AD-762 614

ESTIMATION OF GAS-PHASE THERMOKINETIC PARAMETERS. VOLUME I. A FORTRAN PROGRAM FOR COMPUTING THE THERMOCHEMICAL PROPERTIES OF COMPLEX GAS MOLECULES BY THE METHOD OF GROUP ADDITIVITY. BOOK 2-APPENDIXES

Maria Ramos Martinez

McDonnell Douglas Astronautics Company

Prepared for:

Army Ballistic Research Laboratories

April 1973

BRL

AD

CONTRACT REPORT NO. 107

ESTIMATION OF GAS-PHASE THERMOKINETIC PARAMETERS

VOLUME I

A FORTRAN PROGRAM FOR COMPUTING THE THERMOCHEMICAL PROPERTIES OF COMPLEX GAS MOLECULES BY THE METHOD OF GROUP ADDITIVITY

BOOK 2 - APPENDIXES

Prepared by

McDonnell Douglas Astronautics Company
Huntington Beach, California

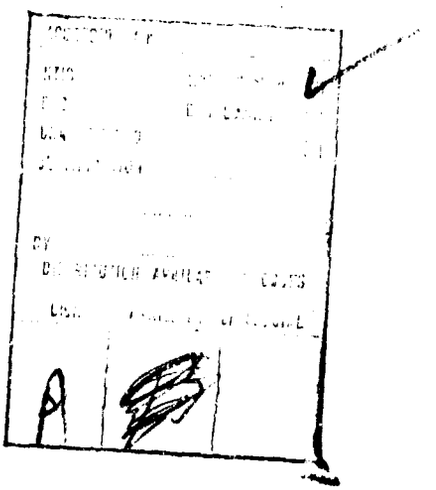
April 1973

USA BALLISTIC RESEARCH LABORATORIES
ABERDEEN PROVING GROUND, MARYLAND

Destroy this report when it is no longer needed.
Do not return it to the originator.

Secondary distribution of this report by originating or sponsoring activity is prohibited.

Additional copies of this report may be obtained from the National Technical Information Service U.S. Department of Commerce, Springfield, Virginia 22151.



The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

CONTRACT REPORT NO. 107

REPORT NO. MDC G4388

ESTIMATION OF GAS-PHASE THERMOKINETIC PARAMETERS

VOLUME I

A FORTRAN PROGRAM FOR COMPUTING THE THERMOCHEMICAL PROPERTIES OF COMPLEX GAS MOLECULES BY THE METHOD OF GROUP ADDITIVITY

BOOK 2 – APPENDIXES

APRIL 1973

BY

DR. MARIA RAMOS MARTINEZ
ADVANCE AERO/THERMODYNAMICS AND NUCLEAR EFFECTS

FOR

BALLISTIC RESEARCH LABORATORIES

PREPARED BY

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY
HUNTINGTON BEACH, CALIFORNIA

PREPARED UNDER CONTRACT NO. DAAD05-72-C-0098

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

McDonnell Douglas Astronautics Company
Huntington Beach, California

2a. REPORT SECURITY CLASSIFICATION

UNCLASSIFIED

2b. GROUP

3. REPORT TITLE

ESTIMATION OF GAS-PHASE THERMOKINETIC PARAMETERS
Volume I - A Fortran Program for Computing the Thermochemical Properties of Complex Gas Molecules by the Method of Group Additivity Book 2-Appendixes

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

Dr. Maria Ramos Martinez

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
APRIL 1973	220	

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
DAAD05-72-C-0098	BRL CONTRACT REPORT NO. 107
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	REPORT NO. MDC G4388

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
BOOK 1 BOOK 2 - APPENDIXES	U.S. Army Ballistic Research Laboratories Aberdeen Proving Ground, Maryland 21005.

13. ABSTRACT

A FORTRAN program for calculating the thermodynamic properties of complex gas molecules by the method of group additivity is documented herein. Included are descriptions of the computational method, characteristics, functions, input/output formats, and logic structure of the program. Seventeen case calculations for diversely structured molecules are presented. Also depicted are a listing of the program, a complete glossary of the program variables and the tables of data comprising the data library input.

10

DD FORM 1 NOV 68 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Chemical Kinetics						

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Book 1

Section 1 INTRODUCTION

Section 2 PRINCIPAL PROPERTIES AND CHARACTERISTICS

- 2.1 Computational Method
- 2.2 Program Functions
- 2.3 Computational Accuracy
- 2.4 Dimension Limits of Principal Variables
- 2.5 Computer Core Requirements
- 2.6 Run Time Requirements

Section 3 INPUT/OUTPUT PROCEDURES

- 3.1 Input Formats
- 3.2 Output Format

Section 4 EXAMPLES OF COMPUTATIONAL OUTPUT

Section 5 LOGIC FRAMEWORK OF PROGRAM

- 5.1 Procedure for Group Identification
- 5.2 Procedures for Chain and Ring Identification
- 5.3 Procedures for Determination of Nongroup Interactions and Symmetry

Section 6 REFERENCES

Book 2

Appendix A TGAP GROUP ADDITIVITY DATA LIBRARY 1

- A.1 Thermochemical Group Contributions 1
- A.2 Corrections for Ring Structures 2
- A.3 Next-Nearest Neighbor Corrections 2

Appendix B FLOW LIST OF TGAP PROGRAM 10

Appendix C GLOSSARY OF PROGRAM VARIABLES 130

TABLES

A-1	Tables of Thermochemical Group Contributors	3
A-2	Corrections for Ring Structures	6
A-3	Next-Nearest Neighbor Corrections	9

Appendix A
TGAP GROUP ADDITIVITY DATA LIBRARY

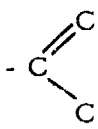
The Group Additivity Data Library is the permanent source of thermochemical data of the TGAP program. These data were derived from Benson's book (1). The data are listed below in three separate tables. A few revisions are included in the tables. These were obtained directly through the courtesy of Dr. Sidney Benson, via telephone communication.

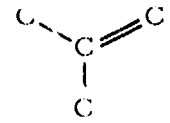
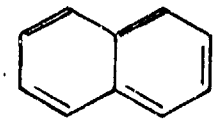
The heat capacity data of Benson has been curve fitted to the equation $C_p = c_0 + c_1 T + c_2 T^2 + c_3 T^4$. The program makes use of these coefficients instead of the tabular heat capacity data to compute the various thermodynamic properties of the molecule. The values of the coefficients are listed in the tables together with the corresponding value for the heat of formation and entropy at 298 K.

A.1 THERMOCHEMICAL GROUP CONTRIBUTIONS

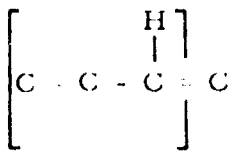
Table A-1 contains the Thermochemical group contributions for hydrocarbons and for oxygen- and nitrogen-containing compounds. The table constitutes a listing of a printout option of the program (See Section 3.1.1).

In the table, the composition of the groups is defined by specific symbols. In addition to the symbols C, H, O and N which have their usual chemical significance other symbols are also used. These are defined below:

Symbol	Definition
D	-C = C
T	-C ≡ C
Z	 (in benzene)

Symbol	Definition
Z1	 central carbon group in fused aromatic rings
	e.g. 
A	C = C = C
N1	-N = C
N2	-N = N

The core atom of the group is always listed first followed by the symbols for the group ligands. For example, DDH represents the group



The thermochemical group values for Z1 were computed by the author. The results were derived by subtracting the sum of eight Z groups from the corresponding thermodynamic property value for gaseous naphthalene and dividing the result by two. The data for naphthalene were obtained from Reference 4 (Book 1).

A.2 CORRECTIONS FOR RING STRUCTURES

Table A-2 contains the nongroup ring corrections for a series of hydrocarbon, oxygen- and nitrogen-containing rings. Saturated and partially unsaturated as well as single and fused ring structures are included. These data are stored in the program per group function. Consequently, as missing data become available, they can be readily incorporated into the program.

A.3 NEXT-NEAREST NEIGHBOR CORRECTIONS

The corrections for next-nearest neighbor interactions are given in Table A-3. The special molecular assignments for $H^{\circ}_{f,298}$ and S°_{298} were described in Section 5.3.3.

Table A-1

TABLES OF THERMOCHEMICAL GROUP CONTRIBUTIONS

GROUP	WEIGHT	HEAT OF FORMATION KCAL/MOLE 298 K	ENTROPY CAL/DEG-MOLE 298 K	HEAT CAPACITY COEFFICIENTS CAL/DEG	CAL/DEG**2-MOLE	CAL/DEG**3-MOLE	CAL/DEG**4-MOLE
DATA FOR HYDROCARBONS							
CHHHC	296	-10.3800	30.4100	-.1679238340	.0244328426	-.0000116714	.0000000022
CHHCC	399	-4.9500	9.4200	-.2553911080	.0228786843	-.0000132553	.0000000030
CHCCC	502	-1.9000	-12.0700	-1.7233179000	.0265408883	-.0000204192	.0000000056
CCCCC	605	.5000	-35.1000	-4.0784320400	.0381256429	-.0000360168	.0000000107
DHH	1031	6.2600	27.6100	.1557868560	.0194481085	-.0000106666	.0000000023
DHC	1134	8.5900	7.9700	.8458673370	.0128226360	-.0000063208	.0000000011
DCC	1237	10.3400	-12.7000	2.2178483200	.0077124957	-.0000049459	.0000000011
DDH	2008	6.7800	6.3800	-1.0380755700	.0238648291	-.0000196512	.0000000058
DDC	2111	6.8800	-14.6000	.3339054170	.0187546888	-.0000182763	.0000000057
DZH	2011	6.7800	6.3800	-1.0380755700	.0238648291	-.0000196512	.0000000058
DZC	2114	8.6400	-14.6000	.3339054170	.0187546888	-.0000182763	.0000000057
DTH	2009	6.7800	6.3800	-1.0380755700	.0238648291	-.0000196512	.0000000058
CDCHH	1273	-4.7600	9.8000	-1.9711603300	.0292831432	-.0000199328	.0000000051
CDDHH	2147	-4.2900	10.2900	-3.7385440300	.0357678715	-.0000269846	.0000000075
CDZHH	2150	-4.2900	10.2900	-3.7385440300	.0357678715	-.0000269846	.0000000079
CTCHH	1274	-4.7300	10.3000	-1.3719938600	.0254749983	-.0000156771	.0000000037
CZCHH	1276	-4.8600	9.3400	-1.2931592200	.0300487840	-.0000221458	.0000000059
CDCCH	1376	-1.4800	-11.6900	-3.6040155200	.0337235204	-.0000282179	.0000000082
CTCCH	1377	-1.7200	-11.1900	-2.8389206600	.0291372022	-.0000228410	.0000000064
CZCCH	1379	-.9800	-12.1500	-2.8510800400	.0341308621	-.0000296252	.0000000087
CDCCC	1479	1.6800	-34.7200	-5.1657888900	.0437931905	-.0000417271	.0000000125
CZCCC	1482	2.6100	-35.1600	-6.3731489100	.0499863257	-.0000501916	.0000000154
TH	1024	26.9300	24.7000	2.5622169400	.0115236646	-.0000087032	.0000000025
TC	1127	27.5500	6.3500	1.4668673000	.0069433901	-.0000054627	.0000000020
TI	2011	29.2000	6.4300	-1.1143259500	.0154355057	-.0000117472	.0000000030
TZ	2014	29.2000	6.4300	-1.1143259500	.0154355057	-.0000117472	.0000000030
ZH	1016	3.3000	11.5300	-1.6891748400	.0201929000	-.0000135394	.0000000034
ZC	1119	5.5100	-7.6500	-.7148721610	.0076700918	-.0000024892	-.0000000001
ZI	1993	5.6800	-7.8000	2.0392057700	.0057575321	-.0000021371	-.0000000000
ZT	1994	5.6800	-7.8000	-2.0392057700	.0057575321	-.0000021371	-.0000000000
ZZ	1996	4.5600	-8.6400	-.7111083760	.0179264165	-.0000159957	.0000000047
A	997	34.2000	6.0300	1.8864443100	.0087392659	-.0000071663	.0000000020
Z1	999	4.8400	-4.6325	-.9445622480	.0128993697	-.0000102398	.0000000029

Table A-1

TABLES OF THERMOCHEMICAL GROUP CONTRIBUTIONS (Continued)

DATA FOR OXYGEN-CONTAINING GROUPS

(CO)(CO)C	1502	-29,2000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
(CO)OC	1855	-33,5000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
(CO)OZ	1858	-46,0000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
(CO)OC	961	-33,4000	14,7800	3,1924243400	,0102961486	-,0000033500	-,0000000008
(CO)OH	878	-29,5000	34,9300	4,6330096300	,0065844185	-,0000060340	-,0000000051
(CO)OH	1717	-31,7000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
(CO)ZZ	2760	-39,1000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
(CO)ZC	1823	-37,6000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
(CO)ZH	1720	-31,7000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
(CO)CC	946	-31,5000	15,0100	3,1089144000	,0071636992	,0000023332	-,0000000032
(CO)CH	843	-29,6000	34,9300	4,6330096300	,0065844185	,0000060340	-,0000000051
(CO)HH	740	-27,7000	53,6700	6,2603870200	,0045815020	,0000112434	-,0000000073
O(CO)(CO)	1524	-50,5000	0,0000	0,0000000000	0,0000000000	0.0000000000	0,0000000000
O(CO)O	1003	-19,0000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
O(CO)O	1842	-41,3000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
O(CO)C	968	-41,3000	8,3900	0,0000100000	0,0000000000	0,0000000000	0,0000000000
O(CO)H	865	-60,3000	24,5200	-1,8852445720	,0261105691	-,0000266194	,0000000102
OCC	447	-4,5000	9,4000	3,9937598500	-,0019631858	,0000034509	-,0000000012
OCO	482	19,0000	9,4000	3,9937598500	-,0019631858	,0000034509	-,0000000012
OCH	344	-16,2700	27,8500	2,4972053200	,0114323353	-,0000091129	,0000000027
OOC	2160	-32,8000	8,1800	0,0000000000	0,0000000000	0,0000000000	0,0000000000
OOC	1286	-31,3000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
OZZ	2156	-19,3000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
OZC	1289	-22,6000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
OZH	1186	-37,9000	29,1100	4,8016326300	-,0052472681	,0000140016	-,0000000070
OCC	412	-23,7000	8,6800	3,7350787000	-,0027931175	,0000073722	-,0000050037
OCH	309	-37,8800	26,0700	5,0581702900	-,0067142502	,0000165871	-,0000000083
O(CO)O	1828	6,3000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
O(CO)C	1793	9,4000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
O(CO)H	1690	7,6800	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
OCO	2146	8,9000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
OCC	1272	10,3000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
OCH	1169	6,4000	8,0000	,8458673370	,0128226360	-,0000063208	,0000000011
Z(CO)	1675	9,7000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
ZC	1154	-1,8000	-10,2000	-4,1497723400	,0406574195	-,0000483862	,0000000191
C(CO)(CO)HH	1511	-7,2000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
C(CO)CC	1058	-1,8000	-12,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
C(CO)CH	955	-5,0000	9,6000	,3596201380	,0251711439	-,0000206641	,0000000073
C(CO)CC	1161	-1,6000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
C(CO)HH	852	-10,0800	30,4100	-,1679238340	,0244328426	-,0000116714	,0000000022
COOCC	675	-16,8000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
COOCH	572	-17,2000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
COOHH	469	-17,7000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
OCZHH	1311	-6,6000	9,7000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
COOHH	1308	-6,9000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
COCCC	640	-6,6000	-33,5600	-6,2148738100	,0516292395	-,0000619198	,0000000248
COCCH	537	-7,0000	-11,3000	-5,3897782000	,0478986890	-,0000525420	,0000000204
COCHH	434	-8,5000	10,0300	-3,4611257600	,0369163436	-,0000324811	,0000000114
COHHH	331	-10,0800	30,4100	-,1758713840	,0244607515	-,0000116904	,0000000022

Table A-1

TABLES OF THERMOCHEMICAL GROUP CONTRIBUTIONS (Continued)

DATA FOR NITROGEN-CONTAINING GROUPS

C(CN)CHH	679	22,5000	40,2000	1,5936711300	,0380737392	-,0000228626	,0000000045
C(CN)CCH	782	25,8000	19,8000	4,4547547800	,0259552780	-,0000145811	,0000000028
C(CN)CCC	865	0,0000	-2,8000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
C(CN)(CN)CC	1165	0,0000	28,4000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
D(CN)H	1414	37,4000	36,5800	1,6278912400	,0341575604	-,0000250052	,0000000065
D(CN)(CN)	1747	84,1000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
D(ONO)H	1247	0,0000	44,4000	1,3059065240	,0294788384	-,0000378314	,0000000102
T(CN)	1397	63,8000	35,4000	4,3208018800	,0166236963	-,0000118637	,0000000032
C(ONO)CHH	512	-15,1000	48,4000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
C(ONO)CCH	615	-15,8000	26,9000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
C(ONO)CCC	718	0,0000	3,9000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
C(ONO)(ONO)CH	728	-14,9000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
O(NO)C	508	-5,9000	41,9000	4,3909442100	,0191590634	-,0000122788	,0000000027
O(ONO)C	525	-19,4000	48,5000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
CNHHH	712	-10,0800	30,4100	-1,1679238340	,0244328424	-,0000116714	,0000000022
CNCHH	815	-6,6000	9,8000	-1,8919676100	,0301279382	-,0000233047	,0000000074
CNCCH	918	-5,2000	-11,7000	-3,7379120600	,0382564597	-,0000382709	,0000000140
CNCCC	1021	-3,2000	-34,1000	-5,6117685700	,0475192186	-,0000538081	,0000000201
NCHH	699	4,8000	29,7100	2,8959473800	,0100598791	-,0000023272	-,0000000001
NCCH	802	15,4000	6,9400	1,0095954513	,0170255903	-,0000109772	,0000000026
NCCC	905	24,4000	-13,4600	-1,8110294200	,0233777734	-,0000208418	,0000000060
NNHH	1115	11,4000	29,1300	1,9759068520	,0211752446	-,0000146100	,0000000039
NNCH	1218	20,9000	9,6100	1,9568461210	,0164883897	-,0000126832	,0000000035
NNCC	1321	29,2000	-13,8000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
NNZH	2095	22,1000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
N1H	479	0,0000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
N1C	583	21,3000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
N1Z	1459	16,7000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
N2H	527	25,1000	26,8000	2,4119114000	,0071523525	-,0000022180	,0000000001
N2C	630	32,5000	-8,0400	0,0000000000	0,0000000000	0,0000000000	0,0000000000
NZHH	1576	4,8000	29,7100	2,8959473800	,0100598791	-,0000023272	-,0000000001
NZCH	1679	17,9000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
NZCC	1782	26,2000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
NZZH	2596	16,3000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
ZN	1535	-,5000	-9,6900	-3,5715753900	,0369462129	-,0000448672	,0000000181
N2N	1646	23,0000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
(CO)NH	1259	-29,6000	34,9300	4,6330096300	,0065844185	-,0000060340	-,0000000051
(CO)NC	1362	-32,8000	16,2000	3,6876785500	,0038940571	-,0000070014	-,0000000034
N(CO)NH	1255	-14,9000	24,6900	-3,2529205200	,0309737873	-,0000242737	,0000000078
N(CO)CH	1356	-4,4000	3,9000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
N(CO)CC	1461	0,0000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
N(CO)ZH	2235	,4000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
N(CO)(CO)H	1914	-18,5000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
N(CO)(CO)C	2017	-5,9000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000
N(CO)(CO)Z	2894	-,5000	0,0000	0,0000000000	0,0000000000	0,0000000000	0,0000000000

Table A-2

CORRECTIONS FOR RING STRUCTURES

Ring	$\Delta H^{o}_{f,298}$ kcal/mole	S^{o}_{298} cal/deg-mole	C^{o}_{p} cal/deg-mole	Coefficients cal/deg^2-mole	cal/deg^3-mole	cal/deg^4-mole
Cyclopropane	27.6	32.1	-5.5952	1.1577×10^{-2}	-1.115×10^{-5}	3.50×10^{-9}
Cyclopropene	53.7	33.6	-			
Cyclobutane	26.2	29.8	-7.9609	1.3812×10^{-2}	-9.772×10^{-6}	2.54×10^{-9}
Cyclobutene	29.8	29.0	-4.1570	7.0499×10^{-3}	-5.930×10^{-6}	1.69×10^{-9}
Cyclopentane	6.3	27.3	-1.2010	1.7322×10^{-2}	-8.845×10^{-6}	1.65×10^{-9}
Cyclopentene	5.9	25.8	-7.9649	6.4757×10^{-3}	4.718×10^{-7}	-1.15×10^{-9}
Cyclopentadiene	6.0					
Cyclohexane	0	18.8				
Cyclohexene	1.4	21.5				
1, 3 Cyclohexadiene	4.8					
1, 4 Cyclohexadiene	0.5					
Cycloheptane	6.4	15.9				
Cycloheptene	5.4					
1, 3 Cycloheptadiene	6.6					
1, 3, 5 Cycloheptatriene	4.7	23.7				
Cyclooctane	9.9	16.5				
Cis-cyclooctene	6.0					
Trans-cyclooctene	15.3					
1, 3, 5 Cyclooctatriene	8.9					
Cyclooctatetraene	17.1					
Cyclononane	12.8					
Cis-cyclononene	9.9					
Trans-cyclononene	12.8					
Spiropentane	63.5	67.6				
Bicyclo-(1, 1, 0)-butane	68.4	69.2				
Bicyclo-(2, 1, 0)-pentane	55.3					
Bicyclo-(3, 1, 0)-hexane	32.7					
Bicyclo-(4, 1, 0)-heptane	28.9					
Bicyclo-(5, 1, 0)-octane	29.6					
Bicyclo-(6, 1, 0)-nonane	31.1					
	27.6	31.4	4.5377	-3.6617×10^{-2}	5.709×10^{-5}	-2.73×10^{-8}
	26.4	27.7	-5.3154	2.4026×10^{-3}	-4.325×10^{-6}	7.39×10^{-9}
	6.7					

Table A-2

CORRECTIONS FOR RING STRUCTURES (Continued)

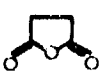
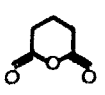
Ring	$\Delta H^{o}_{f,298}$ kcal/mole	S^{o}_{298} cal/deg-mole	C^{o}_{p} cal/deg-mole	Coefficients cal/deg^2-mole	cal/deg^3-mole	cal/deg^4-mole
	2.2					
	3.5					
	5.4					
	3.4					
	-6.2					
	2.5					
	6.0					
	3.4					
	1.1					
	1.4					
	4.6					
	27.7	31.6				
	26.2	29.3				

Table A-2
CORRECTIONS FOR RING STRUCTURES (Continued)

Ring	$\Delta H^{o}_{f,298}$ kcal/mole	S^{o}_{298} cal/deg-mole	C^{o}_{p} cal/deg-mole	Coefficients cal/deg^{2}-mole	cal/deg^{3}-mole	cal/deg^{4}-mole
	6.8	26.7	-7.4924	1.4742×10^{-3}	1.186×10^{-5}	-8.02×10^{-9}
	1.0					
	3.4					
	8.5					

Table A-3
NEXT-NEAREST NEIGHBOR CORRECTIONS

Interaction	$H^o_{f,298}$ kcal/mole	S^o_{298} cal/deg-mole	C^o_p cal/deg-mole	Coefficients cal/deg^2-mole	cal/deg^3-mole	cal/deg^4-mole
Alkane gauche	0.80					
Alkene gauche	0.50					
Ether oxygen gauche	0.3					
Di-tertiary ethers	8.4					
Cis	1.00 (a)	0 (a)	-2.7198	5.8531×10^{-3}	-4.785×10^{-6}	1.39×10^{-9}
Ortho (benzene)	0.57	-1.61	3.6765×10^{-1}	4.4047×10^{-3}	-6.267×10^{-6}	2.10×10^{-9}
Ortho/para(pyridine)	-1.5					

(a) See Table 5-4 for exceptions.

Appendix B
FLOW LIST OF TGAP PROGRAM

The flow list of the TGAP program and the program glossary presented in the following section were generated by the McDonnell Douglas JOYCE Automated Documentation System. The flow list comprises a listing of the source program bounded by brackets or arrows on th 'eft and right margins. The brackets on the left margin delineate the cycles or DO loops of the program. The brackets on the right margin delineate all transfers, both conditional and unconditional. The numbers of the statements to which the transfers are made are also depicted. The flow list of the main routine, TGAP, is presented first followed by the flowlists of the 54 subroutines of the program listed in alphabetical order. The names of the routines and their respection locations in the text are presented below.

	Routine	Page
1	TGAP	12
2	ASSIGN	16
3	ASYMC	18
4	BOND	19
5	CHAINM	20
6	CHANGE	22
7	CIS	23
8	CISCOR	25
9	CORCIG	28
10	CRINGS	34
11	CTWO	36
12	CYCORR	38
13	DATA1	45
14	DELETE	50
15	DELTA1	51
16	DELTA2	52
17	DITERE	53
18	ENTSYM	54
19	EQUAL	56
20	EQUALR	60
21	EXTROT	65

	Routine	Page
22	FIND	70
23	FIRSTR	71
24	FUSION	73
25	GADATA	76
26	GAUCHE	77
27	HEXGON	80
28	IDENT	85
29	INTROT	88
30	LESSEN	89
31	LINEAR	92
32	MAXCHN	93
33	MULTI	95
34	NEWCOL	96
35	NEWKC	97
36	NRINGS	98
37	NUMBER	99
38	ORDER	100
39	OXYATM	101
40	PRINT1	102
41	PRINT2	103
42	RESETR	104
43	RING	105
44	SAME	106
45	SCAN	107
46	SCANBR	109
47	SCANCH	110
48	SEARCH	111
49	SETUP	112
50	SHIFT	114
51	SORNGI	115
52	STAND	117
53	SUMATM	121
54	SYMRNG	122
55	SYMTRY	127

```
      PROGRAM TGAP(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)                    TGAP
C       THIS IS THE MAIN OR CONTROL UNIT OF THE THERMOCHEMICAL GROUP         TGAP
C       ADDITIVITY PROGRAM (TGAP). IT PRINTS OUT THE DATA LIBRARY            TGAP
C       (OPTIONAL), READS AND PRINTS OUT INPUT DATA, TRANSFERS CONTROL       TGAP
C       TO THREE MAIN SECTIONS OF THE PROGRAM, AND COMPUTES AND PRINTS       TGAP
C       THE THERMOCHEMICAL PROPERTIES OF THE MOLECULE.                       TGAP
      INTEGER ENDCS1,ENDCS2,ENDCS3,ENDRUN,CASE,BLANK,ALTER(20,2),            TGAP
     1ORDSUM(180),SEARCH                                                     TGAP
      INTEGER SYMX(4),SYMBOL(9),GRID(50,80)                                  TGAP
      INTEGER WEIGHT(9)                                                      TGAP
      INTEGER SUM(180)                                                       TGAP
      DIMENSION TARRAY(14),CPT(14),ST(14),HST(14),FT(14),HSOT(14),           TGAP
     1HTT(14),CP(4),CPSYM(4),KGRID(103),LABEL(50)                            TGAP
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                           TGAP
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC          TGAP
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),       TGAP
     1IOBR(100),IB(100,8),IRG,NOBR                                           TGAP
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT          TGAP
      COMMON/BLK5/NDATA,NUMATM(5),MBC(50),MBS(2),JW,JY,LFLAG5,LFLAG6         TGAP
      COMMON/BLK6/NUMFRQ(24)                                                 TGAP
      COMMON/BLK7/SUM,HF298(180),S298(180),CPX(180,4)                        TGAP
      EQUIVALENCE (CPT(1),IMATX(5,1)),(ST(1),IMATX(19,1)),(HST(1),           TGAP
     1IMATX(33,1)),(FT(1),IMATX(47,1)),(HSOT(1),IMATX(11,2)),(HTT(1),        TGAP
     1IMATX(25,2)),(KGRID(1),NBC(1,7)),(LABEL(1),NBC(1,49))                  TGAP
      DATA (SYMX(I),I=1,4)/1H-,1H=,1H ,1H /                                  TGAP
      DATA NDATM/ 5/,NOS/ 9/,(SYMBOL(I),WEIGHT(I),I=1,9) /1HH,   1,          TGAP
     1HC,  12,1HO,  156,1HN,  537,1HE,  998,2HCO,  677,2HCN,  401,          TGAP
     22HNO,  217,3HNO2,  234/                                                TGAP
      DATA (MWGT(I),I=1, 9)/0,0,14,5,0,27,0,0,0/                             TGAP
      DATA (NUMFRQ(I),I=1,24)/2,1,3,1,1,0,2,1,4,1,1,0,4,1,3,1,1,0,          TGAP
     14,1,3,2,1,0/                                                           TGAP
      DATA NOVAL/1,4,2,3,1,2,1,1,1/                                          TGAP
      DATA ALTER(1,1)/1170/,ALTER(1,2)/296/,ALTER(2,1)/2985/,ALTER(2,2)      TGAP
     1/2111/,ALTER(3,1)/1171/,ALTER(3,2)/296/,ALTER(4,1)/1173/,              TGAP
     2ALTER(4,2)/296/,NAL/4/                                                 TGAP
      DATA TARRAY/298.0,300.0,400.0,500.0,600.0,700.0,800.0,900.0,           TGAP
     11000.0,1200.0,1400.0,1600.0,1800.0,2000.0/                             TGAP
      DATA ENDCS1/1H*/,ENDCS2/1H*/,ENDCS3/1H*/,ENDRUN/3HEND/,BLANK/1H /,     TGAP
     1ISUM/180*0/                                                            TGAP
C1.0     INITIALIZE VARIABLE.                                                TGAP
      CASE=0                                                                 TGAP
C2.0     READ IN DATA.                                                       TGAP
C          READ IN PRINTOUT OPTIONS (IPRINT) AS WELL AS THE                  TGAP
C          OPTIONAL VARIABLES COMPRISING THE EXTERNAL SYMMETRY NUMBER        TGAP
C          (NOSN), THE NUMBER OF ENANTIOMERS (NENAN), AND THE NUMBER OF      TGAP
C          MESO STRUCTURES (MESO). A MOLECULE IDENTIFICATION LABEL           TGAP
C          (LABEL) MAY BE INPUT IF DESIRED.                                  TGAP
    1 READ(5,2)IEND,IPRINT,NOSN,NENAN,MESO,LABEL                             TGAP
    2 FORMAT(A3,4I3,15X,50A1)                                                TGAP
C           IF END CARD PRESENT, STOP.                                       TGAP
      IF(IEND.EQ.ENDRUN)STOP                                                 TGAP
      NO=0                                                                   TGAP
C         READ IN GRAPHIC STRUCTURE OF MOLECULE.                             TGAP
    3 NO=NO+1                                                                TGAP
      READ(5,4)(GRID(NO,L),L=1,80)                                           TGAP
    4 FORMAT(80A1)                                                           TGAP
      IF(GRID(NO,1).NE.ENDCS1.OR.GRID(NO,2).NE.ENDCS2.OR.GRID(NO,3).NE.      TGAP
     1ENDCS3)GO TO 3                                                         TGAP
C           END OF CASE DATA INPUT CARD REACHED. DELETE LAST CARD            TGAP
C           FROM NO COUNTER AND INCREMENT THE VARIABLE CASE.                 TGAP
      NO=NO-1                                                                TGAP
      IF(NO.LE.0)GO TO 1                                                     TGAP
      CASE=CASE+1                                                            TGAP
C3.0     PRINT OUT GROUP ADDITIVITY DATA TABLES ON FIRST PASS PROVIDING      TGAP
C        JPRINT IS NON-ZERO.                                                 TGAP
      JPRINT=IPRINT/10                                                       TGAP
      JPRINT=IPRINT-10*JPRINT                                                TGAP
      IF(CASE.NE.1)GO TO 498                                                 TGAP
      KPRINT=IPRINT/100                                                      TGAP
      KPRINT=(IPRINT-100*KPRINT)/10                                          TGAP
  498 IF(CASE.EQ.1.AND.JPRINT.NE.0)CALL GADATA                               TGAP
C9.0     HEADING PRINTOUT OPTIONS.                                           TGAP
      WRITE(6,5)                                                             TGAP
    5 FORMAT(1H1)                                                            TGAP
```

17 APR 73 8.02-36

```
      IF(KPRINT.NE.0.AND.KPRINT.NE.2)GO TO 502                          TGAP
C           PRINT OUT TITLE.                                            TGAP
      WRITE(6,500)                                                      TGAP
  500 FORMAT(1H ,37X,54HGROUP  ADDITIVITY  THERMOCHEMICAL  PROPERTIES  P TGAP
     1ROGRAM     //)                                                   TGAP
  502 IF(KPRINT.NE.0.AND.KPRINT.NE.3)GO TO 506                          TGAP
C           PRINT OUT CASE NUMBER.                                      TGAP
      WRITE(6,504)CASE                                                  TGAP
  504 FORMAT(1H ,59X,12HCASE  NUMBER,I6//)                              TGAP
  506 WRITE(6,508)                                                      TGAP
  508 FORMAT(1H ,54X,20HMOLECULAR  STRUCTURE  //)                       TGAP
C5.0      CENTER STRUCTURAL DATA FOR PRINTOUT PURPOSES.                 TGAP
C           FIND LEFT(MBL) AND RIGHT(MBR) BOUNDARIES OF ARRAY LABEL.    TGAP
      MBL=50                                                            TGAP
      MBR=1                                                             TGAP
      DO602   L=1,50                                                    TGAP
      IF(LABEL(L).EQ.BLANK)GO TO 602                                    TGAP
      IF(L.GE.MBL)GO TO 600                                             TGAP
      MBL=L                                                             TGAP
  600 IF(L.LE.MBR)GO TO 602                                             TGAP
      MBR=L                                                             TGAP
  602 CONTINUE                                                          TGAP
C           FIND INITIAL LOCATION IN KGRID INTO WHICH COLUMN OF DATA    TGAP
C           FROM ARRAY LABEL IS TO BE TRANSFERRED.                      TGAP
C         BLANK OUT KGRID ARRAY AND DETERMINE IF IDENTIFICATION NAME    TGAP
C         WAS INPUT.                                                    TGAP
      DO606   L=1,103                                                   TGAP
  606 KGRID(L)=BLANK                                                    TGAP
      IF(MBL.EQ.50.AND.MBR.EQ.1)GO TO 616                               TGAP
      LL=64-(MBR-MBL+1)/2                                               TGAP
C           TRANSFER DATA FROM LABEL INTO DESIGNATED LOCATIONS OF KGRID TGAP
C           AND PRINT OUT.                                              TGAP
      DO610   L=MBL,MBR                                                 TGAP
      KGRID(LL)=LABEL(L)                                                TGAP
  610 LL=LL+1                                                           TGAP
      WRITE(6,612)(KGRID(L),L=1,88)                                     TGAP
  612 FORMAT(1H ,88A1//)                                                TGAP
C           FIND LEFT(MBL) AND RIGHT(MBR) GRID BOUNDARIES.              TGAP
  616 MBL=80                                                            TGAP
      MBR=1                                                             TGAP
      DO8   K=1,NO                                                      TGAP
      DO8   L=1,80                                                      TGAP
      IF(GRID(K,L).EQ.BLANK)GO TO 8                                     TGAP
      IF(L.GE.MBL)GO TO 7                                               TGAP
      MBL=L                                                             TGAP
    7 IF(L.LE.MBR)GO TO 8                                               TGAP
      MBR=L                                                             TGAP
    8 CONTINUE                                                          TGAP
C           FIND INITIAL LOCATION IN KGRID INTO WHICH COLUMN OF DATA FROM TGAP
C           ARRAY GRID IS TO BE TRANSFERRED AND BLANK OUT KGRID ARRAY.  TGAP
      LS=64-(MBR-MBL+1)/2                                               TGAP
      DO9   L=1,103                                                     TGAP
    9 KGRID(L)=BLANK                                                    TGAP
C           TRANSFER COLUMN OF DATA FROM GRID INTO DESIGNATED LOCATIONS  TGAP
C           OF KGRID AND PRINT OUT.                                     TGAP
      DO12    K=1,NO                                                    TGAP
      LL=LS                                                             TGAP
      DO10    L=MBL,MBR                                                 TGAP
      KGRID(LL)=GRID(K,L)                                               TGAP
   10 LL=LL+1                                                           TGAP
      WRITE(6,11)(KGRID(L),L=1,103)                                     TGAP
   11 FORMAT(1H ,103A1)                                                 TGAP
   12 CONTINUE                                                          TGAP
C6.0      TRANSFERS CONTROL TO SECTION ONE OF THE PROGRAM WHICH         TGAP
C           IDENTIFIES THE GROUPS AND THEIR STRUCTURAL AND WEIGHT DATA. TGAP
      IERR=0                                                            TGAP
      CALL STAND(IERR)                                                  TGAP
      IF(IERR)13,14,13                                                  TGAP
C           ERROR EXISTS. TRANSFER TO PRINTOUT ROUTINE AND EXIT FROM    TGAP
C           CASE CALCULATION.                                           TGAP
   13 CALL PRINT1                                                       TGAP
```

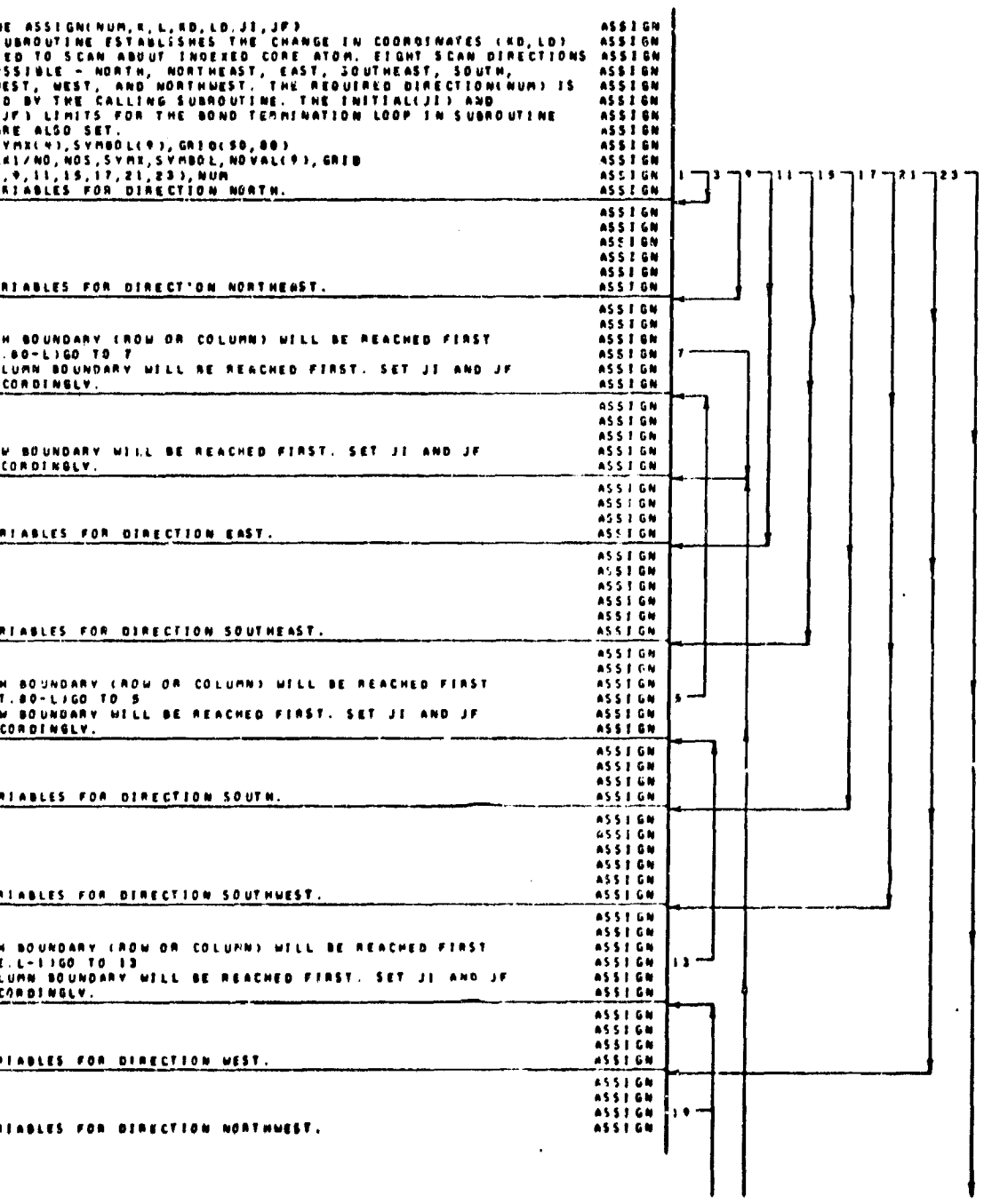
17 APR 73 8.02-36

```
146.            GO TO 1                                                  TGAP
147.   C7.0     TRANSFERS CONTROL TO SECTION TWO OF THE PROGRAM WHICH FINDS THE TGAP
148.   C        CHAIN SEGMENTS AND UNIQUE RINGS (IF ANY).                TGAP
149.        14 CALL CHAINO(LX,IERR)                                      TGAP
150.   C          IF ERROR EXISTS TRANSFER TO PRINTOUT ROUTINE AND EXIT FROM TGAP
151.   C          CASE CALCULATION.                                      TGAP
152.           IF(IERR)13,15,13                                         TGAP
153.   C8.0     TRANSFER CONTROL TO SECTION THREE OF THE PROGRAM WHICH FINDS TGAP
154.   C        ALL SECOND-ORDER AND RING CORRECTIONS AS WELL AS CONTRIBUTIONS TGAP
155.   C        DUE TO INTERNAL AND EXTERNAL ROTATIONAL SYMMETRY AND OPTICAL TGAP
156.   C        ISOMERISM.                                               TGAP
157.        15 CALL CORCIG(NOSN,NENAN,MESO,LX,HSYM,SSYM,CPSYM)          TGAP
158.   C9.0     COMPUTE AND PRINT OUT THERMOCHEMICAL PROPERTIES OF MOLECULE. TGAP
159.   C          TRANSFER THERMOCHEMICAL CONTRIBUTIONS DERIVED FROM SECTION TGAP
160.   C          THREE TO CORRESPONDING VARIABLES DELH, S, AND CP.      TGAP
161.           DELH=HSYM                                                TGAP
162.           S=SSYM                                                   TGAP
163.           DO21   J=1,9                                             TGAP
164.        21 CP(J)=CPSYM(J)                                           TGAP
165.   C          FIND ORDER MAX ELEMENTS OF ARRAY SUM WOULD HAVE IF ARRANGED TGAP
166.   C          IN NUMERICAL ORDER AND STORE SAID ORDER (RELATIVE TO INITIAL TGAP
167.   C          ORDER) IN ORDSUM.                                      TGAP
168.           MAX=163                                                  TGAP
169.           CALL ORDER(MAX,ORDSUM,SUM)                               TGAP
170.   C          FIND LOCATION IN GROUP ADDITIVITY DATA ARRAYS CONTAINING THE TGAP
171.   C          DATA CORRESPONDING TO THE COMPUTED WEIGHT MOLWT(K) OF GROUP K TGAP
172.           LFLAGG=0                                                 TGAP
173.           DO69   K=1,KCC                                           TGAP
174.           ISER=SEARCH(SUM,ORDSUM,163,MOLWT(K),IT)                  TGAP
175.   C          WAS WEIGHT MATCH ACHIEVED                            TGAP
176.           IF(ISER.NE.0)GO TO 25                                    TGAP
177.           IF(IT.GT.0.AND.IT.LE.163)GO TO 67                        TGAP
178.   C          NO. DOES AN ALTERNATE DATA GROUP WEIGHT EXIST FOR GROUP K TGAP
179.        25 DO33   L=1,NAL                                           TGAP
180.           IF(MOLWT(K).NE.ALTER(L,1))GO TO 33                       TGAP
181.   C          YES IT DOES. FIND ITS LOCATION IN DATA ARRAYS. OTHERWISE TGAP
182.   C          PRINT OUT MESSAGE INDICATING DATA ARE MISSING.       TGAP
183.           ISER=SEARCH(SUM,ORDSUM,163,ALTER(L,2),IT)                TGAP
184.           IF(ISER)63,67,63                                         TGAP
185.        33 CONTINUE                                                 TGAP
186.        63 WRITE(6,65)K,MOLWT(K)                                    TGAP
187.        65 FORMAT(1HO,26X,14HDATA FOR GROUP,I4,18H  WITH A WEIGHT OF,I6, TGAP
188.          13SH  ARE NOT IN PROGRAM DATA LIBRARY.)                   TGAP
189.           LFLAGG=1                                                 TGAP
190.   C          MOST LIKELY, A MATCH WAS ACHIEVED. NOW ADD THE THERMO- TGAP
191.   C          CHEMICAL DATA FOR GROUP K TO DELH, S, AND CP.         TGAP
192.        67 DELH=DELH+HF298(IT)                                      TGAP
193.           S=S+S298(IT)                                             TGAP
194.           DO68   J=1,9                                             TGAP
195.        68 CP(J)=CP(J)+CPX(IT,J)                                    TGAP
196.        69 CONTINUE                                                 TGAP
197.   C          DETERMINE IF THERMOCHEMICAL DATA OUTPUT IS COMPLETE. IF NOT, TGAP
198.   C          PRINT OUT WARNING MESSAGE.                           TGAP
199.           IF(LFLAGG.EQ.0)GO TO 75                                  TGAP
200.           WRITE(6,73)                                              TGAP
201.        73 FORMAT(// 1HO,  1X,126HWARNING --- ALL OF THE THERMOCHEMICAL VALUE TGAP
202.          1S FOR ONE OR MORE GROUPS ARE MISSING. THE THERMOCHEMICAL DATA BELO TGAP
203.          2W ARE INCOMPLETE.)                                       TGAP
204.        75 IF(LFLAGS.EQ.0)GO TO 83                                  TGAP
205.           WRITE(6,77)                                              TGAP
206.        77 FORMAT(// 1HO,21X,87HWARNING --- ENTROPY AND FREE ENERGY DATA BELO TGAP
207.          1W MAY BE OFF A FEW PERCENT DUE TO POSSIBLE MIN ,27X,75HCONTRIBUTIO TGAP
208.          2NS) FROM EXTERNAL ROTATIONAL SYMMETRY OR AND OPTICAL ISOMERISM.) TGAP
209.   C          PRINT OUT IDEAL HEAT CAPACITY COEFFICIENTS OF MOLECULE. ALSO TGAP
210.   C          COMPUTE AND PRINT OUT THE HEAT CAPACITY, ENTROPY, ENTHALPY, TGAP
211.   C          GIBBS FREE ENERGY, H-H(298), AND TOTAL ENTHALPY FOR   TGAP
212.   C          TEMPERATURES SPECIFIED IN ARRAY TARRAY.               TGAP
213.        83 HCONS=((((CP(4)*298.0/4.0+CP(3))/3.0)*298.0+CP(2)/2.0)*298.0* TGAP
214.          1CP(1))*298.0)/1000.0                                     TGAP
215.           SCONS=S-CP(1)*ALOG(298.0)-(((CP(4)*298.0/3.0+CP(3)/2.0)*298.0* TGAP
216.          1CP(2))*298.0                                             TGAP
```

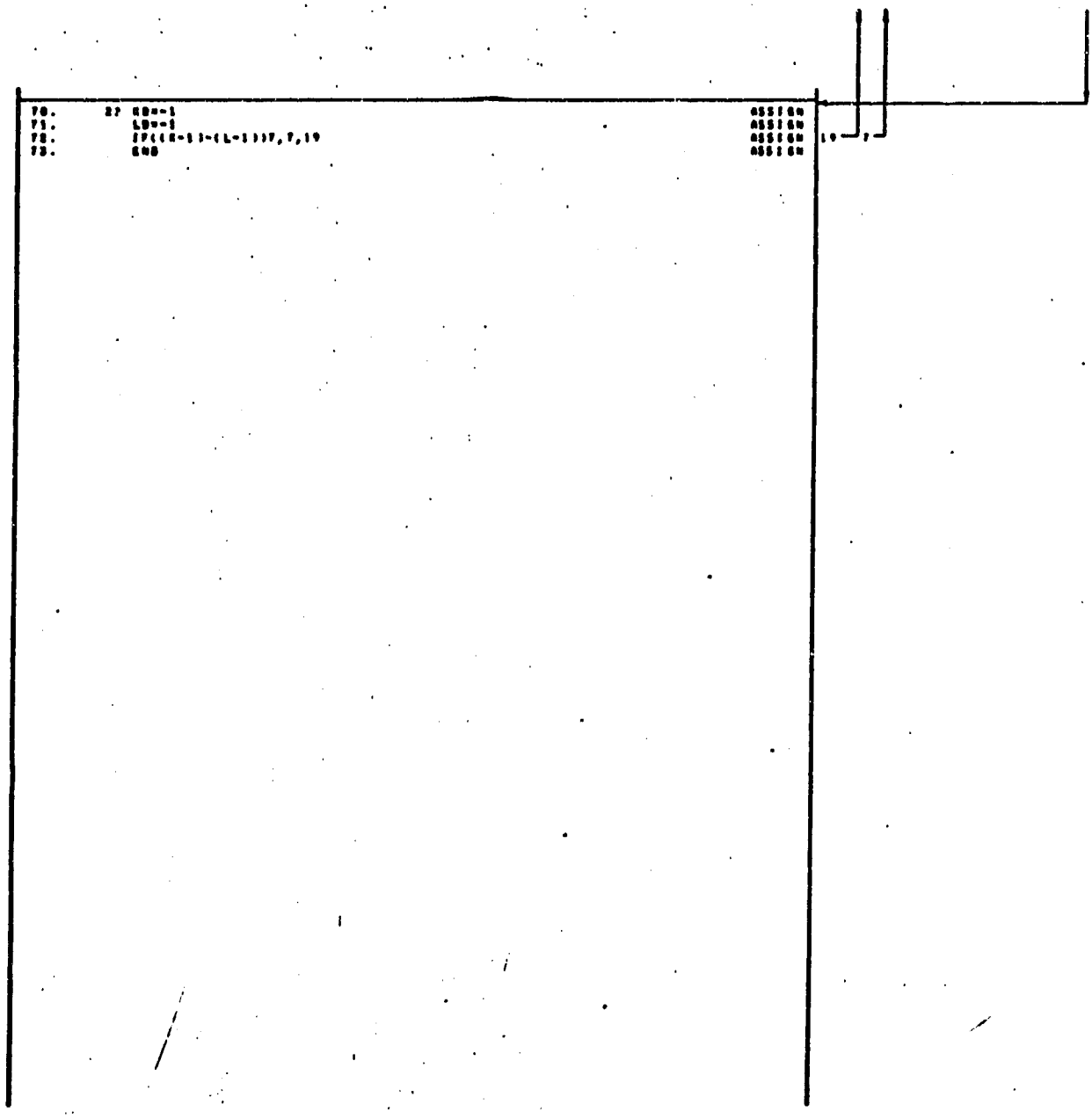
```
      DOBS    J=1,19                                                        TGAP
      CPT(J)=CP(1)+((CP(4)*TARRAY(J)+CP(3))*TARRAY(J)+CP(2))*TARRAY(J)      TGAP
      ST(J)=CP(1)*ALOG(TARRAY(J))+((CP(4)*TARRAY(J)/3.0+CP(3)/2.0)          TGAP
     1*TARRAY(J)+CP(2))*TARRAY(J)+SCONS                                     TGAP
      HST(J)=(((CP(4)*TARRAY(J)/4.0+CP(3)/3.0)*TARRAY(J)+CP(2)/2.0)*       TGAP
     1TARRAY(J)+CP(1))*TARRAY(J))/1000.0                                    TGAP
      HSDT(J)=HST(J)-HCONS                                                  TGAP
      FT(J)=-(HSDT(J)*1000.0-TARRAY(J)*ST(J))/TARRAY(J)                     TGAP
      HTT(J)=HSDT(J)+DELH                                                   TGAP
   85 CONTINUE                                                              TGAP
      WRITE(6,87)(CP(J),J=1,4)                                              TGAP
   87 FORMAT(///1H ,44X,40HTHERMOCHEMICAL  PROPERTIES  OF  MOLECULE  //     TGAP
     11H ,51X,26HHEAT CAPACITY COEFFICIENTS  /1H ,28X,12HCAL/DEG-MOLE,      TGAP
     25X,15HCAL/DEG**2-MOLE,5X,15HCAL/DEG**3-MOLE,5X,15HCAL/DEG**4-MOLE/   TGAP
     31H ,22X,2E19.8,2E20.8)                                                TGAP
      WRITE(6,88)                                                           TGAP
   88 FORMAT(//1H ,13X,1HT,11X,9HC(P),14X,1HS,11X,9HH(T)-H(0),3X,           TGAP
     116H-(G(T)-H(298))/T,5X,11HH(T)-H(298),3X,22HDELHF(298)+H(T)-H(298)    TGAP
     2/1H ,11X,5HDEG K,5X,12HCAL/DEG-MOLE,5X,12HCAL/DEG-MOLE,5X,            TGAP
     39HKCAL/MOLE,5X,12HCAL/DEG-MOLE,6X,9HKCAL/MOLE,11X,9HKCAL/MOLE)        TGAP
      WRITE(6,89)(TARRAY(J),CPT(J),ST(J),HST(J),FT(J),HSDT(J),HTT(J),       TGAP
     1J=1,19)                                                               TGAP
   89 FORMAT(1H , 8X,F8.1,5F16.6,F20.6)                                     TGAP
      GO TO 1                                                               TGAP
      END                                                                   TGAP
```

17 APR 73 8.02-36

```
  1.            SUBROUTINE ASSIGN(NUM,K,L,KD,LD,JI,JF)                    ASSIGN
  2.    C          THIS SUBROUTINE ESTABLISHES THE CHANGE IN COORDINATES (KD,LD)  ASSIGN
  3.    C          REQUIRED TO SCAN ABOUT INDEXED CORE ATOM. EIGHT SCAN DIRECTIONS ASSIGN
  4.    C          ARE POSSIBLE - NORTH, NORTHEAST, EAST, SOUTHEAST, SOUTH,   ASSIGN
  5.    C          SOUTHWEST, WEST, AND NORTHWEST. THE REQUIRED DIRECTION(NUM) IS  ASSIGN
  6.    C          DEFINED BY THE CALLING SUBROUTINE. THE INITIAL(JI) AND     ASSIGN
  7.    C          FINAL(JF) LIMITS FOR THE BOND TERMINATION LOOP IN SUBROUTINE  ASSIGN
  8.    C          BOND ARE ALSO SET.                                        ASSIGN
  9.            INTEGER SYMX(4),SYMBOL(9),GRID(50,80)                      ASSIGN
 10.            COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID               ASSIGN
 11.            GO TO(1,3,9,11,15,17,21,23),NUM                           ASSIGN
 12.    C1.0       SET VARIABLES FOR DIRECTION NORTH.                     ASSIGN
 13.          1 KD=-1                                                     ASSIGN
 14.            LD=0                                                      ASSIGN
 15.            JI=2                                                      ASSIGN
 16.            JF=K-1                                                    ASSIGN
 17.            RETURN                                                    ASSIGN
 18.    C2.0       SET VARIABLES FOR DIRECTION NORTHEAST.                 ASSIGN
 19.          3 KD=-1                                                     ASSIGN
 20.            LD=1                                                      ASSIGN
 21.    C           WHICH BOUNDARY (ROW OR COLUMN) WILL BE REACHED FIRST   ASSIGN
 22.            IF(K-1.LE.80-L)GO TO 7                                    ASSIGN
 23.    C              COLUMN BOUNDARY WILL BE REACHED FIRST. SET JI AND JF  ASSIGN
 24.    C              ACCORDINGLY.                                        ASSIGN
 25.          5 JI=2                                                      ASSIGN
 26.            JF=80-L                                                   ASSIGN
 27             RETURN                                                    ASSIGN
 28.    C              ROW BOUNDARY WILL BE REACHED FIRST. SET JI AND JF   ASSIGN
 29.    C              ACCORDINGLY.                                        ASSIGN
 30           7 JI=2                                                      ASSIGN
 31             JF=K-1                                                    ASSIGN
 32             RETURN                                                    ASSIGN
 33    C3.0       SET VARIABLES FOR DIRECTION EAST.                      ASSIGN
 34.          9 KD=0                                                      ASSIGN
 35.            LD=1                                                      ASSIGN
 36.            JI=L+2                                                    ASSIGN
 37.            JF=80                                                     ASSIGN
 38             RETURN                                                    ASSIGN
 39.    C4.0       SET VARIABLES FOR DIRECTION SOUTHEAST.                 ASSIGN
 40.         11 KD=1                                                      ASSIGN
 41             LD=1                                                      ASSIGN
 42.    C           WHICH BOUNDARY (ROW OR COLUMN) WILL BE REACHED FIRST   ASSIGN
 43             IF(NO-K.GT.80-L)GO TO 5                                   ASSIGN
 44.    C              ROW BOUNDARY WILL BE REACHED FIRST. SET JI AND JF   ASSIGN
 45.    C              ACCORDINGLY.                                        ASSIGN
 46.         13 JI=2                                                      ASSIGN
 47.            JF=NO-K                                                   ASSIGN
 48.            RETURN                                                    ASSIGN
 49.    C5.0       SET VARIABLES FOR DIRECTION SOUTH.                     ASSIGN
 50.         15 KD=1                                                      ASSIGN
 51.            LD=0                                                      ASSIGN
 52.            JI=K+2                                                    ASSIGN
 53.            JF=NO                                                     ASSIGN
 54.            RETURN                                                    ASSIGN
 55.    C6.0       SET VARIABLES FOR DIRECTION SOUTHWEST.                 ASSIGN
 56.         17 KD=1                                                      ASSIGN
 57.            LD=-1                                                     ASSIGN
 58.    C           WHICH BOUNDARY (ROW OR COLUMN) WILL BE REACHED FIRST   ASSIGN
 59.            IF(NO-K.LE.L-1)GO TO 13                                   ASSIGN
 60.    C              COLUMN BOUNDARY WILL BE REACHED FIRST. SET JI AND JF  ASSIGN
 61.    C              ACCORDINGLY.                                        ASSIGN
 62.         19 JI=2                                                      ASSIGN
 63.            JF=L-1                                                    ASSIGN
 64.            RETURN                                                    ASSIGN
 65.    C7.0       SET VARIABLES FOR DIRECTION WEST.                      ASSIGN
 66.         21 KD=0                                                      ASSIGN
 67.            LD=-1                                                     ASSIGN
 68             GO TO 19                                                  ASSIGN
 69.    C8.0       SET VARIABLES FOR DIRECTION NORTHWEST.                 ASSIGN
```



19 FEB 73 8.02-39



19 PRO 73 0.02-39

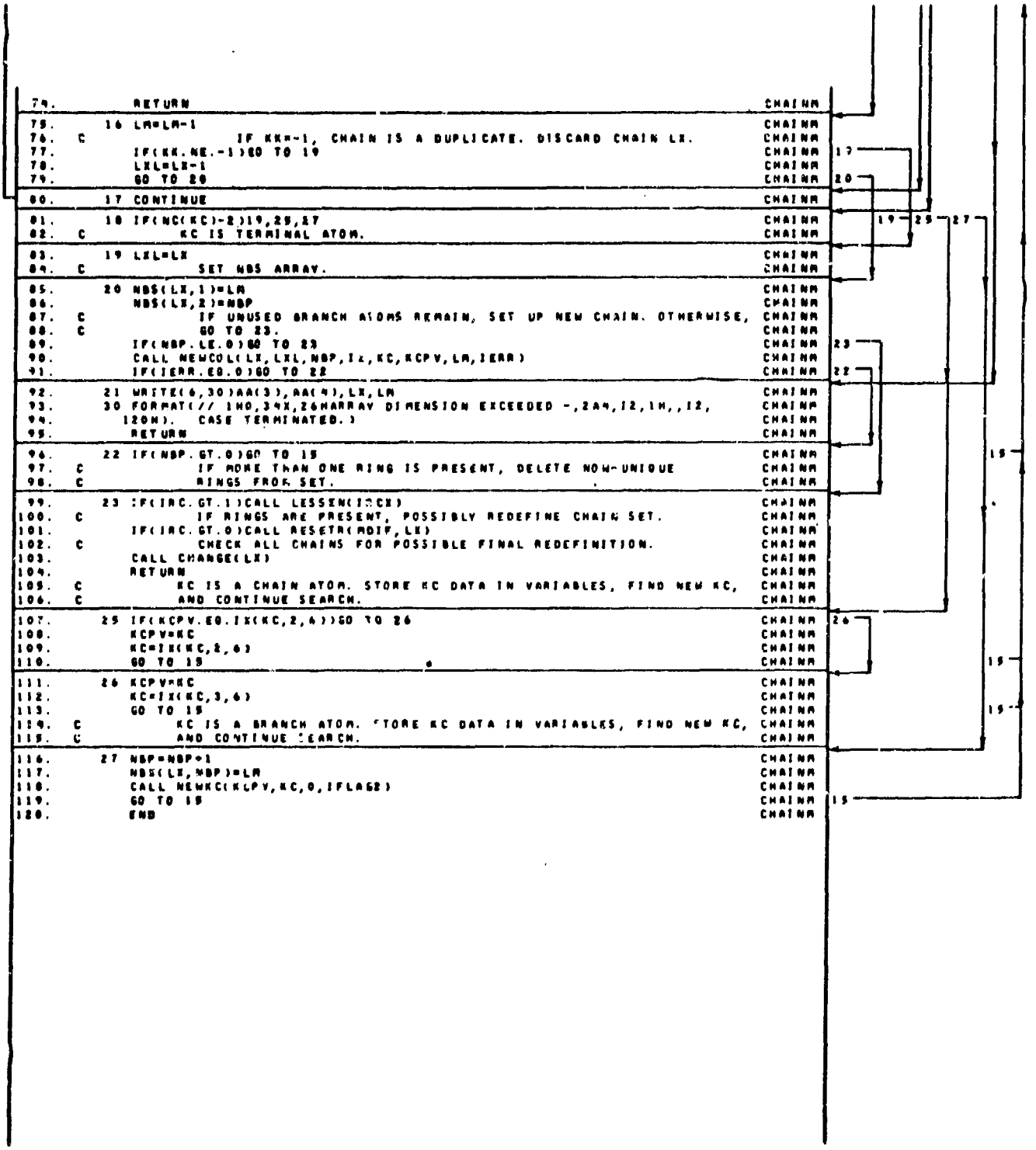
```
      SUBROUTINE ASYMC(NASYMC)                                          ASYMC
C        THIS SUBROUTINE DETERMINES THE NUMBER OF ASYMMETRIC CARBON     ASYMC
C        ATOMS PRESENT IN THE MOLECULE.                                 ASYMC
      INTEGER SYMI(4),SYMBOL(9),GRID(50,80)                             ASYMC
      INTEGER WEIGHT(9)                                                 ASYMC
      DIMENSION IOPATM(100),JBR(100),KCSAME(6,100)                      ASYMC
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                      ASYMC
      COMMON/BLK2/WEIGHT,AWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC     ASYMC
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),  ASYMC
     1IDBR(100),IB(100,8),IRG,NOBR                                      ASYMC
      EQUIVALENCE (IOPATM(1),IMATX(2,49)),(JBR(1),IMATX(2,43)),         ASYMC
     1(KCSAME(1,1),GRID(4,42))                                          ASYMC
C1.0     INITIALIZE COUNTER NASYMC.                                     ASYMC
      NASYMC=0                                                          ASYMC
C2.0     EXECUTE SEARCH FOR ASYMMETRIC ATOMS.                           ASYMC
      DO 5   K=1,KCC                                                    ASYMC
      IF(KCSAME(1,K).NE.0.OR.IX(K,1,1).NE.2.OR.IBC(K).NE.0.OR.          ASYMC
     1JBR(K).NE.4)GO TO 5                                               ASYMC
C        CORE ATOM K IS A NON-RING CARBON ATOM WITH FOUR LIGANDS ALL    ASYMC
C        OF WHICH ARE DISSIMILAR.                                       ASYMC
      NASYMC=NASYMC+1                                                   ASYMC
      IOPATM(NASYMC)=K                                                  ASYMC
    5 CONTINUE                                                          ASYMC
      RETURN                                                            ASYMC
      END                                                               ASYMC
```

19 FEB 73 8.02-38

```
  1.            SUBROUTINE BOND(KX,LX,KD,LD,JI,JF,I,IERR)                    BOND
  2.    C          THIS SUBROUTINE DETERMINES THE TYPE OF BOND AT A SPECIFIED  BOND
  3.    C          DIRECTION OF THE INDEXED CORE ATOM. IT ALSO SCANS SYMBOL INPUT BOND
  4.    C          ARRAY ALONG THIS DIRECTION UNTIL A SYMBOL UNEQUAL TO THE    BOND
  5.    C          DESIGNATED BOND IS DETECTED.                               BOND
  6.            INTEGER SYMX(4),SYMBOL(9),GRID(50,80)                        BOND
  7.            COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                 BOND
  8.    C1.0       DETERMINE IF INPUT SYMBOL IS A BOND SYMBOL.               BOND
  9.            DO1    I=1,3                                                 BOND
 10.            IF(GRID(KX,LX).EQ.SYMX(I))GO TO 11                           BOND   11
 11.          1 CONTINUE                                                     BOND
 12.    C          NO, IT IS NOT. PRINT ERROR MESSAGE, SET ERROR FLAG, AND EXIT. BOND
 13.            WRITE(6,5)KX,LX                                              BOND
 14.          5 FORMAT(// 1HO,28X,22HERROR - BOND SYMBOL AT,I4,1H,,I4,41H  MISSING BOND
 15.           1 OR INCORRECT.  CASE TERMINATED.)                            BOND
 16.          7 IERR=1                                                       BOND
 17.            RETURN                                                       BOND
 18.    C          YES, IT IS.                                               BOND
 19.    C2.0       FIND LOCATION OF NON-BOND SYMBOL AND EXIT.                BOND
 20.         11 DO 13     JJ=JI,JF                                          BOND
 21.            KX=KX+KD                                                     BOND
 22.            LX=LX+LD                                                     BOND
 23.            IF(GRID(KX,LX).NE.SYMX(I))RETURN                             BOND
 24.         13 CONTINUE                                                     BOND
 25.    C          THERE IS NO NON-BOND SYMBOL. PRINT ERROR MESSAGE, SET ERROR BOND
 26.    C          FLAG, AND EXIT.                                           BOND
 27.            WRITE(6,15)KX,LX                                             BOND
 28.         15 FORMAT(1HO,15X,32HFREE RADICAL AT GRID COORDINATES,I5,1H,,I5) BOND
 29.            GO TO 7                                                      BOND   7
 30.            END                                                          BOND
```

```
      SUBROUTINE CHAINA(LX,IERR)                                        CHAINA
C        THIS SUBROUTINE IS THE CONTROL ELEMENT FOR SECTION TWO OF THE  CHAINA
C        PROGRAM. THIS SECTION IDENTIFIES THE CHAIN FORMATIONS PRESENT  CHAINA
C        IN THE MOLECULE AS WELL AS THE NUMBER OF UNIQUE RINGS AND THE  CHAINA
C        RING COMPONENTS.                                               CHAINA
      INTEGER WEIGHT(9)                                                 CHAINA
      DIMENSION AA(4),JUNCT(9,100),NOBRD(100)                           CHAINA
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,9,6),NC(100),KCC     CHAINA
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),  CHAINA
     1IDBR(100),IB(100,8),IRG,NOBR                                      CHAINA
      COMMON/BLK4/NBC(40,50),NBS(40,2),NBX(40,20),IRC,NONFUS,IRCTOT     CHAINA
      EQUIVALENCE (JUNCT(1,1),IMATX(3,1)),(NOBRD(1),IMATX(3,11))        CHAINA
      DATA AA/4H  IR,4HING(,1H ,4HNBC(/                                 CHAINA
C1.0     INITIALIZE VARIABLES.                                          CHAINA
      LM=1                                                              CHAINA
      LX=1                                                              CHAINA
      NBP=0                                                             CHAINA
      MDIF=0                                                            CHAINA
      DO1    L=1,KCC                                                    CHAINA
    1 IBC(L)=0                                                          CHAINA
C2.0     SEARCH FOR TERMINAL GROUP.                                     CHAINA
      DO3    L=1,KCC                                                    CHAINA
      IF(NC(L).EQ.1)GO TO 5                                             CHAINA   5
    3 CONTINUE                                                          CHAINA
C           NO TERMINAL GROUP IS PRESENT.                               CHAINA
      KC=1                                                              CHAINA
      MDIF=KC                                                           CHAINA
      GO TO 7                                                           CHAINA   7
C           TERMINAL GROUP IS PRESENT.                                  CHAINA
    5 KC=L                                                              CHAINA
C3.0     IF BRANCH ATOMS ARE PRESENT, SET BRANCH COMPUTATION VARIABLES  CHAINA
C        NOBRD AND JUNCT.                                               CHAINA
    7 IF(NOBR.EQ.0)GO TO 12                                             CHAINA  12
      DO10   L=1,NOBR                                                   CHAINA
      KK=IDBR(L)                                                        CHAINA
      NOBRD(KK)=1                                                       CHAINA
      JF=NC(KK)                                                         CHAINA
      DO10   J=1,JF                                                     CHAINA
      JUNCT(J,KK)=IX(KK,J+1,6)                                          CHAINA
   10 CONTINUE                                                          CHAINA
C4.0     STORE FIRST ATOM OF FIRST CHAIN IN NBC. IS IT A BRANCH ATOM    CHAINA
   12 NBC(LX,LM)=IX(KC,1,6)                                             CHAINA
      IF(NC(KC).LE.2)GO TO 13                                           CHAINA  13
C           KC IS A BRANCH ATOM. FIND NEW KC.                           CHAINA
      NBP=NBP+1                                                         CHAINA
      NBX(LX,NBP)=LM                                                    CHAINA
      NOBRD(KC)=0                                                       CHAINA
      IFLAG1=1                                                          CHAINA
      CALL NEWKC(KCPV,KC,IFLAG1,IFLAG2)                                 CHAINA
      GO TO 15                                                          CHAINA  15
C           KC IS NOT A BRANCH ATOM. FIND NEW KC.                       CHAINA
   13 KCPV=KC                                                           CHAINA
      KC=IX(KC,2,6)                                                     CHAINA
C5.0     START OF CHAIN EVALUATION CYCLE.                               CHAINA
   15 LM=LM+1                                                           CHAINA
C           SET LM. IF NUMBER OF CHAINS EXCEEDS DIMENSION (50) OF ARRAY CHAINA
C           NBC, PRINT OUT MESSAGE AND EXIT.                            CHAINA
      IF(LM.GT.50)GO TO 21                                              CHAINA  21
      NBC(LX,LM)=IX(KC,1,6)                                             CHAINA
C           IS RING STRUCTURE PRESENT IN MOLECULE                       CHAINA
      IF(IRG.EQ.0)GO TO 18                                              CHAINA  18
C              YES, DETERMINE IF RING STRUCTURE IS CONTAINED IN CHAIN.  CHAINA
      LX=LM-1                                                           CHAINA
      DO17   KK=1,LX                                                    CHAINA
      IF(NBC(LX,KK).NE.NBC(LX,LM))GO TO 17                              CHAINA  17
C                 RING STRUCTURE CONTAINED IN CHAIN. STORE RING         CHAINA
C                 COMPONENTS.                                           CHAINA
      CALL RING(LX,LM,KK,IERR)                                          CHAINA
      IF(IERR.EQ.0)GO TO 16                                             CHAINA  16
C                 NUMBER OF RING COMPONENTS EXCEEDS DIMENSION OF ARRAY  CHAINA
C                 IRING. PRINT OUT MESSAGE AND EXIT.                    CHAINA
      NALT=LM-KK                                                        CHAINA
      WRITE(6,30)AA(1),AA(2),IRC,NALT                                   CHAINA
```

19 FEB 73 6.02-39



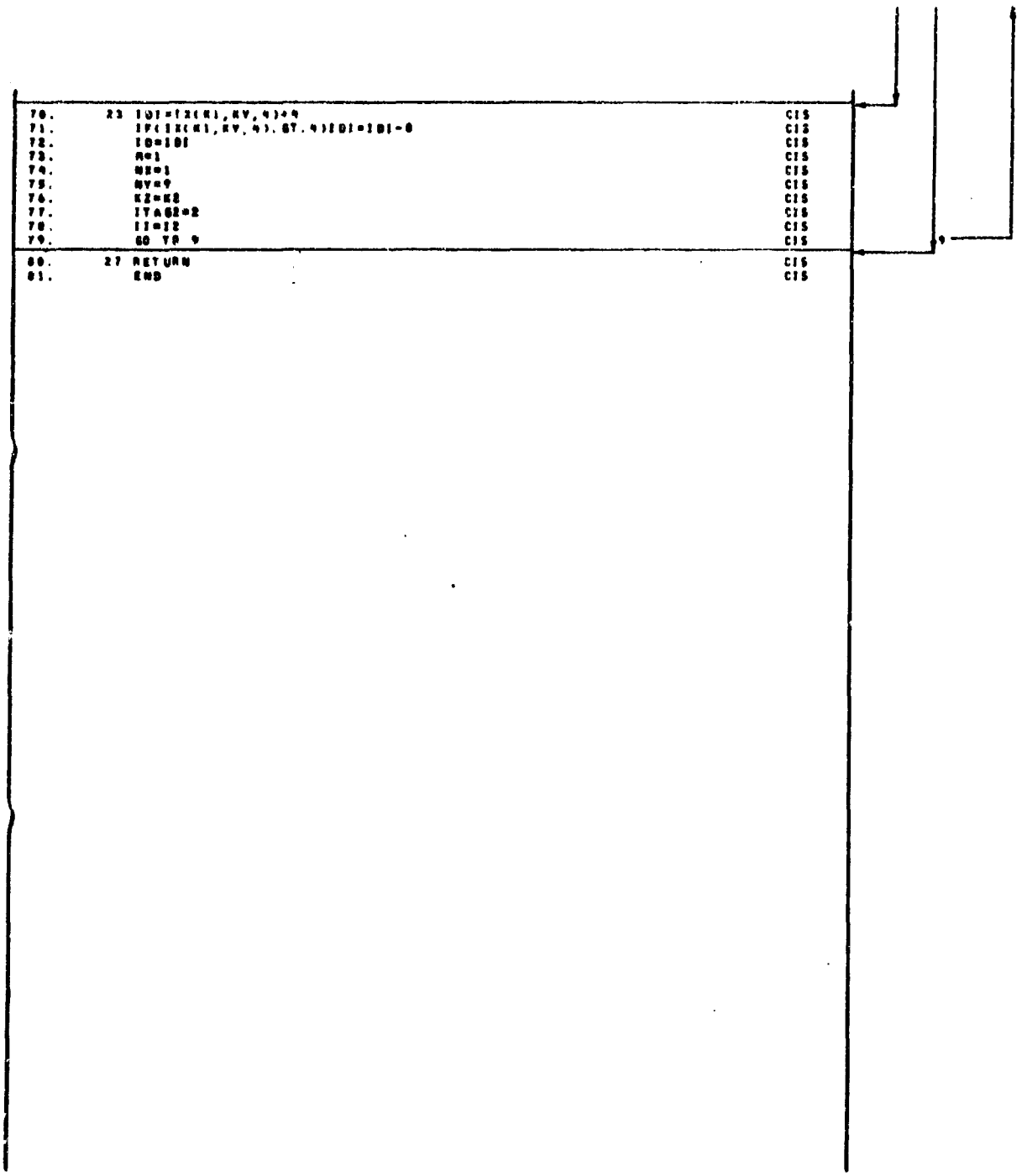
```
 79.          RETURN                                                    CHAINA
 75.       16 LR=LR-1                                                   CHAINA
 76.  C                 IF KK=-1, CHAIN IS A DUPLICATE. DISCARD CHAIN LX.CHAINA
 77.          IF(KK.NE.-1)GO TO 19                                      CHAINA  17
 78.          LXL=LX-1                                                  CHAINA
 79.          GO TO 20                                                  CHAINA  20
 80.       17 CONTINUE                                                  CHAINA
 81.       18 IF(NC(KC)-2)19,25,27                                      CHAINA  19 25 27
 82.  C           KC IS TERMINAL ATOM.                                  CHAINA
 83.       19 LXL=LX                                                    CHAINA
 84.  C              SET NBS ARRAY.                                     CHAINA
 85.       20 NBS(LX,1)=LR                                              CHAINA
 86.          NBS(LX,2)=NBP                                             CHAINA
 87.  C              IF UNUSED BRANCH ATOMS REMAIN, SET UP NEW CHAIN. OTHERWISE,CHAINA
 88.  C              GO TO 23.                                          CHAINA
 89.          IF(NBP.LE.0)GO TO 23                                      CHAINA  23
 90.          CALL NEWCOL(LX,LXL,NBP,IX,KC,KCPV,LR,IERR)                CHAINA
 91.          IF(IERR.EQ.0)GO TO 22                                     CHAINA  22
 92.       21 WRITE(6,30)AA(3),AA(4),LX,LR                              CHAINA
 93.       30 FORMAT(// 1H0,34X,26HARRAY DIMENSION EXCEEDED -,2A4,I2,1H,,I2,CHAINA
 94.         120H).  CASE TERMINATED.)                                  CHAINA
 95.          RETURN                                                    CHAINA
 96.       22 IF(NBP.GT.0)GO TO 15                                      CHAINA  15
 97.  C              IF MORE THAN ONE RING IS PRESENT, DELETE NON-UNIQUE CHAINA
 98.  C              RINGS FROM SET.                                    CHAINA
 99.       23 IF(IRC.GT.1)CALL LESSEN(ICCK)                             CHAINA
100.  C              IF RINGS ARE PRESENT, POSSIBLY REDEFINE CHAIN SET. CHAINA
101.          IF(IRC.GT.0)CALL RESETR(MDIF,LX)                          CHAINA
102.  C              CHECK ALL CHAINS FOR POSSIBLE FINAL REDEFINITION.  CHAINA
103.          CALL CHANGE(LX)                                           CHAINA
104.          RETURN                                                    CHAINA
105.  C           KC IS A CHAIN ATOM. STORE KC DATA IN VARIABLES, FIND NEW KC,CHAINA
106.  C           AND CONTINUE SEARCH.                                  CHAINA
107.       25 IF(KCPV.EQ.IX(KC,2,4))GO TO 26                            CHAINA  26
108.          KCPV=KC                                                   CHAINA
109.          KC=IX(KC,2,6)                                             CHAINA
110.          GO TO 15                                                  CHAINA  15
111.       26 KCPV=KC                                                   CHAINA
112.          KC=IX(KC,3,6)                                             CHAINA
113.          GO TO 15                                                  CHAINA  15
114.  C           KC IS A BRANCH ATOM. STORE KC DATA IN VARIABLES, FIND NEW KC,CHAINA
115.  C           AND CONTINUE SEARCH.                                  CHAINA
116.       27 NBP=NBP+1                                                 CHAINA
117.          NBS(LX,NBP)=LR                                            CHAINA
118.          CALL NEWKC(KCPV,KC,0,IFLAG2)                              CHAINA
119.          GO TO 15                                                  CHAINA  15
120.          END                                                       CHAINA
```

19 FEB 73 8.02.38

```
      1.            SUBROUTINE CHANGE(LX)                                    CHANGE
      2.      C        THIS SUBROUTINE REDEFINES ALL EXISTING CHAINS IN ARRAY NBC IF  CHANGE
      3.      C        A CHAIN EXISTS THAT (1) HAS ONE BRANCH ATOM ONLY (2) AND ITS   CHANGE
      4.      C        UPPER RESIDUAL (I2) IS GREATER THAN ITS LOWER RESIDUAL (I1).   CHANGE
      5.      C        IF MORE THAN ONE SUCH CHAIN EXISTS, THE ONE WITH THE MAXIMUM   CHANGE
      6.      C        I2-I1 IS CHOSEN AS THE REFERENCE.                              CHANGE
      7.               COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT  CHANGE
      8.      C1.0     INITIALIZE VARIABLE.                                           CHANGE
      9.               MAXD=0                                                         CHANGE
     10.      C2.0     INITIALIZE CYCLE THAT SEARCHES FOR CHAIN WITH ONE BRANCH ATOM  CHANGE
     11.      C         AND MAXIMUM I2-I1.                                            CHANGE
     12.               DO 4   L=1,LX                                                  CHANGE
     13.               IF(NBS(L,2).NE.1)GO TO 4                                       CHANGE  4
     14.      C           CHAIN L HAS ONLY ONE BRANCH ATOM.                           CHANGE
     15.               I1=NBX(L,1)-1                                                  CHANGE
     16.               I2=NBS(L,1)-NBX(L,1)                                           CHANGE
     17.               MDIF=I2-I1                                                     CHANGE
     18.               IF(MDIF.LE.MAXD)GO TO 4                                        CHANGE  4
     19.      C           MDIF MAXIMUM VALUE THUS FAR. STORE CHAIN PROPERTIES.       CHANGE
     20.               MAXD=MDIF                                                      CHANGE
     21.               MAXL=L                                                         CHANGE
     22.            4  CONTINUE                                                     CHANGE
     23.               IF(MAXD.EQ.0)RETURN                                            CHANGE
     24.      C3.0     POSITIVE DIFFERENTIAL OBTAINED. REDEFINE ALL CHAINS EXCEPT MAXL CHANGE
     25.               MAXIZ=NBX(MAXL,1)                                              CHANGE
     26.               DO 18    L=1,LX                                                CHANGE
     27.               IF(L.EQ.MAXL)GO TO 18                                          CHANGE 18
     28.      C           CHAIN L IS NOT CHAIN MAXL (REFERENCE). REDEFINE ITS VARIABLES CHANGE
     29.               JF=NBS(L,1)                                                    CHANGE
     30.               MX=JF+1                                                        CHANGE
     31.               DO 10    J=MAXIZ,JF                                           CHANGE
     32.               MX=MX-1                                                        CHANGE
     33.               MY=MX+MAX9                                                     CHANGE
     34.           10  NBC(L,MY)=NBC(L,MX)                                           CHANGE
     35.               MX=NBS(MAXL,1)+1                                               CHANGE
     36.               JF=NBS(MAXL,1)-MAXIZ                                           CHANGE
     37.               DO 12    J=1,JF                                               CHANGE
     38.               MX=MX-1                                                        CHANGE
     39.           12  NBC(L,J)=NBC(MAXL,MX)                                         CHANGE
     40.               NBS(L,1)=NBS(L,1)+MAXD                                         CHANGE
     41.               JF=NBS(L,2)                                                    CHANGE
     42.               DO 16    J=1,JF                                               CHANGE
     43.           16  NBX(L,J)=NBX(L,J)+MAXD                                        CHANGE
     44.           18  CONTINUE                                                     CHANGE
     45.               JF=NBS(MAXL,1)/2                                               CHANGE
     46.               MX=NBS(MAXL,1)+1                                               CHANGE
     47.      C4.0     REDEFINE VARIABLES OF CHAIN MAXL.                            CHANGE
     48.               DO 24    J=1,JF                                               CHANGE
     49.               MX=MX-1                                                        CHANGE
     50.               MY=NBC(MAXL,J)                                                 CHANGE
     51.               NBC(MAXL,J)=NBC(MAXL,MX)                                       CHANGE
     52.           24  NBC(MAXL,MX)=MY                                               CHANGE
     53.               NBX(MAXL,1)=NBX(MAXL,1)+MAXD                                   CHANGE
     54.               RETURN                                                         CHANGE
     55.               END                                                            CHANGE
```

19 FEB 73 8.02-35

```
      SUBROUTINE CIS(K1,K2,I1,I2,IEXIT)                                 CIS
C         THIS SUBROUTINE IDENTIFIES EACH PAIR OF LIGANDS CIS TO THE    CIS
C         LINKAGE C=C, WHERE BOTH CARBON ATOMS (K1 AND K2) HAVE A       CIS
C         CONNECTIVITY OF THREE OR CIS TO THE LINKAGE C-C WHERE K1 AND K2 CIS
C         ARE BOTH ADJACENT CARBON ATOMS IN A PARTICULAR RING EACH      CIS
C         WITH A CONNECTIVITY OF THREE OR FOUR.                         CIS
      INTEGER WEIGHT(9)                                                 CIS
      DIMENSION MS(9),MAT(9)                                            CIS
      COMMON/BLK2/WEIGHT,MUST(9),MOLWT(100),IX(100,9,6),NC(100),KCC     CIS
      COMMON/BLK3/IRING(40,30),IMAX(50,80),NW(100),IBC(100),KON(100),   CIS
     1IDBR(100),IB(100,8),IRG,NDBR                                      CIS
      EQUIVALENCE (MS(1),IB(2,1)),(MAT(1),IB(4,1))                      CIS
C1.0      INITIALIZE EXIT FLAG.                                         CIS
      IEXIT=0                                                           CIS
C2.0      ARE K1 AND K2 CARBON ATOMS AND ARE EACH BONDED TO THREE       CIS
C         OR MORE LIGANDS                                               CIS
      IF(IX(K1,1,1).EQ.2.AND.IX(K2,1,1).EQ.2.AND.KON(K1).GE.2.AND.      CIS
     1KON(K2).GE.3)GO TO 3                                              CIS
C           NO. SET EXIT FLAG AND EXIT FROM ROUTINE.                    CIS
      IEXIT=1                                                           CIS
      RETURN                                                            CIS
C           YES. INITIALIZE VARIABLES.                                  CIS
    3 DO 4   KK=3,9                                                     CIS
      MS(KK)=0                                                          CIS
    4 MAT(KK)=0                                                         CIS
      KF=NC(K1)+1                                                       CIS
C3.0      FIND LOCATION (KV) OF K2 DATA IN K1 SECTION OF IX ARRAY.      CIS
      DO 5   KV=2,KF                                                    CIS
      IF(IX(K1,KV,6).EQ.K2)GO TO 7                                      CIS
    5 CONTINUE                                                          CIS
C4.0      SET VARIABLES USED IN THE SCANNING OF ATOM K1.                CIS
    7 IDI=IX(K1,KV,4)                                                   CIS
      ID=IDI                                                            CIS
      MM=1                                                              CIS
      MF=0                                                              CIS
      MV=0                                                              CIS
      KV=K1                                                             CIS
      ITAG2=1                                                           CIS
      I1=I1                                                             CIS
      M=0                                                               CIS
C5.0      * FIND LIGANDS FOR ATOM K1 OR K2.                             CIS
    9 ITAG1=1                                                           CIS
   11 ID=ID+2                                                           CIS
      IF(ID.EQ.MV)ID=MF                                                 CIS
      IF(ID.EQ.IDI)GO TO 18                                             CIS
      DO 15   KK=2,9                                                    CIS
      IF(IX(K2,KK,4)-ID)15,17,15                                        CIS
   15 CONTINUE                                                          CIS
C           THERE IS NO LIGAND ALONG DIRECTION ID.                      CIS
      GO TO 11                                                          CIS
C           LIGANDS PRESENT ALONG ID. IF IT IS NON-RING, TRANSFER TO 170. CIS
C           OTHERWISE, DETERMINE WHETHER LIGAND DATA ARE TO BE STORED.  CIS
   17 IF(IX(K2,KK,6).EQ.0.OR.IX(K2,KK,6).EQ.0)GO TO 170                 CIS
      K22=IX(K2,KK,6)                                                   CIS
      IF(IBC(K22).EQ.0)GO TO 170                                        CIS
      IF(I1)11,170,169                                                  CIS
  169 I1=-1                                                             CIS
C           STORE CHEMICAL SYMBOL CODE AND GROUP NUMBER (IF ANY) OF     CIS
C           LIGAND.                                                     CIS
  170 M=M+1                                                             CIS
      MS(M)=IX(K2,KK,1)                                                 CIS
      MAT(M)=IX(K2,KK,6)                                                CIS
C           REPEAT PROCEDURE ONCE FOR EACH ATOM K1 AND K2.              CIS
   18 GO TO(19,21),ITAG1                                                CIS
   19 ITAG1=2                                                           CIS
      GO TO 11                                                          CIS
   21 GO TO(23,27),ITAG2                                                CIS
C4.3      SET VARIABLES USED IN THE SCANNING OF ATOM K2 AND TRANSFER    CIS
C         TO THE SCAN OPERATION.                                        CIS
```



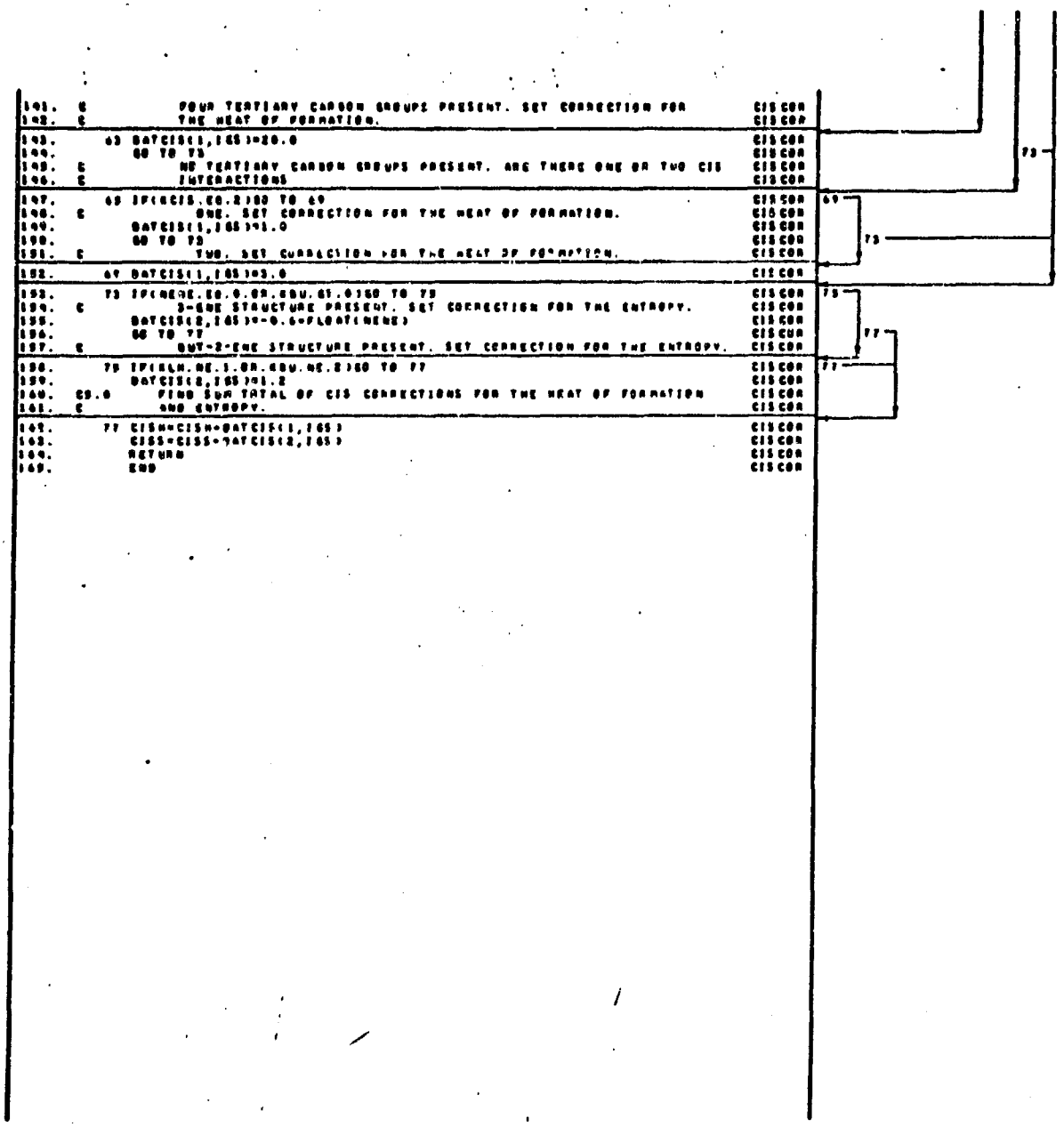
```
70.      23 IDI=IX(K1,KV,4)+9                                    CIS
71.         IF(IX(K1,KV,4).GT.4)IDI=IDI-8                        CIS
72.         I0=IDI                                               CIS
73.         K=1                                                  CIS
74.         KK=1                                                 CIS
75.         KV=9                                                 CIS
76.         K2=K8                                                CIS
77.         ITAG2=2                                              CIS
78.         II=I2                                                CIS
79.         GO TO 9                                              CIS
80.      27 RETURN                                               CIS
81.         END                                                  CIS
```

FEB 75 5.02-39

```
      SUBROUTINE CISCOR(K1,K2,I1,I2,IGS,IGSCIS,ICIS,CISM,CISS)                  CISCOR
C        THIS SUBROUTINE DETERMINES WHETHER THE LIGANDS OF ATOMS K1 AND         CISCOR
C        K2 EXHIBIT A CIS INTERACTION. IF SO, IT ESTABLISHES THE NUMBER,        CISCOR
C        TYPE, AND MAGNITUDE OF THIS CORRECTION.                                CISCOR
      INTEGER WEIGHT(9)                                                         CISCOR
      DIMENSION KCSAUS(3,150),KFGAUS(150),KTGAUS(150),DATCIS(2,150),            CISCOR
     1MS(9),MET(9)                                                              CISCOR
      COMMON/BLK2/WEIGHT,MWST(9),MOLWT(100),IX(100,9,6),NC(100),KCC             CISCOR
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),          CISCOR
     1IDBR(100),IBI(100,8),IR8,NOBR                                             CISCOR
      EQUIVALENCE (KCSAUS(1,1),IMATX(1,72)),(KFGAUS(1),IMATX(1,63)),            CISCOR
     1(KTGAUS(1),IMATX(1,40)),(DATCIS(1,1),IMATX(1,66)),(MS(1),                 CISCOR
     2IB(2,1)),(MET(1),IB(6,1))                                                 CISCOR
C1.0     IDENTIFY THE CIS COMPONENTS OF ATOMS K1 AND K2.                        CISCOR
      CALL CIS(K1,K2,I1,I2,IEXIT)                                               CISCOR
C        IF K1 AND K2 ARE BOTH CARBON ATOMS WITH A CONNECTIVITY OF              CISCOR
C        THREE OR MORE CONTINUE. OTHERWISE EXIT FROM ROUTINE.                   CISCOR
      IF(IEXIT.NE.0)RETURN                                                      CISCOR
C2.0     INITIALIZE VARIABLES.                                                  CISCOR
      KLK=0                                                                     CISCOR
      KTETRA=0                                                                  CISCOR
      KBU=0                                                                     CISCOR
      KTBL=0                                                                    CISCOR
      KTBR=0                                                                    CISCOR
      KCIS=0                                                                    CISCOR
      NEWE=0                                                                    CISCOR
C3.0     EXECUTE LOOP THAT TESTS CIS COMPONENTS.                                CISCOR
      DO 45   L=1,2                                                             CISCOR
      IF(MS(L).NE.1.OR.MS(L+2).NE.1)GO TO 3                                     CISCOR
C        BOTH CIS ATOMS ARE HYDROGEN ATOMS. THERE IS NO CIS CORRECTION          CISCOR
C        IN THIS CASE. INCREMENT HYDROGEN COUNTER AND TRANSFER.                 CISCOR
      KLH=KLH+1                                                                 CISCOR
      GO TO 45                                                                  CISCOR
C        ARE BOTH CIS ATOMS CARBON ATOMS EACH WITH A CONNECTIVITY               CISCOR
C        GREATER THAN TWO.                                                      CISCOR
    3 LI=MET(L)                                                                 CISCOR
      L2=MET(L+2)                                                               CISCOR
      IF(MS(L).NE.2)GO TO 5                                                     CISCOR
      IF(KON(LI)-2)45,45,7                                                      CISCOR
    5 IF(MS(L).NE.6)GO TO 45                                                    CISCOR
      IF(KON(LI)-1)45,45,7                                                      CISCOR
    7 IF(MS(L+2).NE.2)GO TO 9                                                   CISCOR
      IF(KON(L2)-2)45,45,10                                                     CISCOR
    9 IF(MS(L+2).NE.6)GO TO 45                                                  CISCOR
      IF(KON(L2)-1)45,45,10                                                     CISCOR
C        YES THEY ARE. ARE BOTH CIS ATOMS RING ATOMS AND PART OF THE            CISCOR
C        SAME RING SYSTEM. IF SO, THERE IS NO CIS INTERACTION                   CISCOR
C        BETWEEN THIS PAIR. TRANSFER TO 45.                                     CISCOR
   10 IF(IBC(LI)*IBC(L2).NE.0.AND.IBC(LI).EQ.IBC(L2))GO TO 45                   CISCOR
C           CIS INTERACTION EXISTS. SET VARIABLES AND INCREMENT CIS             CISCOR
C           COUNTER KCIS.                                                       CISCOR
      KTBLL=0                                                                   CISCOR
      KCL=0                                                                     CISCOR
      KCP=0                                                                     CISCOR
      KC3=0                                                                     CISCOR
      KCIS=KCIS+1                                                               CISCOR
      IF(KON(LI).NE.4)GO TO 25                                                  CISCOR
C           LIGAND ON ATOM K1 HAS A CONNECTIVITY OF FOUR.                       CISCOR
      KS=NC(LI)+1                                                               CISCOR
      GO TO(45,11,13,15,13),KS                                                  CISCOR
C           LIGAND HAS NO CORE LIGANDS EXCEPT K1. INCREMENT COUNTER             CISCOR
C           AND TRANSFER.                                                       CISCOR
   11 KBU=KBU+1                                                                 CISCOR
      GO TO 25                                                                  CISCOR
C           LIGAND HAS THREE CORE LIGANDS IN ADDITION TO K1. PROCESS            CISCOR
C           VARIABLES.                                                          CISCOR
   13 KTBL=KTBL+1                                                               CISCOR
      KTBLL=1                                                                   CISCOR
C           LIGAND HAS ONE OR TWO CORE LIGANDS IN ADDITION TO K1. IF            CISCOR
C           AT LEAST ONE IS A CARBON ATOM, INCREMENT KCL COUNTER. IF            CISCOR
C           LATTER LIGAND HAS ONLY ONE CORE ATOM AND PARENT LIGAND HAS          CISCOR
C           ONLY TWO, INCREMENT KC3 COUNTER.                                    CISCOR
```

```
      19 DO21   K=2,KS                                                 CISCOR
         IF(IE(L1,K,6).EQ.K1.OR.IE(L1,K,1).NE.2)GO TO 21               CISCOR
         KCL=KCL+1                                                     CISCOR
         KC1=IE(L1,K,6)                                                CISCOR
         IF(NC(KC1).GT.1.OR.NC(L1).NE.2)GO TO 21                       CISCOR
         KC3=KC3+1                                                     CISCOR
         GO TO 29                                                      CISCOR
      21 CONTINUE                                                      CISCOR
      29 IF(NCON(L2).NE.4)GO TO 45                                     CISCOR
C                 LIGAND ON ATOM K2 HAS A CONNECTIVITY OF FOUR.        CISCOR
         KS=NC(L2)+1                                                   CISCOR
         GO TO(45,31,35,35,33),KS                                      CISCOR
C                 LIGAND HAS NO CORE LIGANDS EXCEPT K2. INCREMENT COUNTER CISCOR
C                 AND TRANSFER.                                        CISCOR
      31 KSU=KSU+1                                                     CISCOR
         GO TO 45                                                      CISCOR
C                 LIGAND HAS THREE CORE LIGANDS IN ADDITION TO K2. PROCESS CISCOR
C                 VARIABLES.                                           CISCOR
      33 KTBN=KTBN+1                                                   CISCOR
         IF(KTBLL.EQ.1)KTETRA=KTETRA+1                                 CISCOR
C                 LIGAND HAS ONE OR TWO CORE LIGANDS IN ADDITION TO K2. IF CISCOR
C                 AT LEAST ONE IS A CARBON ATOM, INCREMENT KCR COUNTER. IF CISCOR
C                 LATTER LIGAND HAS ONLY ONE CORE ATOM AND PARENT LIGAND HAS CISCOR
C                 ONLY TWO, INCREMENT KC3 COUNTER.                     CISCOR
      35 DO37   K=2,KS                                                 CISCOR
         IF(IE(L2,K,6).EQ.K2.OR.IE(L2,K,1).NE.2)GO TO 37               CISCOR
         KCR=KCR+1                                                     CISCOR
         KC2=IE(L2,K,6)                                                CISCOR
         IF(NC(KC2).GT.1.OR.NC(L2).NE.2)GO TO 37                       CISCOR
         KC3=KC3+1                                                     CISCOR
         GO TO 41                                                      CISCOR
      37 CONTINUE                                                      CISCOR
C             IF KCL, KCR, AND KC3 WERE ALL ACTIVATED, SET COUNTER NENE. CISCOR
      41 IF(KCL.GT.0.AND.KCR.GT.0.AND.KC3.GT.0)NENE=NENE+1             CISCOR
      45 CONTINUE                                                      CISCOR
         IF(KCIS.EQ.0)RETURN                                           CISCOR
C4.0         CIS INTERACTION(S) PRESENT. FIND AND APPLY APPROPRIATE    CISCOR
C            CORRECTION(S).                                            CISCOR
C               PROCESS VARIABLES.                                     CISCOR
         IGSCIS=IGSCIS+1                                               CISCOR
         ICIS=ICIS+KCIS                                                CISCOR
         IGS=IGS+1                                                     CISCOR
         KTGAUS(IGS)=4                                                 CISCOR
         KFGAUS(IGS)=1                                                 CISCOR
         KCGAUS(1,IGS)=K1                                              CISCOR
         KCGAUS(2,IGS)=K2                                              CISCOR
         IF(KCIS.EQ.2)KFGAUS(IGS)=2                                    CISCOR
         KSUM=KTBL+KTBN+1                                              CISCOR
         GO TO(65,49,53,59,63),KSUM                                    CISCOR
C               ONE TERTIARY CARBON GROUP PRESENT. SET CORRECTION FOR THE CISCOR
C               HEAT OF FORMATION.                                     CISCOR
      49 DATCIS(1,IGS)=4.0                                             CISCOR
      53 IF(KCIS.NE.2)GO TO 73                                         CISCOR
C                 TWO CIS INTERACTIONS PRESENT. ADD THIS CORRECTION    CISCOR
         DATCIS(1,IGS)=DATCIS(1,IGS)+2.0                               CISCOR
         GO TO 73                                                      CISCOR
C               TWO TERTIARY CARBON GROUPS PRESENT. ARE THEY BOTH CIS  CISCOR
      55 IF(KTETRA.NE.1)GO TO 57                                       CISCOR
C                 YES. SET CORRECTION FOR THE HEAT OF FORMATION        CISCOR
         DATCIS(1,IGS)=10.0                                            CISCOR
         GO TO 53                                                      CISCOR
C                 NO. SET CORRECTION FOR THE HEAT OF FORMATION         CISCOR
      57 DATCIS(1,IGS)=8.0                                             CISCOR
         GO TO 73                                                      CISCOR
C               THREE TERTIARY CARBON GROUPS PRESENT  SET CORRECTION FOR CISCOR
C               THE HEAT OF FORMATION                                  CISCOR
      59 DATCIS(1,IGS)=14.0                                            CISCOR
         GO TO 73                                                      CISCOR
```

19 FEB 73 8.02-33



```
C          FOUR TERTIARY CARBON GROUPS PRESENT. SET CORRECTION FOR
C          THE HEAT OF FORMATION.
      63 DATCIS(1,165)=20.0
         GO TO 73
C          NO TERTIARY CARBON GROUPS PRESENT. ARE THERE ONE OR TWO CIS
C          INTERACTIONS
      65 IF(ICIS.EQ.2)GO TO 69
C                ONE. SET CORRECTION FOR THE HEAT OF FORMATION.
         DATCIS(1,165)=1.0
         GO TO 73
C                TWO. SET CORRECTION FOR THE HEAT OF FORMATION.
      69 DATCIS(1,165)=3.0
      73 IF(NENE.EQ.0.OR.ABU.GT.0)GO TO 75
C             3-ENE STRUCTURE PRESENT. SET CORRECTION FOR THE ENTROPY.
         DATCIS(2,165)=-0.6*FLOAT(NENE)
         GO TO 77
C             OUT-2-ENE STRUCTURE PRESENT. SET CORRECTION FOR THE ENTROPY.
      75 IF(ALM.NE.1.OR.ABU.NE.2)GO TO 77
         DATCIS(2,165)=1.2
C.0      FIND SUM TOTAL OF CIS CORRECTIONS FOR THE HEAT OF FORMATION
C          AND ENTROPY.
      77 CISH=CISH+DATCIS(1,165)
         CISS=CISS+DATCIS(2,165)
         RETURN
         END
```

```
      SUBROUTINE CORCIG(NOSN,NENAN,MESO,LX,HSYM,SSYM,CPSYM)                  CORCIG
C        THIS SUBROUTINE IS THE CONTROL ELEMENT FOR SECTION THREE OF THE     CORCIG
C        PROGRAM. THIS SECTION SEARCHES FOR AND ACCOUNTS FOR ALL SECOND-     CORCIG
C        ORDER INTERACTIONS AND RING CORRECTIONS AS WELL AS CONTRIBU-         CORCIG
C        TIONS DUE TO INTERNAL AND EXTERNAL ROTATIONAL SYMMETRY AND          CORCIG
C        OPTICAL ISOMERISM.                                                  CORCIG
      INTEGER SYMX(4),SYMBOL(9),GRID(50,80)                                  CORCIG
      INTEGER WEIGHT(9)                                                      CORCIG
      INTEGER RINGLI(40)                                                     CORCIG
      DIMENSION CPSYM(4),NOKOMB(11),KCGAUS(3,150),KFGAUS(150),               CORCIG
     1KTGAUS(150),DATCIS(2,150),MDEL(40),IM(150),CPALT(4),NRING(40),         CORCIG
     2KRCLOR(100),KRCNWT(100),NONARO(40),MNIX(2,200),MARO(100)               CORCIG
      DIMENSION TYPE(4,8),CPCIS(4),DORTHO(6),RINGD(6,50)                     CORCIG
      DIMENSION A1(6),A2(6),A3(6),A4(6),A5(6),A6(6),A7(6),A8(6),A9(6),       CORCIG
     1A10(6),A11(6),A12(6),A13(6),A14(6),A15(6),A16(6),A17(6),A18(6),        CORCIG
     2A19(6),A20(6),A21(6),A22(6),A23(6),A24(6),A25(6),A26(6),A27(6),        CORCIG
     3A28(6),A29(6),A30(6)                                                   CORCIG
      DIMENSION A31(6),A32(6),A33(6),A34(6),A35(6),A36(6),A37(6),A38(6),     CORCIG
     1A39(6),A40(6),A41(6),A42(6),A43(6),A44(6)                              CORCIG
      DIMENSION A45(6),A46(6),A47(6),A48(6),A49(6),A50(6)                    CORCIG
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                           CORCIG
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC          CORCIG
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),       CORCIG
     1IDORO(100),IB(100,8),IRG,NOBR                                          CORCIG
      COMMON/BLK4/KBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT          CORCIG
      COMMON/BLK5/NOATM,NUMATM(5),MDC(50),MBS(2),JW,JY,LFLAGS,LFLAGG         CORCIG
      EQUIVALENCE (RINGD(1,1),A1),(RINGD(1,2),A2),(RINGD(1,3),A3),           CORCIG
     1(RINGD(1,4),A4),(RINGD(1,5),A5),(RINGD(1,6),A6),(RINGD(1,7),A7),       CORCIG
     2(RINGD(1,8),A8),(RINGD(1,9),A9),(RINGD(1,10),A10),(RINGD(1,11),        CORCIG
     3A11),(RINGD(1,12),A12),(RINGD(1,13),A13),(RINGD(1,14),A14),            CORCIG
     4(RINGD(1,15),A15),(RINGD(1,16),A16),(RINGD(1,17),A17),                 CORCIG
     5(RINGD(1,18),A18),(RINGD(1,19),A19),(RINGD(1,20),A20),                 CORCIG
     6(RINGD(1,21),A21),(RINGD(1,22),A22),(RINGD(1,23),A23),                 CORCIG
     7(RINGD(1,24),A24),(RINGD(1,25),A25),(RINGD(1,26),A26),                 CORCIG
     8(RINGD(1,27),A27),(RINGD(1,28),A28),(RINGD(1,29),A29),                 CORCIG
     9(RINGD(1,30),A30)                                                      CORCIG
      EQUIVALENCE (RINGD(1,31),A31),(RINGD(1,32),A32),(RINGD(1,33),A33),     CORCIG
     1(RINGD(1,34),A34),(RINGD(1,35),A35),(RINGD(1,36),A36),                 CORCIG
     2(RINGD(1,37),A37),(RINGD(1,38),A38),(RINGD(1,39),A39),                 CORCIG
     3(RINGD(1,40),A40),(RINGD(1,41),A41),(RINGD(1,42),A42),                 CORCIG
     4(RINGD(1,43),A43),(RINGD(1,44),A44),(RINGD(1,45),A45),                 CORCIG
     5(RINGD(1,46),A46),(RINGD(1,47),A47),(RINGD(1,48),A48),                 CORCIG
     6(RINGD(1,49),A49),(RINGD(1,50),A50)                                    CORCIG
      EQUIVALENCE (NBENZ,ID(10,1)),(NOSIX,IB(11,1)),(MD,IB(12,1)),           CORCIG
     1(MDEL(1),IB(13,1)),(RINGLI(1),ID(95,1)),(NOKOMB(1),IB(80,8)),          CORCIG
     2(KCGAUS(1,1),IMATX(1,72)),(KFGAUS(1),IMATX(1,63)),(KTGAUS(1),          CORCIG
     3IMATX(1,60)),(DATCIS(1,1),IMATX(1,66)),(IM(1),IMATX(1,57)),            CORCIG
     4(CPALT(1),ID(91,8)),(NRING(1),GRID(1,75)),(KRCLOR(1),GRID(1,76)),      CORCIG
     5(KRCNWT(1),GRID(1,78)),(NNBENZ,GRID(1,80)),(NONARO(1),                 CORCIG
     6IMATX(2,49)),(MNIX(1,1),GRID(4,1)),(NIBENZ,ID(96,8)),(MARO(1),         CORCIG
     7GRID(1,25))                                                            CORCIG
      DATA TYPE(1,1)/4H RIN/,TYPE(2,1)/4HNG CO/,TYPE(3,1)/4HBREC/,           CORCIG
     1TYPE(4,1)/4HTION/,TYPE(1,2)/4HNORTH/,TYPE(2,2)/4HO CO/,TYPE(3,2)       CORCIG
     2/4HBREC/,TYPE(4,2)/4HTION/,TYPE(1,3)/4H PAR/,TYPE(2,3)/4HA CO/,        CORCIG
     3TYPE(3,3)/4HBREC/,TYPE(4,3)/4HTION/,TYPE(1,4)/1H /,TYPE(2,4)           CORCIG
     4/4H   C/,TYPE(3,4)/2HIS/,TYPE(4,4)/1H /,TYPE(1,5)/4H  GA/,            CORCIG
     5TYPE(2,5)/4HUCHE/,TYPE(3,5)/4H ALK/,TYPE(4,5)/3HENE/,TYPE(1,6)        CORCIG
     6/4H  CO/,TYPE(2,6)/4HUCHE/,TYPE(3,6)/4H ALK/,TYPE(4,6)/3HENE/,        CORCIG
     7TYPE(1,7)/4H  GA/,TYPE(2,7)/4HUCHE/,TYPE(3,7)/4H ETH/,TYPE(4,7)       CORCIG
     8/2HR /,TYPE(1,8)/4HDITE/,TYPE(2,8)/4HRATIA/,TYPE(3,8)/4HRY E/,        CORCIG
     9TYPE(4,8)/4HTHER/                                                      CORCIG
      DATA CPCIS/-2.719C0850E+0,5.85308034E-3,-4.78484651E-6,1.39424152      CORCIG
     1E-9/                                                                   CORCIG
      DATA  DORTHO /0.57,-1.61,3.67654979E-1, 4.40466325E-3,                 CORCIG
     1-6.24[5104?C-6, 2.09796396E-9/,DORPAR/-1.50/                          CORCIG
C        CORRECTIONS FOR HYDROCARBON, OXYGEN-CONTAINING, AND                 CORCIG
C        NITROGEN-CONTAINING RINGS.                                          CORCIG
      DATA A1/ 27.6,   32.1,    -5.595190?1E+0,    1.15774509E-2,            CORCIG
     1 -1.11538029E-5,    3.49006739E-9/                                     CORCIG
      DATA A2/43.7,33.6,4*0.0/                                              CORCIG
      DATA A3/ 26.2,   26.0,    -7.96086542E+0,    1.38122922E-2,            CORCIG
     1  -9.77191219E-6,    2.53582451E-9/                                   CORCIG
      DATA A4/  29.0,   29.0,    -9.15702016E+0,    7.04990289E-3,           CORCIG
     1  -5.93029990E-6,    1.69080107E-9/                                   CORCIG
      DATA A5/  6.3,    27.3,    -1.20102313E+1,    1.73221158E-2,           CORCIG
```

```
     1  -8.84512613E-6,   1.64956435E-9/                                    CORCIG
      DATA A6/  5.9,  25.8,  -7.96493944E+0,   6.47567798E-3,               CORCIG
     1  4.71777450E-7,   -1.15478934E-9/                                    CORCIG
      DATA A7/6.0,5*0.0/                                                   CORCIG
      DATA A8/  0.00000001,  18.8, -1.31204886E+1,  2.55471733E-2,         CORCIG
     1   -1.13372646E-5,   1.03305952E-9/                                  CORCIG
      DATA A9/  1.4,  21.5,  -9.86261273E+0,   2.40236698E-2,              CORCIG
     1  -1.96337388E-5,   5.97394852E-9/                                   CORCIG
      DATA A10/4.8,5*0.0/                                                  CORCIG
      DATA A11/0.5,5*0.0/                                                  CORCIG
      DATA A12/6.4,15.9,4*0.0/                                             CORCIG
      DATA A13/5.4,5*0.0/                                                  CORCIG
      DATA A14/6.6,5*0.0/                                                  CORCIG
      DATA A15/4.7,23.7,4*0.0/                                             CORCIG
      DATA A16/5.9,16.5,4*0.0/                                             CORCIG
      DATA A17/6.0,5*0.0/                                                  CORCIG
      DATA A18/15.5,5*0.0/                                                 CORCIG
      DATA A19/8.9,5*0.0/                                                  CORCIG
      DATA A20/17.1,5*0.0/                                                 CORCIG
      DATA A21/12.8,5*0.0/                                                 CORCIG
      DATA A22/9.9,5*0.0/                                                  CORCIG
      DATA A23/12.8,5*0.0/                                                 CORCIG
      DATA A24/63.5,67.6,4*0.0/                                            CORCIG
      DATA A25/60.4,69.2,4*0.0/                                            CORCIG
      DATA A26/55.2,5*0.0/                                                 CORCIG
      DATA A27/32.7,5*0.0/                                                 CORCIG
      DATA A28/28.9,5*0.0/                                                 CORCIG
      DATA A29/29.6,5*0.0/                                                 CORCIG
      DATA A30/31.1,5*0.0/                                                 CORCIG
      DATA A31/27.6, 31.9, 4.53771953E+0, -3.66170519E-2,                  CORCIG
     1   5.70918434E-5, -2.73325817E-8/                                    CORCIG
      DATA A32/26.9, 27.7, -5.31540263E+0, 2.40262420E-3,                  CORCIG
     1  -4.32548458E-6,   7.30512351E-9/                                   CORCIG
      DATA A33/6.7,5*0.0/                                                  CORCIG
      DATA A34/2.2,5*0.0/                                                  CORCIG
      DATA A35/3.5,5*0.0/                                                  CORCIG
      DATA A36/5.4,5*0.0/                                                  CORCIG
      DATA A37/3.4,5*0.0/                                                  CORCIG
      DATA A38/-6.2,5*0.0/                                                 CORCIG
      DATA A39/2.5,5*0.0/                                                  CORCIG
      DATA A40/6.0,5*0.0/                                                  CORCIG
      DATA A41/3.4,5*0.0/                                                  CORCIG
      DATA A42/1.1,5*0.0/                                                  CORCIG
      DATA A43/1.4,5*0.0/                                                  CORCIG
      DATA A44/4.6,5*0.0/                                                  CORCIG
      DATA A45/27.7,31.6,4*0.0/                                            CORCIG
      DATA A46/26.2,29.3,4*0.0/                                            CORCIG
      DATA A47/6.8,  26.7,  -7.49239560E+0, 1.47921187E-3,                 CORCIG
     1   1.18614544E-5, -8.02285243E-9/                                    CORCIG
      DATA A48/1.0,5*0.0/                                                  CORCIG
      DATA A49/3.4,5*0.0/                                                  CORCIG
      DATA A50/8.5,5*0.0/                                                  CORCIG
C1.0      INITIALIZE VARIABLES FOR THIS ROUTINE AND SUBSEQUENT             CORCIG
C         ROUTINES IN SECTION THREE OF THE PROGRAM.                        CORCIG
      IGS=0                                                                CORCIG
      IRNG2=0                                                              CORCIG
      IRNG3=0                                                              CORCIG
      ICRTHO=0                                                             CORCIG
      IORPAR=0                                                             CORCIG
      IGO=0                                                                CORCIG
      GAUCHH=0.0                                                           CORCIG
      ICIS=0                                                               CORCIG
      IGSCIS=0                                                             CORCIG
      CISH=0.0                                                             CORCIG
      CISS=0.0                                                             CORCIG
      IBH=0                                                                CORCIG
      IOTE=0                                                               CORCIG
      OTEH=0.0                                                             CORCIG
      NTBENZ=0                                                             CORCIG
      DO9   J=1,KCC                                                        CORCIG
      MARO(J)=0                                                            CORCIG
      KFGAUS(J)=0                                                          CORCIG
      KTGAUS(J)=0                                                          CORCIG
      KRCMOR(J)=0                                                          CORCIG
      KRCNUT(J)=1                                                          CORCIG
```

17 APR 73 0:02:36

```
      DO1   K=1,2                                                       CORCIG
      DATCIS(K,J)=0.0                                                   CORCIG
    1 CONTINUE                                                          CORCIG
      DO3   L=1,3                                                       CORCIG
      KCGAUS(L,J)=0                                                     CORCIG
    3 CONTINUE                                                          CORCIG
    5 CONTINUE                                                          CORCIG
      KF=2*KCC                                                          CORCIG
      DO9   K=1,KF                                                      CORCIG
      MNIX(1,K)=0                                                       CORCIG
      MNIX(2,K)=0                                                       CORCIG
    9 CONTINUE                                                          CORCIG
      MN=0                                                              CORCIG
      IF(IRC.EQ.0)GO TO 12                                              CORCIG
C         RING STRUCTURES ARE PRESENT, SET RING ARRAYS NRING AND NONARO CORCIG
      DO11  K=1,IRC                                                     CORCIG
      NONARO(K)=1                                                       CORCIG
      NRING(K)=0                                                        CORCIG
   11 CONTINUE                                                          CORCIG
C2.0     THIS SECTION SEARCHES FOR GAUCHE AND CIS INTERACTIONS FOR WHICH CORCIG
C        THE CENTRAL ATOMS ARE NON-RING ATOMS AND FINDS THEIR CONTRIBUT- CORCIG
C        IONS IF THESE INTERACTIONS OCCUR.                              CORCIG
C          FIND BOND PAIR K1-K2 FOR TESTING. BOTH K1 AND K2 MUST BE     CORCIG
C          NON-RING ATOMS.                                              CORCIG
   12 DO57  K1=1,KCC                                                    CORCIG
      IF(IRC(K1).NE.0)GO TO 57                                          CORCIG
C            K1 IS NOT RING ATOM.                                       CORCIG
      JF=NC(K1)+1                                                       CORCIG
      DO55  J=2,JF                                                      CORCIG
      K2=IX(K1,J,6)                                                     CORCIG
      IF(IRC(K2).NE.0)GO TO 55                                          CORCIG
C            K2 IS NOT RING ATOM.                                       CORCIG
C            HAS BOND PAIR K1-K2 BEEN TESTED PREVIOUSLY                 CORCIG
      IF(MN.EQ.0)GO TO 17                                               CORCIG
      DO13  LL=1,MN                                                     CORCIG
      IF(MNIX(1,LL).EQ.K1.AND.MNIX(2,LL).EQ.K2.OR.MNIX(1,LL).EQ.K2.AND. CORCIG
     1MNIX(2,LL).EQ.K1)GO TO 55                                         CORCIG
   13 CONTINUE                                                          CORCIG
C                 NO IT HAS NOT. SET ARRAY INDICATORS FOR THIS BOND PAIR. CORCIG
   17 MN=MN+1                                                           CORCIG
      MNIX(1,MN)=K1                                                     CORCIG
      MNIX(2,MN)=K2                                                     CORCIG
C          IF K1 DATA ARE NOT IN PARENT LOCATION IN THE IX ARRAY IN THE CORCIG
C          IX DATA OF K2, PLACE THEM THERE.                             CORCIG
      IF(IX(K2,1,4).NE.IX(K1,1,6))CALL SHIFT(K2,K1,0)                   CORCIG
C            WHAT IS THE BOND TYPE BETWEEN K1 AND K2                    CORCIG
      IF(IX(K2,2,5)-2)35,47,55                                          CORCIG
C            SINGLE BOND PRESENT. FIND GAUCHE CONTRIBUTIONS (IF ANY).   CORCIG
   35 CALL GAUCHE(K1,K2,IGS,IGH,IGG,GAUCHN)                             CORCIG
      GO TO 55                                                          CORCIG
C                 DOUBLE BOND PRESENT. FIND CIS CONTRIBUTIONS (IF ANY). CORCIG
   47 CALL CISCOR(K1,K2,0,0,IGS,IGSCIS,ICIS,CISH,CISS)                  CORCIG
   55 CONTINUE                                                          CORCIG
   57 CONTINUE                                                          CORCIG
C3.0     SET THE THERMODYNAMIC AND PRINTOUT VARIABLES.                  CORCIG
      NRING=0                                                           CORCIG
      SOIR.G=0.0                                                        CORCIG
      DO63  L=1,4                                                       CORCIG
   63 CPSYM(L)=0.0                                                      CORCIG
      NTITLE=0                                                          CORCIG
      NE=0                                                              CORCIG
      ND=0                                                              CORCIG
      IF(IRC.NE.0)GO TO 71                                              CORCIG
C4.0     RING STRUCTURES ARE NOT PRESENT. PRINT OUT STRUCTURAL DATA     CORCIG
C        AND GO TO 87.                                                  CORCIG
      CALL PRINT1                                                       CORCIG
      GO TO 87                                                          CORCIG
C5.0     RING STRUCTURES ARE PRESENT. SEARCH FOR AND IDENTIFY FUSED     CORCIG
C        RING SYSTEMS AND FIND RING CORRECTIONS.                        CORCIG
   71 NJTOT(1)=0                                                        CORCIG
C          IDENTIFY SET(S) OF FUSED RING SYSTEMS PRESENT IN MOLECULE    CORCIG
C          (IF ANY).                                                    CORCIG
```

17 APR 73 8.02-38

```
223.            CALL FUSION                                                    CORCIG
224.            NBENZ=0                                                        CORCIG
225.            NNBENZ=0                                                       CORCIG
226.            NOSIX=0                                                        CORCIG
227.      C          FIND SUBSCRIPT CORRESPONDING TO LOCATION OF THERMOCHEMICAL CORCIG
228.      C          CORRECTIONS FOR RING K.                                   CORCIG
229.            DO75   K=1,IRC                                                 CORCIG
230.            CALL CYCORR(K,L1)                                              CORCIG
231.            IF(L1.GT.0)GO TO 73                                            CORCIG   73
232.            ME=ME+1                                                        CORCIG
233.         73 RINGL1(K)=L1                                                   CORCIG
234.         75 CONTINUE                                                       CORCIG
235.            IF(NBENZ.GT.0.OR.NNBENZ.GT.0)GO TO 77                          CORCIG   77
236.      C          BENZENE OR PYRIDINE-LIKE RINGS ARE NOT PRESENT. PRINT OUT CORCIG
237.      C          STRUCTURAL DATA AND GO TO 79.                             CORCIG
238.            CALL PRINT1                                                    CORCIG
239.            GO TO 79                                                       CORCIG   79
240.      C          FIND WEIGHT CORRECTIONS AND ORTHO (PARA) CORRECTIONS FOR  CORCIG
241.      C          BENZENE AND PYRIDINE-LIKE RINGS.                          CORCIG
242.         77 CALL HEXGON(IGS,IORTHO,IORPAR,DORTHO,DORPAR,HRING,SRING,CPSYM) CORCIG
243.         79 IF(NOKOMB(1).EQ.0)GO TO 83                                     CORCIG   83
244.      C          FUSED RINGS ARE PRESENT. FIND RING CORRECTIONS FOR CERTAIN CORCIG
245.      C          HYDROCARBON FUSED RING SYSTEMS.                           CORCIG
246.            CALL CRINGS(IGS,IRNG2,RINGD(1,24),HRING,SRING,CPSYM)           CORCIG
247.      C          FIND CORRECTIONS FOR A NITROGEN-CONTAINING FUSED RING SYSTEM. CORCIG
248.            CALL NRINGS(IGS,IRNG3,RINGD(1,49),HRING,SRING,CPSYM)           CORCIG
249.      C          DELETE RING CORRECTION FLAGS FROM RING SETS WHICH FORM PART CORCIG
250.      C          OF FUSED RING SETS AND HENCE HAVE THEIR RING CORRECTIONS   CORCIG
251.      C          ALREADY INCLUDED THEREIN.                                 CORCIG
252.         83 J=0                                                            CORCIG
253.         85 J=J+1                                                          CORCIG
254.            IF(J.GT.MD)GO TO 86                                            CORCIG   86
255.            K=MOEL(J)                                                      CORCIG
256.            RINGL1(K)=0                                                    CORCIG
257.            GO TO 85                                                       CORCIG   85
258.      C          SEARCH FOR ADDITIONAL RING CORRECTIONS AS WELL AS FOR     CORCIG
259.      C          GAUCHE AND CIS INTERACTIONS IN WHICH ONE OR BOTH OF THE   CORCIG
260.      C          CENTRAL ATOMS ARE NON-AROMATIC RING ATOMS AND FIND THEIR  CORCIG
261.      C          CONTRIBUTIONS IF THESE OCCUR.                             CORCIG
262.         86 CALL SORNGI(IGS,IGH,IGO,IGSCIS,ICIS,GAUCHH,CISH,CISS)          CORCIG
263.         87 IF(IGS.EQ.0)GO TO 95                                           CORCIG   95
264.      C6.0     RING AND/OR SECOND-ORDER INTERACTIONS PRESENT. ESTABLISH THE CORCIG
265.      C        NUMERICAL ORDER OF THESE DATA.                              CORCIG
266.            CALL ORDER(IGS,IM,KTGAUS)                                      CORCIG
267.            IF(IGO.EQ.0)GO TO 95                                           CORCIG   95
268.      C7.0     GAUCHE ETHER STRUCTURES PRESENT. SEARCH FOR THE PRESENCE OF CORCIG
269.      C        DITERTIARY ETHER STRUCTURES.                                CORCIG
270.            CALL DITER(IGS,IGO,IDTE,DTEM)                                  CORCIG
271.      C8.0     FIND THE LIGAND ATOMIC COMPOSITION OF EACH CORE ATOM AND THE CORCIG
272.      C        SYMMETRY ELEMENTS OF EACH NON-RING CORE ATOM IN THE MOLECULE. CORCIG
273.         95 JW=5                                                           CORCIG
274.            JY=5                                                           CORCIG
275.            CALL SYMTRY(LX)                                                CORCIG
276.      C9.0     FIND NUMBER OF ASYMMETRIC ATOMS PRESENT.                    CORCIG
277.            CALL ASYMC(NASYMC)                                             CORCIG
278.      C10.0    FIND LONGEST CHAIN IN MOLECULE.                             CORCIG
279.            CALL MAXCHN(LX,NASYMC)                                         CORCIG
280.      C11.0    FIND EXTERNAL ROTATIONAL SYMMETRY OF MOLECULE.              CORCIG
281.            CALL EXTROT(NDSYC,NASYMC,NPSUDA,KCSUDA,NOMESO)                 CORCIG
282.      C12.0    COMPUTE INTERNAL ROTATIONAL ENTROPY CONTRIBUTIONS (IF ANY). CORCIG
283.            CALL INTROT(IGS,INR,ROTINS)                                    CORCIG
284.            IRCMD=IRC-MD-ME                                                CORCIG
285.            KSUM=IGS+IRCMD                                                 CORCIG
286.            IF(KSUM.EQ.0)GO TO 251                                         CORCIG   251
287.      C13.0    PRINT OUT DATA FOR RING CORRECTIONS AND SECOND-ORDER RING   CORCIG
288.      C        INTERACTIONS.                                               CORCIG
289.      C          PRINT OUT TITLE.                                          CORCIG
290.            CALL PRINT2(NTITLE)                                            CORCIG
291.            IF(IRCMD.EQ.0)GO TO 171                                        CORCIG   171
292.      C          PRINT OUT DATA FOR RING CORRECTIONS.                      CORCIG
293.            DO169   K=1,IRC                                                CORCIG
```

17 APR 73 0.02-36

```
294.          LI=RINGLI(K)                                                    CORCIG
295.          IF(LI.EQ.0)GO TO 169                                            CORCIG
296.          HRING=HRING+RINGD(1,LI)                                         CORCIG
297.          SRING=SRING+RINGD(2,LI)                                         CORCIG
298.          DO161   KK=1,9                                                  CORCIG
299.      161 CPSYM(KK)=CPSYM(KK)+RINGD(KK+2,LI)                              CORCIG
300.          WRITE(6,165)(TYPE(L,I),L=1,4),K,(RINGD(KK,LI),KK=1,6)           CORCIG
301.      165 FORMAT(1H ,4A4,9X,I4,12X,1H1,F13.3,4X,F13.3,3X,4E13.4)          CORCIG
302.      169 CONTINUE                                                        CORCIG
303.      171 IF(IGS.EQ.0)GO TO 251                                           CORCIG
304.    C           THERE ARE ADDITIONAL RING CORRECTIONS AND/OR SECOND-ORDER  CORCIG
305.    C           INTERACTIONS PRESENT IN THE MOLECULE. PRINT PERTINENT DATA. CORCIG
306.          IF(IRNG2.EQ.0)GO TO 177                                         CORCIG
307.          DO175   K=1,IRNG2                                               CORCIG
308.    C           THERE ARE FUSED CARBON RING CORRECTIONS. PRINT OUT DATA.   CORCIG
309.          IBX=IM(K)                                                       CORCIG
310.          I=KFGAUS(IBX)                                                   CORCIG
311.          WRITE(6,173)(TYPE(L,I),L=1,4),KCGAUS(1,IBX),KCGAUS(2,IBX),      CORCIG
312.         1(RINGD(L,I),L=1,6)                                              CORCIG
313.      173 FORMAT(1H ,4A4,7X,2I4,10X,1H1,F13.3,4X,F13.3,3X,4E13.4)         CORCIG
314.      175 CONTINUE                                                        CORCIG
315.      177 IF(IRNG3.EQ.0)GO TO 183                                         CORCIG
316.    C           THERE ARE FUSED NITROGEN RING CORRECTIONS. PRINT OUT DATA. CORCIG
317.          KI=IRNG2+1                                                      CORCIG
318.          KF=IRNG2+IRNG3                                                  CORCIG
319.          DO181   K=KI,KF                                                 CORCIG
320.          IBX=IM(K)                                                       CORCIG
321.          WRITE(6,179)(TYPE(L,I),L=1,4),(KCGAUS(L,IBX),L=1,3),RINGD(1,49) CORCIG
322.      179 FORMAT(1H ,4A4,5X,3I4,8X,1H1,F13.3,4X,F13.3,3X,4E13.4)          CORCIG
323.      181 CONTINUE                                                        CORCIG
324.      183 IF(IORTHO.  0 TO 189                                            CORCIG
325.    C           THERE ARE ORTHO BENZENE CORRECTIONS. PRINT OUT DATA.       CORCIG
326.          KI=IRNG2+IRNG3+1                                                CORCIG
327.          KF=IRNG2+IRNG3+IORTHO                                           CORCIG
328.          DO187   K=KI,KF                                                 CORCIG
329.          IBX=IM(K)                                                       CORCIG
330.          WRITE(6,185)(TYPE(L,2),L=1,4),KCGAUS(1,IBX),KCGAUS(2,IBX),      CORCIG
331.         1(ORTHO(KK),KK=1,6)                                              CORCIG
332.      185 FORMAT(1H ,4A4,7X,2I4,10X  1,F13.3,4X,F13.3,3X,4E13.4)          CORCIG
333.      187 CONTINUE                                                        CORCIG
334.      189 IF(IORPAR.EQ.0)GO TO 195                                        CORCIG
335.    C           THERE ARE ORTHO AND/OR PARA PYRIDINE CORRECTIONS. PRINT    CORCIG
336.    C           OUT DATA.                                                  CORCIG
337.          KI=IRNG2+IRNG3+IORTHO+1                                         CORCIG
338.          KF=KI-1+IORPAR                                                  CORCIG
339.          DO193   K=KI,KF                                                 CORCIG
340.          IBX=IM(K)                                                       CORCIG
341.          J=KTGAUS(IBX)                                                   CORCIG
342.          WRITE(6,191)(TYPE(L,J),L=1,4),KCGAUS(1,IBX),KCGAUS(2,IBX),IORPAR CORCIG
343.      191 FORMAT(1H ,4A4,/X,2I4,10X,1H1,F13.3)                            CORCIG
344.      193 CONTINUE                                                        CORCIG
345.      195 IF(IGSCIS.EQ.0)GO TO 211                                        CORCIG
346.    C           THERE ARE CIS CORRECTIONS. PRINT OUT DATA.                 CORCIG
347.          ICIS=ICIS                                                       CORCIG
348.          DO199   L=1,9                                                   CORCIG
349.      199 CPSYM(L)=CPSYM(L)+ICIS*CPCIS(L)                                 CORCIG
350.          KI=IRNG2+IRNG3+IORTHO+IORPAR+1                                  CORCIG
351.          KF=KI-1+IGSCIS                                                  CORCIG
352.          DO207   K=KI,KF                                                 CORCIG
353.          IBX=IM(K)                                                       CORCIG
354.          DO201   L=1,9                                                   CORCIG
355.      201 CPALT(L)=FLOAT(KFGAUS(IBX))*CPCIS(L)                            CORCIG
356.          WRITE(6,205)(TYPE(L,4),L=1,4),KCGAUS(1,IBX),KCGAUS(2,IBX),      CORCIG
357.         1KFGAUS(IBX),DATCIS(1,IBX),DATCIS(2,IBX),(CPALT(L),L=1,4)        CORCIG
358.      205 FORMAT(1H ,4A4,7X,2I4,I11,F13.3,4X,F13.3,3X,4E13.4)             CORCIG
359.      207 CONTINUE                                                        CORCIG
360.      211 IF(IGH.EQ.0)GO TO 239                                           CORCIG
361.    C           THERE ARE GAUCHE CORRECTIONS. PRINT OUT DATA.              CORCIG
362.          KI=IRNG2+IRNG3+IORTHO+IORPAR+IGSCIS+1                           CORCIG
363.          KF=KI-1+IGH                                                     CORCIG
364.          DO221   K=KI,KF                                                 CORCIG
365.          IBX=IM(K)                                                       CORCIG
```

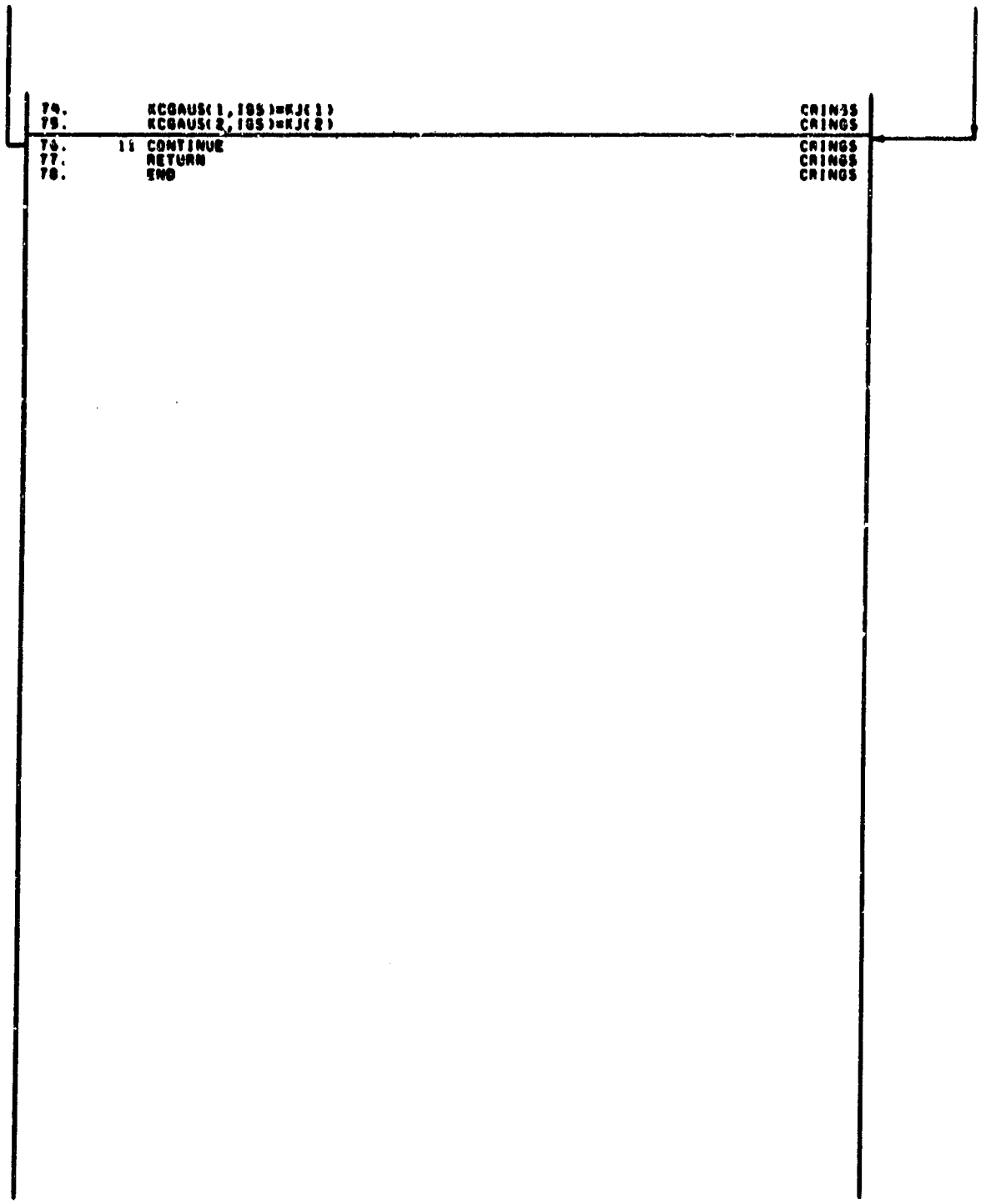
17 APR 73 0.02-36

```
      J=KTGAUS(IBX)                                                  CORCIG
      WRITE(6,219)(TYPE(L,J),L=1,4),KCGAUS(1,IBX),KCGAUS(2,IBX),     CORCIG
     1KFGAUS(IBX),DATCIS(1,IBX)                                      CORCIG
  219 FORMAT(1H ,4A4,7X,2I4,I11,F13.3)                               CORCIG
  221 CONTINUE                                                       CORCIG
      IF(IDTE.EQ.0)GO TO 239                                         CORCIG
C          THERE ARE DITERTIARY ETHER STRUCTURES. PRINT OUT DATA.    CORCIG
      KI=IRNG2+IRNG3+IORTHO+IORPAR+IGSCIS+IGH+1                      CORCIG
      KF=KI-1+IDTE                                                   CORCIG
      DO235   K=KI,KF                                                CORCIG
      WRITE(6,231)(TYPE(L,8),L=1,4),(KCGAUS(L,K),L=1,3),KFGAUS(K),   CORCIG
     1DATCIS(1,K)                                                    CORCIG
  231 FORMAT(1H ,4A4,5X,3I4,I9,F13.3)                                CORCIG
  235 CONTINUE                                                       CORCIG
  239 IF(INR.EQ.0)GO TO 251                                          CORCIG
C          THERE ARE INTERNAL ROTATION CONTRIBUTIONS. PRINT OUT DATA. CORCIG
      KI=IGS-INR+1                                                   CORCIG
      DO243   K=KI,IGS                                               CORCIG
      WRITE(6,241)KCGAUS(1,K),DATCIS(2,K)                            CORCIG
  241 FORMAT(18H INTERNAL ROTATION,6X,I6,12X,1H1,17X,F13.3)          CORCIG
  243 CONTINUE                                                       CORCIG
C14.0     FIND CONTRIBUTIONS TO THE ENTROPY ARISING FROM INTERNAL AND CORCIG
C         EXTERNAL ROTATIONAL SYMMETRY AND OPTICAL ISOMERISM.        CORCIG
  251 CALL ENTSYM(NOSN,NENAN,MESO,MASYMC,NOSNC,NPSUDA,KCSUDA,NOMESO,  CORCIG
     1SSN,SOPTS)                                                     CORCIG
C15.0     FIND SUMS OF ALL AFOREMENTIONED CONTRIBUTIONS TO THE HEAT OF CORCIG
C         FORMATION AND ENTROPY.                                     CORCIG
      HSYM=CISH+GAUCHH+DTEH+HRINR                                    CORCIG
      SSYM=CISS+ROTINS+SRING+SSN+SOPTS                               CORCIG
      RETURN                                                         CORCIG
      END                                                            CORCIG
```

17 APR 73 0.02-36

```
      SUBROUTINE CRINGS(IGS,IRNG2,RDATA,NRING,SRING,CPSYM)                  CRINGS
C        THIS SUBROUTINE SEARCHES FOR CERTAIN FUSED CARBON RING SYSTEMS.    CRINGS
C        IF PRESENT, IT APPLIES APPROPRIATE RING CORRECTIONS TO THE         CRINGS
C        THERMODYNAMIC PROPERTIES.                                          CRINGS
      INTEGER WEIGHT(9)                                                     CRINGS
      DIMENSION RDEL(40),KOMB(40,10),NOKOMB(11),CPSYM(9),RDATA(6,7),        CRINGS
     1KRPROP(6,40),NOK(10,10),KJ(2),KTGAUS(150),KFGAUS(150),                CRINGS
     2KCGAUS(3,150)                                                         CRINGS
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC         CRINGS
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),      CRINGS
     1IORR(100),IB(100,8),IRG,NORR                                          CRINGS
      COMMON/BLK4/NDC(60,50),NBS(60,2),NRX(60,20),IRC,NONFUS,IRCTOT         CRINGS
      EQUIVALENCE (NO,IB(12,1)),(NDEL(1),IB(13,1)),(KOMB(1,1),IB(62,4)),    CRINGS
     1(NOKOMB(1),IB(80,8)),(NOK(1,1),IMATX(1,55)),(KRPROP(1,1),             CRINGS
     2NBX(2,1)),(KTGAUS(1),IMATX(1,60)),(KFGAUS(1),IMATX(1,63)),            CRINGS
     3(KCCAUS(1,1),IMATX(1,72)),(KJ(1),NBX(28,20))                          CRINGS
C1.0     INITIALIZE CYCLE THAT TESTS EACH FUSED RING SET.                   CRINGS
      IC=NOKOMB(1)                                                          CRINGS
      DO11   I=1,IC                                                         CRINGS
      IF(NOKOMB(I+1).NE.2)GO TO 11                                          CRINGS
C2.0     THERE ARE TWO RINGS IN FUSED RING SET I.                           CRINGS
      K1=KOMB(1,I)                                                          CRINGS
      K2=KOMB(2,I)                                                          CRINGS
C3.0     FIND THE FIRST 3-CENTERED RING (IF ANY) IN SET I.                  CRINGS
      IF(IRING(K1,1).NE.3)GO TO 1                                           CRINGS
C           RING 1 IS 3-CENTERED.                                           CRINGS
      KJ(1)=K1                                                              CRINGS
      KJ(2)=K2                                                              CRINGS
      GO TO 9                                                               CRINGS
    1 IF(IRING(K2,1).NE.3)GO TO 11                                          CRINGS
C           RING 2 IS 3-CENTERED.                                           CRINGS
      KJ(1)=K2                                                              CRINGS
      KJ(2)=K1                                                              CRINGS
C4.0     AT LEAST ONE OF THE TWO RINGS IS 3-CENTERED. NOW DETERMINE         CRINGS
C        OTHER RING PROPERTIES.                                             CRINGS
    3 KSU[illegible]RP(1,K1)+KRPROP(3,K1)+KRPROP(5,K1)+KRPROP(1,K2)+        CRINGS
     1KRPROP(3,K2)+KRPROP(5,K2)                                             CRINGS
      IF(KSUM.NE.0)GO TO 11                                                 CRINGS
C5.0     BOTH RINGS ARE SATURATED CARBON BASED RINGS.                       CRINGS
      N1=KJ(2)                                                              CRINGS
      N2=IRING(N1,1)                                                        CRINGS
C6.0     HOW MANY COMMON ATOMS DO RINGS POSSESS                             CRINGS
      IF(NOK(2,I)-2)5,7,11                                                  CRINGS
C           RINGS HAVE 1 ATOM IN COMMON.                                    CRINGS
    5 IF(N2.NE.3)GO TO 11                                                   CRINGS
C              SECOND RING IS ALSO 3-CENTERED. SET RING CORRECTION FLAG     CRINGS
C              FOR SPIROPENTANE.                                            CRINGS
      N3=1                                                                  CRINGS
      GO TO 9                                                               CRINGS
C           RINGS HAVE 2 ATOMS IN COMMON.                                   CRINGS
    7 IF(N2.GT.8)GO TO 11                                                   CRINGS
C              SECOND RING IS 3,4,5,6,7, OR 8-CENTERED. SET RING            CRINGS
C              CORRECTION FLAG FOR BICYCLO-(1,1,0)-BUTANE,                  CRINGS
C              -(2,1,0)-PENTANE, -(3,1,0)-HEXANE, -(4,1,0)-HEPTANE,         CRINGS
C              -(5,1,0)-OCTANE, OR -(6,1,0)-NONANE.                         CRINGS
      N3=N2-1                                                               CRINGS
C4.0     STORE TWO RING NUMBERS IN RDEL.                                    CRINGS
    9 NO=NO+1                                                               CRINGS
      RDEL(NO)=KJ(1)                                                        CRINGS
      NO=NO+1                                                               CRINGS
      RDEL(NO)=KJ(2)                                                        CRINGS
C8.0     SET I IS ONE OF THE AFOREMENTIONED FUSED CARBON RING SYSTEMS.      CRINGS
C        ADD CORRECTIONS TO THERMODYNAMIC PROPERTIES, STORE PERTINENT       CRINGS
C        PRINTOUT DATA, AND CONTINUE THE TEST OF OTHER FUSED RING SETS.     CRINGS
      HRING=HRING+RDATA(1,N3)                                               CRINGS
      SRING=SRING+RDATA(2,N3)                                               CRINGS
      DO10   KK=1,9                                                         CRINGS
      CPSYM(KK)=CPSYM(KK)+RDATA(KK+2,N3)                                    CRINGS
   10 CONTINUE                                                              CRINGS
      IRNG2=IRNG2+1                                                         CRINGS
      IGS=IGS+1                                                             CRINGS
      KTGAUS(IGS)=1                                                         CRINGS
      KFGAUS(IGS)=N3+23                                                     CRINGS
```


IF A/3 73 G.02-36

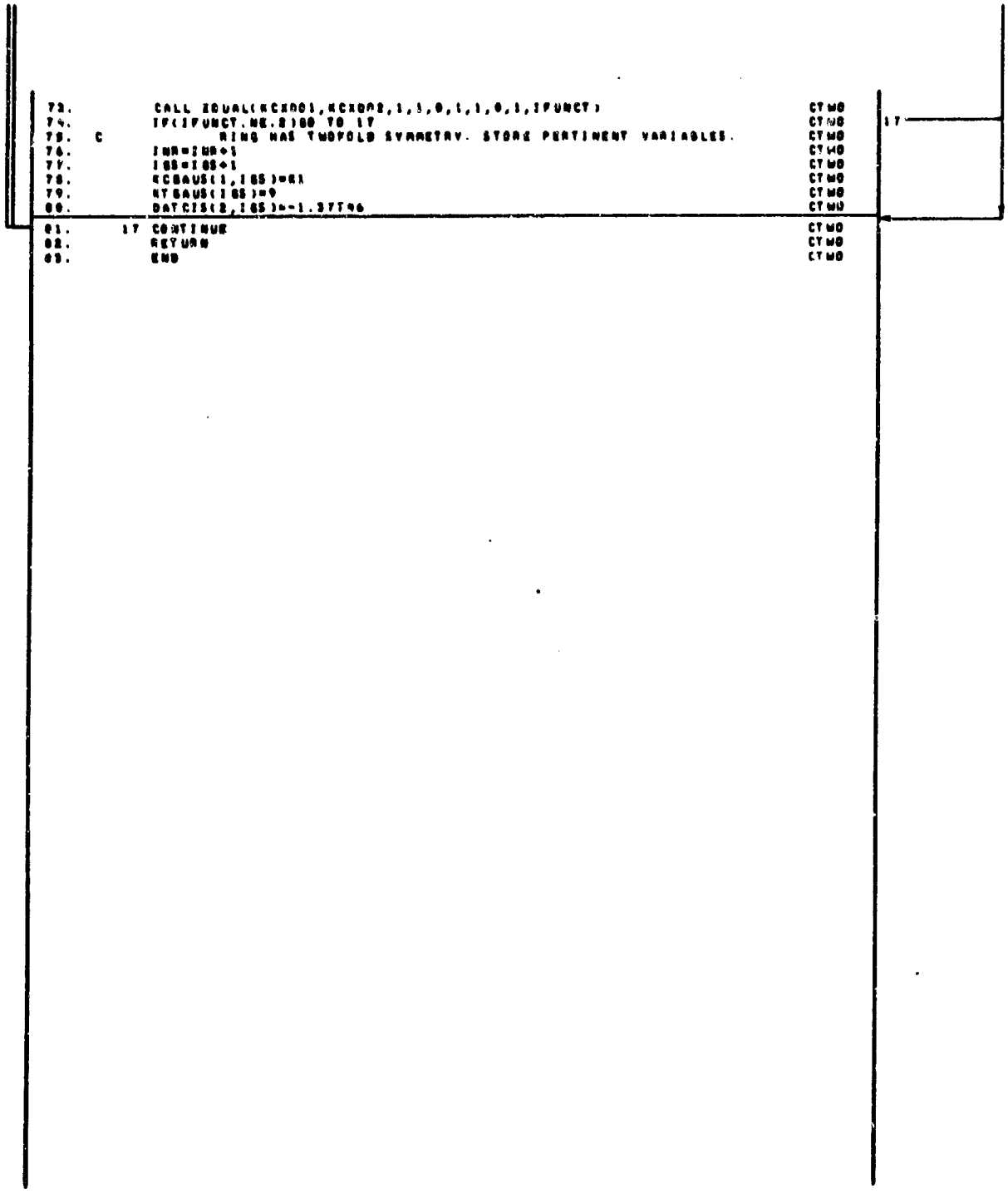


```
      KCGAUS(1,IGS)=KJ(1)                                    CRINGS
      KCGAUS(2,IGS)=KJ(2)                                    CRINGS
   11 CONTINUE                                               CRINGS
      RETURN                                                 CRINGS
      END                                                    CRINGS
```

17 APR 73 8.02-36

```
      SUBROUTINE CTWO(IGS,INR)                                          CTWO
C        THIS SUBROUTINE CHECKS FOR THE PRESENCE OF NO2 GROUPS AND      CTWO
C        MONOCYCLIC AROMATIC RINGS WHICH EXHIBIT TWOFOLD INTERNAL       CTWO
C        ROTATIONAL SYMMETRY ABOUT AN AXIS WITH A NONLINEAR             CTWO
C        CONFIGURATION. IF PRESENT, IT STORES THE PERTINENT I.D.        CTWO
C        NUMBERS AND CONTRIBUTIONS TO THE ENTROPY.                      CTWO
      INTEGER SYMM(4),SYMBOL(9),GRID(50,80)                             CTWO
      INTEGER WEIGHT(9)                                                 CTWO
      DIMENSION KCRDO1(3),KCGAUS(3,150),DATCIS(2,150),KTGAUS(150),      CTWO
     1KTBENZ(40)                                                        CTWO
      COMMON/BLK1/ND,NOS,SYMM,SYMBOL,MOVAL(9),GRID                      CTWO
      COMMON/BLK2/WEIGHT,NWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC     CTWO
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),  CTWO
     1IDBR(100),IB(100,8),IRG,NOBR                                      CTWO
      EQUIVALENCE (KCGAUS(1,1),IMATX(1,72)),(DATCIS(1,1),IMATX(1,66)),  CTWO
     1(NTBENZ,IB(96,8)),(KTBENZ(1),GRID(41,27)),(KTGAUS(1),IMATX(1,60)) CTWO
C1.0     CHECK FOR NO2 GROUPS WHICH ARE NOT BONDED TO LINEAR LIGANDS.   CTWO
      DO3    I=1,KCC                                                    CTWO
      IF(NC(I).EQ.KON(I))GO TO 3                                        CTWO
      JI=NC(I)+2                                                        CTWO
      JF=KON(I)+1                                                       CTWO
      DO1    J=JI,JF                                                    CTWO
      IF(IX(I,J,1).NE.9)GO TO 1                                         CTWO
      IF(IBC(I).NE.0)GO TO 100                                          CTWO
C           NO2 GROUP PRESENT.                                          CTWO
      CALL LINEAR(1000,I,LINE,KSYM)                                     CTWO
      IF(LINE.EQ.1)GO TO 1                                              CTWO
C           NO2 LIGAND NONLINEAR.                                       CTWO
  100 INR=INR+1                                                         CTWO
      IGS=IGS+1                                                         CTWO
      KCGAUS(1,IGS)=I                                                   CTWO
      KTGAUS(IGS)=9                                                     CTWO
      DATCIS(2,IGS)=-1.37746                                            CTWO
    1 CONTINUE                                                          CTWO
    3 CONTINUE                                                          CTWO
C2.0     IF MONOCYCLIC AROMATIC RINGS ARE NOT PRESENT, RETURN. OTHERWISE CTWO
C        CONTINUE.                                                      CTWO
      IF(NTBENZ.EQ.0)RETURN                                             CTWO
C3.0     INITIALIZE MAIN CYCLE THAT SEARCHES FOR THE PRESENCE OF        CTWO
C        INTERNAL TWOFOLD SYMMETRY.                                     CTWO
      DO17   N=1,NTBENZ                                                 CTWO
      K=KTBENZ(N)                                                       CTWO
C           TEST ALL COMPONENTS OF RING K.                              CTWO
      DO17   J=2,7                                                      CTWO
      K1=IRING(K,J)                                                     CTWO
      IF(KON(K1).LE.2)GO TO 17                                          CTWO
C           CONNECTIVITY OF ATOM K1 IS GREATER THAN 2.                  CTWO
      IO=1                                                              CTWO
      JF=KON(K1)+1                                                      CTWO
C           FIND THE RING AND NON-RING LIGANDS OF RING ATOM K1.         CTWO
      DO11   I=2,JF                                                     CTWO
      KC=IX(K1,I,6)                                                     CTWO
      IF(KC.EQ.0)GO TO 7                                                CTWO
      IF(IBC(KC).NE.IBC(K1))GO TO 5                                     CTWO
C           LIGAND IS A RING ATOM. STORE ITS GROUP NUMBER.              CTWO
      IO=IO+1                                                           CTWO
      IF(IO.GT.3)GO TO 17                                               CTWO
      KCRDO1(IO)=KC                                                     CTWO
      GO TO 11                                                          CTWO
C           LIGAND IS A NON-RING CORE ATOM. IF NONLINEAR, TRANSFER TO   CTWO
C           9. OTHERWISE, TRANSFER TO 17.                               CTWO
    5 IF(IBC(KC).NE.0)GO TO 9                                           CTWO
      CALL LINEAR(K1,KC,LINE,KSYM)                                      CTWO
      IF(LINE-1)9,17,9                                                  CTWO
C           LIGAND IS A NON-RING NON-CORE ATOM. IF NONLINEAR, (I.E.,    CTWO
C           NO OR NO2) CONTINUE. OTHERWISE, TRANSFER TO 17.             CTWO
    7 IF(IX(K1,4,1).LT.0)GO TO 11                                       CTWO
    9 KCRDO1(1)=K1                                                      CTWO
      KCRDO2=KCRDO1(1)                                                  CTWO
   11 CONTINUE                                                          CTWO
      IFUNCT=1                                                          CTWO
C           DETERMINE IF RING IS SYMMETRICAL.                           CTWO
```

19 FEB 73 8:02:39



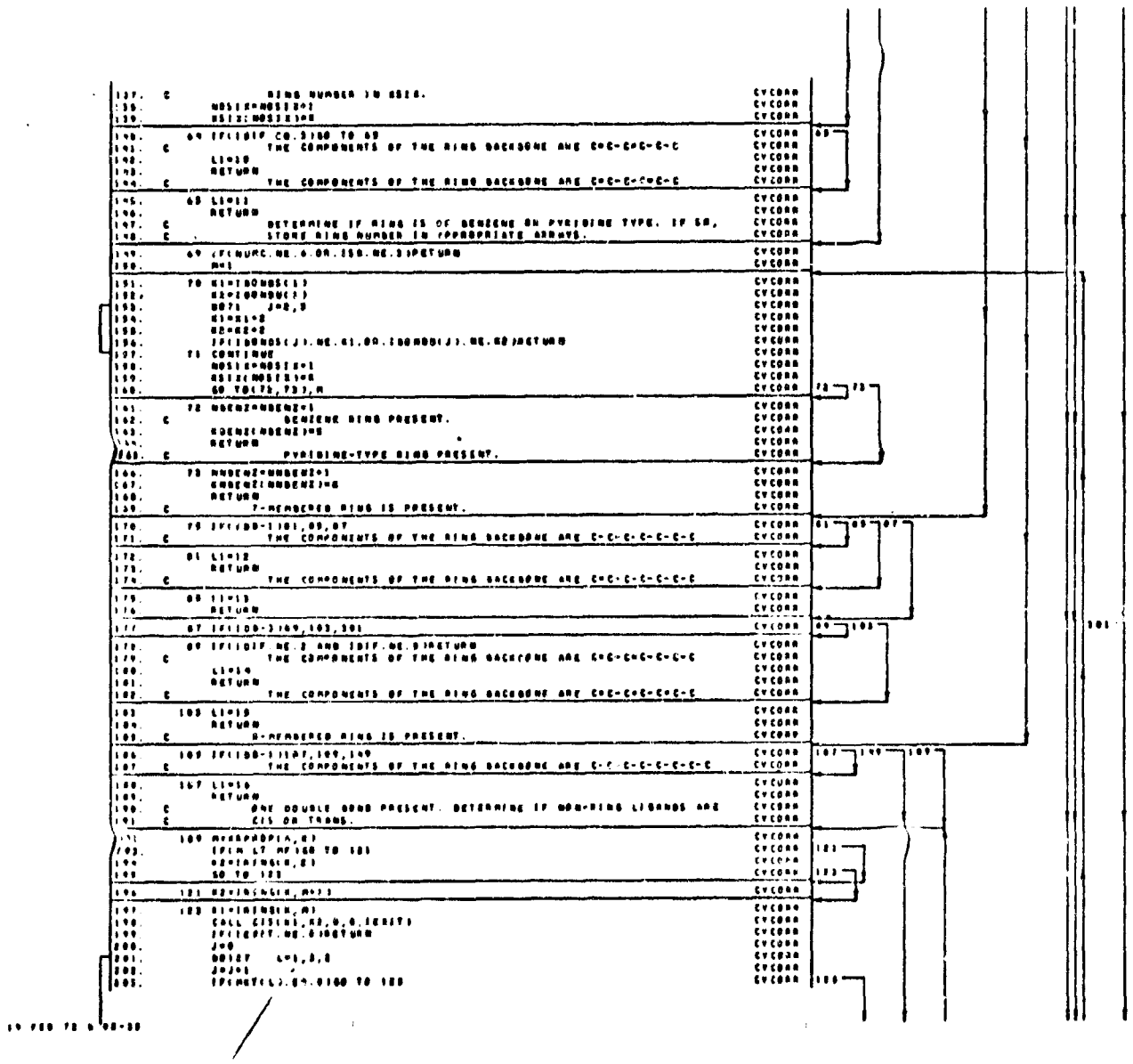
```
      CALL EQUAL(KCXR01,KCXR02,1,1,0,1,1,0,1,IFUNCT)                    CTMB
      IF(IFUNCT.NE.2)GO TO 17                                           CTMB
C           RING HAS TWOFOLD SYMMETRY. STORE PERTINENT VARIABLES.       CTMB
      IWR=IWR+1                                                         CTMB
      IGS=IGS+1                                                         CTMB
      KCSAVS(1,IGS)=K1                                                  CTMB
      NTSAVS(IGS)=9                                                     CTMB
      DATCIS(2,IGS)=-1.37796                                            CTMB
   17 CONTINUE                                                          CTMB
      RETURN                                                            CTMB
      END                                                               CTMB
```

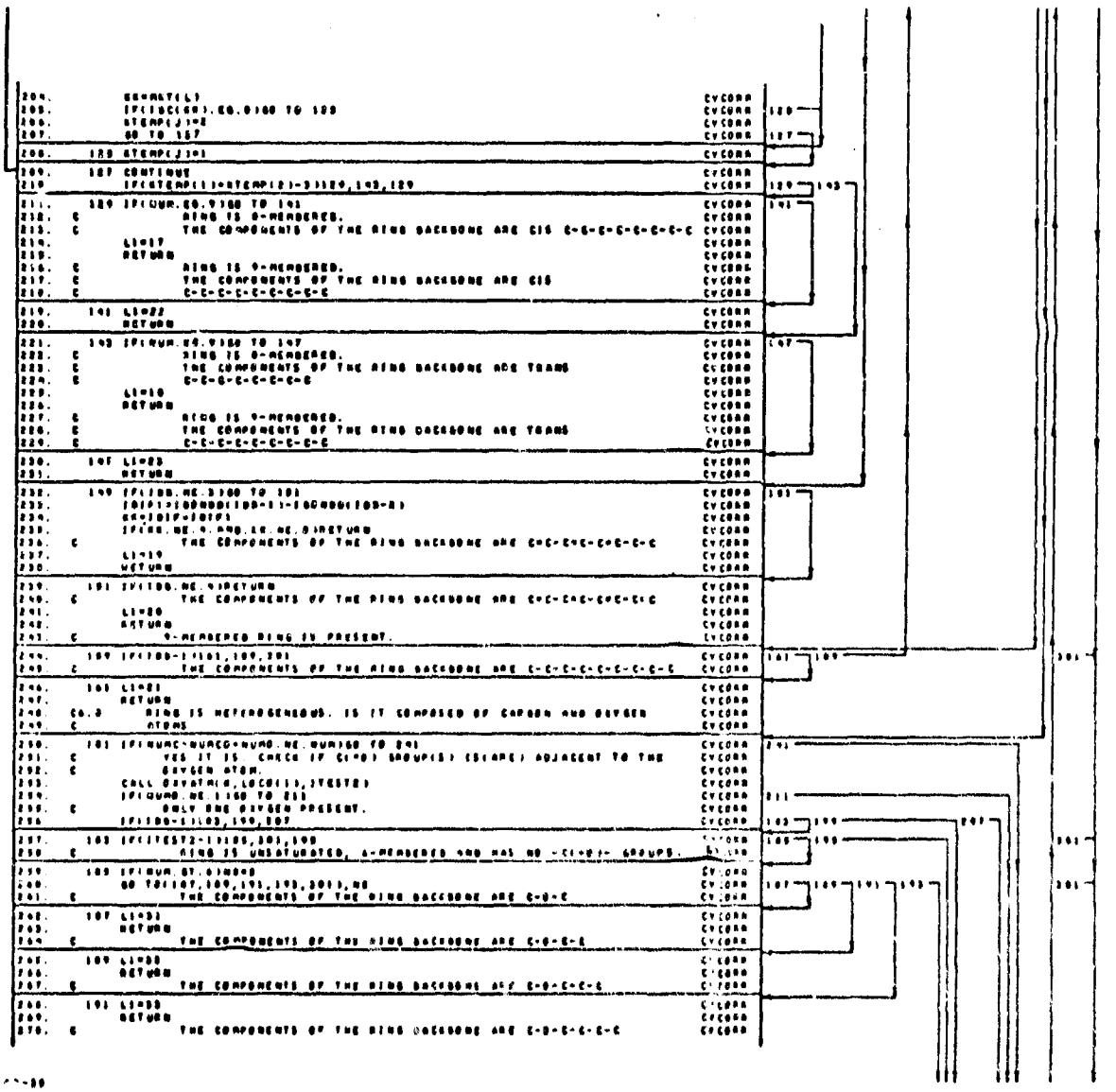
19 FEB 73 8.02-38

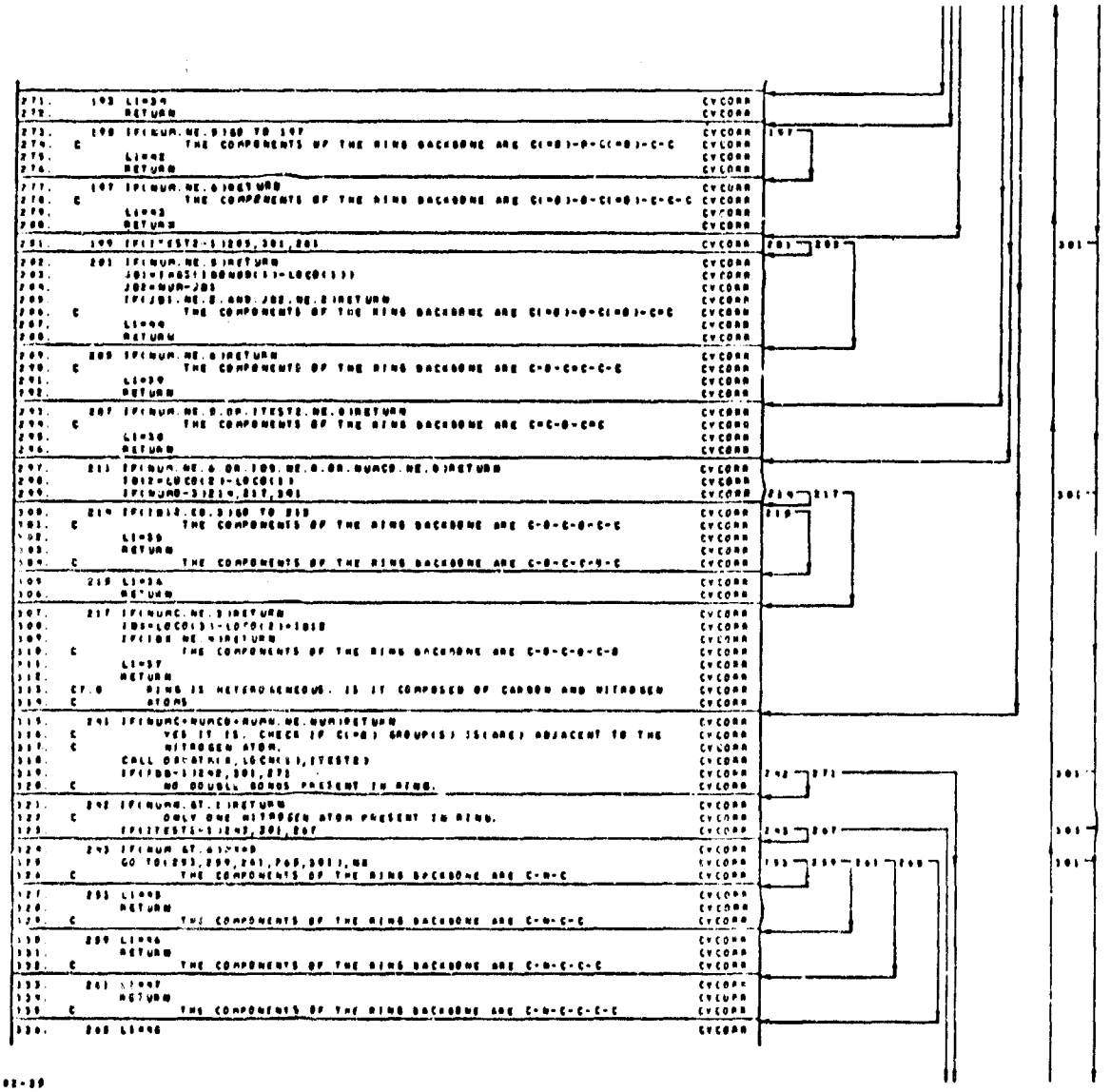
```
      SUBROUTINE CYCORR(K,LI)                                                CYCORR
C        THIS SUBROUTINE DETERMINES THE TYPE OF RING CORRECTION THAT         CYCORR
C        IS TO BE APPLIED TO A NON-FUSED RING. IT ALSO SETS VARIOUS          CYCORR
C        RING ARRAYS AND OTHER INDICATORS, IF BENZENE OR PYRIDINE            CYCORR
C        TYPE RINGS ARE PRESENT.                                             CYCORR
      INTEGER SYMX(4),SYMBOL(9),GRID(50,80)                                  CYCORR
      INTEGER WEIGHT(9)                                                      CYCORR
      DIMENSION MS(4),MRT(4),XTEMP(2),IBONDS(30),IBONDD(30),                 CYCORR
     1KBENZ(40),KSIX(40),LOCO(30),LOCN(30),LOCCO(30),KRPROP(6,40),           CYCORR
     2MVW(2),NRING(40),KRCNOR(100),KRCNWT(100),KNBENZ(40)                    CYCORR
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                           CYCORR
      COMMON/BLK2/WEIGHT,MWBT(9),MOLWT(100),IX(100,9,6),NC(100),KCC          CYCORR
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),       CYCORR
     1IDBR(100),IB(100,8),IRB,NDBR                                           CYCORR
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IAC,NONFUS,IRCTOT          CYCORR
      EQUIVALENCE (MS(1),IB(2,1)),(MRT(1),IB(6,1)),(XTEMP(1),IB(93,1)),      CYCORR
     1(IBONDS(1),IB(2,3)),(IBONDD(1),IB(51,3)),(NUMC,IB(76,2)),(NUMO,        CYCORR
     2IB(77,2)),(LOCO(1),IMATX(1,47)),(NUMN,IB(78,2)),(LOCN(1),              CYCORR
     3IMATX(21,46)),(NUMCO,IB(79,2)),(LOCCO(1),IMATX(41,45)),(NBENZ,         CYCORR
     4IB(10,1)),(NOSIX,IB(11,1)),(KBENZ(1),IB(53,1)),(KSIX(1),IB(81,2)),     CYCORR
     5(KRPROP(1,1),NBX(2,1)),(MVW(1),GRID(41,75)),(NRING(1),GRID(1,75)),     CYCORR
     6(KRCNOR(1),GRID(1,76)),(KRCNWT(1),GRID(1,78)),(KNBENZ,GRID(1,80)),     CYCORR
     7(KNBENZ(1),GRID(2,80))                                                 CYCORR
C1.0     INITIALIZE VARIABLES.                                               CYCORR
      LI=0                                                                   CYCORR
      DO1   M=1,6                                                            CYCORR
      KRPROP(M,K)=0                                                          CYCORR
    1 CONTINUE                                                               CYCORR
      NUM=IRING(K,1)                                                         CYCORR
      IF(NUM.GT.9)RETURN                                                     CYCORR
      MF=NUM+1                                                               CYCORR
      IDIF=0                                                                 CYCORR
      ISB=0                                                                  CYCORR
      IDB=0                                                                  CYCORR
      NUMC=0                                                                 CYCORR
      NUMO=0                                                                 CYCORR
      NUMN=0                                                                 CYCORR
      NUMCO=0                                                                CYCORR
C2.0     FIND NUMBER OF CARBON, OXYGEN, NITROGEN, -C(=O)-, SINGLE BONDS,     CYCORR
C        AND DOUBLE BONDS PRESENT IN RING BACKBONE AND THE LOCATION OF       CYCORR
C        EACH ONE, EXCEPT FOR THE CARBON ATOMS.                              CYCORR
      DO21    M=1,MF                                                         CYCORR
      K1=IRING(K,M)                                                          CYCORR
      IF(IX(K1,1,1).NE.2)GO TO 3                                             CYCORR
      NUMC=NUMC+1                                                            CYCORR
      GO TO 9                                                                CYCORR
    3 IF(IX(K1,1,1).NE.3)GO TO 5                                             CYCORR
      NUMO=NUMO+1                                                            CYCORR
      LOCO(NUMO)=M                                                           CYCORR
      GO TO 9                                                                CYCORR
    5 IF(IX(K1,1,1).NE.4)GO TO 7                                             CYCORR
      NUMN=NUMN+1                                                            CYCORR
      LOCN(NUMN)=M                                                           CYCORR
      GO TO 9                                                                CYCORR
    7 IF(IX(K1,1,1).NE.6)RETURN                                              CYCORR
      NUMCO=NUMCO+1                                                          CYCORR
      LOCCO(NUMCO)=M                                                         CYCORR
    9 IF(M.LT.MF)GO TO 11                                                    CYCORR
      K2=IRING(K,2)                                                          CYCORR
      GO TO 13                                                               CYCORR
   11 K2=IRING(K,M+1)                                                        CYCORR
   13 KONMAX=NC(K1)+1                                                        CYCORR
      DO15    N=2,KONMAX                                                     CYCORR
      IF(IX(K1,N,6).EQ.K2)GO TO 17                                           CYCORR
   15 CONTINUE                                                               CYCORR
   17 IF(IX(K1,N,5).NE.1)GO TO 19                                            CYCORR
      ISB=ISB+1                                                              CYCORR
      IBONDS(ISB)=M                                                          CYCORR
      GO TO 21                                                               CYCORR
   19 IF(IX(K1,N,5).NE.2)GO TO 21                                            CYCORR
      IDB=IDB+1                                                              CYCORR
```

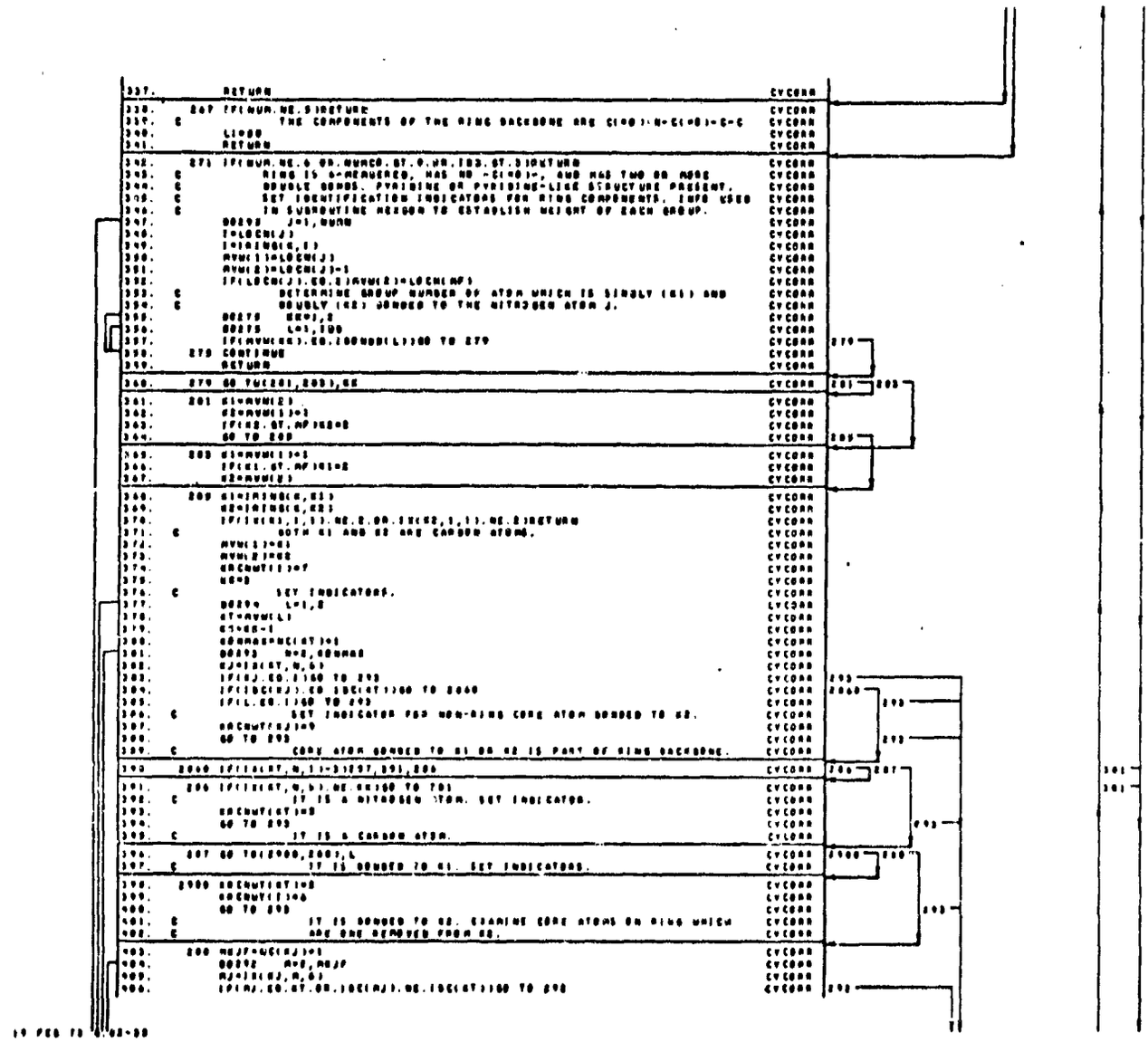
```
      IBONDD(IDD)=M                                                     CVCORR
   21 CONTINUE                                                          CVCORR
C3.0     IF MORE THAN ONE DOUBLE BOND IS PRESENT, FIND SEPARATION       CVCORR
C        BETWEEN THE FIRST AND SECOND.                                  CVCORR
      IF(IDD.GT.1)IDIF=IBONDD(IDD)-IBONDD(IDD-1)                        CVCORR
C4.0     STORE VARIABLES ASSOCIATED WITH RING 6 IN KRPROP.              CVCORR
      KRPROP(1,6)=NUMR                                                  CVCORR
      KRPROP(2,6)=LOCO(1)                                               CVCORR
      KRPROP(3,6)=NUMN                                                  CVCORR
      KRPROP(4,6)=LOCN(1)                                               CVCORR
      KRPROP(5,6)=IDD                                                   CVCORR
      KRPROP(6,6)=IBONDD(1)                                             CVCORR
      NS=NUM-2                                                          CVCORR
      IF(NUMC+NUMCO-NUM)101,25,301                                      CVCORR
C5.0      RING IS HOMOGENEOUS - ALL ATOMS IN RING BACKBONE ARE CARBON.  CVCORR
   25 IF(NUMCO-1)29,27,301                                              CVCORR
C          5- OR 6-MEMBERED RING PRESENT CONTAINING ONE -C(=O)- GROUP.  CVCORR
   27 IF(NUM.NE.5.AND.NUM.NE.6)RETURN                                   CVCORR
   29 GO TO(31,37,43,53,75,100,159),NS                                  CVCORR
C           3-MEMBERED RING IS PRESENT.                                 CVCORR
   31 IF(IDD-1)33,35,301                                                CVCORR
C              THE COMPONENTS OF THE RING BACKBONE ARE C-C-C            CVCORR
   33 LI=1                                                              CVCORR
      RETURN                                                            CVCORR
C              THE COMPONENTS OF THE RING BACKBONE ARE C=C-C            CVCORR
   35 LI=2                                                              CVCORR
      RETURN                                                            CVCORR
C           4-MEMBERED RING IS PRESENT.                                 CVCORR
   37 IF(IDD-1)39,41,301                                                CVCORR
C              THE COMPONENTS OF THE RING BACKBONE ARE C-C-C-C          CVCORR
   39 LI=3                                                              CVCORR
      RETURN                                                            CVCORR
C              THE COMPONENTS OF THE RING BACKBONE ARE C=C-C-C          CVCORR
   41 LI=4                                                              CVCORR
      RETURN                                                            CVCORR
C           5-MEMBERED RING IS PRESENT.                                 CVCORR
   43 IF(IDD-1)45,49,51                                                 CVCORR
   45 IF(NUMCO.EQ.1)GO TO 47                                            CVCORR
C              THE COMPONENTS OF THE RING BACKBONE ARE C-C-C-C-C        CVCORR
      LI=5                                                              CVCORR
      RETURN                                                            CVCORR
C              THE COMPONENTS OF THE RING BACKBONE ARE C-C-C-C-C(=O)    CVCORR
   47 LI=40                                                             CVCORR
      RETURN                                                            CVCORR
C              THE COMPONENTS OF THE RING BACKBONE ARE C=C-C-C-C        CVCORR
   49 LI=6                                                              CVCORR
      RETURN                                                            CVCORR
C              THE COMPONENTS OF THE RING BACKBONE ARE C=C-C=C-C        CVCORR
   51 LI=7                                                              CVCORR
      RETURN                                                            CVCORR
C           6-MEMBERED RING IS PRESENT.                                 CVCORR
   53 IF(IDD-1)55,59,61                                                 CVCORR
   55 IF(NUMCO.EQ.1)GO TO 57                                            CVCORR
C              THE COMPONENTS OF THE RING BACKBONE ARE C-C-C-C-C-C      CVCORR
      LI=8                                                              CVCORR
      RETURN                                                            CVCORR
C              THE COMPONENTS OF THE RING BACKBONE ARE C-C-C-C-C-C(=O)  CVCORR
   57 LI=41                                                             CVCORR
      RETURN                                                            CVCORR
C              THE COMPONENTS OF THE RING BACKBONE ARE C=C-C-C-C-C      CVCORR
   59 LI=9                                                              CVCORR
      RETURN                                                            CVCORR
   61 IF(IDD-2)63,69,301                                                CVCORR
   63 IF(NUMCO.GT.0)GO TO 69                                            CVCORR
C              RING HAS TWO DOUBLE BONDS AND NO -C(=O)- GROUPS. STORE   CVCORR
```

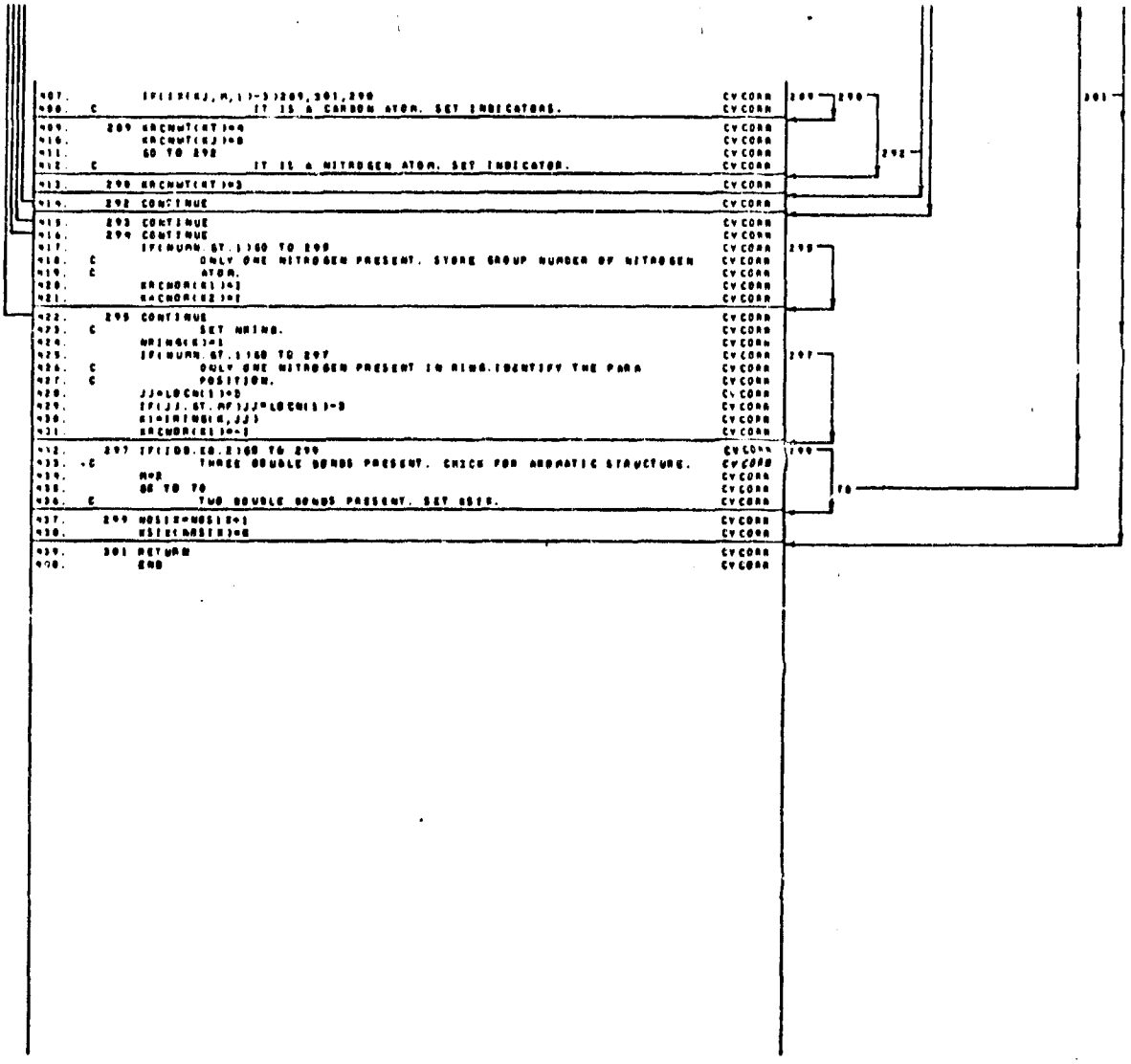
10 FEB 75 8.48-55










IF(I.EQ.J.OR.I.EQ.K)GO TO 299
IF(I.EQ.J.OR.I.EQ.K)GO TO 299
IF(NUMN.GT.1)GO TO 299
IF(NUMN.GT.1)GO TO 297
IF(IDD.EQ.2)GO TO 299
GO TO 298
GO TO 301

```
      BLOCK DATA DATA1                                                      DATA1
C        THIS BLOCK DATA CONTAINS THE DATA FROM BENSONS GROUP ADDITIVITY     DATA1
C        TABLES. THE TABULAR INPUT PER GROUP COMPRISES THE (1)CHEMICAL       DATA1
C        GROUP SYMBOL (GROUP1,GROUP2,GROUP3), (2)ASSIGNED GROUP             DATA1
C        WEIGHT(SUM), (3)HEAT OF FORMATION(HF298) AND (4)ENTROPY(S298)      DATA1
C        AT 298 DEG. K., AND THE COEFFICIENTS FOR THE HEAT CAPACITY         DATA1
C        EQUATION  CP = CP1 + CP2(T) + CP3(T)**2 + CP4(T)**3.               DATA1
      INTEGER GROUP1(180),GROUP2(180),GROUP3(180)                           DATA1
      INTEGER SUM(180)                                                      DATA1
      DIMENSION CP1(180),CP2(180),CP3(180),CP4(180)                         DATA1
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT         DATA1
      COMMON/BLK7/SUM,HF298(180),S298(180),CPX(180,4)                       DATA1
      EQUIVALENCE (GROUP1(1),NBC(2,1)),(GROUP2(1),NBC(2,12)),               DATA1
     1(GROUP3(1),NBC(2,28))                                                 DATA1
      EQUIVALENCE (CPX(1,1),CP1(1)),(CPX(1,2),CP2(1)),(CPX(1,3),CP3(1)),    DATA1
     1(CPX(1,4),CP4(1))                                                     DATA1
C1.0      DATA FOR HYDROCARBONS.                                           DATA1
      DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),          DATA1
     ACP1(L),CP2(L),CP3(L),CP4(L),L=1,9)/                                   DATA1
     B4HCHHH,4HC   ,4H    ,     296,     -10.08,     30.41,                 DATA1
     C-1.67923034E-1,   2.49328426E-2, -1.16714356E-5,   2.18035135E-9,    DATA1
     D4HCHHC,4HC   ,4H    ,     399,      -4.95,      9.42,                 DATA1
     E-2.45391108E-1,   2.28706843E-2, -1.32552624E-5,   2.96652455E-9,    DATA1
     F4HCHCC,4HC   ,4H    ,     502,      -1.90,    -12.07,                 DATA1
     G-1.72331790E+0,   2.65408083E-2, -2.04191614E-5,   5.63747170E-9,    DATA1
     H4HCCCC,4HC   ,4H    ,     605,       0.50,    -35.10,                 DATA1
     I-4.07843204E+0,   3.81256429E-2, -3.60167656E-5,   1.06820609E-8,    DATA1
     J4HOHH ,4H    ,4H    ,    1031,       6.26,     27.61,                 DATA1
     K 1.35786856E-1,   1.94481085E-2, -1.06666187E-5,   2.32949382E-9,    DATA1
     L4HOHC ,4H    ,4H    ,    1134,       8.59,      7.97,                 DATA1
     M 8.45867337E-1,   1.28226160E-2, -6.32082053E-6,   1.11437915E-9,    DATA1
     N4HOCC ,4H    ,4H    ,    1237,      10.34,    -12.70,                 DATA1
     O 2.21784832E+0,   7.71249971E-3, -4.94590091E-6,   1.09689094E-9,    DATA1
     P4HODH ,4H    ,4H    ,    2008,       6.78,      6.38,                 DATA1
     Q-1.03007557E+0,   2.38648291E-2, -1.96511772E-5,   5.76029184E-9,    DATA1
     R4HODC ,4H    ,4H    ,    2111,       8.86,    -14.60,                 DATA1
     S 3.33905417E-1,   1.87546888E-2, -1.02762576E-5,   5.74280363E-9/     DATA1
      DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),          DATA1
     ACP1(L),CP2(L),CP3(L),CP4(L),L=10,18)/                                 DATA1
     B4HOZH ,4H    ,4H    ,    2011,       6.78,      6.38,                 DATA1
     C-1.03007557E+0,   2.38648291E-2, -1.96511772E-5,   5.76029184E-9,    DATA1
     D4HOZC ,4H    ,4H    ,    2114,       8.64,    -14.60,                 DATA1
     E 3.33905417E-1,   1.87546888E-2, -1.02762576E-5,   5.74280363E-9,    DATA1
     F4HOTH ,4H    ,4H    ,    2009,       6.76,      6.38,                 DATA1
     G-1.03007557E+0,   2.38648291E-2, -1.96511772E-5,   5.76029184E-9,    DATA1
     H4HCOCH,4HH   ,4H    ,    1273,      -4.76,      9.80,                 DATA1
     I-1.97016033E+0,   2.92031432E-2, -1.99328415E-5,   5.14872893E-9,    DATA1
     J4HCOOH,4HH   ,4H    ,    2147,      -4.29,     10.20,                 DATA1
     K-3.73859403E+0,   3.57687115E-2, -2.69846033E-5,   7.46954866E-9,    DATA1
     L4HCOZH,4HH   ,4H    ,    2150,      -4.29,     10.20,                 DATA1
     M-3.73859403E+0,   3.57687115E-2, -2.69846033E-5,   7.46954866E-9,    DATA1
     N4HCTCH,4HH   ,4H    ,    1274,      -4.73,     10.30,                 DATA1
     O-1.37099386E+0,   2.94749983E-2, -1.56770613E-5,   3.74328008E-9,    DATA1
     P4HCZCH,4HH   ,4H    ,    1276,      -4.66,      9.34,                 DATA1
     Q-1.29319222E+0,   3.00487840E-2, -2.21458365E-5,   5.87065982E-9,    DATA1
     R4HCDCC,4HH   ,      ,    1376,      -1.48,     -1.69,                 DATA1
     S-3.60401992E+0,   1.37235204E-2, -2.02179473E-5,   8.23527062E-9/     DATA1
      DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),          DATA1
     ACP1(L),CP2(L),CP3(L),CP4(L),L=19,27)/                                 DATA1
     B4HCTCC,4HH   ,4H    ,    1377,      -1.72,    -11.19,                 DATA1
     C-2.03892066E+0,   2.91372022E-2, -2.28409603E-5,   6.41422727E-9,    DATA1
     D4HCZCC,4HH   ,      ,    1379,      -0.98,    -12.15,                 DATA1
     E-2.05100004E+0,   3.41308621E-2, -2.90251609E-5,   8.72526375E-9,    DATA1
     F4HCDCC,4HC   ,4H    ,    1479,       1.60,    -34.72,                 DATA1
     G-5.66570889E+0,   4.37931905E-2, -4.17270981E-5,   1.24739551E-8,    DATA1
     H4HCZCC,4HC   ,4H    ,    1482,       2.81,    -35.18,                 DATA1
     I-6.37314891E+0,   4.99863297E-2, -5.01915550E-5,   1.53979446E-8,    DATA1
     J4HTH  ,4H    ,4H    ,    1085,      26.93,     24.70,                 DATA1
     K 2.56220694E+0,   1.15236646E-2, -8.70323055E-6,   2.54450214E-9,    DATA1
     L4HTC  ,4H    ,4H    ,    1127,      27.55,      6.35,                 DATA1
     M 1.46686730E+0,   6.94339008E-3, -5.46269683E-6,   2.00196494E-9,    DATA1
     N4HTB  ,4H    ,4H    ,    2001,      29.20,      6.43,                 DATA1
     O-1.11432595E+0,   1.54395057E-2, -1.17471535E-5,   3.01714266E-9,    DATA1
     P4HTZ  ,4H    ,4H    ,    2004,      29.20,      6.43,                 DATA1
     R-1.11432595E+0,   1.54395057E-2, -1.17471535E-5,   3.01714266E-9,    DATA1
```

1- FEB 73 8.02-38

```
      A4HZH   ,4H     ,4H     ,      1016,       3.30,      11.53,                    DATA1
     S-1.68917484E+0,    2.01929000E-2, -1.35394127E-5,    3.43535328E-9/             DATA1
       DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),                    DATA1
      ACP1(L),CP2(L),CP3(L),CP4(L),L=28,33)/                                           DATA1
      B4HZC   ,4H     ,4H     ,      1119,       5.51,      -7.69,                    DATA1
     C 7.14972161E-1,    7.07999179E-3, -2.19924970E-6,   -1.21381310E-10,           DATA1
      D4HZD   ,4H     ,4H     ,      1993,       5.68,      -7.80,                    DATA1
     E 2.03920577E+0,    5.75753208E-3, -2.13712848E-6,   -3.50617941E-11,           DATA1
      F4HZT   ,4H     ,4H     ,      1994,       5.68,      -7.80,                    DATA1
     G-2.03920577E+0,    5.75753208E-3, -2.13712848E-6,   -3.50617941E-11,           DATA1
      H4HZZ   ,4H     ,4H     ,      1996,       4.96,      -4.64,                    DATA1
     I-7.11108376E-1,    1.79264165E-2, -1.59957077E-5,    4.70062916E-9,           DATA1
      J4HA    ,4H     ,4H     ,       997,      34.20,       6.00,                    DATA1
     K 1.88644431E+0,    8.73926593E-3, -7.16630431E-6,    2.02379753E-9,           DATA1
      L4HZ1   ,4H     ,4H     ,       999,       4.84,      -4.6325,                  DATA1
     M-4.45622480E-2,    1.28983697E-2, -1.02396205E-5,    2.93610786E-9/           DATA1
C2.0       DATA FOR OXYGEN-CONTAINING GROUPS.                                         DATA1
       DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),                   DATA1
      ACP1(L),CP2(L),CP3(L),CP4(L),L=51,57)/                                           DATA1
      B4H(CO),4H(CO),4HC    ,      1502,      -29.2 ,       0.0 ,                    DATA1
     C 4*0.0,                                                                        DATA1
      D4H(CO),4HOD    ,4H     ,      1055,      -33.5 ,       0.0 ,                    DATA1
     E 4*0.0,                                                                        DATA1
      F4H(CO),4HOZ    ,4H     ,      1058,      -46.0 ,       0.0 ,                    DATA1
     G 4*0.0,                                                                        DATA1
      H4H(CO),4HOC    ,4H     ,       981,      -33.4 ,      14.78,                    DATA1
     I 3.19242434E+00, 1.02961486E-02, -3.35001576E-06, -7.82103825E-10,             DATA1
      J4H(CO),4HOH    ,4H     ,       878,      -29.5 ,      34.93,                    DATA1
     K 4.63300963E+00, 6.58441850E-03,  6.03395343E-06, -5.05496292E-09,             DATA1
      L4H(CO),4HOH    ,4H     ,      1717,      -31.7 ,       0.0 ,                    DATA1
     M 4*0.0,                                                                        DATA1
      N4H(CO),4HZZ    ,4H     ,      2700,      -39.1 ,       0.0 ,                    DATA1
     O 4*0.0/                                                                        DATA1
       DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),                   DATA1
      ACP1(L),CP2(L),CP3(L),CP4(L),L=58,64)/                                           DATA1
      B4H(CO),4HZC    ,4H     ,      1823,      -37.6 ,       0.0 ,                    DATA1
     C 4*0.0,                                                                        DATA1
      D4H(CO),4HZH    ,4H     ,      1720,      -31.7 ,       0.0 ,                    DATA1
     E 4*0.0,                                                                        DATA1
      F4H(CO),4HCC    ,4H     ,       946,      -31.5 ,      15.01,                    DATA1
     G 3.30891440E+00, 7.16369917E-03,  2.33322493E-06, -5.19746180E-09,             DATA1
      H4H(CO),4HCH    ,4H     ,       843,      -29.6 ,      34.93,                    DATA1
     I 4.63300963E+00, 6.58441850E-03,  6.03395343E-06, -5.05496292E-09,             DATA1
      J4H(CO),4HHH    ,4H     ,       740,      -27.7 ,      53.67,                    DATA1
     K 6.26038702E+00, 4.58150196E-03,  1.12433979E-05, -7.28097943E-09,             DATA1
      L4HO(CO,4H)(CO,4H)    ,      1324,      -50.9 ,       0.0 ,                    DATA1
     M 4*0.0,                                                                        DATA1
      N4HO(CO,4H)O    ,4H     ,      1003,      -19.0 ,       0.0 ,                    DATA1
     O 4*0.0/                                                                        DATA1
       DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),                   DATA1
      ACP1(L),CP2(L),CP3(L),CP4(L),L=65,71)/                                           DATA1
      B4HO(CO,4H)O    ,4H     ,      1842,      -91.4 ,       0.0 ,                    DATA1
     C 4*0.0,                                                                        DATA1
      D4HO(CO,4H)C    ,4H     ,       968,      -41.3 ,       8.39,                    DATA1
     E 4*0.0,                                                                        DATA1
      F4HO(CO,4H)H    ,4H     ,       865,      -60.3 ,      24.52,                    DATA1
     G-1.86524457E+00, 2.61105691E-02, -2.66193574E-05,    1.01501091E-08,           DATA1
      H4HOOC  ,4H     ,4H     ,       447,       -4.5 ,       9.4 ,                   DATA1
     I 3.99375985E+00,-1.96318580E-03,  3.45093879E-06, -1.19120711E-09,             DATA1
      J4HOOO  ,4H     ,4H     ,       482,       19.0 ,       9.4 ,                   DATA1
     K 3.99375985E+00,-1.96318580E-03,  3.45093879E-06, -1.19120711E-09,             DATA1
      L4HOOH  ,4H     ,4H     ,       344,      -16.27,      27.85,                    DATA1
     M 2.49720532E+00, 1.13423353E-02, -9.11292325E-06,    2.67551005E-09,           DATA1
      N4HOOD  ,4H     ,4H     ,      2160,      -32.8 ,       8.1 ,                    DATA1
     O 4*0.0/                                                                        DATA1
       DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),                   DATA1
      ACP1(L),CP2(L),CP3(L),CP4(L),L=72,78)/                                           DATA1
      B4HOOC  ,4H     ,4H     ,      1286,      -31.3 ,       0.0 ,                    DATA1
     C 4*0.0,                                                                        DATA1
      D4HOZZ  ,4H     ,4H     ,      2166,      -19.3 ,       0.0 ,                    DATA1
     E 4*0.0,                                                                        DATA1
      F4HO'C  ,4H     ,4H     ,      1289,      -22.6 ,       0.0 ,                    DATA1
     G 4*0.0,                                                                        DATA1
      H4HOZH  ,4H     ,4H     ,      1186,      -32.9 ,      29.1 ,                    DATA1
     I 4.00163263E+00,-5.24726011E-03,  1.40015611E-05,   -6.95655002E-09,           DATA1
```

19 FEB 73 8.02-35

```
     J4HOCC ,4H     ,4H     ,      412,      -23.7 ,       8.48,                      DATA1
     K 3.73567870E-00,-2.79311793E-03,  7.37218601E-06, -3.49635604E-09,   DATA1
     L4HOCH ,4H     ,4H     ,      309,      -37.88,       29.67,                     DATA1
     M 5.05817029E+00,-6.71425025E-03,  1.65070951E-05, -8.32505317E-09,   DATA1
     N4HO'CO,4H)O   ,4H     ,     1028,        6.3 ,        0.0 ,                     DATA1
     O 4*0.0/                                                                         DATA1
      DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),                    DATA1
     ACP1(L),CP2(L),CP3(L),CP4(L),L=79,85)/                                           DATA1
     B4HO(CO,4H)C   ,4H     ,     1793,        9.4 ,        0.0 ,                     DATA1
     C 4*0.0,                                                                         DATA1
     D4HO(CO,4H)H   ,4H     ,     1690,        7.60,        0.0 ,                     DATA1
     E 4*0.0,                                                                         DATA1
     F4HOOO ,4H     ,4H     ,     2146,        8.9 ,        0.0 ,                     DATA1
     G 4*0.0,                                                                         DATA1
     H4HOOC ,4H     ,4H     ,     1272,       10.3 ,        0.0 ,                     DATA1
     I 4*0.0,                                                                         DATA1
     J4HOOH ,4H     ,4H     ,     1169,        8.6 ,        8.0 ,                     DATA1
     K 8.45867337E-01, 1.28226360E-02, -6.32082053E-06,  1.11437915E-09,   DATA1
     L4H2(CO,4H)    ,4H     ,     1675,        9.7 ,        0.0 ,                     DATA1
     M 4*0.0,                                                                         DATA1
     N4H2O  ,4H     ,4H     ,     1154,       -1.8 ,      -10.2 ,                     DATA1
     O-4.44977234E+00, 4.06574195E-02, -4.83862365E-05,  1.90830840E-08/   DATA1
      DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),                    DATA1
     ACP1(L),CP2(L),CP3(L),CP4(L),L=86,92)/                                           DATA1
     B4HC(CO,4H)(CO,4H)HH ,      1911,       -7.2 ,        0.0 ,                     DATA1
     C 4*0.0,                                                                         DATA1
     D4HC(CO,4H)CCH,4H    ,      1058,       -1.8 ,      -12.0 ,                     DATA1
     E 4*0.0,                                                                         DATA1
     F4HC(CO,4H)CHH,4H    ,       955,       -5.0 ,        9.6 ,                     DATA1
     G 3.39620138E-01, 2.51711439E-02, -2.06641082E-05,  7.34898747E-09,   DATA1
     H4HC(CO,4H)CCC,4H    ,      1161,        1.6 ,        0.0 ,                     DATA1
     I 4*0.0,                                                                         DATA1
     J4HC(CO,4H)HHH,4H    ,       852,      -10.08,       30.41,                     DATA1
     K-1.67923834E-01, 2.44328426E-02, -1.16714356E-05,  2.10055135E-09,   DATA1
     L4HCOOC,4HC    ,4H     ,      675,      -16.8 ,        0.0 ,                     DATA1
     M 4*0.0,                                                                         DATA1
     N4HCOOC,4HH    ,4H     ,      572,      -17.2 ,        0.0 ,                     DATA1
     O 4*0.0/                                                                         DATA1
      DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),                    DATA1
     ACP1(L),CP2(L),CP3(L),CP4(L),L=93,99)/                                           DATA1
     B4HCOOH,4HH    ,4H     ,      429,      -17.7 ,        0.0 ,                     DATA1
     C 4*0.0,                                                                         DATA1
     D4HCO2H,4HH    ,4H     ,     1311,       -6.6 ,        9.7 ,                     DATA1
     E 4*0.0,                                                                         DATA1
     F4HCOOH,4HH    ,4H     ,     1308,       -3.9 ,        0.0 ,                     DATA1
     G 4*0.0,                                                                         DATA1
     H4HCOCC,4HC    ,4H     ,      646,       -6.50,      -33.56,                     DATA1
     I-6.21487381E+00, 5.16292395E-02, -6.39197997E-05,  2.47600205E-08,   DATA1
     J4HCOCC,4HH    ,4H     ,      537,       -7.00,      -11.90,                     DATA1
     K-5.35977820E+00, 4.78986890E-02, -5.25420190E-05,  2.04415825E-08,   DATA1
     L4HCOCH,4HH    ,4H     ,      434,       -8.5 ,       10.63,                     DATA1
     M-3.46110576E+00, 3.69163436E-02, -3.24811370E-05,  1.13583539E-08,   DATA1
     N4HCOHH,4HH    ,4H     ,      331,      -10.08,       30.41,                     DATA1
     O-1.75871384E-01, 2.44607515E-02, -1.16903894E-05,  2.10285358E-09/   DATA1
C3.0      DATA FOR NITROGEN-CONTAINING GROUPS.                                         DATA1
      DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),                    DATA1
     ACP1(L),CP2(L),CP3(L),CP4(L),L=118,124)/                                         DATA1
     B4HCNHH,4HH    ,4H     ,      718,      -10.08,       30.41,                     DATA1
     C-1.67923834E-01, 2.44328424E-02, -1.16714356E-05,  2.10035135E-09,   DATA1
     D4HCNCH,4HH    ,4H     ,      815,       -6.6 ,        9.8 ,                     DATA1
     E-1.09196751E+00, 3.01279382E-02, -2.33046702E-05,  7.40890103E-09,   DATA1
     F4HCNCC,4HH    ,4H     ,      918,       -5.2 ,      -11.7 ,                     DATA1
     G-3.73791206E+00, 3.82564597E-02, -3.82709119E-05,  1.39873957E-08,   DATA1
     H4HCNCC,4HC    ,4H     ,     1021,       -3.2 ,      -34.1 ,                     DATA1
     I-5.61176857E+00, 4.75192106E-02, -5.35080656E-05,  2.01088418E-08,   DATA1
     J4HNCHH,4H     ,4H     ,      699,        4.8 ,       29.71,                     DATA1
     K 2.89594738E+00, 1.60598791E-02, -2.32720336E-06, -1.39650146E-10,   DATA1
     L4HNCCH,4H     ,4H     ,      802,       15.4 ,        8.94,                     DATA1
     M 9.59545130E-03, 1.70255903E-02, -1.09771703E-05,  2.57828880E-09,   DATA1
     N4HNCCC,4H     ,4H     ,      935,       24.4 ,      -13.46,                     DATA1
     O-1.81102942E+00, 2.33777734E-02, -2.09418857E-05,  6.03393047E-09/   DATA1
      DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),                    DATA1
     ACP1(L),CP2(L),CP3(L),CP4(L),L=125,131)/                                         DATA1
     B4HNNHH,4H     ,4H     ,     1119,       11.4 ,       29.13,                     DATA1
     C 9.75906822E-01, 2.11752446E-02, -1.46099632E-05,  3.94852974E-09,   DATA1
```

9 FEB 73 8.02-39

```
226.        D*HNNCH,*H    ,*H     ,     1218,       20.9 ,      9.61,                    DATA1
227.        E 9.56846121E-01, 1.64803897E-02, -1.26831746E-05,  3.51125561E-09,          DATA1
228.        F*HNNCC,*H    ,*H     ,     1321,       29.2 ,     13.80,                    DATA1
229.        G *=0.0,                                                                     DATA1
230.        H*HNNZH,*H    ,*H     ,     2095,       22.1 ,      0.0 ,                    DATA1
231.        I *=0.0,                                                                     DATA1
232.        J*HN1H ,*H    ,*H     ,      479,        0.0 ,      0.0 ,                    DATA1
233.        K *=0.0,                                                                     DATA1
234.        L*HN1C ,*H    ,*H     ,      563,       21.3 ,      0.7 ,                    DATA1
235.        M *=0.0,                                                                     DATA1
236.        N*HN1Z ,*H    ,*H     ,     1459,       16.7 ,      0.0 ,                    DATA1
237.        O *=0.0/                                                                     DATA1
238.          DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),             DATA1
239.        ACP1(L),CP2(L),CP3(L),CP4(L),L=132,138)/                                     DATA1
240.        B*HNZH ,*H    ,*H     ,      527,       25.1 ,     26.0 ,                    DATA1
241.        C 2.41191140E+00, 7.15235247E-03, -2.21801632E-06,  8.55800298E-11,          DATA1
242.        D*HNZC ,*H    ,*H     ,      630,       32.5 ,      8.0 ,                    DATA1
243.        E *=0.0,                                                                     DATA1
244.        F*HNZHH,*H    ,*H     ,     1576,        4.8 ,     29.71,                    DATA1
245.        G 2.8*5947*8E+00, 1.00598791E-02, -2.32720336E-06, -1.39650146E-10,          DATA1
246.        H*HNZCH,*H    ,*H     ,     1679,       14.9 ,      0.0 ,                    DATA1
247.        I *=0.0,                                                                     DATA1
248.        J*HNZCC,*H    ,*H     ,     1782,       26.2 ,      0.0 ,                    DATA1
249.        K *=0.0,                                                                     DATA1
250.        L*HNZZH,*H    ,*H     ,     2556,       10.3 ,      0.0 ,                    DATA1
251.        M *=0.0,                                                                     DATA1
252.        N*HZN  ,*H    ,*H     ,     1538,       -0.5 ,     -9.69,                    DATA1
253.        C-3.57157539E+00, 3.69462129E-02, -4.48672434E-05,  1.80561699E-08/          DATA1
254.          DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),             DATA1
255.        ACP1(L),CP2(L),CP3(L),CP4(L),L=139,145)/                                     DATA1
256.        B*HN2H ,*H    ,*H     ,     1046,       23.0 ,      0.0 ,                    DATA1
257.        C *=0.C,                                                                     DATA1
258.        D*N(CO),*HNH  ,*H     ,     1259,      -29.6 ,     34.93,                    DATA1
259.        E 4.63300763E+00, 6.59441850E-03,  6.03395343E-06, -5.05496292E-09,          DATA1
260.        F*H(CO),*HNC  ,*H     ,     1362,      -32.8 ,     16.2 ,                    DATA1
261.        G 3.6*767855E+00, 3.89405707E-03,  7.00136594E-06, -3.37854793E-09,          DATA1
262.        H*HN(CO,*H)HH ,*H     ,     1255,      -14.9 ,     24.69,                    DATA1
263.        I-3.25292052E+00, 3.09737873E-02, -2.42737089E-05,  7.77065898E-09,          DATA1
264.        J*HN(CO,*H)CH ,*H     ,     1358,       -4.4 ,      3.9 ,                    DATA1
265.        K *.9.0,                                                                     DATA1
266.        L*HN(CO,*H)CC ,*H     ,     1461,        0.0 ,      0.0 ,                    DATA1
267.        M *=0.0,                                                                     DATA1
268.        N*HN(CO,*H)ZH ,*H     ,     2233,        0.4 ,      0.0 ,                    DATA1
269.        O *=0.0/                                                                     DATA1
270.          DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),             DATA1
271.        ACP1(L),CP2(L),CP3(L),CP4(L),L=146,152)/                                     DATA1
272.        B*HN(CO,*H)(CO,*H)H  ,     1914,      -18.5 ,      0.0 ,                    DATA1
273.        C *=0.0,                                                                     DATA1
274.        D*HN(CO,*H (CO,*H)C  ,     2017,       -5.9 ,      0.0 ,                    DATA1
275.        E *=0.C,                                                                     DATA1
276.        F*HN(CO,*H)(CO,*H)Z  ,     2894,       -0.5 ,      0.0 ,                    DATA1
277.        G *=0.0,                                                                     DATA1
278.        H*HC(CN,*H)CHH,*H     ,      679,       22.9 ,     40.20,                    DATA1
279         I 1.59367113E+00, 3.80737392E-02, -2.26624252E-05,  4.49026495E-09,          DATA1
280.        J*HC(CN,*H)CCH,*H     ,      792,       25.8 ,     19.80,                    DATA1
281.        K *.45475478E+00, 2.59552780E-02, -1.45011110E-05,  2.77018921E-09,          DATA1
282.        L*HC(CN,*H)CCC,*H     ,      895,        0.0 ,     -2.80,                    DATA1
283.        M *=0.0,                                                                     DATA1
284.        N*HC(CN,*H)(CN,*H)CC  ,     1165,        0.0 ,     28.40,                    DATA1
285.        O *=0.9/                                                                     DATA1
286.          DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),S298(L),             DATA1
287.        ACP1(L),CP2(L),CP3(L),CP4(L),L=153,159)/                                     DATA1
288         B*HD(CN,*H)H  ,*H     ,     1414,       37.4 ,     36.58,                    DATA1
289         C 1.62789124E+00, 3.41575604E-02, -2.50052036E-05,  6.52183340E-09,          DATA1
290.        D*HD(CN,*H)(CN,*H)    ,     1797,       84.1 ,      0.0 ,                    DATA1
291.        E *=0.0,                                                                     DATA1
292.        F*HD(CN,*HO)H ,*H     ,     1247,        0.0 ,     44.4 ,                    DATA1
293.        G 3.05906329E-01, 9.04780384E-02, -3.70313510E-05,  1.01920047E-08,          DATA1
294.        H*HZ(CN,*H)   ,*H     ,     1501,       35.8 ,     20.50,                    DATA1
295.        I 3.13119650E+00, 2.99963575E-02, -2.70020814E-05   9.27540003E-09,          DATA1
296.        J*HT(CN,*H)   ,*H     ,     1397,       63.8 ,     35.40,                    DATA1
297.        K 6.32003189E+00  1.66234963E-02, -1.18636033E-05,  3.10238362E-09,          DATA1
298.        L*HC(ON,*HO)CH,*HH    ,      912,      -19.1 ,     48.4 ,                    DATA1
299.        M *=0.0,                                                                     DATA1
300.        N*HC(ON,*HO)CC,*HH    ,      619,      -19.8 ,     26.9 ,                    DATA1
```

9 FEB 73 3 02-38

```
      D 4*0.0/                                                          DATA1
       DATA (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),KF298(L),S298(L),     DATA1
      ACP1(L),CP2(L),CP3(L),CP4(L),L=160,163)/                          DATA1
      B4HC(ON,4HO)CC,4HC    ,      710,        0.0 ,     3.9 ,          DATA1
      C 4*0.0,                                                          DATA1
      D4HC(ON,4HO)(O,5HNO)CH,      720,      -14.9 ,     0.0 ,          DATA1
      E 4*0.0,                                                          DATA1
      F4HO(NO,4H)C  ,4H     ,      500,       -5.9 ,    41.9 ,          DATA1
      G 4.39094421E+00, 1.91590634E-02, -1.22787841E-05,  2.64523631E-09, DATA1
      H4HO(ON,4HO)C ,4H     ,      529,      -15.9 ,    48.50,          DATA1
      I 4*0.0/                                                          DATA1
       END                                                              DATA1
```

9 FEB 73 6.02-39

```
      SUBROUTINE DELETE(KIRD,ISBE,KTT)                                  DELETE
C        THIS SUBROUTINE DISENGAGES FROM USE ALL THE PAIRS OF CHAIN      DELETE
C        RING ATOMS AND THEIR NON-RING LIGANDS WHICH HAVE BEEN FOUND     DELETE
C        TO BE DISSIMILAR BY THE COMPARISON TESTS OF SUBROUTINE EQUALR.  DELETE
      INTEGER SYMX(4),SYMBOL(9),GRID(50,80)                             DELETE
      DIMENSION JDONE(100),KCCR(2,30),IDX(8,99)                         DELETE
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                      DELETE
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),  DELETE
     1IDBK(100),IB(100,8),IRG,NOBR                                      DELETE
      EQUIVALENCE (JDONE(1),IMATX(2,47)),(KCCR(1,1),GRID(1,23)),        DELETE
     1((IDX(1,1),GRID(3,1)))                                            DELETE
C1.0     THIS SECTION DISENGAGES FROM USE ALL THOSE PAIRS OF CHAIN      DELETE
C        ATOMS FROM THE TWO RINGS, A AND B, UNDER COMPARISON WHICH ARE  DELETE
C        EACH BONDED TO TWO OTHER RING ATOMS AND WHICH FOLLOW THE       DELETE
C        PREVIOUS BRANCH RING ATOM. THE LATTER IS AN ATOM BONDED TO     DELETE
C        THREE OTHER RING ATOMS, NAMELY, ATOMS THAT FORM PART OF THE    DELETE
C        RING BACKBONE.                                                 DELETE
      KK=KTT                                                            DELETE
      DO3   K=1,KTT                                                     DELETE
      IF(KCCR(1,KK).EQ.KIRD)GO TO 5                                     DELETE
      K1=KCCR(1,KK)                                                     DELETE
      JDONE(K1)=0                                                       DELETE
      K2=KCCR(2,KK)                                                     DELETE
      JDONE(K2)=0                                                       DELETE
      KK=KK-1                                                           DELETE
    3 CONTINUE                                                          DELETE
C2.0     IF NONE OF THE AFOREMENTIONED CHAIN RING ATOMS ARE PRESENT,    DELETE
C        EXIT FROM ROUTINE.                                             DELETE
    5 KD=KTT-KK                                                         DELETE
      IF(KD.EQ.0)RETURN                                                 DELETE
C3.0     THIS SECTION ALSO DISENGAGES FROM USE ANY PAIRS OF NON-RING    DELETE
C        ATOMS BONDED TO THE AFOREMENTIONED CHAIN RING ATOMS.           DELETE
      K1=KTT                                                            DELETE
      KTT=KK                                                            DELETE
      K2=ISBE                                                           DELETE
      DO11  K=1,KD                                                      DELETE
      IF(KCCR(1,K1).NE.IDX(1,K2))GO TO 9                                DELETE
      K2=K2-1                                                           DELETE
    9 K1=K1-1                                                           DELETE
   11 CONTINUE                                                          DELETE
      ISBE=K2                                                           DELETE
      RETURN                                                            DELETE
      END                                                               DELETE
```

```
 1.          SUBROUTINE DELTA1(KT)                                          DELTA1
 2.  C         THIS SUBROUTINE DETERMINES THAT PART OF THE WEIGHT OF GROUP KT DELTA1
 3.  C         THAT IS COMPOSED OF THE SUM OF THE ASSIGNED WEIGHTS OF THE    DELTA1
 4.  C         GROUP CORE ATOM AND OF THE ATOMIC CONSTITUENTS BONDED TO THE  DELTA1
 5.  C         CORE ATOM.                                                    DELTA1
 6.          INTEGER WEIGHT(9)                                              DELTA1
 7.          COMMON/BLK2/WEIGHT,RWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC  DELTA1
 8.  C1.       SET KSUM.                                                    DELTA1
 9.          KSUM=0                                                         DELTA1
10.          I=IX(KT,1,1)                                                   DELTA1
11.  C2.       ADD WEIGHT OF CORE ATOM KT TO MOLWT(KT).                     DELTA1
12.          MOLWT(KT)=MOLWT(KT)+WEIGHT(I)                                  UELTA1
13.  C3.       ADD WEIGHTS OF LIGAND ATOMS BONDED TO KT.                    DELTA1
14.          DO11   KM=2,5                                                  DELTA1
15.          I=IX(KT,KM,1)                                                  DELTA1
16.          IF(I.EQ.0)GO TO 12                                             DELTA1
17.  C             ADD WEIGHT OF LIGAND KM.                                DELTA1
18.          MOLWT(KT)=MOLWT(KT)+WEIGHT(I)                                  DELTA1
19.  C           DETERMINE WHETHER WEIGHT OF ATOM KM HAS TO BE MODIFIED.     DELTA1
20.  C             IS BOND BETWEEN KT AND KM MULTIPLE                       DELTA1
21.          IF(IX(KT,KM,5).EQ.1)GO TO 11                                   DELTA1
22.  C                 YES, IT IS.                                          DELTA1
23.          IF(IX(KT,1,1).NE.2)GO TO 3                                     DELTA1
24.  C               ATOM KT IS CARBON ATOM.                                DELTA1
25.          IF(IX(KM,1).NE.2)GO TO 9                                       DELTA1
26.  C               ATOM KM IS ALSO CARBON ATOM.                           DELTA1
27.  C                 IS BOND BETWEEN KT AND KM TRIPLE                     DELTA1
28.          IF(IX(KT,KM,5).EQ.3)GO TO 1                                    DELTA1
29.  C                   NO, IT IS DOUBLE. APPLY CORRECTION FOR C=C.         DELTA1
30.  C                    !NCREMENT KSUM.                                   DELTA1
31.          MOLWT(KT)=MOLWT(KT)+ 788                                       DELTA1
32.          KSUM=KSUM+1                                                    DELTA1
33.          GO TO 11                                                       DELTA1
34.  C                     YES, IT IS. APPLY CORRECTION FOR C C             DELTA1
35.        1 MOLWT(KT)=MOLWT(KT)+ 764                                       DELTA1
36.          GO TO 11                                                       DELTA1
37.        3 IF(IX(KT,1,1).NE.4.OR.IX(KT,KM,5).NE.2)GO TO 11                DELTA1
38.  C               KT IS OF TYPE N=. NOW DETERMINE KM.                    DELTA1
39.          IF(IX(KT,KM,1)-3)5,11,7                                        DELTA1
40.  C                 APPLY CORRECTION FOR N=C OR C=N                      DELTA1
41.        5 MOLWT(KT)=MOLWT(KT)-202                                        DELTA1
42.          GO TO 11                                                       DELTA1
43.  C                   APPLY CORRECTION FOR N=N                           DELTA1
44.        7 MOLWT(KT)=MOLWT(KT)-570                                        DELTA1
45.          GO TO 11                                                       DELTA1
46.  C               IF KM IS OF TYPE N=, APPLY CORRECTION FOR C=N (GO TO 5). DELTA1
47.        9 IF(IX(KT,KM,1).EQ.4.AND.IX(KT,KM,5).EQ.2)GO TO 5               DELTA1
48.       11 CONTINUE                                                       DELTA1
49.  C4.       TWO CORRECTIONS HAVE BEEN APPLIED FOR C=C. BOND IS ACTUALLY  DELTA1
50.  C         C=C=C. APPLY WEIGHT CORRECTION FOR ALLENE.                   DELTA1
51.       13 IF(KSUM.NE.2)RETURN                                           DELTA1
52.          MOLWT(KT)=MOLWT(KT) - 672                                      DELTA1
53.          RETURN                                                         DELTA1
54.          END                                                            DELTA1
```

DELTA2

```
      SUBROUTINE DELTA2(KT,JT)                                          DELTA2
C        THIS SUBROUTINE ADDS ALL OF THE SECOND-ORDER MULTIPLE-BOND     DELTA2
C        CONTRIBUTIONS TO THE GROUP (KT) WEIGHT VALUE. THE CONTRIBUTIONS DELTA2
C        CONSIDERED HERE ARE THOSE ARISING FROM ANY MULTIPLE BONDS WHICH DELTA2
C        LIGANDS BONDED TO THE CORE ATOM (JT) MAY HAVE. THE BOND        DELTA2
C        BETWEEN KT AND JT IS, OF COURSE, NOT INCLUDED.                 DELTA2
      INTEGER WEIGHT(9)                                                 DELTA2
      COMMON/BLK2/WEIGHT,AWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC     DELTA2
C1.      IS ATOM JT A CARBON ATOM                                       DELTA2
      IF(IX(JT,1,1).NE.2)RETURN                                         DELTA2
C           YES, IT IS.                                                 DELTA2
C2.0     FIND LOCATION OF DATA IN GROUP JT ASSOCIATED WITH LIGAND KT.   DELTA2
      DO5   KM=2,5                                                      DELTA2
      IF(IX(KT,1,2).EQ.IX(JT,KM,2).AND.IX(KT,1,3).EQ.IX(JT,KM,3))GO TO 7 DELTA2  7
    5 CONTINUE                                                          DELTA2
    7 KV=KM                                                             DELTA2
C3.0     INITIATE CYCLE THAT ASSIGNS THE SECOND-ORDER CORRECTIONS.      DELTA2
      DO13   KM=2,5                                                     DELTA2
      IF(KM.EQ.KV)GO TO 13                                              DELTA2  13
C           BOND BETWEEN KT AND JT IS EXCLUDED.                         DELTA2
      IF(IX(JT,KM,1).NE.2.OR.IX(JT,KM,5).EQ.1)GO TO 13                  DELTA2  13
C        ATOM KM IS ALSO A CARBON ATOM AND BOND BETWEEN JT AND KM       DELTA2
C        IS MULTIPLE.                                                   DELTA2
C           IS BOND BETWEEN JT AND KM TRIPLE                            DELTA2
      IF(IX(JT,KM,5).EQ.3)GO TO 3                                       DELTA2  3
C              NO, IT IS DOUBLE. IS IT AN ALLENE TYPE BOND              DELTA2
      IF(IX(KT,1,1).EQ.2.AND.IX(JT,KV,5).EQ.2)GO TO 1                   DELTA2  1
C                 NO, APPLY CORRECTION FOR C=C                          DELTA2
      MOLWT(KT)=MOLWT(KT)+ 874                                          DELTA2
      RETURN                                                            DELTA2
C                    YES, APPLY CORRECTION FOR C=C=C                    DELTA2
    1 MOLWT(KT)=MOLWT(KT)+0                                             DELTA2
      RETURN                                                            DELTA2
C                 YES, IT IS TRIPLE. APPLY CORRECTION FOR C C           DELTA2
    3 MOLWT(KT)=MOLWT(KT)+ 875                                          DELTA2
      RETURN                                                            DELTA2
   13 CONTINUE                                                          DELTA2
      RETURN                                                            DELTA2
      END                                                               DELTA2
```

17 APR 73 8.02-36

```
      SUBROUTINE DITERE(IGS,IGO,IDTE,DTEM)                              DITERE
C        THIS SUBROUTINE DETERMINES WHETHER THERE ARE ONE OR MORE       DITERE
C        DITERTIARY ETHER GROUPS PRESENT. IF SO, IT STORES THE          DITERE
C        MAGNITUDE OF EACH CORRECTION AND SETS THE OTHER PERTINENT      DITERE
C        VARIABLES.                                                     DITERE
      INTEGER SUM1,SUM2                                                 DITERE
      INTEGER WEIGHT(9)                                                 DITERE
      DIMENSION KCGAUS(3,150),KFGAUS(150),KTGAUS(150),DATCIS(2,150),    DITERE
     1IM(150)                                                           DITERE
      COMMON/BLK2/WEIGHT,AWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC     DITERE
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),  DITERE
     1IDBR(100),ID(100,8),IRO,NOBR                                      DITERE
      EQUIVALENCE (KCGAUS(1,1),IMATX(1,72)),(KFGAUS(1),IMATX(1,63)),    DITERE
     1(KTGAUS(1),IMATX(1,60)),(DATCIS(1,1),IMATX(1,66)),(IM(1),         DITERE
     2IMATX(1,57))                                                      DITERE
C1.0     IF NUMBER OF GAUCHE ETHER CORRECTIONS ARE EVEN, CONTINUE.      DITERE
C        OTHERWISE PRINT ERROR MESSAGE AND EXIT.                        DITERE
      JJ=IGO/2                                                          DITERE
      IF(IGO.EQ.2*JJ)GO TO 63                                           DITERE  63
      WRITE(6,61)                                                       DITERE
   61 FORMAT(///1H0,44HERROR IN GAUCHE ETHER CALCULATION. CONTINUE.///) DITERE
      RETURN                                                            DITERE
   63 LI=IGS-IGO+1                                                      DITERE
      LF=IGS                                                            DITERE
C2.0     TEST EACH PAIR OF GAUCHE ETHERS FOR DITERTIARY ETHER STRUCTURE. DITERE
      DO 65   L=LI,LF,2                                                 DITERE
C           FIND GROUP NUMBERS OF NON-OXYGEN CORE ATOMS.                DITERE
      IBX=IM(L)                                                         DITERE
      KC1=KCGAUS(1,IBX)                                                 DITERE
      KC2=KCGAUS(1,IBX+1)                                               DITERE
      SUM1=IX(KC1,1,1)+IX(KC2,1,1)                                      DITERE
      SUM2=NW(KC1)+NW(KC2)                                              DITERE
      IF(SUM1.NE.4.OR.SUM2.NE.8)GO TO 65                                DITERE  65
C           BOUNDARY LIGANDS ARE CARBON ATOMS AND EACH HAS A            DITERE
C           CONNECTIVITY OF FOUR. DITERTIARY GROUP IS PRESENT. SET      DITERE
C           VARIABLES AND CONTINUE.                                     DITERE
      IGS=IGS+1                                                         DITERE
      KCGAUS(1,IGS)=KC1                                                 DITERE
      KCGAUS(2,IGS)=KCGAUS(2,IBX)                                       DITERE
      KCGAUS(3,IGS)=KC2                                                 DITERE
      KTGAUS(IGS)=0                                                     DITERE
      KFGAUS(IGS)=1                                                     DITERE
      DATCIS(1,IGS)=0.9                                                 DITERE
      DTEM=DTEM+DATCIS(1,IGS)                                           DITERE
      IDTE=IDTE+1                                                       DITERE
   65 CONTINUE                                                          DITERE
      RETURN                                                            DITERE
      END                                                               DITERE
```

17 APR 73 8.02-36

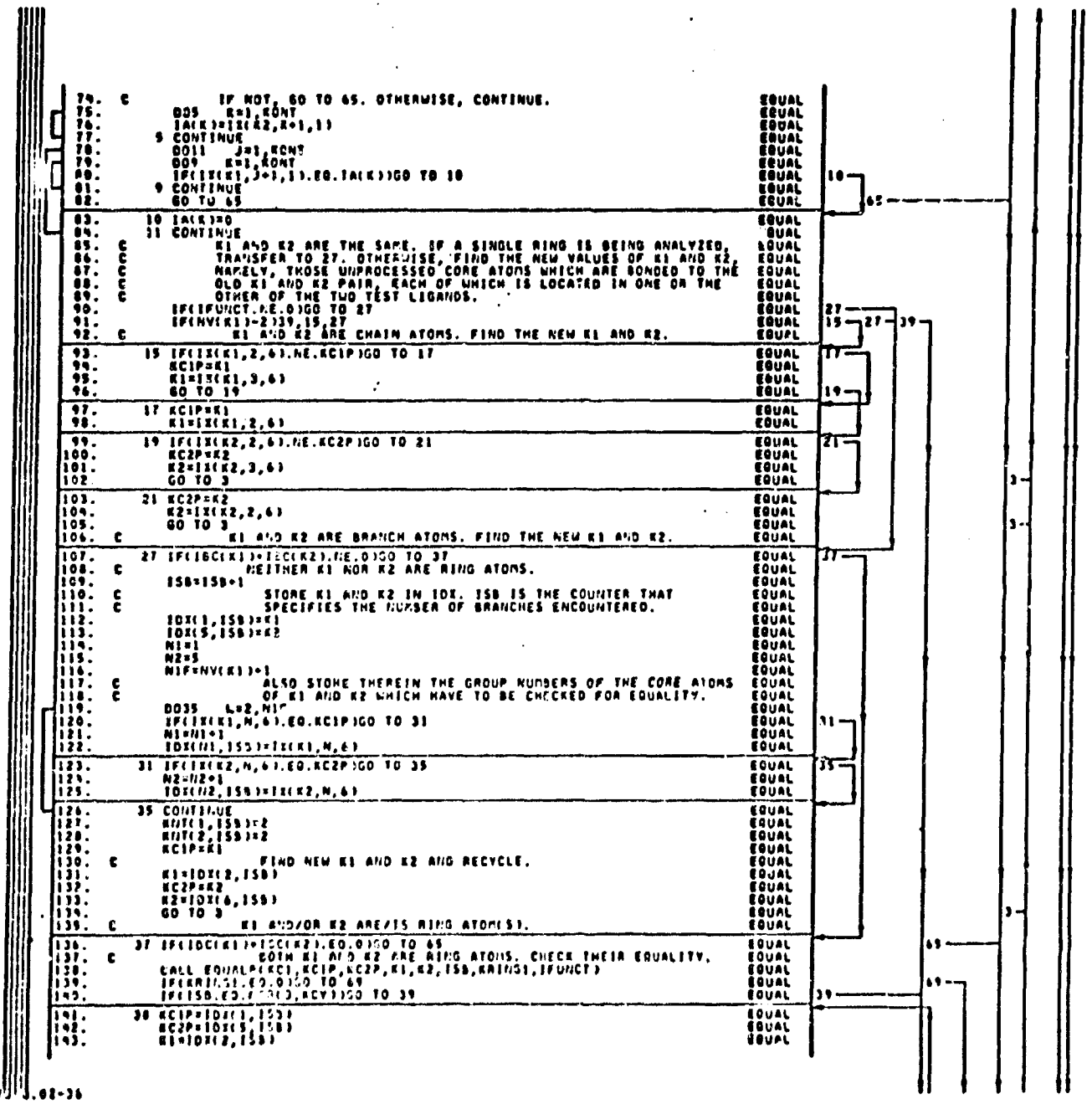
```
  1.          SUBROUTINE ENTSYM(NOSN,NENAN,MESO,NASYMC,NOSNC,NPSUDA,NCSUDA,      ENTSYM
  2.         1NOMESO,SSN,SOPTS)                                                  ENTSYM
  3.    C        THIS SUBROUTINE COMPUTES THE CONTRIBUTIONS TO THE ENTROPY DUE   ENTSYM
  4.    C        TO INTERNAL AND EXTERNAL ROTATIONAL SYMMETRY AND OPTICAL        ENTSYM
  5.    C        ISOMERISM USING COMPUTED DATA AS WELL AS INPUT DATA (IF ANY).   ENTSYM
  6.          DIMENSION TOUT(12),IOPATM(100)                                    ENTSYM
  7.          COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),  ENTSYM
  8.         1IDBR(100),I8(100,8),IRG,NOBR                                      ENTSYM
  9.          COMMON/BLK5/NDATM,NUMATM(5),MBC(50),MBS(2),JM,JV,LFLAGS,LFLAGG    ENTSYM
 10.          EQUIVALENCE (IOPATM(1),IMATX(2,45))                               ENTSYM
 11.          DATA TOUT/1H ,4H IMP,4HUT  ,1H ,1H ,4HCOMP,4HUTED,1H ,4H NOT,     ENTSYM
 12.         1 4H CAL,4HCULA,4HBLE /                                            ENTSYM
 13.    C1.0      INITIALIZE VARIABLE.                                          ENTSYM
 14.          LFLAGS=0                                                          ENTSYM
 15.    C2.0      WAS THE NUMBER OF ENANTIOMERS INPUT                           ENTSYM
 16.    C           YES. FIND TOTAL NUMBER, NOPTS, AND ENTROPY OF MIXING.       ENTSYM
 17.          IF(NENAN.EQ.0)GO TO 1                                             ENTSYM   1
 18.          NOPTS=NENAN*MESO                                                  ENTSYM
 19.          OPTS=NOPTS                                                        ENTSYM
 20.          SOPTS1=1.98726*ALOG(OPTS)                                         ENTSYM
 21.          SOPTS=SOPTS1                                                      ENTSYM
 22.          GO TO 3                                                           ENTSYM   3
 23.    C           NO. SET NOPTS=0.                                            ENTSYM
 24.        1 NOPTS=0                                                         ENTSYM
 25.        3 MESOC=0                                                         ENTSYM
 26.          AFREQ=1.000                                                       ENTSYM
 27.    C3.0      WERE ANY ASYMMETRIC ATOMS FOUND BY PROGRAM                    ENTSYM
 28.          IF(NASYMC.GT.0)GO TO 9                                            ENTSYM   9
 29.    C           NO. SET RELEVANT VARIABLES.                                 ENTSYM
 30.          NENANC=0                                                          ENTSYM
 31.          NOPTSC=0                                                          ENTSYM
 32.          SOPTSC=0.0                                                        ENTSYM
 33.          GO TO 19                                                          ENTSYM                19
 34.    C           YES. COMPUTE ENTROPY OF MIXING USING THE NUMBER OF ASYMMETRIC ENTSYM
 35.    C           ATOMS CALCULATED.                                           ENTSYM
 36.        9 IF(NOMESO.EQ.0)GO TO 13                                         ENTSYM   13
 37.          NX=NASYMC+NPSUDA                                                  ENTSYM
 38.          NENANC=2**(NX-1)                                                  ENTSYM
 39           IF(NX/2.EQ.2*NX)GO TO 11                                          ENTSYM   11
 40.          MESOC=2**((NX-1)/2)                                               ENTSYM
 41.          NENANC=NENANC-MESOC                                               ENTSYM
 42.          GO TO 15                                                          ENTSYM   15
 43.       11 MESOC=2**((NX-2)/2)                                               ENTSYM
 44.          GO TO 15                                                          ENTSYM   15
 45.       13 NENANC=2**NASYMC                                                  ENTSYM
 46.       15 NOPTSC=NENANC*MESOC                                               ENTSYM
 47.          IF(NOMESO.EQ.2)AFREQ=FLOAT(MESOC*2/NOPTSC)                        ENTSYM
 48.          OPTS=NOPTSC                                                       ENTSYM
 49.          SOPTSC=1.98726*ALOG(OPTS)                                         ENTSYM
 50.    C9.0      WAS AN EXTERNAL SYMMETRY NUMBER INPUT                         ENTSYM
 51.       19 IF(NOSN.EQ.0)GO TO 23                                             ENTSYM   23
 52.    C           YES. COMPUTE EXTERNAL ROTATIONAL ENTROPY CONTRIBUTION FROM   ENTSYM
 53.    C           INPUT DATA.                                                 ENTSYM
 54.          SN=NOSN                                                           ENTSYM
 55.          IF(MESO.LE.0)GO TO 21                                             ENTSYM   21
 56.          AFREQ=FLOAT(MESO*2/NOPTS)                                         ENTSYM
 57.          GO TO 22                                                          ENTSYM   22
 58.       21 IF(MESOC.LE.0)GO TO 22                                            ENTSYM   22
 59.          AFREQ=FLOAT(MESOC*2/NOPTSC)                                       ENTSYM
 60.       22 SSN1=-1.98726*ALOG(SN)                                            ENTSYM
 61.          SSN=SSN1                                                          ENTSYM
 62.    C           NOW COMPUTE SAME PROPERTY FROM COMPUTED DATA.                ENTSYM
 63.       23 SN=NOSNC                                                          ENTSYM
 64.          SSNC=-1.98726*AFREQ*ALOG(SN)                                      ENTSYM
 65.    C9.9      IF SYMMETRY AND/OR OPTICAL DATA WERE INPUT, THE ENTROPY       ENTSYM
 66.    C         CONTRIBUTIONS USED IN THE THERMO CALCULATIONS ARE THOSE DERIVED ENTSYM
 67.    C         FROM THE INPUT DATA.                                          ENTSYM
 68.          IF(NOSN.EQ.0)SSN=SSNC                                             ENTSYM
 69.          IF(NOPTS.EQ.0)SOPTS=SOPTSC                                        ENTSYM
 70.    C6.0      SET FLAG IF DATUM WAS NOT INPUT AND PROPERTY IS NOT COMPUTABLE. ENTSYM
```

```
      IF(JW.EQ.9.AND.NOSN.EQ.0.OR.JV.EQ.9.AND.NOPTS.EQ.0)LFLAGS=1          ENTSVA
C7.0      PRINT OUT EXTERNAL ROTATIONAL SYMMETRY CONTRIBUTION.             ENTSVA
      WRITE(6,25)                                                          ENTSVA
   25 FORMAT(//1H0,42X,45HEXTERNAL  ROTATIONAL  SYMMETRY  CONTRIBUTION     ENTSVA
     1//1H ,45X,6HSOURCE, 9X,15HSYMMETRY NUMBER,7X,7HENTROPY)              ENTSVA
      IF(NOSN.EQ.0)GO TO 31                                                ENTSVA
C         SYMMETRY DATA WAS INPUT. PRINT OUT CONTRIBUTION COMPUTED         ENTSVA
C         THEREWITH.                                                       ENTSVA
      WRITE(6,29)(TOUT(K),K=1,4),NOSN,SSNI                                 ENTSVA
   29 FORMAT(1H ,38X,4A4,7X,I5,7X,F14.5)                                   ENTSVA
C         PRINT OUT CONTRIBUTION FROM COMPUTED DATUM.                      ENTSVA
   31 WRITE(6,29)TOUT(JW),TOUT(JW+1),TOUT(JW+2),TOUT(JW+3),NOSNC,SSNC      ENTSVA
C8.0      PRINT OUT OPTICAL ISOMERS CONTRIBUTION.                          ENTSVA
      WRITE(6,35)                                                          ENTSVA
   35 FORMAT(//1H0,49X,30HOPTICAL  ISOMER  CONTRIBUTION   //1H ,11X,      ENTSVA
     16HSOURCE, 9X,10HASYMMETRIC,4X,16HPSEUDOASYMMETRIC,4X,11HENANTIOMER  ENTSVA
     25,4X,4HMESO,4X,21HTOTAL OPTICAL ISOMERS,7X,7HENTROPY)                ENTSVA
      IF(NOPTS.EQ.0)GO TO 41                                               ENTSVA
C         OPTICAL DATA WERE INPUT. PRINT OUT CONTRIBUTION COMPUTED         ENTSVA
C         THEREWITH.                                                       ENTSVA
      WRITE(6,39)(TOUT(K),K=1,4),NENAN,MESO,NOPTS,SOPTSI                   ENTSVA
   39 FORMAT(1H ,6X,4A4,39X,I5,7X,I5,11X,I5,10X,F14.5)                     ENTSVA
C         PRINT OUT CONTRIBUTION FROM COMPUTED DATA.                       ENTSVA
   41 WRITE(6,45)TOUT(JV),TOUT(JV+1),TOUT(JV+2),TOUT(JV+3),NASYMC,         ENTSVA
     1NPSUDA,NENANC,MESOC,NOPTSC,SOPTSC                                    ENTSVA
   45 FORMAT(1H ,6X,4A4,5X,I5,12X,I5,12X,I5,7X,I5,11X,I5,10X,F14.5)        ENTSVA
      IF(NASYMC.EQ.0)RETURN                                                ENTSVA
C9.0      IF ASYMMETRIC ATOMS WERE FOUND BY PROGRAM, PRINT OUT THEIR       ENTSVA
C         GROUP NUMBERS.                                                   ENTSVA
      WRITE(6,49)(IOPATM(K),K=1,NASYMC)                                    ENTSVA
   49 FORMAT(//1H0,16X,46HGROUP NUMBER OF ASYMMETRIC CARBON ATOM(S):  *    ENTSVA
     1,10(I5,1H,),/(1H ,32X,10(I5,1H,)))                                   ENTSVA
      IF(NPSUDA.EQ.0)RETURN                                                ENTSVA
C         IF PSEUDOASYMMETRIC ATOM IS PRESENT, PRINT OUT ITS GROUP         ENTSVA
C         NUMBER.                                                          ENTSVA
      WRITE(6,51)ICSUBA                                                    ENTSVA
   51 FORMAT(//1H0,18X,49HGROUP NUMBER OF PSEUDOASYMMETRIC CARBON ATOM     ENTSVA
     1H ,I5)                                                              ENTSVA
      RETURN                                                               ENTSVA
      END                                                                  ENTSVA
```

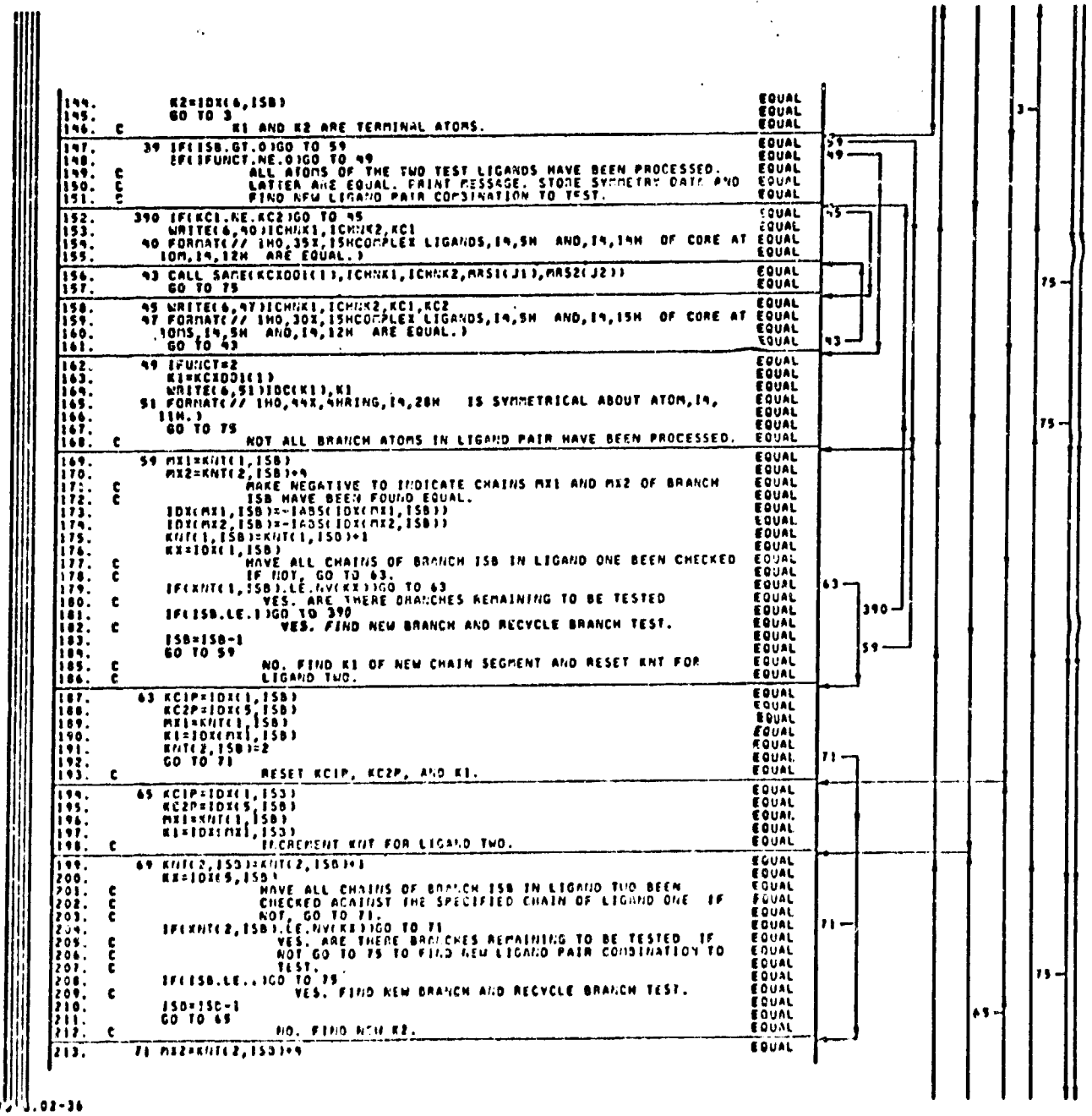
19 FEB 73 8.08-38

```
      SUBROUTINE EQUAL(KCXD01,KCXD02,NOD01,NOD02,MULT1,LF,KF,MULT2,KY,       EQUAL
     1IFUNCT)                                                                 EQUAL
C        THIS SUBROUTINE PROCESSES THE OPERATION WHICH CHECKS THE             EQUAL
C        STRUCTURE OF TWO OR MORE LIGANDS AND DETERMINES WHICH IS EQUAL       EQUAL
C        TO WHICH (IF ANY).                                                   EQUAL
      INTEGER SYMX(49),SYMBOL(9),GRID(50,80)                                  EQUAL
      INTEGER WEIGHT(9)                                                       EQUAL
      INTEGER PERTKC(4,100),PERLIG(4,100)                                     EQUAL
      DIMENSION KSAME(4,3,2),IA(4),KNT(2,99),IDX(8,99),NV(100),MCY(100),      EQUAL
     1MER(3,40),MKT(40),NDONE(100),KCXD01(3),KCXD02(3),MAS1(4),MAS2(4)        EQUAL
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                            EQUAL
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC           EQUAL
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),        EQUAL
     1IDBR(100),IB(100,8),IRO,NOBR                                            EQUAL
      COMMON/BLK4/NBC(60,50),NOS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT           EQUAL
      EQUIVALENCE (KSAME(1,1,1),IB(14,1)),(PERTKC(1,1),NBX(2,5)),             EQUAL
     1(IA(1),IB(75,8)),(KNT(1,1),GRID(1,33)),(IDX(1,1),GRID(3,1)),            EQUAL
     2(MCY(1),GRID(1,31)),(NV(1),GRID(1,29)),(MER(1,1),GRID(1,37)),           EQUAL
     3(KCY,IB(97,8)),(MKT(1),IB(35,2)),(PERLIG(1,1),NBX(42,11)),              EQUAL
     4(MAS1(1),NBX(24,20)),(MAS2(1),NBX(24,20)),(NDONE(1),GRID(50,78))        EQUAL
C1.0     INITIALIZE ARRAY NDONE.                                              EQUAL
      DO1    K=1,KCC                                                          EQUAL
    1 NDONE(K)=0                                                              EQUAL
C2.0     START EXECUTION OF MAIN CYCLE THAT CHECKS THE STRUCTURES OF TWO      EQUAL
C        OR MORE LIGANDS TO ESTABLISH WHETHER THEY ARE IDENTICAL.             EQUAL
C        II INCREMENTS THE SETS OF IDENTICAL LIGANDS WHOSE ATOMIC             EQUAL
C        COMPOSITION VARY FROM SET TO SET.                                    EQUAL
      DO95   II=1,KY                                                          EQUAL
C        MM AND NN CONTROL THE SETTING OF THE CORE ATOMS (KC1 AND KC2)        EQUAL
C        TO WHICH THE LIGANDS ARE BONDED.                                     EQUAL
      DO93   MM=1,NOD01                                                       EQUAL
      J1=MM                                                                   EQUAL
      KC1=KCXD01(MM)                                                          EQUAL
      NI=1+MULT1*MM                                                           EQUAL
      DO93   NN=NI,NOD02                                                      EQUAL
      J2=NN                                                                   EQUAL
      KC2=KCXD02(NN)                                                          EQUAL
C        JJ AND KK CONTROL THE SETTING OF THE TWO LIGANDS WHOSE               EQUAL
C        STRUCTURES ARE TO BE COMPARED. THE FIRST CORE ATOMS OF THESE         EQUAL
C        LIGANDS ARE K1 AND K2, RESPECTIVELY.                                 EQUAL
      DO85   JJ=1,LF                                                          EQUAL
      IF(NOD02.EQ.1)J1=JJ                                                     EQUAL
      IF(IFUNCT.EQ.0)GO TO 300                                                EQUAL
      K1=KCXD01(2)                                                            EQUAL
      GO TO 302                                                               EQUAL
  300 IF(KSAME(J1,II,1).EQ.0)GO TO 85                                         EQUAL
      LG1=PERLIG(JJ,KC1)                                                      EQUAL
      K1=PERTKC(LG1,KC1)                                                      EQUAL
  302 ICHNK1=K1                                                               EQUAL
      KI=1+MULT2*JJ                                                           EQUAL
      DO75   KK=KI,KF                                                         EQUAL
      IF(NOD02.EQ.1)J2=KK                                                     EQUAL
      IF(IFUNCT.EQ.0)GO TO 304                                                EQUAL
      K2=KCXD01(3)                                                            EQUAL
      GO TO 306                                                               EQUAL
  304 IF(KSAME(J2,II,2).EQ.0)GO TO 75                                         EQUAL
      LG2=PERLIG(KK,KC2)                                                      EQUAL
      K2=PERTKC(LG2,KC2)                                                      EQUAL
  306 K1=ICHNK1                                                               EQUAL
      IF(NDONE(K1).NE.0.AND.NDONE(K2).NE.0)GO TO 75                           EQUAL
C           K1 AND K2 HAVE NOT BEEN PROCESSED.                                EQUAL
      NDONE(K1)=1                                                             EQUAL
      NDONE(K2)=1                                                             EQUAL
      KC1P=KC1                                                                EQUAL
      KC2P=KC2                                                                EQUAL
      ISO=0                                                                   EQUAL
      ICHNK2=K2                                                               EQUAL
    3 IF(IX(K1,1,1).NE.IX(K2,1,1).OR.NV(K1).NE.NV(K2).OR.KON(K1).NE.         EQUAL
     1KON(K2))GO TO 65                                                        EQUAL
C        ATOMS K1 AND K2 ARE THE SAME, ARE BONDED TO THE SAME NUMBER          EQUAL
C        OF CORE ATOMS, AND HAVE THE SAME OVERALL CONNECTIVITIES.             EQUAL
      KONT=KON(K1)                                                            EQUAL
C        DETERMINE IF K1 AND K2 ARE BONDED TO THE SAME TYPES OF ATOMS.        EQUAL
```

17 APR 7[illegible] [illegible].02-36



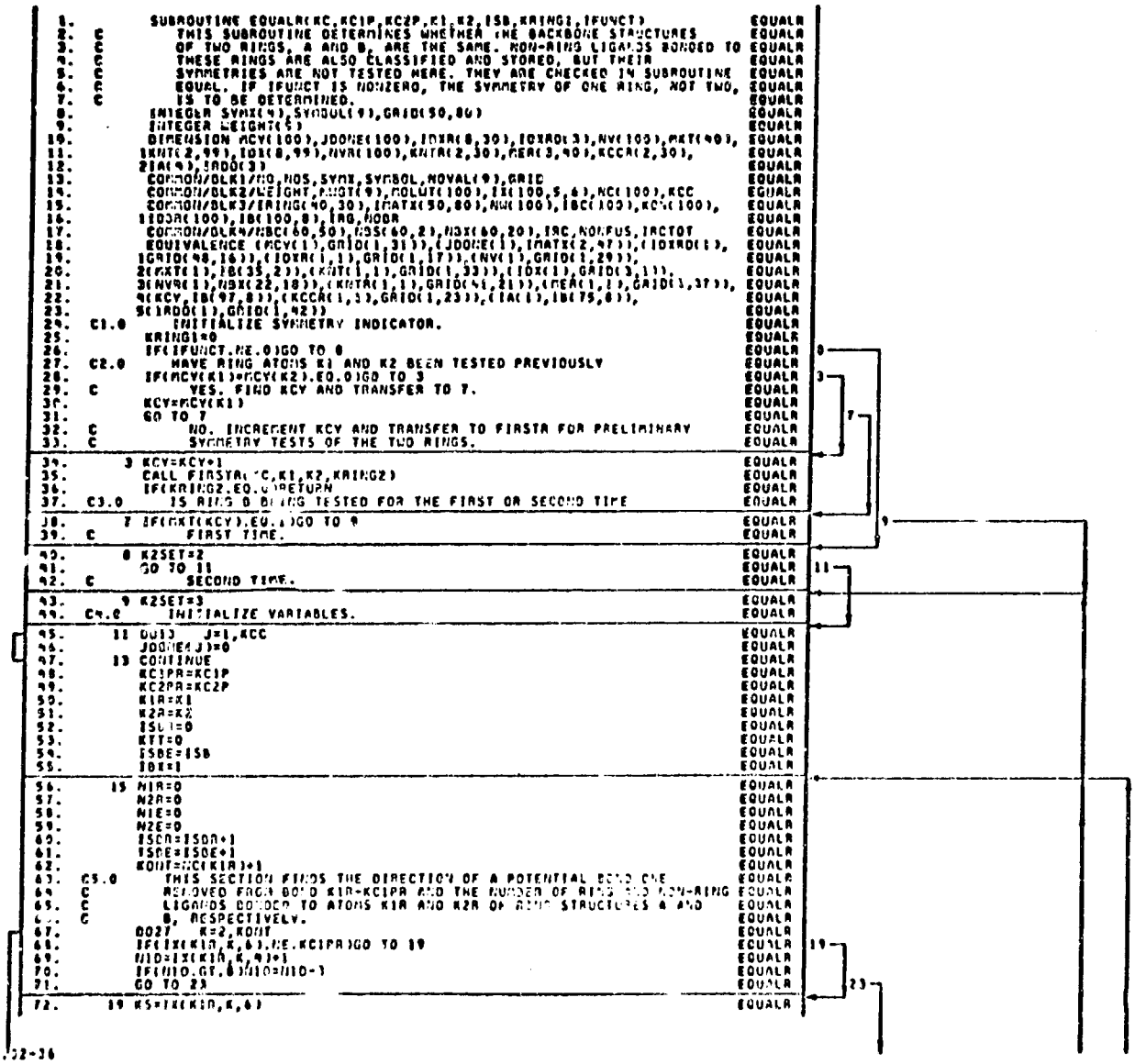
```
      C          IF NOT, GO TO 65. OTHERWISE, CONTINUE.                    EQUAL
           DO5   K=1,KONT                                                  EQUAL
           IACK)=IX(K2,K+1,1)                                              EQUAL
         5 CONTINUE                                                        EQUAL
           DO11  J=1,KONT                                                  EQUAL
           DO9   K=1,KONT                                                  EQUAL
           IF(IX(K1,J+1,1).EQ.IACK))GO TO 10                               EQUAL
         9 CONTINUE                                                        EQUAL
           GO TO 65                                                        EQUAL
        10 IACK)=0                                                         EQUAL
        11 CONTINUE                                                        EQUAL
      C          K1 AND K2 ARE THE SAME. IF A SINGLE RING IS BEING ANALYZED, EQUAL
      C          TRANSFER TO 27. OTHERWISE, FIND THE NEW VALUES OF K1 AND K2, EQUAL
      C          NAMELY, THOSE UNPROCESSED CORE ATOMS WHICH ARE BONDED TO THE EQUAL
      C          OLD K1 AND K2 PAIR, EACH OF WHICH IS LOCATED IN ONE OR THE EQUAL
      C          OTHER OF THE TWO TEST LIGANDS.                            EQUAL
           IF(IFUNCT.NE.0)GO TO 27                                         EQUAL
           IF(NV(K1)-2)39,15,27                                            EQUAL
      C          K1 AND K2 ARE CHAIN ATOMS. FIND THE NEW K1 AND K2.        EQUAL
        15 IF(IX(K1,2,6).NE.KC1P)GO TO 17                                  EQUAL
           KC1P=K1                                                         EQUAL
           K1=IX(K1,3,6)                                                   EQUAL
           GO TO 19                                                        EQUAL
        17 KC1P=K1                                                         EQUAL
           K1=IX(K1,2,6)                                                   EQUAL
        19 IF(IX(K2,2,6).NE.KC2P)GO TO 21                                  EQUAL
           KC2P=K2                                                         EQUAL
           K2=IX(K2,3,6)                                                   EQUAL
           GO TO 3                                                         EQUAL
        21 KC2P=K2                                                         EQUAL
           K2=IX(K2,2,6)                                                   EQUAL
           GO TO 3                                                         EQUAL
      C          K1 AND K2 ARE BRANCH ATOMS. FIND THE NEW K1 AND K2.       EQUAL
        27 IF(IBC(K1)+IBC(K2).NE.0)GO TO 37                                EQUAL
      C               NEITHER K1 NOR K2 ARE RING ATOMS.                    EQUAL
           ISB=ISB+1                                                       EQUAL
      C               STORE K1 AND K2 IN IDX. ISB IS THE COUNTER THAT      EQUAL
      C               SPECIFIES THE NUMBER OF BRANCHES ENCOUNTERED.        EQUAL
           IDX(1,ISB)=K1                                                   EQUAL
           IDX(5,ISB)=K2                                                   EQUAL
           N1=1                                                            EQUAL
           N2=5                                                            EQUAL
           N1F=NV(K1)+1                                                    EQUAL
      C               ALSO STORE THEREIN THE GROUP NUMBERS OF THE CORE ATOMS EQUAL
      C               OF K1 AND K2 WHICH HAVE TO BE CHECKED FOR EQUALITY.  EQUAL
           DO35  L=2,N1F                                                   EQUAL
           IF(IX(K1,N,6).EQ.KC1P)GO TO 31                                  EQUAL
           N1=N1+1                                                         EQUAL
           IDX(N1,ISB)=IX(K1,N,6)                                          EQUAL
        31 IF(IX(K2,N,6).EQ.KC2P)GO TO 35                                  EQUAL
           N2=N2+1                                                         EQUAL
           IDX(N2,ISB)=IX(K2,N,6)                                          EQUAL
        35 CONTINUE                                                        EQUAL
           NUT(1,ISB)=2                                                    EQUAL
           NUT(2,ISB)=2                                                    EQUAL
           KC1P=K1                                                         EQUAL
      C               FIND NEW K1 AND K2 AND RECYCLE.                      EQUAL
           K1=IDX(2,ISB)                                                   EQUAL
           KC2P=K2                                                         EQUAL
           K2=IDX(6,ISB)                                                   EQUAL
           GO TO 3                                                         EQUAL
      C               K1 AND/OR K2 ARE/IS RING ATOM(S).                    EQUAL
        37 IF(IBC(K1)+IBC(K2).EQ.0)GO TO 65                                EQUAL
      C               BOTH K1 AND K2 ARE RING ATOMS. CHECK THEIR EQUALITY. EQUAL
           CALL EQUALP(KC1,KC1P,KC2P,K1,K2,ISB,KRING1,IFUNCT)              EQUAL
           IF(IFUNCT.EQ.0)GO TO 69                                         EQUAL
           IF(ISB.EQ.0)GO TO 39                                            EQUAL
        38 KC1P=IDX(1,ISB)                                                 EQUAL
           KC2P=IDX(5,ISB)                                                 EQUAL
           K1=IDX(2,ISB)                                                   EQUAL
```



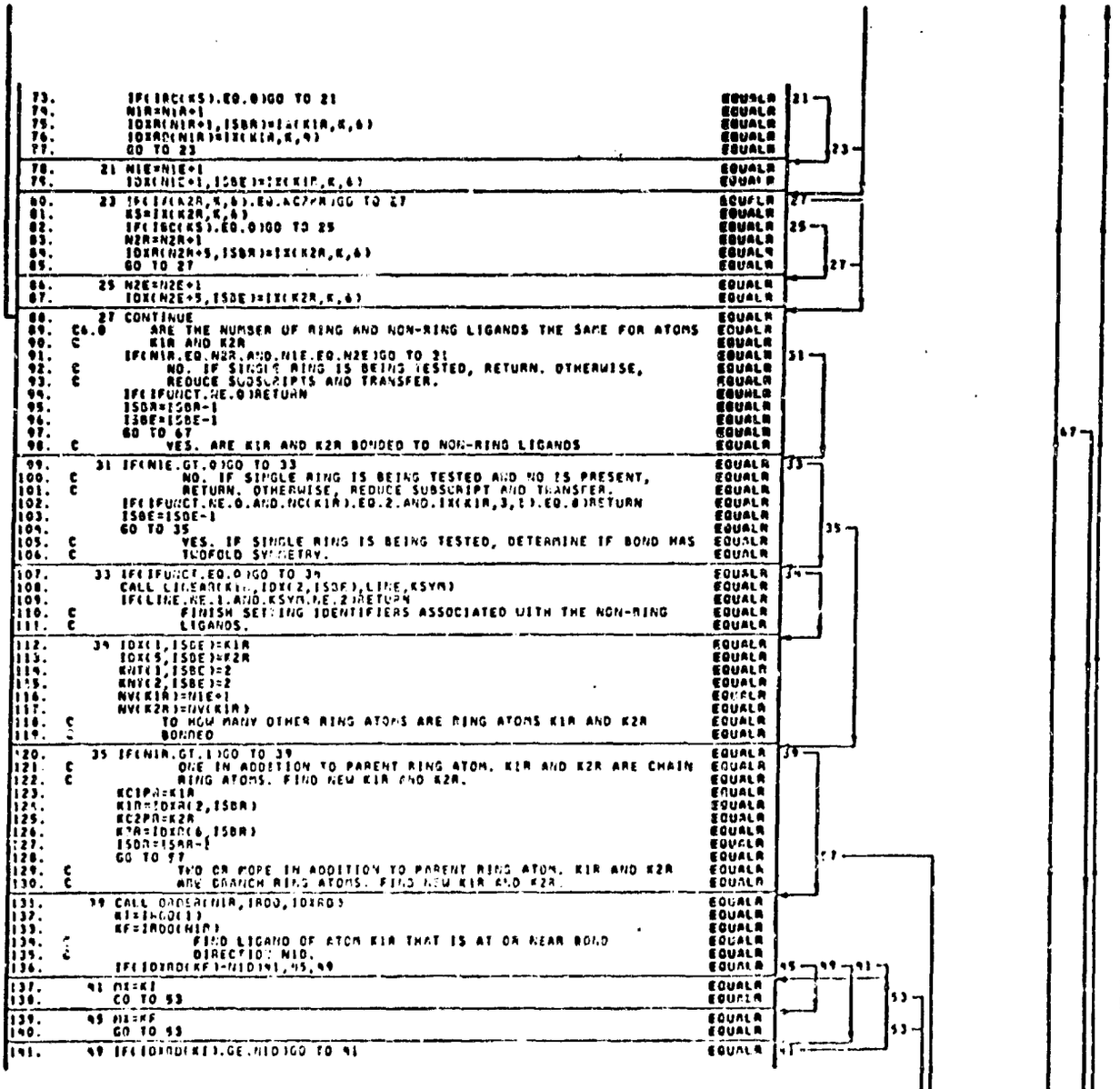
```
144.          K2=IDX(6,ISB)                                                    EQUAL
145.          GO TO 3                                                          EQUAL
146.    C               K1 AND K2 ARE TERMINAL ATOMS.                          EQUAL
147.       39 IF(ISB.GT.0)GO TO 59                                             EQUAL
148.          IF(IFUNCT.NE.0)GO TO 49                                          EQUAL
149.    C               ALL ATOMS OF THE TWO TEST LIGANDS HAVE BEEN PROCESSED. EQUAL
150.    C               LATTER ARE EQUAL. PRINT MESSAGE. STORE SYMETRY DATA AND EQUAL
151.    C               FIND NEW LIGAND PAIR COMBINATION TO TEST.              EQUAL
152.      390 IF(KC1.NE.KC2)GO TO 45                                           EQUAL
153.          WRITE(6,40)ICHNK1,ICHNK2,KC1                                     EQUAL
154.       40 FORMAT(// 1HO,35X,15HCOMPLEX LIGANDS,I4,5H  AND,I4,14H  OF CORE AT EQUAL
155.         1OM,I4,12H  ARE EQUAL.)                                           EQUAL
156.          CALL SAME(KCIDO1(1),ICHNK1,ICHNK2,MRS1(J1),MRS2(J2))             EQUAL
157.          GO TO 75                                                         EQUAL
158.       45 WRITE(6,47)ICHNK1,ICHNK2,KC1,KC2                                 EQUAL
159.       47 FORMAT(// 1HO,30X,15HCOMPLEX LIGANDS,I4,5H  AND,I4,15H  OF CORE AT EQUAL
160.         1OMS,I4,5H  AND,I4,12H  ARE EQUAL.)                               EQUAL
161.          GO TO 43                                                         EQUAL
162.       49 IFUNCT=2                                                         EQUAL
163.          K1=KCIDO1(1)                                                     EQUAL
164.          WRITE(6,51)IDC(K1),K1                                            EQUAL
165.       51 FORMAT(// 1HO,44X,4HRING,I4,28H  IS SYMMETRICAL ABOUT ATOM,I4,   EQUAL
166.         1H.)                                                              EQUAL
167.          GO TO 75                                                         EQUAL
168.    C               NOT ALL BRANCH ATOMS IN LIGAND PAIR HAVE BEEN PROCESSED. EQUAL
169.       59 MI1=KNT(1,ISB)                                                   EQUAL
170.          MI2=KNT(2,ISB)+9                                                 EQUAL
171.    C               MAKE NEGATIVE TO INDICATE CHAINS MI1 AND MI2 OF BRANCH EQUAL
172.    C               ISB HAVE BEEN FOUND EQUAL.                             EQUAL
173.          IDX(MI1,ISB)=-IABS(IDX(MI1,ISB))                                 EQUAL
174.          IDX(MI2,ISB)=-IABS(IDX(MI2,ISB))                                 EQUAL
175.          KNT(1,ISB)=KNT(1,ISB)+1                                          EQUAL
176.          KX=IDX(1,ISB)                                                    EQUAL
177.    C               HAVE ALL CHAINS OF BRANCH ISB IN LIGAND ONE BEEN CHECKED EQUAL
178.    C               IF NOT, GO TO 63.                                      EQUAL
179.          IF(KNT(1,ISB).LE.KX)GO TO 63                                     EQUAL
180.    C               YES. ARE THERE BRANCHES REMAINING TO BE TESTED         EQUAL
181.          IF(ISB.LE.1)GO TO 390                                            EQUAL
182.    C                     YES. FIND NEW BRANCH AND RECYCLE BRANCH TEST.    EQUAL
183.          ISB=ISB-1                                                        EQUAL
184.          GO TO 59                                                         EQUAL
185.    C               NO. FIND K1 OF NEW CHAIN SEGMENT AND RESET KNT FOR     EQUAL
186.    C               LIGAND TWO.                                            EQUAL
187.       63 KC1P=IDX(1,ISB)                                                  EQUAL
188.          KC2P=IDX(5,ISB)                                                  EQUAL
189.          MI1=KNT(1,ISB)                                                   EQUAL
190.          K1=IDX(MI1,ISB)                                                  EQUAL
191.          KNT(2,ISB)=2                                                     EQUAL
192.          GO TO 71                                                         EQUAL
193.    C               RESET KC1P, KC2P, AND K1.                              EQUAL
194.       65 KC1P=IDX(1,ISB)                                                  EQUAL
195.          KC2P=IDX(5,ISB)                                                  EQUAL
196.          MI1=KNT(1,ISB)                                                   EQUAL
197.          K1=IDX(MI1,ISB)                                                  EQUAL
198.    C               INCREMENT KNT FOR LIGAND TWO.                          EQUAL
199.       69 KNT(2,ISB)=KNT(2,ISB)+1                                          EQUAL
200.          KX=IDX(5,ISB)                                                    EQUAL
201.    C               HAVE ALL CHAINS OF BRANCH ISB IN LIGAND TWO BEEN       EQUAL
202.    C               CHECKED AGAINST THE SPECIFIED CHAIN OF LIGAND ONE  IF  EQUAL
203.    C               NOT, GO TO 71.                                         EQUAL
204.          IF(KNT(2,ISB).LE.KX)GO TO 71                                     EQUAL
205.    C                  YES. ARE THERE BRANCHES REMAINING TO BE TESTED  IF  EQUAL
206.    C                  NOT GO TO 75 TO FIND NEW LIGAND PAIR COMBINATION TO EQUAL
207.    C                  TEST.                                               EQUAL
208.          IF(ISB.LE..)GO TO 75                                             EQUAL
209.    C                        YES. FIND NEW BRANCH AND RECYCLE BRANCH TEST.  EQUAL
210.          ISB=ISB-1                                                        EQUAL
211.          GO TO 69                                                         EQUAL
212.    C                        NO. FIND NEW K2.                             EQUAL
213.       71 MI2=KNT(2,ISB)+9                                                 EQUAL
```

17 APR 7[illegible].02-36

```
      IF(IDX(MX2,ISB).LT.0)GO TO 69                                     EQUAL
      K2=IDX(MX2,ISB)                                                   EQUAL
C                   IF NEITHER K1 NOR K2 ARE RING ATOMS, GO TO 3 (RECYCLE EQUAL
C                   TEST).                                              EQUAL
      IF(IBC(K1)+IBC(K2).EQ.0)GO TO 3                                   EQUAL
      IF(IBC(K1)*IBC(K2).EQ.0)GO TO 65                                  EQUAL
C                   BOTH K1 AND K2 ARE RING ATOMS. CHECK THEIR EQUALITY. EQUAL
      IF(MCY(K1)+MCY(K2).EQ.0)GO TO 72                                  EQUAL
      IF(MCY(K1).NE.MCY(K2))GO TO 65                                    EQUAL
      KCY=MCY(K1)                                                       EQUAL
      ISB=MER(3,KCY)                                                    EQUAL
      KC1P=IDX(1,ISB)                                                   EQUAL
      KC2P=IDX(5,ISB)                                                   EQUAL
      K1=MER(1,KCY)                                                     EQUAL
      K2=MER(2,KCY)                                                     EQUAL
      IF(MXT(KCY).EQ.1)GO TO 69                                         EQUAL
   72 CALL EQUALR(KC1,KC1P,KC2P,K1,K2,ISB,KRING1,IFUNCT)                EQUAL
      IF(KRING1.EQ.0)GO TO 69                                           EQUAL
      IF(ISB.EQ.MER(3,KCY))GO TO 38                                     EQUAL
      GO TO 3                                                           EQUAL
   75 CONTINUE                                                          EQUAL
   85 CONTINUE                                                          EQUAL
   93 CONTINUE                                                          EQUAL
   95 CONTINUE                                                          EQUAL
      RETURN                                                            EQUAL
      END                                                               EQUAL
```



```
      SUBROUTINE EQUALR(KC,KC1P,KC2P,K1,K2,ISB,KRING2,IFUNCT)                   EQUALR
C        THIS SUBROUTINE DETERMINES WHETHER THE BACKBONE STRUCTURES             EQUALR
C        OF TWO RINGS, A AND B, ARE THE SAME. NON-RING LIGANDS BONDED TO        EQUALR
C        THESE RINGS ARE ALSO CLASSIFIED AND STORED, BUT THEIR                  EQUALR
C        SYMMETRIES ARE NOT TESTED HERE. THEY ARE CHECKED IN SUBROUTINE         EQUALR
C        EQUAL. IF IFUNCT IS NONZERO, THE SYMMETRY OF ONE RING, NOT TWO,        EQUALR
C        IS TO BE DETERMINED.                                                   EQUALR
      INTEGER SYMX(9),SYMBOL(9),GRID(50,80)                                     EQUALR
      INTEGER WEIGHT(5)                                                         EQUALR
      DIMENSION MCY(100),JDONE(100),IDXR(8,30),IDXRO(3),NVC(100),MXT(40),       EQUALR
     1KNTC(2,99),IDX(8,99),NVR(100),KNTR(2,30),MER(3,40),KCCR(2,30),            EQUALR
     2IAC(9),IRDO(3)                                                            EQUALR
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                              EQUALR
      COMMON/BLK2/WEIGHT,MXT(9),MOLWT(100),IE(100,5,6),NC(100),KCC              EQUALR
      COMMON/BLK3/ERING(40,30),IMATX(50,80),NW(100),IBC(100),KCY(100),          EQUALR
     1IDXR(100),IB(100,8),IRG,NOOR                                              EQUALR
      COMMON/BLK4/XBC(60,50),XDSC(60,2),NDX(60,20),IRC,NOMPUS,IRCTOT            EQUALR
      EQUIVALENCE (MCY(1),GRID(1,31)),(JDONE(1),IMATX(2,47)),(IDXRO(1),         EQUALR
     1GRID(48,16)),(IDXR(1,1),GRID(1,7)),(NVC(1),GRID(1,29)),                   EQUALR
     2(MXT(1),IB(35,2)),(KNTC(1,1),GRID(1,33)),(IDX(1),GRID(3,1)),              EQUALR
     3(NVR(1),NDX(22,18)),(KNTR(1,1),GRID(41,21)),(MER(1,1),GRID(1,37)),        EQUALR
     4(KCY,IB(97,8)),(KCCR(1,1),GRID(1,23)),(IAC(1),IB(75,8)),                  EQUALR
     5(IRDO(1),GRID(1,42))                                                      EQUALR
C1.0     INITIALIZE SYMMETRY INDICATOR.                                         EQUALR
      KRING1=0                                                                  EQUALR
      IF(IFUNCT.NE.0)GO TO 8                                                    EQUALR
C2.0     HAVE RING ATOMS K1 AND K2 BEEN TESTED PREVIOUSLY                       EQUALR
      IF(MCY(K1)+MCY(K2).EQ.0)GO TO 3                                           EQUALR
C           YES. FIND KCY AND TRANSFER TO 7.                                    EQUALR
      KCY=MCY(K1)                                                               EQUALR
      GO TO 7                                                                   EQUALR
C           NO. INCREMENT KCY AND TRANSFER TO FIRSTA FOR PRELIMINARY           EQUALR
C           SYMMETRY TESTS OF THE TWO RINGS.                                    EQUALR
    3 KCY=KCY+1                                                                 EQUALR
      CALL FIRSTA(KC,K1,K2,KRING2)                                              EQUALR
      IF(KRING2.EQ.0)RETURN                                                     EQUALR
C3.0     IS RING 8 BEING TESTED FOR THE FIRST OR SECOND TIME                    EQUALR
    7 IF(MXT(KCY).EQ.0)GO TO 9                                                  EQUALR
C           FIRST TIME.                                                         EQUALR
    8 K2SET=2                                                                   EQUALR
      GO TO 11                                                                  EQUALR
C           SECOND TIME.                                                        EQUALR
    9 K2SET=3                                                                   EQUALR
C9.0     INITIALIZE VARIABLES.                                                  EQUALR
   11 DO13   J=1,KCC                                                            EQUALR
      JDONE(J)=0                                                                EQUALR
   13 CONTINUE                                                                  EQUALR
      KC1PR=KC1P                                                                EQUALR
      KC2PR=KC2P                                                                EQUALR
      K1R=K1                                                                    EQUALR
      K2R=K2                                                                    EQUALR
      ISUB=0                                                                    EQUALR
      KTT=0                                                                     EQUALR
      ISBE=ISB                                                                  EQUALR
      IBE=1                                                                     EQUALR
   15 N1R=0                                                                     EQUALR
      N2R=0                                                                     EQUALR
      N1E=0                                                                     EQUALR
      N2E=0                                                                     EQUALR
      ISBR=ISBR+1                                                               EQUALR
      ISBE=ISBE+1                                                               EQUALR
      KONT=NC(K1R)+1                                                            EQUALR
C5.0     THIS SECTION FINDS THE DIRECTION OF A POTENTIAL BOND ONE               EQUALR
C        REMOVED FROM BOND K1R-KC1PR AND THE NUMBER OF RING AND NON-RING        EQUALR
C        LIGANDS BONDED TO ATOMS K1R AND K2R OF RING STRUCTURES A AND           EQUALR
C        B, RESPECTIVELY.                                                       EQUALR
      DO27   K=2,KONT                                                           EQUALR
      IF(IE(K1R,K,6).NE.KC1PR)GO TO 19                                          EQUALR
      N1D=IE(K1R,K,4)+1                                                         EQUALR
      IF(N1D.GT.8)N1D=N1D-1                                                     EQUALR
      GO TO 23                                                                  EQUALR
   19 KS=IE(K1R,K,6)                                                            EQUALR
```



```
      IF(IRC(K$).EQ.0)GO TO 21                                          EQUALR
      NIR=NIR+1                                                         EQUALR
      IDXR(NIR+1,ISBR)=IX(K1R,K,6)                                      EQUALR
      IDXR(NIR)=IX(K1R,K,9)                                             EQUALR
      GO TO 23                                                          EQUALR
   21 NIE=NIE+1                                                         EQUALR
      IDXE(NIE+1,ISBE)=IX(K1P,K,6)                                      EQUALR
   23 IF(IX(K2R,K,6).EQ.ACZPR)GO TO 27                                  EQUALR
      K$=IX(K2R,K,6)                                                    EQUALR
      IF(IRC(K$).EQ.0)GO TO 25                                          EQUALR
      N2R=N2R+1                                                         EQUALR
      IDXR(N2R+5,ISBR)=IX(K2R,K,6)                                      EQUALR
      GO TO 27                                                          EQUALR
   25 N2E=N2E+1                                                         EQUALR
      IDX(N2E+5,ISBE)=IX(K2R,K,6)                                       EQUALR
   27 CONTINUE                                                          EQUALR
C6.0     ARE THE NUMBER OF RING AND NON-RING LIGANDS THE SAME FOR ATOMS EQUALR
C        K1R AND K2R                                                    EQUALR
      IF(NIR.EQ.N2R.AND.NIE.EQ.N2E)GO TO 31                             EQUALR
C           NO. IF SINGLE RING IS BEING TESTED, RETURN. OTHERWISE,      EQUALR
C           REDUCE SUBSCRIPTS AND TRANSFER.                             EQUALR
      IF(IFUNCT.NE.0)RETURN                                             EQUALR
      ISBR=ISBR-1                                                       EQUALR
      ISBE=ISBE-1                                                       EQUALR
      GO TO 67                                                          EQUALR
C           YES. ARE K1R AND K2R BONDED TO NON-RING LIGANDS             EQUALR
   31 IF(NIE.GT.0)GO TO 33                                              EQUALR
C              NO. IF SINGLE RING IS BEING TESTED AND NO ES PRESENT,    EQUALR
C              RETURN. OTHERWISE, REDUCE SUBSCRIPT AND TRANSFER.        EQUALR
      IF(IFUNCT.NE.0.AND.NC(K1R).EQ.2.AND.IX(K1R,3,2).EQ.0)RETURN       EQUALR
      ISBE=ISBE-1                                                       EQUALR
      GO TO 35                                                          EQUALR
C              YES. IF SINGLE RING IS BEING TESTED, DETERMINE IF BOND HAS EQUALR
C              TWOFOLD SYMMETRY.                                        EQUALR
   33 IF(IFUNCT.EQ.0)GO TO 34                                           EQUALR
      CALL LINEARCK(,IDX(2,ISBE),LINE,KSYM)                             EQUALR
      IF(LINE.NE.1.AND.KSYM.NE.2)RETURN                                 EQUALR
C              FINISH SETTING IDENTIFIERS ASSOCIATED WITH THE NON-RING  EQUALR
C              LIGANDS.                                                 EQUALR
   34 IDX(1,ISBE)=K1R                                                   EQUALR
      IDX(5,ISBE)=K2R                                                   EQUALR
      KNT(1,ISBE)=2                                                     EQUALR
      KNY(2,ISBE)=2                                                     EQUALR
      NVE(K1R)=NIE+1                                                    EQUALR
      NVE(K2R)=NVE(K1R)                                                 EQUALR
C              TO HOW MANY OTHER RING ATOMS ARE RING ATOMS K1R AND K2R  EQUALR
C              BONDED                                                   EQUALR
   35 IF(NIR.GT.1)GO TO 39                                              EQUALR
C              ONE IN ADDITION TO PARENT RING ATOM, K1R AND K2R ARE CHAIN EQUALR
C              RING ATOMS. FIND NEW K1R AND K2R.                        EQUALR
      KC1P=K1R                                                          EQUALR
      K1R=IDXR(2,ISBR)                                                  EQUALR
      KC2P=K2R                                                          EQUALR
      K2R=IDXR(6,ISBR)                                                  EQUALR
      ISBR=ISBR-1                                                       EQUALR
      GO TO 47                                                          EQUALR
C              TWO OR MORE IN ADDITION TO PARENT RING ATOM, K1R AND K2R EQUALR
C              ARE BRANCH RING ATOMS. FIND NEW K1R AND K2R.             EQUALR
   39 CALL ORDER(NIR,IROD,IDXR0)                                        EQUALR
      NI=IROD(1)                                                        EQUALR
      KF=IROD(NIR)                                                      EQUALR
C                 FIND LIGAND OF ATOM K1R THAT IS AT OR NEAR BOND      EQUALR
C                 DIRECTION NI0.                                       EQUALR
      IF(IDXR0(KF)-NI0)41,45,49                                         EQUALR
   41 NI=KI                                                             EQUALR
      GO TO 53                                                          EQUALR
   45 NI=KF                                                             EQUALR
      GO TO 53                                                          EQUALR
   49 IF(IDXR0(KF).GE.NI0)GO TO 41                                      EQUALR
```

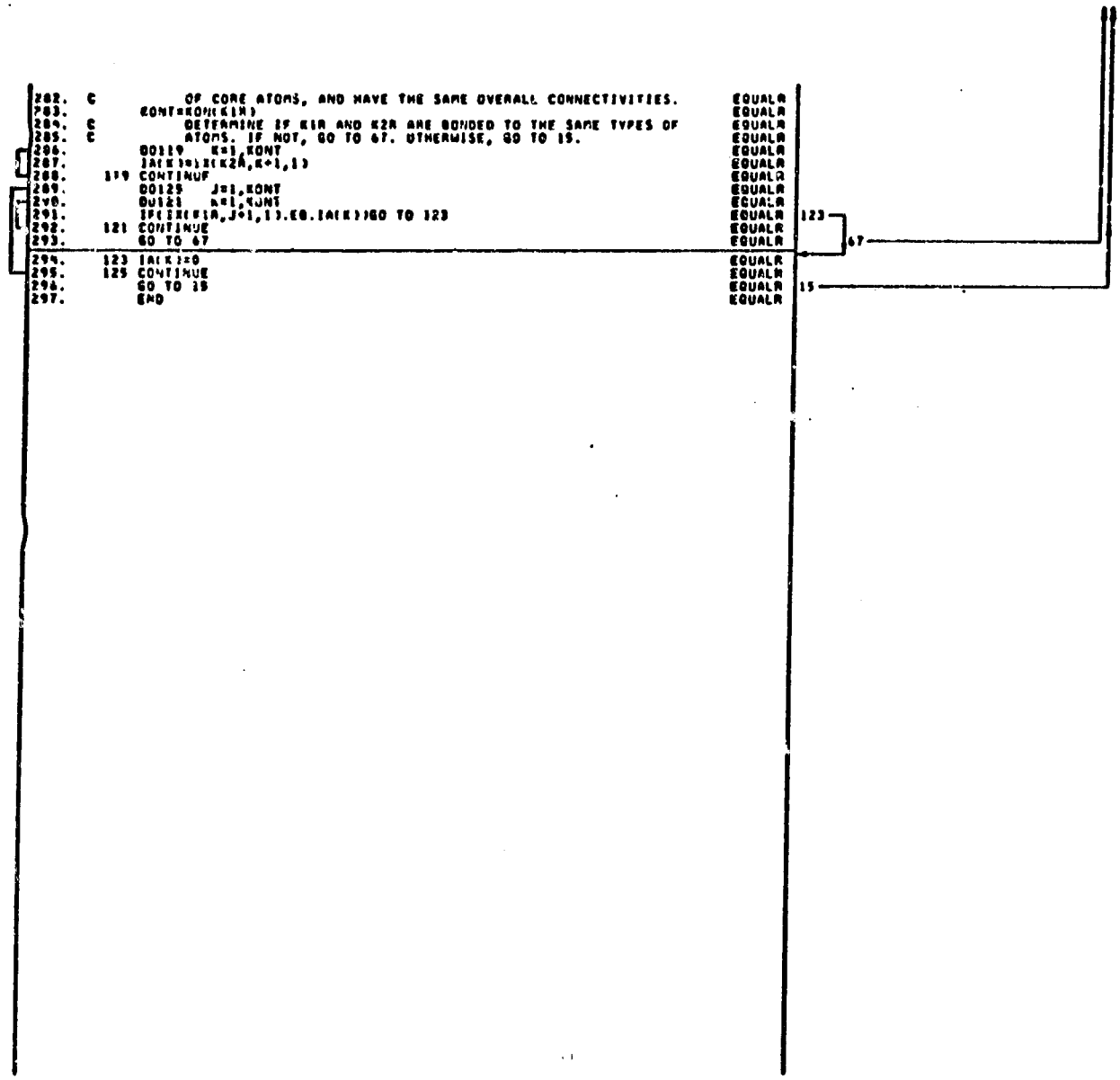
IF APR 73 8.02-36

```
      KS=1                                                              EQUALR
   51 KS=KS+1                                                           EQUALR
      MI=IRODE(KS)                                                      EQUALR
      IF(IDXR(MI).LT.NID)GO TO 51                                       EQUALR
   53 K=IDXR(2,ISBR)                                                    EQUALR
      IDXR(2,ISBR)=IDXR(MI+1,ISBR)                                      EQUALR
      IDXR(MI+1,ISBR)=K                                                 EQUALR
C        SET IDENTIFIERS ASSOCIATED WITH RING LIGANDS AND FIND          EQUALR
C           NEW K1R AND K2R.                                            EQUALR
      KI=0                                                              EQUALR
      IF(K1R.EQ.K1)KI=1                                                 EQUALR
      NVR(K1R)=NLR+1-KI                                                 EQUALR
      NVR(K2R)=NVR(K1R)                                                 EQUALR
      KNTR(1,ISBR)=2                                                    EQUALR
      KNTR(2,ISBR)=2                                                    EQUALR
      IF(ISBR.EQ.1)KNTR(2,ISBR)=K2SET                                   EQUALR
      IDXR(1,ISBR)=K1R                                                  EQUALR
      IDXR(5,ISBR)=K2R                                                  EQUALR
      KC1PR=K1R                                                         EQUALR
      K1R=IDXR(2,ISBR)                                                  EQUALR
      KC2PR=K2R                                                         EQUALR
      MI=KNTR(2,ISBR)+4                                                 EQUALR
      K2R=IDXR(MI,ISBR)                                                 EQUALR
   57 IF(IFUNCT.EQ.0.OR.K1R.NE.K2R)GO TO 63                             EQUALR
C           SINGLE RING IS BEING TESTED AND K1R EQUALS K2R.             EQUALR
C           DETERMINE IF FINAL LIGAND TO BE TESTED WHICH IS BONDED      EQUALR
C           TO THE RING HAS TWOFOLD SYMMETRY.                           EQUALR
      IF(NC(K1R).EQ.2)GO TO 62                                          EQUALR
      DO 58 K=2,4                                                       EQUALR
      KS=IX(K1R,K,6)                                                    EQUALR
      IF(IOC(KS).EQ.0)GO TO 59                                          EQUALR
   58 CONTINUE                                                          EQUALR
   59 CALL LINEAR(K1R,KS,LINE,KSYM)                                     EQUALR
      IF(LINE.NE.1.AND.KSYM.NE.2)RETURN                                 EQUALR
   60 KRING1=1                                                          EQUALR
      KCY=1                                                             EQUALR
      FER(3,KCY)=ISB                                                    EQUALR
   61 RETURN                                                            EQUALR
   62 IF(IX(K1R,4,1)-8)60,61,60                                         EQUALR
   63 JDON(KC1PR)=1                                                     EQUALR
      JDON(KC2PR)=1                                                     EQUALR
      KTT=KTT+1                                                         EQUALR
      KCCR(1,KTT)=KC1PR                                                 EQUALR
      KCCR(2,KTT)=KC2PR                                                 EQUALR
C4.0     ARE THE NEW K1R AND K2R ATOMS EQUAL TO THE STARTING VALUES     EQUALR
C           K1 AND K2, RESPECTIVELY.  IF SO, GO TO 85 AND CHANGE IBX.   EQUALR
      IF(K1R.EQ.K1.AND.K2R.EQ.K2)GO TO 85                               EQUALR
      GO TO(65,87),IBX                                                  EQUALR
C7.0     TRANSFER POINT WHEN IBX=1.                                     EQUALR
   65 IF(JDON(K2R).EQ.0)GO TO 115                                       EQUALR
C           K2R HAS BEEN TESTED PREVIOUSLY.  ARE THERE ANY BRANCH RING  EQUALR
C           ATOMS LEFT TO TEST.  IF NOT, GO TO 73.                      EQUALR
   67 IF(ISBR.LE.0)GO TO 73                                             EQUALR
C           DISENGAGE INTERVENING CHAIN ATOMS(IF ANY). FIND NEW K2R (AND EQUALR
C           K1R IF ALTERED).                                            EQUALR
      IF(KTT.GT.0)CALL DELETE(IDXR(1,ISBR),ISBE,KTT)                    EQUALR
      KC1PR=IDXR(1,ISBR)                                                EQUALR
      KC2PR=IDXR(5,ISBR)                                                EQUALR
      MI=KNTR(1,ISBR)                                                   EQUALR
      K1R=IDXR(MI,ISBR)                                                 EQUALR
      KNTR(2,ISBR)=KNTR(2,ISBR)+1                                       EQUALR
      MI=IDXR(5,ISBR)                                                   EQUALR
      IF(KNTR(2,ISBR).GT.NVR(MI))GO TO 71                               EQUALR
      MI=KNTR(2,ISBR)+4                                                 EQUALR
      K2R=IDXR(MI,ISBR)                                                 EQUALR
      GO TO 65                                                          EQUALR
C           ALL LIGANDS OF BRANCH ISBR IN RING B HAVE BEEN TESTED.  ARE EQUALR
C           THERE ANY FURTHER BRANCHES TO TEST                          EQUALR
   71 IF(ISBR.LE.1)GO TO 73                                             EQUALR
C              YES.  FIND NEW BRANCH SUBSCRIPT AND RETURN TO 67.        EQUALR
```


07 APR 73 8.02-36

```
            ISBR=ISBR-1                                                  EQUALR
            GO TO 67                                                     EQUALR
         73 IF(MKT(KCY).EQ.1)GO TO 103                                   EQUALR
C                   NO. REPEAT ENTIRE TEST CYCLE, SCANNING RING B IN THE EQUALR
C                   REVERSE DIRECTION.                                   EQUALR
            MKT(KCY)=1                                                   EQUALR
            GO TO 9                                                      EQUALR
C               RESET IBX AND GO TO 91.                                  EQUALR
         65 IBX=2                                                        EQUALR
            GO TO 91                                                     EQUALR
C8.0          TRANSFER POINT WHEN IBX=2.                                 EQUALR
C9.0          HAVE ATOMS K1R OR/AND K2R BEEN TESTED PREVIOUSLY  IF NOT,  EQUALR
C             GO TO 115.                                                 EQUALR
         67 IF(JDONE(K1R).EQ.1)GO TO 91                                  EQUALR
            IF(JDONE(K2R).EQ.0)GO TO 115                                 EQUALR
C               YES. FIND NEW K1R.                                       EQUALR
         91 KNTR(1,ISBR)=KNTR(1,ISBR)+1                                  EQUALR
            MX=IDXR(1,ISBR)                                              EQUALR
C                 HAVE ALL LIGANDS OF BRANCH ISBR IN RING A BEEN TESTED  EQUALR
C                  IF NOT, GO TO 99.                                     EQUALR
            IF(KNTR(1,ISBR).LE.NVR(MX))GO TO 99                          EQUALR
C                  ARE THERE ANY MORE BRANCHES TO TEST                   EQUALR
            IF(ISBR.LE.1)GO TO 92                                        EQUALR
C                    YES. FIND NEW BRANCH SUBSCRIPT AND RETURN TO 91.    EQUALR
            ISBR=ISBR-1                                                  EQUALR
            GO TO 91                                                     EQUALR
C                    NO. BACKBONE STRUCTURE OF BOTH RINGS IS IDENTICAL. SET EQUALR
C                    PERTINENT ARRAYS AND EXIT FROM ROUTINE.             EQUALR
         92 KRING1=1                                                     EQUALR
            MER(1,KCY)=K1                                                EQUALR
            MER(2,KCY)=K2                                                EQUALR
            MER(3,KCY)=ISB                                               EQUALR
            IF(ISBE.EQ.ISB)RETURN                                        EQUALR
            KI=ISB+1                                                     EQUALR
            DO93   K=KI,ISBE                                             EQUALR
            MX=IDX(1,K)                                                  EQUALR
            MCV(MX)=KCY                                                  EQUALR
            MX=IDX(5,K)                                                  EQUALR
            MCV(MX)=KCY                                                  EQUALR
         93 CONTINUE                                                     EQUALR
            RETURN                                                       EQUALR
C               SET K1R AND FIND NEW K2R TO TEST (IF ANY).               EQUALR
         99 KC1PR=IDXR(1,ISBR)                                           EQUALR
            KC2PR=IDXR(5,ISBR)                                           EQUALR
            MX=KNTR(1,ISBR)                                              EQUALR
            K1R=IDXR(MX,ISBR)                                            EQUALR
            KNTR(2,ISCR)=2                                               EQUALR
        101 MX=KNTR(2,ISCR)+4                                            EQUALR
            K2R=IDXR(MX,ISCR)                                            EQUALR
            IF(JDONE(K2R).EQ.0)GO TO 115                                 EQUALR
            KNTR(2,ISCR)=KNTR(2,ISCR)+1                                  EQUALR
            MX=IDXR(5,ISCR)                                              EQUALR
            IF(KNTR(2,ISCR).LE.NVR(MX))GO TO 101                         EQUALR
C               NO MORE NEW K2R ATOMS AVAILABLE.                         EQUALR
        103 MKT(KCY)=0                                                   EQUALR
            KCY=KCY-1                                                    EQUALR
            IF(MCV(K1).EQ.0)RETURN                                       EQUALR
C                   RESET ALL VALUES ASSOCIATED WITH KCY TO THEIR INITIAL EQUALR
C                   VALUES AND RETURN.                                   EQUALR
            DO107   K=1,KCC                                              EQUALR
            IF(MCV(K).LE.KCY)GO TO 107                                   EQUALR
            MX=MCV(K)                                                    EQUALR
            MST(MX)=0                                                    EQUALR
            MCV(K)=0                                                     EQUALR
        107 CONTINUE                                                     EQUALR
            RETURN                                                       EQUALR
C10.0        CHECK SIMILARITY OF RING ATOMS K1R AND K2R.                 EQUALR
        115 IF(IH(K1R,1,1).NE.IH(K2R,1,1).OR.NC(K1R).NE.NC(K2R).OR.KON(K1R). EQUALR
           1NE.KON(K2R))GO TO 67                                         EQUALR
C                 ATOMS K1R AND K2R ARE THE SAME, ARE BONDED TO THE SAME NUMBER EQUALR
```

17 APR 73 8.02-36



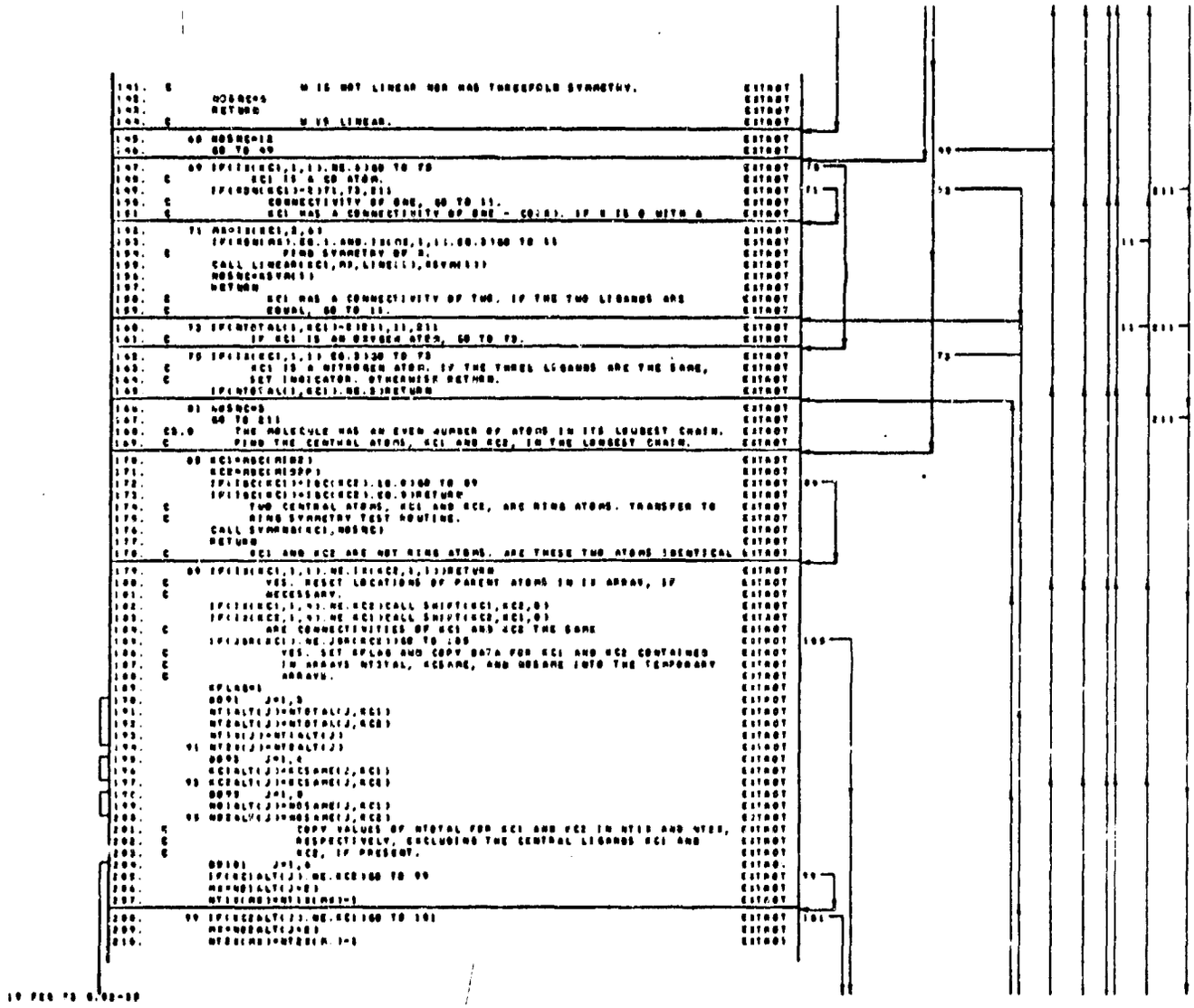
17 APR 73 0.02-36

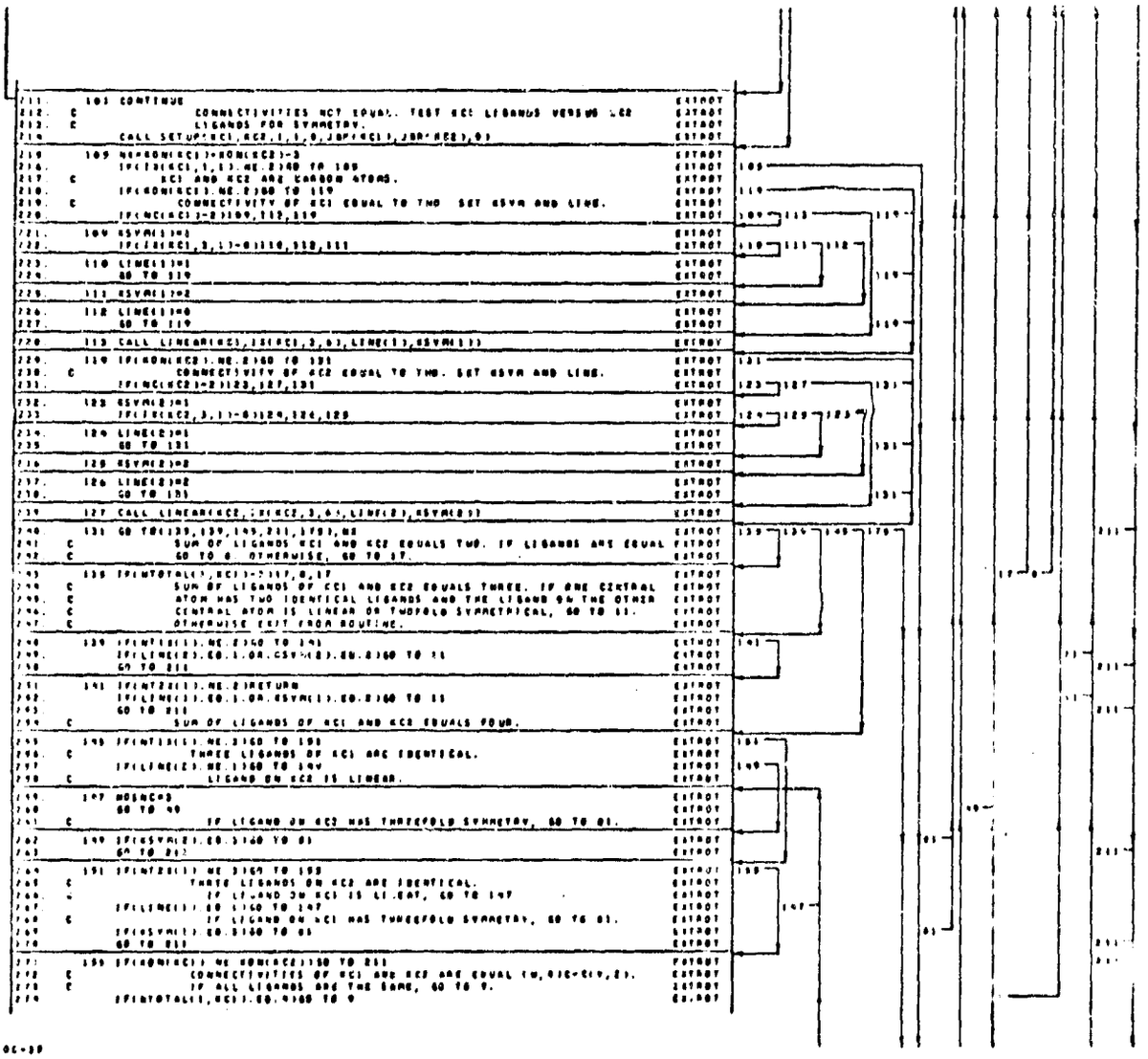
```
      SUBROUTINE EXTROT(NOSMC,NOSYMC,NPSUDO,ACSURA,NOMESO)
C        THIS SUBROUTINE COMPUTES THE EXTERNAL ROTATION SYMMETRY NUMBER
C        OF THE MOLECULE.
      INTEGER SYMB(9),SYMBOL(9),GRID(20,80)
      INTEGER WEIGHT(9)
      DIMENSION KINT(100),LINE(3),ASYM(3),KCSAME(6,100),NOSAME(6,100),
     1NTOTAL(3,100),KC1ALT(6),KC2ALT(6),NO1ALT(6),NO2ALT(6),NT1ALT(6),
     2NT2ALT(6),NT1S(3),NT2S(3),NS(9),NNT(9),JBR(100)
      COMMON/BLK1/NO,NOS,SYMI,SYMBOL,NOVAL(9),GRID
      COMMON/BLK2/WEIGHT,NWGT(9),MOLWT(100),IE(100,5,6),NC(100),KCC
      COMMON/BLK3/IRING(40,30),IMAT3(40,40),NH(100),IBC(100),NBN(100),
     1IBR(100),IB(100,6),IB8,NBRA
      COMMON/BLK5/NDATA,NUMAT(9),NDC(50),NDS(2),JW,JY,LFLAG,LFLAGG
      EQUIVALENCE (INTMIN,GRID(4,70)),(KINT(1),GRID(5,70)),(LINE(1),
     1GRID(6,70)),(ASYM(1),GRID(7,70)),(KCSAME(1,1),GRID(9,40)),
     2(NOSAME(1,1),GRID(9,59)),(NTOTAL(1,1),GRID(4,70)),(KC1ALT(1),
     3GRID(9,70)),(KC2ALT(1),GRID(10,70)),(NO1ALT(1),GRID(11,70)),
     4(NO2ALT(1),GRID(20,70)),(NT1ALT(1),GRID(37,70)),(NT2ALT(1),
     5GRID(40,70)),(NT1S(1),GRID(43,70)),(NT2S(1),GRID(46,70)),(NS(1),
     6IG(2,5)),(NNT(1),IB(6,1)),(JBR(1),IMAT3(2,45))
C1.0      INITIALIZE VARIABLES.
      LFLAG=0
      NOSMC=1
      NOMESO=0
      NPSUDO=0
      INTMIN=0
      MIDE=NDS(1)/2
      MIDEP=MIDE+1
      IF(MIDE*2.EQ.NDS(1))GO TO 65
C2.0      THE MOLECULE HAS AN ODD NUMBER OF ATOMS IN ITS LONGEST CHAIN.
      KC1=NDC(MIDEP)
      NCP=NC(KC1)+1
      NONF=NBN(KC1)+1
      IF(IBC(KC1).EQ.0)GO TO 5
C            THE CENTRAL ATOM KC1 IS A RING ATOM. TRANSFER TO RING
C            SYMMETRY TEST ROUTINE.
      CALL SYMRNG(KC1,NOSMC)
      RETURN
C            THE CENTRAL ATOM KC1 IS NOT A RING ATOM.
    5 IF(IE(KC1,1,1).NE.2)GO TO 69
C            KC1 IS A CARBON ATOM.
      NB=NBN(KC1)-1
      GO TO(7,25,35),NB
C              KC1 HAS A CONNECTIVITY OF TWO - KEY. DETERMINE LINEARITY
C              OF LIGAND X.
    7 KCHEST=IE(KC1,2,6)
      CALL LINEAR(KC1,KCHEST,LINE(1),ASYM(1))
C                 ARE LIGANDS X AND Y IDENTICAL
      IF(NTOTAL(1,KC1).NE.2)GO TO 13
C                     YES THEY ARE. SO THE TWO LIGANDS HAVE TWOFOLD
C                     ROTATIONAL SYMMETRY
    8 IF(LINE(1).EQ.1.OR.ASYM(1).NE.2)GO TO 11
C                       YES.
    9 NOSMC=4
      GO TO 211
C                       NO.
   11 NOSMC=2
      GO TO 211
C                     LIGANDS ARE NOT IDENTICAL. DETERMINE LINEARITY OF
C                     LIGAND Y.
   13 KCHEST=IE(KC1,3,6)
      CALL LINEAR(KC1,KCHEST,LINE(2),ASYM(2))
   17 IF(LINE(1)+LINE(2)-1)19,370,211
C                     EITHER LIGAND X OR LIGAND Y IS LINEAR, NOT BOTH. SET
C                     SYMMETRY INDICATOR. DELETE INTERNAL ROTATION
C                     CONTRIBUTION OF KC1 IF SYMMETRY IS THREEFOLD.
  370 NOSMC=LINE(1)*ASYM(2)+LINE(2)*ASYM(1)
      IF(ASYM(1).EQ.3)GO TO 18
      IF(ASYM(2).NE.3)GO TO 211
      INTMIN=INTMIN+1
      KINT(INTMIN)=IE(KC1,3,6)
```

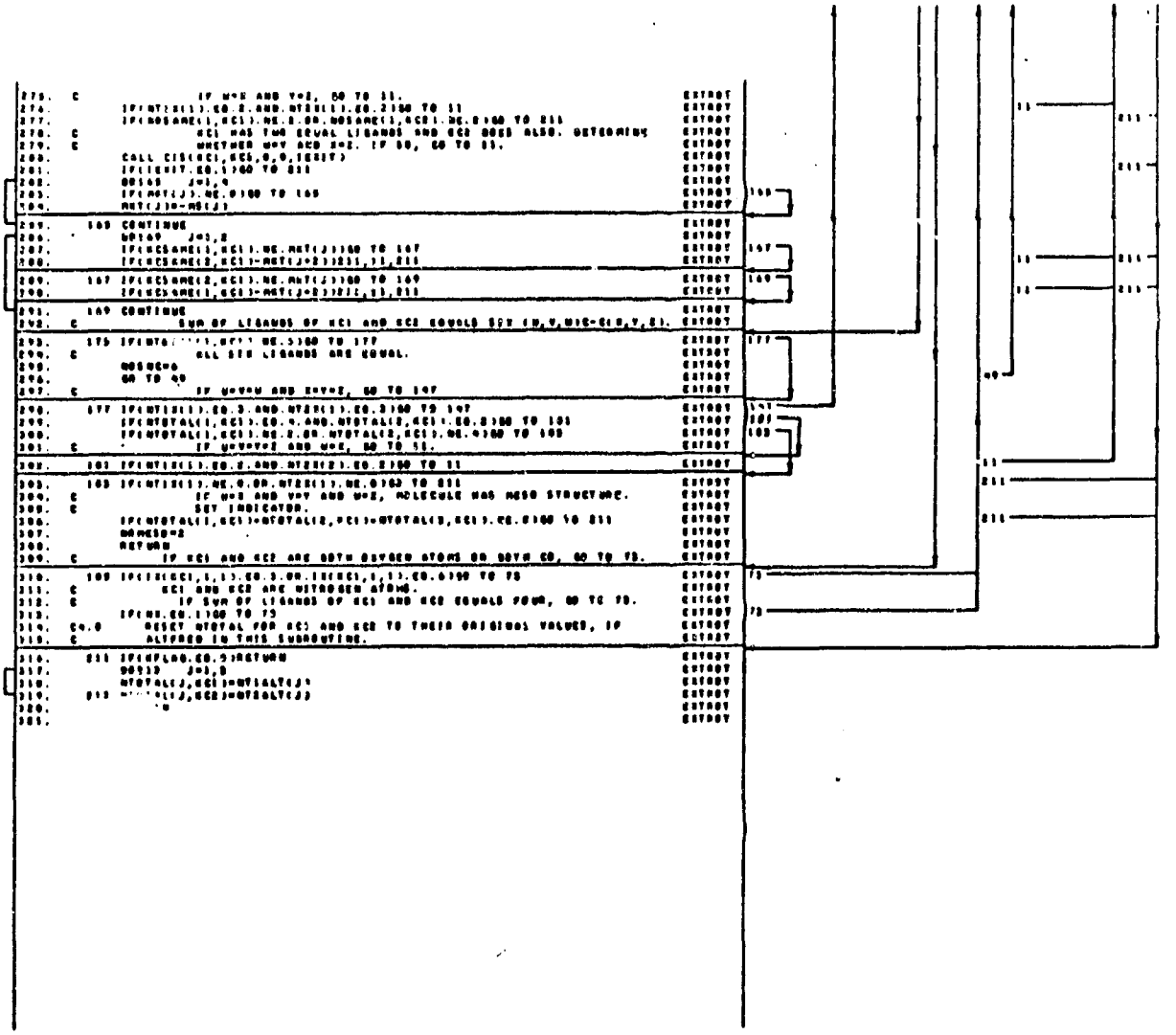
19 FEB 75 8.02-58

```
      GO TO 211                                                         EXTROT
   10 INTMIN=INTMIN+1                                                   EXTROT
      RINT(INTMIN)=FI(KC2,2,6)                                          EXTROT
      GO TO 211                                                         EXTROT
C                 NEITHER X NOR Y ARE LINEAR. ARE THEY OF SAME SYMMETRY EXTROT
   19 IF(ASYM(1).NE.ASYM(2).OR.ASYM(1).LT.2)RETURN                      EXTROT
C                 YES.                                                  EXTROT
      NOTWC=ASYM(1)                                                     EXTROT
      GO TO 211                                                         EXTROT
C              KCI HAS CONNECTIVITY OF THREE. IF TWO OF THE LIGANDS ARE EXTROT
C              NOT IDENTICAL, EXIT FROM ROUTINE.                        EXTROT
   25 IF(NTOTAL(1,KCI).NE.2)RETURN                                      EXTROT
C                 FIND LIGAND Z WHICH IS DOUBLE BONDED TO KCI.          EXTROT
      DO29   K=2,NCF                                                    EXTROT
      IF(FI(KCI,K,5).NE.2)GO TO 29                                      EXTROT
      KCHENT=FI(KCI,K,6)                                                EXTROT
      CALL LINEAR(KCI,KCHENT,LINE(1),ASYM(1))                           EXTROT
C                 IF Z IS LINEAR OR HAS TWOFOLD SYMMETRY, GO TO 11.     EXTROT
      IF(LINE(1).EQ.1.OR.ASYM(1).EQ.2)GO TO 11                          EXTROT
      RETURN                                                            EXTROT
   29 CONTINUE                                                          EXTROT
      J=9                                                               EXTROT
      RETURN                                                            EXTROT
C              KCI HAS CONNECTIVITY OF FOUR - C(W,X,Y,Z). IF W=X AND    EXTROT
C              Y=Z, GO TO 11.                                           EXTROT
   35 IF(NTOTAL(1,KCI).EQ.2.AND.NTOTAL(2,KCI).EQ.2)GO TO 11             EXTROT
      IF(NTOTAL(1,KCI)-3)37,43,65                                       EXTROT
   37 IF(NASYMC.LT.2.OR.NOSAME(1,KCI).NE.2)GO TO 211                    EXTROT
C                 LIGANDS W=X, Y NOT EQUAL TO Z, AND 2 OR MORE ASYMMETRIC EXTROT
C                 ATOMS ARE PRESENT. IS KCI PSEUDOASYMMETRIC            EXTROT
      KC2=MBC(MID2)                                                     EXTROT
      KC3=MBC(MID2P+1)                                                  EXTROT
      IF(FI(KC2,1,1).NE.2.OR.CON(KC2).NE.4.OR.NOSAME(1,KC2).NE.0)GO     EXTROT
     1TO 211                                                            EXTROT
      IF(FI(KC3,1,1).NE.2.OR.CON(KC3).NE.4.OR.NOSAME(1,KC3).NE.0)GO     EXTROT
     1TO 211                                                            EXTROT
C                 YES IT IS. SET INDICATORS.                            EXTROT
      NOMESO=1                                                          EXTROT
      NPSUDO=1                                                          EXTROT
      KCSUDO=KCI                                                        EXTROT
      RETURN                                                            EXTROT
C              LIGANDS W=X=Y. DOES THE CONNECTIVITY EXCEED THE NUMBER   EXTROT
C              OF CORE LIGANDS                                          EXTROT
   43 NOSWC=3                                                           EXTROT
      IF(CON(KCI).GT.NC(KCI))GO TO 49                                   EXTROT
C                 NO. FIND THE LOCATION OF THE DISSIMILAR LIGAND Z IN   EXTROT
C                 THE FI ARRAY AS WELL AS LINEARITY OF Z. IF LINEAR, GO EXTROT
C                 TO 49.                                                EXTROT
      DO44   K=2,NCNF                                                   EXTROT
      DO430  L=1,3                                                      EXTROT
      IF(FI(KCI,K,6).EQ.KCSAME(L,KCI))GO TO 44                          EXTROT
  430 CONTINUE                                                          EXTROT
      MO=FI(KCI,K,6)                                                    EXTROT
      CALL LINEAR(KCI,FI(KCI,MO,6),LINE(1),ASYM(1))                     EXTROT
      IF(LINE(1)-1)211,49,211                                           EXTROT
   44 CONTINUE                                                          EXTROT
C                 YES. TRANSFER TO 49 UNLESS LIGAND EQUALS NO OR NO2.   EXTROT
   45 IF(FI(KCI,KCNF,1).GE.8)GO TO 211                                  EXTROT
C                 DELETE INTERNAL ROTATION CONTRIBUTION OF KCI.         EXTROT
   49 INTMIN=INTMIN+1                                                   EXTROT
      RINT(INTMIN)=KCI                                                  EXTROT
      GO TO 211                                                         EXTROT
C              LIGANDS W=X=Y=Z. FIND SYMMETRY OF W.                     EXTROT
   65 CALL LINEAR(KCI,FI(KCI,2,6),LINE(1),ASYM(1))                      EXTROT
      IF(ASYM(1).NE.3)GO TO 67                                          EXTROT
C                 W HAS THREEFOLD SYMMETRY.                             EXTROT
      NOSWC=10                                                          EXTROT
      RETURN                                                            EXTROT
   67 IF(LINE(1).EQ.1)GO TO 68                                          EXTROT
```

1Y FEB 73 8 02-38



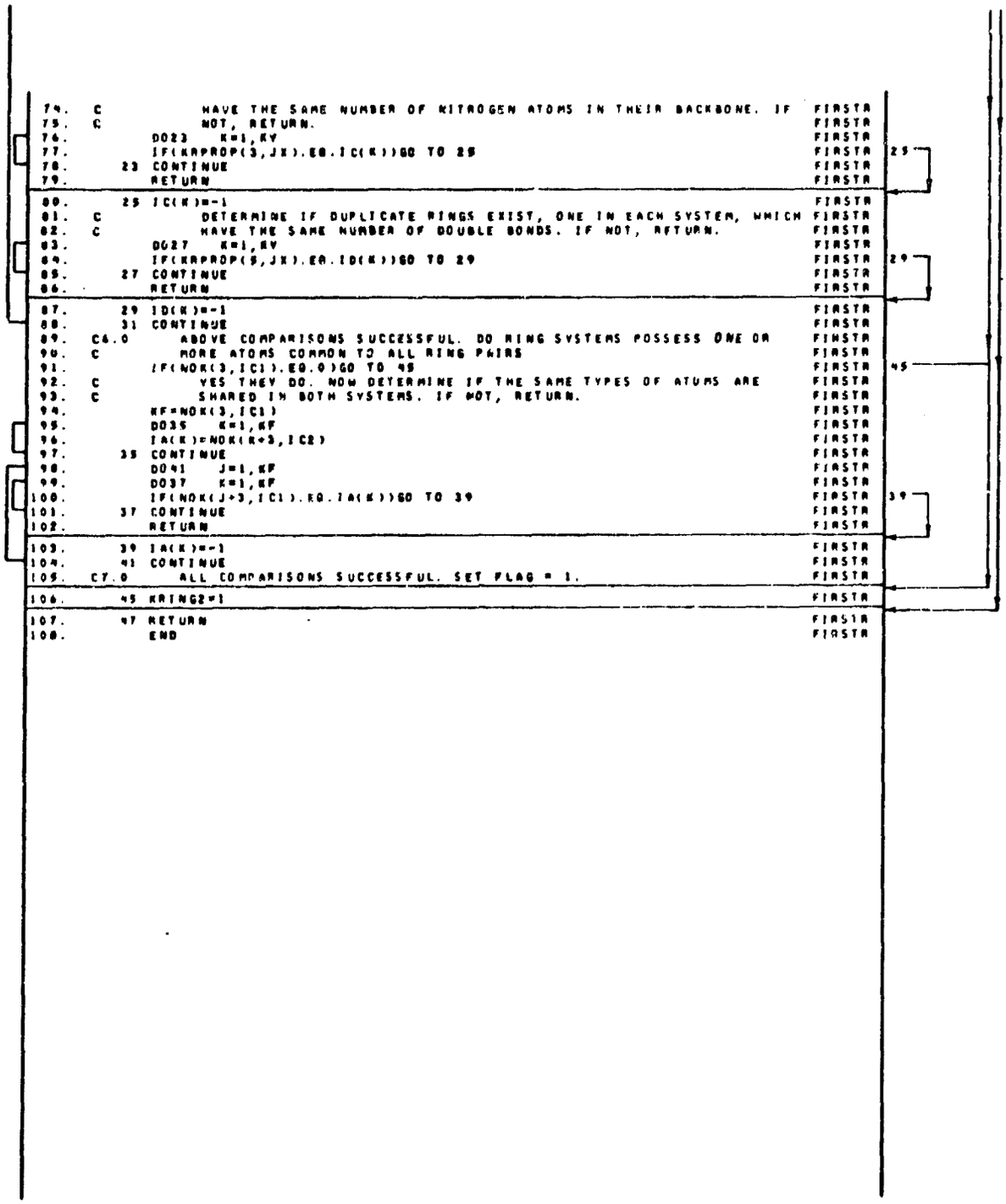




```
      SUBROUTINE FIND(KC,X,L,M,NUM,MR,IERR)                              FIND
C        THIS SUBROUTINE DETERMINES WHETHER THE FIRST ATOM TO BE         FIND
C        INDEXED IS A CORE ATOM OR AN UNIVALENT ATOM. IF THE LATTER, IT  FIND
C        DETERMINES THE LOCATION OF ONE OF THE CORE ATOMS BONDED TO      FIND
C        THIS NON-CORE AND FIRST INDEXED ATOM. THE DATA FOR THIS NON-    FIND
C        CORE ATOM ARE STORED IN ARRAY X.                                FIND
      INTEGER SYMX(9),SYMBOL(9),GRID(90,80)                              FIND
      INTEGER WEIGHT(9)                                                  FIND
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                       FIND
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC      FIND
C1.0     FIND NUMBER OF LIGANDS(MP) BONDED TO INDEXED ATOM. HOW MANY     FIND
C        ARE THERE                                                       FIND
      CALL NUMBER(NUM,NX,K,L,MP,KX,LX,KD,LD,JI,JF)                       FIND
      IF(MP-1)3,7,9                                                      FIND
C           ZERO LIGANDS - PRINT ERROR MESSAGE, SET ERROR FLAG, AND EXIT. FIND
    3 WRITE(6,5)K,L                                                      FIND
    5 FORMAT(// 1HO, 39H,24HERROR - ISOLATED ATOM AT,I4,1H,,I4,19H.  CAS FIND
     1E TERMINATED.)                                                     FIND
      IERR=1                                                             FIND
      RETURN                                                             FIND
C           ONE LIGAND - INDEXED ATOM IS NOT CORE ATOM. FIND CORE ATOM.  FIND
C             STORE DATA OF INDEXED ATOM IN IX ARRAY IN LOCATION         FIND
C             ASSIGNED TO A LIGAND.                                      FIND
    7 IX(KC,2,1)=M                                                       FIND
      IX(KC,2,2)=K                                                       FIND
      IX(KC,2,3)=L                                                       FIND
C              FIND BOND TYPE AND LOCATION OF SYMBOL OF CORE LIGAND      FIND
C              BONDED TO INDEXED ATOM.                                   FIND
      CALL BOND(KX,LX,KD,LD,JI,JF,IX(KC,2,5),IERR)                       FINC
      IF(IERR.EQ.1)RETURN                                                FIND
C              IDENTIFY SYMBOL.                                          FIND
      CALL IDENT(NX,KX,LX,M,IERR)                                        FIND
      IF(IERR.EQ.1)RETURN                                                FIND
C              COMPUTE BOND DIRECTION FROM CORE ATOM TO INDEXED ATOM AND FIND
C              STORE.                                                    FIND
      NZ=NX+4                                                            FIND
      IF(NZ.GT.8)NZ=NZ-8                                                 FIND
      IX(KC,2,4)=NZ                                                      FIND
C              SET COORDINATES OF CORE ATOM, MR, AND NUM.                FIND
      K=KX                                                               FIND
      L=LX                                                               FIND
      MR=3                                                               FIND
      GO TO 11                                                           FIND
C           TWO LIGANDS - INDEXED ATOM IS CORE ATOM. SET MR AND NUM.     FIND
    9 MR=2                                                               FIND
   11 NUM=NX                                                             FIND
      RETURN                                                             FIND
      END                                                                FIND
```

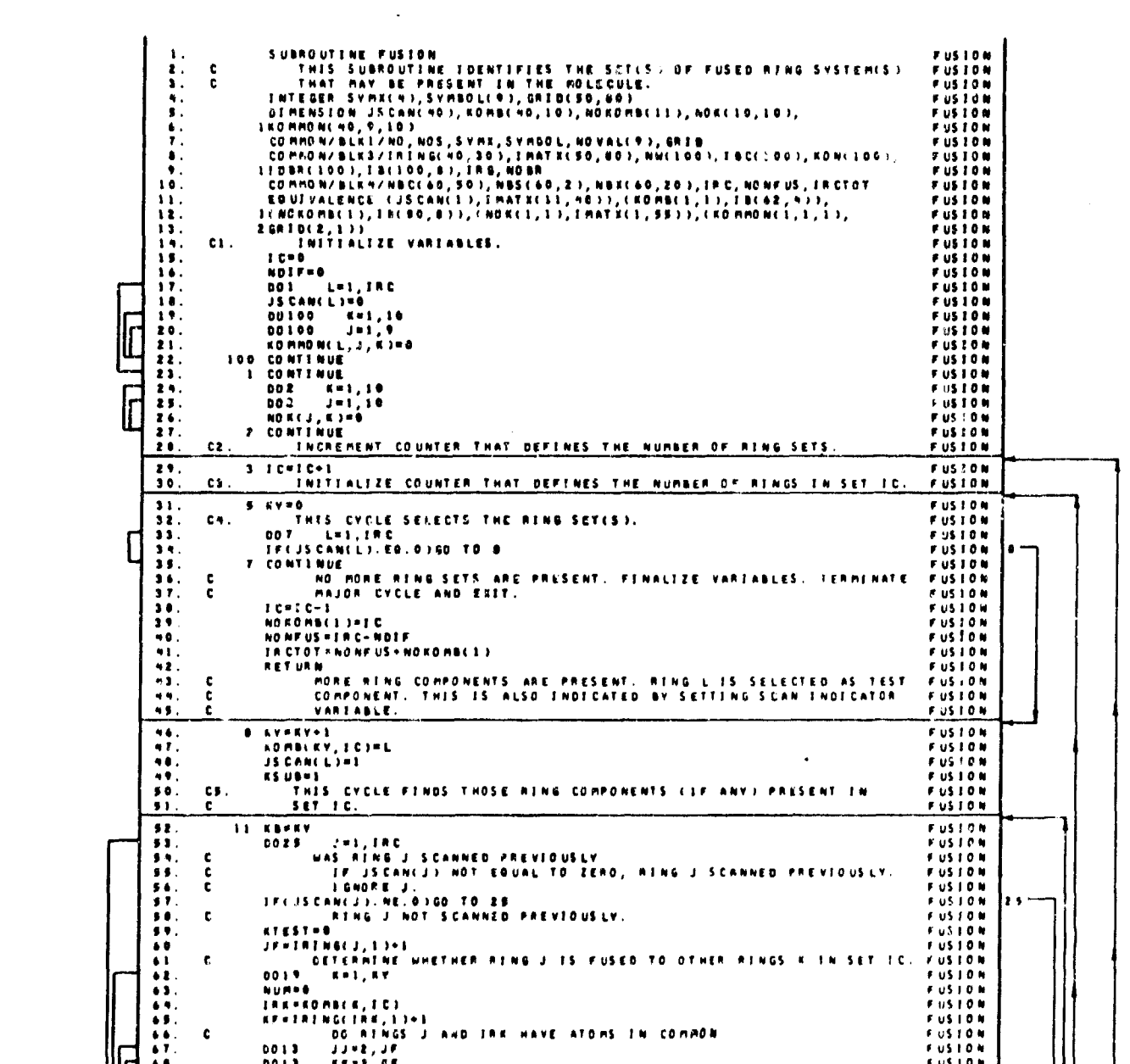
```
      SUBROUTINE FIRSTR(KC,K1,K2,KRING2)                                FIRSTR
C        THIS SUBROUTINE DETERMINES IF ATOMS K1 AND K2 FORM PART OF TWO  FIRSTR
C        DIFFERENT FUSED RING SYSTEMS. IF SO, IT COMPARES SEVERAL OF THE FIRSTR
C        COMPOSITION AND STRUCTURAL PROPERTIES OF THE TWO FUSED RING     FIRSTR
C        SYSTEMS TO DETERMINE WHETHER OR NOT THEY ARE SIMILAR. IF THE    FIRSTR
C        TWO RINGS ARE NOT FUSED, IT COMPARES THEIR RING SIZE.           FIRSTR
      INTEGER SYMX(4),SYMBOL(9),GRID(80,80)                             FIRSTR
      DIMENSION NOKOMB(11),KOMB(40,10),NOK(10,10),KRPROP(6,40),IA(40),  FIRSTR
     1IE(40),IC(40),ID(40)                                              FIRSTR
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                      FIRSTR
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),  FIRSTR
     1IDBR(100),IB(100,8),IRG,NOBR                                      FIRSTR
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT     FIRSTR
      COMMON/BLK5/NDATM,NUMATM(5),MBC(50),MBS(2),JW,JV,LFLAGS,LFLAGG    FIRSTR
      EQUIVALENCE (NOKOMB(1),IB(80,8)),(KOMB(1,1),IB(42,4)),(NOK(1,1),  FIRSTR
     1IMATX(1,55)),(KRPROP(1,1),NBX(2,1)),(IA(1),GRID(1,27)),(IE(1),    FIRSTR
     2IB(22,3)),(IC(1),IMATX(2,53)),(ID(1),IMATX(42,53))                FIRSTR
C1.0     INITIALIZE VARIABLE KRING2.                                    FIRSTR
      KRING2=0                                                          FIRSTR
C2.0     ARE ATOMS K1 AND K2 IN SAME RING SYSTEM                        FIRSTR
      IF(IBC(K1).NE.IBC(K2))GO TO 5                                     FIRSTR
C           YES. SET FLAGS AND EXIT FROM ROUTINE.                       FIRSTR
      WRITE(6,3)K1,K2,KC                                                FIRSTR
    3 FORMAT(// 1H0,23X,9HATOMS ,I4,6H   AND,I4,40H   ARE IN SAME RING SY FIRSTR
     1STEM. ASYMMETRY OV,I4,20H   NOT DETERMINABLE.)                    FIRSTR
      JW=0                                                              FIRSTR
      JV=0                                                              FIRSTR
      RETURN                                                            FIRSTR
C           NO. CONTINUE WITH TEST.                                     FIRSTR
    5 IF((IBC(K1)+IBC(K2))/100-1)7,47,9                                 FIRSTR
C3.0     K1 AND K2 ARE NON-FUSED RINGS. IF THEIR RING SIZE IS EQUAL,    FIRSTR
C           GO TO 45. OTHERWISE, RETURN.                                FIRSTR
    7 IC1=IBC(K1)                                                       FIRSTR
      IC2=IBC(K2)                                                       FIRSTR
      IF(IRING(IC1,1).EQ.IRING(IC2,1))GO TO 45                          FIRSTR
      RETURN                                                            FIRSTR
C4.0     K1 AND K2 BELONG TO SEPARATE FUSED RING SYSTEMS.               FIRSTR
    9 IC1=IBC(K1)-100                                                   FIRSTR
      IC2=IBC(K2)-100                                                   FIRSTR
      IF(NOKOMB(IC1+1).NE.NOKOMB(IC2+1))RETURN                          FIRSTR
C5.0     SAME NUMBER OF FUSED RINGS IN BOTH SYSTEMS.                    FIRSTR
      IF(NOK(1,IC1).NE.NOK(1,IC2).OR.NOK(2,IC1).NE.NOK(2,IC2).OR.       FIRSTR
     1NOK(3,IC1).NE.NOK(3,IC2))RETURN                                   FIRSTR
C6.0     BOTH SYSTEMS HAVE THE SAME NUMBER OF RING PAIRS, SAME NUMBER OF FIRSTR
C           ATOMS COMMON TO ANY FUSED RING PAIR, AND SAME NUMBER OF ATOMS FIRSTR
C           COMMON TO ALL RING PAIRS.                                   FIRSTR
      KY=NOKOMB(IC1+1)                                                  FIRSTR
C7.0     TEST THE EQUALITY OF THE TWO FUSED RING SYSTEMS FURTHER.       FIRSTR
      DO11   K=1,KY                                                     FIRSTR
      JX=KOMB(K,IC2)                                                    FIRSTR
      IA(K)=IRING(JX,1)                                                 FIRSTR
      IE(K)=KRPROP(1,JX)                                                FIRSTR
      IC(K)=KRPROP(3,JX)                                                FIRSTR
      ID(K)=KRPROP(5,JX)                                                FIRSTR
   11 CONTINUE                                                          FIRSTR
      DO33   J=1,KY                                                     FIRSTR
      JX=KOMB(J,IC1)                                                    FIRSTR
C           DETERMINE IF RINGS EXIST, ONE ON EACH SYSTEM, WHICH HAVE THE FIRSTR
C           SAME NUMBER OF COMPONENTS. IF NOT, RETURN.                  FIRSTR
      DO19   K=1,KY                                                     FIRSTR
      IF(IRING(JX,1).EQ.IA(K))GO TO 17                                  FIRSTR
   19 CONTINUE                                                          FIRSTR
      RETURN                                                            FIRSTR
   17 IA(K)=-1                                                          FIRSTR
C           DETERMINE IF DUPLICATE RINGS EXIST, ONE IN EACH SYSTEM, WHICH FIRSTR
C           HAVE THE SAME NUMBER OF OXYGEN ATOMS IN THEIR BACKBONE. IF  FIRSTR
C           NOT, RETURN.                                                FIRSTR
      DO19   K=1,KY                                                     FIRSTR
      IF(KRPROP(1,JX).EQ.IE(K))GO TO 21                                 FIRSTR
   19 CONTINUE                                                          FIRSTR
      RETURN                                                            FIRSTR
   21 IE(K)=-1                                                          FIRSTR
C           DETERMINE IF DUPLICATE RINGS EXIST, ONE IN EACH SYSTEM, WHICH FIRSTR
```

19 FEB 73 6:08:38



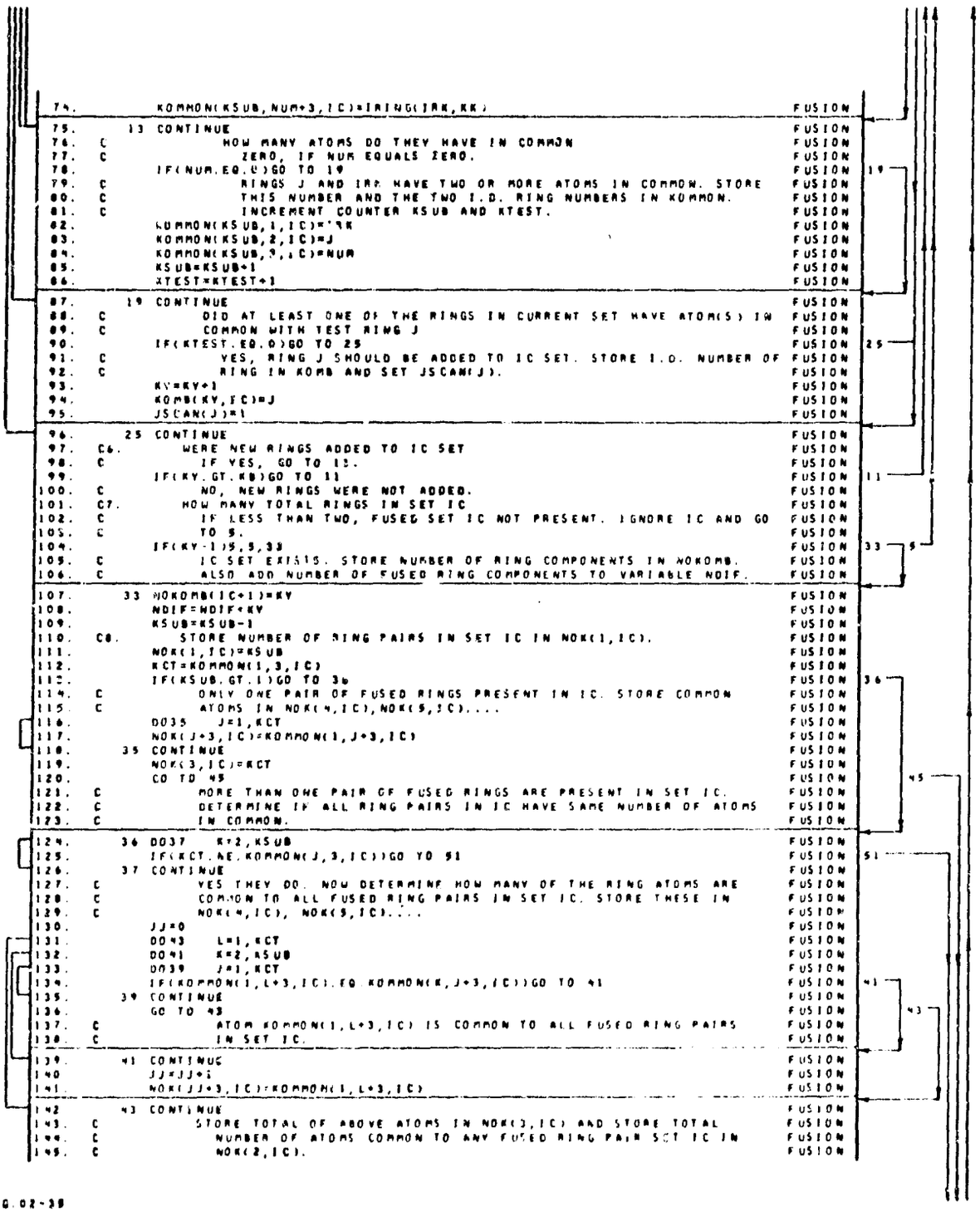
```
74.   C           HAVE THE SAME NUMBER OF NITROGEN ATOMS IN THEIR BACKBONE. IF   FIRSTR
75.   C           NOT, RETURN.                                                   FIRSTR
76.         DO23    K=1,KV                                                       FIRSTR
77.         IF(KRPROP(3,JX).EQ.IC(K))GO TO 25                                    FIRSTR  25
78.      23 CONTINUE                                                             FIRSTR
79.         RETURN                                                               FIRSTR
80.      25 IC(K)=-1                                                             FIRSTR
81.   C           DETERMINE IF DUPLICATE RINGS EXIST, ONE IN EACH SYSTEM, WHICH  FIRSTR
82.   C           HAVE THE SAME NUMBER OF DOUBLE BONDS. IF NOT, RETURN.          FIRSTR
83.         DO27    K=1,KV                                                       FIRSTR
84.         IF(KRPROP(5,JX).EQ.ID(K))GO TO 29                                    FIRSTR  29
85.      27 CONTINUE                                                             FIRSTR
86.         RETURN                                                               FIRSTR
87.      29 ID(K)=-1                                                             FIRSTR
88.         CONTINUE                                                             FIRSTR
89.   C4.0      ABOVE COMPARISONS SUCCESSFUL. DO RING SYSTEMS POSSESS ONE OR    FIRSTR
90.   C         MORE ATOMS COMMON TO ALL RING PAIRS                              FIRSTR
91.         IF(NOK(3,IC1).EQ.0)GO TO 45                                          FIRSTR  45
92.   C           YES THEY DO. NOW DETERMINE IF THE SAME TYPES OF ATOMS ARE      FIRSTR
93.   C           SHARED IN BOTH SYSTEMS. IF NOT, RETURN.                        FIRSTR
94.         KF=NOK(3,IC1)                                                        FIRSTR
95.         DO35    K=1,KF                                                       FIRSTR
96.         IA(K)=NOK(K+3,IC2)                                                   FIRSTR
97.      35 CONTINUE                                                             FIRSTR
98.         DO41    J=1,KF                                                       FIRSTR
99.         DO37    K=1,KF                                                       FIRSTR
100.        IF(NOK(J+3,IC1).EQ.IA(K))GO TO 39                                    FIRSTR  39
101.     37 CONTINUE                                                             FIRSTR
102.        RETURN                                                               FIRSTR
103.     39 IA(K)=-1                                                             FIRSTR
104.     41 CONTINUE                                                             FIRSTR
105.  C7.0      ALL COMPARISONS SUCCESSFUL. SET FLAG = 1.                        FIRSTR
106.     45 KRING2=1                                                             FIRSTR
107.     47 RETURN                                                               FIRSTR
108.        END                                                                  FIRSTR
```

19 FEB 73 6 02-38



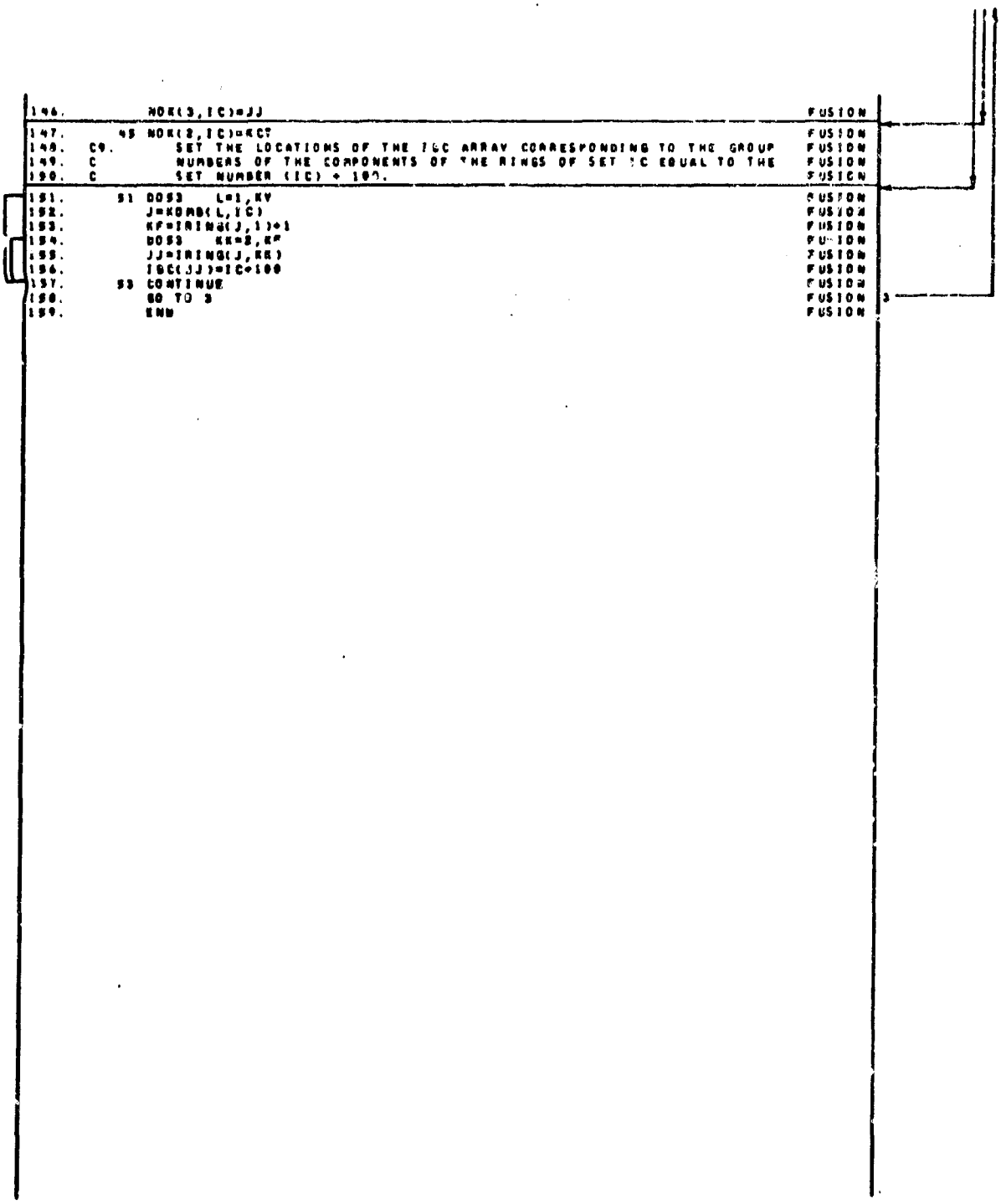
```
      SUBROUTINE FUSION                                                 FUSION
C         THIS SUBROUTINE IDENTIFIES THE SET(S) OF FUSED RING SYSTEM(S)  FUSION
C         THAT MAY BE PRESENT IN THE MOLECULE.                          FUSION
      INTEGER SYMX(4),SYMBOL(9),GRID(50,60)                             FUSION
      DIMENSION JSCAN(40),KOMB(40,10),NOKOMB(11),NOK(10,10),            FUSION
     1KOMMON(40,9,10)                                                   FUSION
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                      FUSION
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),  FUSION
     1IDBR(100),IB(100,8),IRS,NOBR                                      FUSION
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT     FUSION
      EQUIVALENCE (JSCAN(1),IMATX(1,40)),(KOMB(1,1),IB(62,4)),          FUSION
     1(NOKOMB(1),IB(90,8)),(NOK(1,1),IMATX(1,55)),(KOMMON(1,1,1),       FUSION
     2GRID(2,1))                                                        FUSION
C1.       INITIALIZE VARIABLES.                                         FUSION
      IC=0                                                              FUSION
      NDIF=0                                                            FUSION
      DO1    L=1,IRC                                                    FUSION
      JSCAN(L)=0                                                        FUSION
      DO100    K=1,10                                                   FUSION
      DO100    J=1,9                                                    FUSION
      KOMMON(L,J,K)=0                                                   FUSION
  100 CONTINUE                                                          FUSION
    1 CONTINUE                                                          FUSION
      DO2    K=1,10                                                     FUSION
      DO2    J=1,10                                                     FUSION
      NOK(J,K)=0                                                        FUSION
    2 CONTINUE                                                          FUSION
C2.       INCREMENT COUNTER THAT DEFINES THE NUMBER OF RING SETS.       FUSION
    3 IC=IC+1                                                           FUSION
C3.       INITIALIZE COUNTER THAT DEFINES THE NUMBER OF RINGS IN SET IC. FUSION
    5 KV=0                                                              FUSION
C4.       THIS CYCLE SELECTS THE RING SET(S).                           FUSION
      DO7    L=1,IRC                                                    FUSION
      IF(JSCAN(L).EQ.0)GO TO 8                                          FUSION  8
    7 CONTINUE                                                          FUSION
C           NO MORE RING SETS ARE PRESENT. FINALIZE VARIABLES. TERMINATE FUSION
C           MAJOR CYCLE AND EXIT.                                       FUSION
      IC=IC-1                                                           FUSION
      NOKOMB(1)=IC                                                      FUSION
      NONFUS=IRC-NDIF                                                   FUSION
      IRCTOT=NONFUS+NOKOMB(1)                                           FUSION
      RETURN                                                            FUSION
C           MORE RING COMPONENTS ARE PRESENT. RING L IS SELECTED AS TEST FUSION
C           COMPONENT. THIS IS ALSO INDICATED BY SETTING SCAN INDICATOR FUSION
C           VARIABLE.                                                   FUSION
    8 KV=KV+1                                                           FUSION
      KOMB(KV,IC)=L                                                     FUSION
      JSCAN(L)=1                                                        FUSION
      KSUB=1                                                            FUSION
C9.       THIS CYCLE FINDS THOSE RING COMPONENTS (IF ANY) PRESENT IN    FUSION
C           SET IC.                                                     FUSION
   11 KB=KV                                                             FUSION
      DO29    J=1,IRC                                                   FUSION
C             WAS RING J SCANNED PREVIOUSLY                             FUSION
C               IF JSCAN(J) NOT EQUAL TO ZERO, RING J SCANNED PREVIOUSLY. FUSION
C               IGNORE J.                                               FUSION
      IF(JSCAN(J).NE.0)GO TO 29                                         FUSION  29
C               RING J NOT SCANNED PREVIOUSLY.                          FUSION
      KTEST=0                                                           FUSION
      JF=IRING(J,1)+1                                                   FUSION
C           DETERMINE WHETHER RING J IS FUSED TO OTHER RINGS K IN SET IC. FUSION
      DO19    K=1,KV                                                    FUSION
      NUM=0                                                             FUSION
      IRK=KOMB(K,IC)                                                    FUSION
      KF=IRING(IRK,1)+1                                                 FUSION
C               DO RINGS J AND IRK HAVE ATOMS IN COMMON                 FUSION
      DO13    JJ=2,JF                                                   FUSION
      DO13    KK=2,KF                                                   FUSION
C                 IF NOT, GO TO 13.                                     FUSION
      IF(IRING(IRK,KK).NE.IRING(J,JJ))GO TO 13                          FUSION  13
C                   YES - RINGS J AND IRK ARE FUSED. INCREMENT COUNTER AND FUSION
C                   STORE FUSED ATOMS IN KOMMON.                        FUSION
      NUM=NUM+1                                                         FUSION
```

19 FEB 73 8:02:39



```
74.            KOMMON(KSUB,NUM+3,IC)=IRING(IRK,KK)                    FUSION
75.         13 CONTINUE                                               FUSION
76.   C              HOW MANY ATOMS DO THEY HAVE IN COMMON             FUSION
77.   C                ZERO, IF NUM EQUALS ZERO.                       FUSION
78.            IF(NUM.EQ.0)GO TO 19                                   FUSION
79.   C                RINGS J AND IRK HAVE TWO OR MORE ATOMS IN COMMON. STORE  FUSION
80.   C                THIS NUMBER AND THE TWO I.D. RING NUMBERS IN KOMMON.     FUSION
81.   C                INCREMENT COUNTER KSUB AND KTEST.               FUSION
82.            KOMMON(KSUB,1,IC)=IRK                                  FUSION
83.            KOMMON(KSUB,2,IC)=J                                    FUSION
84.            KOMMON(KSUB,3,IC)=NUM                                  FUSION
85.            KSUB=KSUB+1                                            FUSION
86.            KTEST=KTEST+1                                          FUSION
87.         19 CONTINUE                                               FUSION
88.   C             DID AT LEAST ONE OF THE RINGS IN CURRENT SET HAVE ATOM(S) IN  FUSION
89.   C             COMMON WITH TEST RING J                           FUSION
90.            IF(KTEST.EQ.0)GO TO 25                                 FUSION
91.   C                YES, RING J SHOULD BE ADDED TO IC SET. STORE I.D. NUMBER OF  FUSION
92.   C                RING IN KOMB AND SET JSCAN(J).                  FUSION
93.            KY=KY+1                                                FUSION
94.            KOMB(KY,IC)=J                                          FUSION
95.            JSCAN(J)=1                                             FUSION
96.         25 CONTINUE                                               FUSION
97.   C6.          WERE NEW RINGS ADDED TO IC SET                     FUSION
98.   C               IF YES, GO TO 11.                               FUSION
99.            IF(KY.GT.KB)GO TO 11                                   FUSION
100.  C               NO, NEW RINGS WERE NOT ADDED.                   FUSION
101.  C7.          HOW MANY TOTAL RINGS IN SET IC                     FUSION
102.  C               IF LESS THAN TWO, FUSED SET IC NOT PRESENT. IGNORE IC AND GO  FUSION
103.  C               TO 5.                                           FUSION
104.           IF(KY-1)5,5,33                                         FUSION
105.  C               IC SET EXISTS. STORE NUMBER OF RING COMPONENTS IN NOKOMB.  FUSION
106.  C               ALSO ADD NUMBER OF FUSED RING COMPONENTS TO VARIABLE NDIF.  FUSION
107.        33 NOKOMB(IC+1)=KY                                        FUSION
108.           NDIF=NDIF+KY                                           FUSION
109.           KSUB=KSUB-1                                            FUSION
110.  C8.          STORE NUMBER OF RING PAIRS IN SET IC IN NOK(1,IC).  FUSION
111.           NOK(1,IC)=KSUB                                         FUSION
112.           KCT=KOMMON(1,3,IC)                                     FUSION
113.           IF(KSUB.GT.1)GO TO 36                                  FUSION
114.  C               ONLY ONE PAIR OF FUSED RINGS PRESENT IN IC. STORE COMMON  FUSION
115.  C               ATOMS IN NOK(4,IC),NOK(5,IC)....                FUSION
116.           DO35    J=1,KCT                                        FUSION
117.           NOK(J+3,IC)=KOMMON(1,J+3,IC)                           FUSION
118.        35 CONTINUE                                               FUSION
119.           NOK(3,IC)=KCT                                          FUSION
120.           GO TO 45                                               FUSION
121.  C               MORE THAN ONE PAIR OF FUSED RINGS ARE PRESENT IN SET IC.  FUSION
122.  C               DETERMINE IF ALL RING PAIRS IN IC HAVE SAME NUMBER OF ATOMS  FUSION
123.  C               IN COMMON.                                      FUSION
124.        36 DO37    K=2,KSUB                                       FUSION
125.           IF(KCT.NE.KOMMON(J,3,IC))GO TO 51                      FUSION
126.        37 CONTINUE                                               FUSION
127.  C               YES THEY DO. NOW DETERMINE HOW MANY OF THE RING ATOMS ARE  FUSION
128.  C               COMMON TO ALL FUSED RING PAIRS IN SET IC. STORE THESE IN  FUSION
129.  C               NOK(4,IC), NOK(5,IC)....                        FUSION
130.           JJ=0                                                   FUSION
131.           DO43    L=1,KCT                                        FUSION
132.           DO41    K=2,KSUB                                       FUSION
133.           DO39    J=1,KCT                                        FUSION
134.           IF(KOMMON(1,L+3,IC).EQ.KOMMON(K,J+3,IC))GO TO 41       FUSION
135.        39 CONTINUE                                               FUSION
136.           GO TO 43                                               FUSION
137.  C                ATOM KOMMON(1,L+3,IC) IS COMMON TO ALL FUSED RING PAIRS  FUSION
138.  C                IN SET IC.                                     FUSION
139.        41 CONTINUE                                               FUSION
140.           JJ=JJ+1                                                FUSION
141.           NOK(JJ+3,IC)=KOMMON(1,L+3,IC)                          FUSION
142.        43 CONTINUE                                               FUSION
143.  C             STORE TOTAL OF ABOVE ATOMS IN NOK(3,IC) AND STORE TOTAL  FUSION
144.  C               NUMBER OF ATOMS COMMON TO ANY FUSED RING PAIR SET IC IN  FUSION
145.  C               NOK(2,IC).                                      FUSION
```

,4 FEB 73 0.02-38



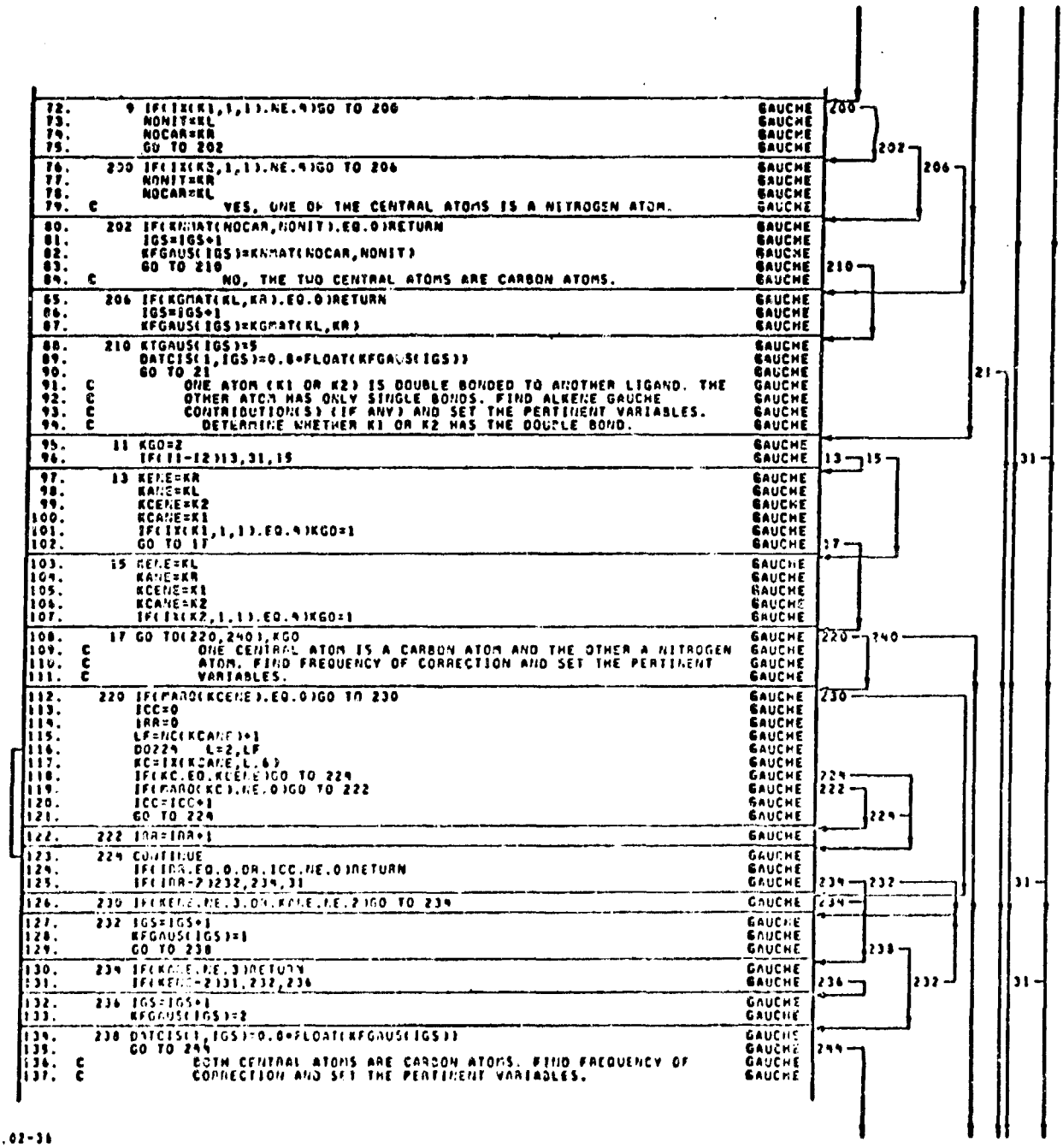
```
      NOK(3,IC)=JJ                                                      FUSION
   45 NOK(2,IC)=KCT                                                     FUSION
C9.   SET THE LOCATIONS OF THE IGC ARRAY CORRESPONDING TO THE GROUP     FUSION
C     NUMBERS OF THE COMPONENTS OF THE RINGS OF SET IC EQUAL TO THE     FUSION
C     SET NUMBER (IC) + 100.                                           FUSION
   51 DO53   L=1,KV                                                     FUSION
      J=KOMB(L,IC)                                                      FUSION
      KF=IRING(J,1)+1                                                   FUSION
      DO53   KK=2,KF                                                    FUSION
      JJ=IRING(J,KK)                                                    FUSION
      IGC(JJ)=IC+100                                                    FUSION
   53 CONTINUE                                                          FUSION
      GO TO 3                                                           FUSION
      END                                                               FUSION
```

19 FEB 73 8.02-39

```
      SUBROUTINE GADATA                                                 GADATA
C        THIS SUBROUTINE PRINTS OUT THE GROUP ADDITIVITY LIBRARY DATA   GADATA
C        IF CALLED FOR BY THE INPUT DATA.                               GADATA
      INTEGER GROUP1(100),GROUP2(100),GROUP3(100)                       GADATA
      INTEGER SUM(100)                                                  GADATA
      DIMENSION CP1(100),CP2(100),CP3(100),CP4(100)                     GADATA
      COMMON/BLK4/NBC(40,50),NBS(40,2),NBX(40,20),IAC,NONFUS,IRCTOT     GADATA
      COMMON/BLK7/SUM,HF298(100),S298(100),CPX(100,4)                   GADATA
      EQUIVALENCE (CPX(1,1),CP1(1)),(CPX(1,2),CP2(1)),(CPX(1,3),CP3(1)),GADATA
     1(CPX(1,4),CP4(1))                                                 GADATA
      EQUIVALENCE (GROUP1(1),NBC(2,1)),(GROUP2(1),NBC(2,12)),           GADATA
     1(GROUP3(1),NBC(2,23))                                             GADATA
C1.0     PRINT OUT HEADING AND UNITS OF OUTPUT.                         GADATA
      WRITE(6,998)                                                      GADATA
  998 FORMAT(1HI,35X,48HTABLES OF THERMOCHEMICAL GROUP CONTRIBUTIONS    GADATA
     1//)                                                               GADATA
      WRITE(6,996)                                                      GADAYA
  996 FORMAT(1H0, 4X,5HGROUP, 7X,6HWEIGHT, 2X,17HHEAT OF FORMATION, 2X, GADATA
     1 7HENTROPY, 20X,24HHEAT CAPACITY COEFFICIENTS/ 1H ,28X,9HKCAL/MOLE,GADATA
     2 4X,12HCAL/DEG-MOLE, 3X,12HCAL/DEG-MOLE, 2X,15HCAL/DEG**2-MOLE,  GADATA
     3 2X,15HCAL/DEG**3-MOLE, 2X,15HCAL/DEG**4-MOLE/1H ,30X,5H298 K,   GADATA
     49X,5H298 K)                                                       GADATA
C2.0     PRINT OUT TABLE OF HYDROCARBONS.                               GADATA
      WRITE(6,994)                                                      GADATA
  994 FORMAT(1H0,    49X,21HDATA FOR HYDROCARBONS//)                    GADATA
      WRITE(6,3) (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),        GADATA
     1S298(L),CP1(L),CP2(L),CP3(L),CP4(L),L=1,33)                       GADATA
    3 FORMAT(1H ,2A4,A5,I8,4X,F10.4,4X,F10.4,F17.10,F16.10,2F17.10)     GADATA
C3.0     PRINT OUT TABLE OF OXYGEN-CONTAINING GROUPS.                   GADATA
      WRITE(6,992)                                                      GADATA
  992 FORMAT(1H0,    43X,33HDATA FOR OXYGEN-CONTAINING GROUPS//)        GADATA
      WRITE(6,5) (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),        GADATA
     1S298(L),CP1(L),CP2(L),CP3(L),CP4(L),L=51,99)                      GADATA
    5 FORMAT(1H ,2A4,A5,I8,4X,F10.4,4X,F10.4,F17.10,F16.10,2F17.10)     GADATA
C4.0     PRINT OUT TABLE OF NITROGEN-CONTAINING GROUPS.                 GADATA
      WRITE(6,990)                                                      GADATA
  990 FORMAT(1H0, 42X, 35HDATA FOR NITROGEN-CONTAINING GROUPS//)        GADATA
      WRITE(6,7) (GROUP1(L),GROUP2(L),GROUP3(L),SUM(L),HF298(L),        GADATA
     1S298(L),CP1(L),CP2(L),CP3(L),CP4(L),L=110,163)                    GADATA
    7 FORMAT(1H ,2A4,A5,I8,4X,F10.4,4X,F10.4,F17.10,F16.10,2F17.10)     GADATA
      RETURN                                                            GADATA
      END                                                               GADATA
```

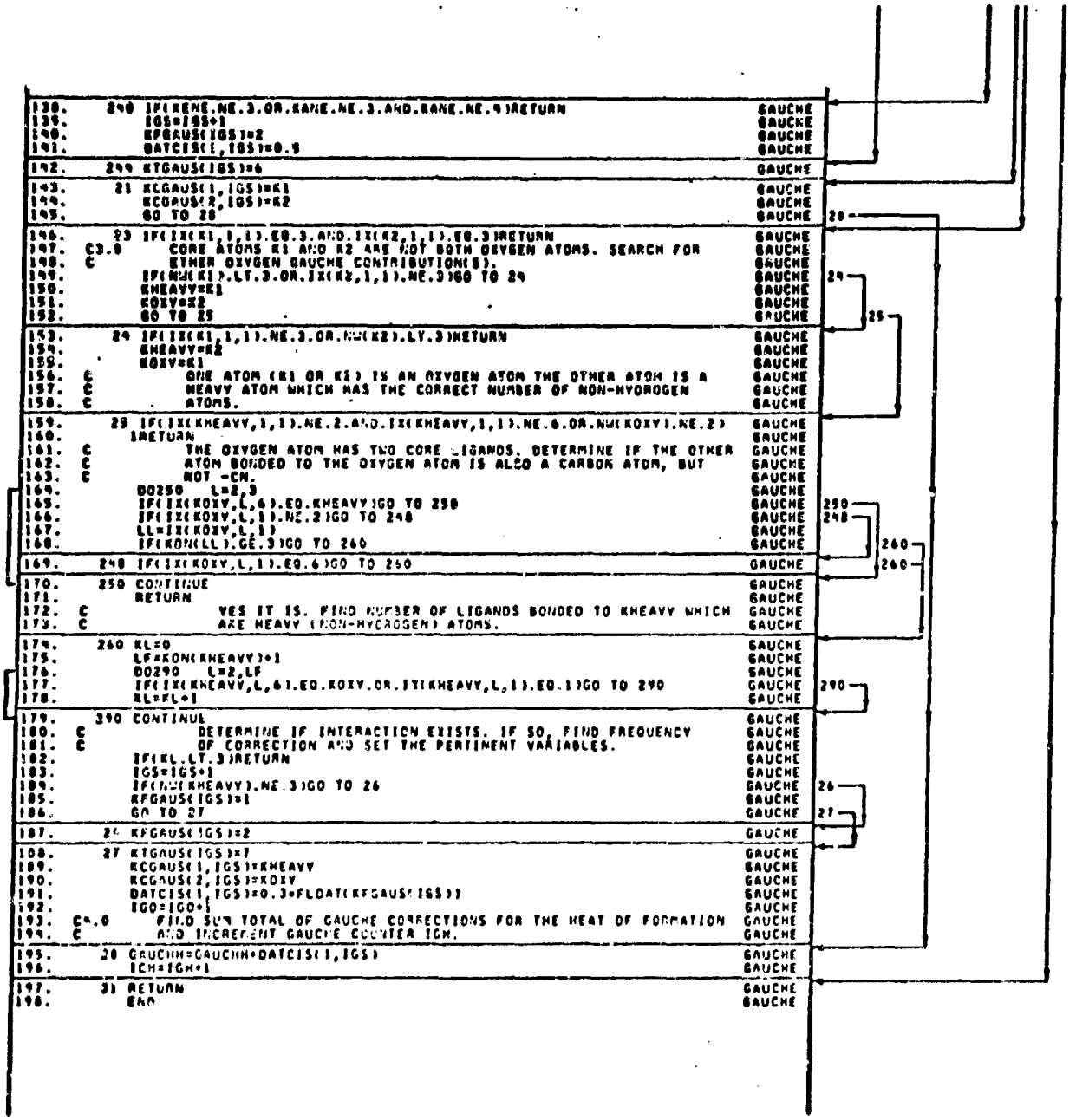
```
      SUBROUTINE GAUCHE(K1,K2,IBS,IGM,IGO,GAUCHN)                       GAUCHE
C        THIS SUBROUTINE DETERMINES WHETHER THE LIGANDS OF ATOMS K1 AND  GAUCHE
C        K2 ARE GAUCHE TO EACH OTHER. IF SO, IT DETERMINES THE NUMBER,   GAUCHE
C        TYPE, AND MAGNITUDE OF THE GAUCHE CORRECTION.                   GAUCHE
      INTEGER SYM1(9),SYMBOL(9),GRID(50,60)                             GAUCHE
      INTEGER WEIGHT(9)                                                 GAUCHE
      DIMENSION KCGAUS(3,150),KFGAUS(150),KTGAUS(150),DATCIS(2,150),    GAUCHE
     1KGMAT(9,9),KNMAT(9,9),MARO(100)                                   GAUCHE
      COMMON/BLK1/NO,NOS,SYM1,SYMBOL,NOVAL(9),GRID                      GAUCHE
      COMMON/BLK2/WEIGHT,MUST(9),MOLWT(100),IX(100,9,6),NC(100),KCC     GAUCHE
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NWI(100),IBO(100),KON(100), GAUCHE
     1IOCR(100),IB(100,8),IRG,NOBR                                      GAUCHE
      EQUIVALENCE (KCGAUS(1,1),IMATX(1,72)),(KFGAUS(1),IMATX(1,63)),    GAUCHE
     1(KTGAUS(1),IMATX(1,60)),(DATCIS(1,1),IMATX(1,66)),(MARO(1),       GAUCHE
     2GRID(1,29))                                                       GAUCHE
C        KGMAT DEFINES THE NUMBER OF ALKANE GAUCHE CORRECTIONS WHEN K1   GAUCHE
C        AND K2 ARE BOTH CARBON ATOMS.                                   GAUCHE
      DATA KGMAT(1,1)/0/,KGMAT(1,2)/0/,KGMAT(1,3)/0/,KGMAT(1,4)/0/,     GAUCHE
     1KGMAT(2,1)/0/,KGMAT(2,2)/0/,KGMAT(2,3)/1/,KGMAT(2,4)/2/,          GAUCHE
     2KGMAT(3,1)/0/,KGMAT(3,2)/1/,KGMAT(3,3)/2/,KGMAT(3,4)/3/,          GAUCHE
     3KGMAT(4,1)/0/,KGMAT(4,2)/2/,KGMAT(4,3)/3/,KGMAT(4,4)/6/          GAUCHE
C        KNMAT DEFINES THE NUMBER OF ALKANE GAUCHE CORRECTIONS WHEN K1   GAUCHE
C        OR K2 IS A NITROGEN ATOM.                                       GAUCHE
      DATA KNMAT(1,1)/0/,KNMAT(1,2)/0/,KNMAT(1,3)/0/,KNMAT(1,4)/0/,     GAUCHE
     1KNMAT(2,1)/0/,KNMAT(2,2)/1/,KNMAT(2,3)/1/,KNMAT(2,4)/2/,          GAUCHE
     2KNMAT(3,1)/0/,KNMAT(3,2)/1/,KNMAT(3,3)/2/,KNMAT(3,4)/3/,          GAUCHE
     3KNMAT(4,1)/0/,KNMAT(4,2)/2/,KNMAT(4,3)/3/,KNMAT(4,4)/6/          GAUCHE
C.1.0    IF BOTH K1 AND K2 ARE NITROGEN ATOMS, EXIT FROM ROUTINE.        GAUCHE
      IF(IX(K1,1,1).EQ.4.AND.IX(K2,1,1).EQ.4)RETURN                     GAUCHE
C        IF K1 OR K2 IS AN OXYGEN ATOM, GO TO 23.                        GAUCHE
      IF(IX(K1,1,1).EQ.3.OR.IX(K2,1,1).EQ.3)GO TO 23                    GAUCHE
      IF(KON(K1).LT.3.AND.IX(K1,1,1).NE.6)RETURN                        GAUCHE
      IF(KON(K2).LT.3.AND.IX(K2,1,1).NE.6)RETURN                        GAUCHE
C.2.0    CORE ATOMS K1 AND K2 ARE EITHER CARBON OR NITROGEN ATOMS.       GAUCHE
C          FIND NUMBER OF LIGANDS BONDED TO K1 AND K2 WHICH ARE HEAVY    GAUCHE
C          (NON-HYDROGEN) ATOMS.                                         GAUCHE
      KL=0                                                              GAUCHE
      KR=0                                                              GAUCHE
      LF=KON(K1)+1                                                      GAUCHE
      DO150   L=2,LF                                                    GAUCHE
      IF(IX(K1,L,6).EQ.K2.OR.IX(K1,L,1).EQ.1)GO TO 150                  GAUCHE
      KL=KL+1                                                           GAUCHE
  150 CONTINUE                                                          GAUCHE
      LF=KON(K2)+1                                                      GAUCHE
      DO180   L=2,LF                                                    GAUCHE
      IF(IX(K2,L,6).EQ.K1.OR.IX(K2,L,1).EQ.1)GO TO 180                  GAUCHE
      KR=KR+1                                                           GAUCHE
  180 CONTINUE                                                          GAUCHE
      KL=KL+1                                                           GAUCHE
      KR=KR+1                                                           GAUCHE
C          DETERMINE IF GAUCHE ALKANE OR GAUCHE ALKENE INTERACTIONS      GAUCHE
C          ARE PRESENT.                                                  GAUCHE
      I1=0                                                              GAUCHE
      I2=0                                                              GAUCHE
C          DETERMINE IF K1 OR K2 ARE JOINED BY A DOUBLE BOND TO OTHER    GAUCHE
C          LIGANDS. IF SO, SET THE CORRESPONDING FLAG(S) EQUAL TO 1.     GAUCHE
      DO12    L=2,5                                                     GAUCHE
      IF(IX(K1,L,5)-2)10,6,31                                           GAUCHE
    6 I1=1                                                              GAUCHE
    8 IF(IX(K2,L,5)-2)12,10,31                                          GAUCHE
   10 I2=1                                                              GAUCHE
   12 CONTINUE                                                          GAUCHE
      IF(IX(K1,1,1).NE.6)GO TO 190                                      GAUCHE
      KL=KL+1                                                           GAUCHE
      I1=1                                                              GAUCHE
  190 IF(IX(K2,1,1).NE.6)GO TO 192                                      GAUCHE
      KR=KR+1                                                           GAUCHE
      I2=1                                                              GAUCHE
  192 IF(I1+I2-1)9,11,31                                                GAUCHE
C          ALL BONDS OF ATOMS K1 AND K2 ARE SINGLE. FIND ALKANE GAUCHE   GAUCHE
C          CONTRIBUTION(S) (IF ANY) AND SET THE PERTINENT VARIABLES.     GAUCHE
C            IS K1 OR K2 A NITROGEN ATOM                                 GAUCHE
```

17 APR 73 0.02-36



```
      9 IF(IX(K1,1,1).NE.9)GO TO 206                                    GAUCHE
        NONIT=KL                                                        GAUCHE
        NOCAR=KR                                                        GAUCHE
        GO TO 202                                                       GAUCHE
  200 IF(IX(K2,1,1).NE.9)GO TO 206                                      GAUCHE
        NONIT=KR                                                        GAUCHE
        NOCAR=KL                                                        GAUCHE
C               YES, ONE OF THE CENTRAL ATOMS IS A NITROGEN ATOM.       GAUCHE
  202 IF(KNMAT(NOCAR,NONIT).EQ.0)RETURN                                 GAUCHE
        IGS=IGS+1                                                       GAUCHE
        KFGAUS(IGS)=KNMAT(NOCAR,NONIT)                                  GAUCHE
        GO TO 210                                                       GAUCHE
C               NO, THE TWO CENTRAL ATOMS ARE CARBON ATOMS.             GAUCHE
  206 IF(KCMAT(KL,KR).EQ.0)RETURN                                       GAUCHE
        IGS=IGS+1                                                       GAUCHE
        KFGAUS(IGS)=KCMAT(KL,KR)                                        GAUCHE
  210 KTGAUS(IGS)=5                                                     GAUCHE
        DATCIS(1,IGS)=0.0+FLOAT(KFGAUS(IGS))                            GAUCHE
        GO TO 21                                                        GAUCHE
C          ONE ATOM (K1 OR K2) IS DOUBLE BONDED TO ANOTHER LIGAND. THE  GAUCHE
C          OTHER ATOM HAS ONLY SINGLE BONDS. FIND ALKENE GAUCHE         GAUCHE
C          CONTRIBUTION(S) (IF ANY) AND SET THE PERTINENT VARIABLES.    GAUCHE
C            DETERMINE WHETHER K1 OR K2 HAS THE DOUBLE BOND.            GAUCHE
   11 KGO=2                                                             GAUCHE
      IF(I1-I2)13,31,15                                                 GAUCHE
   13 KENE=KR                                                           GAUCHE
      KANE=KL                                                           GAUCHE
      KCENE=K2                                                          GAUCHE
      KCANE=K1                                                          GAUCHE
      IF(IX(K1,1,1).EQ.9)KGO=1                                          GAUCHE
      GO TO 17                                                          GAUCHE
   15 KENE=KL                                                           GAUCHE
      KANE=KR                                                           GAUCHE
      KCENE=K1                                                          GAUCHE
      KCANE=K2                                                          GAUCHE
      IF(IX(K2,1,1).EQ.9)KGO=1                                          GAUCHE
   17 GO TO(220,240),KGO                                                GAUCHE
C          ONE CENTRAL ATOM IS A CARBON ATOM AND THE OTHER A NITROGEN   GAUCHE
C          ATOM. FIND FREQUENCY OF CORRECTION AND SET THE PERTINENT     GAUCHE
C          VARIABLES.                                                   GAUCHE
  220 IF(MAROM(KCENE).EQ.0)GO TO 230                                    GAUCHE
      ICC=0                                                             GAUCHE
      IRR=0                                                             GAUCHE
      LF=NC(KCANE)+1                                                    GAUCHE
      DO229   L=2,LF                                                    GAUCHE
      KC=IX(KCANE,L,6)                                                  GAUCHE
      IF(KC.EQ.KCENE)GO TO 229                                          GAUCHE
      IF(MAROM(KC).NE.0)GO TO 222                                       GAUCHE
      ICC=ICC+1                                                         GAUCHE
      GO TO 229                                                         GAUCHE
  222 IRR=IRR+1                                                         GAUCHE
  229 CONTINUE                                                          GAUCHE
      IF(IRR.EQ.0.OR.ICC.NE.0)RETURN                                    GAUCHE
      IF(IRR-2)232,239,31                                               GAUCHE
  230 IF(KENE.NE.3.OR.KANE.NE.2)GO TO 239                               GAUCHE
  232 IGS=IGS+1                                                         GAUCHE
      KFGAUS(IGS)=1                                                     GAUCHE
      GO TO 238                                                         GAUCHE
  239 IF(KANE.NE.3)RETURN                                               GAUCHE
      IF(KENE-2)31,232,236                                              GAUCHE
  236 IGS=IGS+1                                                         GAUCHE
      KFGAUS(IGS)=2                                                     GAUCHE
  238 DATCIS(1,IGS)=0.0+FLOAT(KFGAUS(IGS))                              GAUCHE
      GO TO 299                                                         GAUCHE
C          BOTH CENTRAL ATOMS ARE CARBON ATOMS. FIND FREQUENCY OF       GAUCHE
C          CORRECTION AND SET THE PERTINENT VARIABLES.                  GAUCHE
```

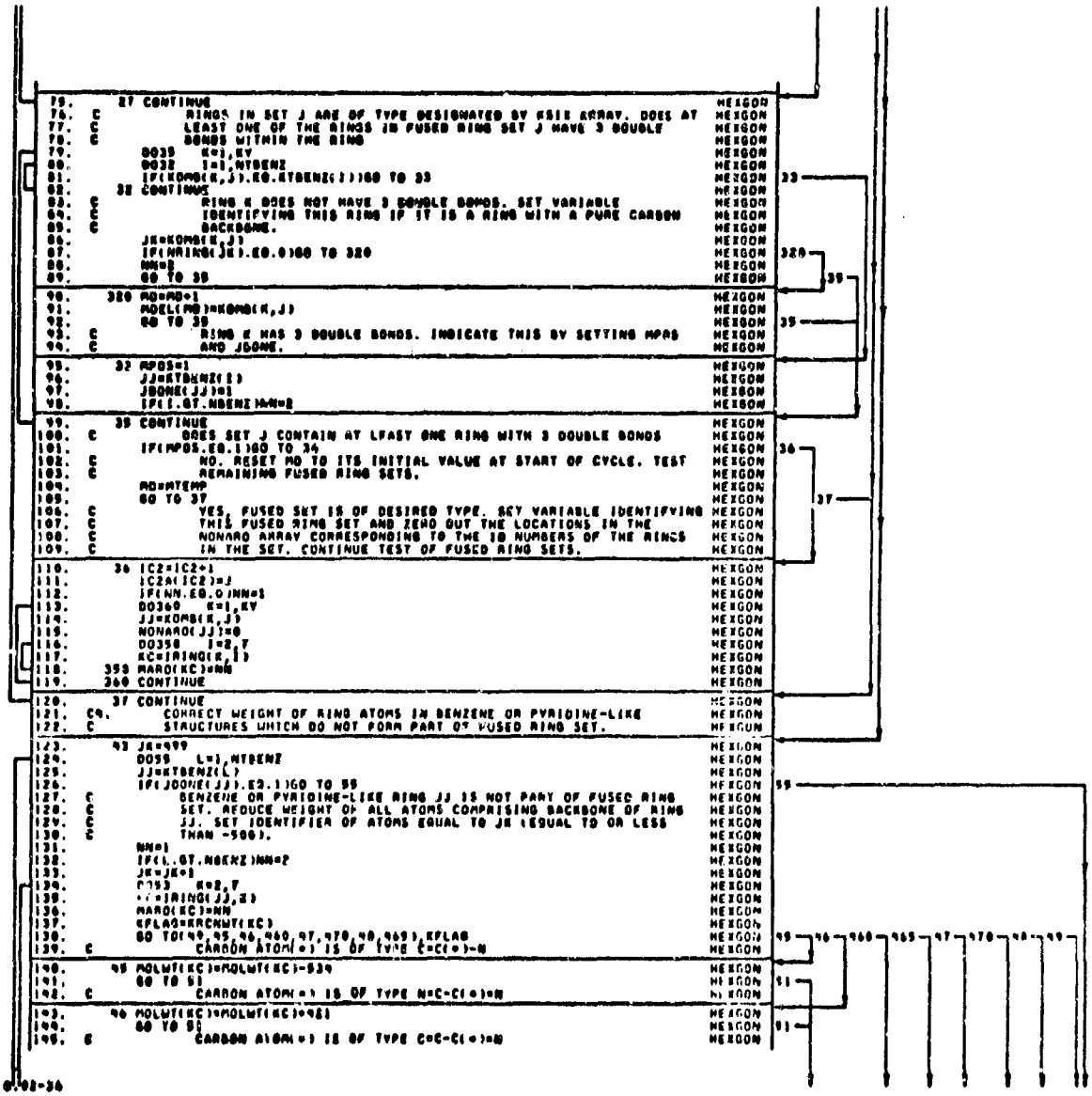
17 APR 73 0.02-36



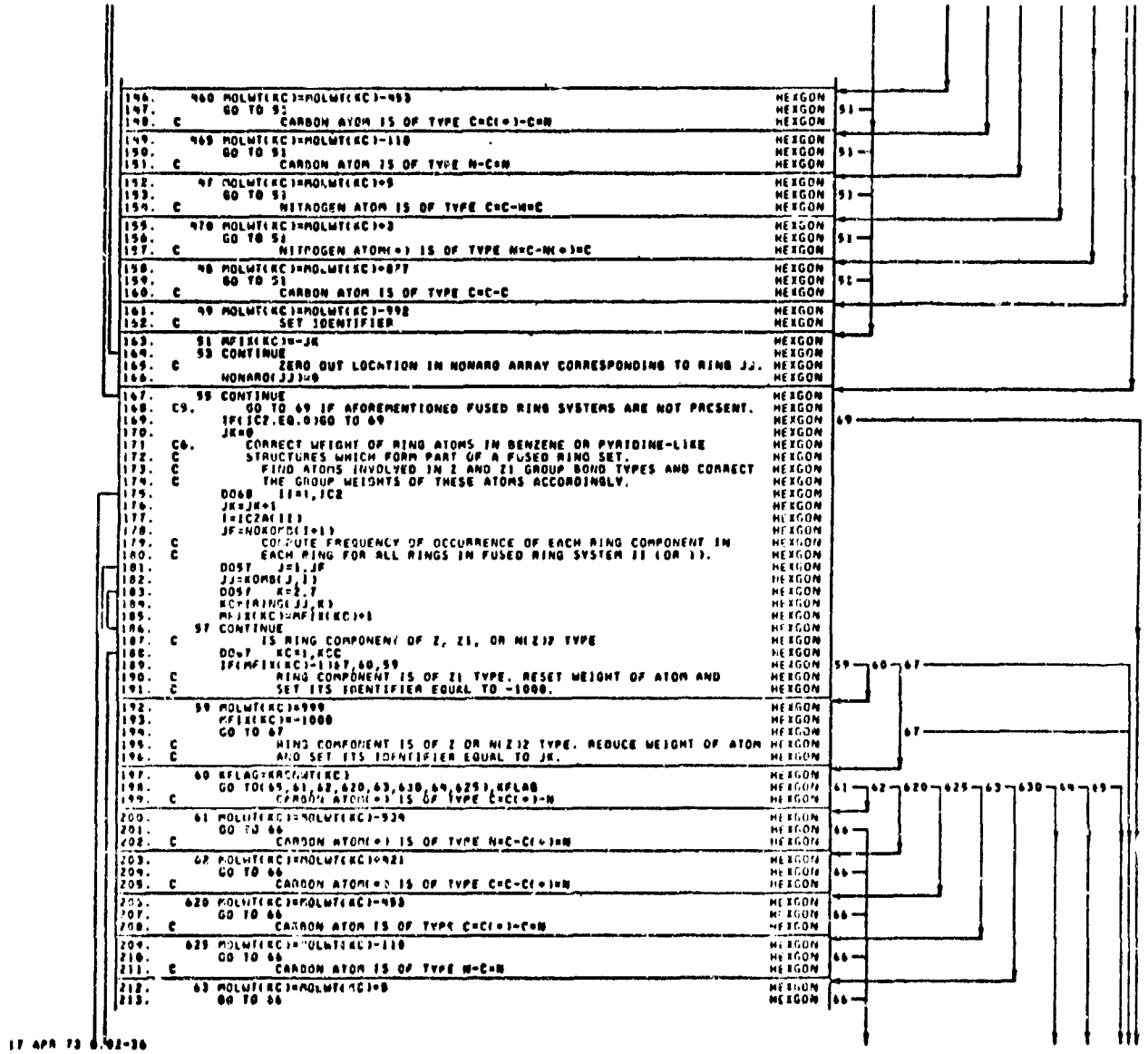
```
      240 IF(KENE.NE.3.OR.KANE.NE.3.AND.KANE.NE.4)RETURN                  GAUCHE
          IGS=IGS+1                                                      GAUCHE
          KFGAUS(IGS)=2                                                  GAUCHE
          DATCIS(I,IGS)=0.5                                              GAUCHE
      249 KTGAUS(IGS)=6                                                  GAUCHE
       21 KCGAUS(1,IGS)=K1                                               GAUCHE
          KCGAUS(2,IGS)=K2                                               GAUCHE
          GO TO 28                                                       GAUCHE   28
       23 IF(IX(K1,1,1).EQ.3.AND.IX(K2,1,1).EQ.3)RETURN                  GAUCHE
C3.0      CORE ATOMS K1 AND K2 ARE NOT BOTH OXYGEN ATOMS. SEARCH FOR     GAUCHE
C         EITHER OXYGEN GAUCHE CONTRIBUTION(S).                          GAUCHE
          IF(NW(K1).LT.3.OR.IX(K2,1,1).NE.3)GO TO 24                     GAUCHE   24
          KHEAVY=K1                                                      GAUCHE
          KOXY=K2                                                        GAUCHE
          GO TO 25                                                       GAUCHE   25
       24 IF(IX(K1,1,1).NE.3.OR.NW(K2).LT.3)RETURN                       GAUCHE
          KHEAVY=K2                                                      GAUCHE
          KOXY=K1                                                        GAUCHE
C              ONE ATOM (K1 OR K2) IS AN OXYGEN ATOM THE OTHER ATOM IS A  GAUCHE
C              HEAVY ATOM WHICH HAS THE CORRECT NUMBER OF NON-HYDROGEN    GAUCHE
C              ATOMS.                                                     GAUCHE
       25 IF(IX(KHEAVY,1,1).NE.2.AND.IX(KHEAVY,1,1).NE.4.OR.NW(KOXY).NE.2)GAUCHE
          RETURN                                                         GAUCHE
C              THE OXYGEN ATOM HAS TWO CORE LIGANDS. DETERMINE IF THE OTHER GAUCHE
C              ATOM BONDED TO THE OXYGEN ATOM IS ALSO A CARBON ATOM, BUT  GAUCHE
C              NOT -CN.                                                   GAUCHE
          DO250   L=2,3                                                  GAUCHE
          IF(IX(KOXY,L,6).EQ.KHEAVY)GO TO 250                            GAUCHE   250
          IF(IX(KOXY,L,1).NE.2)GO TO 248                                 GAUCHE   248
          LL=IX(KOXY,L,1)                                                GAUCHE
          IF(KON(LL).GE.3)GO TO 260                                      GAUCHE   260
      248 IF(IX(KOXY,L,1).EQ.6)GO TO 250                                 GAUCHE   260
      250 CONTINUE                                                       GAUCHE
          RETURN                                                         GAUCHE
C                  YES IT IS. FIND NUMBER OF LIGANDS BONDED TO KHEAVY WHICH GAUCHE
C                  ARE HEAVY (NON-HYDROGEN) ATOMS.                      GAUCHE
      260 KL=0                                                           GAUCHE
          LF=KON(KHEAVY)+1                                               GAUCHE
          DO290   L=2,LF                                                 GAUCHE
          IF(IX(KHEAVY,L,6).EQ.KOXY.OR.IX(KHEAVY,L,1).EQ.1)GO TO 290     GAUCHE   290
          KL=KL+1                                                        GAUCHE
      290 CONTINUE                                                       GAUCHE
C              DETERMINE IF INTERACTION EXISTS. IF SO, FIND FREQUENCY     GAUCHE
C              OF CORRECTION AND SET THE PERTINENT VARIABLES.             GAUCHE
          IF(KL.LT.3)RETURN                                              GAUCHE
          IGS=IGS+1                                                      GAUCHE
          IF(NW(KHEAVY).NE.3)GO TO 26                                    GAUCHE   26
          KFGAUS(IGS)=1                                                  GAUCHE
          GO TO 27                                                       GAUCHE   27
       26 KFGAUS(IGS)=2                                                  GAUCHE
       27 KTGAUS(IGS)=7                                                  GAUCHE
          KCGAUS(1,IGS)=KHEAVY                                           GAUCHE
          KCGAUS(2,IGS)=KOXY                                             GAUCHE
          DATCIS(1,IGS)=0.3*FLOAT(KFGAUS(IGS))                           GAUCHE
          IGO=IGO+1                                                      GAUCHE
C*.0      FIND SUM TOTAL OF GAUCHE CORRECTIONS FOR THE HEAT OF FORMATION GAUCHE
C         AND INCREMENT GAUCHE COUNTER IGH.                              GAUCHE
       28 GAUCHN=GAUCHN+DATCIS(1,IGS)                                    GAUCHE
          IGH=IGH+1                                                      GAUCHE
       31 RETURN                                                         GAUCHE
          END                                                            GAUCHE
```

```
      SUBROUTINE HEXGON(IGS,IORTHO,IORPAR,DORTHO,DORPAR,NRING,SRING,     HEXGON
     1CPSYM)                                                             HEXGON
C        THIS SUBROUTINE ASSIGNS THE NECESSARY WEIGHT CORRECTIONS TO     HEXGON
C        GROUPS OF FUSED RING SYSTEMS CONTAINING BENZENE CARBONS AND/OR  HEXGON
C        PYRIDINE-LIKE STRUCTURES WHICH FORM CONJUGATED SIX-MEMBERED     HEXGON
C        RINGS AND/OR TO NON-FUSED RINGS OF THE AFOREMENTIONED           HEXGON
C        STRUCTURES AND TO ALL GROUPS BONDED TO THESE RING ATOMS. IT     HEXGON
C        ALSO PRINTS OUT THE ASSIGNED WEIGHTS, THE COMPOSITION OF THE    HEXGON
C        MOLECULE, AND THE IX ARRAY.                                     HEXGON
      INTEGER SYM1(4),SYMBOL(9),GRID(50,80)                              HEXGON
      INTEGER WEIGHT(9)                                                  HEXGON
      DIMENSION KBENZ(40),KSIX(40),MDEL(40),IC2A(10),JDONE(40),          HEXGON
     1KOMB(40,10),NOKOMB(11),MFIX(100),CPSYM(4),DORTHO(6),JBOND(100),    HEXGON
     2NOK(10,10),NRING(40),KRCNOR(100),KRCNWT(100),KNBENZ(40),KTBENZ(40) HEXGON
     3,NONARO(40),KTGAUS(150),KCGAUS(3,150),NARO(100)                   HEXGON
      COMMON/BLK1/NO,NOS,SYM1,SYMBOL,NOVAL(9),GRID                       HEXGON
      COMMON/BLK2/WEIGHT,NWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC      HEXGON
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),   HEXGON
     1IDBR(100),IB(100,8),IRG,NOER                                       HEXGON
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT      HEXGON
      EQUIVALENCE (NBENZ,IB(10,1)),(NOSIX,IB(11,1)),(MD,IB(12,1)),       HEXGON
     1(MDEL(1),IB(13,1)),(KBENZ(1),IB(53,1)),(KSIX(1),IB(81,2)),        HEXGON
     2(JDONE(1),IB(81,3)),(IC2,IB(21,4)),(IC2A(1),IB(22,4)),(KOMB(1,1), HEXGON
     3IB(62,4)),(NOKOMB(1),IB(80,8)),(MFIX(1),IMATX(1,53)),(JBOND(1),    HEXGON
     4IMATX(1,51)),(NOK(1,1),IMATX(1,55)),(NRING(1),GRID(1,75)),         HEXGON
     5(KRCNOR(1),GRID(1,76)),(KRCNWT(1),GRID(1,78)),(NNBENZ,GRID(1,80)), HEXGON
     6(KNBENZ(1),GRID(2,80)),(NTBENZ,IB(96,8)),(KTBENZ(1),GRID(41,27)),  HEXGON
     7(NONARO(1),IMATX(2,49)),(KTGAUS(1),IMATX(1,40)),(KCGAUS(1,1),      HEXGON
     8IMATX(1,72)),(NARO(1),GRID(1,25))                                  HEXGON
C1.      INITIALIZE VARIABLES.                                           HEXGON
      IC2=0                                                              HEXGON
      DO1   L=1,IRC                                                      HEXGON
      JDONE(L)=0                                                         HEXGON
    1 CONTINUE                                                           HEXGON
      DO3   KC=1,KCC                                                     HEXGON
      MFIX(KC)=0                                                         HEXGON
      JBOND(KC)=0                                                        HEXGON
    3 CONTINUE                                                           HEXGON
      IF(NBENZ.EQ.0)GO TO 9                                              HEXGON  9
      DO7   L=1,NBENZ                                                    HEXGON
    7 KTBENZ(L)=KBENZ(L)                                                 HEXGON
    9 IF(NNBENZ.EQ.0)GO TO 15                                            HEXGON 15
      I=NBENZ+1                                                          HEXGON
      DO11  L=1,NNBENZ                                                   HEXGON
      KTBENZ(I)=KNBENZ(L)                                                HEXGON
   11 I=I+1                                                              HEXGON
   15 NTBENZ=NBENZ+NNBENZ                                                HEXGON
C2.      IF FUSED RING SYSTEMS ARE NOT PRESENT, GO TO 43. OTHERWISE,     HEXGON
C        CONTINUE.                                                       HEXGON
      IF(NOKOMB(1).EQ.0)GO TO 43                                         HEXGON 43
      IC=NOKOMB(1)                                                       HEXGON
C3.      DETERMINE WHETHER ANY OF FUSED RING SETS ARE COMPOSED OF PURE   HEXGON
C        CARBON OR CARBON-NITROGEN BACKBONE, 6-CENTERED RINGS OF THE     HEXGON
C        TYPE DESIGNATED IN ARRAY KSIX, AT LEAST ONE OF WHICH MUST HAVE  HEXGON
C        A BENZENE OR PYRIDINE-LIKE STRUCTURE WITH 3 DOUBLE BONDS PER    HEXGON
C        RING.                                                           HEXGON
      DO37  J=1,IC                                                       HEXGON
C           DOES EACH ADJOINING RING PAIR IN FUSED RING SET J HAVE TWO   HEXGON
C           ATOMS IN COMMON AND A CARBON OR PYRIDINE-TYPE UNSATURATED    HEXGON
C           STRUCTURE                                                    HEXGON
      IF(NOK(2,J).NE.2)GO TO 37                                          HEXGON 37
      NN=0                                                               HEXGON
C              EACH RING IN SET J HAS 2 ATOMS IN COMMON WITH EACH RING TO HEXGON
C              WHICH IT IS FUSED. NOW TEST WHETHER RINGS IN SET J ARE    HEXGON
C              UNSATURATED, 6-CENTERED CARBON OR PYRIDINE-LIKE RINGS OF  HEXGON
C              TYPE DESIGNATED BY ARRAY KSIX.                            HEXGON
      MPOS=0                                                             HEXGON
      MTEMP=MD                                                           HEXGON
      KY=NOKOMB(J+1)                                                     HEXGON
      DO27  K=1,KY                                                       HEXGON
      DO25  I=1,NOSIX                                                    HEXGON
      IF(KOMB(K,J).EQ.KSIX(I))GO TO 27                                   HEXGON 27
   25 CONTINUE                                                           HEXGON
      GO TO 37                                                           HEXGON    37
```

17 APR 73 8.02-36

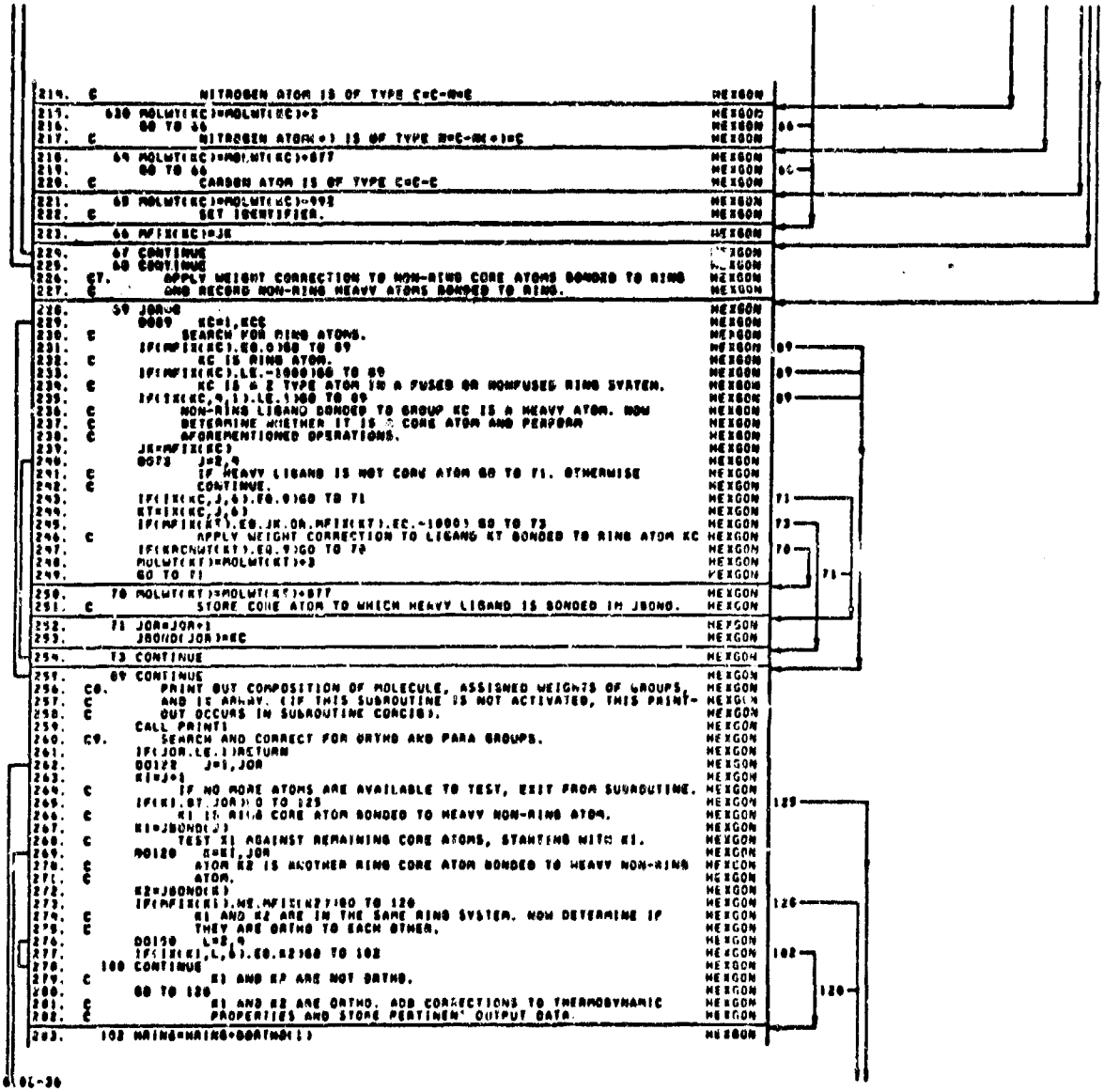


```
      37 CONTINUE                                                       MEXGON
C           RINGS IN SET J ARE OF TYPE DESIGNATED BY KSIX ARRAY. DOES AT  MEXGON
C           LEAST ONE OF THE RINGS IN FUSED RING SET J HAVE 3 DOUBLE      MEXGON
C           BONDS WITHIN THE RING                                         MEXGON
      DO35   K=1,KV                                                       MEXGON
      DO32   I=1,NTBENZ                                                   MEXGON
      IF(KOMB(K,J).EQ.ITBENZ(I))GO TO 33                                  MEXGON
   32 CONTINUE                                                            MEXGON
C           RING K DOES NOT HAVE 3 DOUBLE BONDS. SET VARIABLE             MEXGON
C           IDENTIFYING THIS RING IF IT IS A RING WITH A PURE CARBON      MEXGON
C           BACKBONE.                                                     MEXGON
      JE=KOMB(K,J)                                                        MEXGON
      IF(NRING(JE).EQ.6)GO TO 320                                         MEXGON
      NN=2                                                                MEXGON
      GO TO 35                                                            MEXGON
  320 MP=MP+1                                                             MEXGON
      MDEL(MP)=KOMB(K,J)                                                  MEXGON
      GO TO 35                                                            MEXGON
C           RING K HAS 3 DOUBLE BONDS. INDICATE THIS BY SETTING MPAS      MEXGON
C           AND JBONE.                                                    MEXGON
   33 MPOS=1                                                              MEXGON
      JJ=ITBENZ(I)                                                        MEXGON
      JBONE(JJ)=1                                                         MEXGON
      IF(I.GT.NBENZ)NN=2                                                  MEXGON
   35 CONTINUE                                                            MEXGON
C           DOES SET J CONTAIN AT LEAST ONE RING WITH 3 DOUBLE BONDS      MEXGON
      IF(MPOS.EQ.1)GO TO 34                                               MEXGON
C           NO. RESET MP TO ITS INITIAL VALUE AT START OF CYCLE. TEST     MEXGON
C           REMAINING FUSED RING SETS.                                    MEXGON
      MP=MTEMP                                                            MEXGON
      GO TO 37                                                            MEXGON
C           YES, FUSED SET IS OF DESIRED TYPE. SET VARIABLE IDENTIFYING   MEXGON
C           THIS FUSED RING SET AND ZERO OUT THE LOCATIONS IN THE         MEXGON
C           NONARO ARRAY CORRESPONDING TO THE ID NUMBERS OF THE RINGS     MEXGON
C           IN THE SET. CONTINUE TEST OF FUSED RING SETS.                 MEXGON
   36 IC2=IC2+1                                                           MEXGON
      ICZA(IC2)=J                                                         MEXGON
      IF(NN.EQ.0)NN=1                                                     MEXGON
      DO360   K=1,KV                                                      MEXGON
      JJ=KOMB(K,J)                                                        MEXGON
      NONARO(JJ)=0                                                        MEXGON
      DO358   I=2,7                                                       MEXGON
      KC=IRING(K,I)                                                       MEXGON
  358 NARO(KC)=NN                                                         MEXGON
  360 CONTINUE                                                            MEXGON
   37 CONTINUE                                                            MEXGON
C.          CORRECT WEIGHT OF RING ATOMS IN BENZENE OR PYRIDINE-LIKE      MEXGON
C           STRUCTURES WHICH DO NOT FORM PART OF FUSED RING SET.          MEXGON
   41 JE=999                                                              MEXGON
      DO59   L=1,NTBENZ                                                   MEXGON
      JJ=ITBENZ(L)                                                        MEXGON
      IF(JBONE(JJ).EQ.1)GO TO 59                                          MEXGON
C           BENZENE OR PYRIDINE-LIKE RING JJ IS NOT PART OF FUSED RING    MEXGON
C           SET. REDUCE WEIGHT OF ALL ATOMS COMPRISING BACKBONE OF RING   MEXGON
C           JJ. SET IDENTIFIER OF ATOMS EQUAL TO JE (EQUAL TO OR LESS     MEXGON
C           THAN -999).                                                   MEXGON
      NN=1                                                                MEXGON
      IF(L.GT.NBENZ)NN=2                                                  MEXGON
      JE=JE-1                                                             MEXGON
      DO53   K=2,7                                                        MEXGON
      KC=IRING(JJ,K)                                                      MEXGON
      NARO(KC)=NN                                                         MEXGON
      KFLAG=KRCHWT(KC)                                                    MEXGON
      GO TO(99,95,96,960,97,970,98,969),KFLAG                             MEXGON
C           CARBON ATOM(=) IS OF TYPE C=C(=O)-N                           MEXGON
   99 MOLWT(KC)=MOLWT(KC)-539                                             MEXGON
      GO TO 53                                                            MEXGON
C           CARBON ATOM(=) IS OF TYPE N=C-C(=O)=N                         MEXGON
   96 MOLWT(KC)=MOLWT(KC)+421                                             MEXGON
      GO TO 53                                                            MEXGON
C           CARBON ATOM(=) IS OF TYPE C=C-C(=O)=N                         MEXGON
```



```
196.     960 MOLWT(KC)=MOLWT(KC)-953                                    HEXGON
197.         GO TO 91                                                   HEXGON
198.   C             CARBON ATOM IS OF TYPE C=C(=)-C=N                  HEXGON
199.     965 MOLWT(KC)=MOLWT(KC)-110                                    HEXGON
200.         GO TO 91                                                   HEXGON
201.   C             CARBON ATOM IS OF TYPE N-C=N                       HEXGON
202.      97 MOLWT(KC)=MOLWT(KC)+9                                      HEXGON
203.         GO TO 91                                                   HEXGON
204.   C             NITROGEN ATOM IS OF TYPE C=C-N=C                   HEXGON
205.     970 MOLWT(KC)=MOLWT(KC)+3                                      HEXGON
206.         GO TO 91                                                   HEXGON
207.   C             NITROGEN ATOM(=) IS OF TYPE N=C-N(=)=C             HEXGON
208.      98 MOLWT(KC)=MOLWT(KC)+877                                    HEXGON
209.         GO TO 91                                                   HEXGON
210.   C             CARBON ATOM IS OF TYPE C=C-C                       HEXGON
211.      99 MOLWT(KC)=MOLWT(KC)-998                                    HEXGON
212.   C             SET IDENTIFIER                                     HEXGON
213.      91 MFI(KC)=-JK                                                HEXGON
214.      93 CONTINUE                                                   HEXGON
215.   C             ZERO OUT LOCATION IN NONARO ARRAY CORRESPONDING TO RING JJ. HEXGON
216.         NONARO(JJ)=0                                               HEXGON
217.      95 CONTINUE                                                   HEXGON
218.   C9.           GO TO 69 IF AFOREMENTIONED FUSED RING SYSTEMS ARE NOT PRESENT. HEXGON
219.         IF(IC2.EQ.0)GO TO 69                                       HEXGON
220.         JK=0                                                       HEXGON
221.   C6.       CORRECT WEIGHT OF RING ATOMS IN BENZENE OR PYRIDINE-LIKE HEXGON
222.   C         STRUCTURES WHICH FORM PART OF A FUSED RING SET.        HEXGON
223.   C           FIND ATOMS INVOLVED IN Z AND Z1 GROUP BOND TYPES AND CORRECT HEXGON
224.   C           THE GROUP WEIGHTS OF THESE ATOMS ACCORDINGLY.        HEXGON
225.         DO68   II=1,IC2                                            HEXGON
226.         JK=JK+1                                                    HEXGON
227.         I=IC2A(II)                                                 HEXGON
228.         JF=NOKOMB(I+1)                                             HEXGON
229.   C           COMPUTE FREQUENCY OF OCCURRENCE OF EACH RING COMPONENT IN HEXGON
230.   C           EACH RING FOR ALL RINGS IN FUSED RING SYSTEM II (OR I). HEXGON
231.         DO97   J=I,JF                                              HEXGON
232.         JJ=KOMB(J,I)                                               HEXGON
233.         DO97   K=2,7                                               HEXGON
234.         KC=IRING(JJ,K)                                             HEXGON
235.         MFI(KC)=MFI(KC)+1                                          HEXGON
236.      97 CONTINUE                                                   HEXGON
237.   C             IS RING COMPONENT OF Z, Z1, OR NIZ)2 TYPE          HEXGON
238.         DO67   KC=1,KSC                                            HEXGON
239.         IF(MFI(KC)-1)67,60,59                                      HEXGON
240.   C             RING COMPONENT IS OF Z1 TYPE. RESET WEIGHT OF ATOM AND HEXGON
241.   C             SET ITS IDENTIFIER EQUAL TO -1000.                 HEXGON
242.      59 MOLWT(KC)=999                                              HEXGON
243.         MFI(KC)=-1000                                              HEXGON
244.         GO TO 67                                                   HEXGON
245.   C             RING COMPONENT IS OF Z OR NIZ)2 TYPE. REDUCE WEIGHT OF ATOM HEXGON
246.   C             AND SET ITS IDENTIFIER EQUAL TO JK.                HEXGON
247.      60 KFLAG=KACWT(KC)                                            HEXGON
248.         GO TO(69,61,62,620,63,630,64,625),KFLAG                    HEXGON
249.   C             CARBON ATOM(=) IS OF TYPE C=C(=)-N                 HEXGON
250.      61 MOLWT(KC)=MOLWT(KC)-939                                    HEXGON
251.         GO TO 66                                                   HEXGON
252.   C             CARBON ATOM(=) IS OF TYPE N=C-C(=)=N               HEXGON
253.      62 MOLWT(KC)=MOLWT(KC)+921                                    HEXGON
254.         GO TO 66                                                   HEXGON
255.   C             CARBON ATOM(=) IS OF TYPE C=C-C(=)=N               HEXGON
256.     620 MOLWT(KC)=MOLWT(KC)-953                                    HEXGON
257.         GO TO 66                                                   HEXGON
258.   C             CARBON ATOM IS OF TYPE C=C(=)-C=N                  HEXGON
259.     625 MOLWT(KC)=MOLWT(KC)-110                                    HEXGON
260.         GO TO 66                                                   HEXGON
261.   C             CARBON ATOM IS OF TYPE N-C=N                       HEXGON
262.      63 MOLWT(KC)=MOLWT(KC)+9                                      HEXGON
263.         GO TO 66                                                   HEXGON
```

17 APR 73 0:02:36

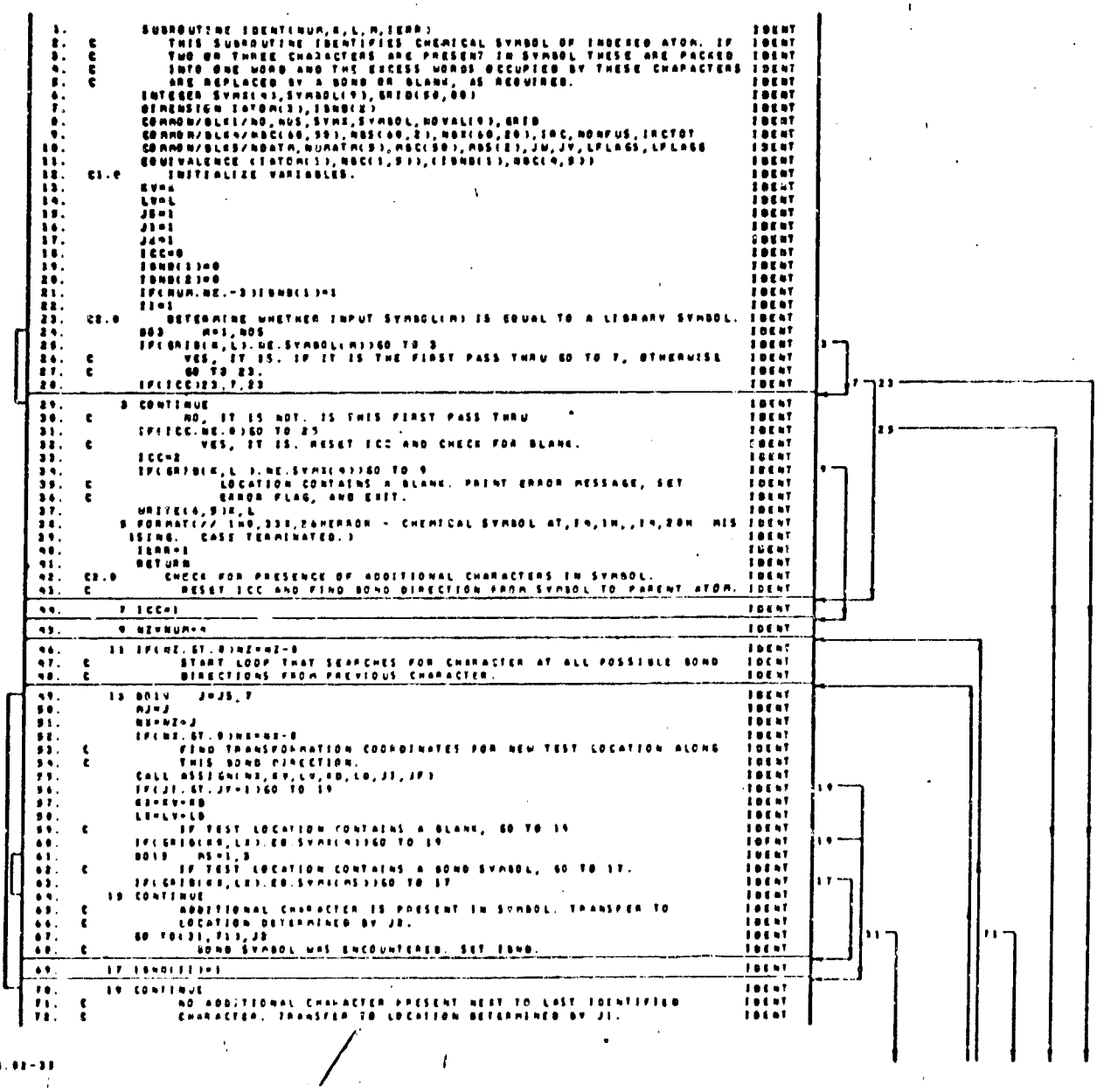


```
C              NITROGEN ATOM IS OF TYPE C=C-N=C
  630 MOLWT(KC)=MOLWT(KC)+3
      GO TO 66
C              NITROGEN ATOM(+) IS OF TYPE N=C-N(+)=C
   64 MOLWT(KC)=MOLWT(KC)+877
      GO TO 66
C              CARBON ATOM IS OF TYPE C=C-C
   65 MOLWT(KC)=MOLWT(KC)-992
C              SET IDENTIFIER.
   66 MFIX(KC)=JR
   67 CONTINUE
   68 CONTINUE
C7.      APPLY WEIGHT CORRECTION TO NON-RING CORE ATOMS BONDED TO RING
C        AND RECORD NON-RING HEAVY ATOMS BONDED TO RING.
   69 JBON=0
      DO69    KC=1,KCC
C            SEARCH FOR RING ATOMS.
      IF(MFIX(KC).EQ.0)GO TO 69
C              KC IS RING ATOM.
      IF(MFIX(KC).LE.-1000)GO TO 69
C              KC IS A Z TYPE ATOM IN A FUSED OR NONFUSED RING SYSTEM.
      IF(IX(KC,9,1).LE.1)GO TO 69
C            NON-RING LIGAND BONDED TO GROUP KC IS A HEAVY ATOM. NOW
C            DETERMINE WHETHER IT IS A CORE ATOM AND PERFORM
C            AFOREMENTIONED OPERATIONS.
      JK=MFIX(KC)
      DO73    J=2,9
C              IF HEAVY LIGAND IS NOT CORE ATOM GO TO 71. OTHERWISE
C              CONTINUE.
      IF(IX(KC,J,6).EQ.0)GO TO 71
      KT=IX(KC,J,6)
      IF(MFIX(KT).EQ.JK.OR.MFIX(KT).EC.-1000) GO TO 73
C              APPLY WEIGHT CORRECTION TO LIGAND KT BONDED TO RING ATOM KC
      IF(KRCNWT(KT).EQ.9)GO TO 70
      MOLWT(KT)=MOLWT(KT)+3
      GO TO 71
   70 MOLWT(KT)=MOLWT(KT)+877
C              STORE CORE ATOM TO WHICH HEAVY LIGAND IS BONDED IN JBOND.
   71 JOR=JOR+1
      JBOND(JOR)=KC
   73 CONTINUE
   69 CONTINUE
C9.      PRINT OUT COMPOSITION OF MOLECULE, ASSIGNED WEIGHTS OF GROUPS,
C        AND IS ARRAY. (IF THIS SUBROUTINE IS NOT ACTIVATED, THIS PRINT-
C        OUT OCCURS IN SUBROUTINE CONCIS).
      CALL PRINT1
C9.      SEARCH AND CORRECT FOR ORTHO AND PARA GROUPS.
      IF(JOR.LE.1)RETURN
      DO122   J=1,JOR
      KI=J+1
C              IF NO MORE ATOMS ARE AVAILABLE TO TEST, EXIT FROM SUBROUTINE.
      IF(KI.GT.JOR))GO TO 125
C              KI IS RING CORE ATOM BONDED TO HEAVY NON-RING ATOM.
      KI=JBOND(J)
C              TEST KI AGAINST REMAINING CORE ATOMS, STARTING WITH KI.
      DO120   K=KI,JOR
C              ATOM K2 IS ANOTHER RING CORE ATOM BONDED TO HEAVY NON-RING
C              ATOM.
      K2=JBOND(K)
      IF(MFIX(KI).NE.MFIX(K2))GO TO 120
C              KI AND K2 ARE IN THE SAME RING SYSTEM. NOW DETERMINE IF
C              THEY ARE ORTHO TO EACH OTHER.
      DO130   L=2,9
      IF(IX(K,L,6).EQ.K2)GO TO 102
  130 CONTINUE
C              KI AND K2 ARE NOT ORTHO.
      GO TO 120
C              KI AND K2 ARE ORTHO. ADD CORRECTIONS TO THERMODYNAMIC
C              PROPERTIES AND STORE PERTINENT OUTPUT DATA.
  102 HRING=HRING+DORTHO(I)
```

17 APR 73 8:01:36

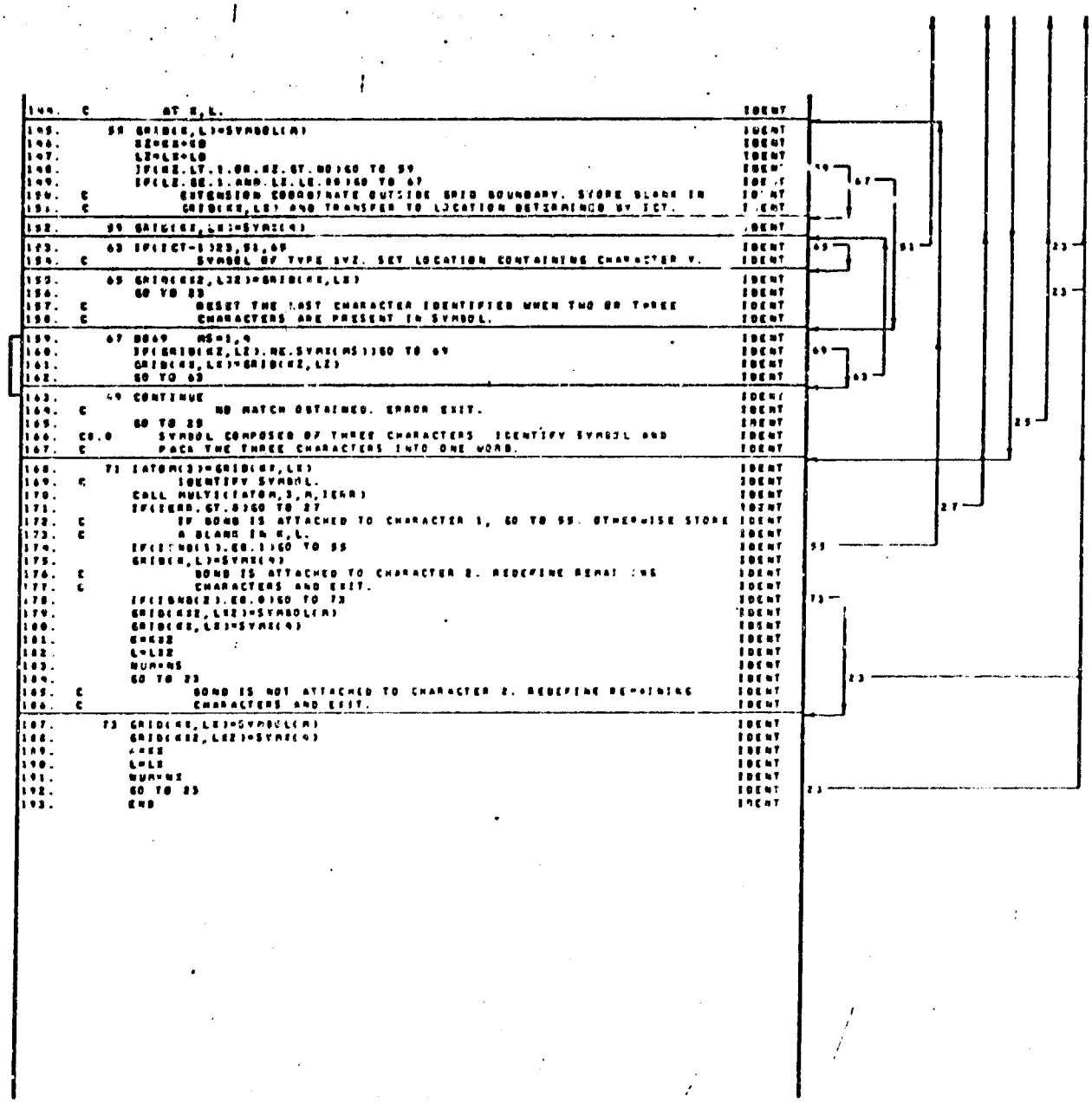
```
284.           SRING=SRING+DORTHO(2)                                    HEXGON
285.           DO109   KK=1,9                                           HEXGON
286.           CPSYM(KK)=CPSYM(KK)+DORTHO(KK+2)                         HEXGON
287.       109 CONTINUE                                                 HEXGON
288.           IORTHO=IORTHO+1                                          HEXGON
289.           IGS=IGS+1                                                HEXGON
290.           KTGAUS(IGS)=2                                            HEXGON
291.           KCGAUS(1,IGS)=K1                                         HEXGON
292.           KCGAUS(2,IGS)=K2                                         HEXGON
293.       120 CONTINUE                                                 HEXGON
294.       122 CONTINUE                                                 HEXGON
295.     C10   SEARCH AND CORRECT FOR HEAVY GROUPS BONDED TO RING AND WHICH  HEXGON
296.     C     ARE ORTHO OR PARA TO NITROGEN ATOM IN PYRIDINE-LIKE STRUCTURES. HEXGON
297.     C     THIS CORRECTION APPLIES ONLY TO RINGS CONTAINING ONLY ONE   HEXGON
298.     C     NITROGEN ATOM.                                             HEXGON
299.       125 DO139   KC=1,KCC                                         HEXGON
300.           IF(KRCNOR(KC).EQ.0)GO TO 139                             HEXGON  139
301.     C        GROUP ATOM KC CONTAINED IN PYRIDINE-LIKE STRUCTURE. NOW  HEXGON
302.     C        DETERMINE WHETHER KC IS BONDED TO HEAVY NON-RING ATOM.   HEXGON
303.           DO137   J=1,JOR                                          HEXGON
304.           IF(KC.NE.JBOND(J))GO TO 137                              HEXGON  137
305.     C        YES, KC IS BONDED TO HEAVY NON-RING ATOM. ADD CORRECTION  HEXGON
306.     C        FOR THE HEAT OF FORMATION AND STORE PERTINENT OUTPUT DATA. HEXGON
307.           HRING=HRING+DORPAR                                       HEXGON
308.           IORPAR=IORPAR+1                                          HEXGON
309.           IGS=IGS+1                                                HEXGON
310.           IF(KRCNOR(KC).LT.0)GO TO 131                             HEXGON  131
311.     C           HEAVY ATOM IS ORTHO TO NITROGEN ATOM.              HEXGON
312.           KTGAUS(IGS)=2                                            HEXGON
313.           KCGAUS(1,IGS)=KRCNOR(KC)                                 HEXGON
314.           KCGAUS(2,IGS)=JBOND(J)                                   HEXGON
315.           GO TO 137                                                HEXGON  137
316.     C           HEAVY ATOM IS PARA TO NITROGEN ATOM.               HEXGON
317.       131 KTGAUS(IGS)=3                                            HEXGON
318.           KCGAUS(1,IGS)=IABS(KRCNOR(KC))                           HEXGON
319.           KCGAUS(2,IGS)=JBOND(J)                                   HEXGON
320.       137 CONTINUE                                                 HEXGON
321.       139 CONTINUE                                                 HEXGON
322.           RETURN                                                   HEXGON
323.           END                                                      HEXGON
```

17 APR 73 8.02-36



```
      SUBROUTINE IDENT(NUM,K,L,M,IERR)                                  IDENT
C        THIS SUBROUTINE IDENTIFIES CHEMICAL SYMBOL OF INDEXED ATOM. IF  IDENT
C        TWO OR THREE CHARACTERS ARE PRESENT IN SYMBOL THESE ARE PACKED  IDENT
C        INTO ONE WORD AND THE EXCESS WORDS OCCUPIED BY THESE CHARACTERS IDENT
C        ARE REPLACED BY A BOND OR BLANK, AS REQUIRED.                   IDENT
      INTEGER SYMBL(3),SYMBOL(9),GRID(50,80)                            IDENT
      DIMENSION IATOM(3),ISNB(2)                                        IDENT
      COMMON/BLK1/NO,NUS,SYMS,SYMBOL,NOVALE(9),GRID                     IDENT
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBR(60,20),IRC,NONPUS,IRCTOT     IDENT
      COMMON/BLK5/NBATM,NUMATM(9),MBC(50),MBS(2),JM,JY,LPLASS,LPLASS    IDENT
      EQUIVALENCE (IATOM(1),NBC(1,51)),(ISNB(1),NBC(4,51))              IDENT
C1.0     INITIALIZE VARIABLES.                                          IDENT
      KM=K                                                              IDENT
      LM=L                                                              IDENT
      JB=1                                                              IDENT
      J1=1                                                              IDENT
      J4=1                                                              IDENT
      ICC=0                                                             IDENT
      ISNB(1)=0                                                         IDENT
      ISNB(2)=0                                                         IDENT
      IF(NUM.NE.-3)ISNB(1)=1                                            IDENT
      J1=1                                                              IDENT
C2.0     DETERMINE WHETHER INPUT SYMBOL(M) IS EQUAL TO A LIBRARY SYMBOL. IDENT
      DO3    M=1,NOS                                                    IDENT
      IF(GRID(K,L).NE.SYMBOL(M))GO TO 3                                 IDENT
C           YES, IT IS. IF IT IS THE FIRST PASS THRU GO TO 7, OTHERWISE  IDENT
C           GO TO 23.                                                   IDENT
      IF(ICC)23,7,23                                                    IDENT
    3 CONTINUE                                                          IDENT
C        NO, IT IS NOT. IS THIS FIRST PASS THRU                         IDENT
      IF(ICC.NE.0)GO TO 25                                              IDENT
C           YES, IT IS. RESET ICC AND CHECK FOR BLANK.                  IDENT
      ICC=2                                                             IDENT
      IF(GRID(K,L ).NE.SYMS(9))GO TO 9                                  IDENT
C              LOCATION CONTAINS A BLANK. PRINT ERROR MESSAGE, SET      IDENT
C              ERROR FLAG, AND EXIT.                                    IDENT
      WRITE(6,5)K,L                                                     IDENT
    5 FORMAT(// 1H0,33X,26HERROR - CHEMICAL SYMBOL AT,I4,1H,,I4,20H  HAS IDENT
     1ISING.  CASE TERMINATED.)                                         IDENT
      IERR=1                                                            IDENT
      RETURN                                                            IDENT
C2.0     CHECK FOR PRESENCE OF ADDITIONAL CHARACTERS IN SYMBOL.         IDENT
C           RESET ICC AND FIND BOND DIRECTION FROM SYMBOL TO PARENT ATOM. IDENT
    7 ICC=1                                                             IDENT
    9 NEWNUM=4                                                          IDENT
   11 IF(NZ.ST.0)NZ=NZ-8                                                IDENT
C           START LOOP THAT SEARCHES FOR CHARACTER AT ALL POSSIBLE BOND  IDENT
C           DIRECTIONS FROM PREVIOUS CHARACTER.                         IDENT
   13 DO19    J=J5,7                                                    IDENT
      MJ=J                                                              IDENT
      NB=NZ+J                                                           IDENT
      IF(NB.ST.0)NB=NB-8                                                IDENT
C           FIND TRANSFORMATION COORDINATES FOR NEW TEST LOCATION ALONG  IDENT
C           THIS BOND DIRECTION.                                        IDENT
      CALL ASSIGN(NB,KV,LV,KB,LB,J1,JF)                                 IDENT
      IF(J1.ST.JF+1)GO TO 19                                            IDENT
      KB=KV+KB                                                          IDENT
      LB=LV+LB                                                          IDENT
C           IF TEST LOCATION CONTAINS A BLANK, GO TO 19                 IDENT
      IF(GRID(KB,LB).EQ.SYMS(9))GO TO 19                                IDENT
      DO17    MS=1,3                                                    IDENT
C           IF TEST LOCATION CONTAINS A BOND SYMBOL, GO TO 17.          IDENT
      IF(GRID(KB,LB).EQ.SYMS(MS))GO TO 17                               IDENT
   19 CONTINUE                                                          IDENT
C           ADDITIONAL CHARACTER IS PRESENT IN SYMBOL. TRANSFER TO       IDENT
C           LOCATION DETERMINED BY J8.                                  IDENT
      GO TO(31,713,J8                                                   IDENT
C              BOND SYMBOL WAS ENCOUNTERED. SET ISNB.                   IDENT
   17 IF(ISNB(1))19                                                     IDENT
   19 CONTINUE                                                          IDENT
C           NO ADDITIONAL CHARACTER PRESENT NEXT TO LAST IDENTIFIED      IDENT
C           CHARACTER. TRANSFER TO LOCATION DETERMINED BY J1.           IDENT
```

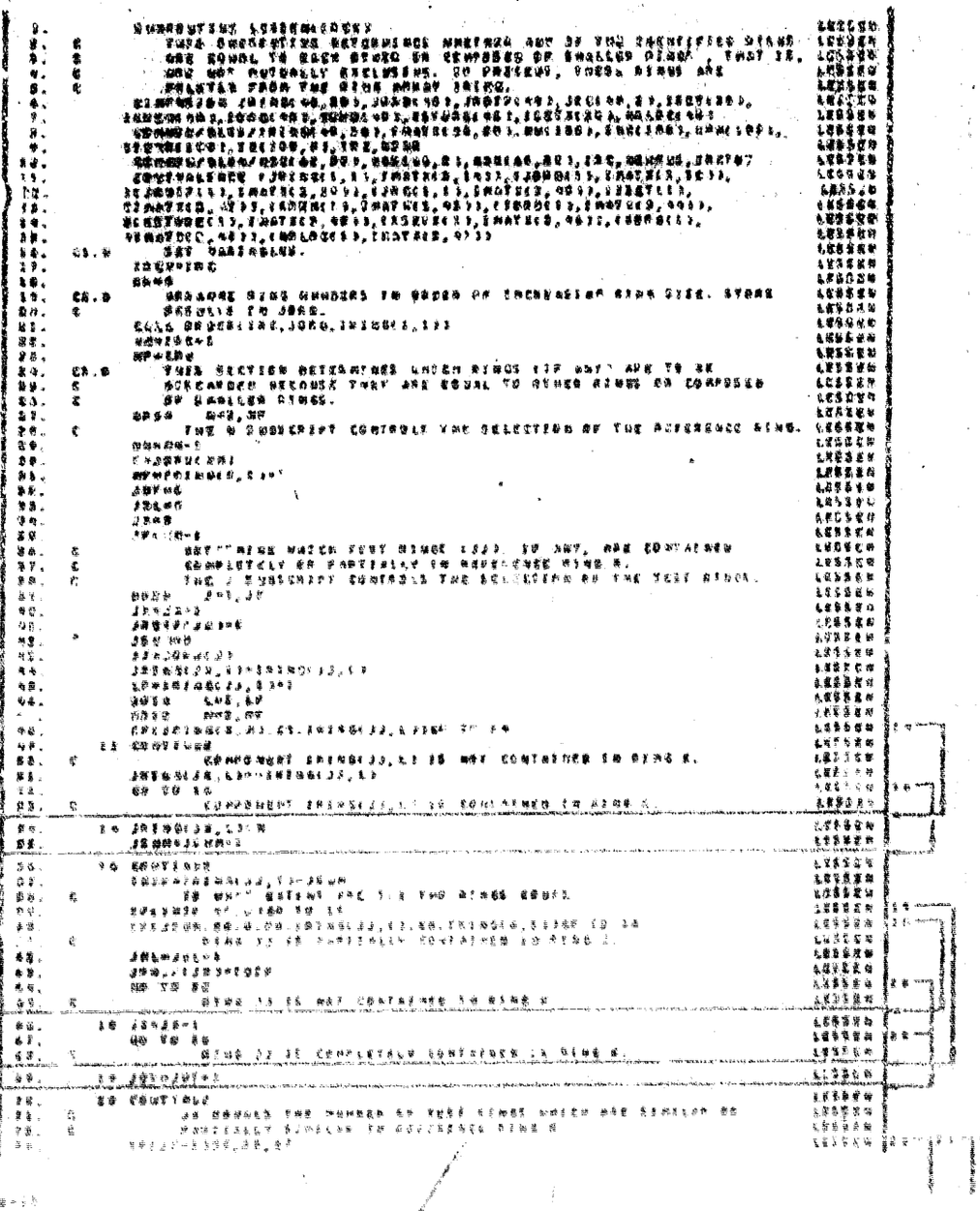
```
                GO TO(21,35,37),J1                                      IDENT
C               IF FIRST SYMBOL WAS NOT A LIBRARY SYMBOL(ICC=2), GO TO 25.  IDENT
      21 IF(ICC.EQ.2)GO TO 29                                           IDENT
C9.0     SYMBOL HAS BEEN IDENTIFIED. SUM ATOMS IN SYMBOL M INTO NUMATM  IDENT
C        AND EXIT.                                                      IDENT
      23 KP=1                                                           IDENT
         CALL SUMATM(M,KP,NUMAT(1))                                     IDENT
         RETURN                                                         IDENT
C           SET ERROR FLAG.                                             IDENT
      25 IERR=1                                                         IDENT
C           PRINT ERROR MESSAGE AND EXIT.                               IDENT
      27 WRITE(6,29)                                                    IDENT
      29 FORMAT(1HO,15X,54HERROR - INCORRECT SPECIES DESIGNATION.  CASE TER IDENT
     $MINATED.)                                                         IDENT
         RETURN                                                         IDENT
C9.0     SYMBOL CONTAINS AT LEAST TWO AND POSSIBLY THREE CHARACTERS.    IDENT
C        SEARCH FOR THIRD CHARACTER Z IN SYMBOL ZXV WHERE X, LOCATED    IDENT
C        AT K,L, IS FIRST CHARACTER OF SYMBOL IDENTIFIED.               IDENT
C          STORE FIRST CHARACTER IN IATOM(1) AND SECOND CHARACTER       IDENT
C          IN IATOM(2).                                                 IDENT
      31 IATOM(1)=GRID(K,L)                                             IDENT
         IATOM(2)=GRID(K1,L1)                                           IDENT
C           STORE AND SET PERTINENT VARIABLES AND RETURN TO SEARCH FOR  IDENT
C           THIRD CHARACTER.                                            IDENT
         K2=K1                                                          IDENT
         L2=L1                                                          IDENT
         KO2=KO                                                         IDENT
         LO2=LO                                                         IDENT
         NS=NX                                                          IDENT
         J2=2                                                           IDENT
         JS=MJ+1                                                        IDENT
         IF(JS.GT.7)GO TO 35                                            IDENT
         J1=2                                                           IDENT
         ICT=1                                                          IDENT
         GO TO 13                                                       IDENT
C9.0     SYMBOL IS NOT ZXV. NOW SEARCH FOR THIRD CHARACTER Z IN SYMBOL  IDENT
C        ZYZ WHERE Y IS LOCATED AT K,L.                                 IDENT
C          STORE AND SET PERTINENT VARIABLES AND RETURN TO SEARCH FOR   IDENT
C          THIRD CHARACTER.                                             IDENT
      35 JS=1                                                           IDENT
         KYM=K32                                                        IDENT
         LYM=L32                                                        IDENT
         ICT=2                                                          IDENT
         J1=3                                                           IDENT
         MI=MS+4                                                        IDENT
         I1=2                                                           IDENT
         GO TO 13                                                       IDENT
C4.0     SYMBOL COMPOSED OF TWO CHARACTERS. IDENTIFY SYMBOL AND PACK    IDENT
C        THE TWO CHARACTERS INTO ONE WORD.                              IDENT
C          IDENTIFY SYMBOL.                                             IDENT
      37 CALL MULTIE(ATOM,2,M,IERR)                                     IDENT
         IF(IERR.EQ.1)GO TO 27                                          IDENT
C           IF THIS IS FIRST SYMBOL OF MOLECULE TO BE IDENTIFIED, GO TO IDENT
C           39.                                                         IDENT
         IF(NUM.EQ.-3)GO TO 39                                          IDENT
C           IF BOND IS ATTACHED TO CHARACTER 1, GO TO 51.               IDENT
         IF(IBOND(1).EQ.1)GO TO 51                                      IDENT
C              CASE 1 - PACK TWO CHARACTERS INTO ONE WORD.              IDENT
      39 GRID(K+2,L32)=SYMBOL(M)                                        IDENT
         GRID(K,L)=SYMBOL(1)                                            IDENT
         K=K32                                                          IDENT
         L=L32                                                          IDENT
         NUM=NS                                                         IDENT
         GO TO 23                                                       IDENT
C              CASE 2 - PACK TWO CHARACTERS INTO ONE WORD.              IDENT
      51 ICT=0                                                          IDENT
         J2=K32                                                         IDENT
         L2=L32                                                         IDENT
         KO=KO2                                                         IDENT
         LO=LO2                                                         IDENT
CF.0     FINAL SETTINGS FOR TWO OR THREE CHARACTER CASES. STORE SYMBOL  IDENT
```

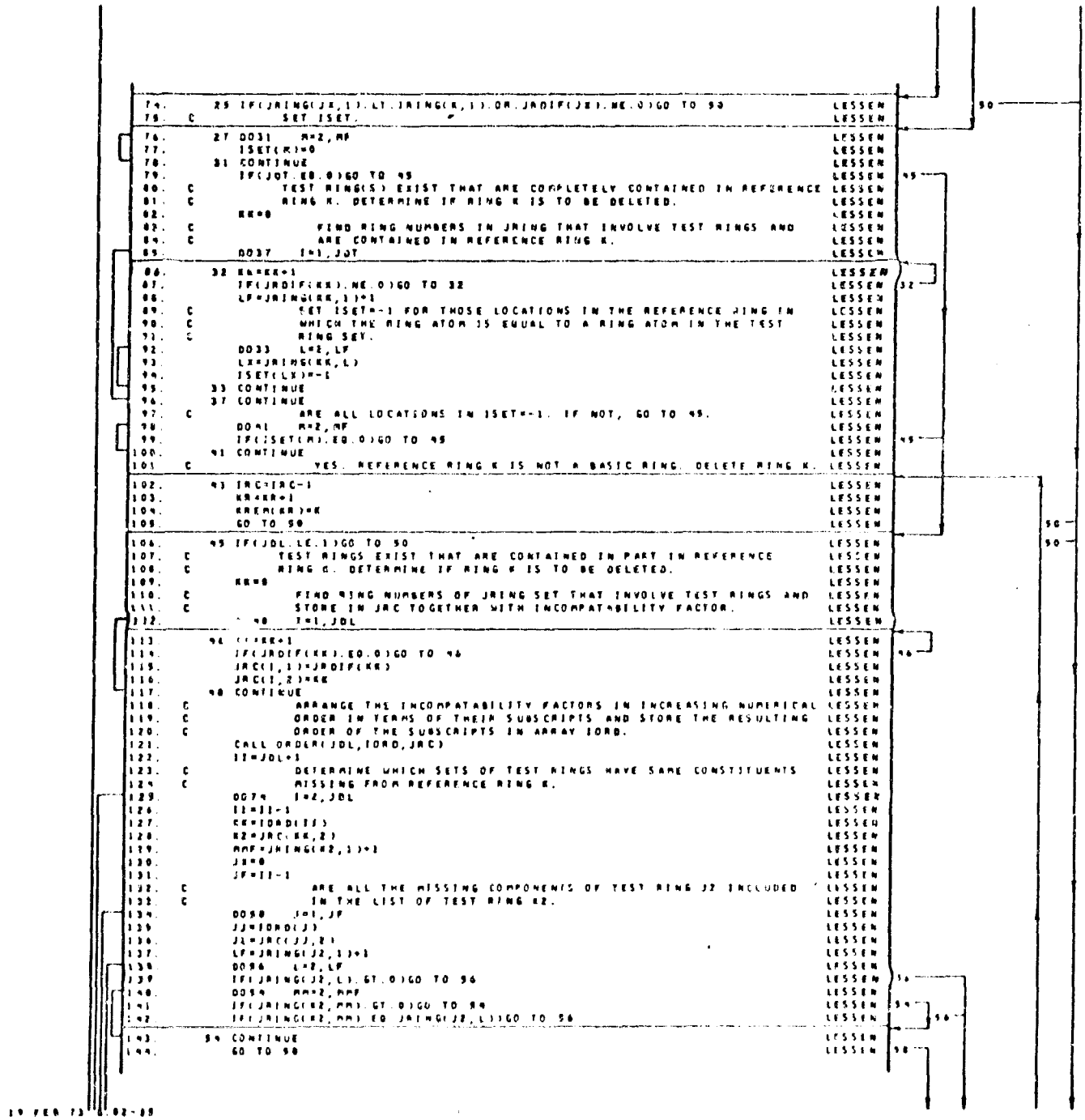


```
144.   C        AT K,L.                                                    IDENT
145.        58 GRID(K,L)=SYMBOL(M)                                        IDENT
146.           KZ=KZ+KD                                                   IDENT
147.           LZ=LZ+LD                                                   IDENT
148.           IF(KZ.LT.1.OR.KZ.GT.80)GO TO 59                            IDENT
149.           IF(LZ.GE.1.AND.LZ.LE.60)GO TO 67                           IDENT
150.   C            EXTENSION COORDINATE OUTSIDE GRID BOUNDARY. STORE BLANK IN  IDENT
151.   C            GRID(KZ,LZ) AND TRANSFER TO LOCATION DETERMINED BY ICT.    IDENT
152.        59 GRID(KZ,LZ)=SYM2(4)                                        IDENT
153.           IF(ICT-1)23,61,63                                          IDENT
154.   C            SYMBOL OF TYPE XYZ. SET LOCATION CONTAINING CHARACTER Y.   IDENT
155.        65 GRID(KZ2,LZ2)=GRID(KZ,LZ)                                  IDENT
156.           GO TO 23                                                   IDENT
157.   C            RESET THE LAST CHARACTER IDENTIFIED WHEN TWO OR THREE      IDENT
158.   C            CHARACTERS ARE PRESENT IN SYMBOL.                          IDENT
159.        67 DO69   MS=1,4                                              IDENT
160.           IF(GRID(KZ,LZ).NE.SYM2(MS))GO TO 69                        IDENT
161.           GRID(KZ,LZ)=GRID(KZ,LZ)                                    IDENT
162.           GO TO 63                                                   IDENT
163.        69 CONTINUE                                                   IDENT
164.   C            NO MATCH OBTAINED. ERROR EXIT.                             IDENT
165.           GO TO 25                                                   IDENT
166.   C0.0     SYMBOL COMPOSED OF THREE CHARACTERS  IDENTIFY SYMBOL AND   IDENT
167.   C        PACK THE THREE CHARACTERS INTO ONE WORD.                   IDENT
168.        71 IATOM(3)=GRID(KP,LK)                                       IDENT
169.   C            IDENTIFY SYMBOL.                                           IDENT
170.           CALL MULTY(IATOM,3,M,IERR)                                 IDENT
171.           IF(IERR.GT.0)GO TO 27                                      IDENT
172.   C            IF BOND IS ATTACHED TO CHARACTER 1, GO TO 55. OTHERWISE STORE  IDENT
173.   C            A BLANK IN K,L.                                            IDENT
174.           IF(ICTNB(1).EQ.1)GO TO 55                                  IDENT
175.           GRID(K,L)=SYM2(4)                                          IDENT
176.   C            BOND IS ATTACHED TO CHARACTER 2. REDEFINE REMAINING        IDENT
177.   C            CHARACTERS AND EXIT.                                       IDENT
178.           IF(ICTNB(2).EQ.0)GO TO 73                                  IDENT
179.           GRID(KZ2,LZ2)=SYMBOL(M)                                    IDENT
180.           GRID(KD,LD)=SYM2(4)                                        IDENT
181.           K=KZ2                                                      IDENT
182.           L=LZ2                                                      IDENT
183.           NUM=NS                                                     IDENT
184.           GO TO 23                                                   IDENT
185.   C            BOND IS NOT ATTACHED TO CHARACTER 2. REDEFINE REMAINING    IDENT
186.   C            CHARACTERS AND EXIT.                                       IDENT
187.        73 GRID(KD,LD)=SYMBOL(M)                                      IDENT
188.           GRID(KZ2,LZ2)=SYM2(4)                                      IDENT
189.           K=KZ2                                                      IDENT
190.           L=LZ2                                                      IDENT
191.           NUM=NS                                                     IDENT
192.           GO TO 23                                                   IDENT
193.           END                                                        IDENT
```

19 FEB 73 8.03.39

```
      SUBROUTINE INTROT(IGS,INR,ROTINS)                                 INTROT
C0.0      THIS SUBROUTINE DETERMINES WHICH CARBON ATOMS (IF ANY) HAVE    INTROT
C         THREEFOLD INTERNAL ROTATIONAL SYMMETRY AND HAVE NOT BEEN       INTROT
C         INCLUDED IN THE EXTERNAL ROTATIONAL SYMMETRY CONTRIBUTIONS.    INTROT
C         IT ALSO DETERMINES THE PRESENCE OF STRUCTURES WITH TWOFOLD     INTROT
C         INTERNAL ROTATIONAL SYMMETRY SUCH AS NO2 GROUPS AND CERTAIN    INTROT
C         MONOCYCLIC AROMATIC RINGS. IF ANY OF THESE ARE PRESENT, THE    INTROT
C         TOTAL INTERNAL ROTATION CONTRIBUTION TO THE ENTROPY IS COMPUTED INTROT
      INTEGER SYMX(9),SYMBOL(9),GRID(90,80)                             INTROT
      INTEGER WEIGHT(9)                                                 INTROT
      DIMENSION IKC(100),KINT(100),NTOTAL(3,150),KCGAUS(3,150),         INTROT
     1DATCIS(2,150),KTGAUS(150)                                         INTROT
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                      INTROT
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC     INTROT
      COMMON/BLK3/RING(40,30),IMATX(90,80),NW(100),IBC(100),KON(100),   INTROT
     1IDBR(100),IB(100,6),IRG,NOBR                                      INTROT
      EQUIVALENCE (IKC(1),GRID(1,79)),(INTMIN,GRID(4,76)),(KINT(1),     INTROT
     1GRID(5,76)),(NTOTAL(1,1),GRID(4,70)),(KCGAUS(1,1),IMATX(1,72)),   INTROT
     2(DATCIS(1,1),IMATX(1,66)),(KTGAUS(1),IMATX(1,60))                 INTROT
C1.0      INITIALIZE VARIABLES.                                         INTROT
      INR=0                                                             INTROT
      ROTINS=0.0                                                        INTROT
      KS=IGS                                                            INTROT
C2.0      DETERMINE IF THERE ARE CARBON ATOMS WITH A CONNECTIVITY OF FOUR INTROT
C         AND THREEFOLD+ ROTATIONAL SYMMETRY. IF SO, STORE GROUP NUMBER  INTROT
C         OF ATOM.                                                      INTROT
      DO2   K=1,KCC                                                     INTROT
      IF(IX(K,1,1).NE.2.OR.KON(K).NE.4.OR.NTOTAL(1,K).NE.3)GO TO 2      INTROT  2
      INR=INR+1                                                         INTROT
      IGS=IGS+1                                                         INTROT
      KCGAUS(1,IGS)=K                                                   INTROT
      NTOTAL(1,IGS)=1                                                   INTROT
      DATCIS(2,IGS)=-2.0323                                             INTROT
    2 CONTINUE                                                          INTROT
      IF(INR.EQ.0.OR.INTMIN.EQ.0)GO TO 10                               INTROT 10
C3.0      AFOREMENTIONED CARBON ATOMS ARE PRESENT AND SYMMETRY OF SOME OF INTROT
C         THESE ATOMS HAS BEEN INCLUDED IN THE EXTERNAL ROTATIONAL      INTROT
C         SYMMETRY. FIND THESE ATOMS AND DISCARD THEM.                  INTROT
      M=KS                                                              INTROT
      DO4   K=1,INR                                                     INTROT
      M=M+1                                                             INTROT
    4 IKC(K)=KCGAUS(1,M)                                                INTROT
      LL=0                                                              INTROT
      M=KS                                                              INTROT
      DO8   K=1,INR                                                     INTROT
      DO6   L=1,INTMIN                                                  INTROT
      IF(IKC(K).EQ.KINT(L))GO TO 8                                      INTROT  8
    6 CONTINUE                                                          INTROT
      LL=LL+1                                                           INTROT
      M=M+1                                                             INTROT
      KCGAUS(1,M)=IKC(K)                                                INTROT
    8 CONTINUE                                                          INTROT
      INR=LL                                                            INTROT
      IGS=KS+INR                                                        INTROT
C4.0      DETERMINE IF THERE ARE STRUCTURES WITH TWOFOLD INTERNAL        INTROT
C         ROTATIONAL SYMMETRY.                                          INTROT
   10 CALL CTWO(IGS,INR)                                                INTROT
      IF(INR.EQ.0)RETURN                                                INTROT
C5.0      COMPUTE TOTAL INTERNAL ROTATIONAL CONTRIBUTION.               INTROT
      M=KS                                                              INTROT
      DO26  K=1,INR                                                     INTROT
      M=M+1                                                             INTROT
      ROTINS=ROTINS+DATCIS(2,M)                                         INTROT
   26 CONTINUE                                                          INTROT
      RETURN                                                            INTROT
      END                                                               INTROT
```





```
      25 IF(JRING(J2,1).LT.JRING(K,1).OR.JRDIF(J2).NE.0)GO TO 90
C           SET ISET.
      27 DO31    M=2,MF
         ISET(K)=0
      31 CONTINUE
         IF(JDT.EQ.0)GO TO 95
C           TEST RING(S) EXIST THAT ARE COMPLETELY CONTAINED IN REFERENCE
C           RING K. DETERMINE IF RING K IS TO BE DELETED.
         KK=0
C                FIND RING NUMBERS IN JRING THAT INVOLVE TEST RINGS AND
C                ARE CONTAINED IN REFERENCE RING K.
         DO37    I=1,JDT
      32 KK=KK+1
         IF(JRDIF(KK).NE.0)GO TO 32
         LF=JRING(KK,1)+1
C              SET ISET=-1 FOR THOSE LOCATIONS IN THE REFERENCE JRING IN
C              WHICH THE RING ATOM IS EQUAL TO A RING ATOM IN THE TEST
C              RING SET.
         DO33    L=2,LF
         LX=JRING(KK,L)
         ISET(LX)=-1
      33 CONTINUE
      37 CONTINUE
C                ARE ALL LOCATIONS IN ISET=-1. IF NOT, GO TO 95.
         DO41    M=2,MF
         IF(ISET(M).EQ.0)GO TO 95
      41 CONTINUE
C                YES. REFERENCE RING K IS NOT A BASIC RING. DELETE RING K.
      43 IRC=IRC-1
         KR=KR+1
         KREM(KR)=K
         GO TO 90
      45 IF(JDL.LE.1)GO TO 50
C           TEST RINGS EXIST THAT ARE CONTAINED IN PART IN REFERENCE
C           RING G. DETERMINE IF RING K IS TO BE DELETED.
         KK=0
C              FIND RING NUMBERS OF JRING SET THAT INVOLVE TEST RINGS AND
C              STORE IN JRC TOGETHER WITH INCOMPATABILITY FACTOR.
         DO48    I=1,JDL
      46 KK=KK+1
         IF(JRDIF(KK).EQ.0)GO TO 46
         JRC(I,1)=JRDIF(KK)
         JRC(I,2)=KK
      48 CONTINUE
C              ARRANGE THE INCOMPATABILITY FACTORS IN INCREASING NUMERICAL
C              ORDER IN TERMS OF THEIR SUBSCRIPTS AND STORE THE RESULTING
C              ORDER OF THE SUBSCRIPTS IN ARRAY IORD.
         CALL ORDER(JDL,IORD,JRC)
         II=JDL+1
C              DETERMINE WHICH SETS OF TEST RINGS HAVE SAME CONSTITUENTS
C              MISSING FROM REFERENCE RING K.
         DO79    I=2,JDL
         II=II-1
         KK=IORD(II)
         K2=JRC(KK,2)
         MMF=JRING(K2,1)+1
         JJ=0
         JF=II-1
C                ARE ALL THE MISSING COMPONENTS OF TEST RING J2 INCLUDED
C                IN THE LIST OF TEST RING K2.
         DO58    J=1,JF
         JJ=IORD(J)
         JL=JRC(JJ,2)
         LF=JRING(J2,1)+1
         DO86    L=2,LF
         IF(JRING(J2,L).GT.0)GO TO 56
         DO59    MM=2,MMF
         IF(JRING(K2,MM).GT.0)GO TO 59
         IF(JRING(K2,MM).EQ.JRING(J2,L))GO TO 56
      59 CONTINUE
         GO TO 90
```

19 FEB 73 8:02:08

```
   56 CONTINUE                                                        LESSEN
C            AS THEY ARE. SET KSTORE.                                 LESSEN
      JX=JX+1                                                         LESSEN
      KSTORE(JX)=J2                                                   LESSEN
   58 CONTINUE                                                        LESSEN
      IF(JX-1)74,60,62                                                LESSEN
   60 IF(JRDIF(J2).NE.JRDIF(K2))GO TO 74                              LESSEN
C                TEST RINGS EXIST THAT HAVE ALL OR SOME OF THE SAME   LESSEN
C                CONSTITUENTS AS TEST RING K2 MISSING FROM REFERENCE  LESSEN
C                RING K.                                              LESSEN
   62 JX=JX+1                                                         LESSEN
      KSTORE(JX)=K2                                                   LESSEN
C                SET ISETX.                                           LESSEN
      DO64   M=2,MF                                                   LESSEN
      ISETX(M)=ISET(M)                                                LESSEN
   64 CONTINUE                                                        LESSEN
C                SET ISETX=-1 FOR THOSE LOCATIONS IN THE REFERENCE RING LESSEN
C                IN WHICH THE RING ATOM IS EQUAL TO A RING ATOM IN THE LESSEN
C                (PARTIAL) TEST RING SET JUST DEFINED.                LESSEN
      DO68   J=1,JX                                                   LESSEN
      JJ=KSTORE(J)                                                    LESSEN
      LF=JRING(JJ,1)+1                                                LESSEN
      DO66   L=2,LF                                                   LESSEN
      LX=JRING(JJ,L)                                                  LESSEN
      IF(LX.LT.0)GO TO 66                                             LESSEN
      ISETX(LX)=-1                                                    LESSEN
   66 CONTINUE                                                        LESSEN
   68 CONTINUE                                                        LESSEN
C                IF ALL ISETX LOCATIONS =-1, REFERENCE RING K SHALL BE LESSEN
C                DELETED. OTHERWISE, CONTINUE WITH TEST.              LESSEN
      DO70   M=2,MF                                                   LESSEN
      IF(ISETX(M).EQ.0)GO TO 74                                       LESSEN
   70 CONTINUE                                                        LESSEN
      GO TO 93                                                        LESSEN
   74 CONTINUE                                                        LESSEN
   53 CONTINUE                                                        LESSEN
      IF(IRCX.EQ.IXC)RETURN                                           LESSEN
C4.0     THIS SECTION DELETES THE RINGS THAT ARE EQUAL OR COMPOSITES. LESSEN
      KORD(1)=1                                                       LESSEN
C            DETERMINE THE ORDER THE SUBSCRIPTS OF THE RING NUMBERS IN LESSEN
C            KREM WOULD HAVE IF THE RING NUMBERS WERE ARRANGED IN     LESSEN
C            NUMERICAL ORDER.                                         LESSEN
      IF(KR.GT.1)CALL ORDER(KR,KORD,KREM)                             LESSEN
      J=0                                                             LESSEN
      KK=1                                                            LESSEN
      KX=KORD(KK)                                                     LESSEN
C            FIND RING NUMBER (J) CONTAINED IN KREM. SET JSTOP.       LESSEN
   80 J=J+1                                                           LESSEN
      IF(J.NE.KREM(KX))GO TO 80                                       LESSEN
      JSTOP=J                                                         LESSEN
C            INCREMENT J                                              LESSEN
   82 J=J+1                                                           LESSEN
C                IF J EXCEEDS INITIAL NUMBER OF RINGS, EXIT.          LESSEN
      IF(J.GT.IRCX)RETURN                                             LESSEN
   85 IF(J-KREM(KX))84,83,83                                          LESSEN
C            FIND NEW VALUE OF KX.                                    LESSEN
   83 KK=KK+1                                                         LESSEN
      IF(KK.GT.KR)GO TO 84                                            LESSEN
      KX=KORD(KK)                                                     LESSEN
      GO TO 85                                                        LESSEN
C            TRANSFER DATA FROM RING J TO RING JSTOP.                 LESSEN
   84 LF=IRING(J,1)+1                                                 LESSEN
      DO86   L=1,LF                                                   LESSEN
      IRING(JSTOP,L)=IRING(J,L)                                       LESSEN
   86 CONTINUE                                                        LESSEN
      NRLOC(JSTOP)=NRLOC(J)                                           LESSEN
      JSTOP=JSTOP+1                                                   LESSEN
      GO TO 82                                                        LESSEN
      END                                                             LESSEN
```

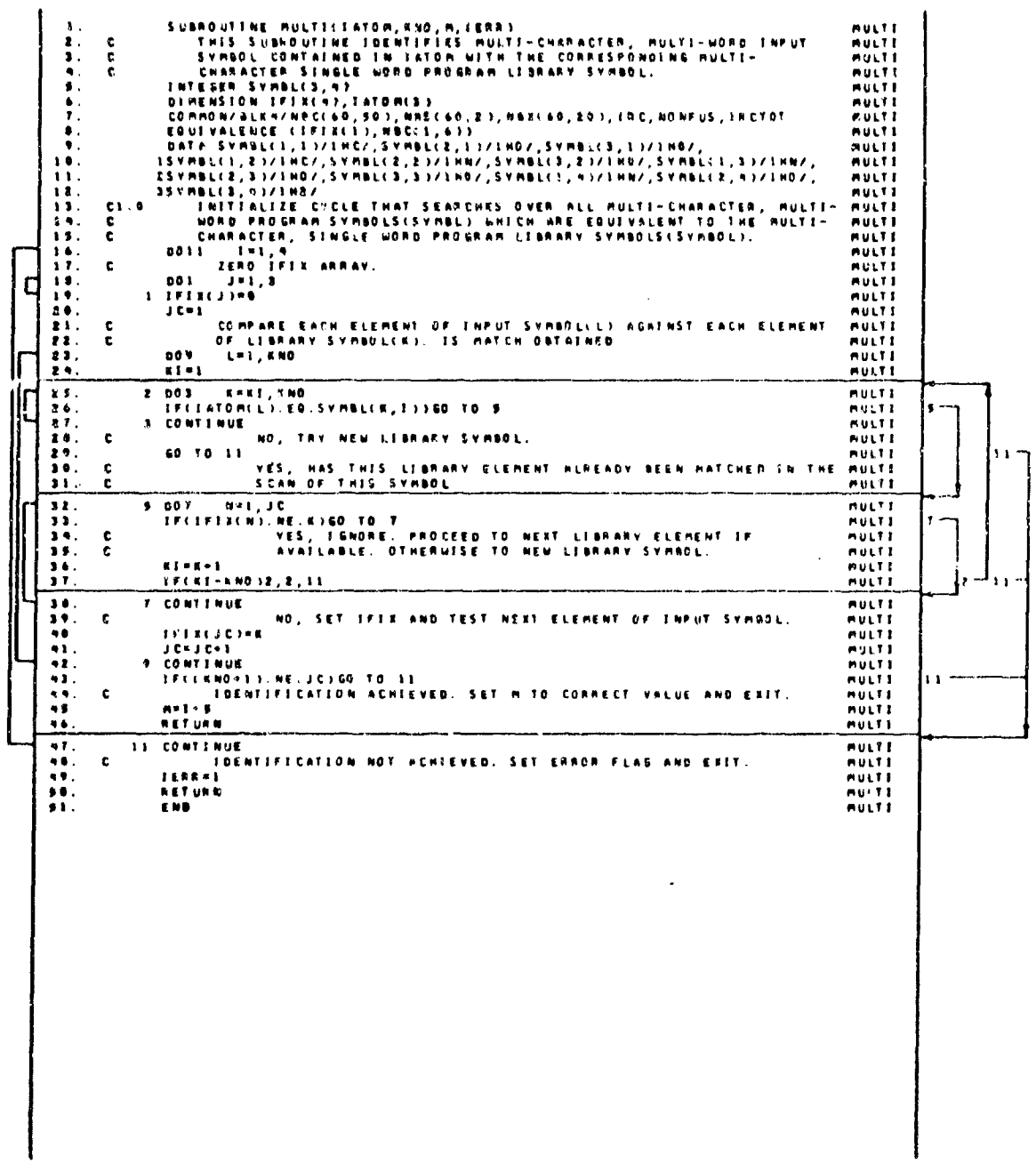
19 FEB 73 8 02-32

```
      SUBROUTINE LINEAR(KC,KCNEXT,LINE,KSYM)                                    LINEAR
C        THIS SUBROUTINE ESTABLISHES THE SYMMETRY PROPERTIES OF A               LINEAR
C        PARTICULAR LIGAND (KCNEXT) BONDED TO KC. IT DETERMINES THE             LINEAR
C        LINEARITY AND THE ROTATIONAL SYMMETRY OF THE LIGAND.                   LINEAR
      INTEGER SYMX(9),SYMBOL(9),GRID(96,98)                                     LINEAR
      INTEGER WEIGHT(9)                                                         LINEAR
      DIMENSION KCSAME(6,100),NTOTAL(3,100)                                     LINEAR
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                              LINEAR
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KC(             LINEAR
     COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KONC(100),          LINEAR
     1IOBR(100),IX(100,8),IRG,NOBR                                              LINEAR
      COMMON/BLK5/NOATM,NUMATM(5),MBC(50),MBS(2),JW,JV,LFLAGS,LFLAGG            LINEAR
      EQUIVALENCE (KCSAME(1,1),GRID(4,42)),(NTOTAL(1,1),GRID(4,70))             LINEAR
C1.0     INITIALIZE VARIABLES.                                                  LINEAR
      LINE=0                                                                    LINEAR
      KSYM=1                                                                    LINEAR
      K1=KC                                                                     LINEAR
      K2=KCNEXT                                                                 LINEAR
    3 IF(IBC(K2).EQ.0)GO TO 15                                                  LINEAR
C2.0     ATOM KCNEXT IS A RING ATOM. SET FLAG AND RETURN                        LINEAR
      JW=9                                                                      LINEAR
      RETURN                                                                    LINEAR
C3.0     ATOM KCNEXT IS NOT A RING ATOM.                                        LINEAR
   15 IF(IX(K2,1,1).NE.2.AND.IX(K2,1,1).NE.6)RETURN                             LINEAR
C4.0     ATOM IS EITHER CARBON OR CO. HOW MANY CORE ATOMS ARE BONDED            LINEAR
C        TO LIGAND ATOM                                                         LINEAR
      IF(NC(K2)-2)25,35,43                                                      LINEAR
C           ONE CORE ATOM.                                                     LINEAR
   25 IF(IX(K2,1,1).NE.3.OR.KON(K2).NE.2.OR.IX(K2,3,1).GE.8)GO TO 31            LINEAR
C           LIGAND IS CARBON ATOM AND HAS A CONNECTIVITY OF TWO.               LINEAR
   29 LINE=1                                                                    LINEAR
      RETURN                                                                    LINEAR
C              IF LIGAND IS CO AND HAS A CONNECTIVITY OF ONE, GO TO 29         LINEAR
   31 IF(IX(K2,1,1).EQ.6.AND.KON(K2).EQ.1)GO TO 29                              LINEAR
      IF(NTOTAL(1,K2).NE.KON(K2)-1)RETURN                                       LINEAR
C              THE NUMBER OF SYMMETRICAL GROUPS BONDED TO THE LIGAND           LINEAR
C              EQUALS ITS CONNECTIVITY MINUS ONE                               LINEAR
      KSYM=NTOTAL(1,K2)                                                         LINEAR
      RETURN                                                                    LINEAR
C           TWO CORE ATOMS.  IF THE CONNECTIVITY OF THE LIGAND EXCEEDS ITS     LINEAR
C           NUMBER OF CORE ATOMS, OR IF THE LIGAND IS CO, RETURN               LINEAR
   35 IF(KON(K2).GT.NC(K2).OR.IBC(K2,1,1).EQ.6)RETURN                           LINEAR
C              FIND THE NEXT ATOM DOWN THE CHAIN AND REPEAT TESTS              LINEAR
      IF(IX(K2,2,6).NE.K1)GO TO 37                                              LINEAR
      K1=K2                                                                     LINEAR
      K2=IX(K2,3,6)                                                             LINEAR
      GO TO 3                                                                   LINEAR
   37 K1=K2                                                                     LINEAR
      K2=IX(K2,2,6)                                                             LINEAR
      GO TO 3                                                                   LINEAR
C           THREE OR MORE CORE ATOMS                                           LINEAR
   43 NF=NC(K2)-1                                                               LINEAR
      IF(KON(K2).NE.NC(K2).OR.NTOTAL(3,K2).NE.NF)RETURN                         LINEAR
C              LIGAND HAS A CONNECTIVITY EQUAL TO ITS NUMBER OF CORE           LINEAR
C              LIGANDS (NC(KC)), AND IT HAS NC(KC)-1 IDENTICAL LIGANDS         LINEAR
C              ARE ANY OF THESE IDENTICAL LIGANDS EQUAL TO THE PREVIOUS        LINEAR
C              ATOM IN THE C-SIG- IF SO, RETURN. OTHERWISE, SET SYMMETRY       LINEAR
C              VALUE.                                                          LINEAR
      DO 47  K=1,NF                                                             LINEAR
      IF(KCSAME(K,K2).EQ.K1)RETURN                                              LINEAR
   47 CONTINUE                                                                  LINEAR
      KSYM=NTOTAL(1,K2)                                                         LINEAR
      RETURN                                                                    LINEAR
      END                                                                       LINEAR
```

```
      SUBROUTINE MAXCHN(LX,NASYMC)                                      MAXCHN
C        THIS SUBROUTINE FINDS THE LONGEST CHAIN IN THE MOLECULE AND     MAXCHN
C        PRINTS OUT THE GROUP NUMBERS OF THE CORE ATOM CONSTITUENTS OF   MAXCHN
C        THE CHAIN. IF ASYMMETRIC ATOMS ARE PRESENT, IT ALSO DETERMINES  MAXCHN
C        IF ANOTHER CHAIN OF THE SAME (MAXIMUM) LENGTH IS PRESENT WHICH  MAXCHN
C        CONTAINS MORE ASYMMETRIC ATOMS THAN THE FIRST MAXIMUM CHAIN     MAXCHN
C        DETECTED. IF SO, A NEW MAXIMUM CHAIN IS DEFINED.                MAXCHN
      DIMENSION IOPATM(100)                                             MAXCHN
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),  MAXCHN
     1IOBR(100),IB(100,8),IRB,NOBR                                      MAXCHN
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT     MAXCHN
      COMMON/BLK5/NOATM,NUMATM(5),MBC(50),MBS(2),JW,JV,LFLAGS,LFLAGB    MAXCHN
      EQUIVALENCE (IOPATM(1),IMATX(2,45))                               MAXCHN
C1.0     SET VARIABLES.                                                 MAXCHN
      KSAME=1                                                           MAXCHN
      LV=1                                                              MAXCHN
      KMAX=NBS(1,1)                                                     MAXCHN
C2.0     FIND THE LONGEST CHAIN IN THE MOLECULE.                        MAXCHN
      IF(LX.LE.1)GO TO 12                                               MAXCHN
      DO11   K=2,LX                                                     MAXCHN
      IF(NBS(K,1)-KMAX)11,9,7                                           MAXCHN
    9 KSAME=KSAME+1                                                     MAXCHN
      GO TO 11                                                          MAXCHN
    7 KSAME=1                                                           MAXCHN
      KMAX=NBS(K,1)                                                     MAXCHN
      LV=K                                                              MAXCHN
   11 CONTINUE                                                          MAXCHN
   12 LVP=LV                                                            MAXCHN
C3.0     IF ASYMMETRIC ATOMS ARE PRESENT, DETERMINE IF ANOTHER CHAIN OF MAXCHN
C        THE SAME (MAXIMUM) LENGTH IS PRESENT WHICH CONTAINS MORE       MAXCHN
C        ASYMMETRIC ATOMS. IF SO, UTILIZE THIS ONE AS THE MAXIMUM CHAIN. MAXCHN
      IF(NASYMC.EQ.0.OR.KSAME.EQ.1)GO TO 35                             MAXCHN
C           ONE OR MORE ASYMMETRIC ATOMS ARE PRESENT.                   MAXCHN
      KOPMAX=0                                                          MAXCHN
      KI=LV+1                                                           MAXCHN
   13 KOP=0                                                             MAXCHN
C           FIND HOW MANY ASYMMETRIC ATOMS (KOP) ARE PRESENT IN MAXIMUM MAXCHN
C           CHAIN LV.                                                   MAXCHN
      DO17   L=1,NASYMC                                                 MAXCHN
      DO15   K=1,KMAX                                                   MAXCHN
      IF(IOPATM(L).NE.NBC(LV,K))GO TO 15                                MAXCHN
      KOP=KOP+1                                                         MAXCHN
      GO TO 17                                                          MAXCHN
   15 CONTINUE                                                          MAXCHN
   17 CONTINUE                                                          MAXCHN
      IF(KOP.EQ.NASYMC)GO TO 35                                         MAXCHN
C           KOP IS LESS THAN TOTAL NUMBER OF ASYMMETRIC ATOMS PRESENT.  MAXCHN
C           CONTINUE SEARCH.                                            MAXCHN
      IF(KOP.LE.KOPMAX)GO TO 21                                         MAXCHN
C           KOP IS GREATER THAN PREVIOUS MAXIMUM NUMBER OF ASYMMETRIC   MAXCHN
C           ATOMS CONTAINED IN CHAIN. DEFINE NEW MAXIMUM CHAIN.         MAXCHN
      KOPMAX=KOP                                                        MAXCHN
      LVP=LV                                                            MAXCHN
   21 IF(KI.GT.LX)GO TO 29                                              MAXCHN
C           NOT ALL CHAINS HAVE BEEN TESTED. FIND NEW CHAIN TO TEST THAT MAXCHN
C           CONTAINS A MAXIMUM NUMBER OF CORE ATOM CONSTITUENTS.        MAXCHN
      DO25   L=KI,LX                                                   MAXCHN
      IF(NBS(L,1).GE.KMAX)GO TO 27                                      MAXCHN
   25 CONTINUE                                                          MAXCHN
      GO TO 29                                                          MAXCHN
   27 LV=L                                                              MAXCHN
      KI=L+1                                                            MAXCHN
      GO TO 13                                                          MAXCHN
   29 JV=0                                                              MAXCHN
C.0      STORE GROUP NUMBERS OF LONGEST CHAIN IN MBC. STORE OTHER       MAXCHN
C        PERTINENT CHAIN DATA IN MBS.                                   MAXCHN
   35 DO37   K=1,KMAX                                                   MAXCHN
   37 MBC(K)=NBC(LVP,K)                                                 MAXCHN
      MBS(1)=KMAX                                                       MAXCHN
      MBS(2)=LVP                                                        MAXCHN
```

```
C5.0      PRINT OUT THE TOTAL NUMBER OF CORE ATOM CONSTITUENTS IN THE      MAXCHN
C         CHAIN AND THEIR GROUP NUMBERS.                                   MAXCHN
      WRITE(6,41)KMAX                                                      MAXCHN
   41 FORMAT(/// 1H ,20X,42HNUMBER OF CORE ATOMS IN LONGEST CHAIN  =  ,    MAXCHN
     1I5)                                                                  MAXCHN
      WRITE(6,45)(MBC(M),M=1,KMAX)                                         MAXCHN
   45 FORMAT(1H0, 7X,55HGROUP NUMBER OF CORE CONSTITUENTS OF LONGEST CHA   MAXCHN
     1N  =  ,10(I5,1H,)/(1H ,62X,10(I5,1H,)))                             MAXCHN
      RETURN                                                               MAXCHN
      END                                                                  MAXCHN
```

17 APR 73 8.02-36



```
      SUBROUTINE MULTI(IATOM,KNO,M,IERR)                                    MULTI
C        THIS SUBROUTINE IDENTIFIES MULTI-CHARACTER, MULTI-WORD INPUT       MULTI
C        SYMBOL CONTAINED IN IATOM WITH THE CORRESPONDING MULTI-            MULTI
C        CHARACTER SINGLE WORD PROGRAM LIBRARY SYMBOL.                      MULTI
      INTEGER SYMBL(3,9)                                                    MULTI
      DIMENSION IFIX(9),IATOM(3)                                            MULTI
      COMMON/BLK1/NPC(60,50),NRE(60,2),NBX(60,20),IRC,NONFUS,IRCYOT         MULTI
      EQUIVALENCE (IFIX(1),NBC(1,6))                                        MULTI
      DATA SYMBL(1,1)/1HC/,SYMBL(2,1)/1HO/,SYMBL(3,1)/1HO/,                 MULTI
     1SYMBL(1,2)/1HC/,SYMBL(2,2)/1HN/,SYMBL(3,2)/1HO/,SYMBL(1,3)/1HN/,      MULTI
     2SYMBL(2,3)/1HO/,SYMBL(3,3)/1HO/,SYMBL(1,4)/1HN/,SYMBL(2,4)/1HO/,      MULTI
     3SYMBL(3,4)/1HB/                                                       MULTI
C1.0     INITIALIZE CYCLE THAT SEARCHES OVER ALL MULTI-CHARACTER, MULTI-    MULTI
C        WORD PROGRAM SYMBOLS(SYMBL) WHICH ARE EQUIVALENT TO THE MULTI-     MULTI
C        CHARACTER, SINGLE WORD PROGRAM LIBRARY SYMBOLS(SYMBOL).            MULTI
      DO11   I=1,9                                                          MULTI
C           ZERO IFIX ARRAY.                                                MULTI
      DO1    J=1,3                                                          MULTI
    1 IFIX(J)=0                                                             MULTI
      JC=1                                                                  MULTI
C           COMPARE EACH ELEMENT OF INPUT SYMBOL(L) AGAINST EACH ELEMENT    MULTI
C           OF LIBRARY SYMBOL(K). IS MATCH OBTAINED                         MULTI
      DO9    L=1,KNO                                                        MULTI
      KI=1                                                                  MULTI
    2 DO3    K=KI,4NO                                                       MULTI
      IF(IATOM(L).EQ.SYMBL(K,I))GO TO 5                                     MULTI
    3 CONTINUE                                                              MULTI
C                NO, TRY NEW LIBRARY SYMBOL.                                MULTI
      GO TO 11                                                              MULTI
C                YES, HAS THIS LIBRARY ELEMENT ALREADY BEEN MATCHED IN THE  MULTI
C                SCAN OF THIS SYMBOL                                        MULTI
    5 DO7    N=1,JC                                                         MULTI
      IF(IFIX(N).NE.K)GO TO 7                                               MULTI
C                   YES, IGNORE. PROCEED TO NEXT LIBRARY ELEMENT IF         MULTI
C                   AVAILABLE. OTHERWISE TO NEW LIBRARY SYMBOL.             MULTI
      KI=K+1                                                                MULTI
      IF(KI-KNO)2,2,11                                                      MULTI
    7 CONTINUE                                                              MULTI
C                NO, SET IFIX AND TEST NEXT ELEMENT OF INPUT SYMBOL.        MULTI
      IFIX(JC)=K                                                            MULTI
      JC=JC+1                                                               MULTI
    9 CONTINUE                                                              MULTI
      IF((KNO+1).NE.JC)GO TO 11                                             MULTI
C           IDENTIFICATION ACHIEVED. SET M TO CORRECT VALUE AND EXIT.       MULTI
      M=I-8                                                                 MULTI
      RETURN                                                                MULTI
   11 CONTINUE                                                              MULTI
C           IDENTIFICATION NOT ACHIEVED. SET ERROR FLAG AND EXIT.           MULTI
      IERR=1                                                                MULTI
      RETURN                                                                MULTI
      END                                                                   MULTI
```

19 FEB 73 8.02-35

NEWCOL

```
      SUBROUTINE NEWCOL(LX,LXL,NBP,IZ,KC,KCPV,LM,IERR)                  NEWCOL
C        THIS SUBROUTINE DEFINES THE INITIAL CONSTITUENTS OF A NEW      NEWCOL
C        CHAIN.  THE GROUP NUMBERS OF THE PREVIOUS CHAIN COMMON TO THE  NEWCOL
C        CHAIN ARE STORED IN A NEW ROW OF ARRAY NBC.  THE RELATED BRANCH NEWCOL
C        DATA FOR THE NEW CHAIN ARE STORED IN ARRAY NBX.                NEWCOL
      INTEGER WEIGHT(9)                                                 NEWCOL
      DIMENSION JUNCT(5,100),NOBRD(100)                                 NEWCOL
      COMMON/BLK2/WEIGHT,AWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC     NEWCOL
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),  NEWCOL
     1IOBR(100),IB(100,8),IRG,NOBR                                      NEWCOL
      COMMON/BLK4/NBC(60,50),NBS(40,2),NBX(60,20),IRC,NONFUS,IRCTOT     NEWCOL
      EQUIVALENCE (JUNCT(1,1),IMATX(3,1)),(NOBRD(1),IMATX(3,11))        NEWCOL
C1.0     FIND CORE ATOM IN CHAIN FORMATION WHICH IS ALSO A BRANCH POINT. NEWCOL
C        DETERMINE WHETHER THIS ATOM STILL HAS LIGANDS WHICH HAVE NOT   NEWCOL
C        YET BEEN INCORPORATED IN A CHAIN LINE.                         NEWCOL
      LF=NBP                                                            NEWCOL
      DO3    LL=1,LF                                                    NEWCOL
      LM=NBX(LX,NBP)                                                    NEWCOL
      KC=NBC(LX,LM)                                                     NEWCOL
      IFLAG1=0                                                          NEWCOL
      LV=LM-1                                                           NEWCOL
      IF(LV.GT.0)GO TO 1                                                NEWCOL  1
      IFLAG1=1                                                          NEWCOL
      GO TO 2                                                           NEWCOL  2
    1 KCPV=NBC(LX,LV)                                                   NEWCOL
C           FIND THE TENTATIVE BRANCH ATOM KC.                          NEWCOL
    2 CALL NEWKC(KCPV,KC,IFLAG1,IFLAG2)                                 NEWCOL
C           HAS KC ALREADY BEEN PROCESSED  IF NOT, GO TO 5.  OTHERWISE, NEWCOL
C           FIND NEW BRANCH ATOM.                                       NEWCOL
      IF(IFLAG2.EQ.0)GO TO 5                                            NEWCOL  5
      NBP=NBP-1                                                         NEWCOL
    3 CONTINUE                                                          NEWCOL
      LX=LXL                                                            NEWCOL
      RETURN                                                            NEWCOL
C3.0     RESET TO THEIR INITIAL VALUE OF 1 THE LOCATIONS OF NOBRD       NEWCOL
C        ASSOCIATED WITH THOSE BRANCH ATOMS WHICH FOLLOW KC.            NEWCOL
    5 LF=NBS(LX,2)                                                      NEWCOL
      DO6    LL=1,LF                                                    NEWCOL
      IF(NBX(LX,LL).LE.LM)GO TO 6                                       NEWCOL  6
      J=NBX(LX,LL)                                                      NEWCOL
      KCX=NBC(LX,J)                                                     NEWCOL
      NOBRD(KCX)=1                                                      NEWCOL
    6 CONTINUE                                                          NEWCOL
C3.0     COPY GROUP NUMBERS IN NBC(LX,J) INTO NBC(LX+1,J) UP TO         NEWCOL
C        DESIGNATED BRANCH POINT.  ALSO COPY NBX ARRAY.                 NEWCOL
      LV=LXL+1                                                          NEWCOL
      IF(LV.LE.60)GO TO 17                                              NEWCOL  17
      IERR=1                                                            NEWCOL
      LX=LV                                                             NEWCOL
      RETURN                                                            NEWCOL
   17 DO19    J=1,LM                                                    NEWCOL
   19 NBC(LV,J)=NBC(LX,J)                                               NEWCOL
      DO21    J=1,NBP                                                   NEWCOL
   21 NBX(LV,J)=NBX(LX,J)                                               NEWCOL
      LX=LV                                                             NEWCOL
      IZ=LM                                                             NEWCOL
      RETURN                                                            NEWCOL
      END                                                               NEWCOL
```

19 FEB 73 8.02.38

NEWKC

```
  1.          SUBROUTINE NEWKC(KCPV,KC,IFLAG1,IFLAG2)                          NEWKC
  2.    C        THIS SUBROUTINE FINDS THE NEXT CHAIN ATOM WHEN THE LAST       NEWKC
  3.    C        IDENTIFIED ATOM (KC) IN THE CHAIN IS A BRANCH ATOM. THE NEW   NEWKC
  4.    C        CHAIN ATOM IS SET EQUAL TO ONE OF THE UNUSED CORE ATOM LIGANDS NEWKC
  5.    C        OF KC AND IS STORED IN KC. IF ALL OF THESE LIGANDS HAVE       NEWKC
  6.    C        ALREADY BEEN INCORPORATED IN SOME CHAIN LINK, THE PROGRAM EXITS NEWKC
  7.    C        FROM THE SUBROUTINE UNLESS KCPV IS NOT STORED IN THE REQUIRED NEWKC
  8.    C        ORDER.                                                        NEWKC
  9.          INTEGER WEIGHT(9)                                               NEWKC
 10.          DIMENSION JUNCT(5,100),NOBRD(100)                               NEWKC
 11.          COMMON/BLK2/WEIGHT,NWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC   NEWKC
 12.          COMMON/BLK3/IRING(10,30),IMATX(50,80),NW(100),IBC(100),KOM(100),NEWKC
 13.         1IOBR(100),IB(100,8),IFG,NOBR                                    NEWKC
 14.          EQUIVALENCE (JUNCT(1,1),IMATX(3,1)),(NOBRD(1),IMATX(3,11))      NEWKC
 15.    C1.0     SET IFLAG2 AND INCREMENT NOBRD.                              NEWKC
 16.          IFLAG2=0                                                        NEWKC
 17.          NOBRD(KC)=NOBRD(KC)+1                                           NEWKC
 18.    C2.0     IS IFLAG1 EQUAL TO ONE                                       NEWKC
 19.          IF(IFLAG1.NE.1)GO TO 1                                          NEWKC
 20.    C           YES. IF ALL LIGANDS OF ATOM KC HAVE BEEN USED, GO TO 2.   NEWKC
 21.    C           OTHERWISE, GO TO 9.                                       NEWKC
 22.          IF(NOBRD(KC)-NC(KC))9,9,2                                       NEWKC
 23.    C           NO, IT IS NOT.                                            NEWKC
 24.    C3.0     ALL CORE ATOM LIGANDS OF BRANCH ATOM KC MAY HAVE BEEN USED.  NEWKC
 25.        1 IF(NOBRD(KC).LE.NC(KC))GO TO 4                                  NEWKC
 26.    C           ALL CORE ATOM LIGANDS OF BRANCH ATOM KC HAVE BEEN USED.   NEWKC
 27.          IF(JUNCT(1,KC).NE.KCPV)GO TO 3                                  NEWKC
 28.    C           KCPV IS IN CORRECT LOCATION. SET IFLAG2 AND EXIT.         NEWKC
 29.        2 IFLAG2=1                                                        NEWKC
 30.          RETURN                                                          NEWKC
 31.    C           KCPV IS NOT IN REQUIRED POSITION. RESET NOBRD.            NEWKC
 32.        3 NOBRD(KC)=2                                                     NEWKC
 33.    C           IF CONDITIONS REQUIRE IT, STORE VALUE OF KCPV IN JUNCT(1,KC) NEWKC
 34.    C           SWITCHING LOCATIONS WITH GROUP NUMBER PREVIOUSLY THERE.   NEWKC
 35.        4 JF=NC(KC)                                                       NEWKC
 36.          DO 5   J=1,JF                                                   NEWKC
 37.          IF(JUNCT(J,KC).EQ.KCPV)GO TO 7                                  NEWKC
 38.        5 CONTINUE                                                        NEWKC
 39.        7 IF(J.LT.NOBRD(KC))GO TO 9                                       NEWKC
 40.          ITEMP=JUNCT(J,KC)                                               NEWKC
 41.          JUNCT(J,KC)=JUNCT(1,KC)                                         NEWKC
 42.          JUNCT(1,KC)=ITEMP                                               NEWKC
 43.    C4.0     SET KCPV EQUAL TO CURRENT KC. THEN SET KC EQUAL TO ONE OF THE NEWKC
 44.    C           STILL UNUSED CORE ATOM LIGANDS BONDED TO BRANCH ATOM KC   NEWKC
 45.        9 KCPV=KC                                                         NEWKC
 46.          ITEMP=NOBRD(KC)                                                 NEWKC
 47.          KC=JUNCT(ITEMP,KC)                                              NEWKC
 48.          RETURN                                                          NEWKC
 49.          END                                                             NEWKC
```

```
      SUBROUTINE NRINGS(IGS,IRNG3,RDATA,HRING,SRING,CPSYM)                    NRINGS
C        THIS SUBROUTINE DETERMINES WHETHER THE EXISTING SIX-MEMBERED         NRINGS
C        NITROGEN-CONTAINING RINGS FORM PART OF THE TRICYCLIC FUSED RING      NRINGS
C        SYSTEM N(2)C(6)H(12), WHERE EACH RING IS FUSED AND HAS 4 POINTS      NRINGS
C        IN COMMON WITH EACH OF OTHER TWO RINGS. IF PRESENT IT ADDS           NRINGS
C        RING CORRECTIONS TO THERMODYNAMIC PROPERTIES.                        NRINGS
      INTEGER WEIGHT(9)                                                       NRINGS
      DIMENSION KOMB(40,10),NOKOMB(11),NOK(10,10),KRPROP(6,40),               NRINGS
     1JSTORE(3),RDATA(6),CPSYM(9),KTGAUS(150),KCGAUS(3,150)                   NRINGS
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC           NRINGS
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),        NRINGS
     1IOBR(100),IB(100,8),IRG,NOBR                                            NRINGS
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT           NRINGS
      EQUIVALENCE (KOMB(1,1),IB(62,4)),(NOKOMB(1),IB(80,8)),(JSTORE(1),       NRINGS
     1IB(98,8)),(NOK(1,1),IMATX(1,55)),(KRPROP(1,1),NBX(2,1)),                NRINGS
     2(KTGAUS(1),IMATX(1,60)),(KCGAUS(1,1),IMATX(1,72))                       NRINGS
C1.0     INITIALIZE CYCLE THAT TESTS EACH FUSED RING SET.                     NRINGS
      IC=NOKOMB(1)                                                            NRINGS
C2.0     HOW MANY RINGS ARE THERE IN FUSED RING SET J                         NRINGS
      DO7   J=1,IC                                                            NRINGS
      KV=NOKOMB(J+1)                                                          NRINGS
C           IF NUMBER UNEQUAL TO 3 IGNORE SET J.                              NRINGS
      IF(KV.NE.3)GO TO 7                                                      NRINGS  7
C             SET J COMPOSED OF 3 RINGS.                                      NRINGS
      IF(NOK(1,J).NE.3.OR.NOK(2,J).NE.4.OR.NOK(3,J).NE.2)GO TO 7              NRINGS  7
C             EACH RING IS FUSED AT 4 POINTS TO EACH OF THE OTHER TWO         NRINGS
C             RINGS AND TWO OF THESE FOUR ATOMS ARE COMMON TO ALL THREE       NRINGS
C             RINGS                                                           NRINGS
      K1=NOK(4,J)                                                             NRINGS
      K2=NOK(5,J)                                                             NRINGS
      IF(IX(K1,1,1).NE.4.OR.IX(K2,1,1).NE.4)GO TO 7                           NRINGS  7
C           THE TWO JUNCTURE ATOMS ARE NITROGEN.                              NRINGS
      NX3=0                                                                   NRINGS
C3.0     DETERMINE WHETHER THE 3 RINGS IN SET J ARE EACH 6-CENTERED           NRINGS
C        CARBON-NITROGEN RINGS WITH NO DOUBLE BONDS.                          NRINGS
      DO1   K=1,KV                                                            NRINGS
      I=KOMB(K,J)                                                             NRINGS
      IF(IRING(I,1).NE.6)GO TO 7                                              NRINGS  7
C             RING I IS 6-CENTERED.                                           NRINGS
      IF(KRPROP(1,I)+KRPROP(5,I).NE.0.OR.KRPROP(3,I).NE.2)GO TO 7             NRINGS  7
C           RING I COMPOSED ONLY OF 4 CARBON AND 2 NITROGEN ATOMS WITH        NRINGS
C           NO DOUBLE BONDS. STORE I.D. NUMBER OF RING INVOLVED.              NRINGS
      NX3=NX3+1                                                               NRINGS
      JSTORE(NX3)=I                                                           NRINGS
    1 CONTINUE                                                                NRINGS
C4.0     DETERMINE SEPARATION OF TWO NITROGEN ATOMS.                          NRINGS
      K2=KRPROP(4,I)+3                                                        NRINGS
      K3=IRING(I,K2)                                                          NRINGS
      IF(IX(K3,1,1).NE.4)GO TO 7                                              NRINGS  7
C             THE TWO NITROGEN ATOMS ARE IN THE 1 AND 4 POSITIONS.            NRINGS
C5.0     SET J IS THE TRICYCLIC FUSED RING SYSTEM N(2)C(6)H(12). ADD          NRINGS
C        CORRECTIONS TO THERMODYNAMIC PROPERTIES, STORE PERTINENT             NRINGS
C        PRINTOUT DATA, AND CONTINUE THE TEST OF OTHER FUSED RING SETS.       NRINGS
      HRING=HRING+RDATA(1)                                                    NRINGS
      SRING=SRING+RDATA(2)                                                    NRINGS
      DO3   KK=1,9                                                            NRINGS
      CPSYM(KK)=CPSYM(KK)+RDATA(KK+2)                                         NRINGS
    3 CONTINUE                                                                NRINGS
      IRNG3=IRNG3+1                                                           NRINGS
      IGS=IGS+1                                                               NRINGS
      KTGAUS(IGS)=1                                                           NRINGS
      DO5   KK=1,3                                                            NRINGS
    5 KCGAUS(KK,IGS)=JSTORE(KK)                                               NRINGS
    7 CONTINUE                                                                NRINGS
      RETURN                                                                  NRINGS
      END                                                                     NRINGS
```

17 APR 73 0.02-36

```
      SUBROUTINE NUMBER(NUM,NXX,K,L,MP,KXX,LXX,KOD,LOD,JII,JFF)
C        THIS SUBROUTINE DETERMINES THE NUMBER OF LIGANDS(MP) ATTACHED
C        TO CORE ATOM NUMBER ONE OR THE NUMBER OF LIGANDS MINUS ONE OF
C        ANY CORE ATOM WHOSE GROUP NUMBER IS GREATER THAN ONE.
      INTEGER SYMX(9),SYMBOL(9),GRID(50,60)
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID
C1.0     INITIALIZE MP AND FIND THE REVERSE OF DIRECTION NUM.
      MP=0
      NUM=NUM+9
      IF(NUM.GT.8)NUM=NUM-8
C2.0     INITIALIZE CYCLE THAT FINDS THE NUMBER OF LIGANDS.
      DO1   J=1,7
C           COMPUTE BOND DIRECTION TO BE TESTED.
      NX=NUM+J
      IF(NX.GT.8)NX=NX-8
C           FIND NEW TRANSFORMATION COORDINATES FOR THIS BOND DIRECTION.
      CALL ASSIGN(NX,K,L,KB,LB,JI,JF)
      IF(JI.GT.JF)GO TO 1
      KX=K+KB
      LX=L+LB
C           DOES THIS LOCATION HAVE A NON-BLANK CHARACTER
      IF(GRID(KX,LX).EQ.SYMX(9))GO TO 1
C             YES, IT DOES. INCREMENT MP.
      MP=MP+1
C               IS THIS THE FIRST LIGAND IDENTIFIED
      IF(MP.GT.1)GO TO 1
C                 YES, IT IS. SET ALL IDENTIFYING VARIABLES FOR THIS
C                 LIGAND. CONTINUE SEARCH.
      NXX=NX
      KXX=KX
      LXX=LX
      KOD=KB
      LOD=LB
      JII=JI
      JFF=JF
    1 CONTINUE
      RETURN
      END
```

19 FEB 73 8.02-39

```
 1.          SUBROUTINE ORDER(LIM,IA,IB)                                    ORDER
 2.  C          THIS SUBROUTINE STORES IN ARRAY IA THE ORDER THAT THE LIM    ORDER
 3.  C          ELEMENTS OF ARRAY IB WOULD HAVE IF ARRANGED IN ORDER OF      ORDER
 4.  C          ASCENDING VALUE.                                             ORDER
 5.          DIMENSION IA(100),IB(100)                                      ORDER
 6.  C1.0       STORE INITIAL ORDER OF ELEMENTS IN IA.                       ORDER
 7.          DO1    J=1,LIM                                                 ORDER
 8.          IA(J)=J                                                        ORDER
 9.        1 CONTINUE                                                       ORDER
10.          IF(LIM.EQ.1)RETURN                                             ORDER
11.  C2.        NOW THAN ONE ELEMENT CONTAINED IN IB. INITIALIZE CYCLE.     ORDER
12.          JF=LIM-1                                                       ORDER
13.          DO9    J=1,JF                                                  ORDER
14.  C            CHECK ORDER OF TWO ADJACENT ELEMENTS IN IA, NAMELY, J AND ORDER
15.  C            J+1. IF IB(K1) CORRESPONDING TO ELEMENT J IS GREATER THAN ORDER
16.  C            IB(K2) CORRESPONDING TO ELEMENT J+1, THEIR ORDER IN IA IS ORDER
17.  C            SWITCHED. ALL PREVIOUS ELEMENTS IN IA ARE THEN ALSO CHECKED ORDER
18.  C            AGAINST J+1 IN LIKE MANNER AND THEIR ORDER IN IA REARRANGED ORDER
19.  C            IF NECESSARY.                                              ORDER
20.          K3=J                                                           ORDER
21.          K4=J+1                                                         ORDER
22.        3 IF(K4.LE.1)GO TO 9                                             ORDER
23.          K1=IA(K3)                                                      ORDER
24.          K2=IA(K4)                                                      ORDER
25.          IF(IB(K1).LE.IB(K2))GO TO 9                                    ORDER
26.          JX=IA(K3)                                                      ORDER
27.          IA(K3)=IA(K4)                                                  ORDER
28.          IA(K4)=JX                                                      ORDER
29.          K3=K3-1                                                        ORDER
30.          K4=K3+1                                                        ORDER
31.          GO TO 3                                                        ORDER
32.  C            NOW CHECK REMAINING ELEMENTS IN SAME MANNER AND CONTINUE  ORDER
33.  C            UNTIL ALL ELEMENTS HAVE BEEN CHECKED.                    ORDER
34.        9 CONTINUE                                                       ORDER
35.          RETURN                                                         ORDER
36.          END                                                            ORDER
```

```
      SUBROUTINE OXYATM(K,LOCX,ITEST)                                   OXYATM
C        THIS SUBROUTINE DETERMINES WHETHER THERE ARE 0, 1, OR 2 -C(=O)- OXYATM
C        GROUPS PRESENT IN RING K WHICH ARE ADJACENT TO AN OXYGEN ATOM  OXYATM
C        LOCX (OR NITROGEN ATOM) LIKEWISE PART OF THE RING BACKBONE     OXYATM
C        STRUCTURE.                                                     OXYATM
      INTEGER WEIGHT(9)                                                 OXYATM
      DIMENSION MA(30)                                                  OXYATM
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),NCC     OXYATM
      COMMON/BLK3/IRING(40,30),IMATX(50,60),NW(100),IBC(100),KON(100),  OXYATM
     1IOSN(100),IB(100,3),INB,NOBR                                      OXYATM
      EQUIVALENCE (MA(1),IMATX(31,47))                                  OXYATM
C1.0     FIND LOCATION OF C(=O) GROUPS IN RING K.                       OXYATM
      NOXY=0                                                            OXYATM
      MF=IRING(K,1)+1                                                   OXYATM
      DO5   M=2,MF                                                      OXYATM
      IF(M.EQ.LOCX)GO TO 5                                              OXYATM
C          RING COMPONENT K1 IS NOT AN OXYGEN (OR NITROGEN) ATOM.       OXYATM
      K1=IRING(K,M)                                                     OXYATM
      IF(IX(K1,1,1).NE.5)GO TO 5                                        OXYATM
C          RING COMPONENT K1 CONTAINS C(=O) LINKAGE.                    OXYATM
      NOXY=NOXY+1                                                       OXYATM
      MA(NOXY)=M                                                        OXYATM
    5 CONTINUE                                                          OXYATM
C2.0     SET ITEST FLAG ACCORDING TO NUMBER OF C(=O) LINKAGES PRESENT.  OXYATM
      IF(NOXY.NE.2)GO TO 15                                             OXYATM
C          TWO C(=O) GROUPS PRESENT. ARE THEY ADJACENT TO THE OXYGEN    OXYATM
C          OR NITROGEN ATOM LOCX                                        OXYATM
      JD1=MA(2)-MA(1)                                                   OXYATM
      JD2=IABS(LOCX-MA(1))                                              OXYATM
      IF(JD1.NE.2.OR.JD2.NE.1.AND.JD2.NE.IRING(K,1)-1)GO TO 17          OXYATM
C             YES. SET FLAG=2.                                          OXYATM
    9 ITEST=2                                                           OXYATM
      RETURN                                                            OXYATM
   15 IF(NOXY.NE.1)GO TO 17                                             OXYATM
C             ONE C(=O) GROUP PRESENT. SET FLAG=1.                      OXYATM
      ITEST=1                                                           OXYATM
      RETURN                                                            OXYATM
C             NONE PRESENT. SET FLAG=0.                                 OXYATM
   17 ITEST=0                                                           OXYATM
      RETURN                                                            OXYATM
      END                                                               OXYATM
```

19 FEB 73 8.02-39

```
      SUBROUTINE PRINT1                                                 PRINT1
C        THIS SUBROUTINE PRINTS OUT THE ATOMIC COMPOSITION, THE ASSIGNED PRINT1
C        WEIGHTS, AND THE STRUCTURE AND COORDINATE ARRAY IX. IT ALSO     PRINT1
C        WILL OUTPUT, IF PRESENT, THE NUMBER OF COMPONENTS IN EACH RING  PRINT1
C        AND THEIR GROUP NUMBERS.                                        PRINT1
      INTEGER SYMX(4),SYMBOL(9),GRID(80,80)                             PRINT1
      INTEGER WEIGHT(9)                                                 PRINT1
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                      PRINT1
      COMMON/BLK2/WEIGHT,NWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC     PRINT1
      COMMON/BLK3/IRING(40,30),IMATX(30,80),NW(100),IBC(100),KON(100),  PRINT1
     1IDER(100),IS(100,8),IRA,NOBR                                      PRINT1
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRTOT      PRINT1
      COMMON/BLK5/NDATM,NUMATM(5),MBC(50),MBS(2),JW,JV,LFLAGS,LFLAGG    PRINT1
C.0      THIS SECTION PRINTS OUT THE ATOMIC COMPOSITION OF THE MOLECULE. PRINT1
      WRITE(6,1)                                                        PRINT1
    1 FORMAT(// 1HO,49X,33HATOMIC  COMPOSITION  OF  MOLECULE           PRINT1
     1//1HO,55X,4HATOM,8X,6HNUMBER)                                     PRINT1
      DO5   J=1,NDATM                                                   PRINT1
      IF(NUMATM(J).EQ.0)GO TO 5                                         PRINT1  5
      WRITE(6,3)SYMBOL(J),NUMATM(J)                                     PRINT1
    3 FORMAT(1H ,57X,A4,5X,I5)                                          PRINT1
C                                                                       PRINT1
      CONTINUE                                                          PRINT1
C2.      THIS SECTION PRINTS OUT THE ASSIGNED WEIGHTS OF THE GROUPS     PRINT1
C        AND THE IX ARRAY. DESCRIPTION OF IX OUTPUT, IX(M,N,J) -        PRINT1
C        DATA FOR CORE ATOM(N) AT N=1 AND FOR LIGANDS AT N=2 TO N=5,    PRINT1
C        CONTENTS OF J COLUMN - J=1 CHEMICAL SYMBOL OF COMPONENT,       PRINT1
C        J=2 GRID ROW COORDINATE. J=3 GRID COLUMN COORDINATE,           PRINT1
C        J=4 DIRECTION OF LIGAND BOND, J=5 BOND TYPE, J=6 GROUP NUMBER  PRINT1
C        OF CORE ATOM.                                                  PRINT1
      WRITE(6,9)                                                        PRINT1
    9 FORMAT(// 1HO,47X,34HGROUP  STRUCTURE  AND  COMPOSITION          PRINT1
     1//1H ,23X,85HGROUP     GROUP     CHEMICAL SYMBOL   GRID ROW  GRID CO PRINT1
     2LUMN   BOND   BOND  GROUP NUMBER                 /1H ,21X,85HNUMBER  PRINT1
     3WEIGHT    OF COMPONENT    COORDINATE  COORDINATE  VECTOR  TYPE  OF  PRINT1
     4 CORE ATOM       )                                                PRINT1
      DO14   M=1,KCC                                                    PRINT1
      L=IX(M,1,1)                                                       PRINT1
      WRITE(6,11)M,MOLWT(M),SYMBOL(L),IX(M,1,2),IX(M,1,3),IX(M,1,6)     PRINT1
   11 FORMAT(1H ,22X,I3,5X,I5,11X,A4,7X,I5,7X,I5,22X,I5)                PRINT1
      NF=KON(M)+1                                                       PRINT1
      DO15   N=2,NF                                                     PRINT1
      L=IX(M,N,1)                                                       PRINT1
      WRITE(6,13)SYMBOL(L),(IX(M,N,J),J=2,6)                            PRINT1
   13 FORMAT(1H ,46X,A4,7X,I5,7X,I5,7X,I3,3X,I4,5X,I5)                  PRINT1
   15 CONTINUE                                                          PRINT1
      WRITE(6,17)                                                       PRINT1
   17 FORMAT(1H )                                                       PRINT1
   19 CONTINUE                                                          PRINT1
C3.0     IF RING STRUCTURES ARE PRESENT, THIS SECTION PRINTS OUT THE    PRINT1
C        NUMBER OF COMPONENTS IN EACH RING AND THEIR GROUP NUMBERS.      PRINT1
      IF(IRC.EQ.0)RETURN                                                PRINT1
      DO37   J=1,IRC                                                    PRINT1
      WRITE(6,31)J,IRING(J,1)                                           PRINT1
   31 FORMAT(/// 1H ,24X,11HRING NUMBER,I3,4H  IS,I3,10H  MEMBERED)    PRINT1
      NF=IRING(J,1)+1                                                   PRINT1
      WRITE(6,35)(IRING(J,N),N=2,NF)                                    PRINT1
   35 FORMAT(1HO,23X,39HGROUP NUMBERS OF RING CONSTITUENTS  =  ,10(I5,  PRINT1
     1(1H,)/(1H ,62X,10(I5,1H,))))                                      PRINT1
   37 CONTINUE                                                          PRINT1
      RETURN                                                            PRINT1
      END                                                               PRINT1
```

19 FEB 73 8.08.39

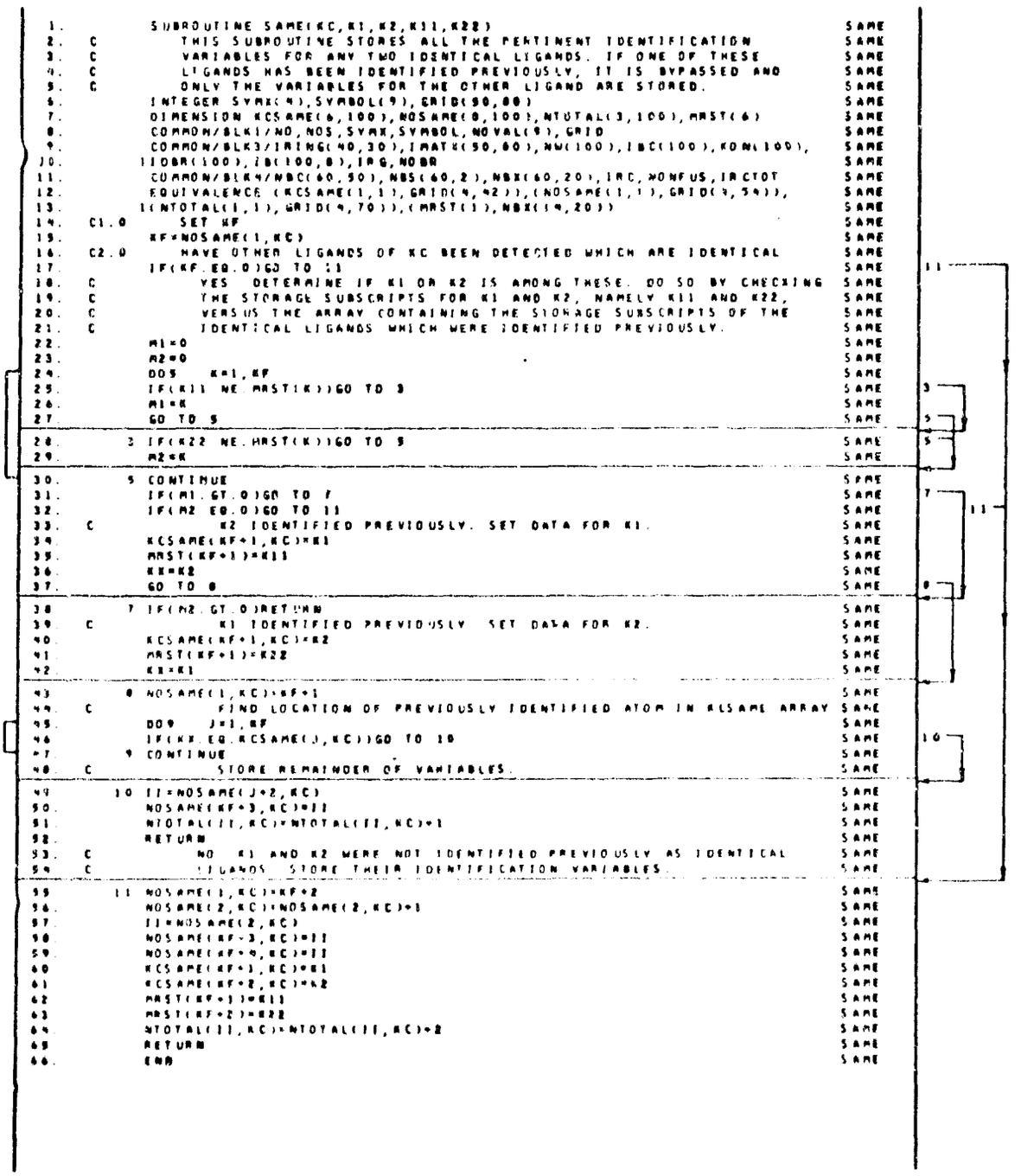
```
      SUBROUTINE PRINT2(NTITLE)                                          PRINT2
C        THIS SUBROUTINE OUTPUTS THE TITLE PRECEDING THE PRINTOUT OF     PRINT2
C        RING, SECOND-ORDER INTERACTION, AND CERTAIN SYMMETRY            PRINT2
C        CORRECTIONS. THE FLAG NTITLE IS ALSO RESET TO 1 WHEN THIS       PRINT2
C        ROUTINE IS EXECUTED.                                            PRINT2
      NTITLE=1                                                           PRINT2
      WRITE(6,1)                                                         PRINT2
    1 FORMAT(///1H0,28X,73HNONGROUP  INTERACTION  AND  INTERNAL  ROTATION PRINT2
     1AL  SYMMETRY  CONTRIBUTIONS     /1H0,6X,9HTYPE,11X,14HRING NUMBER O PRINT2
     2R,3X,6HAMOUNT,2X,17HHEAT OF FORMATION,3X,7HENTROPY,14X,26HHEAT CAP PRINT2
     3ACITY COEFFICIENTS     /1H ,22X,12HGROUP NUMBER,17X,9HKCAL,10X,   PRINT2
     49HCAL/DEG K,5X,9HCAL/DEG K,2X,12HCAL/DEG K**2,1X,12HCAL/DEG K**3, PRINT2
     51X,12HCAL/DEG K**4)                                               PRINT2
      RETURN                                                             PRINT2
      END                                                                PRINT2
```

```
      SUBROUTINE RESETR(MDIF,LX)                                        RESETR
C        THIS SUBROUTINE SETS THE RING ARRAY IBC AND ALSO ADDS THE      RESETR
C        RESIDUAL NON-BRANCH RING ATOMS TO ALL THE CHAINS, IF THE FIRST RESETR
C        CHAIN ATOM DEFINED WAS A NON-BRANCH RING ATOM AND NO TERMINAL  RESETR
C        CORE ATOMS ARE PRESENT IN THE MOLECULE. THESE RESIDUAL ATOMS   RESETR
C        WERE NOT PREVIOUSLY INCLUDED IN THE CHAIN DEFINITIONS.         RESETR
      INTEGER WEIGHT(9)                                                 RESETR
      DIMENSION NRLOC(40)                                               RESETR
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC     RESETR
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KOM(100),  RESETR
     1IDBR(100),IB(100,8),IRG,NDBR                                      RESETR
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT     RESETR
      EQUIVALENCE (NRLOC(1),IMATX(3,47))                                RESETR
C1.0     SET RING ARRAY IBC.                                            RESETR
      DO 3    K=1,IRC                                                   RESETR
      LF=IRING(K,1)+1                                                   RESETR
      DO 3    L=2,LF                                                    RESETR
      KC=IRING(K,L)                                                     RESETR
      IBC(KC)=K                                                         RESETR
    3 CONTINUE                                                          RESETR
      KC=NBC(1,1)                                                       RESETR
      IF(MDIF.EQ.0.OR.IBC(KC).EQ.0.OR.NC(KC).GT.2)RETURN                RESETR
C2.0     FIRST ATOM ON FIRST CHAIN IS A RING SPECIES BUT NOT A CHAIN    RESETR
C        ATOM AND NO TERMINAL CORE ATOMS ARE CONTAINED IN MOLECULE.     RESETR
C        FIND CHAIN LXX THAT IDENTIFIES RING K.                         RESETR
      K=IBC(KC)                                                         RESETR
      LXX=NRLOC(K)                                                      RESETR
      NBP=NBS(LXX,2)                                                    RESETR
      IF(NBP.EQ.0)RETURN                                                RESETR
C3.0     CHAIN ATOMS EXIST IN RING. FIND NUMBER OF NON-BRANCH ATOMS     RESETR
C        JDIF TO BE ADDED TO ALL CHAINS (IF ANY).                       RESETR
      JDIF=IRING(K,1)-NBX(LXX,NBP)                                      RESETR
      IF(JDIF.LE.0)RETURN                                               RESETR
C4.0     ADD THE JDIF NON-BRANCH RING ATOMS TO ALL CHAINS AND CHANGE    RESETR
C        NBS AND NBX ACCORDINGLY.                                       RESETR
      MZ=NBX(LXX,NBP)+1                                                 RESETR
      DO 11   L=1,LX                                                    RESETR
      IF(L.NE.LXX)GO TO 6                                               RESETR
C           CHAIN L IS RING IDENTIFIER CHAIN. MOVE THE FIRST MF ELEMENTS RESETR
C           IN ARRAY NBC CORRESPONDING TO CHAIN L JDIF LOCATIONS.       RESETR
      MF=NBX(LXX,NBP)                                                   RESETR
      MX=MF+1                                                           RESETR
      DO 5    M=1,MF                                                    RESETR
      MX=MX-1                                                           RESETR
      MY=MX+JDIF                                                        RESETR
      NBC(L,MY)=NBC(L,MX)                                               RESETR
    5 CONTINUE                                                          RESETR
      GO TO 8                                                           RESETR
C           CHAIN L IS NOT RING IDENTIFIER CHAIN. MOVE ALL THE ELEMENTS RESETR
C           IN ARRAY NBC CORRESPONDING TO CHAIN L JDIF LOCATIONS.       RESETR
    6 MF=NBS(L,1)                                                       RESETR
      MX=MF+1                                                           RESETR
      DO 7    M=1,MF                                                    RESETR
      MX=MX-1                                                           RESETR
      MY=MX+JDIF                                                        RESETR
      NBC(L,MY)=NBC(L,MX)                                               RESETR
    7 CONTINUE                                                          RESETR
      NBS(L,1)=NBS(L,1)+JDIF                                            RESETR
C           STORE THE ADDITIONAL (REMAINING) RING ELEMENTS IN THE FIRST RESETR
C           JDIF LOCATIONS OF NBC CORRESPONDING TO CHAIN L.             RESETR
    8 MX=MZ                                                             RESETR
      DO 9    M=1,JDIF                                                  RESETR
      MX=MX+1                                                           RESETR
      NBC(L,M)=IRING(K,MX)                                              RESETR
    9 CONTINUE                                                          RESETR
      NBPA=NBS(L,2)                                                     RESETR
      DO 11   N=1,NBPA                                                  RESETR
      NBX(L,N)=NBX(L,N)+JDIF                                            RESETR
   11 CONTINUE                                                          RESETR
      RETURN                                                            RESETR
      END                                                               RESETR
```

```
      SUBROUTINE RING(LX,LR,KK,IERR)                                    RING
C        THIS SUBROUTINE DETERMINES IF A SIMILAR RING WAS NOT PREVIOUSLY RING
C        IDENTIFIED. IF NOT, THE GROUP NUMBERS OF THE RING CONSTITUENTS  RING
C        ARE STORED IN ARRAY IRING. IF RING IS A DUPLICATE, IT IS NOT    RING
C        USED.                                                           RING
      DIMENSION NRLOC(90)                                               RING
      COMMON/BLK3/IRING(40,30),IMATX(50,99),NW(100),IBC(100),KUN(100),  RING
     1IDBR(10^),IA(100,8),IRS,NONN                                      RING
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,KONFUS,IRCTOT     RING
      EQUIVALENCE (NRLOC(1),IMATX(3,47))                                RING
C1.0     CHECK IF IDENTICAL RING HAS ALREADY BEEN IDENTIFIED AND STORED RING
C        IN IRING ARRAY. IF SO, DELETE RING AND IF THE CHAIN IS ALSO     RING
C        EQUAL, DELETE CHAIN AS WELL.                                    RING
      NALT=LR-KE                                                        RING
      KF=NALT+1                                                         RING
      IF(IRC.EQ.0)GO TO 9                                               RING
      DO7    LL=1,IRC                                                   RING
      IF(IRING(LL,1).NE.NALT)GO TO 7                                    RING
C           RING LL HAS SAME NUMBER OF RING MEMBERS. TEST IT.           RING
      K1=LR                                                             RING
C             ARE RING MEMBERS THE SAME EXCEPT POSITIONED IN REVERSE ORDER RING
      DO8    K=2,KF                                                     RING
      IF(IRING(LL,K).NE.NBC(LX,K1))GO TO 3                              RING
      K'=K1-1                                                           RING
    5 CONTINUE                                                          RING
C                YES. ARE THE TWO CHAINS THE SAME LENGTH                RING
      LXX=NRLOC(LL)                                                     RING
      IF(NBS(LXX,1).NE.LR-1)RETURN                                      RING
C                  YES. IF THERE ARE NO CHAINS IN ADDITION TO THE RINGS, GO RING
C                  TO 2. OTHERWISE, CONTINUE.                           RING
      IF(NBS(LXX,1).EQ.IRING(LL,1))GO TO 2                              RING
      K1=KK-1                                                           RING
C             ARE THE REMAINDER OF THE CHAIN COMPONENTS THE SAME         RING
      DO9    J=1,K1                                                     RING
      IF(NBC(LX,J).NE.NBC(LXX,J))RETURN                                 RING
    9 CONTINUE                                                          RING
C                YES. SET KK=-1 WHICH DELETES CHAIN FORMATION LX.       RING
    2 KK=-1                                                             RING
      RETURN                                                            RING
C                NO. ARE THE RING COMPONENTS THE SAME EXCEPT FOR BEING  RING
C                INVERTED AND DISPLACED BY ONE IN THEIR STORAGE ORDER   RING
    3 K1=LR-1                                                           RING
      DO6    K=2,KF                                                     RING
      IF(IRING(LL,K).NE.NBC(LX,K1))GO TO 7                              RING
      K1=K1-1                                                           RING
    6 CONTINUE                                                          RING
      RETURN                                                            RING
    7 CONTINUE                                                          RING
C2.0     RING IS NEW. STORE THE TOTAL NUMBER OF COMPONENTS (NALT) OF    RING
C        RING IRC IN IRING(IRC,1) AND THE GROUP NUMBERS OF THESE        RING
C        COMPONENTS IN IRING(IRC,K) FROM K=2 TO NALT+1.                 RING
    9 IRA=IRC+1                                                         RING
C             ARE THERE TOO MANY RINGS PRESENT                          RING
      IF(IRA.LE.40)GO TO 10                                             RING
C                YES. DETERMINE IF SOME ARE NOT UNIQUE. IF SO, GO TO 10. RING
C                OTHERWISE, EXIT.                                       RING
      CALL LESSEN(IRC1)                                                 RING
      IF(IRC.LT.IRCX)GO TO 10                                           RING
      IERR=1                                                            RING
      IRC=IRC+1                                                         RING
      RETURN                                                            RING
C           STORE RING DATA IN IRING ARRAY.                             RING
   10 IRC=IRC+1                                                         RING
      K1=KK                                                             RING
      IRING(IRC,1)=NALT                                                 RING
      DO11   K=2,KF                                                     RING
      IRING(IRC,K)=NBC(LX,K1)                                           RING
   11 K1=K1+1                                                           RING
      NRLOC(IRC)=LX                                                     RING
      RETURN                                                            RING
      END                                                               RING
```

19 FEB 73 8.02-38

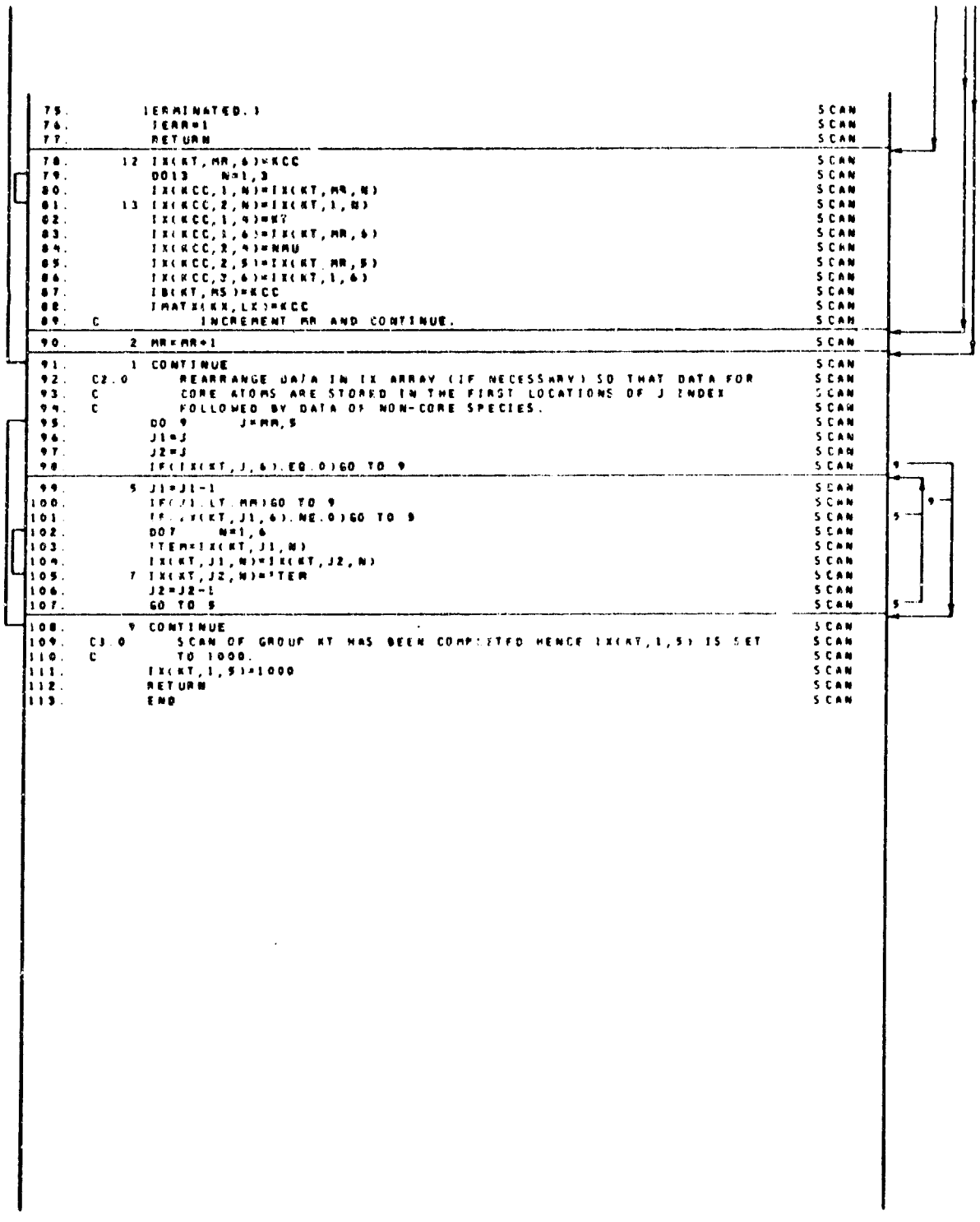
SAME



```
      SUBROUTINE SAME(KC,K1,K2,K11,K22)                                  SAME
C        THIS SUBROUTINE STORES ALL THE PERTINENT IDENTIFICATION         SAME
C        VARIABLES FOR ANY TWO IDENTICAL LIGANDS. IF ONE OF THESE        SAME
C        LIGANDS HAS BEEN IDENTIFIED PREVIOUSLY, IT IS BYPASSED AND      SAME
C        ONLY THE VARIABLES FOR THE OTHER LIGAND ARE STORED.             SAME
      INTEGER SYMX(9),SYMBOL(9),GRID(90,80)                              SAME
      DIMENSION KCSAME(6,100),NOSAME(8,100),NTOTAL(3,100),MRST(6)        SAME
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                       SAME
      COMMON/BLK3/IRING(40,30),IMAT(90,80),NW(100),IBC(100),KON(100),    SAME
     1IOBR(100),ZB(100,8),IRG,NOBR                                       SAME
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT      SAME
      EQUIVALENCE (KCSAME(1,1),GRID(4,42)),(NOSAME(1,1),GRID(4,54)),     SAME
     1(NTOTAL(1,1),GRID(4,70)),(MRST(1),NBX(14,20))                      SAME
C1.0     SET KF                                                          SAME
      KF=NOSAME(1,KC)                                                    SAME
C2.0     HAVE OTHER LIGANDS OF KC BEEN DETECTED WHICH ARE IDENTICAL      SAME
      IF(KF.EQ.0)GO TO 11                                                SAME
C           YES  DETERMINE IF K1 OR K2 IS AMONG THESE. DO SO BY CHECKING SAME
C           THE STORAGE SUBSCRIPTS FOR K1 AND K2, NAMELY K11 AND K22,    SAME
C           VERSUS THE ARRAY CONTAINING THE STORAGE SUBSCRIPTS OF THE    SAME
C           IDENTICAL LIGANDS WHICH WERE IDENTIFIED PREVIOUSLY.          SAME
      M1=0                                                               SAME
      M2=0                                                               SAME
      DO 5   K=1,KF                                                      SAME
      IF(K11.NE.MRST(K))GO TO 3                                          SAME
      M1=K                                                               SAME
      GO TO 5                                                            SAME
    3 IF(K22.NE.MRST(K))GO TO 5                                          SAME
      M2=K                                                               SAME
    5 CONTINUE                                                           SAME
      IF(M1.GT.0)GO TO 7                                                 SAME
      IF(M2.EQ.0)GO TO 11                                                SAME
C           K2 IDENTIFIED PREVIOUSLY. SET DATA FOR K1.                   SAME
      KCSAME(KF+1,KC)=K11                                                SAME
      MRST(KF+1)=K11                                                     SAME
      KK=K2                                                              SAME
      GO TO 8                                                            SAME
    7 IF(M2.GT.0)RETURN                                                  SAME
C           K1 IDENTIFIED PREVIOUSLY  SET DATA FOR K2.                   SAME
      KCSAME(KF+1,KC)=K22                                                SAME
      MRST(KF+1)=K22                                                     SAME
      KK=K1                                                              SAME
    8 NOSAME(1,KC)=KF+1                                                  SAME
C           FIND LOCATION OF PREVIOUSLY IDENTIFIED ATOM IN KCSAME ARRAY  SAME
      DO 9   J=1,KF                                                      SAME
      IF(KK.EQ.KCSAME(J,KC))GO TO 10                                     SAME
    9 CONTINUE                                                           SAME
C           STORE REMAINDER OF VARIABLES.                                SAME
   10 II=NOSAME(J+2,KC)                                                  SAME
      NOSAME(KF+3,KC)=II                                                 SAME
      NTOTAL(II,KC)=NTOTAL(II,KC)+1                                      SAME
      RETURN                                                             SAME
C        NO  K1 AND K2 WERE NOT IDENTIFIED PREVIOUSLY AS IDENTICAL       SAME
C        LIGANDS  STORE THEIR IDENTIFICATION VARIABLES.                  SAME
   11 NOSAME(1,KC)=KF+2                                                  SAME
      NOSAME(2,KC)=NOSAME(2,KC)+1                                        SAME
      II=NOSAME(2,KC)                                                    SAME
      NOSAME(KF+3,KC)=II                                                 SAME
      NOSAME(KF+4,KC)=II                                                 SAME
      KCSAME(KF+1,KC)=K11                                                SAME
      KCSAME(KF+2,KC)=K22                                                SAME
      MRST(KF+1)=K11                                                     SAME
      MRST(KF+2)=K22                                                     SAME
      NTOTAL(II,KC)=NTOTAL(II,KC)+2                                      SAME
      RETURN                                                             SAME
      END                                                                SAME
```

19 FEB 73 8.02.38

```
      SUBROUTINE SCAN(KT,MR,MR,NUM,K,L,IERR)                                SCAN
C        THIS SUBROUTINE LOCATES AND IDENTIFIES ALL ATOMS BONDED TO THE     SCAN
C        CORE ATOM OF GROUP KT WHICH HAVE NOT YET BEEN SCANNED AND          SCAN
C        STORES PERTINENT DATA OF EACH OF THESE LIGANDS IN IX ARRAY. IT     SCAN
C        ALSO REARRANGES DATA OF GROUP KT IN IX ARRAY SO THAT DATA OF       SCAN
C        CORE SPECIES ARE STORED FIRST, FOLLOWED BY THE DATA OF NON-CORE    SCAN
C        SPECIES.                                                           SCAN
      INTEGER SYMX(9),SYMBOL(9),GRID(50,80)                                 SCAN
      INTEGER WEIGHT(9)                                                     SCAN
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVOL(9),GRID                          SCAN
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC         SCAN
      COMMON/BLK3/IRING(50,30),IMATX(50,80),NM(100),IBC(100),KON(100),      SCAK
     1IOBR(100),IB(100,8),IRG,NOBR                                          SCAN
      COMMON/BLK5/NDATM,NUMATM(9),MBC(50),MBS(2),JM,JV,LFLAGS,LFLAGE        SCAN
C1.0     INITIALIZE MS AND FIND BOND DIRECTION FROM CORE ATOM KT TO         SCAN
C        PARENT ATOM.                                                       SCAN
      MS=0                                                                  SCAN
      NZ=NUM+4                                                              SCAN
      IF(NZ.GT.8)NZ=NZ-8                                                    SCAN
C2.0     START SEARCH AND IDENTIFICATION CYCLE.                             SCAN
      DO1   J=1,7                                                           SCAN
C           FIND TEST BOND DIRECTION.                                       SCAN
      NX=NZ+J                                                               SCAN
      IF(NX.GT.8)NX=NX-8                                                    SCAN
C           FIND TRANSFORMATION COORDINATES FOR THIS BOND DIRECTION.        SCAN
      CALL ASSIGN(NX,K,L,KD,LD,JI,JF)                                       SCAN
      IF(JI.GT.JF)GO TO 1                                                   SCAN
      KX=R+KD                                                               SCAN
      LX=L+LD                                                               SCAN
      IF(GRID(KX,LX).EQ.SYMX(4))GO TO 1                                     SCAN
C           TEST LOCATION IS NOT BLANK.                                     SCAN
      IF(MR.LE.5)GO TO 4                                                    SCAN
C             EXCESSIVE LIGANDS PRESENT IN GROUP. PRINT ERROR MESSAGE,      SCAN
C             SET ERROR FLAG, AND EXIT.                                     SCAN
      WRITE(6,3)KT                                                          SCAN
    3 FORMAT(// 1H0,31X,43HEXCESSIVE NUMBER OF LIGANDS ABOUT CORE ATOM,     SCAN
     1I4,19H.  CASE TERMINATED.)                                            SCAN
      IERR=1                                                                SCAN
      RETURN                                                                SCAN
C           LOCATE AND IDENTIFY NON-BOND SYMBOL.                            SCAN
    4 CALL BOND(KX,LX,KD,LD,JI,JF,IX(KT,MR,5),IERR)                         SCAN
      IF(IERR.EQ.1)RETURN                                                   SCAN
      CALL IDENT(NZ,KX,LX,M,IERR)                                           SCAN
      IF(IERR.EQ.1)RETURN                                                   SCAN
C           STORE DATA OF LIGAND IN ARRAY IX.                               SCAN
      IX(KT,MR,1)=M                                                         SCAN
      IX(KT,MR,2)=KX                                                        SCAN
      IX(KT,MR,3)=LX                                                        SCAN
      IX(KT,MR,4)=NZ                                                        SCAN
C           IS LIGAND ALSO A CORE ATOM                                      SCAN
      NMU=IX(KT,MR,4)                                                       SCAN
      CALL NUMBER(NMU,NJ,KX,LX,MP,KKX,LXX,KD,LD,JI,JF)                      SCAN
      IF(MP.EQ.0)GO TO 2                                                    SCAN
C             YES, IT IS. INCREMENT NC AND MS.                              SCAN
      NC(KT)=NC(KT)+1                                                       SCAN
      MS=MS+1                                                               SCAN
C           HAS LIGAND BEEN SCANNED PREVIOUSLY, THAT IS, IS LIGAND PART     SCAN
C           OF RING SYSTEM                                                  SCAN
      IF(IMATX(KX,LX).EQ.0)GO TO 11                                         SCAN
C             YES, IT IS.  OBTAIN REQUIRED DATA FROM IMATX.                 SCAN
      IX(KT,MR,6)=IMATX(KX,LX)                                              SCAN
      IB(KT,MS)=IMATX(KX,LX)                                                SCAN
C             DELETE ATOM(S) IN SYMBOL IX(KT,MR,1) FROM NUMATM.             SCAN
      CALL SUMATM(IX(KT,MR,1),-1,NUMATM)                                    SCAN
C             INCREMENT IRG AND CONTINUE.                                   SCAN
      IRG=IRG+1                                                             SCAN
      GO TO 2                                                               SCAN
C             LIGAND NOT PART OF RING SYSTEM. ASSIGN NEW GROUP              SCAN
C             NUMBER(KCC) TO IT AND STORE DATA FOR THIS GROUP IN IX.        SCAN
C             SET IMATX.                                                    SCAN
   11 KCC=KCC+1                                                             SCAN
      IF(KCC.LE.100)GO TO 12                                                SCAN
      WRITE(6,110)                                                          SCAN
  110 FORMAT(//1H0,36 ,56HNUMBER OF ALLOWED CORE ATOMS EXCEEDED   CASE T SCAN
```



```
      TERMINATED.)                                                      SCAN
       IERR=1                                                           SCAN
       RETURN                                                           SCAN
   12  IX(KT,MR,6)=KCC                                                  SCAN
       DO13   N=1,3                                                     SCAN
       IX(KCC,1,N)=IX(KT,MR,N)                                          SCAN
   13  IX(KCC,2,N)=IX(KT,1,N)                                           SCAN
       IX(KCC,1,4)=KT                                                   SCAN
       IX(KCC,1,6)=IX(KT,MR,6)                                          SCAN
       IX(KCC,2,4)=NAU                                                  SCAN
       IX(KCC,2,5)=IX(KT,MR,5)                                          SCAN
       IX(KCC,2,6)=IX(KT,1,6)                                           SCAN
       IB(KT,MS)=KCC                                                    SCAN
       IMAT(KX,LX)=KCC                                                  SCAN
C            INCREMENT MR AND CONTINUE.                                 SCAN
    2  MR=MR+1                                                          SCAN
    1  CONTINUE                                                         SCAN
C2.0       REARRANGE DATA IN IX ARRAY (IF NECESSARY) SO THAT DATA FOR   SCAN
C          CORE ATOMS ARE STORED IN THE FIRST LOCATIONS OF J INDEX       SCAN
C          FOLLOWED BY DATA OF NON-CORE SPECIES.                         SCAN
       DO 9       J=MR,5                                                SCAN
       J1=J                                                             SCAN
       J2=J                                                             SCAN
       IF(IX(KT,J,6).EQ.0)GO TO 9                                       SCAN
    5  J1=J1-1                                                          SCAN
       IF(J1.LT.MR)GO TO 9                                              SCAN
       IF(IX(KT,J1,6).NE.0)GO TO 9                                      SCAN
       DO7     N=1,6                                                    SCAN
       ITEM=IX(KT,J1,N)                                                 SCAN
       IX(KT,J1,N)=IX(KT,J2,N)                                          SCAN
    7  IX(KT,J2,N)=ITEM                                                 SCAN
       J2=J2-1                                                          SCAN
       GO TO 5                                                          SCAN
    9  CONTINUE                                                         SCAN
C3.0       SCAN OF GROUP KT HAS BEEN COMPLETED HENCE IX(KT,1,5) IS SET  SCAN
C          TO 1000.                                                      SCAN
       IX(KT,1,5)=1000                                                  SCAN
       RETURN                                                           SCAN
       END                                                              SCAN
```

19 FEB 73 6.02.39

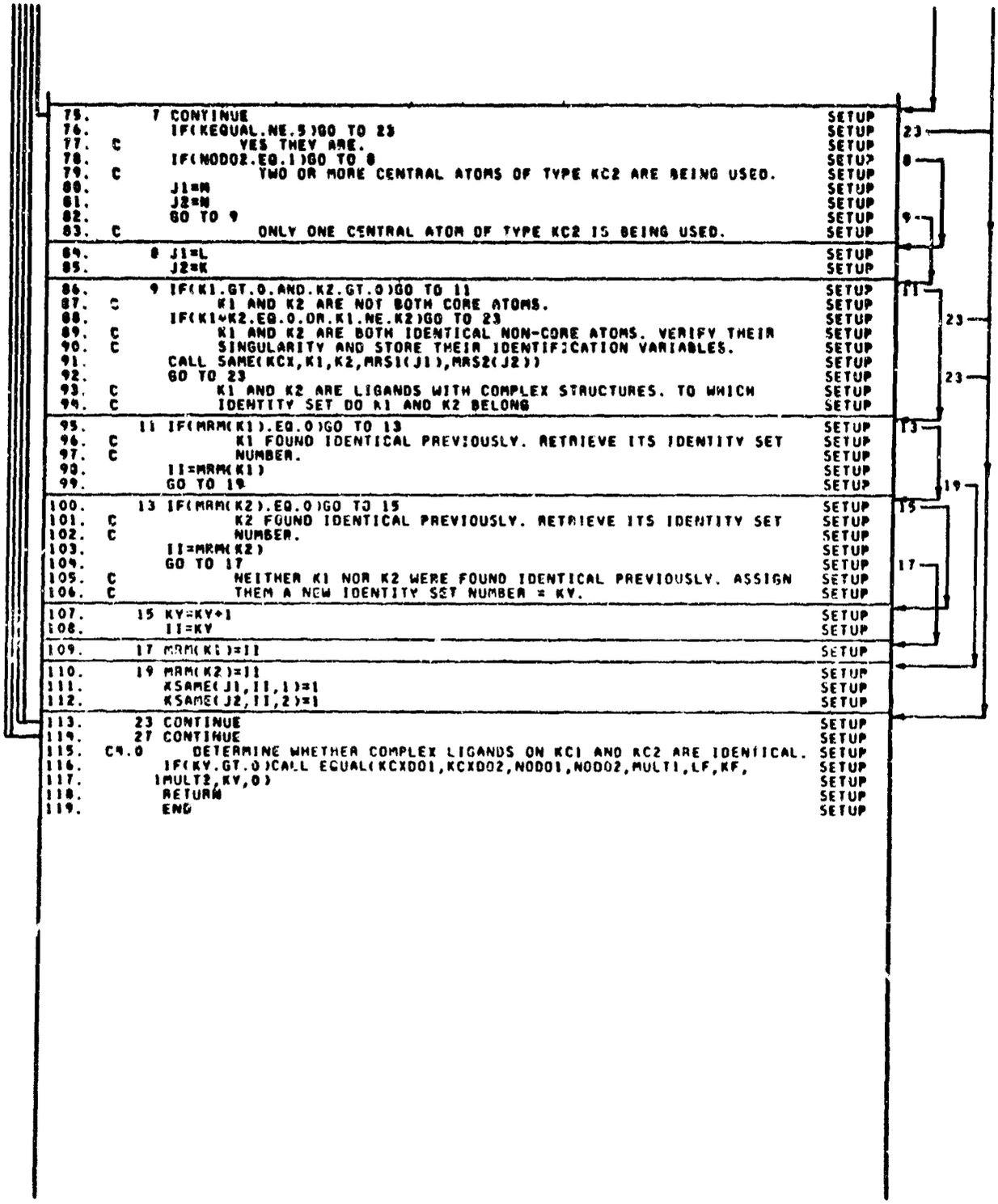
```
  1.          SUBROUTINE SCANBR(KCNEXT,KAFTER)                                  SCANBR
  2.  C          THIS SUBROUTINE COMPUTES THE ATOMIC COMPOSITION OF ALL LIGANDS  SCANBR
  3.  C          BONDED TO BRANCH ATOM KCNEXT AND ESTABLISHES THE SIMILARITIES   SCANBR
  4.  C          OF THESE LIGANDS PROVIDING KCNEXT IS NOT A RING ATOM.           SCANBR
  5.          INTEGER WEIGHT(9)                                                 SCANBR
  6.          INTEGER PERTKC(9,100),PERLIG(9,100)                               SCANBR
  7.          DIMENSION KTOT(5),JBR(100),LIGAND(5,9,100)                        SCANBR
  8.          COMMON/BLK2/WEIGHT,MAXY(9),MOLWT(100),IX(100,5,6),NC(100),KCC     SCANBR
  9.          COMMON/BLK3/IRING(90,30),IMATX(50,60),NW(100),IBC(100),KON(100),  SCANBR
 10.         1IDBR(100),IA(100,8),IRG,NOBR                                      SCANBR
 11.          COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONPUS,IRCTOT     SCANBR
 12.          COMMON/BLK5/NDATM,NUMATM(5),MBC(50),MBS(2),JM,JV,LFLAGS,LFLAGG    SCANBR
 13.          EQUIVALENCE (KTOT(1),IB(62,8)),(JBR(1),IMATX(2,93)),              SCANBR
 14.         1(LIGAND(1,1,1),IMATX(2,1)),(PERTKC(1,1),NBX(2,5)),(PERLIG(1,1),   SCANBR
 15.         2NBX(42,11))                                                       SCANBR
 16.  C1.0      INITIALIZE VARIABLES.                                           SCANBR
 17.          DO1    J=1,NDATM                                                  SCANBR
 18.          KTOT(J)=0                                                         SCANBR
 19.        1 CONTINUE                                                          SCANBR
 20.          KSUB=JBR(KCNEXT)                                                  SCANBR
 21.          IF(KSUB.LT.2)GO TO 5                                              SCANBR
 22.  C2.0      STORE SUM TOTAL OF THE ATOMIC COMPOSITIONS OF THE KCNEXT        SCANBR
 23.  C         LIGANDS IDENTIFIED THUS FAR IN KTOT.                            SCANBR
 24.          DO3    LX=2,KSUB                                                  SCANBR
 25.          DO3    J=1,NDATM                                                  SCANBR
 26.          KTOT(J)=KTOT(J)+LIGAND(J,LX,KCNEXT)                               SCANBR
 27.        3 CONTINUE                                                          SCANBR
 28.        5 IF(NC(KCNEXT).EQ.KON(KCNEXT))GO TO 9                              SCANBR
 29.  C3.0      KCNEXT HAS NON-CORE ATOM LIGANDS. COMPUTE THEIR ATOMIC          SCANBR
 30.  C         COMPOSITIONS.                                                   SCANBR
 31.          LS=NC(KCNEXT)+2                                                   SCANBR
 32.          LF=KON(KCNEXT)+1                                                  SCANBR
 33.          DO7    LX=LS,LF                                                   SCANBR
 34.          KSUB=KSUB+1                                                       SCANBR
 35.          PERTKC(KSUB,KCNEXT)=-IX(KCNEXT,LX,1)                              SCANBR
 36.          CALL SUMATM(IX(KCNEXT,LX,1),1,LIGAND(1,KSUB,KCNEXT))              SCANBR
 37.          CALL SUMATM(IX(KCNEXT,LX,1),1,KTOT(1))                            SCANBR
 38.        7 CONTINUE                                                          SCANBR
 39.        9 IF(KAFTER.EQ.1000)GO TO 12                                        SCANBR
 40.  C4.0      KAFTER IS NOT THE FIRST BRANCH ATOM IDENTIFIED. FIND ITS ATOMIC SCANBR
 41.  C         COMPOSITION.                                                    SCANBR
 42.          CALL SUMATM(IX(KCNEXT,1,1),1,KTOT(1))                             SCANBR
 43.          DO11   J=1,NDATM                                                  SCANBR
 44.          LIGAND(J,1,KCNEXT)=NUMATM(J)-KTOT(J)                              SCANBR
 45.       11 CONTINUE                                                          SCANBR
 46.          PERTKC(1,KCNEXT)=KAFTER                                           SCANBR
 47.  C5.0      SET VARIABLES JBR AND PERLIG.                                   SCANBR
 48.       12 JBR(KCNEXT)=KSUB                                                  SCANBR
 49.          DO13   J=1,KSUB                                                   SCANBR
 50.       13 PERLIG(J,KCNEXT)=J                                                SCANBR
 51.  C6.0      IF KCNEXT IS NOT A RING ATOM, DETERMINE WHICH OF ITS LIGANDS    SCANBR
 52.  C         ARE IDENTICAL TO WHICH (IF ANY).                                SCANBR
 53.          IF(IBC(KCNEXT).EQ.0)CALL SETUP(KCNEXT,KCNEXT,1,1,0,JBR(KCNEXT)-1,SCANBR
 54.         1JBR(KCNEXT),1)                                                    SCANBR
 55.          RETURN                                                            SCANBR
 56.          END                                                               SCANBR
```

```
      SUBROUTINE SCANCH(KCBEF,KC,KCNEXT)                                SCANCH
C        THIS SUBROUTINE COMPUTES THE ATOMIC COMPOSITION OF ALL LIGANDS  SCANCH
C        BONDED TO CHAIN ATOM KC AND ESTABLISHES THE SIMILARITIES OF     SCANCH
C        THESE LIGANDS PROVIDING KC IS NOT A RING ATOM.                  SCANCH
      INTEGER WEIGHT(9)                                                 SCANCH
      INTEGER PERTKC(9,100),PERLIG(9,100)                               SCANCH
      DIMENSION KTOT(5),JBR(100),LIGAND(5,9,100)                        SCANCH
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC     SCANCH
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),  SCANCH
     1IDBR(100),IB(100,8),IRG,NOBR                                      SCANCH
      COMMON/BLK4/NBC(60,50),NBS(30,2),NBX(60,20),IRC,NONFUS,IRCTOT     SCANCH
      COMMON/BLK5/NOATM,NUMATM(5),MBC(50),MBS(2),JM,JY,LFLAGS,LFLAGG    SCANCH
      EQUIVALENCE (KTOT(1),IB(62,8)),(LIGAND(1,1,1),IMATX(2,1)),        SCANCH
     )(JBR(1),IMATX(2,43)),(PERTKC(1,1),NBX(2,5)),(PERLIG(1,1),         SCANCH
     2NBX(42,11))                                                       SCANCH
C1.0     INITIALIZE VARIABLES. IF KCNEXT DOES NOT EXIST, SET IN=0.      SCANCH
      IN=1                                                              SCANCH
      IF(KCNEXT.EQ.1000)IN=0                                            SCANCH
      IF(KCBEF.EQ.1000)GO TO 3                                          SCANCH
C2.0     KC IS NOT THE FIRST ELEMENT IN THE CHAIN. THEREFORE KCBEF       SCANCH
C        EXISTS. COMPUTE ATOMIC COMPOSITION OF LIGAND KCBEF.             SCANCH
      IN=IN+1                                                           SCANCH
      DO1   J=1,NOATM                                                   SCANCH
      LIGAND(J,IN,KC)=KTOT(J)                                           SCANCH
    1 CONTINUE                                                          SCANCH
      PERTKC(IN,KC)=KCBEF                                               SCANCH
    3 IF(NC(KC).EQ.KON(KC))GO TO 7                                      SCANCH
C3.0     KC HAS NON-CORE ATOM LIGANDS. COMPUTE THEIR ATOMIC COMPOSITIONS SCANCH
      LS=NC(KC)+2                                                       SCANCH
      LF=KON(KC)+1                                                      SCANCH
      DO5   LX=LS,LF                                                    SCANCH
      IN=IN+1                                                           SCANCH
      PERTKC(IN,KC)=IX(KC,LX,1)                                         SCANCH
      CALL SUMATM(IX(KC,LX,1),1,LIGAND(1,IN,KC))                        SCANCH
      CALL SUMATM(IX(KC,LS,1),1,KTOT(1))                                SCANCH
    5 CONTINUE                                                          SCANCH
C4.0     STORE ATOMIC COMPOSITION OF ATOM KC IN KTOT                    SCANCH
    7 CALL SUMATM(IX(KC,1,1),1,KTOT(1))                                 SCANCH
      IF(KCNEXT.EQ.1000)GO TO 11                                        SCANCH
C5.0     KCNEXT EXISTS. COMPUTE ITS ATOMIC COMPOSITION.                 SCANCH
      DO9   J=1,NOATM                                                   SCANCH
      LIGAND(J,1,KC)=NUMATM(J)-KTOT(J)                                  SCANCH
    9 CONTINUE                                                          SCANCH
      PERTKC(1,KC)=KCNEXT                                               SCANCH
C6.0     SET VARIABLES JBR AND PERLIG.                                  SCANCH
   11 JBR(KC)=IN                                                        SCANCH
      DO13   J=1,IN                                                     SCANCH
   13 PERLIG(J,KC)=J                                                    SCANCH
C7.0     IF KC IS NOT RING ATOM, DETERMINE WHICH OF ITS LIGANDS ARE     SCANCH
C        IDENTICAL TO WHICH (IF ANY).                                    SCANCH
      IF(IBC(KC).EQ.0)CALL SETUP(KC,KC,1,1,0,JBR(KC)-1,JBR(KC),1)       SCANCH
      RETURN                                                            SCANCH
      END                                                               SCANCH
```

```
      INTEGER FUNCTION SEARCH(SUM,ORDSUM,LIM,KVALUE,KI)                  SEARCH
C        THIS ROUTINE FINDS LOCATION (KI) IN THE GROUP ADDITIVITY ARRAYS SEARCH
C        WHERE THE DATA FOR THE TEST GROUP WITH WEIGHT KVALUE ARE        SEARCH
C        STORED.                                                         SEARCH
      INTEGER SUM(100),ORDSUM(100)                                       SEARCH
C1.0     FIND LOCATION I TO BE CHECKED.                                  SEARCH
      I1=1                                                               SEARCH
      I2=LIM                                                             SEARCH
    1 I=(I1+I2)/2                                                        SEARCH
C2.      FIND CORRESPONDING LOCATION IN SUM.                             SEARCH
      KI=ORDSUM(I)                                                       SEARCH
C3.0     DO WEIGHTS AGREE                                                SEARCH
      IF(SUM(KI)-KVALUE)5,3,9                                            SEARCH
C           YES, A SOLUTION HAS BEEN FOUND. SET SEARCH TO ZERO AND EXIT. SEARCH
    3 SEARCH=0                                                           SEARCH
      RETURN                                                             SEARCH
C           NO, TEST VALUE IS LESS THAN DESIRED VALUE. ARE THE TWO       SEARCH
C           LOCATION LIMITS EQUAL                                        SEARCH
    5 IF(I1.EQ.I2)GO TO 7                                                SEARCH
C              NO, NOW FIND NEW TEST LOCATION AND REPEAT.                SEARCH
      I1=I+1                                                             SEARCH
      I1=MIN0(I1,I2)                                                     SEARCH
      GO TO 1                                                            SEARCH
C              YES, DESIRED VALUE EXCEEDS VALUE OF MAXIMUM ELEMENT IN SUM. SEARCH
C              SET SEARCH TO -1 AND EXIT.                                SEARCH
    7 SEARCH=-1                                                          SEARCH
      RETURN                                                             SEARCH
C           NO, TEST VALUE IS GREATER THAN DESIRED VALUE. ARE THE TWO    SEARCH
C           TEST LOCATION LIMITS EQUAL                                   SEARCH
    9 IF(I1.EQ.I2)GO TO 11                                               SEARCH
C              NO, NOW FIND NEW TEST LOCATION AND REPEAT.                SEARCH
      I2=I-1                                                             SEARCH
      I2=MAX0(I1,I2)                                                     SEARCH
      GO TO 1                                                            SEARCH
C              YES, DESIRED VALUE IS LESS THAN VALUE OF MINIMUM ELEMENT IN SEARCH
C              SUM. SET SEARCH TO 1 AND EXIT.                            SEARCH
   11 SEARCH=1                                                           SEARCH
      RETURN                                                             SEARCH
      END                                                                SEARCH
```

```
      SUBROUTINE SETUP(KCXDO1,KCXDO2,NODO1,NODO2,MULT1,LF,KF,MULT2)      SETUP
C        THIS SUBROUTINE PROCESSES THE COMPARISON OF LIGANDS BONDED TO   SETUP
C        THE SAME OR DIFFERENT CENTRAL CORE ATOMS BY FIRST COMPARING     SETUP
C        THEIR ATOMIC COMPOSITIONS. IF (1) THE LATTER ARE IDENTICAL AND  SETUP
C        (2) THE LIGANDS ARE BOTH NON-CORE TYPES, THE LIGANDS ARE        SETUP
C        IDENTICAL. IF (1) IS TRUE, BUT THE LIGANDS ARE CORE ATOMS       SETUP
C        THEIR IDENTITY MUST BE CHECKED FURTHER.                         SETUP
      INTEGER SYMX(9),SYMBOL(9),GRID(50,80)                              SETUP
      INTEGER WEIGHT(9)                                                  SETUP
      INTEGER PERTKC(9,100),PERLIG(9,100)                                SETUP
      DIMENSION KCXDO1(3),KCXDO2(3),KSAME(9,3,2),LIGAND(5,9,100),        SETUP
     1MRM(100),NOSAME(8,100),NTOTAL(3,100),MRS1(9),MRS2(9)               SETUP
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                       SETUP
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC      SETUP
      COMMON/BLK3/IRING(40,36),IMATX(50,80),NW(100),IBC(100),KON(100),   SETUP
     1IDBR(100),IB(100,8),IRB,NOBR                                       SETUP
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT      SETUP
      COMMON/BLK5/NDATM,NUMATM(5),MBC(50),MBS(2),JW,JY,LFLAGS,LFLAGB     SETUP
      EQUIVALENCE (KSAME(1,1,1),IB(14,1)),(LIGAND(1,1,1),IMATX(2,1)),    SETUP
     1(PERTKC(1,1),NBX(2,5)),(PERLIG(1,1),NBX(42,11)),(MRM(1),IB(62,5)), SETUP
     2(NOSAME(1,1),GRID(4,54)),(NTOTAL(1,1),GRID(4,70)),(MRS1(1),        SETUP
     3NBX(20,20)),(MRS2(1),NBX(24,20))                                   SETUP
C1.0     INITIALIZE VARIABLES.                                          SETUP
      KCX=KCXDO1(1)                                                      SETUP
      DO1   K=1,9                                                        SETUP
      NOSAME(K,KCX)=0                                                    SETUP
      NTOTAL(K,KCX)=0                                                    SETUP
      DO1   J=1,9                                                        SETUP
      KSAME(J,K,1)=0                                                     SETUP
      KSAME(J,K,2)=0                                                     SETUP
    1 CONTINUE                                                           SETUP
      DO100   K=1,KCC                                                    SETUP
  100 MRM(K)=0                                                           SETUP
C2.0     ARE LIGANDS TO BE COMPARED BONDED TO SAME OR DIFFERENT CORE     SETUP
C        ATOMS                                                          SETUP
      IF(KCXDO1(1).EQ.KCXDO2(1))GO TO 3                                  SETUP  3
C           DIFFERENT. SET ARRAYS ACCORDINGLY.                          SETUP
      DO2   K=1,9                                                        SETUP
      MRS1(K)=K                                                          SETUP
      MRS2(K)=K+9                                                        SETUP
    2 CONTINUE                                                           SETUP
      GO TO 5                                                            SETUP  5
C           SAME. SET ARRAYS ACCORDINGLY.                               SETUP
    3 DO4   K=1,9                                                        SETUP
      MRS1(K)=K                                                          SETUP
      MRS2(K)=K                                                          SETUP
    4 CONTINUE                                                           SETUP
    5 KY=0                                                               SETUP
      II=1                                                               SETUP
C3.0     START EXECUTION OF CYCLE THAT ESTABLISHES THE SIMILARITIES OF   SETUP
C        THE VARIOUS LIGANDS.                                           SETUP
      DO27   M=1,NODO1                                                   SETUP
      KC1=KCXDO1(M)                                                      SETUP
      NI=1+MULT1*M                                                       SETUP
      DO27   N=NI,NODO2                                                  SETUP
      KC2=KCXDO2(N)                                                      SETUP
      DO23   L=1,LF                                                      SETUP
      LG1=PERLIG(L,KC1)                                                  SETUP
C           K1 IS A LIGAND BONDED TO KC1.                               SETUP
      K1=PERTKC(LG1,KC1)                                                 SETUP
      KI=1+MULT2*L                                                       SETUP
      DO23   K=KI,KF                                                     SETUP
      LG2=PERLIG(K,KC2)                                                  SETUP
C           K2 IS A LIGAND BONDED TO KC2.                               SETUP
      K2=PERTKC(LG2,KC2)                                                 SETUP
      IF(K1.EQ.KC2.OR.K2.EQ.KC1)GO TO 23                                 SETUP 23
C           NEITHER K1 NOR K2 ARE EQUAL TO THE CORE ATOM TO WHICH THE   SETUP
C           OTHER IS BONDED.                                            SETUP
      KEQUAL=0                                                           SETUP
C           CHECK WHETHER THE ATOMIC COMPOSITIONS OF LIGANDS K1 AND K2  SETUP
C           ARE EQUAL.                                                  SETUP
      DO7   J=1,NDATM                                                    SETUP
      IF(LIGAND(J,LG1,KC1).NE.LIGAND(J,LG2,KC2))GO TO 7                  SETUP  7
      KEQUAL=KEQUAL+1                                                    SETUP
```

17 APR 73 6:02:36



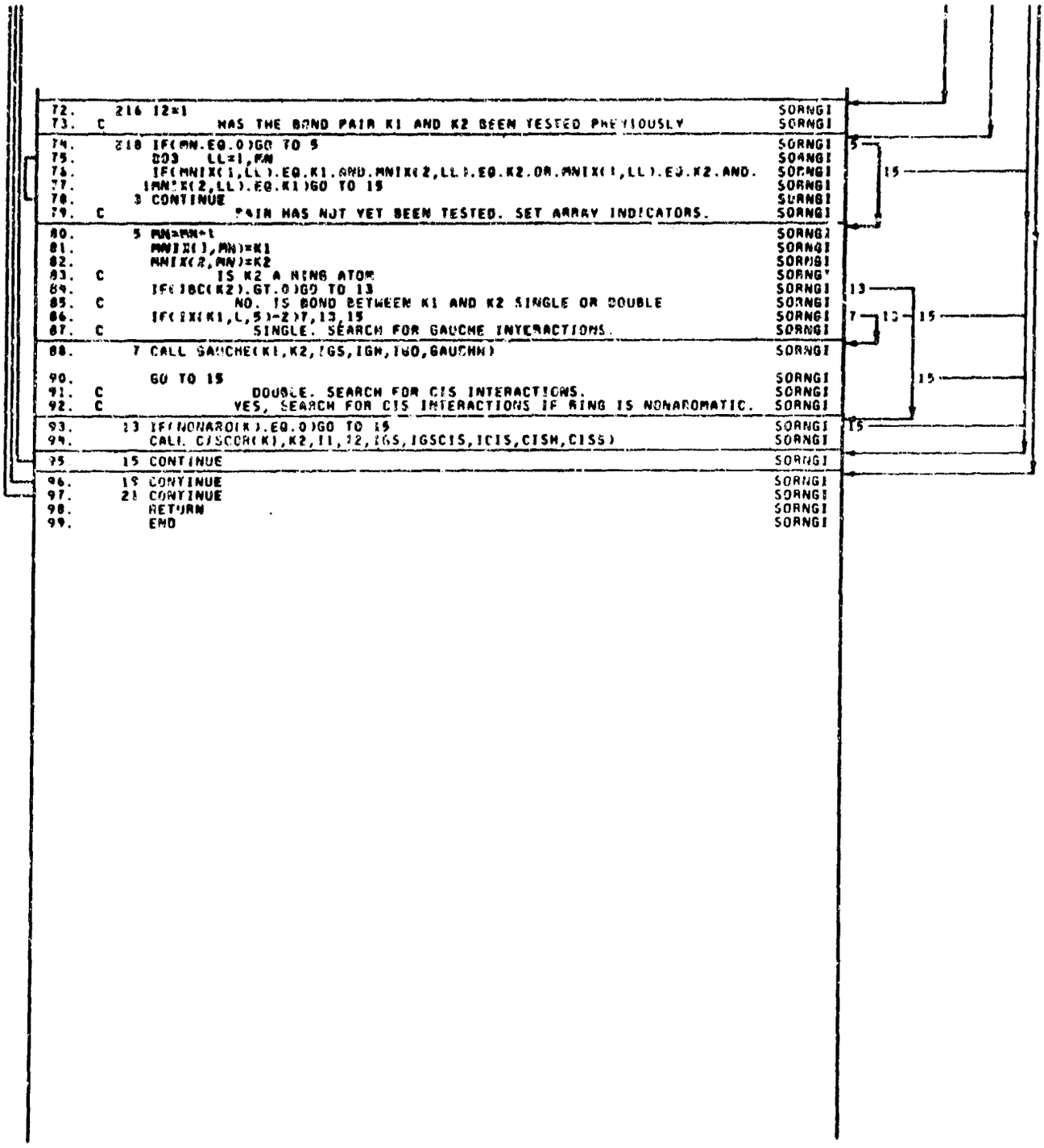
```
      7 CONTINUE                                                        SETUP
        IF(KEQUAL.NE.5)GO TO 23                                         SETUP
C                YES THEY ARE.                                          SETUP
        IF(NOD02.EQ.1)GO TO 8                                           SETUP
C                TWO OR MORE CENTRAL ATOMS OF TYPE KC2 ARE BEING USED.  SETUP
        J1=M                                                            SETUP
        J2=N                                                            SETUP
        GO TO 9                                                         SETUP
C                   ONLY ONE CENTRAL ATOM OF TYPE KC2 IS BEING USED.    SETUP
      8 J1=L                                                            SETUP
        J2=K                                                            SETUP
      9 IF(K1.GT.0.AND.K2.GT.0)GO TO 11                                 SETUP
C           K1 AND K2 ARE NOT BOTH CORE ATOMS.                          SETUP
        IF(K1*K2.EQ.0.OR.K1.NE.K2)GO TO 23                              SETUP
C           K1 AND K2 ARE BOTH IDENTICAL NON-CORE ATOMS. VERIFY THEIR   SETUP
C           SINGULARITY AND STORE THEIR IDENTIFICATION VARIABLES.       SETUP
        CALL SAME(KCX,K1,K2,MRS1(J1),MRS2(J2))                          SETUP
        GO TO 23                                                        SETUP
C           K1 AND K2 ARE LIGANDS WITH COMPLEX STRUCTURES. TO WHICH     SETUP
C           IDENTITY SET DO K1 AND K2 BELONG                            SETUP
     11 IF(MRM(K1).EQ.0)GO TO 13                                        SETUP
C              K1 FOUND IDENTICAL PREVIOUSLY. RETRIEVE ITS IDENTITY SET SETUP
C              NUMBER.                                                  SETUP
        II=MRM(K1)                                                      SETUP
        GO TO 19                                                        SETUP
     13 IF(MRM(K2).EQ.0)GO TO 15                                        SETUP
C              K2 FOUND IDENTICAL PREVIOUSLY. RETRIEVE ITS IDENTITY SET SETUP
C              NUMBER.                                                  SETUP
        II=MRM(K2)                                                      SETUP
        GO TO 17                                                        SETUP
C              NEITHER K1 NOR K2 WERE FOUND IDENTICAL PREVIOUSLY. ASSIGN SETUP
C              THEM A NEW IDENTITY SET NUMBER = KV.                     SETUP
     15 KV=KV+1                                                         SETUP
        II=KV                                                           SETUP
     17 MRM(K1)=II                                                      SETUP
     19 MRM(K2)=II                                                      SETUP
        KSAME(J1,II,1)=1                                                SETUP
        KSAME(J2,II,2)=1                                                SETUP
     23 CONTINUE                                                        SETUP
     27 CONTINUE                                                        SETUP
C*.0       DETERMINE WHETHER COMPLEX LIGANDS ON KC1 AND KC2 ARE IDENTICAL. SETUP
        IF(KV.GT.0)CALL EQUAL(KCXD01,KCXD02,NOD01,NOD02,MULT1,LF,KF,    SETUP
       1MULT2,KV,0)                                                     SETUP
        RETURN                                                          SETUP
        END                                                             SETUP
```

17 APR 73 6.02-36

```
      SUBROUTINE SHIFT(JC,KC,MFLAG)                                     SHIFT
C        THIS SUBROUTINE SWITCHES DATA OF LIGAND CONTAINED IN COLUMN     SHIFT
C        J=2 OF IX(JC,J,I) WITH DATA OF LIGAND CORRESPONDONG TO CORE     SHIFT
C        ATOM KC.                                                        SHIFT
      INTEGER WEIGHT(9)                                                 SHIFT
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC     SHIFT
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NW(100),IBC(100),KON(100),  SHIFT
     1IOBR(100),IB(100,6),ING,MOBR                                      SHIFT
C1.0     SET IX(JC,1,4) EQUAL TO NEW PARENT CORE ATOM.                  SHIFT
      IX(JC,1,4)=IX(KC,1,6)                                             SHIFT
C2.0     FIND LOCATION J IN IX(JC,J,I) CONTAINING CORE ATOM DATA. THEN  SHIFT
C        SHIFT DATA FROM COLUMN J=2 TO J.                                SHIFT
      DO33   J=3,5                                                      SHIFT
      IF(IX(JC,J,6).NE.IX(JC,1,4))GO TO 33                              SHIFT   33
      DO32   LL=1,6                                                     SHIFT
      IX(JC,J,LL)=IX(JC,2,LL)                                           SHIFT
   32 CONTINUE                                                          SHIFT
      GO TO 39                                                          SHIFT   39
   33 CONTINUE                                                          SHIFT
C3.0     NOW STORE DATA FOR CORE ATOM IX(KC,1,6) INTO J=2 COLUMN        SHIFT
C        OF IX(JC,J,I). IN SO DOING, MAKE APPROPRIATE CHANGES FOR        SHIFT
C        VECTOR DIRECTION.                                               SHIFT
   39 DO34   J=1,3                                                      SHIFT
   34 IX(JC,2,J)=IX(KC  ,1,J)                                           SHIFT
      KS=IX(JC,1,2)                                                     SHIFT
      LS=IX(JC,1,3)                                                     SHIFT
      DO36   L=3,5                                                      SHIFT
      IF(IX(KC  ,L,2).EQ.KS.AND.IX(KC  ,L,3).EQ.LS)GO TO 38             SHIFT   38
   36 CONTINUE                                                          SHIFT
   38 IX(JC,2,4)=IX(KC  ,L,4) + 4                                       SHIFT
      IF(IX(JC,2,4).GT.8)IX(JC,2,4)=IX(JC,2,4)-8                        SHIFT
      IX(JC,2,5)=IX(KC  ,L,5)                                           SHIFT
      IX(JC,2,6)=IX(KC  ,1,6)                                           SHIFT
C4.0     RESET VALUES OF IB PERTAINING TO JC IF MFLAG IS NOT ZERO.      SHIFT
C        MFLAG IS NOT ZERO IN FIRST SECTION OF PROGRAM.                  SHIFT
      IF(MFLAG.EQ.0)RETURN                                              SHIFT
      IB(JC,1)=IX(JC,3,6)                                               SHIFT
      IB(JC,2)=IX(JC,4,6)                                               SHIFT
      IB(JC,3)=IX(JC,5,6)                                               SHIFT
      RETURN                                                            SHIFT
      END                                                               SHIFT
```

```
      SUBROUTINE SORNGI(IGS,IGM,IGO,IBSCIS,ICIS,CAUCHH,CISH,CISS)
C        THIS SUBROUTINE CONTROLS THE SEARCH FOR GAUCHE AND CIS
C        INTERACTIONS FOR THE CASE WHERE ONE OF THE TWO CENTRAL ATOMS
C        (K1 OR K2) IS A RING ATOM AND THE OTHER IS NOT. THESE CIS
C        INTERACTIONS APPLY ONLY TO NONAROMATIC RINGS. THE GAUCHE
C        INTERACTIONS APPLY TO BOTH AROMATIC AND NONAROMATIC RINGS.
      INTEGER SYMX(9),SYMBOL(9),GRID(50,30)
      INTEGER WEIGHT(9)
      DIMENSION NONARO(40),MNIX(2,200)
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NUMBL(9),GRID
      COMMON/BLK2/WEIGHT,AWGT(9),MOLWT(100),IX(100,5,6),NC(100),NCC
      COMMON/BLK3/IRING(40,30),IMATX(90,80),NWI(100),IBC(100),KONC(100),
     1)DBR(100),IB(100,8),IRG,NOBR
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT
      EQUIVALENCE (NONARO(1),IMATX(2,49)),(MNIX(1,1),GRID(9,1))
C1.0     INITIALIZE ARRAY.
      KF=2*KCC
      DO1   K=1,KF
      MNIX(1,K)=0
      MNIX(2,K)=0
    1 CONTINUE
      MN=0
C2.0     START EXECUTION OF MAIN CYCLE THAT SEARCHES FOR GAUCHE AND CIS
C        INTERACTIONS BETWEEN LIGANDS IN EACH NON-AROMATIC RING SYSTEM.
      DO21   K=1,IRC
      JF=IRING(K,1)+1
C           TEST COMPONENTS AND NON-RING LIGANDS OF RING K.
      DO19   J=2,JF
      K1=IRING(K,J)
      LF=NC(K1)+1
C           DETERMINE IF K1 HAS AT LEAST ONE NON-RING LIGAND. IF NOT,
C           FIND NEW K1.
      ISUM=0
      ITEST=0
      DO2   L=2,LF
      K2=IX(K1,L,5)
      IF(IBC(K2).GT.0)GO TO 190
      ITEST=1
      GO TO 2
  190 ISUM=ISUM+1
    2 CONTINUE
      IF(KONC(K1).LE.NC(K1).AND.ITEST.EQ.0)GO TO 19
C           SET STORAGE INDICATOR OF K1 WHICH IS USED IN SUBROUTINE CIS.
      II=0
      IF(KONC(K1).LE.3)GO TO 209
      IF(ISUM-3)200,202,19
  200 II=-1
      GO TO 209
  202 II=1
C           EXAMINE CORE LIGANDS OF RING ATOM K1
  209 DO15   L=2,LF
      K2=IX(K1,L,6)
C           DETERMINE IF K2 HAS AT LEAST ONE NON-RING LIGAND. IF NOT,
C           FIND NEW K2.
      ISUM=0
      ITEST=0
      MF=NC(K2)+1
      DO212   M=2,MF
      K3=IX(K2,M,5)
      IF(IBC(K3).GT.0)GO TO 210
      ITEST=1
      GO TO 212
  210 ISUM=ISUM+1
  212 CONTINUE
      IF(KONC(K2).LE.NC(K2).AND.ITEST.EQ.0)GO TO 15
C           SET STORAGE INDICATOR OF K2 WHICH IS USED IN SUBROUTINE CIS.
      I2=0
      IF(KONC(K2).LE.3)GO TO 218
      IF(ISUM-3)219,216,15
  216 I2=-1
      GO TO 218
```


17 APR 73 0:02-36



```
      216 I2=1                                                          SORNGI
C               HAS THE BOND PAIR K1 AND K2 BEEN TESTED PREVIOUSLY      SORNGI
      218 IF(MN.EQ.0)GO TO 5                                            SORNGI
          DO3    LL=1,MN                                                SORNGI
          IF(MNIX(1,LL).EQ.K1.AND.MNIX(2,LL).EQ.K2.OR.MNIX(1,LL).EQ.K2.AND.  SORNGI
         1MNIX(2,LL).EQ.K1)GO TO 15                                     SORNGI
        3 CONTINUE                                                      SORNGI
C                       PAIR HAS NOT YET BEEN TESTED. SET ARRAY INDICATORS.  SORNGI
        5 MN=MN+1                                                       SORNGI
          MNIX(1,MN)=K1                                                 SORNGI
          MNIX(2,MN)=K2                                                 SORNGI
C                 IS K2 A RING ATOM                                     SORNGI
          IF(IBC(K2).GT.0)GO TO 13                                      SORNGI
C                       NO. IS BOND BETWEEN K1 AND K2 SINGLE OR DOUBLE  SORNGI
          IF(IX(K1,L,5)-2)7,13,15                                       SORNGI
C                     SINGLE. SEARCH FOR GAUCHE INTERACTIONS.           SORNGI
        7 CALL GAUCHE(K1,K2,IGS,IGM,IGO,GAUCHN)                         SORNGI
          GO TO 15                                                      SORNGI
C                     DOUBLE. SEARCH FOR CIS INTERACTIONS.              SORNGI
C                   YES, SEARCH FOR CIS INTERACTIONS IF RING IS NONAROMATIC.  SORNGI
       13 IF(NONARO(K).EQ.0)GO TO 15                                    SORNGI
          CALL CISCOR(K1,K2,I1,I2,IGS,IGSCIS,ICIS,CISM,CISS)            SORNGI
       15 CONTINUE                                                      SORNGI
       19 CONTINUE                                                      SORNGI
       21 CONTINUE                                                      SORNGI
          RETURN                                                        SORNGI
          END                                                           SORNGI
```

17 APR 73 0.02-36

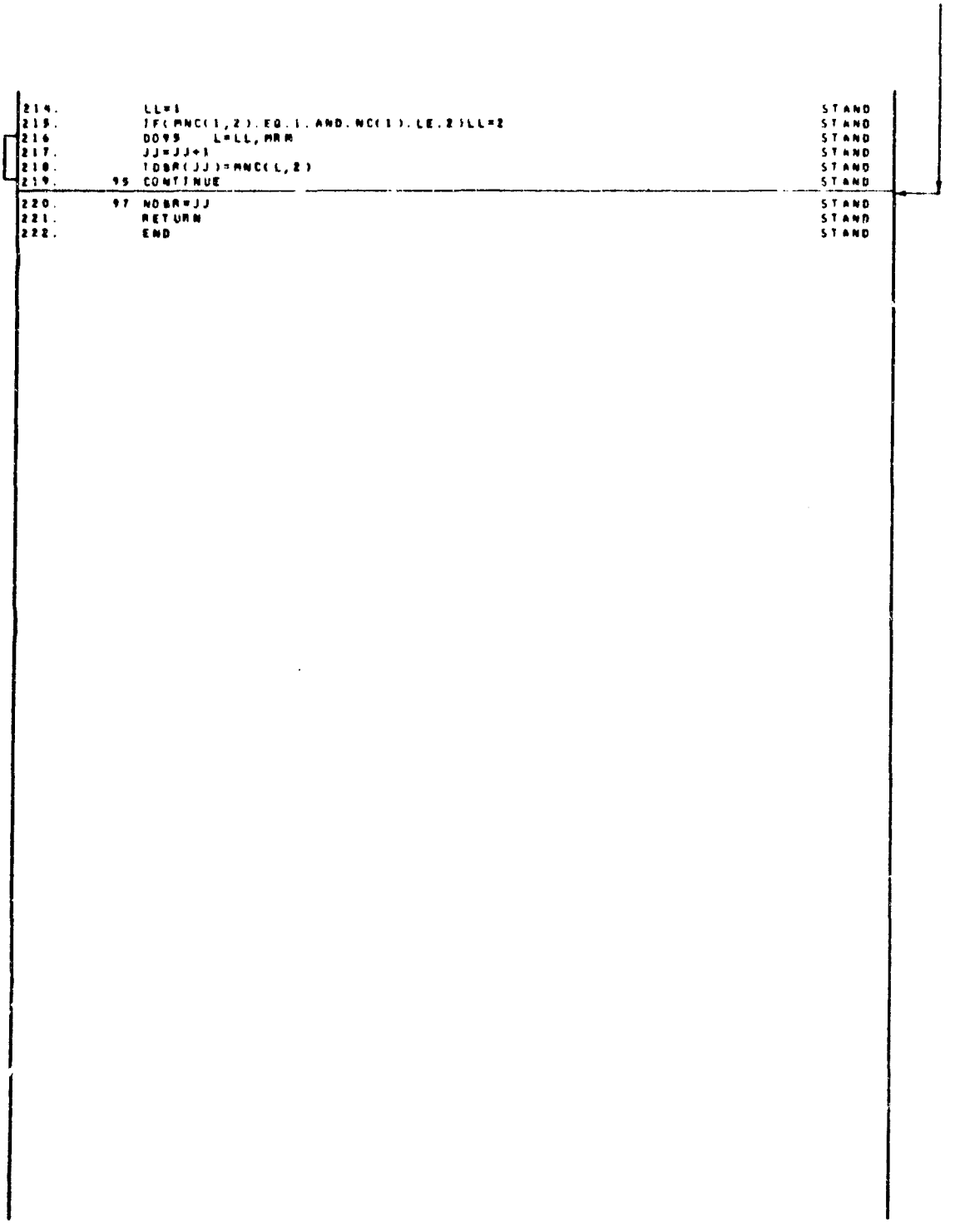
```
      SUBROUTINE STAND(IERR)                                                STAND
C        THIS SUBROUTINE IS THE CONTROL ELEMENT FOR SECTION ONE OF THE      STAND
C        PROGRAM. THIS SECTION IDENTIFIES THE GROUP, AND GROUP              STAND
C        COMPONENTS OF THE MOLECULE AS WELL AS THE ATOMIC COORDINATES,       STAND
C        BOND VECTORS AND BOND TYPES. THE GROUP WEIGHTS ARE ALSO            STAND
C        ASSIGNED. FOR SOME UNSATURATED RING COMPOUNDS THE LATTER IS        STAND
C        ALTERED IN SECTION THREE OF THE PROGRAM.                           STAND
      INTEGER SYMB(4),SYMBOL(9),GRID(50,80)                                 STAND
      INTEGER WEIGHT(9)                                                     STAND
      DIMENSION RNC(100,2),ICNC(100)                                        STAND
      COMMON/BLK1/NO,NOS,SYMB SYMBOL,NOVAL(9),GRID                          STAND
      COMMON/BLK2/WEIGHT,RWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC         STAND
      COMMON/BLK3/IRING(40,30),IMAT(50,80),NM(100),IBL(100),KON(100),       STAND
     1IORR(100),IB(100,8),IRG NOBR                                          STAND
      COMMON/BLK4/NBC(60,50),ADS(60,2),NBX(60,20),IRC,NONPOS,IRCTOT         STAND
      COMMON/BLK5/NOATM,NUMATM(5),MBC(50),MBS(2),JW,JY,LFLAGS,LFLAGG        STAND
      EQUIVALENCE (MRM,NBC(3,1)),(RNC(1,1),NBC(4,1)),(ISUB,NBC(2,7)),       STAND
     1(ICNC(1),NBC(3,7))                                                    STAND
C1.0      INITIALIZE VARIABLES.                                             STAND
      MRM=0                                                                 STAND
      IRG=0                                                                 STAND
      IY2=6                                                                 STAND
C           CONTENTS OF IX ARRAY. IX(M,N,J) CONTAINS GRID DATA FOR CORE     STAND
C           ATOM(M) AT N=1 AND FOR LIGANDS AT N=2 TO N=5. CONTENTS OF       STAND
C           J COLUMN - J=1 CHEMICAL SYMBOL OF COMPONENT, J=2 GRID ROW       STAND
C           COORDINATE, J=3 GRID COLUMN COORDINATE, N=1 AND J=4 GROUP       STAND
C           NUMBER OF PARENT CORE ATOM, N NOT 1 AND J=4 DIRECTION OF        STAND
C           BOND FROM CORE ATOM TO LIGAND, J=5 BOND TYPE, J=6 GROUP         STAND
C           NUMBER OF CORE ATOM.                                            STAND
      DO 5    M=1,100                                                       STAND
      DO 1    N=1,5                                                         STAND
      DO 1    J=1,6                                                         STAND
    1 IX(M,N,J)=0                                                           STAND
      DO 3    J=1,8                                                         STAND
    3 IB(M,J)=0                                                             STAND
      MOLWT(M)=0                                                            STAND
      NC(M)=0                                                               STAND
    5 CONTINUE                                                              STAND
      DO 7    L=1,80                                                        STAND
      DO 7    K=1,50                                                        STAND
    7 IMAT(K,L)=0                                                           STAND
      DO 9    L=1,5                                                         STAND
      NUMATM(L)=0                                                           STAND
    9 CONTINUE                                                              STAND
      IRC=0                                                                 STAND
C2.0      FIND FIRST NON-BLANK CHARACTER ON INPUT CARD. PROCEED TO 17.       STAND
      DO 11   L=1,80                                                        STAND
      DO 11   K=1,NO                                                        STAND
      IF(GRID(K,L).NE.SYMB(N))GO TO 17                                      STAND
   11 CONTINUE                                                              STAND
C           PRINT ERROR MESSAGE, SET ERROR FLAG, AND TERMINATE CASE         STAND
C           CALCULATION.                                                    STAND
   12 WRITE(6,13)                                                           STAND
   13 FORMAT(// 1HO,39*,50HSTRUCTURE INPUT ARRAY INCORRECT   CASE TERMIN   STAND
     1ATED.)                                                                STAND
   19 IERR=1                                                                STAND
      RETURN                                                                STAND
C3.0      IS CHARACTER A CHEMICAL SYMBOL                                    STAND
   17 DO 23   L=1,8                                                         STAND
      IF(GRID(K,L)-SYMB(LL))23,19,23                                        STAND
C           NO, MOLECULE IS A FREE RADICAL. PRINT OUT ERROR MESSAGE         STAND
   19 WRITE(6,21)K,L                                                        STAND
   21 FORMAT(// 1HO,*3X,32HFREE RADICAL AT GRID COORDINATES,I4,1H,,I4,      STAND
     1 1H.)                                                                 STAND
      GO TO 18                                                              STAND
C           YES, CHARACTER IS A CHEMICAL SYMBOL. NOW IDENTIFY IT.           STAND
   23 CONTINUE                                                              STAND
      NUM=3                                                                 STAND
      CALL IDENT(NUM,K,L,M,IER)                                             STAND
      IF(IER.EQ.1)RETURN                                                    STAND
      K(M)                                                                  STAND
      KCC=KCC+1                                                             STAND
      MM=2                                                                  STAND
```

```
      C4.0     DETERMINE IF FIRST CHEMICAL SYMBOL IS A CORE ATOM. IF NOT, FIND STAND
      C          CORE ATOM(KC).                                              STAND
               CALL FIND(KC,R,L,M,NUM,MR,IERR)                               STAND
               IF(IERR.EQ.1)RETURN                                           STAND
      C            SET VARIABLES IN IX ARRAY.                                STAND
               IX(KC,1,1)=M                                                  STAND
               IX(KC,1,2)=R                                                  STAND
               IX(KC,1,3)=L                                                  STAND
               IX(KC,1,4)=KC                                                 STAND
               IHATM(R,L)=KC                                                 STAND
      C5.0     IDENTIFY LIGANDS BONDED TO CORE ATOM KC.                      STAND
               CALL SCAN(KC,MM,MR,NUM,R,L,IERR)                              STAND
               IF(IERR.EQ.1)RETURN                                           STAND
               IX(KC,1,4)=IX(KC,2,4)                                         STAND
      C6.0     FIND PRIMARY WEIGHT OF GROUP KC.                              STAND
               CALL DELTA1(KC)                                               STAND
               ISUM=1                                                        STAND
      C7.0     SET VARIABLES DEPENDING ON BONDING PROPERTIES OF KC.          STAND
               IF(NC(KC)-1)12,29,27                                          STAND
      C          KC HAS TWO OR MORE CORE LIGANDS.                            STAND
            27 MRM=MRM+1                                                     STAND
               ANC(MRM,1)=NC(KC)-1                                           STAND
               ANC(MRM,2)=KC                                                 STAND
      C          KC IS A CORE TERMINAL ATOM.                                 STAND
            29 ICN(IS(5))=KC                                                 STAND
      C8.0     INITIALIZE SCAN VARIABLES.                                    STAND
               NM=1                                                          STAND
               NN=1                                                          STAND
               GO TO 35                                                      STAND
            31 NN=2                                                          STAND
            35 JF=NC(KC)                                                     STAND
      C9.0     START PROCESSING CYCLE THAT IDENTIFIES REMAINDER OF MOLECULE. STAND
               DO 43   JJ=1,JF                                               STAND
               NN=NN+1                                                       STAND
               JC=IA(KC,JJ)                                                  STAND
      C          HAS CORE ATOM JC BEEN SCANNED BEFORE                        STAND
               IF(IX(JC,1,5).NE.1000)GO TO 37                                STAND
      C              YES, RING STRUCTURE IS PRESENT. SET ITZ.                STAND
               ITZ=1                                                         STAND
               GO TO 41                                                      STAND
      C              NO, RING STRUCTURE NOT YET EVIDENT. IDENTIFY LIGANDS OF JC STAND
            37 MM=3                                                          STAND
               CALL SCAN(JC,MM,MR,IX(KC,NN,4),IX(KC,NN,2),IX(KC,NN,3),IERR)  STAND
               IF(IERR.EQ.1)RETURN                                           STAND
      C          IF KC IS NOT IN PARENT CORE ATOM POSITION, PUT IT SO.       STAND
            41 IF(IX(JC,1,4).NE.IX(KC,1,4))CALL SHIFT(JC,KC,1)               STAND
      C          IF JC IS TERMINAL ATOM, GO TO 43. OTHERWISE, ADD SECOND-ORDER STAND
      C          WEIGHT CORRECTIONS.                                         STAND
               IF(NC(JC).LT.1)GO TO 43                                       STAND
               CALL DELTA2(KC,JC)                                            STAND
            43 CONTINUE                                                      STAND
      C10.0    FIND NEW ATOM TO BE PROCESSED.                                STAND
               KCPV=KC                                                       STAND
               JF=NC(KC)                                                     STAND
      C          DETERMINE IF CORE ATOM KC HAS BEEN PROCESSED BEFORE.        STAND
               DO 49   JA=1,JF                                               STAND
               KC=IA(KCPV,JA)                                                STAND
               DO 47   LL=1,ISUM                                             STAND
               IF(KC.EQ.ICN(LL))GO TO 49                                     STAND
            47 CONTINUE                                                      STAND
               GO TO 51                                                      STAND
      C          YES IT HAS. FIND NEW KC. IF NONE, FIND NEW BRANCH ATOM      STAND
            49 CONTINUE                                                      STAND
               ITZ=0                                                         STAND
               GO TO 63                                                      STAND
      C              NO IT HAS NOT.                                          STAND
      C          IF KCPV NOT IN PARENT CORE ATOM POSITION, PUT IT SO         STAND
            51 IF(IX(KC,1,4).NE.IX(KCPV,1,4))CALL SHIFT(KC,KCPV,1)           STAND
      C          DETERMINE WEIGHT OF GROUP KC.                               STAND
               CALL DELTA1(KC)                                               STAND
               CALL DELTA2(KC,IX(KC,1,4))                                    STAND
```

17 FEB 73 0 02-38

```
      ISUB=ISUB+1                                                       STAND
C           SET VARIABLES DEPENDING UPON BONDING PROPERTIES OF KC.      STAND
      IF(NC(KC)-2)57,55,53                                              STAND
C              KC IS BRANCH ATOM.                                       STAND
   53 MAM=MAM+1                                                         STAND
      MNC(MAM,1)=NC(KC)-2                                               STAND
      MNC(MAM,2)=KC                                                     STAND
C              KC IS CHAIN ATOM. RESTART CYCLE.                         STAND
   55 ICH(ISUB)=KC                                                      STAND
      GO TO 51                                                          STAND
C              KC IS TERMINAL CORE ATOM.                                STAND
   57 IF(ITZ.EQ.0)GO TO 61                                              STAND
      ITZ=0                                                             STAND
      IF(IX(KC,1,4).NE.IX(KCPV,1,6))CALL SHIFT(KC,KCPV,1)               STAND
      GO TO 55                                                          STAND
   61 ICH(ISUB)=KC                                                      STAND
C11.0    FIND NEW BRANCH ATOM.                                          STAND
C            IF NO BRANCH ATOM PRESENT, GO TO 85                        STAND
   63 IF(MAM.EQ.0)GO TO 85                                              STAND
C            FIND POTENTIAL BRANCH ATOM. IF NONE PRESENT, GO TO 85.     STAND
   65 DO67   J=1,MAM                                                    STAND
      IF(MNC(J,1).NE.0)GO TO 69                                         STAND
   67 CONTINUE                                                          STAND
      GO TO 85                                                          STAND
   69 MNC(J,1)=MNC(J,1)-1                                               STAND
      KC=MNC(J,2)                                                       STAND
      KCPV=KC                                                           STAND
C           FIND CORE ATOM BONDED TO BRANCH ATOM WHICH HAS NOT YET      STAND
C           BEEN SCANNED AND TRANSFER TO 51. IF NONE, FIND ANOTHER      STAND
C           BRANCH ATOM.                                                STAND
      JF=NC(KC)                                                         STAND
      DO74   JA=1,JF                                                    STAND
      KC=ICB(KCPV,JA)                                                   STAND
      DO72   LL=1,ISUB                                                  STAND
      IF(KC.EQ.ICH(LL))GO TO 74                                         STAND
   72 CONTINUE                                                          STAND
      IF(KC.EQ.0)GO TO 65                                               STAND
      GO TO 51                                                          STAND
   74 CONTINUE                                                          STAND
      GO TO 65                                                          STAND
C12.0    SET NC AND NW COUNTER ARRAYS AND WEIGHT CONNECTIONS TO CORE    STAND
C        ATOMS. CHECK VALENCE OF ALL ATOMS.                             STAND
   85 DO93   KC=1,NCC                                                   STAND
      NW(KC)=0                                                          STAND
      JC=0                                                              STAND
      DO89   L=2,5                                                      STAND
      JC=JC+IX(KC,L,5)                                                  STAND
      IF(IX(KC,L,6).EQ.0)GO TO 86                                       STAND
      NW(KC)=NW(KC)+1                                                   STAND
      GO TO 87                                                          STAND
   86 IF(IX(KC,L,1).EQ.0)GO TO 91                                       STAND
      JJ=IX(KC,L,1)                                                     STAND
      IF(IX(KC,L,5).NE.NOVAL(JJ))GO TO 12                               STAND
C              NON-CORE ATOM BONDED TO KC HAS CORRECT VALENCE.          STAND
   87 NON(KC)=L-1                                                       STAND
   89 CONTINUE                                                          STAND
   91 JJ=IX(KC,1,1)                                                     STAND
      IF(JC.NE.NOVAL(JJ))GO TO 12                                       STAND
C              CORE ATOM KC HAS CORRECT VALENCE.                        STAND
      NC(KC)=NW(KC)                                                     STAND
      IF(IX(KC,1,1).EQ.6)NW(KC)=NW(KC)+1                                STAND
      MX=IX(KC,1,1)                                                     STAND
      MOLWT(KC)=MOLWT(KC)+MWST(MX)                                      STAND
      IX(KC,1,5)=0                                                      STAND
   93 CONTINUE                                                          STAND
C13.0    STORE GROUP NUMBER OF BRANCH CORE ATOMS IN IBUR AND TOTAL      STAND
C        NUMBER OF ENTRIES IN NOBR.                                     STAND
      JJ=0                                                              STAND
      IF(MAM.EQ.0)GO TO 97                                              STAND
```

19 FEB 73 8.02.39



```
214.            LL=1                                                    STAND
215.            IF(MNC(1,2).EQ.1.AND.NC(1).LE.2)LL=2                    STAND
216             DO95    L=LL,MRM                                        STAND
217.            JJ=JJ+1                                                 STAND
218.            IDBR(JJ)=MNC(L,2)                                       STAND
219.         95 CONTINUE                                                STAND
220.         97 NOBR=JJ                                                 STAND
221.            RETURN                                                  STAND
222.            END                                                     STAND
```

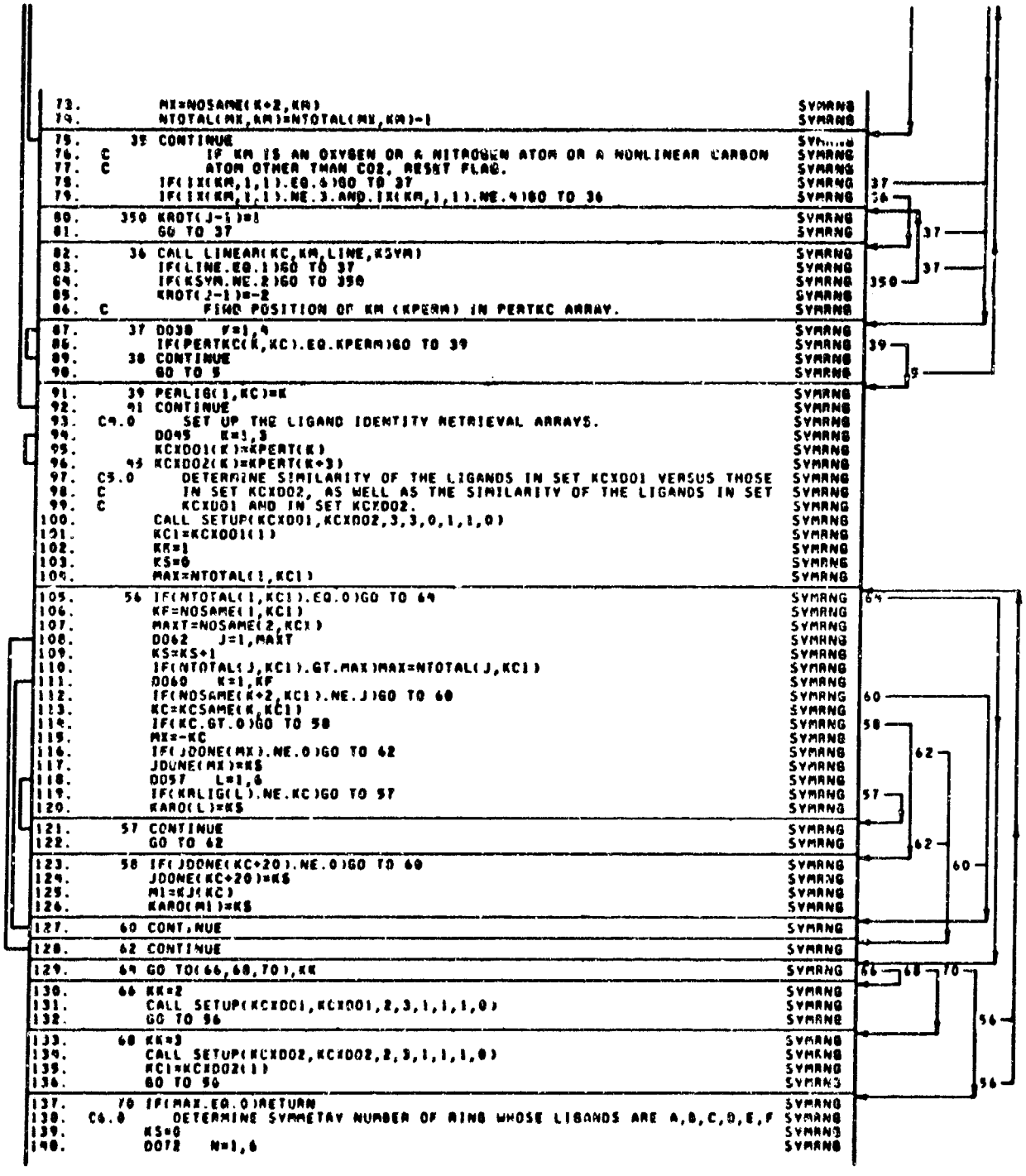
19 FEB 73 8 02:38

```
 1.          SUBROUTINE SUMATM(M,KP,NUMATM)                                  SUMATM
 2.    C        THIS SUBROUTINE ADDS OR SUBTRACTS (WHEN KP=1 OR-1) THE NUMBER SUMATM
 3.    C        OF EACH ELEMENT PRESENT IN THE CHEMICAL SYMBOL REPRESENTED BY SUMATM
 4.    C        M TO (FROM) THE LOCATION ASSIGNED THAT PARTICULAR ELEMENT IN SUMATM
 5.    C        ARRAY NUMATM.                                                 SUMATM
 6.          DIMENSION NUMATM(5)                                             SUMATM
 7.          COMMON/BLK4/NUMFRQ(24)                                          SUMATM
 8.    C1.0     TEST SYMBOL CODE                                            SUMATM
 9.          IF(M.LT.6)GO TO 4                                               SUMATM
10.    C           SYMBOL CODE IS 6 OR GREATER, HENCE IT REPRESENTS MULTI-   SUMATM
11.    C           CHARACTER SPECIES. FIND THE NUMBER OF EACH ELEMENT IT     SUMATM
12.    C           CONTAINS AND ADD (OR SUBTRACT) TO (FROM) CORRESPONDING    SUMATM
13.    C           LOCATION IN NUMATM DEPENDING ON VALUE OF KP.              SUMATM
14.          MX=1+(M-6)*6                                                    SUMATM
15.          DO2    J=1,5,2                                                  SUMATM
16.          MZ=MX+J-1                                                       SUMATM
17.          MY=NUMFRQ(MZ)                                                   SUMATM
18.          NUMATM(MY)=NUMATM(MY)+NUMFRQ(MZ+1)*KP                           SUMATM
19.        2 CONTINUE                                                        SUMATM
20.          RETURN                                                          SUMATM
21.    C           SYMBOL REPRESENTS A SINGLE ELEMENT. ADD KP (EQUAL TO +1 OR SUMATM
22.    C           -1) TO LOCATION OF NUMATM THAT CORRESPONDS TO THIS ELEMENT. SUMATM
23.        4 NUMATM(M)=NUMATM(M)+KP                                          SUMATM
24.          RETURN                                                          SUMATM
25.          END                                                             SUMATM
```

19 FEB 72 8 02-38

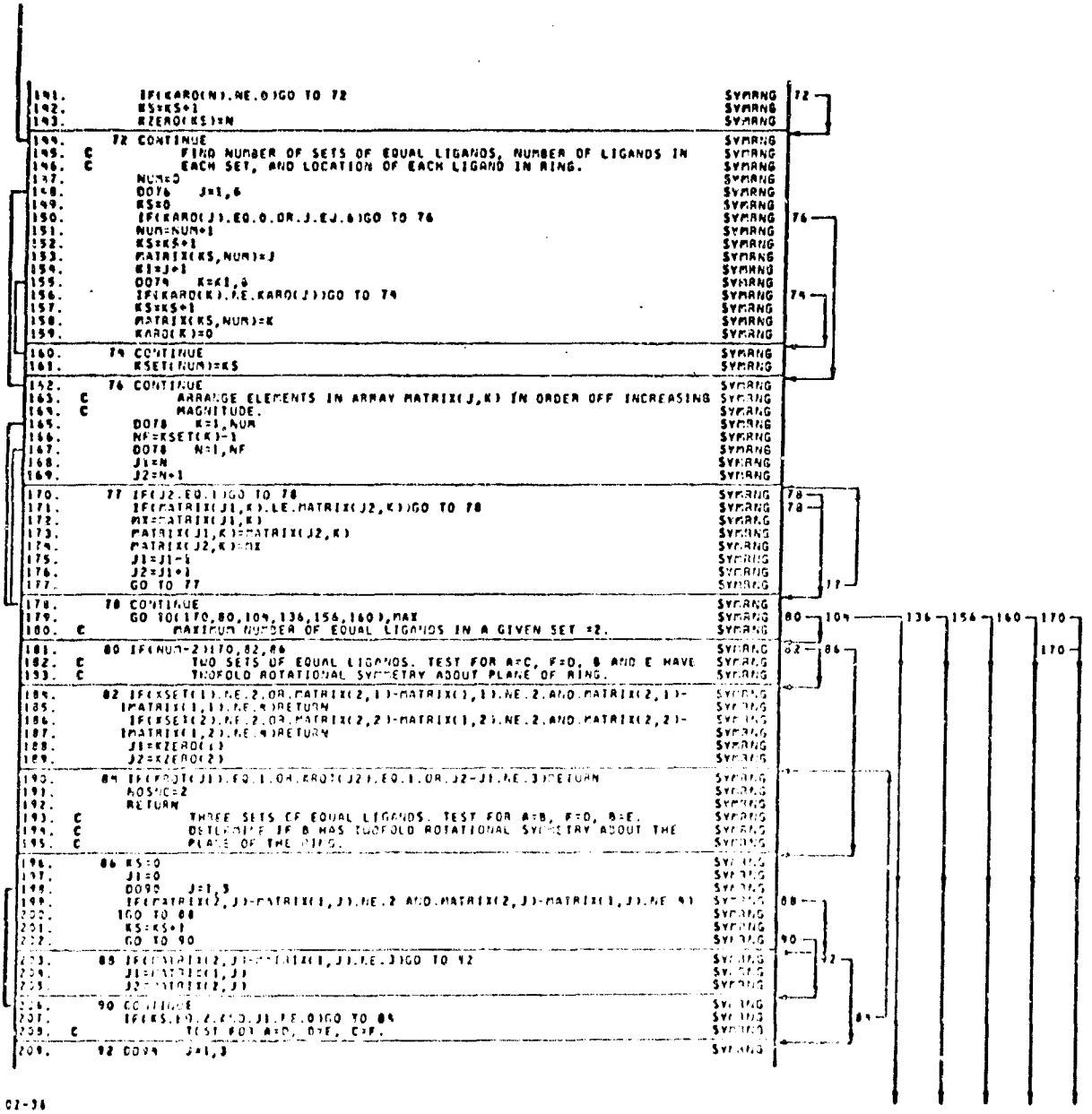
```
      SUBROUTINE SYMRNG(KCX,NOSRC)                                          SYMRNG
C        THIS SUBROUTINE COMPUTES THE EXTERNAL ROTATIONAL SYMMETRY          SYMRNG
C        NUMBER OF AROMATIC, MONOCYCLIC BENZENE-TYPE STRUCTURES.            SYMRNG
      INTEGER SYMX(9),SYMBOL(9),GRID(90,80)                                 SYMRNG
      INTEGER WEIGHT(9)                                                     SYMRNG
      INTEGER PERTKC(9,100),PERLIG(9,100)                                   SYMRNG
      DIMENSION KBENZ(40),KCSAME(6,100),NOSAME(8,100),NTOTAL(3,100),        SYMRNG
     1KJ(100),KPERT(6),KCIDO1(3),KCIDO2(3),KROT(6),MATRIX(6,3),KARO(6),     SYMRNG
     2KZERO(6),KSET(3),JDONE(120),KRLIG(6)                                  SYMRNG
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                          SYMRNG
      COMMON/BLK2/WEIGHT,MWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC         SYMRNG
      COMMON/BLK3/IRING(40,30),IMATX(90,80),NW(100),IBC(100),KON(100),      SYMRNG
     1IDBR(100),IB(100,8),IAG,NDBR                                          SYMRNG
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT         SYMRNG
      COMMON/BLK5/NDATM,NUMATM(5),MBC(50),MBS(2),JW,JV,LFLAGS,LFLAGB        SYMRNG
      EQUIVALENCE (NBENZ,IR(10,1)),(KBENZ(1),IB(53,1)),(KCSAME(1,1),        SYMRNG
     1GRID(9,92)),(NOSAME(1,1),GRID(9,54)),(NTOTAL(1,1),GRID(9,70)),        SYMRNG
     2(PERTKC(1,1),NBX(2,51)),(PERLIG(1,1),NBX(42,11)),(KRUT(1),            SYMRNG
     3NBX(2,20)),(KPERT(1),NBX(8,20)),(MATRIX(1,1),NBX(30,20)),(KARO(1),    SYMRNG
     4NBX(48,20)),(KZERO(1),NBX(54,20)),(KJ(1),NBC(1,49)),(JDONE(1),        SYMRNG
     5NBC(1,48)),(KSET(1),IB(72,8)),(KRLIG(1),NBC(1,47))                    SYMRNG
C1.0     ESYABLISH THE RING PROPERTIES.                                     SYMRNG
      IF(IBC(KCX).LE.IRC.AND.NBENZ.EQ.1)GO TO 7                             SYMRNG  7
C           RING IS FUSED OR/AND BENZENE-TYPE RINGS ARE NOT PRESENT.        SYMRNG
    5 JW=9                                                                  SYMRNG
      RETURN                                                                SYMRNG
C           RING IS NOT FUSED AND BENZENE-TYPE RINGS ARE PRESENT.           SYMRNG
    7 KW=IBC(KCX)                                                           SYMRNG
C           DETERMINE IF ATOM KCX IS A COMPONENT OF ONE OF THE BENZENE-     SYMRNG
C           TYPE RINGS. IF NOT, RETURN TO 5.                                SYMRNG
      DO9   N=1,NBENZ                                                       SYMRNG
      IF(KBENZ(N).EQ.KW)GO TO 13                                            SYMRNG 13
    9 CONTINUE                                                              SYMRNG
      GO TO 5                                                               SYMRNG  5
C           ATOM KCX IS A COMPONENT OF A BENZENE-TYPE RING (KW).            SYMRNG
C2.0     INITIALIZE VARIABLES.                                              SYMRNG
   13 DO14   N=1,6                                                          SYMRNG
      KRLIG(N)=0                                                            SYMRNG
      KARO(N)=0                                                             SYMRNG
   14 KROT(N)=2                                                             SYMRNG
      NF=KCC+20                                                             SYMRNG
      DO15   N=1,NF                                                         SYMRNG
   15 JDONE(N)=0                                                            SYMRNG
      DO16   N=1,3                                                          SYMRNG
   16 KSET(N)=0                                                             SYMRNG
      KS=0                                                                  SYMRNG
C3.0     EXECUTE RING CLASSIFICATION CYCLE.                                 SYMRNG
      DO91   J=2,7                                                          SYMRNG
      KC=IRING(KW,J)                                                        SYMRNG
      KS=KS+1                                                               SYMRNG
      KPERT(KS)=KC                                                          SYMRNG
      IF(NC(KC)-3)17,21,5                                                   SYMRNG 17 21  5
C           NON-RING ATOM BONDED TO RING ATOM KC IS A NON-CORE ATOM.        SYMRNG
C           RESET FLAG IF LIGAND IS NO.                                     SYMRNG
   17 KPERM=-IX(KC,9,1)                                                     SYMRNG
      KRLIG(J-1)=KPERM                                                      SYMRNG
      IF(IX(KC,9,1).EQ.3)KROT(J-1)=1                                        SYMRNG
      GO TO 37                                                              SYMRNG 37
C           NON-RING ATOM BONDED TO RING ATOM KC IS A CORE ATOM. FIND       SYMRNG
C           THIS CORE ATOM (KM) IN IX ARRAY.                                SYMRNG
   21 NF=NC(KC)+1                                                           SYMRNG
      DO27   K=2,NF                                                         SYMRNG
      KM=IX(KC,K,6)                                                         SYMRNG
      IF(IBC(KM).EQ.0)GO TO 33                                              SYMRNG 33
   27 CONTINUE                                                              SYMRNG
   33 KPERM=KM                                                              SYMRNG
      KJ(KM)=J-1                                                            SYMRNG
      KRLIG(J-1)=KM                                                         SYMRNG
C           IF KC IS ONE OF THE IDENTICAL LIGANDS OF ATOM KM, REDUCE THE    SYMRNG
C           TOTAL NUMBER OF IDENTICAL LIGANDS OF KM BY ONE.                 SYMRNG
      DO35   K=1,6                                                          SYMRNG
      IF(KCSAME(K,KM).NE.KC)GO TO 35                                        SYMRNG 35
```

17 APR 73 0:02:56



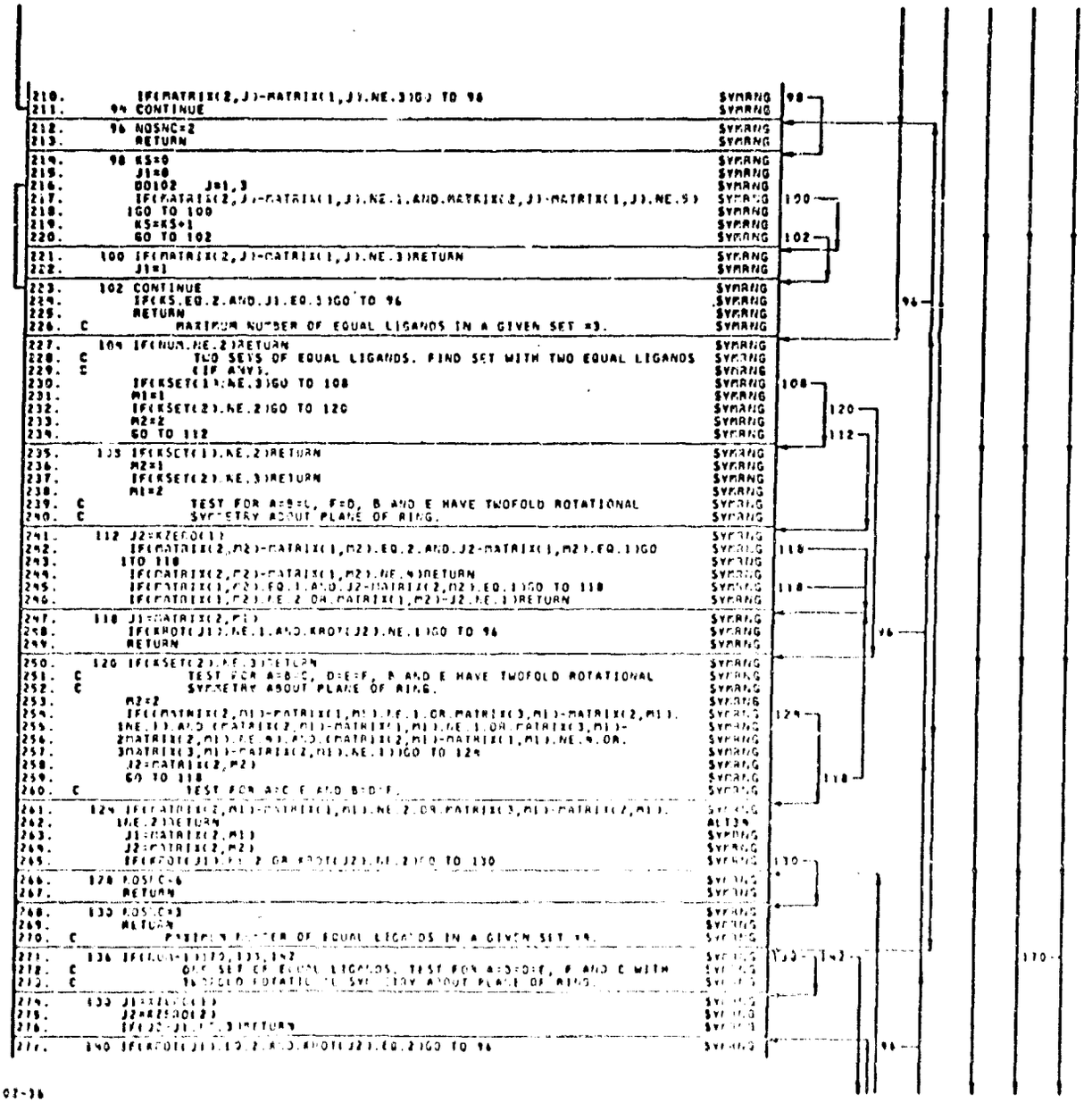
```
 73.          MX=NOSAME(K+2,KM)                                          SYMRNG
 74.          NTOTAL(MX,KM)=NTOTAL(MX,KM)-1                              SYMRNG
 75.       35 CONTINUE                                                   SYMRNG
 76.  C            IF KM IS AN OXYGEN OR A NITROGEN ATOM OR A NONLINEAR CARBON  SYMRNG
 77.  C            ATOM OTHER THAN CO2, RESET FLAG.                     SYMRNG
 78.          IF(IX(KM,1,1).EQ.6)GO TO 37                                SYMRNG  37
 79.          IF(IX(KM,1,1).NE.3.AND.IX(KM,1,1).NE.4)GO TO 36            SYMRNG  36
 80.      350 KROT(J-1)=1                                                SYMRNG
 81.          GO TO 37                                                   SYMRNG  37
 82.       36 CALL LINEAR(KC,KM,LINE,KSYM)                               SYMRNG
 83.          IF(LINE.EQ.1)GO TO 37                                      SYMRNG  37
 84.          IF(KSYM.NE.2)GO TO 350                                     SYMRNG 350
 85.          KROT(J-1)=-2                                               SYMRNG
 86.  C            FIND POSITION OF KM (KPERM) IN PERTKC ARRAY.         SYMRNG
 87.       37 DO38    K=1,4                                             SYMRNG
 88.          IF(PERTKC(K,KC).EQ.KPERM)GO TO 39                          SYMRNG  39
 89.       38 CONTINUE                                                   SYMRNG
 90.          GO TO 5                                                    SYMRNG   5
 91.       39 PERLIG(I,KC)=K                                             SYMRNG
 92.       41 CONTINUE                                                   SYMRNG
 93.  C9.0       SET UP THE LIGAND IDENTITY RETRIEVAL ARRAYS.           SYMRNG
 94.          DO45    K=1,3                                             SYMRNG
 95.          KCXDO1(K)=KPERT(K)                                         SYMRNG
 96.       45 KCXDO2(K)=KPERT(K+3)                                       SYMRNG
 97.  C9.0       DETERMINE SIMILARITY OF THE LIGANDS IN SET KCXDO1 VERSUS THOSE  SYMRNG
 98.  C           IN SET KCXDO2, AS WELL AS THE SIMILARITY OF THE LIGANDS IN SET  SYMRNG
 99.  C           KCXDO1 AND IN SET KCXDO2.                              SYMRNG
100.          CALL SETUP(KCXDO1,KCXDO2,3,3,0,1,1,0)                      SYMRNG
101.          KC1=KCXDO1(1)                                              SYMRNG
102.          KK=1                                                       SYMRNG
103.          KS=0                                                       SYMRNG
104.          MAX=NTOTAL(1,KC1)                                          SYMRNG
105.       56 IF(NTOTAL(1,KC1).EQ.0)GO TO 64                             SYMRNG  64
106.          KF=NOSAME(1,KC1)                                           SYMRNG
107.          MAXT=NOSAME(2,KC1)                                         SYMRNG
108.          DO62    J=1,MAXT                                           SYMRNG
109.          KS=KS+1                                                    SYMRNG
110.          IF(NTOTAL(J,KC1).GT.MAX)MAX=NTOTAL(J,KC1)                  SYMRNG
111.          DO60    K=1,KF                                             SYMRNG
112.          IF(NOSAME(K+2,KC1).NE.J)GO TO 60                           SYMRNG  60
113.          KC=KCSAME(K,KC1)                                           SYMRNG
114.          IF(KC.GT.0)GO TO 58                                        SYMRNG  58
115.          MX=-KC                                                     SYMRNG
116.          IF(JDONE(MX).NE.0)GO TO 62                                 SYMRNG  62
117.          JDONE(MX)=KS                                               SYMRNG
118.          DO57    L=1,6                                              SYMRNG
119.          IF(KRLIG(L).NE.KC)GO TO 57                                 SYMRNG  57
120.          KARO(L)=KS                                                 SYMRNG
121.       57 CONTINUE                                                   SYMRNG
122.          GO TO 62                                                   SYMRNG  62
123.       58 IF(JDONE(KC+20).NE.0)GO TO 60                              SYMRNG  60
124.          JDONE(KC+20)=KS                                            SYMRNG
125.          M1=KJ(KC)                                                  SYMRNG
126.          KARO(M1)=KS                                                SYMRNG
127.       60 CONTINUE                                                   SYMRNG
128.       62 CONTINUE                                                   SYMRNG
129.       64 GO TO(66,68,70),KK                                         SYMRNG  66  68  70
130.       66 KK=2                                                       SYMRNG
131.          CALL SETUP(KCXDO1,KCXDO1,2,3,1,1,1,0)                      SYMRNG
132.          GO TO 56                                                   SYMRNG  56
133.       68 KK=3                                                       SYMRNG
134.          CALL SETUP(KCXDO2,KCXDO2,2,3,1,1,1,0)                      SYMRNG
135.          KC1=KCXDO2(1)                                              SYMRNG
136.          GO TO 56                                                   SYMRNG  56
137.       70 IF(MAX.EQ.0)RETURN                                         SYMRNG
138.  C6.0       DETERMINE SYMMETRY NUMBER OF RING WHOSE LIGANDS ARE A,B,C,D,E,F  SYMRNG
139.          KS=0                                                       SYMRNG
140.          DO72    N=1,6                                              SYMRNG
```

17 APR 73 6.02-36

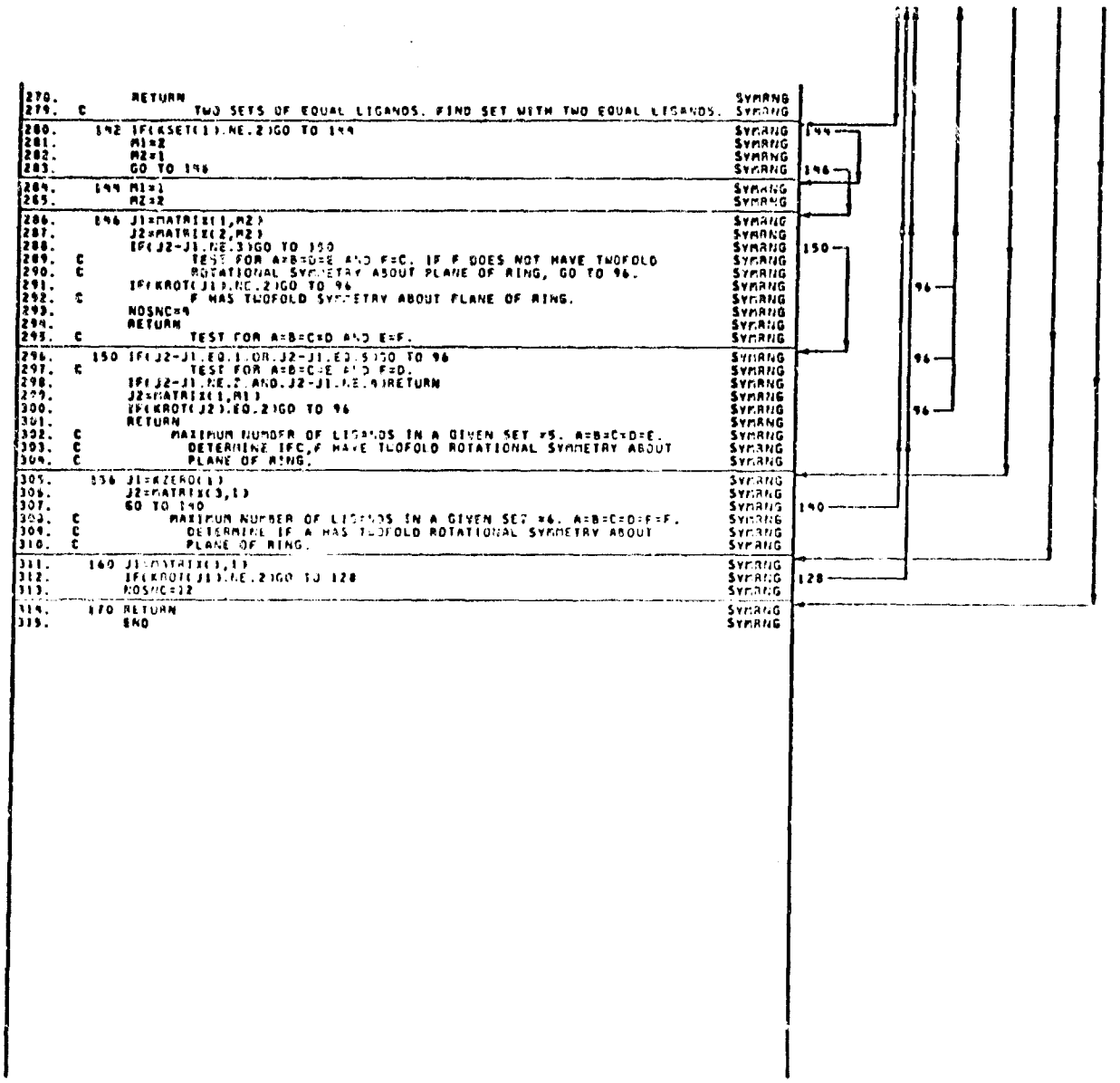


```
      IF(KARO(N).NE.0)GO TO 72                                          SYMRNG
      KS=KS+1                                                           SYMRNG
      KZERO(KS)=N                                                       SYMRNG
   72 CONTINUE                                                          SYMRNG
C          FIND NUMBER OF SETS OF EQUAL LIGANDS, NUMBER OF LIGANDS IN   SYMRNG
C          EACH SET, AND LOCATION OF EACH LIGAND IN RING.               SYMRNG
      NUM=0                                                             SYMRNG
      DO76   J=1,6                                                      SYMRNG
      KS=0                                                              SYMRNG
      IF(KARO(J).EQ.0.OR.J.EQ.6)GO TO 76                                SYMRNG
      NUM=NUM+1                                                         SYMRNG
      KS=KS+1                                                           SYMRNG
      MATRIX(KS,NUM)=J                                                  SYMRNG
      K1=J+1                                                            SYMRNG
      DO74   K=K1,6                                                     SYMRNG
      IF(KARO(K).NE.KARO(J))GO TO 74                                    SYMRNG
      KS=KS+1                                                           SYMRNG
      MATRIX(KS,NUM)=K                                                  SYMRNG
      KARO(K)=0                                                         SYMRNG
   74 CONTINUE                                                          SYMRNG
      KSET(NUM)=KS                                                      SYMRNG
   76 CONTINUE                                                          SYMRNG
C          ARRANGE ELEMENTS IN ARRAY MATRIX(J,K) IN ORDER OFF INCREASING SYMRNG
C          MAGNITUDE.                                                   SYMRNG
      DO78   K=1,NUM                                                    SYMRNG
      NF=KSET(K)-1                                                      SYMRNG
      DO78   N=1,NF                                                     SYMRNG
      J1=N                                                              SYMRNG
      J2=N+1                                                            SYMRNG
   77 IF(J2.EQ.1)GO TO 78                                               SYMRNG
      IF(MATRIX(J1,K).LE.MATRIX(J2,K))GO TO 78                          SYMRNG
      MX=MATRIX(J1,K)                                                   SYMRNG
      MATRIX(J1,K)=MATRIX(J2,K)                                         SYMRNG
      MATRIX(J2,K)=MX                                                   SYMRNG
      J1=J1-1                                                           SYMRNG
      J2=J1+1                                                           SYMRNG
      GO TO 77                                                          SYMRNG
   78 CONTINUE                                                          SYMRNG
      GO TO(170,80,104,136,156,160),MAX                                 SYMRNG
C          MAXIMUM NUMBER OF EQUAL LIGANDS IN A GIVEN SET =2.           SYMRNG
   80 IF(NUM-2)170,82,86                                                SYMRNG
C             TWO SETS OF EQUAL LIGANDS. TEST FOR A=C, F=D, B AND E HAVE SYMRNG
C             THOFOLD ROTATIONAL SYMMETRY ABOUT PLANE OF RING.          SYMRNG
   82 IF(KSET(1).NE.2.OR.MATRIX(2,1)-MATRIX(1,1).NE.2.AND.MATRIX(2,1)-  SYMRNG
     1MATRIX(1,1).NE.4)RETURN                                           SYMRNG
      IF(KSET(2).NE.2.OR.MATRIX(2,2)-MATRIX(1,2).NE.2.AND.MATRIX(2,2)-  SYMRNG
     1MATRIX(1,2).NE.4)RETURN                                           SYMRNG
      J1=KZERO(1)                                                       SYMRNG
      J2=KZERO(2)                                                       SYMRNG
   84 IF(KARO(J1).EQ.1.OR.KARO(J2).EQ.1.OR.J2-J1.NE.3)RETURN            SYMRNG
      NOSYM=2                                                           SYMRNG
      RETURN                                                            SYMRNG
C             THREE SETS OF EQUAL LIGANDS. TEST FOR A=B, F=D, B=E.      SYMRNG
C             DETERMINE IF B HAS THOFOLD ROTATIONAL SYMMETRY ABOUT THE  SYMRNG
C             PLANE OF THE RING.                                        SYMRNG
   86 KS=0                                                              SYMRNG
      J1=0                                                              SYMRNG
      DO90   J=1,3                                                      SYMRNG
      IF(MATRIX(2,J)-MATRIX(1,J).NE.2.AND.MATRIX(2,J)-MATRIX(1,J).NE.4)SYMRNG
     1GO TO 88                                                          SYMRNG
      KS=KS+1                                                           SYMRNG
      GO TO 90                                                          SYMRNG
   88 IF(KS.NE.2)GO TO 92                                               SYMRNG
      J1=MATRIX(1,J)                                                    SYMRNG
      J2=MATRIX(2,J)                                                    SYMRNG
   90 CONTINUE                                                          SYMRNG
      IF(KS.EQ.2.AND.J1.NE.0)GO TO 84                                   SYMRNG
C             TEST FOR A=D, B=E, C=F.                                   SYMRNG
   92 DO94   J=1,3                                                      SYMRNG
```


17 APR 73 8.02.36



```
210.          IF(MATRIX(2,J)-MATRIX(1,J).NE.3)GO TO 96                    SYMRNG   98
211.       96 CONTINUE                                                    SYMRNG
212.       96 NOSNC=2                                                     SYMRNG
213.          RETURN                                                      SYMRNG
214.       98 KS=0                                                        SYMRNG
215.          J1=0                                                        SYMRNG
216.          DO102   J=1,3                                               SYMRNG
217.          IF(MATRIX(2,J)-MATRIX(1,J).NE.1.AND.MATRIX(2,J)-MATRIX(1,J).NE.9)  SYMRNG  100
218.         1GO TO 100                                                   SYMRNG
219.          KS=KS+1                                                     SYMRNG
220.          GO TO 102                                                   SYMRNG  102
221.      100 IF(MATRIX(2,J)-MATRIX(1,J).NE.3)RETURN                      SYMRNG
222.          J1=1                                                        SYMRNG
223.      102 CONTINUE                                                    SYMRNG
224.          IF(KS.EQ.2.AND.J1.EQ.3)GO TO 96                             SYMRNG
225.          RETURN                                                      SYMRNG
226.  C             MAXIMUM NUMBER OF EQUAL LIGANDS IN A GIVEN SET =3.   SYMRNG
227.      104 IF(NUM.NE.2)RETURN                                          SYMRNG
228.  C             TWO SETS OF EQUAL LIGANDS. FIND SET WITH TWO EQUAL LIGANDS  SYMRNG
229.  C                 (IF ANY).                                         SYMRNG
230.          IF(KSET(1).NE.3)GO TO 108                                   SYMRNG  108
231.          M1=1                                                        SYMRNG
232.          IF(KSET(2).NE.2)GO TO 120                                   SYMRNG  120
233.          M2=2                                                        SYMRNG
234.          GO TO 112                                                   SYMRNG  112
235.      108 IF(KSET(1).NE.2)RETURN                                      SYMRNG
236.          M2=1                                                        SYMRNG
237.          IF(KSET(2).NE.3)RETURN                                      SYMRNG
238.          M1=2                                                        SYMRNG
239.  C             TEST FOR A=B=C, F=D, E AND E HAVE TWOFOLD ROTATIONAL  SYMRNG
240.  C             SYMMETRY ABOUT PLANE OF RING.                       SYMRNG
241.      112 J2=KZERO(1)                                                 SYMRNG
242.          IF(MATRIX(2,M2)-MATRIX(1,M2).EQ.2.AND.J2-MATRIX(1,M2).EQ.1)GO  SYMRNG  118
243.         1TO 118                                                      SYMRNG
244.          IF(MATRIX(2,M2)-MATRIX(1,M2).NE.9)RETURN                    SYMRNG
245.          IF(MATRIX(1,M2).EQ.1.AND.J2-MATRIX(2,M2).EQ.1)GO TO 118     SYMRNG  118
246.          IF(MATRIX(1,M2).NE.2.OR.MATRIX(1,M2)-J2.NE.1)RETURN         SYMRNG
247.      118 J1=MATRIX(2,M1)                                             SYMRNG
248.          IF(KROT(J1).NE.1.AND.KROT(J2).NE.1)GO TO 96                 SYMRNG
249.          RETURN                                                      SYMRNG  96
250.      120 IF(KSET(2).NE.3)RETURN                                      SYMRNG
251.  C             TEST FOR A=B=C, D=E=F, F AND E HAVE TWOFOLD ROTATIONAL  SYMRNG
252.  C             SYMMETRY ABOUT PLANE OF RING.                       SYMRNG
253.          M2=2                                                        SYMRNG
254.          IF(MATRIX(2,M1)-MATRIX(1,M1).NE.1.OR.MATRIX(3,M1)-MATRIX(2,M1).  SYMRNG  124
255.         1NE.1).AND.(MATRIX(2,M1)-MATRIX(1,M1).NE.1.OR.MATRIX(3,M1)-  SYMRNG
256.         2MATRIX(2,M1).NE.9).AND.(MATRIX(2,M1)-MATRIX(1,M1).NE.9.OR.  SYMRNG
257.         3MATRIX(3,M1)-MATRIX(2,M1).NE.1))GO TO 124                   SYMRNG
258.          J2=MATRIX(2,M2)                                             SYMRNG
259.          GO TO 118                                                   SYMRNG  118
260.  C             TEST FOR A=C, E AND B=D=F.                           SYMRNG
261.      124 IF(MATRIX(2,M1)-MATRIX(1,M1).NE.2.OR.MATRIX(3,M1)-MATRIX(2,M1).  SYMRNG
262.         1NE.2)RETURN                                                 ALT124
263.          J1=MATRIX(2,M1)                                             SYMRNG
264.          J2=MATRIX(2,M2)                                             SYMRNG
265.          IF(KROT(J1).EQ.2.OR.KROT(J2).NE.2)GO TO 130                 SYMRNG  130
266.      128 POSIC=6                                                     SYMRNG
267.          RETURN                                                      SYMRNG
268.      130 POSIC=3                                                     SYMRNG
269.          RETURN                                                      SYMRNG
270.  C             MAXIMUM NUMBER OF EQUAL LIGANDS IN A GIVEN SET =4.   SYMRNG
271.      136 IF(KSET(1).EQ.2,133,142                                     SYMRNG  133  142
272.  C             ONE SET OF EQUAL LIGANDS. TEST FOR A=B=D=E, F AND C WITH  SYMRNG
273.  C             TWOFOLD ROTATIONAL SYMMETRY ABOUT PLANE OF RING.     SYMRNG
274.      133 J1=KZERO(1)                                                 SYMRNG
275.          J2=KZERO(2)                                                 SYMRNG
276.          IF(J2-J1.NE.3)RETURN                                        SYMRNG
277.      140 IF(KROT(J1).EQ.2.AND.KROT(J2).EQ.2)GO TO 96                 SYMRNG  96
```



```
270.          RETURN                                                    SYMRNG
279.    C           TWO SETS OF EQUAL LIGANDS. FIND SET WITH TWO EQUAL LIGANDS.  SYMRNG
280.      142 IF(KSET(1).NE.2)GO TO 144                                   SYMRNG
281.          M1=2                                                      SYMRNG
282.          M2=1                                                      SYMRNG
283.          GO TO 146                                                 SYMRNG
284.      144 M1=1                                                      SYMRNG
285.          M2=2                                                      SYMRNG
286.      146 J1=MATRIX(1,M2)                                           SYMRNG
287.          J2=MATRIX(2,M2)                                           SYMRNG
288.          IF(J2-J1.NE.3)GO TO 150                                   SYMRNG
289.    C           TEST FOR A=B=D=E AND F=C. IF F DOES NOT HAVE TWOFOLD  SYMRNG
290.    C           ROTATIONAL SYMMETRY ABOUT PLANE OF RING, GO TO 96.  SYMRNG
291.          IF(KROT(J1).NE.2)GO TO 96                                 SYMRNG
292.    C           F HAS TWOFOLD SYMMETRY ABOUT PLANE OF RING.         SYMRNG
293.          NOSNC=9                                                   SYMRNG
294.          RETURN                                                    SYMRNG
295.    C           TEST FOR A=B=C=D AND E=F.                           SYMRNG
296.      150 IF(J2-J1.EQ.1.OR.J2-J1.EQ.5)GO TO 96                      SYMRNG
297.    C           TEST FOR A=B=C=E AND F=D.                           SYMRNG
298.          IF(J2-J1.NE.2.AND.J2-J1.NE.4)RETURN                       SYMRNG
299.          J2=MATRIX(1,M1)                                           SYMRNG
300.          IF(KROT(J2).EQ.2)GO TO 96                                 SYMRNG
301.          RETURN                                                    SYMRNG
302.    C           MAXIMUM NUMBER OF LIGANDS IN A GIVEN SET =5. A=B=C=D=E.  SYMRNG
303.    C           DETERMINE IF C HAVE TWOFOLD ROTATIONAL SYMMETRY ABOUT  SYMRNG
304.    C           PLANE OF RING.                                      SYMRNG
305.      156 J1=KZERO(1)                                               SYMRNG
306.          J2=MATRIX(3,1)                                            SYMRNG
307.          GO TO 140                                                 SYMRNG
308.    C           MAXIMUM NUMBER OF LIGANDS IN A GIVEN SET =6. A=B=C=D=E=F.  SYMRNG
309.    C           DETERMINE IF A HAS TWOFOLD ROTATIONAL SYMMETRY ABOUT  SYMRNG
310.    C           PLANE OF RING.                                      SYMRNG
311.      160 J1=MATRIX(1,1)                                            SYMRNG
312.          IF(KROT(J1).NE.2)GO TO 128                                SYMRNG
313.          NOSYC=12                                                  SYMRNG
314.      170 RETURN                                                    SYMRNG
315.          END                                                       SYMRNG
```

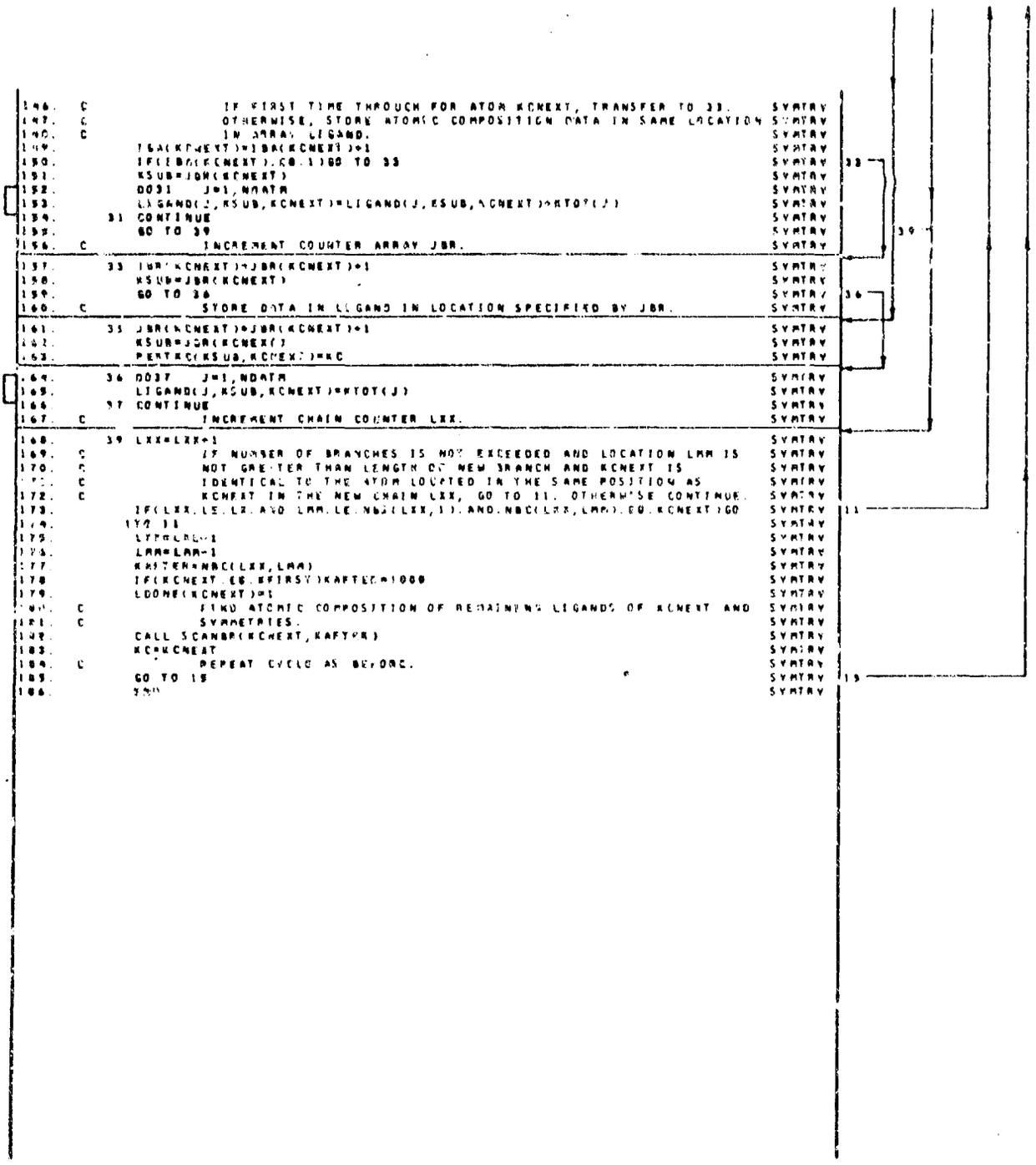

17 APR 73 0.02.36

```
      SUBROUTINE SYMTRY(LE)                                                 SYMTRY
C        THIS SUBROUTINE CONTROLS THE OPERATION WHICH FINDS THE ATOMIC      SYMTRY
C        COMPOSITION OF EACH LIGAND OF EACH CORE ATOM KC IN THE             SYMTRY
C        MOLECULE. IF KC IS NOT A RING ATOM, THE ROUTINE DETERMINES         SYMTRY
C        WHETHER KC POSSESSES IDENTICAL LIGANDS. KC MAY HAVE MORE THAN      SYMTRY
C        ONE SET OF IDENTICAL LIGANDS.                                      SYMTRY
      INTEGER SYMX(4),SYMBOL(9),GRID(50,80)                                 SYMTRY
      INTEGER WEIGHT(9)                                                     SYMTRY
      INTEGER PERTKC(4,100)                                                 SYMTRY
      DIMENSION KTOT(5),LDONE(100),JBR(100),LIGAND(5,4,100),MKT(40),        SYMTRY
     1MER(3,40),NV(100),MCV(100),IBA(100),KCSAME(6,100),NOSAME(6,100),      SYMTRY
     2NTOTAL(3,100)                                                         SYMTRY
      COMMON/BLK1/NO,NOS,SYMX,SYMBOL,NOVAL(9),GRID                          SYMTRY
      COMMON/BLK2/WEIGHT,AWGT(9),MOLWT(100),IX(100,5,6),NC(100),KCC         SYMTRY
      COMMON/BLK3/IRING(40,30),IMATX(50,80),NN(100),IBC(100),KON(100),      SYMTRY
     1IBBR(100),IB(100,8),IRG,NOBR                                          SYMTRY
      COMMON/BLK4/NBC(60,50),NBS(60,2),NBX(60,20),IRC,NONFUS,IRCTOT         SYMTRY
      COMMON/BLK5/NDATM,NUMATM(5),MBC(50),MBS(2),JM,JV,LFLAGS,LFLAGG        SYMTRY
      EQUIVALENCE (KTOT(1),IB(42,8)),(LDONE(1),IMATX(2,41)),(JBR(1),        SYMTRY
     1IMATX(2,43)),(LIGAND(1,1,1),IMATX(2,1)),(PERTKC(1,1),NBX(2,5)),       SYMTRY
     2(KCV,IB(97,8)),(MKT(1),IB(35,2)),(MER(1,1),GRID(1,37)),               SYMTRY
     3(NV(1),GRID(1,29)),(MCV(1),GRID(1,31)),(IBA(1),GRID(1,40)),           SYMTRY
     4(KCSAME(1,1),GRID(4,42)),(NOSAME(1,1),GRID(4,54)),(NTOTAL(1,1),       SYMTRY
     5GRID(4,70))                                                           SYMTRY
C1.0     INITIALIZE VARIABLES USED HERE AND IN SUBSEQUENT SUBROUTINES.      SYMTRY
      DO1   I=1,NDATM                                                       SYMTRY
      KTOT(I)=0                                                             SYMTRY
    1 CONTINUE                                                              SYMTRY
      DO3   I=1,KCC                                                         SYMTRY
      JBR(I)=1                                                              SYMTRY
      IBA(I)=0                                                              SYMTRY
      LDONE(I)=0                                                            SYMTRY
      NV(I)=NCC(I)                                                          SYMTRY
      MCV(I)=0                                                              SYMTRY
      KCSAME(1,I)=0                                                         SYMTRY
      DO2   K=1,4                                                           SYMTRY
      PERTKC(K,I)=0                                                         SYMTRY
      DO2   J=1,NDATM                                                       SYMTRY
      LIGAND(J,K,I)=0                                                       SYMTRY
    2 CONTINUE                                                              SYMTRY
      DO3   L=1,3                                                           SYMTRY
      KCSAME(L,I)=0                                                         SYMTRY
      NTOTAL(L,I)=0                                                         SYMTRY
    3 CONTINUE                                                              SYMTRY
      KCV=0                                                                 SYMTRY
      DO4   K=1,40                                                          SYMTRY
      MKT(K)=0                                                              SYMTRY
      DO4   J=1,3                                                           SYMTRY
      MER(J,K)=0                                                            SYMTRY
    4 CONTINUE                                                              SYMTRY
C2.0     EVALUATE THE ATOMIC COMPOSITION AND SYMMETRY OF THE LIGANDS        SYMTRY
C        UNTIL A BRANCH ATOM IS REACHED OR ALL CORE ATOMS HAVE BEEN         SYMTRY
C        PROCESSED.                                                         SYMTRY
      LXX=1                                                                 SYMTRY
      LMM=1                                                                 SYMTRY
      KCBEF=1000                                                            SYMTRY
    6 LN=LMM+1                                                              SYMTRY
      KCNEXT=NBC(LXX,LN)                                                    SYMTRY
    7 KC=NBC(LXX,LMM)                                                       SYMTRY
C           COMPUTE ATOMIC COMPOSITION AND CHECK SIMILARITY OF LIGANDS      SYMTRY
C           OF CORE ATOM KC.                                                SYMTRY
      CALL SCANCH(KCBEF,KC,KCNEXT)                                          SYMTRY
      LDONE(KC)=1                                                           SYMTRY
      IF(KCNEXT.EQ.1000)GO TO 140                                           SYMTRY
      IF(NC(KCNEXT).GE.3)GO TO 8                                            SYMTRY
C           CORE ATOM KCNEXT IS NOT A BRANCH ATOM. RESET VARIABLES AND      SYMTRY
C           REPEAT CYCLE.                                                   SYMTRY
      KCBEF=KC                                                              SYMTRY
      LMM=LMM+1                                                             SYMTRY
      IF(LMM.LT.NBS(LXX,1))GO TO 6                                          SYMTRY
      KCNEXT=1000                                                           SYMTRY
      GO TO 7                                                               SYMTRY
C           KCNEXT IS A BRANCH ATOM. STORE COMPOSITION OF LIGAND.           SYMTRY
    8 KFIRST=KCNEXT                                                         SYMTRY
```


19 FEB 73 8.02.39

```
      KSUB=JBR(KCNEXT)                                                  SYMTRY
      DO 9   J=1,NDATM                                                  SYMTRY
      LIGAND(J,KSUB,KCNEXT)=KTOT(J)                                     SYMTRY
    9 CONTINUE                                                          SYMTRY
      PERTKC(KSUB,KCNEXT)=KC                                            SYMTRY
C3.0  CONTINUE THE EVALUATION OF THE ATOMIC COMPOSITION AND SYMMETRY    SYMTRY
C       OF THE LIGANDS OF THE CHAIN, BRANCH, OR TERMINAL CORE ATOMS.    SYMTRY
      LIMIT=LM                                                          SYMTRY
   11 LM=NBS(LXX,1)                                                     SYMTRY
      LMM=LM                                                            SYMTRY
      DO 13   J=1,NDATM                                                 SYMTRY
      KTOT(J)=0                                                         SYMTRY
   13 CONTINUE                                                          SYMTRY
   15 IF(LMM.GE.LIMIT)GO TO 17                                          SYMTRY
C          SCAN HAS BEEN COMPLETED. PRINT OUT THE ATOMIC COMPOSITION    SYMTRY
C          OF THE LIGANDS AND EXIT.                                     SYMTRY
  140 WRITE(6,150)                                                      SYMTRY
  150 FORMAT(// 1H0,31X,66HATOMIC  COMPOSITION  OF  CORE  ATOM  AND  LIG SYMTRY
     1ANDS  IN  EACH  GROUP/1H0,40X,12HGROUP NUMBER,5X,8HSUBGROUP,5X,     SYMTRY
     21HH,5X,1HC,5X,1HO,5X,1HN)                                         SYMTRY
      DO 156   KC=1,KCC                                                 SYMTRY
      KSUB=JBR(KC)                                                      SYMTRY
      DO 151   J=1,4                                                    SYMTRY
  151 KTOT(J)=0                                                         SYMTRY
      CALL SUMATM(IX(KC,1,1),1,KTOT(1))                                 SYMTRY
      WRITE(6,152)KC,(KTOT(J),J=1,4)                                    SYMTRY
  152 FORMAT(1H ,42X,I5,12X,6HCORE  ,4I6)                               SYMTRY
      WRITE(6,154)((LIGAND(J,I,KC),J=1,4),I=1,KSUB)                     SYMTRY
  154 FORMAT(1H ,58X,7HLIGAND ,4I6)                                     SYMTRY
  156 CONTINUE                                                          SYMTRY
      RETURN                                                            SYMTRY
   17 KCBEF=KC                                                          SYMTRY
      KC=NBC(LXX,LMM)                                                   SYMTRY
C          HAVE ALL LIGANDS OF KC BEEN IDENTIFIED                       SYMTRY
      IF(LDONE(KC).EQ.0)GO TO 19                                        SYMTRY
C             YES. REPEAT CYCLE WITH THE NEXT ELEMENT UP CHAIN LXX.     SYMTRY
      LMM=LMM-1                                                         SYMTRY
      GO TO 15                                                          SYMTRY
C             NO. HOW MANY CORE LIGANDS DOES KC HAVE                    SYMTRY
   19 IF(NC(KC).LT.3)GO TO 21                                           SYMTRY
C                IT IS A BRANCH ATOM. SET VARIABLES.                    SYMTRY
      KCNEXT=KC                                                         SYMTRY
      LMM=LMM+1                                                         SYMTRY
      KC=NBC(LXX,LMM)                                                   SYMTRY
      GO TO 27                                                          SYMTRY
C                IT IS A CHAIN OR TERMINAL ATOM.                        SYMTRY
   21 IF(LMM.NE.LM)GO TO 25                                             SYMTRY
C                CORE ATOM KC IS THE LAST ELEMENT IN CHAIN LXX. SET KCBEF SYMTRY
C                ACCORDINGLY.                                           SYMTRY
      KCBEF=1000                                                        SYMTRY
   25 KCNEXT=NBC(LXX,LMM-1)                                             SYMTRY
C                COMPUTE ATOMIC COMPOSITION AND SYMMETAY OF LIGANDS OF KC SYMTRY
      CALL SCANCH(KCBEF,KC,KCNEXT)                                      SYMTRY
      LDONE(KC)=1                                                       SYMTRY
      LMM=LMM-1                                                         SYMTRY
C                IF CORE LIGAND KCNEXT IS NOT A BRANCH ATOM, GO TO 27. IF SYMTRY
C                IT IS, CONTINUE.                                       SYMTRY
      IF(NC(KCNEXT).LT.3)GO TO 15                                       SYMTRY
C          HAVE ALL LIGANDS OF BRANCH ATOM KCNEXT BEEN IDENTIFIED       SYMTRY
   27 IF(LDONE(KCNEXT).EQ.0)GO TO 29                                    SYMTRY
C             YES. REPEAT CYCLE WITH THE NEXT ELEMENT OF CHAIN LXX.     SYMTRY
      LMM=LMM-1                                                         SYMTRY
      GO TO 15                                                          SYMTRY
C                NO. CONTINUE.                                          SYMTRY
   29 IF(IBC(KCNEXT).EQ.0)GO TO 35                                      SYMTRY
C                KCNEXT IS A RING ATOM.                                 SYMTRY
      IF(LMM.EQ.LM)GO TO 39                                             SYMTRY
C                KCNEXT IS NOT THE LAST ELEMENT IN CHAIN LXX. KC IS THE SYMTRY
C                ELEMENT BELOW KCNEXT IN CHAIN LXX.                     SYMTRY
      IF(IBC(KC).EQ.0)GO TO 39                                          SYMTRY
C                KC IS ALSO A RING ATOM.                                SYMTRY
```

19 FEB 73 8.02.39



```
C                IF FIRST TIME THROUGH FOR ATOM KCNEXT, TRANSFER TO 33.
C                OTHERWISE, STORE ATOMIC COMPOSITION DATA IN SAME LOCATION
C                IN ARRAY LIGAND.
          IBA(KCNEXT)=IBA(KCNEXT)+1
          IF(IBA(KCNEXT).GE.1)GO TO 33
          KSUB=JBR(KCNEXT)
          DO31   J=1,NOATM
          LIGAND(J,KSUB,KCNEXT)=LIGAND(J,KSUB,KCNEXT)+ATOT(J)
       31 CONTINUE
          GO TO 39
C                INCREMENT COUNTER ARRAY JBR.
       33 JBR(KCNEXT)=JBR(KCNEXT)+1
          KSUB=JBR(KCNEXT)
          GO TO 36
C                STORE DATA IN LIGAND IN LOCATION SPECIFIED BY JBR.
       35 JBR(KCNEXT)=JBR(KCNEXT)+1
          KSUB=JBR(KCNEXT)
          PERTKC(KSUB,KCNEXT)=KC
       36 DO37   J=1,NOATM
          LIGAND(J,KSUB,KCNEXT)=ATOT(J)
       37 CONTINUE
C                INCREMENT CHAIN COUNTER LXX.
       39 LXX=LXX+1
C                IF NUMBER OF BRANCHES IS NOT EXCEEDED AND LOCATION LMM IS
C                NOT GREATER THAN LENGTH OF NEW BRANCH AND KCNEXT IS
C                IDENTICAL TO THE ATOM LOCATED IN THE SAME POSITION AS
C                KCNEXT IN THE NEW CHAIN LXX, GO TO 11. OTHERWISE CONTINUE.
          IF(LXX.LE.LX.AND.LMM.LE.NBC(LXX,1).AND.NBC(LXX,LMM).EQ.KCNEXT)GO
         1TO 11
          LXX=LXX-1
          LMM=LMM-1
          KAFTER=NBC(LXX,LMM)
          IF(KCNEXT.EQ.KFIRST)KAFTER=1000
          LDONE(KCNEXT)=1
C                FIND ATOMIC COMPOSITION OF REMAINING LIGANDS OF KCNEXT AND
C                SYMMETRIES.
          CALL SCANBR(KCNEXT,KAFTER)
          KC=KCNEXT
C                REPEAT CYCLE AS BEFORE.
          GO TO 15
          END
```

Appendix C
GLOSSARY OF PROGRAM VARIABLES

The following glossary contains definitions and cross references of all the FORTRAN symbols used in TGAP. The columns entitled "LOC" and "U" require some explanation. The symbols that appear in column LOC (location) have the following significance:

Number - Location within a common block where the variable is stored. If the variable is an array, it represents the storage location of the first element of the array.

* - Variable is a local variable to the routine specified in column BLOCK.

$ - Defines a subprogram either of TGAP or the System's Library.

The letters that appear in column U (usage) have the following significance:

A - Arithmetic statement function.

C - Constant found in a data statement and on the right side of an arithmetic statement.

D - Variable found in a data statement.

E - Entry point.

F - Function reference.

I - Variable found in common or in the argument list and used only on the right side of ar arithmetic statement.

M - Variable found in common or in the argument list and used on both sides of an arithmetic statement (i. e., modified).

O - Variable found in common or in the argument list and used only on the left side of an arithmetic statement.

S - Subroutine call.

W - Working variable found on the left side and then right side of an arithmetic statement which is not in common or in the argument list but is a local variable.

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
AA	(I)	Contains error printout information.	/CHAINM/	(*	CHAINM	C	REAL	AA	(4)
AFREQ		Reduction factor for the symmetry used when meso structures occur.	/ENTSYM/	(*	ENTSYM	M	REAL	AFREQ	
ALOG		Natural logarithm function routine (external routine of system).	/ALOG	/(0	ENTSYM TGAP	F F		ALOG ALOG	
ALTER	(I,J)	J = 1 contains weight of group I which lacks thermochemical data. J 2 contains weight of group whose data will be utilized for group I.	/TGAP	/(*	TGAP	C	INTE	ALTER	(20,2)
ASSIGN		Establishes the change in coordinates required to scan about indexed core atom. Eight directions are possible. The desired direction is defined by calling subroutine. The lower and upper limits for the bond termination loop of subroutine BOND are also set.	/ASSIGN/	(8	ASSIGN IDENT NUMBER SCAN	E S S S		ASSIGN ASSIGN ASSIGN ASSIGN	
ASYMC		Determines the number of asymmetric carbon atoms present in the molecule.	/ASYMC	/(0	ASYMC CORCIG	E S		ASYMC ASYMC	
A1	(I)	Ring correction data for cyclopropane.	/CORCIG/	(*	CORCIG	D	REAL	A1	(4)
A10	(I)	Ring correction data for cyclohexadiene 1,3.	/CORCIG/	(*	CORCIG	D	REAL	A10	(6)
A11	(I)	Ring correction data for cyclohexadiene 1,4.	/CORCIG/	(*	CORCIG	D	REAL	A11	(6)
A12	(I)	Ring correction data for cycloheptane.	/CORCIG/	(*	CORCIG	D	REAL	A12	(6)
A13	(I)	Ring correction data for cycloheptene.	/CORCIG/	(*	CORCIG	D	REAL	A13	(6)
A14	(I)	Ring correction data for cycloheptadiene 1,3.	/CORCIG/	(*	CORCIG	D	REAL	A14	(6)
A15	(I)	Ring correction data for cycloheptatriene 1,3,5.	/CORCIG/	(*	CORCIG	D	REAL	A15	(6)
A16	(I)	Ring correction data for cyclooctane.	/CORCIG/	(*	CORCIG	D	REAL	A16	(6)
A17	(I)	Ring correction data for cis-cyclooctene.	/CORCIG/	(*	CORCIG	D	REAL	A17	(6)
A18	(I)	Ring correction data for trans-cyclooctene.	/CORCIG/	(*	CORCIG	D	REAL	A18	(6)
A19	(I)	Ring correction data for cyclooctatriene 1,3,5.	/CORCIG/	(*	CORCIG	D	REAL	A19	(6)

17 APR 73 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SCOM	U	TYPE	VAR	DIM
A2	(I)	Ring correction data for cyclopropene.	/CORCIG/(*	)	COPCIG	D	REAL	A2	(6)
A20	(I)	Ring correction data for cyclooctatetraene	/CORCIG/(*	)	CORCIG	D	REAL	A20	(6)
A21	(I)	Ring correction data for cyclononane.	/CORCIG/(*	)	CORCIG	D	REAL	A21	(6)
A22	(I)	Ring correction data for cis-cyclononene.	/CORCIG/(*	)	CORCIG	D	REAL	A22	(6)
A23	(I)	Ring correction data for trans-cyclononene.	/CORCIG/(*	)	CORCIG	D	REAL	A23	(6)
A24	(I)	Ring correction data for spiropentane.	/CORCIG/(*	)	CORCIG	D	REAL	A24	(6)
A25	(I)	Ring correction data for bicyclobutane.	/CORCIG/(*	)	CORCIG	D	REAL	A25	(6)
A26	(I)	Ring correction data for bicyclopentane.	/CORCIG/(*	)	CORCIG	D	REAL	A26	(6)
A27	(I)	Ring correction data for bicyclohexane.	/CORCIG/(*	)	CORCIG	D	REAL	A27	(6)
A28	(I)	Ring correction data for bicycloheptane.	/CORCIG/(*	)	CORCIG	D	REAL	A28	(6)
A29	(I)	Ring correction data for bicyclooctane.	/CORCIG/(*	)	CORCIG	D	REAL	A29	(6)
A3	(I)	Ring correction data for cyclobutane.	/CORCIG/(*	)	CORCIG	D	REAL	A3	(6)
A30	(I)	Ring correction data for bicyclononane.	/CORCIG/(*	)	CORCIG	D	REAL	A30	(6)
A31	(I)	This array and all arrays up to array A44 contain the thermochemical corrections for the oxygen-containing rings.	/CORCIG/(*	)	CORCIG	D	REAL	A31	(6)
A32		Description not input	/CORCIG/(*	)	CORCIG	D	REAL	A32	(6)
A33		Description not input	/CORCIG/(*	)	CORCIG	D	REAL	A33	(6)
A34		Description not input	/CORCIG/(*	)	CORCIG	D	REAL	A34	(6)
A35		Description not input	/CORCIG/(*	)	CORCIG	D	REAL	A35	(6)
A36		Description not input	/CORCIG/(*	)	CORCIG	D	REAL	A36	(6)

17 APR 73 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
A37		Description not input	/CORCIG/(	*)	CORCIG	D	REAL	A37	(6)
A38		Description not input	/CORCIG/(	*)	CORCIG	D	REAL	A38	(6)
A39		Description not input	/CORCIG/(	*)	CORCIG	D	REAL	A39	(6)
A4	(1)	Ring correction data for cyclobutene.	/CORCIG/(	*)	CORCIG	D	REAL	A4	(6)
A40		Description not input	/CORCIG/(	*)	CORCIG	D	REAL	A40	(6)
A41		Description not input	/CORCIG/(	*)	CORCIG	D	REAL	A41	(6)
A42		Description not input	/CORCIG/(	*)	CORCIG	D	REAL	A42	(6)
A43		Description not input	/CORCIG/(	*)	CORCIG	D	REAL	A43	(6)
A44		Description not input	/CORCIG/(	*)	CORCIG	D	REAL	A44	(6)
A45	(1)	This array and all arrays up to array A50 contain the thermochemical corrections for the nitrogen-containing rings.	/CORCIG/(	*)	CORCIG	D	REAL	A45	(6)
A46		Description not input	/CORCIG/(	*)	CORCIG	D	REAL	A46	(6)
A47		Description not input	/CORCIG/(	*)	CORCIG	D	REAL	A47	(6)
A48		Description not input	/CORCIG/(	*)	CORCIG	D	REAL	A48	(6)
A49		Description not input	/CORCIG/(	*)	CORCIG	D	REAL	A49	(6)
A5	(1)	Ring correction data for cyclopentane.	/CORCIG/(	*)	CORCIG	D	REAL	A5	(6)
A50		Description not input	/CORCIG/(	*)	CORCIG	D	REAL	A50	(6)
A6	(1)	Ring correction data for cyclopentene.	/CORCIG/(	*)	CORCIG	D	REAL	A6	(6)
A7	(1)	Ring correction data for cyclopentadiene.	/CORCIG/(	*)	CORCIG	D	REAL	A7	(6)
A8	(1)	Ring correction data for cyclohexane.	/CORCIG/(	*)	CORCIG	D	REAL	A8	(6)

17 APR 73 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE SUBR	SUBROUTINE U	SUBROUTINE TYPE	USAGE VAR	USAGE DIM
A9	(I)	Ring correction data for cyclohexene.	/CORCIG/	/(*	) CORCIG	D	REAL	A9	(6)
BLANK			/TGAP	/(*	) TGAP	C	INTE	BLANK	
BOND		Determines bond type present at specified direction of indexed core atom. It also scans symbol input array along this direction until a symbol unequal to the designated bond is detected.	/BOND	/(*	) BOND FIND SCAN	E S S		BOND BOND BOND	
CASE		Case number.	/TGAP	/(*	) TGAP	W	INTE	CASE	
CHAINM		Control routine for section 2 of the program. This section identifies the chain formations present in the molecule as well as the number of unique rings and the ring components.	/CHAINM/	(*	) CHAINM TGAP	E S		CHAINM CHAINM	
CHANGE		Redefines all chains in array NBC if a chain exists that has only one branch atom and an upper residual (I2) that is greater than the lower residual (I1). If more than one such chain exists, the chain with the maximum I2 - I1 is chosen as reference.	/CHANGE/	(*	) CHAINM CHANGE	S E		CHANGE CHANGE	
CIS		Identifies each pair of ligands cis to the linkage C=C, where both carbon atoms (K1 and K2) have a connectivity of three.	/CIS	/(*	) CIS CISCOR CYCORR EXTROT	E S S S		CIS CIS CIS CIS	
CISCOR		Determines whether ligands of K1 and K2 exhibit a cis interaction. If so, it establishes the number, type, and magnitude of the correction.	/CISCOR/	(*	) CISCOR CORCIG SORNGI	E S S		CISCOR CISCOR CISCOR	
CISM		Sum of all cis contributions to the heat of formation.	/CISCOR/	(*	) CISCOR	M	REAL	CISM	
CISM		Sum of all cis contributions to the heat of formation.	/CORCIG/	(*	) CORCIG	M	REAL	CISM	
CISM		Sum of all cis corrections to the heat of formation.	/SORNGI/	(*	) SORNGI	I	REAL	CISM	
CISS		Sum of all cis contributions to the entropy.	/CISCOR/	(*	) CISCOR	M	REAL	CISS	
CISS		Sum of all cis contributions to the entropy.	/CORCIG/	(*	) CORCIG	M	REAL	CISS	
CISS		Sum of all cis corrections to the entropy.	/SORNGI/	(*	) SORNGI	I	REAL	CISS	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE SUBR	SUBROUTINE U	SUBROUTINE TYPE	USAGE VAR	USAGE DIM
CORCIG		Control routine for section 3 of the program. This section finds all second-order interaction and ring corrections as well as contributions due internal and external rotational symmetry and optical isomerism.	/CORCIG/(8	)	CORCIG TGAP	E S		CORCIG CORCIG	
CP	(I)	Heat capacity coefficients of molecule.	/TGAP	/(*)	TGAP	M	REAL	CP	(9)
CPALT	(I)	Values of the heat capacity coefficients for a particular second-order interaction.	/BLK3	/(6391)	CORCIG	M	REAL	CPALT	(9)
CPCIS	(I)	Heat capacity coefficients for cis corrections	/CORCIG/(*	)	CORCIG	C	REAL	CPCIS	(9)
CPSYM	(I)	Total contribution to the heat capacity coefficients from second-order interactions and ring corrections.	/CORCIS/(*	)	CORCIG	M	REAL	CPSYM	(9)
CPSYM	(I)	Total contribution to the heat capacity coefficients from second-order interactions and ring corrections.	/CRINGS/(*	)	CRINGS	M	REAL	CPSYM	(9)
CPSYM	(I)	Total contribution to the heat capacity coefficients from second-order interactions and ring corrections.	/HEXGON/(*	)	HEXGON	M	REAL	CPSYM	(9)
CPSYM	(I)	Total contribution to the heat capacity coefficients from second-order interactions and ring corrections.	/NRINGS/(*	)	NRINGS	M	REAL	CPSYM	(9)
CPSYM	(I)	Sum total of corrections to the heat capacity coefficients derived from section 3 of the program.	/TGAP	/(*)	TGAP	I	REAL	CPSYM	(9)
CPT	(I)	Heat capacity of molecule for temperatures specified in TARRAY. [cal-deg^{-1}-mole^{-1}]	/BLK3	/(1205)	TGAP	M	REAL	CPT	(19)
CPX	(I,J)	Contains the four heat capacity arrays CP1, CP2, CP3, and CP4.	/BLK7	/(541)	DATA1 GADATA TGAP	D I I	REAL REAL REAL	CP1 CP1 CPX	(180) (180) (180,9)
CP1	(I,J)	Contains the four heat capacity arrays CP1, CP2, CP3, and CP4.	/BLK7	/(541)	DATA1 GADATA TGAP	D I I	REAL REAL REAL	CP1 CP1 CPX	(180) (180) (180,9)
CP2	(I)	Coefficients for the temperature dependent term CP2·T of the fitted group additivity heat capacities. [cal-deg^{-2}-mole^{-1}]	/BLK7	/(721)	DATA1 GADATA	D I	REAL REAL	CP2 CP2	(180) (180)

17 APR 73 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
CP3	(I)	Coefficients for the temperature dependent term $CP3 \cdot T^2$ of the fitted group additivity heat capacities. [cal-deg^{-3}-mole^{-1}]	/BLK7	/(901)	DATA1	D	REAL	CP3	(180)
					GADATA	I	REAL	CP3	(180)
CP4	(I)	Coefficients for the temperature dependent term $CP4 \cdot T^3$ of the fitted group additivity heat capacities. [cal-deg^{-4}-mole^{-1}]	/BLK7	/(1081)	DATA1	D	REAL	CP4	(180)
					GADATA	I	REAL	CP4	(180)
CRINGS		Searches for certain carbon fused ring systems and applies ring corrections if present.	/CRINGS/(0)		CORCIG	S		CRINGS	
					CRINGS	E		CRINGS	
CTWO		Checks for the presence of monocyclic aromatic rings and nitro groups which exhibit twofold internal rotational symmetry about an axis with a nonlinear configuration. If present, it stores the pertinent identification numbers and contributions to the entropy.	/CTWO /(0)		CTWO	E		CTWO	
					INTROT	S		CTWO	
CYCORR		Determines the type of ring correction for a non-fused ring. Also sets various ring arrays and other indicators providing benzene- or pyridine-type rings are present.	/CYCORR/(0)		CORCIG	S		CYCORR	
					CYCORR	E		CYCORR	
DATCIS	(I,J)	Contains the heat of formation (I=1) and entropy (I=2) corrections for the second-order interaction J.	/BLK3	/(4451)	CISCOR	M	REAL	DATCIS(2,150)	
					CORCIG	M	REAL	DATCIS(2,150)	
					CTWO	O	REAL	DATCIS(2,150)	
					DITERE	M	REAL	DATCIS(2,150)	
					GAUCHE	M	REAL	DATCIS(2,150)	
					INTROT	M	REAL	DATCIS(2,150)	
DELETE		Disengages from use all the pairs of chain ring atoms and their non-ring ligands which have been found to be dissimilar by the comparison tests of subroutine EQUALR.	/DELETE/(0)		DELETE	E		DELETE	
					EQUALR	S		DELETE	
DELH		Heat of formation of molecule for 298°K. [kcal-mole^{-1}]	/TGAP /(0)		TGAP	W	REAL	DELH	
DELTA1		Determines that part of the group weight composed of the sum of the assigned weights of the group core atom and of the atomic constituents bonded to the core atom.	/DELTA1/(0)		DELTA1	E		DELTA1	
					STAND	S		DELTA1	
DELTA2		Adds all second-order multiple-bond contributions to the group weight value.	/DELTA2/(0)		DELTA2	E		DELTA2	
					STAND	S		DELTA2	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE SUBR	U	TYPE	USAGE VAR	DIM
DITERE		Searches for ditertiary ether groups in molecule and applies the appropriate correction, if present.	/DITERE/(	)	CORCIG DITERE	S E		DITERE DITERE	
DORPAR		Ortho/para correction for pyridine-type structures.	/CORCIG/(	)	CORCIG	C	REAL	DORPAR	
DORPAR		Ortho/para correction for pyridine-type structures.	/HEXGON/(	)	HEXGON	I	REAL	DORPAR	
DORTHO	(I)	Ortho corrections for benzene-type structures.	/CORCIG/(	)	CORCIG	C	REAL	DORTHO	(6)
DORTHO	(I)	Ortho corrections for benzene-type structures.	/HEXGON/(	)	HEXGON	I	REAL	DORTHO	(6)
DTEM		Sum of all ditertiary ether corrections to the heat of formation.	/CORCIG/(	)	CORCIG	W	REAL	DTEM	
DTEM		Sum of all ditertiary ether corrections to the heat of formation.	/DITERE/(	)	DITERE	M	REAL	DTEM	
ENDCS1		First symbol of end of case test variable.	/TGAP /(	)	TGAP	C	INTE	ENDCS1	
ENDCS2		Second symbol of end of case test variable.	/TGAP /(	)	TGAP	C	INTE	ENDCS2	
ENDCS3		Third symbol of end of case test variable.	/TGAP /(	)	TGAP	C	INTE	ENDCS3	
ENDRUN		End of run test variable.	/TGAP /(	)	TGAP	C	INTE	ENDRUN	
ENTSYM		Computes entropy contributions from internal and external rotational symmetry and optical isomerism.as well as auxiliary properties such as the longest chain in the molecule.	/ENTSYM/(	)	CORCIG ENTSYM	S E		ENTSYM ENTSYM	
EQUAL		Processes the operation that determines whether two or more complex (core) ligands have identical structures.	/EQUAL /(	)	CTWO EQUAL SETUP	S E S		EQUAL EQUAL EQUAL	
EQUALR		Determines whether the backbone structures of two rings, a and b, are the same. Non-ring ligands bonded to these rings are also classified and stored, but their symmetries are checked in subroutine EQUAL, not here.	/EQUALR/(	)	EQUAL EQUALR	S E		EQUALR EQUALR	
EXTROT		Computes the external rotation symmetry number of the molecule.	/EXTROT/(	)	CORCIG EXTROT	S E		EXTROT EXTROT	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
FIND		Determines whether the first atom to be indexed is a core atom or an univalent atom. If the latter, it determines the location of one of the core atoms that is bonded to this non-core and first-indexed atom. The data for this non-core atom are stored in IX.	/FIND /(8)		FIND	E		FIND	
					STAND	S		FIND	
FIRSTR		Determines if the two ring atoms under comparison are part of two different fused ring systems. If so, it compares several of the composition and structural properties of the two fused ring systms for similarity. If the two rings are not fused, it compares their ring size.	/FIRSTR/(8)		EQUALR	S		FIRSTR	
					FIRSTR	E		FIRSTR	
FT	(I)	Gibbs free energy of molecule for temperatures specified in TARRAY. [kcal-mole^{-1}]	/BLK3 /(1247)		TGAP	M	REAL	FT	(19)
FUSION		Identifies the set(s) of fused ring system(s) when present.	/FUSION/(8)		CORCIG	S		FUSION	
					FUSION	E		FUSION	
GADATA		Prints out Bensons tables of thermochemical group additivity data if requested by input data.	/GADATA/(8)		GADATA	E		GADATA	
					TGAP	S		GADATA	
GAUCHE		Determines whether the ligands of the test core atoms K1 and K2 are gauche to each other. If so, it finds the number, type, and magnitude of the gauche correction.	/GAUCHE/(8)		CORCIG	S		GAUCHE	
					GAUCHE	E		GAUCHE	
					SORNGI	S		GAUCHE	
GAUCHH		Sum of all gauche contributions to the heat of formation.	/CORCIG/(*)		CORCIG	W	REAL	GAUCHH	
GAUCHH		Sum of all gauche contributions to the heat of formation.	/GAUCHE/(*)		GAUCHE	M	REAL	GAUCHH	
GAUCHH		Sum of all gauche corrections to the heat of formation.	/SORNGI/(*)		SORNGI	I	REAL	GAUCHH	
GRID	(I, J)	Input array containing two-dimensional structure of the molecule (required input).	/BLK1 /(25)		BOND	I	INTE	GRID	(50,80)
					IDENT	M	INTE	GRID	(50,80)
					NUMBER	I	INTE	GRID	(50,80)
					SCAN	I	INTE	GRID	(50,80)
					STAND	I	INTE	GRID	(50,80)
					TGAP	M	INTE	GRID	(50,80)
GROUP1	(I)	First part of the chemical symbol of each group in the data library.	/BLK9 /(2)		DATA1	D	INTE	GROUP1	(180)
					GADATA	I	INTE	GROUP1	(180)
GROUP2	(I)	Second part of the chemical symbol of each group in the data libr y.	/BLK9 /(662)		DATA1	D	INTE	GROUP2	(180)
					GADATA	I	INTE	GROUP2	(180)

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE SUBR	SUBROUTINE U	SUBROUTINE TYPE	USAGE VAR	USAGE DIM
GROUP3	(I)	Third part of the chemical symbol of each group in the data library.	/BLK9	/(1442)	DATA1 GADATA	D I	INTE INTE	GROUP3 GROUP3	(180) (180)
HCONS		Enthalpy constant.	/TGAP	/( )	TGAP	W	REAL	HCONS	
HEXGON		Assigns the necessary weight corrections to the groups of a fused ring system which form part of a benzene or pyridine-type structure (i.e. conjugated, six-membered rings) and/or to non-fused rings of similar structure and to all groups bonded to these ring structures.	/HEXGON/( )		CORCIG HEXGON	S E		HEXGON HEXGON	
HF298	(I)	Heat of formation at 298°K of each group in the data library. [kcal-mole^{-1}]	/BLK7	/(181)	DATA1 GADATA TGAP	D I I	REAL REAL REAL	HF298 HF298 HF298	(180) (180) (180)
HRING		Total of ring contributions to the heat of formation.	/CORCIG/( )		CORCIG	W	REAL	HRING	
HRING		Sum of ring contributions to the heat of formation.	/CRINGS/( )		CRINGS	M	REAL	HRING	
HRING		Description not input	/HEXGON/( )		HEXGON	M	REAL	HRING	
HRING		Sum of ring contributions to the heat of formation.	/NRINGS/( )		NRINGS	M	REAL	HRING	
HSDT	(I)	Enthalpy of molecule at temperature T minus enthalpy at temperature 298°K for temperatures specified in TARRAY. [kcal-mole^{-1}]	/BLK3	/(1261)	TGAP	M	REAL	HSDT	(19)
HST	(I)	Enthalpy of molecule for temperatures specified in TARRAY. [kcal-mole^{-1}]	/BLK3	/(1223)	TGAP	M	REAL	HST	(19)
HSYM		Total contribution to the heat of formation from second-order interations and ring corrections.	/CORCIG/( )		CORCIG	O	REAL	HSYM	
HSYM		Sum total of corrections to the heat of formation at 298°K derived from section 3 of the program. [kcal-mole^{-1}]	/TGAP	/( )	TGAP	I	REAL	HSYM	
HTT	(I)	Total enthalpy of molecule (enthalpy + heat of formation at 298°K) for temperatures specified in TARRAY. [kcal-mole^{-1}]	/BLK3	/(1275)	TGAP	M	REAL	HTT	(19)
I		Subscript that designates bond type.	/BOND	/( )	BOND	M	INTE	I	

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE SUBR	U	TYPE	USAGE VAR	DIM
I		Subscript.	/CORCIG/(*	)	CORCIG	M	INTE	I	
I		Loop counter and subscript.	/CRINGS/(*	)	CRINGS	M	INTE	I	
I		Loop counter and subscript.	/CTWO /(*	)	CTWO	M	INTE	I	
I		Group number of nitrogen atom in ring K.	/CYCORR/(*	)	CYCORR	M	INTE	I	
I		Symbol code for element or radical in program library.	/DELTAI/(*	)	DELTAI	M	INTE	I	
I		Loop counter and subscript.	/HEXGON/(*	)	HEXGON	M	INTE	I	
I		Loop counter and subscript.	/LESSEN/(*	)	LESSEN	M	INTE	I	
I		Loop counter and subscript.	/MULTI /(*	)	MULTI	M	INTE	I	
I		Subscript.	/NRINGS/(*	)	NRINGS	M	INTE	I	
I		Actual location in array ORDSUM used in the calculation	/SEARCH/(*	)	SEARCH	M	INTE	I	
I		Loop counter and subscript.	/SYMTRY/(*	)	SYMTRY	M	INTE	I	
I		Loop counter and subscript.	/TGAP /(*	)	TGAP	M	INTE	I	
IA	(I)	Contains the size of each ring in the fused ring set B.	/BLK1 /(1325)		FIRSTR	M	INTE	IA	(40)
IA	(I)	Array used for temporary storage	/BLK3 /(6375)		EQUAL	M	INTE	IA	(9)
					EQUALR	M	INTE	IA	(9)
IA	(I)	Contains order elements of IB would have if arranged in order of increasing magnitude.	/ORDER /(*	)	ORDER	M	INTE	IA	(180)
IATOM	(I)	Contains characters of multi-character chemical symbol.	/BLK4 /(241)		IDENT	M	INTE	IATOM	(3)
IATOM	(I)	Contains characters of multi-word input symbol.	/MULTI /(*	)	MULTI	I	INTE	IATOM	(3)
IB	(I,J)	Contains group numbers of all core atom ligands in group I except for that of the parent core atom (which has already been scanned).	/BLK3 /(5601)		SCAN	O	INTE	IB	(100,8)
					SHIFT	O	INTE	IB	(100,8)
					STAND	M	INTE	IB	(100,8)
IB	(I)	Array the elements of which are to be checked for order of increasing magnitude.	/ORDER /(*	)	ORDER	I	INTE	IB	(180)

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
IBA	(I)	Transfer flag. =0 ligand composition of core atom I stored in new location. >0 data stored in same location of array LIGAND.	/BLK1	/(1975)	SYMTRY	M	INTE	IBA	(100)
IBC	(I)	Location I contains the ring number of the ring of ring component I. If I is a non-ring atom, location I = 0.	/BLK3	/(5301)	ASYMC	I	INTE	IBC	(100)
					CHAINM	O	INTE	IBC	(100)
					CIS	I	INTE	IBC	(100)
					CISCOR	I	INTE	IBC	(100)
					CORCIG	I	INTE	IBC	(100)
					CTWO	I	INTE	IBC	(100)
					CYCORR	I	INTE	IBC	(100)
					EQUAL	I	INTE	IBC	(100)
					EQUALR	I	INTE	IBC	(100)
					EXTROT	I	INTE	IBC	(100)
					FIRSTR	I	INTE	IBC	(100)
					FUSION	O	INTE	IBC	(100)
					LINEAR	I	INTE	IBC	(100)
					RESETR	M	INTE	IBC	(100)
					SCANBR	I	INTE	IBC	(100)
					SCANCH	I	INTE	IBC	(100)
					SORTGI	I	INTE	IBC	(100)
					SYCRNG	I	INTE	IBC	(100)
					SYMTRY	I	INTE	IBC	(100)
IBND	(I)	Transfer flag array equal to 0 or 1 depending whether or not bond is adjacent to first character (I=1) or second character (I=2).	/BLK4	/(244)	IDENT	M	INTE	IBND	(2)
IBONDD	(I)	Contains ring locations of double bonds.	/BLK3	/(5861)	CYCORR	M	INTE	IBONDD	(30)
IBONDS	(I)	Contains ring locations of single bonds.	/BLK3	/(5821)	CYCORR	M	INTE	IBONDS	(30)
IBX		Temporary storage variable.	/CORCIG/(*)		CORCIG	M	INTE	IBX	
IBX		Subscript.	/DITERE/(*)		DITERE	M	INTE	IBX	
IBX		Transfer flag.	/EQUALR/(*)		EQUALR	M	INTE	IBX	
IC	(I)	Contains the number of nitrogen atoms in each ring in fused set B.	/BLK3	/(3802)	FIRSTR	M	INTE	IC	(40)
IC		Number of distinct fused ring sets in molecule.	/CRINGS/(*)		CRINGS	M	INTE	IC	
IC		Total number of fused ring sets.	/FUSION/(*)		FUSION	M	INTE	IC	
IC		Total number of fused ring sets.	/HEXGON/(*)		HEXGON	M	INTE	IC	
IC		Total number of fused ring sets.	/NRINGS/(*)		NRINGS	M	INTE	IC	

17 APR 73 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE SUBR	I/O	TYPE	USAGE AR	OTR
ICC		Number of non-aromatic core atoms bonded to the central nitrogen atom.	/GAUCHE/(	*)	GAUCHE	W	INTE	ICC	
ICC		Transfer flag.	/IDENT /(	*)	IDENT	W	INTE	ICC	
ICHNK1		Group numb of first core atom in ligand KC1.	/EQUAL /(	*)	EQUAL	W	INTE	ICHNK1	
ICHNK2		Group number of first core atom in ligand KC2.	/EQUAL /(	*)	EQUAL	W	INTE	ICHNK2	
ICIS		Sum total of all cis interactions.	/CISCOR/(	*)	CISCOR	W	INTE	ICIS	
ICIS		Sum total of all cis interactions.	/CORCIG/(	*)	CORCIG	W	INTE	ICIS	
ICIS		Sum total of all cis interactions.	/SORNGI/(	*)	SORNGI	I	INTE	ICIS	
ICN	(I)	Contains group number of core atoms which have been topologically scanned and identified. For identification purpose, the number 10000 is stored prior to the group number of a terminal core atom.	/BLK4 //	363)	STAND	M	INTE	ICN	(100)
ICT		Transfer flag.	/IDENT /(	*)	IDENT	W	INTE	ICT	
IC1		Identification number of non-fused ring or of fused ring set A.	/FIRSTR/(	*)	FIRSTR	W	INTE	IC1	
IC2		Denotes number of entries in array IC2A.	/BLK3 /(	5921)	HEXGON	M	INTE	IC2	
IC.		Identification number of non-fused ring or of fused ring set B.	/FIRSTR/(	*)	FIRSTR	W	INTE	IC2	
IC2A	(I)	Contains the I.d. number of all fused ring sets composed of benzene and/or pyridine-type structures.	/BLK3 /(	5922)	HEXGON	M	INTE	IC2A	(10)
ID	(I)	Contains the number of double bonds in each ring in fused set B.	/BLK3 /(	3842)	FIRSTR	M	INTE	ID	(40)
ID		Direction of chemical bond.	/CIS /(	*)	CIS	W	INTE	ID	
IDB		Number of double bonds in ring K.	/CYCORR/(	*)	CYCORR	W	INTE	IDB	
IDBR	(I)	Group number of branch core atom I.	/BLK3 /(	5501)	CHAINM STAND	I O	INTE INTE	IDBR IDBR	(100) (100)

IF APR 73 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
IDENT		Identifies chemical symbol of indexed atom. If two or three characters are present in symbol these are packed into one word and the excess words occupied by these characters are replaced by a bond symbol or a blank, as required.	/IDENT	/(0)	FIND IDENT SCAN STAND	S E S S		IDENT IDENT IDENT IDENT	
IDI		Direction of chemical bond between central test core atoms K1 and K2.	/CIS	/(4)	CIS	W	INTE	IDI	
IDIF		Number of ring locations between the last two double bonds in ring K.	/CYCORR/	(*)	CYCORR	W	INTE	IDIF	
IDIF		Difference between size of test ring and the number of its components contained in the reference ring.	/LESSEN/	(*)	LESSEN	W	INTE	IDIF	
IDIF1		Number of ring locations between the next to the last and second to the last double bonds in ring K.	/CYCORR/	(*)	CYCORR	W	INTE	IDIF1	
IDTE		Counter that specifies the number of ditertiary ether groups present.	/CORCIG/	(*)	CORCIG	W	INTE	IDTE	
IDTE		Counter that specifies the number of ditertiary ether groups present.	/DITERE/	(*)	DITERE	W	INTE	IDTE	
IDX	(I,J)	Contains the group numbers (I=1 and 5) of a pair of atoms J, one from each branch, and the group numbers (I=2 to 4 and 6 to 8) of their ligands which are to be tested.	/BLK1	/(27)	DELETE EQUAL EQUALR	I M M	INTE INTE INTE	IDX IDX IDX	(8,99) (8,99) (8,99)
IDX		Displacement between the first and third oxygen atoms in ring K.	/CYCORR/	(*)	CYCORR	W	INTE	IDX	
IDXR	(I,J)	Contains the group numbers (I=1 and 5) of a pair of ring atoms J, one from each ring, and the group numbers (I=2 to 4 and 6 to 8) of their ligands which are to be tested.	/BLK1	/(825)	EQUALR	M	INTE	IDXR	(8,30)
IDXRD	(I)	Contains bond directions between atom KCIR and the ring atoms bonded to it, excluding atom KCIPR.	/BLK1	/(822)	EQUALR	M	INTE	IDXRD	(3)
IDX2		Displacement between the first and second oxygen atoms in ring K.	/CYCORR/	(*)	CYCORR	W	INTE	IDX2	
IE	(I)	Contains the number of oxygen atoms in each ring in fused set B.	/BLK3	/(5822)	FIRSTR	M	INTE	IE	(40)
IEND		End of run input symbol.	/TGAP	/(*)	TGAP	W	INTE	IEND	
IERR		Error flag =0 no error. =1 error present.	/BOND	/(*)	BOND	O	INTE	IERR	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
IERR		Error flag =0 no error. =1 error present.	/CHAINM/(*		) CHAINM	:	INTE	IERR	
IERR		Error flag =0 no error =1 error present	/FIND /(*		) FIND	M	INTE	IERR	
IERR		Error flag =0 no error. =1 error present.	/IDENT /(*		) IDENT	M	INTE	IERR	
IERR		Error flag =0 if multi-character input symbol is identified. =1 if multi-character input symbol is not identified.	/MULTI /(*		) MULTI	O	INTE	IERR	
IERR		Error flag =0 no error. =1 error present.	/NEWCOL/(*		) NEWCOL	O	INTE	IERR	
IERR		Error flag =0 no error. =1 error present.	/RING /(*		) RING	O	INTE	IERR	
IERR		Error flag =0 no error. =1 error present.	/SCAN /(*		) SCAN	M	INTE	IERR	
IERR		Error flag =0 no error. =1 error present.	/STAND /(*		) STAND	M	INTE	IERR	
IERR		Error flag. If zero, no error. If non zero, input or computational error exists.	/TGAP /(*		) TGAP	M	INTE	IERR	
IEXIT		Flag that causes an exit from calling subroutine.	/CIS /(*		) CIS	O	INTE	IEXIT	
IEXIT		Exit flag. =0 conditions satisfied in subroutine CIS. Continue. =1 conditions not satisfied. Exit from subroutine CISCOR.	/CISCOR/(*		) CISCOR	I	INTE	IEXIT	
IEXIT		Exit flag =0 subroutine CIS executed. =1 subroutine CIS not executed.	/CYCORR/(*		) CYCORR	I	INTE	IEXIT	
IEXIT		Exit flag from subroutine CIS. =0 cis computation performed. =1 cis computation not performed.	/EXTROT/(*		) EXTROT	I	INTE	IEXIT	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SYBM	U	TYPE	USAGE VAR	USAGE DIM
IFIX	(I)	Identifies characters of a particular symbol in SYMBL which have been matched to characters of input symbol in IATOM.	/BLK4 /(	301)	MULTI	M	INTE	IFIX	(4)
IFLAG1		Transfer flag of subroutine NEWKC.	/CHAINM/(	*)	CHAINM	M	INTE	IFLAG1	
IFLAG1		Transfer flag of subroutine NEWKC.	/NEWCOL/(	*)	NEWCOL	M	INTE	IFLAG1	
IFLAG1		Flag used to skip the intermediate test and search instructions.	/NEWKC /(	*)	NEWKC	I	INTE	IFLAG1	
IFLAG2		Operation flag of subroutine NEWKC.	/CHAINM/(	*)	CHAINM	I	INTE	IFLAG2	
IFLAG2		Operation flag of subroutine NEWKC =0 KC is the required branch junction atom. =1 KC is not the required branch atom junction.	/NEWCOL/(	*)	NEWCOL	I	INTE	IFLAG2	
IFLAG2		Operation flag =0 ligands of KC have all been processed. =1 unused ligands remain.	/NEWKC /(	*)	NEWKC	O	INTE	IFLAG2	
IFUNCT		Function flag. =0 symmetry test is not for a single aromatic ring. =1 symmetry test is for a single aromatic ring. =2 aromatic ring has twofold internal rotational symmetry.	/CTWO /(	*)	CTWO	M	INTE	IFUNCT	
IFUNCT		Function flag. =0 symmetry test is not for a single aromatic ring. =1 symmetry test is for a single aromatic ring. =2 aromatic ring has twofold internal rotational symmetry.	/EQUAL /(	*)	EQUAL	M	INTE	IFUNCT	
IFUNCT		Function flag. =0 symmetry test is not for a single aromatic ring. =1 symmetry test is for a single aromatic ring. =2 aromatic ring has twofold internal rotational symmetry.	/EQUALR/(	*)	EQUALR	I	INTE	IFUNCT	
IGM		Counter that specifies the number of pairs of core atoms (K1 and K2) whose ligands exhibit gauche interactions.	/CORCIG/(	*)	CORCIG	M	INTE	IGM	

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U TYPE	VAR	DIM
IGH		Counter that specifies the number of pairs of core atoms (K1 and K2) whose ligands exhibit gauche interactions.	/GAUCHE/(	*	) GAUCHE	M	INTE	IGH
IGH		Counter that specifies the number of pairs of core atoms (K1 and K2) whose ligands exhibit gauche interactions.	/SORNGI/(	*	) SORNGI	I	INTE	IGH
IGO		Counter that specifies the number of ether oxygen gauche interactions present in the molecule.	/CORCIG/(	*	) CORCIG	M	INTE	IGO
IGO		Counter that specifies the number of ether oxygen gauche interactions present in the molecule.	/DITERE/(	*	) DITERE	I	INTE	IGO
IGO		Counter that specifies the number of ether oxygen gauche interactions present in the molecule.	/GAUCHE/(	*	) GAUCHE	M	INTE	IGO
IGO		Counter that specifies the number of ether oxygen gauche interactions present in the molecule.	/SORNGI/(	*	) SORNGI	I	INTE	IGO
IGS		Counter that specifies the number of pairs of core atoms (K1 and K2) whose ligands exhibit second-order interactions.	/CISCOR/(	*	) CISCOR	M	INTE	IGS
IGS		Counter that specifies the number of pairs of core atoms (K1 and K2) whose ligands exhibit second-order interactions.	/CORCIG/(	*	) CORCIG	M	INTE	IGS
IGS		Counter that specifies the number of ring and second-order corrections.	/CRINGS/(	*	) CRINGS	M	INTE	IGS
IGS		Counter that specifies the number of pairs of core atoms (K1 and K2) whose ligands exhibit second-order interactions.	/CTWO /(	*	) CTWO	M	INTE	IGS
IGS		Counter that specifies the number of pairs of core atoms (K1 and K2) whose ligands exhibit second-order interactions	/DITERE/(	*	) DITERE	M	INTE	IGS
IGS		Counter that specifies the number of pairs of core atoms (K1 and K2) whose ligands exhibit second-order interactions.	/GAUCHE/(	*	) GAUCHE	M	INTE	IGS
IGS		Counter that specifies the number of ring and second-order corrections.	/HEXGON/(	*	) HEXGON	M	INTE	IGS
IGS		Counter that specifies the number of pairs of core atoms (K1 and K2) whose ligands exhibit second-order interactions.	/INTROT/(	*	) INTROT	M	INTE	IGS

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE SUBR	SUBROUTINE U	SUBROUTINE TYPE	USAGE VAR	USAGE DIM
IGS		Counter that specifies the number of ring and second-order corrections.	/NRINGS/	(*	NRINGS	M	INTE	IGS	
IGS		Counter that specifies the number of pairs of core atoms (K1 and K2) whose ligands exhibit second-order interactions.	/SORNGI/	(*	SORNGI	I	INTE	IGS	
IGSCIS		Counter that specifies the number of pairs of core atoms (K1 and K2) whose ligands exhibit one or two cis interactions.	/CISCOR/	(*	CISCOR	M	INTE	IGSCIS	
IGSCIS		Counter that specifies the number of pairs of core atoms (K1 and K2) whose ligands exhibit one or two cis interactions.	/CORCIG/	(*	CORCIG	M	INTE	IGSCIS	
IGSCIS		Counter that specifies the number of pairs of core atoms (K1 and K2) whose ligands exhibit one or two cis interactions.	/SORNGI/	(*	SORNGI	I	INTE	IGSCIS	
II		Storage flag for cis atoms.	/CIS	/(*	CIS	M	INTE	II	
II		Loop counter and subscript.	/EQUAL	/(*	EQUAL	M	INTE	II	
II		Loop counter and subscript.	/HEXGON/	(*	HEXGON	M	INTE	II	
II		Subscript for IBND array.	/IDENT	/(*	IDENT	M	INTE	II	
II		Subscript.	/LESSEN/	(*	LESSEN	M	INTE	II	
II		Ordinal number assigned each of the identical ligands.	/SAME	/(*	SAME	M	INTE	II	
II		Code number that distinguishes each type of identical ligand.	/SETUP	/(*	SETUP	M	INTE	II	
IKC	(I)	Temporary storage array.	/BLK1	/(3925)	INTROT	M	INTE	IKC	(100)
IM	(I)	Contains the order the subscripts of the second-order interaction data would have if the data were arranged in order of increasing numerical order.	/BLK3	/(4001)	CORCIG	I	INTE	IM	(150)
					DITERE	I	INTE	IM	(150)
IMATX	(I,J)	Contains the group number pertaining to a particular core atom in that location of IMATX that corresponds to the location in GRID containing the symbol for the core atom.	/BLK3	/(1201)	SCAN	M	INTE	IMATX	(50,80)
					STAND	O	INTE	IMATX	(50,80)
IN		Subscript of array LIGAND.	/SCANCH/	(*	SCANCH	M	INTE	IN	

17 APR 79 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	USAGE VAR	USAGE DIM
INR		Counter that specifies the number of groups with internal rotational symmetry.	/CORCIG/(*	)	CORCIG	I	INTE	INR	
INR		Counter that specifies the number of groups with internal rotational symmetry.	/CTWO /(*	)	CTWO	M	INTE	INR	
INR		Counter that specifies the number of groups with internal rotational symmetry.	/INTROT/(*	)	INTROT	M	INTE	INR	
INTMIM		Number of entries in array KINT.	/BLK1 /(3778)		EXTROT INTROT	M I	INTE INTE	INTMIM INTMIM	
INTROT		Searches for atoms which have three-fold rotational symmetry and which have not been included in the external rotational symmetry contribution. If present, the data are printed and the total internal contribution to the entropy is computed.	/INTROT/(8	)	CORCIG INTROT	S E		INTROT INTROT	
IO		Subscript.	/CTWO /(*	)	CTWO	M	INTE	IO	
IOPATM	(I)	Contains group numbers of asymmetric atoms.	/BLK3 /(3402)		ASYMC ENTSYM MAXCHN	O I I	INTE INTE INTE	IOPATM(100) IOPATM(100) IOPATM(100)	
IORD	(I)	Contains the order that the I subscripts of the data in JRC(I,J) would have if said data were arranged in increasing numerical order.	/BLK3 /(3353)		LESSEN	I	INTE	IORD	(90)
IORPAR		Counter that specifies the number of ortho and para pyridine corrections.	/CORCIG/(*	)	CORCIG	M	INTE	IORPAR	
IORPAR		Counter that specifies the number of ortho and para pyridine corrections.	/HEXGON/(*	)	HEXGON	M	INTE	IORPAR	
IORTHO		Counter that specifies the number of ortho benzene corrections.	/CORCIG/(*	)	CORCIG	M	INTE	IORTHO	
IORTHO		Counter that specifies the number of ortho benzene corrections.	/HEXGON/(*	)	HEXGON	M	INTE	IORTHO	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE SUBR	SUBROUTINE U	SUBROUTINE TYPE	USAGE VAR	USAGE DIM
IPRINT		Printout options flag (OJK). K=1 print data library. X=0 do not print data library. J=0 print heading and case number. J=1 do not print heading and case number. J=2 print heading but not case number. J=3 print case numbeb but not heading.	/TGAP	/(*)	TGAP	W	INTE	IPRINT	
IRA		Temporary variable equal to total number of rings identified thus far.	/RING	/(*)	RING	W	INTE	IRA	
IRC		Denotes total number of rings in molecule.	/BLK4	/(4321)	CHAINM	I	INTE	IRC	
					CORCIG	I	INTE	IRC	
					FUSION	I	INTE	IRC	
					HEXGON	I	INTE	IRC	
					LESSEN	M	INTE	IRC	
					PRINT1	I	INTE	IRC	
					RESETR	I	INTE	IRC	
					RING	M	INTE	IRC	
					SORNGI	I	INTE	IRC	
					STAND	O	INTE	IRC	
					SYMRNG	I	INTE	IRC	
IRCMD		Net number of ring corrections.	/CORCIG/	/(*)	CORCIG	W	INTE	IRCMD	
IRCTOT		Total number of fused and non-fused ring sets in molecule.	/BLK4	/(4323)	FUSION	O	INTE	IRCTOT	
IRCX		Value of IRC entered in subroutine LESSEN.	/CHAINM/	/(*)	CHAINM	I	INTE	IRCX	
IRCX		Computed value for the total number of rings entered into subroutine LESSEN.	/LESSEN/	/(*)	LESSEN	M	INTE	IRCX	
IRCX		Value of IRC entered in subroutine LESSEN.	/RING	/(*)	RING	I	INTE	IRCX	
IRDD	(I)	Contains the order the subscripts of the data in array IDXRD would have if said data were arranged in numerical order.	/BLK1	/(2075)	EQUALR	I	INTE	IRDD	(3)
IRG		Ring Indicator =0 ring(s) not present in molecule >0 ring(s) present in the molecule.	/BLK3	/(6401)	CHAINM	I	INTE	IRG	
					SCAN	M	INTE	IRG	
					STAND	O	INTE	IRG	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
IRING	(I,J)	Contains ring size in J=1 and group numbers of ring components in J>1 of ring number I.	/BLK3 /(	1)	CRINGS	I	INTE	IRING	(40,30)
					CTWO	I	INTE	IRING	(40,30)
					CYCORR	I	INTE	IRING	(40,30)
					FIRSTR	I	INTE	IRING	(40,30)
					FUSION	I	INTE	IRING	(40,30)
					HEXGON	I	INTE	IRING	(40,30)
					LESSEN	M	INTE	IRING	(40,30)
					NRINGS	I	INTE	IRING	(40,30)
					OXYATM	I	INTE	IRING	(40,30)
					PRINTI	I	INTE	IRING	(40,30)
					RESETR	I	INTE	IRING	(40,30)
					RING	M	INTE	IRING	(40,30)
					SORNGI	I	INTE	IRING	(40,30)
					SYMRNG	I	INTE	IRING	(40,30)
IRK		Ring number of ring forming part of ring pair KSUB.	/FUSION/(*	)	FUSION	W	INTE	IRK	
IRNG2		Counter that specifies the number of fused carbon-ring corrections.	/CORCIG/(*	)	CORCIG	W	INTE	IRNG2	
IRNG2		Counter that specifies the number of fused carbon-ring corrections.	/CRINGS/(*	)	CRINGS	M	INTE	IRNG2	
IRNG3		Counter that specifies the number of fused nitrogen-containing-ring corrections.	/CORCIG/(*	)	CORCIG	W	INTE	IRNG3	
IRNG3		Counter that specifies the number of fused nitrogen-containing-ring corrections.	/NRINGS/(*	)	NRINGS	M	INTE	IRNG3	
IRR		Description not input	/GAUCHE/(*	)	GAUCHE	W	INTE	IRR	
ISB		Number of single bonds in ring K.	/CYCORR/(*	)	CYCORR	W	INTE	ISB	
ISB		Number of pairs of non-ring branch atoms found to be equal.	/EQUAL /(*	)	EQUAL	W	INTE	ISB	
ISB		Number of pairs of non-ring branch atoms found to be equal.	/EQUALR/(*	)	EQUALR	I	INTE	ISB	
ISBE		Number of pairs of non-ring atoms forming part of a pair of ligands bonded to two separate rings and which have been found to be equal.	/DELETE/(*	)	DELETE	M	INTE	ISBE	
ISBE		Number of pairs of non-ring atoms forming part of a pair of ligands bonded to two separate rings and which have been found to be equal.	/EQUALR/(*	)	EQUALR	W	INTE	ISBE	
ISBR		Number of pairs of ring atoms found to be equal.	/EQUALR/(*	)	EQUALR	W	INTE	ISBR	
ISER		Equivalent to integer function SEARCH.	/TGAP /(*	)	TGAP	W	INTE	ISER	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE SUBR	U	TYPE	USAGE VAR	DIM
ISET	(I)	Denotes the locations in the reference ring where the ring atoms are identical to atoms in those test rings that are estimated to be completely contained in the reference ring.	/BLK3	/(3253)	LESSEN	M	INTE	ISET	(30)
ISETX	(I)	Denotes the locations in the reference ring where the ring atoms are identical to atoms in those test rings that are estimated to be partially contained in the reference ring.	/BLK3	/(3453)	LESSEN	M	INTE	ISETX	(30)
ISUB		Number of entries in array ICN.	/BLK4	/(362)	STAND	M	INTE	ISUB	
ISUM		Number of atoms bonded to a particular ring which are also ring atoms.	/SORNGI	/( )	SORNGI	W	INTE	ISUM	
IT		Location in data arrays containing thermochemical data for group K.	/TGAP	/( )	TGAP	I	INTE	IT	
ITAG1		Transfer flag.	/CIS	/( )	CIS	W	INTE	ITAG1	
ITAG2		Transfer flag.	/CIS	/( )	CIS	W	INTE	ITAG2	
ITEM		Variable used for temporary storage.	/SCAN	/( )	SCAN	W	INTE	ITEM	
ITEMP		Variable used for temporary storage in exchange of location operation.	/NEWKC	/( )	NEWKC	W	INTE	ITEMP	
ITEST		Number of CO groups in ring adjacent to ring atom LOCX.	/OXYATM	/( )	OXYATM	O	INTE	ITEST	
ITEST		Flag set to one if non-ring ligands are bonded to the ring. Otherwise equal to zero.	/SORNGI	/( )	SORNGI	W	INTE	ITEST	
ITEST2		Specifies the number of CO atoms adjacent to a designated ring atom.	/CYCORR	/( )	CYCORR	I	INTE	ITEST2	
ITZ		Flag: =0 no ring present in molecule. =1 ring present in molecule.	/STAND	/( )	STAND	W	INTE	ITZ	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE SUBR	SUBROUTINE U	SUBROUTINE TYPE	USAGE VAR	USAGE DIM
IX	(I,J,K)	Contains structural data for core atom I at J=1 and for ligands at J=2 to J=5. Contents of column K K=1 chemical symbol of component, K=2 row coordinate of species (from input structural grid), K=3 column coordinate, J=1 and K=4 group number of parent core atom, J not 1 and K=4 direction of ligand bond, J=5 bond type and J=6 group number of core atom.	/BLK2	/(119)	ASYMC	I	INTE	IX	(100,5,6)
					CHAINM	I	INTE	IX	(100,5,6)
					CIS	I	INTE	IX	(100,5,6)
					CISCOR	I	INTE	IX	(100,5,6)
					CORCIG	I	INTE	IX	(100,5,6)
					CTWO	I	INTE	IX	(100,5,6)
					CYCORR	I	INTE	IX	(100,5,6)
					DELTA1	I	INTE	IX	(100,5,6)
					DELTA2	I	INTE	IX	(100,5,6)
					DITERE	I	INTE	IX	(100,5,6)
					EQUAL	I	INTE	IX	(100,5,6)
					EQUALR	I	INTE	IX	(100,5,6)
					EXTROT	I	INTE	IX	(100,5,6)
					FIND	M	INTE	IX	(100,5,6)
					GAUCHE	I	INTE	IX	(100,5,6)
					HEXGON	I	INTE	IX	(100,5,6)
					INTROT	I	INTE	IX	(100,5,6)
					LINEAR	I	INTE	IX	(100,5,6)
					NRINGS	I	INTE	IX	(100,5,6)
					OXYWTR	I	INTE	IX	(100,5,6)
					PRINT1	I	INTE	IX	(100,5,6)
					SCAN	M	INTE	IX	(100,5,6)
					SCANBR	I	INTE	IX	(100,5,6)
					SCANCH	I	INTE	IX	(100,5,6)
					SHIFT	M	INTE	IX	(100,5,6)
					SORNGI	I	INTE	IX	(100,5,6)
					STAND	M	INTE	IX	(100,5,6)
					SYMRNG	I	INTE	IX	(100,5,6)
					SYMTRY	I	INTE	IX	(100,5,6)
IZ		Location J in NBC(I,J) of uppermost branch atom in previous chain.	/CHAINM/(	)	CHAINM	I	INTE	IZ	
IZ		Location J in NBC(I,J) of uppermost branch atom in new or previous chain.	/NEWCOL/(	)	NEWCOL	O	INTE	IZ	
II		Number of chain elements from the start of the chain up to but excluding the branch atom.	/CHANGE/(	)	CHANGE	M	INTE	II	
II		Storage flag for ligand bonded to atom K1. =-1 ligand is not stored and used in the cis test. =0 ligand is stored and used in the cis test. =1 ligand is stored and used in the cis test but II is reset to -1.	/CIS /(	)	CIS	I	INTE	II	
II		Storage flag for K1 ligands used in subroutine CIS.	/CISCOR/(	)	CISCOR	I	INTE	II	
II		Bond type flag of first central test core atom.	/GAUCHE/(	)	GAUCHE	M	INTE	II	
II		Lower limit of location in ORDSUM used in the calculation.	/SEARCH/(	)	SEARCH	M	INTE	II	
II		Storage flag for K1 ligands used in subroutine CIS.	/SORNGI/(	)	SORNGI	M	INTE	II	

17 APR 73 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
I2		Number of chain elements one location above the branch atom to the end of the chain. (upper residual).	/CHANGE/(*	)	CHANGE	W	INTE	I2	
I2		Storage flag for ligand bonded to atom K2. Conditions are the same as for flag I1.	/CIS /(*	)	CIS		INTE	I2	
I2		Storage flag for K2 ligands used in subroutine CIS.	/CISCOR/(*	)	CISCOR	I	INTE	I2	
I2		Bond type flag of second central test core atom.	/GAUCHE/(*	)	GAUCHE	W	INTE	I2	
I2		Upper limit of location in ORDSUM used in the calculation.	/SEARCH/(*	)	SEARCH	W	INTE	I2	
I2		Storage flag for K2 ligands used in subroutine CIS.	/SORNGI/(*	)	SORNGI	W	INTE	I2	
J		Loop counter and subscript.	/CHAINM/(*	)	CHAINM	W	INTE	J	
J		Loop counter and subscript.	/CHANGE/(*	)	CHANGE	W	INTE	J	
J		Loop counter and subscript.	/CORCIG/(*	)	CORCIG	W	INTE	J	
J		Loop counter and subscript.	/CTWO /(*	)	CTWO	W	INTE	J	
J		Loop counter and subscript.	/CYCORR/(*	)	CYCORR	W	INTE	J	
J		Loop counter and subscript.	/EQUAL /(*	)	EQUAL	W	INTE	J	
J		Loop counter and subscript.	/EQUALR/(*	)	EQUALR	W	INTE	J	
J		Loop counter and subscript.	/EXTROT/(*	)	EXTROT	W	INTE	J	
J		Loop counter and subscript.	/FIRSTR/(*	)	FIRSTR	W	INTE	J	
J		Loop counter and subscript.	/FUSION/(*	)	FUSION	W	INTE	J	
J		Loop counter and subscript.	/HEXGON/(*	)	HEXGON	W	INTE	J	
J		Loop counter and subscript.	/IDENT /(*	)	IDENT	W	INTE	J	
J		Loop counter and subscript.	/LESSEN/(*	)	LESSEN	W	INTE	J	

17 APR 73 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
J		Loop counter and subscript.	/MULTI /(*	)	MULTI	W	INTE	J	
J		Loop counter and subscript.	/NEWCOL/(*	)	NEWCOL	W	INTE	J	
J		Loop counter and subscript.	/NEWKC /(*	)	NEWKC	W	INTE	J	
J		Loop counter and subscript.	/NRINGS/(*	)	NRINGS	W	INTE	J	
J		Loop counter and subscript.	/NUMBER/(*	)	NUMBER	W	INTE	J	
J		Loop counter and subscript.	/ORDER /(*	)	ORDER	W	INTE	J	
J		Loop counter and subscript.	/PRINT1/(*	)	PRINT1	W	INTE	J	
J		Loop counter and subscript	/RING /(*	)	RING	W	INTE	J	
J		Loop counter and subscript.	/SAME /(*	)	SAME	W	INTE	J	
J		Loop counter and subscript.	/SCAN /(*	)	SCAN	W	INTE	J	
J		Loop counter and subscript.	/SCANBR/(*	)	SCANBR	W	INTE	J	
J		Loop counter and subscript.	/SCANCH/(*	)	SCANCH	W	INTE	J	
J		Loop counter and subscript.	/SETUP /(*	)	SETUP	W	INTE	J	
J		Loop counter and subscript.	/SHIFT /(*	)	SHIFT	W	INTE	J	
J		Loop counter and subscript.	/SORNGI/(*	)	SORNGI	W	INTE	J	
J		Loop counter and subscript.	/STAND /(*	)	STAND	W	INTE	J	
J		Loop counter and subscript.	/SUMATM/(*	)	SUMATM	W	INTE	J	
J		Loop counter and subscript.	/SYMRNG/(*	)	SYMRNG	W	INTE	J	
J		Loop counter and subscript.	/SYMTRY/(*	)	SYMTRY	W	INTE	J	
J		Loop counter and subscript.	/TGAP /(*	)	TGAP	W	INTE	J	

17 APR 72 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE SUBR	U	TYPE	USAGE VAR	DIM
JA		Loop counter and subscript.	/STAND /(*)		STAND	W	INTE	JA	
JBOND	(I)	Contains ring core atom to which heavy ligand is bonded.	/BLK3 /(3701)		HEXGON	M	INTE	JBOND	(100)
JBR	(I)	Counter designating the location in array LIGAND where the composition data of a particular ligand of core atom I are stored.	/BLK3 /(3302)		ASYMC	I	INTE	JBR	(100)
					EXTROT	I	INTE	JBR	(100)
					SCANDR	M	INTE	JBR	(100)
					SCANCH	M	INTE	JBR	(100)
					SYMTRY	M	INTE	JBR	(100)
JC		Counter limit and subscript.	/MULTI /(*)		MULTI	W	INTE	JC	
JC		Identification number of group whose ligand data storage position is to be altered.	/SHIFT /(*)		SHIFT	I	INTE	JC	
JC		Group number of core atom bonded to core atom KC whose ligands are identified and classified in the scan operation.	/STAND /(*)		STAND	W	INTE	JC	
JDIF		Number of residual ring atoms which are to be added to all the chains.	/RESETR/(*)		RESETR	W	INTE	JDIF	
JDL		Counter denoting number of test rings partially contained in reference ring.	/LESSEN/(*)		LESSEN	W	INTE	JDL	
JDONE	(I)	Location I is set equal to 1 if the ring atom of group number I has been tested. Otherwise, it equals 0.	/BLK3 /(3502)		DELETE	O	INTE	JDONE	(100)
					EQUALR	M	INTE	JDONE	(100)
JDONE	(I)	Aromatic ring flag =0 ring I is a ben. ne or pyridine-type ring forming part of a fused ring set.	/BLK3 /(5801)		HEXGON	M	INTE	JDONE	(40)
JDONE	(I)	Identifies ring ligands which have (1) or have not (0) been checked against other ligands for similarity.	/BLK9 /(2821)		SYMRNG	M	INTE	JDONE	(120)
JDT		Counter denoting number of test rings completely contained in reference ring.	/LESSEN/(*)		LESSEN	W	INTE	JDT	
JD1		Temporary storage variable.	/CYCORR/(*)		CYCORR	W	INTE	JD1	
JD1		Displacement between the first and second CO groups in ring K.	/OXYATM/(*)		OXYATM	W	INTE	JD1	
JD2		Temporary storage variable.	/CYCORR/(*)		CYCORR	W	INTE	JD2	
JD2		Displacement between atom LOCX and the first CO group in ring K.	/OXYATM/(*)		OXYATM	W	INTE	JD2	

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE SUBN	USAGE U	TYPE	VAR	DIM
JF		Upper limit for bond termination loop in subroutine BOND.	/ASSIGN/(	*	) ASSIGN	D	INTE	JF	
JF		Upper limit for bond termination loop.	/BOND /(	*	) BOND	I	INTE	JF	
JF		Upper limit of loop counter.	/CHAINM/(	*	) CHAINM	M	INTE	JF	
JF		Upper limit of loop counter.	/CHANGE/(	*	) CHANGE	M	INTE	JF	
JF		Upper limit of loop counter.	/CORCIG/(	*	) CORCIG	M	INTE	JF	
JF		Upper limit of loop counter.	/CTWO /(	*	) CTWO	M	INTE	JF	
JF		Upper limit of bond termination loop of indexed atom number 1 ligand.	/FIND /(	*	) FIND	I	INTE	JF	
JF		Upper limit of loop counter.	/FUSION/(	*	) FUSION	M	INTE	JF	
JF		Upper limit of loop counter.	/HEXGON/(	*	) HEXGON	M	INTE	JF	
JF		Upper limit for bond termination loop in subroutine BOND.	/IDENT /(	*	) IDENT	I	INTE	JF	
JF		Upper limit of loop counter.	/LESSEN/(	*	) LESSEN	M	INTE	JF	
JF		Upper limit of loop counter.	/NEWKC /(	*	) NEWKC	M	INTE	JF	
JF		Upper limit of bond termination loop.	/NUMBER/(	*	) NUMBER	I	INTE	JF	
JF		Upper limit of loop counter.	/ORDER /(	*	) ORDER	M	INTE	JF	
JF		Upper limit for bond termination loop.	/SCAN /(	*	) SCAN	I	INTE	JF	
JF		Upper limit of loop counter.	/SORNGI/(	*	) SORNGI	M	INTE	JF	
JF		Upper limit of cycle counter.	/STAND /(	*	) STAND	M	INTE	JF	
JFF		Upper limit of bond termination loop for ligand one.	/NUMBER/(	*	) NUMBER	D	INTE	JFF	
JI		Lower limit of bond termination loop in subroutine BOND.	/ASSIGN/(	*	) ASSIGN	D	INTE	JI	
JI		Lower limit of bond termination loop.	/BOND /(	*	) BOND	I	INTE	JI	

17 APR 73 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
JI		Lower limit of loop counter.	/CTWO /(*	)	CTWO	M	INTE	JI	
JI		Lower limit of bond termination loop of indexed atom number 1 ligand.	/FIND /(*	)	FIND	I	INTE	JI	
JI		Lower limit of bond termination loop in subroutine BOND.	/IDENT /(*	)	IDENT	I	INTE	JI	
JI		Lower limit of bond termination loop.	/NUMBER/(*	)	NUMBER	I	INTE	JI	
JI		Lower limit of bond termination loop.	/SCAN /(*	)	SCAN	I	INTE	JI	
JII		Lower limit of bond termination loop for ligand one.	/NUMBER/(*	)	NUMBER	O	INTE	JII	
JJ		Loop counter.	/BOND /(*	)	BOND	O	INTE	JJ	
JJ		Ring location para to nitrogen atom.	/CYCORR/(*	)	CYCORR	M	INTE	JJ	
JJ		Variable used for temporary storage.	/DITERE/(*	)	DITERE	M	INTE	JJ	
JJ		Loop counter.	/EQUAL /(*	)	EQUAL	M	INTE	JJ	
JJ		Loop counter and subscript.	/FUSION/(*	)	FUSION	M	INTE	JJ	
JJ		Ring number.	/HEXGON/(*	)	HEXGON	M	INTE	JJ	
JJ		Identification number of test ring.	/LESSEN/(*	)	LESSEN	M	INTE	JJ	
JJ		Counter ultimately equal to NOBR.	/STAND /(*	)	STAND	M	INTE	JJ	
JK		Ring number or non-ring ligand weight identifier.	/HEXGON/(*	)	HEXGON	M	INTE	JK	
JOR		Number of entries in array JROND.	/HEXGON/(*	)	HEXGON	M	INTE	JOR	
JORD	(I)	Contains subscripts of IRING data arranged in order of increasing ring size.	/BLK3 /(3053)		LESSEN	I	INTE	JORD	(40)
JPRINT		Equal to digit K of IPRINT (OJK).	/TGAP /(*	)	TGAP	M	INTE	JPRINT	
JRC	(I,J)	Contains incompatability factors and ring numbers of JRING data pertaining to test rings that are partially contained in the reference ring.	/BLK3 /(3153)		LESSEN	M	INTE	JRC	(40,2)

17 JAN 73 0 03:36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
JRDIF	(I)	Contains the computed differences between the ring size and the number of atoms of the test rings that are also contained in the reference ring. Referred to as incompatability factor.	/BLK3	/(3103)	LESSEN	M	INTE	JRDIF	(40)
JRING	(I,J)	Contains the negative group number of test ring atoms not contained in reference ring or the ring locations in the reference ring where the test ring atoms are located. The value is stored in location J which corresponds to the location of the test ring atom.	/BLK3	/(1855)	LESSEN	M	INTE	JRING	(40,30)
JS		Lower limit of character search cycle counter.	/IDENT	/( )	IDENT	M	INTE	JS	
JSCAN	(I)	Scan flag =0 ring I has not been tested. =1 ring I has been tested.	/BLK3	/(3561)	FUSION	M	INTE	JSCAN	(40)
JSTOP		I location in arrays IRING(I,J) and NRLOC(I) to which ring data are shifted when one or more rings are discarded.	/LESSEN/	( )	LESSEN	M	INTE	JSTOP	
JSTORE	(I)	Contains ring numbers of six-membered rings composed of 4 carbon and 2 nitrogen atoms and devoid of double bonds.	/BLK3	/(6398)	NRINGS	M	INTE	JSTORE	(3)
JSUM		Counter denoting number of atoms of a particular test ring that are also present in the reference ring.	/LESSEN/	( )	LESSEN	M	INTE	JSUM	
JT		Identification number of group to be tested for second-order multiple-bond weight contributions. If present, these are added to group weight of XT.	/DELTA2/	( )	DELTA2	I	INTE	JT	
JUNCT	(I,J)	Contains group number of core atom ligand I bonded to branch atom J where J is also the group number of the branch atom.	/BLK3	/(1203)	CHAINM NEWKC	O M	INTE INTE	JUNCT JUNCT	(5,100) (5,100)
JW		Subscript for printout legend. Setting denotes whether the external symmetry number can be computed.	/BLK5	/(59)	CORCIG ENTSYM EXTROT FIRSTR LINEAR SYMING	O I O O O O	INTE INTE INTE INTE INTE INTE	JW JW JW JW JW JW	
JX		Subscript.	/FIRSTR/	( )	FIRSTR	M	INTE	JX	

17 APR 79 8.02-36

FORTRAN SYMBOL	MAYN SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE SUBN	USAGE U	TYPE	VAR	DIM
JX		Subscript.	/LESSEN/(	*	) LESSEN	W	INTE	JX	
JX		Temporary storage variable.	/ORDER /(	*	) ORDER	W	INTE	JX	
JY		Subscript for printout legend. Setting denotes whether the number of enantiomers and meso compounds can be computed in total.	/BLK5 /(	(0)	CORCIG	O	INTE	JY	
					ENTSYM	I	INTE	JY	
					FIRSTR	O	INTE	JY	
					MAXCHN	O	INTE	JY	
J1		Subscript of one of the two ligands under comparison.	/EQUAL /(	*	) EQUAL	W	INTE	J1	
J1		Flag which is activated if character symbol is not found in cycle.	/IDENT /(	*	) IDENT	W	INTE	J1	
J1		Location in IX of data of non-core ligand.	/SCAN /(	*	) SCAN	W	INTE	J1	
J1		Subscript of array KSAME.	/SETUP /(	*	) SETUP	W	INTE	J1	
J1		Subscript and flag.	/SYMRNG/(	*	) SYMRNG	W	INTE	J1	
J2		Subscript of one of the two ligands under comparison.	/EQUAL /(	*	) EQUAL	W	INTE	J2	
J2		Flag which is activated if character symbol is found in cycle.	/IDENT /(	*	) IDENT	W	INTE	J2	
J2		Subscript.	/LESSEN/(	*	) LESSEN	W	INTE	J2	
J2		Location in IX of data of core ligand.	/SCAN /(	*	) SCAN	W	INTE	J2	
J2		Subscript of array KSAME.	/SETUP /(	*	) SETUP	W	INTE	J2	
J2		Subscript.	/SYMRNG/(	*	) SYMRNG	W	INTE	J2	
K		Row coordinate of indexed core atom.	/ASSIGN/(	*	) ASSIGN	I	INTE	K	
K		Loop counter and subscript.	/ASYMC /(	*	) ASYMC	W	INTE	K	
K		Loop counter and subscript.	/CISCOR/(	*	) CISCOR	W	INTE	K	
K		Loop counter and subscript. Also I.D. number of ring examined in subroutine CYCORR.	/CORCIG/(	*	) CORCIG	W	INTE	K	
K		Identification number of ring.	/CTWO /(	*	) CTWO	W	INTE	K	

17 APR 73 0.02.36

FORTRAN SYMBOL	PAYM SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
K		Identification number of ring.	/CYCORR/(		) CYCORR	I	INTE	K	
K		Loop counter and subscript.	/DELETE/(		) DELETE	O	INTE	K	
K		Loop counter and subscript.	/ENTSYM/(		) ENTSYM	M	INTE	K	
K		Loop counter and subscript.	/EQUAL /(		) EQUAL	M	INTE	K	
K		Loop counter and subscript.	/EQUALR/(		) EQUALR	M	INTE	K	
K		Loop counter and subscript.	/EXTROT/(		) EXTROT	M	INTE	K	
K		Entry value: row coordinate of indexed atom. Exit value: row coordinate of core atom.	/FIND /(		) FIND	M	INTE	K	
K		Loop counter and subscript.	/FIRSTR/(		) FIRSTR	M	INTE	K	
K		Loop counter and subscript.	/FUSION/(		) FUSION	M	INTE	K	
K		Loop counter and subscript.	/HEXGON/(		) HEXGON	M	INTE	K	
K		Row coordinate of first character of symbol.	/IDENT /(		) IDENT	M	INTE	K	
K		Loop counter and subscript.	/INTROT/(		) INTROT	M	INTE	K	
K		Identification number of reference ring.	/LESSEN/(		) LESSEN	M	INTE	K	
K		Loop counter and subscript.	/LINEAR/(		) LINEAR	M	INTE	K	
K		Loop counter and subscript.	/MAXCHN/(		) MAXCHN	M	INTE	K	
K		Loop counter and subscript.	/MULTI /(		) MULTI	M	INTE	K	
K		Loop counter and subscript.	/NRINGS/(		) NRINGS	M	INTE	K	
K		Row coordinate of indexed core atom.	/NUMBER/(		) NUMBER	I	INTE	K	
K		Ring number of ring under inspection.	/OXYATM/(		) OXYATM	I	INTE	K	
K		Loop counter and subscript, also ring number.	/RESETR/(		) RESETR	M	INTE	K	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE		SUBROUTINE USAGE			
			D COM	LOC	SCAL	U	TYPE	VAR DIM
K		Loop counter and subscript.	/RING	/(*	) RING	M	INTE	K
K		Loop counter and subscript.	/SAME	/(*	) SAME	M	INTE	K
K		Row coordinate of core atom in array GRID.	/SCAN	/(*	) SCAN	I	INTE	K
K		Loop counter and subscript.	/SETUP	/(*	) SETUP	M	INTE	K
K		Loop counter and subscript.	/SORNG!	/(*	) SORNGI	M	INTE	K
K		Row grid coordinate of core atom.	/STAND	/(*	) STAND	M	INTE	K
K		Loop counter and subscript.	/SYMRNG	/(*	) SYMRNG	M	INTE	K
K		Loop counter and subscript.	/SYMTRY	/(*	) SYMTRY	M	INTE	K
K		Loop counter and subscript.	/TGAP	/(*	) TGAP	M	INTE	K
KAFTER		Transfer flag which prevents storage of the ligand data already defined at the start of the atomic composition evaluation cycle.	/SCANBR	/(*	) SCANBR	I	INTE	KAFTER
KAFTER		Transfer flag which prevents storage of the ligand data already defined at the start of the atomic composition evaluation cycle.	/SYMTRY	/(*	) SYMTRY	M	INTE	KAFTER
KANE		Denotes which of the two central atoms has no unsaturated bonds.	/GAUCHE	/(*	) GAUCHE	M	INTE	KANE
KARO	(I)	Ring locations (I) containing identical ligands are assigned the same identifiers (KS) in this array.	/SLKS	/(4308)	SYMRNG	M	INTE	KARO (6)
KB		Value of KY at the beginning of the cycle.	/FUSION	/(*	) FUSION	M	INTE	KB
KBENZ	(I)	Contains the ring numbers of the benzene type rings present in the molecule.	/BLK3	/(5453)	CYCORR	O	INTE	KBENZ (NO)
					KEXCOM	I	INTE	KBENZ (NO)
					SYLRNG	I	INTE	KBENZ (NO)
KBU		Counter that specifies whether one or both ligands in the cis position are carbon atoms with a connectivity of four and bonded to only one heavy atom.	/CISCOM	/(*	) CISCOM	M	INTE	KBU
KC		Group number of chain atom being processed.	/CHAINM	/(*	) CHAINM	M	INTE	KC

17 APR 73 0.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE SUBR	SUBROUTINE U	SUBROUTINE TYPE	USAGE VAR	USAGE DIM
KC		Group number of core atom bonded to ring atom KI.	/CTWO /(*	)	CTWO	W	INTE	KC	
KC		Group number of atom whose ligands are being compared.	/EQUALR/(*	)	EQUALR	I	INTE	KC	
KC		Group number.	/FIND /(*	)	FIND	I	INTE	KC	
KC		Group number of atom whose ligands are being compared.	/FIRSTR/(*	)	FIRSTR	I	INTE	KC	
KC		Group number of ligand bonded to saturated central atom.	/GAUCHE/(*	)	GAUCHE	W	INTE	KC	
KC		Loop counter and subscript.	/HEXGON/(*	)	HEXGON	W	INTE	KC	
KC		Group number of atom whose ligand is to be tested for linearity and rotational symmetry.	/LINEAR/(*	)	LINEAR	I	INTE	KC	
KC		Test branch atom in chain LX which may serve as the junction for the new chain.	/NEWCOL/(*	)	NEWCOL	M	INTE	KC	
KC		Group number of chain atom being processed.	/NEWKC /(*	)	NEWKC	M	INTE	KC	
KC		Group number of ring component.	/RESETR/(*	)	RESETR	W	INTE	KC	
KC		Group number of core atom under which the data are stored in each of the symmetry identification arrays.	/SAME /(*	)	SAME	I	INTE	KC	
KC		Group number of chain atom currently being processed by SCANCH.	/SCANCH/(*	)	SCANCH	I	INTE	KC	
KC		Identification number of core atom whose data are to be stored in J=2 of core atom J=JC.	/SHIFT /(*	)	SHIFT	I	INTE	KC	
KC		Group number. Also designates core atom of group.	/STAND /(*	)	STAND	W	INTE	KC	
KC		Group number of ring component.	/SYMRNG/(*	)	SYMRNG	W	INTE	KC	
KC		Group number of atom under inspection.	/SYMTRY/(*	)	SYMTRY	W	INTE	KC	
KCANE		Group number of central atom devoid of unsaturated bonds.	/GAUCHE/(*	)	GAUCHE	W	INTE	KCANE	
KCBEF		Group number of atom processed immediately prior to atom KC.	/SCANCH/(*	)	SCANCH	I	INTE	KCBEF	
KCBEF		Group number of chain atom previously processed. If none, it is set to 1000.	/SYMTRY/(*	)	SYMTRY	W	INTE	KCBEF	

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
KCC		Total number of core atoms in molecular structure.	/BLK2	/(3219)	ASYMC	I	INTE	KCC	
					CHAINM	I	INTE	KCC	
					CORCIG	I	INTE	KCC	
					CTWO	I	INTE	KCC	
					EQUAL	I	INTE	KCC	
					EQUALR	I	INTE	KCC	
					HEXGON	I	INTE	KCC	
					INTROT	I	INTE	KCC	
					PRINTI	I	INTE	KCC	
					SCAN	M	INTE	KCC	
					SETUP	I	INTE	KCC	
					SORNGI	I	INTE	KCC	
					STAND	M	INTE	KCC	
					SYMRNG	I	INTE	KCC	
					SYMTRY	I	INTE	KCC	
					TGAP	I	INTE	KCC	
KCCR	(I,J)	Contains the group numbers I of each pair J of ring atoms, one from each ring, which have been tested.	/BLK1	/(1125)	DELETE	I	INTE	KCCR	(2,30)
					EQUALR	O	INTE	KCCR	(2,30)
KCENE		Group number of central atom containing an unsaturated bond.	/GAUCHE/(*)		GAUCHE	M	INTE	KCENE	
KCGAUS	(I,J)	Contains the group numbers I of the central atoms associated with the second-order interaction J.	/BLK3	/(4751)	CISCOR	O	INTE	KCGAUS	(3,150)
					CORCIG	M	INTE	KCGAUS	(3,150)
					CRINGS	O	INTE	KCGAUS	(3,150)
					CTWO	O	INTE	KCGAUS	(3,150)
					DITERE	M	INTE	KCGAUS	(3,150)
					GAUCHE	O	INTE	KCGAUS	(3,150)
					HEXGON	O	INTE	KCGAUS	(3,150)
					INTROT	M	INTE	KCGAUS	(3,150)
					RNINGS	O	INTE	KCGAUS	(3,150)
KCIS		Number of cis interactions (1 or 2) exhibited by a particular pair of test core atoms.	/CISCOR/(*)		CISCOR	M	INTE	KCIS	
KCL		Counter that specifies whether the atom one removed from the first central test core atom is a carbon atom.	/CISCOR/(*)		CISCOR	M	INTE	KCL	
KCNEXT		Group number of atom adjacent to atom KC1.	/EXTROT/(*)		EXTROT	M	INTE	KCNEXT	
KCNEXT		Group number of atom in ligand to be tested which is bonded to atom KC.	/LINEAR/(*)		LINEAR	I	INTE	KCNEXT	
KCNEXT		Group number of branch atom currently being processed by SCANBR.	/SCANBR/(*)		SCANBR	I	INTE	KCNEXT	
KCNEXT		Group number of atom to be processed immediately after atom KC.	/SCANCH/(*)		SCANCH	I	INTE	KCNEXT	
KCNEXT		Group number of next atom down the chain.	/SYMTRY/(*)		SYMTRY	M	INTE	KCNEXT	
KCPV		Group number of chain atom preceding atom KC.	/CHAINM/(*)		CHAINM	M	INTE	KCPV	

17 APR 73 0.01:26

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
KCPV		Chain atom preceding atom KC.	/NEWCOL/(*		NEWCOL	R	INTE	KCPV	
KCPV		Group number of chain atom preceding atom KC.	/NEWKC /(*		NEWKC	R	INTE	KCPV	
KCPV		Group number of parent core atom, namely, core atom processed immediately prior to KC.	/STAND /(*		STAND	W	INTE	KCPV	
KCR		Counter that specifies whether the atom one removed from the second central test core atom is a carbon atom.	/CISCOR/(*		CISCOR	W	INTE	KCR	
KCSAME	(I,J)	Group number of identical ligand I associated with core atom J.	/BLK1 /(2078)		ASYMC	I	INTE	KCSAME(6,100)	
					EXTROT	I	INTE	KCSAME(6,100)	
					LINEAR	I	INTE	KCSAME(6,100)	
					SAME	R	INTE	KCSAME(6,100)	
					SYMRNG	I	INTE	KCSAME(6,100)	
					SYMTRY	O	INTE	KCSAME(6,100)	
KCSUDA		Group number of pseudoasymmetric atom.	/CORCIG/(*		CORCIG	I	INTE	KCSUDA	
KCSUDA		Group number of pseudoasymmetric atom.	/ENTSYM/(*		ENTSYM	I	INTE	KCSUDA	
KCSUDA		Group number of pseudoasymmetric atom.	/EXTROT/(*		EXTROT	O	INTE	KCSUDA	
KCT		Variable equal to the number of atoms in common between ring pair KSUB.	/FUSION/(*		FUSION	W	INTE	KCT	
KCX		Subscript.	/CISCOR/(*		CISCOR	W	INTE	KCX	
KCX		Variable used for temporary storage.	/NEWCOL/(*		NEWCOL	W	INTE	KCX	
KCX		Group number denoting the location in the arrays where the data are to be stored in subroutine SOME.	/SETUP /(*		SETUP	W	INTE	KCX	
KCX		Group number of ring atom.	/SYMRNG/(*		SYMRNG	I	INTE	KCX	
KCXDOI	(I)	Group number information input to subroutine EQUAL. I=1 KC or 1000. I=2 K1.	/CTWO /(*		CTWO	W	INTE	KCXDOI(3)	
KCXDOI	(I)	Contains the group number(s) (set A) of the core atom(s) whose ligands are to be compared structurewise with those of set B.	/EQUAL /(*		EQUAL	I	INTE	KCXDOI(3)	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
KCXDO1	(I)	Contains the group number(s) (set A) of the core atom(s) whose ligands are to be compared structurewise with those of set B.	/SETUP /	(*)	SETUP	I	INTE	KCXDO1	(3)
KCXDO1	(I)	Contains the group numbers of the first three ring components.	/SYMRNG/	(*)	SYMRNG	W	INTE	KCXDO1	(3)
KCXDO2		Same as KCXDO1(1).	/CTWO /	(*)	CTWO	W	INTE	KCXDO2	
KCXDO2	(I)	Contains the group number(s) (set B) of the core atom(s) whose ligands are to be compared structurewise with those of set A.	/EQUAL /	(*)	EQUAL	I	INTE	KCXDO2	(3)
KCXDO2	(I)	Contains the group number(s) (set B) of the core atom(s) whose ligands are to be compared structurewise with those of set A.	/SETUP /	(*)	SETUP	I	INTE	KCXDO2	(3)
KCXDO2	(I)	Contains the group numbers of the last three ring components.	/SYMRNG/	(*)	SYMRNG	W	INTE	KCXDO2	(3)
KCY		Ordinal number used to identify the non-ring ligand data of a pair of rings which has been tested for symmetry. Also number of entries in arrays AKT and MER.	/BLK3	(1 6397)	EQUAL EQUALR SYMTRY	M M O	INTE INTE INTE	KCY KCY KCY	
KC1		Group number of first carbon atom bonded to the oxygen atom.	/DITERE/	(*)	DITERE	W	INTE	KC1	
KC1		Group number of atom contained in array KCXDO1 and whose ligands are being compared.	/EQUAL /	(*)	EQUAL	W	INTE	KC1	
KC1		Group number of central atom in longest chain if chain is odd or one of the two central atoms if chain is even.	/EXTROT/	(*)	EXTROT	W	INTE	KC1	
KC1		Group number of atom contained in array KCXDO1.	/SETUP /	(*)	SETUP	W	INTE	KC1	
KC1		Group number under which the symmetry data of the ring are stored.	/SYMRNG/	(*)	SYMRNG	W	INTE	KC1	
KC1ALT	(I)	Temporary storage for original KCSAVE data computed for KC1.	/BLK1	(1 3883)	EXTROT	M	INTE	KC1ALT	(6)
KC1P		Group number of atom previous (parent) to atom K1.	/EQUAL /	(*)	EQUAL	W	INTE	KC1P	
KC1P		Group number of atom previous to the entry ring atom K1.	/EQUALR/	(*)	EQUALR	I	INTE	KC1P	
KC1PR		Group number of atom previous (parent) to atom K1R.	/EQUALR/	(*)	EQUALR	W	INTE	KC1PR	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
KC2		Group number of second carbon atom bonded to the oxygen atom.	/DITERE/(	)	DITERE	W	INTE	KC2	
KC2		Group number of atom contained in array KCXD02 and whose ligands are being compared.	/EQUAL /(	)	EQUAL	W	INTE	KC2	
KC2		Group number of atom adjacent to atom KC1.	/EXTROT/(	)	EXTROT	W	INTE	KC2	
KC2		Group number of atom contained in array KCXD02.	/SETUP /(	)	SETUP	W	INTE	KC2	
KC2ALT	(I)	Temporary storage for original KCSAME data computed for KC2.	/BLK1 /(	3889)	EXTROT	W	INTE	KC2ALT	(6)
KC2P		Group number of atom previous (parent) to atom K2.	/EQUAL /(	)	EQUAL	W	INTE	KC2P	
KC2P		Group number of atom previous to the entry ring atom K2.	/EQUALR/(	)	EQUALR	I	INTE	KC2P	
KC2PR		Group number of atom previous (parent) to atom K2R.	/EQUALR/(	)	EQUALR	W	INTE	KC2PR	
KC3		Counter used in the search of 3-ene structures.	/CISCOR/(	)	CISCOR	W	INTE	KC3	
KC3		Group number of atom adjacent to atom KC1.	/EXTROT/(	)	EXTROT	W	INTE	KC3	
KD		Assigned change in row coordinate of symbol input array.	/ASSIGN/(	)	ASSIGN	O	INTE	KD	
KD		Change in row coordinate.	/BOND /(	)	BOND	I	INTE	KD	
KD		Upper limit of loop counter.	/DELETE/(	)	DELETE	W	INTE	KD	
KD		Change in row coordinate for indexed atom number 1 ligand.	/FIND /(	)	FIND	I	INTE	KD	
KD		Change in row coordinate	/IDENT /(	)	IDENT	W	INTE	KD	
KD		Change in row coordinate of array GRID.	/NUMBER/(	)	NUMBER	I	INTE	KD	
KD		Change in row coordinate for transformation to NX.	/SCAN /(	)	SCAN	I	INTE	KD	
KDD		Change in row coordinate of ligand one.	/NUMBER/(	)	NUMBER	O	INTE	KDD	
KD2		KD value of second character.	/IDENT /(	)	IDENT	W	INTE	KD2	
KENE		Denotes which of the two central atoms has an unsaturated bond.	/GAUCHE/(	)	GAUCHE	W	INTE	KENE	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
KEQUAL		Counter that specifies the number of different atoms which are contained in equal amounts by two ligands.	/SETUP /(*	)	SETUP	W	INTE	KEQUAL	
KF		Upper limit of loop counter.	/CIS /(*	)	CIS	W	INTE	KF	
KF		Upper limit of loop counter.	/CORCIG/(*	)	CORCIG	W	INTE	KF	
KF		Upper limit of loop counter that controls the selection of the test ligand from set B.	/EQUAL /(*	)	EQUAL	I	INTE	KF	
KF		Subscript corresponding to highest value of data in IDXRD.	/EQUALR/(*	)	EQUALR	W	INTE	KF	
KF		Upper limit of loop counter.	/FIRSTR/(*	)	FIRSTR	W	INTE	KF	
KF		Upper limit of loop counter.	/FUSION/(*	)	FUSION	W	INTE	KF	
KF		Upper limit of loop counter.	/LINEAR/(*	)	LINEAR	W	INTE	KF	
KF		Upper limit of counter.	/RING /(*	)	RING	W	INTE	KF	
KF		Upper limit of loop counter.	/SAME /(*	)	SAME	W	INTE	KF	
KF		Upper limit of loop counter that controls the selection of the test ligand from set B.	/SETUP /(*	)	SETUP	I	INTE	KF	
KF		Upper limit of loop counter	/SOHNGI/(*	)	SOHNGI	W	INTE	KF	
KF		Description not input	/SYMRNG/(*	)	SYMRNG	W	INTE	KF	
KFGAUS	(I)	Frequency or number of corrections of the second-order interaction I. In CRINGS, it contains subscript of data array associated with ring correction.	/BLK3 /(4301	)	CISCOR CORCIG CRINGS DITERE GAUCHE	O M O O M	INTE INTE INTE INTE INTE	KFGAUS(150) KFGAUS(150) KFGAUS(150) KFGAUS(150) KFGAUS(150)	
KFINST		Group number of first atom processed in composition cycle.	/SYMTRY/(*	)	SYMTRY	W	INTE	KFINSY	
KFLAG		Flag that denotes whether data in arrays NTOTAL, KCSAME, and NOSAME were (1) or were not (0) transferred to the temporary storage array).	/EXTROT/(*	)	EXTROT	W	INTE	KFLAG	
KFLAG		Flag that designates the weight correction for ring atoms.	/HEXGON/(*	)	HEXGON	W	INTE	KFLAG	
KGMAT	(I, J)	Contains alkane C-C gauche corrections.	/GAUCHE/(*	)	GAUCHE	C	INTE	KGMAT	(9,9)

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK LOC	SUBROUTINE SUBR U	USAGE TYPE VAR DIM
KGG		Atom type flag. =1 one of the two central atoms is a nitrogen atom. =2 the two central atoms are carbon atoms.	/GAUCHE/(*	) GAUCHE M	INTE KGG
KGRID	(I)	Contents of input card containing structure or label of molecule shifted and centered for printout purposes.	/BLK4 /(361)	TGAP M	INTE KGRID (103)
KHEAVY		Group number of carbon test core atom bonded to oxygen atom forming part of ether linkage.	/GAUCHE/(*	) GAUCHE M	INTE KHEAVY
KI		Lower limit of loop counter.	/CORCIG/(*	) CORCIG M	INTE KI
KI		Lower limit of loop counter that controls the selection of the test ligand from set B.	/EQUAL /(*	) EQUAL M	INTE KI
KI		Subscript corresponding to lowest value of data in IDXRD.	/EQUALR/(*	) EQUALR M	INTE KI
KI		Lower limit of loop counter.	/HEXGON/(*	) HEXGON M	INTE KI
KI		Lower limit of loop counter.	/MAXCHN/(*	) MAXCHN M	INTE KI
KI		Lower limit of loop counter.	/MULTI /(*	) MULTI M	INTE KI
KI		Location in array SUM corresponding to location I in ORDSUM.	/SEARCH/(*	) SEARCH M	INTE KI
KI		Lower limit of loop counter that controls the selection of the test ligand from set B.	/SETUP /(*	) SETUP M	INTE KI
KI		Description not input	/SYMRNG/(*	) SYMRNG M	INTE KI
KINT	(I)	Contains the group number of atoms whose internal rotational symmetry is included in the external rotational symmetry of the molecule.	/BLK1 /(3719)	EXTROT O INTROT I	INTE KINT (100) INTE KINT (100)
KJ	(I)	Contains the location of the ring atom bonded to the non-ring atom I.	/BLK4 /(2641)	SYMRNG M	INTE KJ (100)
KJ	(I)	Contains I.d. numbers of the two rings in the fused ring set.	/BLK4 /(9288)	CRINGS M	INTE KJ (2)
KJ		Group number of ring component.	/CYCORR/(*	) CYCORR M	INTE KJ

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE SUBR	U	TYPE	USAGE VAR	DIM
KK		Counter of ring test cycle. Also denotes location in chain LX that contains the first atom of the ring.	/CHAINM/(*		) CHAINM	W	INTE	KK	
KK		Loop counter and subscript.	/CIS /(*		) CIS	W	INTE	KK	
KK		Temporary storage variable.	/CORCIG/(*		) CORCIG	W	INTE	KK	
KK		Loop counter and subscript.	/CRINGS/(*		) CRINGS	W	INTE	KK	
KK		Test standard for the locations of the double bonds in ring K.	/CYCORR/(*		) CYCORR	W	INTE	KK	
KK		Subscript of array KCCR.	/DELETE/(*		) DELETE	W	INTE	KK	
KK		Loop counter.	/EQUAL /(*		) EQUAL	W	INTE	KK	
KK		Loop counter and subscript.	/FUSION/(*		) FUSION	W	INTE	KK	
KK		Loop counter and subscript.	/HEXGON/(*		) HEXGON	W	INTE	KK	
KK		Subscript.	/LESSEN/(*		) LESSEN	W	INTE	KK	
KK		Loop counter and subscript.	/NRINGS/(*		) NRINGS	W	INTE	KK	
KK		Location J in NBC(I,J) where first atom of ring is located. Also used as a deletion flag if identical chain formations have been defined.	/RING /(*		) RING	M	INTE	KK	
KK		Transfer flag.	/SYMRNG/(*		) SYMRNG	W	INTE	KK	
KL		Number of heavy atoms bonded to the first central test core atom KI that are capable of exhibiting a gauche interaction.	/GAUCHE/(*		) GAUCHE	W	INTE	KL	
KLM		Counter that specifies whether two ligands cis to each other are both hydrogen atoms.	/CISCOR/(*		) CISCOR	W	INTE	KLM	
KM		Loop counter and subscript.	/DELTA1/(*		) DELTA1	W	INTE	KM	
KM		Loop counter and subscript.	/DELTA2/(*		) DELTA2	W	INTE	KM	
KM		Group number of atom bonded to ring atom KC.	/SYMRNG/(*		) SYMRNG	W	INTE	KM	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE SUBR	SUBROUTINE U	SUBROUTINE TYPE	USAGE VAR	USAGE DIM
KMAX		Number of core atom constituents of longest chain.	/MAXCHN/(1	)	MAXCHN	M	INTE	KMAX	
KNBENZ	(I)	Contains ring numbers of pyridine-type ring structures.	/BLK1	/(3976)	CYCORR	U	INTE	KNBENZ	(40)
					HEXGON	I	INTE	KNBENZ	(40)
KNMAT	(I,J)	Contains alkane c-n corrections.	/GAUCHE/(1	)	GAUCHE	C	INTE	KNMAT	(9,5)
KNO		Number of characters in multi-word input symbol.	/MULTI /(1	)	MULTI	I	INTE	KNO	
KNT	(I,J)	Contains the locations I in array IDX where the group numbers of the test ligands associated with the branch pair J are found.	/BLK1	/(1625)	EQUAL	M	INTE	KNT	(2,99)
					EQUALR	O	INTE	KNT	(2,99)
KNTR	(I,J)	Contains the locations I in array IDXR where the group numbers of the test ligands associated with the ring pair J are found.	/BLK1	/(1065)	EQUALR	M	INTE	KNTR	(2,30)
KOMB	(I,J)	Contains the ring number (I) of the ring component in fused ring set (J).	/BLK3	/(5962)	CRIN[illegible]		INTE	KOM[illegible]	,10)
					[illegible]m		INTE	KO[illegible]	0,10)
					FUSION	M	INTE	KO[illegible]	0,10)
					HEXGON	I	INTE	[illegible]	40,10)
					NRINGS	I	INTE	[illegible]	(40,10)
KOMMON	(I,J,K)	Multi-storage array J=1,2 ring numbers of fused ring pair I in fused ring set K. J=3 number of atoms which ring pair has in common. J>3 group numbers of these atoms.	/BLK1	/(26)	FUSION	M	INTE	KOMMON	(40,9,10)
KON	(I)	Number of ligands bonded to core atom I.	/BLK3	/(5401)	CIS	I	INTE	KON	(100)
					CISCOR	I	INTE	KON	(100)
					CYCO[illegible]	I	INTE	KON	(100)
					EQUAL	I	INTE	KON	(100)
					EQUALR	I	INTE	KON	(100)
					EXTROT	I	INTE	KON	(100)
					GAUCHE	I	INTE	KON	(100)
					INTROT	I	INTE	KON	(100)
					LINEAR	I	INTE	KON	(100)
					PRINTI	I	INTE	KON	(100)
					SCANBR	I	INTE	KON	(100)
					SCANCH	I	INTE	KON	(100)
					SORNGI	I	INTE	KON	(100)
					STAND	O	INTE	KON	(100)
KONF		Connectivity of atom KCI plus one	/EXTROT/(1	)	EXTROT	M	INTE	KONF	
KONMAX		Upper limit of loop counter.	/CYCORR/(1	)	CYCORR	M	INTE	KONMAX	

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
KONT		Upper limit of loop counter.	/EQUAL /(	*)	EQUAL	W	INTE	KONT	
KONT		Upper limit of loop counter.	/EQUALR/(	*)	EQUALR	W	INTE	KONT	
KOP		Number of asymmetric atoms present in a particular chain.	/MAXCHN/(	*)	MAXCHN	W	INTE	KOP	
KOPMAX		Greatest number of asymmetric atoms present in a particular chain.	/MAXCHN/(	*)	MAXCHN	W	INTE	KOPMAX	
KORD	(I)	Contains the order that the subscripts of the data in KREM would have if said data were arranged in increasing numerical order.	/BLK3	/(3553)	LESSEN	M	INTE	KORD	(40)
KOXY		Group number of oxygen test core atom.	/GAUCHE/(	*)	GAUCHE	W	INTE	KOXY	
KP		Addition or subtraction operation flag	/IDENT /(	*)	IDENT	W	INTE	KP	
KP		Integer equal to ±1.	/SUMATN/(	*)	SUMATN	I	INTE	KP	
KPERM		Group number of non-ring ligand bonded to ring atom KC.	/SYMRNG/(	*)	SYMRNG	W	INTE	KPERM	
KPERT	(I)	Contains the group numbers of the ring components.	/BLK4	/(4268)	SYMRNG	M	INTE	KPERT	(6)
KPRINT		Equal to digit J of IPRINT (OJK).	/TGAP	/(*)	TGAP	W	INTE	KPRINT	
KR		Number of heavy atoms bonded to the second central test core atom K2 are available of exhibiting a the interaction.	/GAUCHE/(	*)	GAUCHE	W	INTE	KR	
KR		Subscript of array KREM.	/LESSEN/(	*)	LESSEN	W	INTE	KR	
KRCNOR	(I)	This array applies only to rings containing one nitrogen atom in the ring structure. Locations I=K1 and I=K2 contains the group number j of the nitrogen atom which is adjacent to the carbon atoms K1 and K2. Location I=K3, where K3 is the group number of the ring atom para to the nitrogen atom, contains -j.	/(K1	/(3775)	CORCIG CYCORR HEXGON	O O I	INTE INTE INTE	KRCNOR(100) KRCNOR(100) KRCNOR(10	
KRCNWT	(I)	Contains flags that govern the type of identifying weight correction to be applied to ring atom I.	/BLK1	/(3875)	CORCIG CYCORR HEXGON	O O I	INTE INTE INTE	KRCNWT(100) KRCNWT(100) KRCNWT(100)	

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
KREM	(I)	Contains the ring number of the rings that are not mutually exclusive and hence are to be discarded.	/BLK3	/(3303)	LESSEN	M	INTE	KREM	(40)
KRING1		Ring comparison flag. =0 the two ring-containing ligands are not equal. =1 they are equal.	/EQUAL	/( )	EQUAL	I	INTE	KRING1	
KRING1		Ring comparison flag. =0 the two ring-containing ligands are not equal. =1 they are equal.	/EQUALR	/( )	EQUALR	O	INTE	KRING1	
KRING2		Preliminary ring comparison tests flag =0 initial comparison tests of two rings negative. =1 initial comparison tests of two rings positive.	/EQUALR	/( )	EQUALR	I	INTE	KRING2	
KRING2		Preliminary ring comparison tests flag. =0 initial comparison tests of two rings negative. =1 initial comparison tests of two rings positive.	/FIRSTR	/( )	FIRSTR	O	INTE	KRING2	
KRLIG	(I)	Contains I.d. number of non-core ligand of ring location I.	/BLK4	/(2781)	SYMRNG	M	INTE	KRLIG	(6)
KRO·	(I)	Indicates whether ligand I has (2) or does not have (1) twofold rotational symmetry about plane of ring.	/BLK4	/(4262)	SYMRNG	M	INTE	KRO1	(6)
KRPROP	(I,J)	Multi-variable array I=1 number of oxygen atoms in ring K. I=2 ring location of first oxygen atom. I=3 number of nitrogen atoms in ring K. I=4 ring location of first nitrogen atom. I=5 number of double bonds in ring K. I=6 ring location of first double bond.	/BLK4	/(3122)	CRINGS CYCORR FIRSTR NRINGS	I M I I	INTE INTE INTE INTE	KRPROP(6,40) KRPROP(6,40) KRPROP(6,40) KRPROP(6,40)	
KS		Transfer variable.	/CISCOR	/( )	CISCOR	M	INTE	KS	
KS		Group number of ring ligand, also a subscript.	/EQUALR	/( )	EQUALR	M	INTE	KS	
KS		Entry value of IGS.	/INTR				INTE	KS	

17 APR 73 8.32-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
KS		Row coordinate of core atom JC.	/SHIFT /	(*)	SHIFT	W	INTE	KS	
KS		Identifier stored in array KARO or variable counter.	/SYMRNG/	(*)	SYMRNG	W	INTE	KS	
KSAME	(I,J,K)	The array location corresponding to ligand I of type J in set K is set equal to one if ligand I is equal to another ligand.	/BLK3	(5614)	EQUAL	I	INTE	KSAME	(4,3,2)
					SETUP	O	INTE	KSAME	(4,3,2)
KSAME		Denotes the number of chains of maximum length in the molecule.	/MAXCHN/	(*)	MAXCHN	W	INTE	KSAME	
KSET	(I)	Contains the number of identical ring ligands in set I.	/BLK3	(6372)	SYMRNG	M	INTE	KSET	(3)
SIX	(I)	Contains the ring numbers of the six-membered rings with two or more double bonds in their structure.	/BLK3	(5781)	CYCORR	O	INTE	KSIX	(40)
					HEXGON	I	INTE	KSIX	(40)
KSTORE	(I)	Contains identification number of rings which have some but not all of their atoms in common with the reference ring.	/BLK3	(3403)	LESSEN	M	INTE	KSTORE	(40)
KSUB		Number of ring pairs.	/FUSION/	(*)	FUSION	W	INTE	KSUB	
KSUB		Total number of ligands of a particular atom processed to this point.	/SCANBR/	(*)	SCANBR	W	INTE	KSUB	
KSUB		Storage location in array LIGAND for the composition data of a particular ligand.	/SYMTRY/	(*)	SYMTRY	W	INTE	KSUB	
KSUM		Transfer variable.	/CISCOR/	(*)	CISCOR	W	INTE	KSUM	
KSUM		Total number of second order interactions and ring corrections.	/CORCIG/	(*)	CORCIG	W	INTE	KSUM	
KSUM		Sum total of various ring properties.	/CRINGS/	(*)	CRINGS	W	INTE	KSUM	
KSUM		Denotes presence of allene structure when equal to 2.	/DELTA1/	(*)	DELTA1	W	INTE	KSUM	
KSYM	(I)	Contain internal rotational symmetry of ligand I.	/BLK1	(3381)	EXTROT	M	INTE	KSYM	(2)
KSYM		Denotes whether ligand is (1) or is not (0) linear.	/CTWO /	(*)	CTWO	I	INTE	KSYM	
KSYM		Denotes the rotational symmetry of the ligand.	/EQUALR/	(*)	EQUALR	I	INTE	KSYM	
KSYM		Denotes the rotational symmetry of the ligand.	/LINEAR/	(*)	LINEAR	O	INTE	KSYM	

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
KSYM		Description not input	/SYMRNG/	(*)	SYMRNG	I	INTE	KSYM	
KT		Group number of carbon atom adjacent to nitrogen atom in ring K.	/CYCORR/	(*)	CYCORR	M	INTE	KT	
KT		Identification number of group whose first-order contributions to the group weight are to be calculated.	/DELTA1/	(*)	DELTA1	I	INTE	KT	
KT		Identification number of group whose second-order multiple-bond weight contributions are to be calculated.	/DELTA2/	(*)	DELTA2	I	INTE	KT	
KT		Subscript.	/HEXGON/	(*)	HEXGON	M	INTE	KT	
KT		Group number. Also used to refer to core atom of group.	/SCAN /	(*)	SCAN	I	INTE	KT	
KTBENZ	(I)	Ring numbers of benzene- and pyridine-type rings.	/BLK1 /	(1365)	CTWO	I	INTE	KTBENZ	(40)
					HEXGON	M	INTE	KTBENZ	(40)
KTBL		Counter that specifies whether ligand bonded to first central core K1 is a tertiary carbon atom.	/CISCOR/	(*)	CISCOR	M	INTE	KTBL	
KTBLL		Flag set equal to 1 when KTBL is incremented. Serves to indicate when tertiary carbon groups are cis to each other.	/CISCOR/	(*)	CISCOR	M	INTE	KTBLL	
KTBR		Counter that specifies whether ligand bonded to second central core atom K2 is a tertiary carbon atom.	/CISCOR/	(*)	CISCOR	M	INTE	KTBR	
KTEMP	(I)	Used to distinguish between cis and trans ring configurations.	/BLK3 /	(5693)	CYCORR	M	INTE	KTEMP	(2)
KTEST		Ring pair detection flag. =0 ring pair was not detected by test cycle. >0 ring pair(s) detected by test cycle.	/FUSION/	(*)	FUSION	M	INTE	KTEST	
KTETRA		Counter that specifies whether two groups cis to each other are both tertiary butyl groups, that is, carbon atoms bonded to four heavy atoms.	/CISCOR/	(*)	CISCOR	M	INTE	KTETRA	
KTGAUS	(I)	Denotes the type of the second-order interaction I.	/BLK3 /	(4151)	CISCOR	O	INTE	KTGAUS	(150)
					CORCIS	M	INTE	KTGAUS	(150)
					CRINGS	O	INTE	KTGAUS	(150)
					CTWO	O	INTE	KTGAUS	(150)
					GITERE	O	INTE	KTGAUS	(150)
					GAUCHE	O	INTE	KTGAUS	(150)
					HEXGON	O	INTE	KTGAUS	(150)
					INTROT	O	INTE	KTGAUS	(150)
					NRINGS	O	INTE	KTGAUS	(150)

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
KTOT	(I)	Number of atoms of type I in ligand.	/BLK3	/(6362)	SCANBR	M	INTE	KTOT	(5)
					SCANCH	I	INTE	KTOT	(5)
					SYMTRY	M	INTE	KTOT	(5)
KTT		Number of atom pair entries in array KCCR.	/DELETE/	/( )	DELETE	M	INTE	KTT	
KTT		Number of atom pair entries in array KCCR.	/EQUALR/	/( )	EQUALR	M	INTE	KTT	
KV		Location of data in group JT associated with ligand KT.	/DELTA2/	/( )	DELTA2	M	INTE	KV	
KV		Row coordinate of current character.	/IDENT /	/( )	IDENT	M	INTE	KV	
KVALUE		Identification weight of data for which a match is sought in group additivity arrays.	/SEARCH/	/( )	SEARCH	I	INTE	KVALUE	
KW		Identification number of the ring.	/SYMRNG/	/( )	SYMRNG	M	INTE	KW	
KX		Row coordinate of initial bond location and of non-bond symbol.	/BOND /	/( )	BOND	M	INTE	KX	
KX		Subscript.	/EQUAL /	/( )	EQUAL	M	INTE	KX	
KX		Row coordinate of indexed atom number I ligand.	/FIND /	/( )	FIND	I	INTE	KX	
KX		Test row coordinate.	/IDENT /	/( )	IDENT	M	INTE	KX	
KX		Subscript.	/LESSEN/	/( )	LESSEN	M	INTE	KX	
KX		Row coordinate of first location along NX in array GRID.	/NUMBER/	/( )	NUMBER	M	INTE	KX	
KX		Group number of previously identified atom.	/SAME /	/( )	SAME	M	INTE	KX	
KX		Row coordinate of first location along NX in array GRID.	/SCAN /	/( )	SCAN	M	INTE	KX	
KXX		Row coordinate of ligand one.	/NUMBER/	/( )	NUMBER	O	INTE	KXX	
KXX		Row coordinate of first atom bonded to ligand.	/SCAN /	/( )	SCAN	/	INTE	KXX	
KX2		Row coordinate of second character.	/IDENT /	/( )	IDENT	M	INTE	KX2	
KY		Subscript that defines bond direction between atoms K1 and K2.	/CIS /	/( )	CIS	M	INTE	KY	

17 APR 73 4.08.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE SUBR	SUBROUTINE U	SUBROUTINE TYPE	USAGE VAR	USAGE DIM
KV		Number of different types of identical atoms in ligand sets A and B.	/EQUAL /(	*	) EQUAL	I	INTE	KV	
KV		Number of rings if fused ring set A.	/FIRSTR/(	*	) FIRSTR	W	INTE	KV	
KV		Number of rings in a particular fused ring set.	/FUSION/(	*	) FUSION	W	INTE	KV	
KV		Number of rings in a particular fused ring set.	/HEXGON/(	*	) HEXGON	W	INTE	KV	
KV		Number of rings in a particular fused ring set.	/NRINGS/(	*	) NRINGS	W	INTE	KV	
KV		Subscript that defines the identity type number of a ligand the first time said ligand is processed.	/SETUP /(	*	) SETUP	W	INTE	KV	
KZ		Group number of core atom whose cis ligands are to be identified.	/CIS /(	*	) CIS	W	INTE	KZ	
KZ		Row coordinate of first location beyond KX,LX along the radial direction.	/IDENT /(	*	) IDENT	W	INTE	KZ	
KZERO	(1)	Contains locations in ring which have unique ligands.	/BLK9 /(	43(9)	) SYMRNG	M	INTE	KZERO	(6)
KZZ		Identification number of core atom.	/CIS /(	*	) CIS	W	INTE	KZZ	
K1		Group number of first central test core atom.	/CIS /(	*	) CIS	I	INTE	K1	
K1		Group number of first central test core atom.	/CISCOR/(	*	) CISCOR	I	INTE	K1	
K1		Group number of first central test core atom.	/CORCIG/(	*	) CORCIG	W	INTE	K1	
K1		Identification number of ring one.	/CRINGS/(	*	) CRINGS	W	INTE	K1	
K1		Group number of ring component.	/CTWO /(	*	) CTWO	W	INTE	K1	
K1		Group number of ring component.	/CYCORR/(	*	) CYCORR	W	INTE	K1	
K1		Subscript of routine arrays.	/DELETE/(	*	) DELETE	W	INTE	K1	
K1		Group number of one of two ligand atoms currently under comparison.	/EQUAL /(	*	) EQUAL	W	INTE	K1	
K1		Group number of entry atom in ring A.	/EQUALR/(	*	) EQUALR	I	INTE	K1	

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
K1		Group number of entry atom in ring A.	/FIRSTR/(	*	> FIRSTR	/	INTE	K1	
K1		Group number of first central test core atom.	/GAUCHE/(	*	> GAUCHE	I	INTE	K1	
K1		Group number of first of two central core atoms bonded to an ortho ligand.	/HEXGON/(	*	> HEXGON	M	INTE	K1	
K1		Group number of atom previous to or parent to the atom tested.	/LINEAR/(	*	> LINEAR	M	INTE	K1	
K1		Group number of atom common to 3 fused rings.	/NRINGS/(	*	> NRINGS	M	INTE	K1	
K1		Location of element in IB whose weight is to be compared with that of element in location K2.	/ORDER /(	*	> ORDER	M	INTE	K1	
K1		Group number of ring component.	/OXYATM/(	*	> OXYATM	M	INTE	K1	
K1		Subscript and counter limit.	/RING /(	*	> RING	M	INTE	K1	
K1		Group number of atom 1 in identical ligand 1.	/SAME /(	*	> SAME	I	INTE	K1	
K1		Group number of ligand core atom bonded to atom KC1.	/SETUP /(	*	> SETUP	M	INTE	K1	
K1		Group number of first central test core atom.	/SORNGE/(	*	> SORNGE	M	INTE	K1	
K1A		Group number of one of two ring atoms currently under comparison.	/EQUALR/(	*	> EQUALR	M	INTE	K1A	
K1RD		Group number of atom in ring branch set where the deletion stops.	/DELETE/(	*	> DELETE	I	INTE	K1RD	
K11		Ordinal number assigned ligand 1 for identification purposes.	/SAME /(	*	> SAME	I	INTE	K11	
K2		Group number of second central test core atom.	/CIS /(	*	> CIS	I	INTE	K2	
K2		Group number of second central test core atom.	/CISCOR/(	*	> CISCOR	I	INTE	K2	
K2		Group number of second central test core atom.	/CORCIG/(	*	> CORCIG	M	INTE	K2	
K2		Identification number of ring two.	/ERINGS/(	*	> ERINGS	M	INTE	K2	
K2		Group number of ring component.	/CYCOMR/(	*	> CYCOMR	M	INTE	K2	
K2		Subscript of routine arrays.	/DELETE/(	*	> DELETE	M	INTE	K2	

17 APR 73 8.32.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
K2		Group number of one of two ligand atoms currently under comparison.	/EQUAL /(*		) EQUAL	W	INTE	K2	
K2		Group number of entry atom in ring B.	/EQUALR/(*		) EQUALR	I	INTE	K2	
K2		Group number of entry atom in ring B.	/FIRSTR/(*		) FIRSTR	I	INTE	K2	
K2		Group number of second central test core atom.	/GAUCHE/(*		) GAUCHE	I	INTE	K2	
K2		Group number of second of two central core atoms bonded to an ortho ligand.	/HEXGON/(*		) HEXGON	W	INTE	K2	
K2		Subscript.	/LESSEN/(*		) LESSEN	W	INTE	K2	
K2		Group number of the atom tested.	/LINEAR/(*		) LINEAR	W	INTE	K2	
K2		Group number of atom common to 3 fused rings.	/NRINGS/(*		) NRINGS	W	INTE	K2	
K2		Location of element in I5 whose weight is to be compared with that of element in location K1.	/ORDER /(*		) ORDER	W	INTE	K2	
K2		Group number of atom 1 in identical ligand 2.	/SAME /(*		) SAME	I	INTE	K2	
K2		Group number of ligand core atom bonded to atom KC2.	/SETUP /(*		) SETUP	W	INTE	K2	
K2		Group number of second central test core atom.	/SORNGI/(*		) SORNGI	W	INTE	K2	
K2R		Group number of one of two ring atoms currently under comparison.	/EQUALR/(*		) EQUALR	W	INTE	K2R	
K2SET		Storage location variable for KNTR.	/EQUALR/(*		) EQUALR	W	INTE	K2SET	
K2Z		Ordinal number assigned ligand 2 for identification purposes.	/SAME /(*		) SAME	I	INTE	K2Z	
K3		Group atom of nitrogen ring atom.	/NRINGS/(*		) NRINGS	W	INTE	K3	
K3		Lower element of array IA used in order check.	/ORDER /(*		) ORDER	W	INTE	K3	
K3		Identification number of core atom.	/SORNGI/(*		) SORNGI	W	INTE	K3	
K4		Upper element of array IA used in order check.	/ORDER /(*		) ORDER	W	INTE	K4	
L		Column coordinate of indexed core atom.	/ASSIGN/(*		) ASSIGN	I	INTE	L	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
L		Loop counter and subscript.	/CHAINM/(	*)	CHAINM	M	INTE	L	
L		Loop counter and subscript.	/CHANGE/(	*)	CHANGE	M	INTE	L	
L		Loop counter and subscript.	/CISCOR/(	*)	CISCOR	M	INTE	L	
L		Loop counter and subscript.	/CORCIG/(	*)	CORCIG	M	INTE	L	
L		Loop counter and subscript.	/CYCORR/(	*)	CYCORR	M	INTE	L	
L		Loop counter and subscript.	/DATA1 /(	*)	DATA1	M	INTE	L	
L		Loop counter and subscript	/DITERE/(	*)	DITERE	M	INTE	L	
L		Loop counter and subscript.	/EXTROT/(	*)	EXTROT	M	INTE	L	
L		Entry value: column coordinate of indexed atom. Exit value: column coordinate of core atom.	/FIND /(	*)	FIND	M	INTE	L	
L		Loop counter and subscript.	/FUSION/(	*)	FUSION	M	INTE	L	
L		Loop counter and subscript.	/GADATA/(	*)	GADATA	M	INTE	L	
L		Loop counter and subscript.	/GAUCHE/(	*)	GAUCHE	M	INTE	L	
L		Loop counter and subscript.	/HEXGON/(	*)	HEXGON	M	INTE	L	
L		Column coordinate of first character of symbol.	/IDENT /(	*)	IDENT	M	INTE	L	
L		Loop counter and subscript.	/INTROT/(	*)	INTROT	M	INTE	L	
L		Loop counter and subscript.	/LESSEN/(	*)	LESSEN	M	INTE	L	
L		Loop counter and subscript.	/MAICHN/(	*)	MAICHN	M	INTE	L	
L		Loop counter and subscript.	/MULTI /(	*)	MULTI	M	INTE	L	
L		Column coordine of indexed core atom.	/NUMBER/(	*)	NUMBER	:	INTE	L	
L		Subscript of symbol array (SYMBOL).	/PRINT1/(	*)	PRINT1	M	INTE	L	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	U	TYPE	WAR	DIM
L		Loop counter and subscript.	/RESETR/(	)	RESETR	W	INTE	L	
L		Column coordinate of core atom in array GRID.	/SCAN /(	)	SCAN	I	INTE	L	
L		Loop counter and subscript.	/SETUP /(	)	SETUP	W	INTE	L	
L		Loop counter and subscript.	/SHIFT /(	)	SHIFT	W	INTE	L	
L		Loop counter and subscript.	/SORNGI/(	)	SORNGI	W	INTE	L	
L		Column gird coordinate of core atom.	/STAND /(	)	STAND	W	INTE	L	
L		Description not input	/SYMRNG/(	)	SYMRNG	W	INTE	L	
L		Loop counter and subscript.	/SYMTRY/(	)	SYMTRY	W	INTE	L	
L		Loop counter and subscript.	/TGAP /(	)	TGAP	W	INTE	L	
LABEL	( )	Hollerith symbol for blank. Input array containing identification label for molecule (optional).	/BLK4 /(288)	)	TGAP	R	INTE	LABEL (50)	
LD		Assigned change in column coordinate of symbol input array.	/ASSIGN/(	)	ASSIGN	O	INTE	LD	
LD		Change in column coordinate	/BOND /(	)	BOND	I	INTE	LD	
LD		Change in row coordinate for indexed atom number 1 ligand.	/FIND /(	)	FIND	I	INTE	LD	
LD		Change in column coordinate.	/IDENT /(	)	IDENT	W	INTE	LD	
LD		Change in column coordinate of array GRID.	/NUMBER/(	)	NUMBER	I	INTE	LD	
LD		Change in column coordinate for transformation to NX.	/SCAN /(	)	SCAN	I	INTE	LD	
LDD		Change in column coordinate of ligand one.	/NUMBER/(	)	NUMBER	O	INTE	LDD	
LDONE	( )	Indicator array I=0 composition of all ligands of a core atom not yet found. I=1 all compositions have been found.	/BLK3 /(3202)	)	SYMTRY	W	INTE	LDONE (100)	

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
LDZ		LD value of second character.	/IDENT /(	*)	IDENT	W	INTE	LDZ	
LESSEN		Examines the uniqueness of the ring structures identified by the program. Any rings that are equal to other rings or composed of smaller rings are discarded.	/LESSEN/(	*)	CHAINM LESSEN RING	S E S		LESSEN LESSEN LESSEN	
LF		Upper limit of loop counter.	/DITERE/(	*)	DITERE	W	INTE	LF	
LF		Upper limit of loop counter that controls the selection of a test ligand from set A.	/EQUAL /(	*)	EQUAL	I	INTE	LF	
LF		Upper limit of loop counter.	/GAUCHE/(	*)	GAUCHE	W	INTE	LF	
LF		Upper limit of loop counter.	/LESSEN/(	*)	LESSEN	W	INTE	LF	
LF		Upper limit of loop counter.	/NEWCOL/(	*)	NEWCOL	W	INTE	LF	
LF		Upper limit of loop counter.	/RESETR/(	*)	RESETR	W	INTE	LF	
LF		Upper limit of loop counter.	/SCANBR/(	*)	SCANBR	W	INTE	LF	
LF		Upper limit of loop counter.	/SCANCH/(	*)	SCANCH	W	INTE	LF	
LF		Upper limit of loop counter that controls the selection of a test ligand from set A.	/SETUP /(	*)	SETUP	I	INTE	LF	
LF		Upper limit of loop counter.	/SORNGI/(	*)	SORNGI	W	INTE	LF	
LFLAGG		Data flag =0 no group data are missing from data library. =1 one or more group data are missing from data library.	/BLK5 /(	62)	IGAP	M	INTE	LFLAGG	
LFLAGS		Data flag =0 symmetry and optical contributions are computable. =1 symmetry and/or optical contributions noncomputable and data for their computation not input.	/BLK5 /(	61)	ENTSYM IGAP	O I	INTE INTE	LFLAGS LFLAGS	
LGI		Subscript of array PERTKC that assigns the group number of a ligand core atom bonded to atom KCI.	/EQUAL /(	*)	EQUAL	W	INTE	LGI	

17 APR 73 P 02:36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
LG1		Subscript of array PERTKC that assigns the group number of a ligand core atom bonded to atom KC1.	/SETUP /(	*)	SETUP	W	INTE	LG1	
LG2		Subscript of array PERTKC that assigns the group number of a ligand core atom bonded to atom KC2.	/EQUAL /(	*)	EQUAL	W	INTE	LG2	
LG2		Subscript of array PERTKC that assigns the group number of a ligand core atom bonded to atom KC2.	/SETUP /(	*)	SETUP	W	INTE	LG2	
LI		Lower limit of loop counter.	/DITERE/(	*)	DITERE	W	INTE	LI	
LIGAND	(I,J,K)	Contains atomic composition I of ligand J of core atom K.	/BLK3 /(	1202)	SCANBR SCANCH SETUP SYMTRY	M M I M	INTE INTE INTE INTE	LIGAND(5,4,100) LIGAND(5,4,100) LIGAND(5,4,100) LIGAND(5,4,100)	
LIM		Number of elements in array IA or IB.	/ORDER /(	*)	ORDER	I	INTE	LIM	
LIM		Dimension of array to be searched.	/SEARCH/(	*)	SEARCH	I	INTE	LIM	
LIMIT		Chain location occupied by atom at the start of the branch test cycle.	/SYMTRY/(	*)	SYMTRY	W	INTE	LIMIT	
LINE	(I)	Indicates whether ligand I is (1) or is not (0) linear.	/BLK1 /(	3879)	EXTROT	M	INTE	LINE	(2)
LINE		Denotes the rotational symmetry of the ligand.	/CTWO /(	*)	CTWO	I	INTE	LINE	
LINE		Denotes whether ligand is (1) or is not (0) linear.	/EQUALR/(	*)	EQUALR	I	INTE	LINE	
LINE		Denotes whether ligand is (1) or is not (0) linear.	/LINEAR/(	*)	LINEAR	O	INTE	LINE	
LINE		Description not input	/SYMRNG/(	*)	SYMRNG	I	INTE	LINE	
LINEAR		Establishes the linearity and rotational symmetry of a particular ligand.	/LINEAR/(	8)	CTWO EQUALR EXTROT LINEAR SYMRNG	S S S E S		LINEAR LINEAR LINEAR LINEAR LINEAR	
LK		Upper limit of counter KK of ring test cycle.	/CHAINM/(	*)	CHAINM	W	INTE	LK	
LL		Loop counter and subscript.	/CORCIG/(	*)	CORCIG	W	INTE	LL	
LL		Subscript.	/GAUCHE/(	)	GAUCHE	W	INTE	LL	

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE SUBR	U	TYPE	USAGE VAR	DIM
LL		Loop counter and subscript.	/INTROT/(	)	INTROT	W	INTE	LL	
LL		Loop counter and subscript.	/NEWCOL/(	)	NEWCOL	W	INTE	LL	
LL		Loop counter and subscript.	/RING /(	)	RING	W	INTE	LL	
LL		Loop counter and subscript.	/SHIFT /(	)	SHIFT	W	INTE	LL	
LL		Loop counter and subscript.	/SORNGI/(	)	SORNGI	W	INTE	LL	
LL		Lower limit of cycle counter.	/STAND /(	)	STAND	W	INTE	LL	
LL		Subscript.	/TGAP /(	)	TGAP	W	INTE	LL	
LM		Number of atoms or links in chain.	/CHAINM/(	)	CHAINM	W	INTE	LM	
LM		Location J in NBC(I,J) of uppermost element in portion of new chain defined in subroutine NEWCOL.	/NEWCOL/(	)	NEWCOL	M	INTE	LM	
LM		Number of atoms in chain LX.	/RING /(	)	RING	I	INTE	LM	
LM		Number of components of a particular chain.	/SYMTRY/(	)	SYMTRY	W	INTE	LM	
LMA		Chain location of atom below atom LMM.	/SYMTRY/(	)	SYMTRY	W	INTE	LMA	
LMM		Location in chain LXX of core atom under inspection.	/SYMTRY/(	)	SYMTRY	W	INTE	LMM	
LN		Chain location of next atom down the chain.	/SYMTRY/(	)	SYMTRY	W	INTE	LN	
LOCCO	(I)	Contains ring locations of CO atoms.	/BLK3 /(	3441)	CYCORR	O	INTE	LOCCO	(30)
LOCN	(I)	Contains ring locations of nitrogen atoms.	/BLK3 /(	3471)	CYCORR	M	INTE	LOCN	(30)
LOCO	(I)	Contains ring locations of oxygen atoms.	/BLK3 /(	3501)	CYCORR	M	INTE	LOCO	(30)
LOCX		Ring atom which may be adjacent to one or two CO groups.	/OXYATM/(	)	OXYATM	I	INTE	LOCX	
LS		Lower limit of loop counter.	/SCANBR/(	)	SCANBR	W	INTE	LS	
LS		Lower limit of loop counter.	/SCANCH/(	)	SCANCH	W	INTE	LS	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE SUBR	SUBROUTINE U	USAGE TYPE	USAGE VAR	USAGE DIM
LS		Column coordinate of core atom JC.	/SHIFT /	(*)	SHIFT	M	INTE	LS	
LS		Start location in KGRID into which column of data from array GRID is to be transferred.	/TGAP /	(*)	TGAP	M	INTE	LS	
LV		Column coordinate of current character.	/IDENT /	(*)	IDENT	M	INTE	LV	
LX		Column coordinate of initial bond location and of non-bond symbol.	/BOND /	(*)	BOND	M	INTE	LX	
LX		Computed number of chains in molecule.	/CHAINM/	(*)	CHAINM	M	INTE	LX	
LX		Computed number of chains in molecule.	/CHANGE/	(*)	CHANGE	I	INTE	LX	
LX		Computed number of chains in molecule.	/CORCIG/	(*)	CORCIG	I	INTE	LX	
LY		Column coordinate of indexed atom number 1 ligand.	/FIND /	(*)	FIND	I	INTE	LX	
LX		Test column coordinate.	/IDENT /	(*)	IDENT	M	INTE	LX	
LX		Subscript.	/LESSEN/	(*)	LESSEN	M	INTE	LX	
LX		Computed number of chains in molecule.	/MAXCHN/	(*)	MAXCHN	I	INTE	LX	
LX		Computed number of chains in molecule.	/NEWCOL/	(*)	NEWCOL	M	INTE	LX	
LX		Column coordinate of first location along NX in array GRID.	/NUMBER/	(*)	NUMBER	M	INTE	LX	
LX		Computed number of chains in molecule.	/RESETR/	(*)	RESETR	I	INTE	LX	
LX		Computed number of chains in molecule. Refers also to the particular chain under investigation.	/RING /	(*)	RING	I	INTE	LX	
LX		Column coordinate of first location along NX in array GRID.	/SCAN /	(*)	SCAN	M	INTE	LX	
LX		Loop counter and subscript.	/SCANBR/	(*)	SCANBR	M	INTE	LX	
LX		Loop counter and subscript.	/SCANCH/	(*)	SCANCH	M	INTE	LX	
LX		Computed number of chains in molecule.	/SYMTRY/	(*)	SYMTRY	I	INTE	LX	

17 APR 73 6.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE SUBN	U	USAGE TYPE	VAR	DIM
LX		Number of chains in molecule.	/TGAP /(*	)	TGAP	I	INTE	LX	
LXL		Uppermost chain column used in the construction of the new chain.	/CHAINM/(*	)	CHAINM	M	INTE	LXL	
LXL		Last computed chain to be used in the calculation.	/NEWCOL/(*	)	NEWCOL	I	INTE	LXL	
LXX		Column coordinate of ligand one.	/NUMBER/(*	)	NUMBER	O	INTE	LXX	
LXX		Number of chain that identifies ring K.	/RESETR/(*	)	RESETR	M	INTE	LXX	
LXX		Number of chain that contains ring LL.	/RING /(*	)	RING	M	INTE	LXX	
LXX		Column coordinate of first atom bonded to ligand.	/SCAN /(*	)	SCAN	I	INTE	LXX	
LXX		Number of the chain under inspection.	/SYMTRY/(*	)	SYMTRY	M	INTE	LXX	
LX2		Column coordinate of second character.	/IDENT /(*	)	IDENT	M	INTE	LX2	
LY		Chain number of longest chain computed thus far by routine MAXCHN.	/MAXCHN/(*	)	MAXCHN	M	INTE	LY	
LY		Used to denote location of KCPV element in chain as well as the new chain number.	/NEWCOL/(*	)	NEWCOL	U	INTE	LY	
LYP		Chain number of longest chain.	/MAXCHN/(*	)	MAXCHN	M	INTE	LYP	
LZ		Column coordinate of first location beyond KX,LX along the radial direction.	/IDENT /(*	)	IDENT	M	INTE	LZ	
L1		Group number of core atom bonded to first central atom and cis to L2.	/CISCOR/(*	)	CISCOR	M	INTE	L1	
L1		Location of data (subscript of RINGD) for the ring corrections of ring K.	/CORCIG/(*	)	CORCIG	M	INTE	L1	
L1		Location of ring correction data (subscript of RINGD) for ring K.	/CYCORR/(*	)	CYCORR	O	INTE	L1	
L2		Group number of core atom bonded to second central atom and cis to L1.	/CISCOR/(*	)	CISCOR	M	INTE	L2	
M		Amount by which bond direction is changed.	/CIS /(*	)	CIS	M	INTE	M	
M		Loop counter and subscript.	/CYCORR/(*	)	CYCORR	M	INTE	M	

17 APR 73 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	SUBR U	SUBR TYPE	USAGE VAR	USAGE DIM
A		Code symbol of indexed atom.	/FIND	/(*)	FIND	I	INTE	A	
A		Subscript of SYMBOL array used for identification of input symbol.	/IDENT	/(*)	IDENT	A	INTE	A	
A		Loop counter and subscript.	/INTROT	/(*)	INTROT	W	INTE	A	
A		Loop counter and subscript.	/LESSEN	/(*)	LESSEN	W	INTE	A	
A		Loop counter and subscript.	/MAXCHN	/(*)	MAXCHN	W	INTE	A	
A		Subscript of program library symbol that corresponds to input symbol.	/MULTI	/(*)	MULTI	D	INTE	A	
A		Loop counter and subscript.	/OXYATM	/(*)	OXYATM	W	INTE	A	
A		Loop counter and subscript.	/PRINTI	/(*)	PRINTI	W	INTE	A	
A		Loop counter and subscript.	/RESETR	/(*)	RESETR	W	INTE	A	
A		Identifies type of ligand atom.	/SCAN	/(*)	SCAN	I	INTE	A	
A		Loop counter and subscript.	/SETUP	/(*)	SETUP	W	INTE	A	
A		Counter.	/SORNGI	/(*)	SORNGI	W	INTE	A	
A		Subscript of array SYMBOL used to identify a chemical symbol.	/STAND	/(*)	STAND	W	INTE	A	
A		Subscript of array SYMBOL which designates a particular chemical symbol.	/SUMATM	/(*)	SUMATM	I	INTE	A	
AA	(I)	Contains location of CO group in ring K.	/BLK3	/(3531)	OXYATM	A	INTE	AA	(30)
AARO	(I)	Aromatic ring flag. I=1 core atom I is an aromatic benzene-type ring component. I=2 core atom I is an aromatic pyridine-type ring component. I=0 core atom I is not an aromatic ring component.	/BLK1	/(1225)	CORCIC GAUCHE HEXGON	O I O	INTE INTE INTE	AARO AARO AARO	(100) (100) (100)
MATRIX	(I,J)	Contains ring locations (I) of all ligands belonging to set J of identical ligands.	/BLK4	/(4290)	SYMRNG	A	INTE	MATRIX	(6,3)

17 Jan 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
MAI		Description not input	/SYMRNG/	(*	) SYMRNG	W	INTE	MAI	
MAI		Location of last data group contained in Bensons tables.	/TGAP /(	*	) TGAP	W	INTE	MAI	
MAXCHN		Finds the longest chain in the molecule and prints out the group numbers of the core atom constituents of the chain. If asymmetric atoms are present, it also determines if another chain of the same (maximum) length is present which contains more asymmetric atoms than the first maximum chain detected. If so, this chain becomes the new maximum chain.	/MAXCHN/	(	) CORCIB MAXCHN	S E		MAXCHN MAXCHN	
MAXD		Maximum value of MDIF.	/CHANGE/	(*	) CHANGE	W	INTE	MAXD	
MAXIZ		Location of branch atom in chain MAXL.	/CHANGE/	(*	) CHANGE	W	INTE	MAXIZ	
MAXL		Chain number corresponding to MAXD.	/CHANGE/	(*	) CHANGE	W	INTE	MAXI	
MAXT		Upper limit of loop counter.	/SYMRNG/	(*	) SYMRNG	W	INTE	MAXT	
MBC	(I)	Contains group numbers of core atom constituents of longest chain.	/BLK5 /(	7)	EXTROT MAXCHN	I M	INTE INTE	MBC MBC	(50) (50)
MBL		Left-most location of data in column J of array GRID or in array LABEL.	/TGAP /(	*	) TGAP	W	INTE	MBL	
MBR		Right-most location of data in column J of array GRID or in array LABEL.	/TGAP /(	*	) TGAP	W	INTE	MBR	
MBS	(I)	Properties of longest chain. I=1 number of atoms in longest chain. I=2 chain number of longest chain.	/BLK5 /(	57)	EXTROT MAXCHN	I O	INTE INTE	MBS MBS	(2) (2)
MCY	(I)	Contains the ordinal ring number (KCY) of the test ring pair to one of which the non-ring core atom I is bonded.	/BLK1 /(	1525)	EQUAL EQUALR SYMTRY	I M O	INTE INTE INTE	MCY MCY MCY	(100) (100) (100)
MD		Total number of entries in array MDEL.	/BLK3 /(	5412)	CORCIB CRINGS HEXGON	M M M	INTE INTE INTE	MD MD MD	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
MDEL	(I)	Contains the ring number of the rings which form part of an aromatic fused ring system and whose ring corrections are not to be used in the calculation.	/BLK3	/(5613)	CORCIG	I	INTE	MDEL	(40)
					CRINGS	O	INTE	MDEL	(40)
					HEXSCN	O	INTE	MDEL	(40)
MDIF		Terminal group flag. =0 terminal group is present in molecule. =1 terminal group is not present.	/CHAINM/(*	)	CHAINM	W	INTE	MDIF	
MDIF		The difference I2 - I1.	/CHANGE/(*	)	CHANGE	W	INTE	MDIF	
MDIF		Terminal group flag. =0 terminal group is present in molecule. =1 terminal group is not present.	/RESETR/(*	)	RESETR	I	INTE	MDIF	
ME		Number of rings for which there are no ring corrections.	/CORCIG/(*	)	CORCIG	W	INTE	ME	
MER	(I,J)	Multi-variable array I=1,2 group numbers of entry ring atoms in two test rings. I=3 number of non-ring ligands in two test rings.	/BLK1	/(1825)	EQUAL	I	INTE	MER	(3,40)
					EQUALR	O	INTE	MER	(3,40)
					SYMTRY	O	INTE	MER	(3,40)
MESO		Input value for total number of meso structures of molecule (optional).	/CORCIG/(*	)	CORCIG	I	INTE	MESO	
MESO		Input value for total number of meso structures of molecule (optional).	/ENTSYM/(*	)	ENTSYM	I	INTE	MESO	
MESO		Input value for total number of meso structures of molecule (optional).	/TGAP /(*	)	TGAP	W	INTE	MESO	
MESOC		Computed value for the number of meso structures.	/ENTSYM/(*	)	ENTSYM	W	INTE	MESOC	
MF		Upper limit of loop counter.	/CYCORR/(*	)	CYCORR	W	INTE	MF	
MF		Number of components in reference ring.	/LESSEN/(*	)	LESSEN	W	INTE	MF	
MF		Upper limit of loop counter.	/OXYATM/(*	)	OXYATM	W	INTE	MF	
MF		Upper limit of loop counter.	/RESETR/(*	)	RESETR	W	INTE	MF	
MF		Upper limit of loop counter.	/SORNGI/(*	)	SORNGI	W	INTE	MF	

17 APR 73 0.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
MFIX	(I)	Designates the weight correction to be applied to the non-ring core atoms bonded to the ring.	/BLK3	/(3801)	HEXGON	M	INTE	MFIX	(100)
MFLAG		Flag. If non zero, the elements of IB corresponding to JC are reset. Otherwise they are not.	/SHIFT	/( )	SHIFT	I	INTE	MFLAG	
MID2		Number of atoms contained in longest chain divided by 2.	/EXTROT	/( )	EXTROT	M	INTE	MID2	
MID2P		Atom position in longest chain equal to MID2 plus one.	/EXTROT	/( )	EXTROT	M	INTE	MID2P	
MJ		Group number of ring component.	/CYCORR	/( )	CYCORR	M	INTE	MJ	
MJ		Value of counter J.	/IDENT	/( )	IDENT	M	INTE	MJ	
MKJF		Upper limit of loop counter.	/CYCORR	/( )	CYCORR	M	INTE	MKJF	
MKT	(I)	Contains group numbers for cis ligand pairs. If the ligand is not a core atom, the group number is, of course, 0. Ligands I=1 and I=3 are cis to each other as well as ligands I=2 and I=4.	/BLK3	/(5606)	CIS CISCOR CYCORR EXTROT	O I I M	INTE INTE INTE INTE	MKT MKT MKT MKT	(4) (4) (4) (4)
MKT	(I)	Ring scan test indicator array. I=0 ring I has not been scanned in reverse direction. I=1 yes it has.	/BLK3	/(5735)	EQUAL EQUALR SYMTRY	I M O	INTE INTE INTE	MKT MKT MKT	(40) (40) (40)
MM		Loop counter and subscript.	/EQUAL	/( )	EQUAL	M	INTE	MM	
MM		Loop counter and subscript.	/LESSEN	/( )	LESSEN	M	INTE	MM	
MM		Value of j index in IX(I,j,k) where the search for the core atom data commences.	/SCAN	/( )	SCAN	I	INTE	MM	
MM		Value of j index in IX(I,j,k) where the search for the core atom data commences.	/STAND	/( )	STAND	M	INTE	MM	
MMF		Upper limit of loop counter.	/LESSEN	/( )	LESSEN	M	INTE	MMF	
MN		Number of atom pairs stored in array MNIX.	/CORCIG	/( )	CORCIG	M	INTE	MN	
MN		Number of atom pairs stored in array MNIX.	/SORNGI	/( )	SORNGI	M	INTE	MN	

17 APR 73 0.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
MNC	(I,J)	Contains value of NC (J=1) and group number KC (J=2) of branch core atoms. The exception occurs with core atom one which is also included when it has only two core atom ligands.	/BLK9	/(4)	STAND	M	INTE	MNC	(100,2)
MNIX	(I)	Indicator array that specifies whether core atom I has been processed.no if 0, yes if 1.	/BLK1	/(28)	CORCIG	M	INTE	MNIX	(2,200)
					SORNGI	M	INTE	MNIX	(2,200)
MOLWT	(I)	Computed weight of each group in molecule.	/BLK2	/(19)	DELTA1	M	INTE	MOLWT	(100)
					DELTA2	M	INTE	MOLWT	(100)
					HEXGON	M	INTE	MOLWT	(100)
					PRINT1	I	INTE	MOLWT	(100)
					STAND	M	INTE	MOLWT	(100)
					TGAP	I	INTE	MOLWT	(100)
MP		Number of ligands attached to indexed atom.	/FIND	/( )	FIND	I	INTE	MP	
MP		Number of ligands of indexed atom computed in subroutine NUMBER.	/NUMBER	/( )	NUMBER	M	INTE	MP	
MP		Number of atoms bonded to ligand as computed in NUMBER.	/SCAN	/( )	SCAN	I	INTE	MP	
MPOS		Bond-type flag =0 ring does not have 3 double bonds. =1 ring has 3 double bonds.	/HEXGON	/( )	HEXGON	M	INTE	MPOS	
MR		Column coordinate of IX where data of next ligand in group KC are to be stored.	/FIND	/( )	FIND	O	INTE	MR	
MR		Value of J index in IX(I,J,K) where the ligand data are stored.	/SCAN	/( )	SCAN	M	INTE	MR	
MR		Value of J index in IX(I,J,K) where the ligand data are stored.	/STAND	/( )	STAND	M	INTE	MR	
MRR	(I)	Contains number that denotes the identity type of the core ligand whose group number equals I.	/BLK3	/(5862)	SETUP	M	INTE	MRR	(100)
MRR		Subscript I of array MNC(I,J) equal to the number of branch atoms in the molecule.	/BLK9	/(3)	STAND	M	INTE	MRR	
MRST	(I)	Contains ordinal numbers of previously identified identical ligands of core atom KC.	/BLK9	/(4274)	SAME	M	INTE	MRST	(6)
MRS1	(I)	Contains ordinal identification numbers for ligands of atom in set A	/BLK9	/(4280)	EQUAL	I	INTE	MRS1	(9)
					SETUP	M	INTE	MRS1	(9)
MRS2	(I)	Contains ordinal identification numbers for ligands of atom in set B	/BLK9	/(4289)	EQUAL	I	INTE	MRS2	(9)
					SETUP	M	INTE	MRS2	(9)

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE SUBR	U	TYPE	USAGE VAR	DIM
MS	(I)	Contains chemical symbols subscript codes for cis ligand pairs. Ligands I=1 and I=3 are cis to each other as well as ligands I=2 and I=4.	/BLK3	/(5602)	CIS	O	INTE	MS	(4)
					CISCOR	I	INTE	MS	(4)
					EXTROT	I	INTE	MS	(4)
MS		Loop counter and subscript.	/IDENT	/( )	IDENT	W	INTE	MS	
MS		Subscript of array IB.	/SCAN	/( )	SCAN	W	INTE	MS	
MTEMP		Value of MD at start of cycle.	/HEXGON	/( )	HEXGON	W	INTE	MTEMP	
MULTI		Identifies a designated multi-character multi-word input symbol with the corresponding single word Hollerith program library symbol.	/MULTI	/( )	IDENT	S		MULTI	
					MULTI	E		MULTI	
MULT1		Factor that determines the lower limit of a particular loop counter.	/EQUAL	/( )	EQUAL	I	INTE	MULT1	
MULT1		Factor that determines the lower limit of a particular loop counter.	/SETUP	/( )	SETUP	I	INTE	MULT1	
MULT2		Factor that determines the lower limit of a particular loop counter.	/EQUAL	/( )	EQUAL	I	INTE	MULT2	
MULT2		Factor that determines the lower limit of a particular loop counter.	/SETUP	/( )	SETUP	I	INTE	MULT2	
MVW	(I)	Ring location of bonds or group numbers of carbon atoms on either side of nitrogen atom in unsaturated nitrogen-containing rings.	/BLK1	/(3765)	CYCORR	M	INTE	MVW	(2)
MWGT	(I)	Additional identification number added to original weight of an element or radical when present as the core atom of the group.	/BLK2	/(10)	STAND	I	INTE	MWGT	(9)
					TGAP	O	INTE	MWGT	(9)
MX		Subscript of array NBC.	/CHANGE	/( )	CHANGE	W	INTE	MX	
MX		Subscript that selects data of IDXR to be tested, also subscript of MCY.	/EQUALR	/( )	EQUALR	W	INTE	MX	
MX		Subscript. Also group number of atom adjacent to atom KC1.	/EXTROT	/( )	EXTROT	W	INTE	MX	
MX		Subscript of arrays NBC and IRING.	/RESETR	/( )	RESETR	W	INTE	MX	
MX		Subscript.	/STAND	/( )	STAND	W	INTE	MX	
MX		Initial value of subscript MZ.	/SUMATM	/( )	SUMATM	W	INTE	MX	
MX		Subscript.	/SYMRNG	/( )	SYMRNG	W	INTE	MX	

17 APR 73 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
MX1		Subscript of IDX governing the storage of the ligand data for K1.	/EQUAL /(	*)	EQUAL	W	INTE	MX1	
MX2		Subscript of IDX governing the storage of the ligand data for K2.	/EQUAL /(	*)	EQUAL	W	INTE	MX2	
MY		Subscript and temporary storage variable.	/CHANGE/(	*)	CHANGE	W	INTE	MY	
MY		Subscript of array NBC.	/RESETR/(	*)	RESETR	W	INTE	MY	
MY		Number of particular element contained in species M.	/SUMATM/(	*)	SUMATM	W	INTE	MY	
MZ		Subscript.	/RESETR/(	*)	RESETR	W	INTE	MZ	
MZ		Location in NUMFRQ containing data for species M.	/SUMATM/(	*)	SUMATM	W	INTE	MZ	
M1		Indicates whether or not ligand 1 has already been found equal to another ligand.	/SAME /(	*)	SAME	W	INTE	M1	
M1		Subscript.	/SYMRNG/(	*)	SYMRNG	W	INTE	M1	
M2		Indicates whether or not ligand 2 has already been found equal to another ligand.	/SAME /(	*)	SAME	W	INTE	M2	
M2		Subscript.	/SYMRNG/(	*)	SYMRNG	W	INTE	M2	
N		Location where data are stored in arrays MS and MKT.	/CIS /(	*)	CIS	W	INTE	N	
N		Loop counter and subscript.	/CTWO /(	*)	CTWO	W	INTE	N	
N		Loop counter and subscript.	/CYCORR/(	*)	CYCORR	W	INTE	N	
N		Loop counter and subscript.	/EQUAL /(	*)	EQUAL	W	INTE	N	
N		Loop counter.	/LESSEN/(	*)	LESSEN	O	INTE	N	
N		Loop counter and subscript.	/MULTI /(	*)	MULTI	W	INTE	N	
N		Loop counter and subscript.	/PRINT1/(	*)	PRINT1	W	INTE	N	
N		Loop counter and subscript.	/RESETR/(	*)	RESETR	W	INTE	N	
N		Loop counter and subscript.	/SCAN /(	*)	SCAN	W	INTE	N	

17 APR 73 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
N		Loop counter and subscript.	/SETUP /	(*	) SETUP	M	INTE	N	
N		Loop counter and subscript.	/STAND /	(*	) STAND	M	INTE	N	
N		Loop counter and subscript.	/SYMRNG/	(*	) SYMRNG	M	INTE	N	
NAL		Number of groups for which there are no themochemical group additivity values and which have been assigned the thermochemical properties of another similar group.	/TGAP /	(*	) TGAP	C	INTE	NAL	
NALT		Number of atomic constituents that form the backbone of the ring.	/CHAINM/	(*	) CHAINM	M	INTE	NALT	
NALT		Number of atomic constituents that form the backbone of the ring.	/RING /	(*	) RING	M	INTE	NALT	
NASYMC		Number of asymmetric carbon atoms in molecule.	/ASYMC /	(*	) ASYMC	M	INTE	NASYMC	
NASYMC		Number of asymmetric carbon atoms in molecule.	/CORCIG/	(*	) CORCIG	I	INTE	NASYMC	
NASYMC		Number of asymmetric atoms in molecule.	/ENTSYM/	(*	) ENTSYM	I	INTE	NASYMC	
NASYMC		Number of asymmetric carbon atoms in molecule.	/EXTROT/	(*	) EXTROT	I	INTE	NASYMC	
NASYMC		Number of asymmetric atoms in molecule.	/MAXCHN/	(*	) MAXCHN	I	INTE	NASYMC	
NBC	(I,J)	Contains group number of atom I in chain J.	/BLK9 /	(1)	CHAINM	M	INTE	NBC	(60,50)
					CHANGE	M	INTE	NBC	(60,50)
					MAXCHN	I	INTE	NBC	(60,50)
					NEWCOL	M	INTE	NBC	(60,50)
					RESETR	M	INTE	NBC	(60,50)
					RING	I	INTE	NBC	(60,50)
					SYMTRY	I	INTE	NBC	(60,50)
NBENZ		Total number of benzene type rings present in molecule.	/BLK3 /	(5610)	CORCIG	M	INTE	NBENZ	
					CYCORR	M	INTE	NBENZ	
					HEXGON	I	INTE	NBENZ	
					SYMRNG	I	INTE	NBENZ	
NBP		Number of branch points (atoms) in molecule.	/CHAINM/	(*	) CHAINM	M	INTE	NBP	
NBP		Number of branch points (atoms) in molecule.	/NEWCOL/	(*	) NEWCOL	M	INTE	NBP	
NBP		Number of branch atoms in chain LXX.	/RESETR/	(*	) RESETR	M	INTE	NBP	
NBPA		Number of branch atoms in ring.	/RESETR/	(*	) RESETR	M	INTE	NBPA	

17 APR 73 0.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
NBS	(I,J)	Data pertaining to chain I J=1 number of atoms in chain. J=2 number of branch atoms in chain.	/BLK9	/(3001)	CHAINM	O	INTE	NBS	(60,2)
					CHANGE	M	INTE	NBS	(60,2)
					MAXCHN	I	INTE	NBS	(60,2)
					NEWCOL	I	INTE	NBS	(60,2)
					RESETR	M	INTE	NBS	(60,2)
					RING	I	INTE	NBS	(60,2)
					SYMTRY	I	INTE	NBS	(60,2)
NBX	(I,J)	Location J in chain I (stored in arrayNBC) that contains a branch atom.	/BLK9	/(3121)	CHAINM	O	INTE	NBX	(60,20)
					CHANGE	M	INTE	NBX	(60,20)
					NEWCOL	M	INTE	NBX	(60,20)
					RESETR	M	INTE	NBX	(60,20)
NC	(I)	First section of the program: number of input core atoms bonded to core atom I excluding that of the parent core atom. At end of first section: total number of input core atoms bonded to core atom I.	/BLK2	/(3119)	CHAINM	I	INTE	NC	(100)
					CIS	I	INTE	NC	(100)
					CISCOR	I	INTE	NC	(100)
					CORCIG	I	INTE	NC	(100)
					CTWO	I	INTE	NC	(100)
					CYCORR	I	INTE	NC	(100)
					EQUALR	I	INTE	NC	(100)
					EXTROT	I	INTE	NC	(100)
					GAUCHE	I	INTE	NC	(100)
					LINEAR	I	INTE	NC	(100)
					NEWKC	I	INTE	NC	(100)
					RESETR	I	INTE	NC	(100)
					SCAN	M	INTE	NC	(100)
					SCANBR	I	INTE	NC	(100)
					SCANCH	I	INTE	NC	(100)
					SORNGI	I	INTE	NC	(100)
					STAND	M	INTE	NC	(100)
					SYMRNG	I	INTE	NC	(100)
					SYMTRY	I	INTE	NC	(100)
NCF		Upper limit of loop counter.	/EXTROT/(*	)	EXTROT	M	INTE	NCF	
NDATA		Number of elements used in the formation of the chemical groups.	/BLK5	/(1)	PRINTI	I	INTE	NDATA	
					SCANBR	I	INTE	NDATA	
					SCANCH	I	INTE	NDATA	
					SETUP	I	INTE	NDATA	
					SYMTRY	I	INTE	NDATA	
					TGAP	O	INTE	NDATA	
NDIF		Total number of rings which are part of the fused ring sets.	/FUSION/(*	)	FUSION	M	INTE	NDIF	
NDONE	(I)	Designates that the ligand core atom I has been processed when I=1. Otherwise, I=0.	/BLK1	/(3929)	EQUAL	M	INTE	NDONE	(100)
NENAN		Input value for total number of enantiomeric configurations of molecule (optional).	/CORCIG/(*	)	CORCIG	I	INTE	NENAN	
NENAN		Input value for total number of enantiomeric configurations of molecule (optional).	/ENTSYM/(*	)	ENTSYM	I	INTE	NENAN	
NENAN		Input value for total number of enantiomeric configurations of molecule (optional).	/TGAP	/(*)	TGAP	M	INTE	NENAN	

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE COL	SUBROUTINE SUBR	SUBROUTINE U	SUBROUTINE TYPE	USAGE VAR	USAGE DIM
NENANC		Computed value for the total number of enantiomers.	/ENTSYM/(	*	) ENTSYM	W	INTE	NENANC	
NENE		Number of 3-ene structures associated with pair of central core atoms.	/CISCON/(	*	) CISCON	W	INTE	NENE	
NEWCOL		Defines the initial constituents of a new chain. The group numbers of the previous chain common to the new chain are stored in a new row of array NBC. The related array *branch* data for the new chain are stored in the NBX array.	/NEWCOL/(	8	) CHAINM NEWCOL	S E		NEWCOL NEWCOL	
NEWKC		Finds the next chain atom when the last identified atom KC of the chain is a branch atom. The new chain atom is set equal to one of the unused core atom ligands of KC and is stored in KC. If all these ligands have already been incorporated in some chain link, the program exits from the subroutine unless the required storage order is not present.	/NEWKC /(	8	) CHAINM NEWCOL NEWKC	S S E		NEWKC NEWKC NEWKC	
NF		Upper limit of loop counter N.	/LESSEN/(	*	) LESSEN	W	INTE	NF	
NF		Upper limit of loop counter.	/PRINT1/(	*	) PRINT1	W	INTE	NF	
NF		Upper limit of loop counter.	/SYMRNG/(	*	) SYMRNG	W	INTE	NF	
NI		Lower limit of loop counter.	/EQUAL /(	*	) EQUAL	W	INTE	NI	
NI		Lower limit of loop counter.	/SETUP /(	*	) SETUP	W	INTE	NI	
NJ		Bond direction from ligand to first atom bonded to ligand which is not KT.	/SCAN /(	*	) SCAN	)	INTE	NJ	
NMU		Bond direction from core atom to ligand.	/SCAN /(	*	) SCAN	W	INTE	NMU	
NN		Loop counter and subscript.	/EQUAL /(	*	) EQUAL	W	INTE	NN	
NN		Settings for array MARO. 1=benzene-type ring, 2=pyridine-type ring, 0=neither.	/MEIGON/(	*	) MEIGON	W	INTE	NN	
NN		Subscript of array JORD.	/LESSEN/(	*	) LESSEN	W	INTE	NN	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
NN		Value of j subscript in IX(i,j,k) where data required for scan operation are located.	/STAND	/(*)	STAND	W	INTE	NN	
NNBENZ		Number of pyridine-type ring structures.	/BLK1	/(3979)	CORCIG	M	INTE	NNBENZ	
					CYCORA	M	INTE	NNBENZ	
					HEXGON	I	INTE	NNBENZ	
NO		Number of cards used to input structure of molecule.	/BLK1	/(1)	ASSIGN	I	INTE	NO	
					IDENT	I	INTE	NO	
					STAND	I	INTE	NO	
					TGAP	M	INTE	NO	
NOBR		Number of branch core atoms present in molecule.	/BLK3	/(6402)	CHAINR	I	INTE	NOBR	
					STAND	O	INTE	NOBR	
NOBRD	(I)	Counter that denotes which ligand of the branch atom of group number I is to be added to the chain.	/BLK3	/(1703)	CHAINR	O	INTE	NOBRD	(100)
					NEWCOL	O	INTE	NOBRD	(100)
					NEWKC	M	INTE	NOBRD	(100)
NOCAR		Denotes whether the carbon atom is the first or second central atom of the gauche interaction structure.	/GAUCHE	/(*)	GAUCHE	W	INTE	NOCAR	
NOD01		Number of entries in array KCXD01.	/EQUAL	/(*)	EQUAL	I	INTE	NOD01	
NOD01		Number of entries in array KCXD01.	/SETUP	/(*)	SETUP	I	INTE	NOD01	
NOD02		Number of entries in array KCXD02.	/EQUAL	/(*)	EQUAL	I	INTE	NOD02	
NOD02		Number of entries in array KCXD02.	/SETUP	/(*)	SETUP	I	INTE	NOD02	
NOK	(I,J)	Multi-storage array I=1 number of ring pairs in fused ring set J. I=2 total number of atoms common to any ring pair in set. I=3 total number of atoms common to all ring pairs in set. I>3 group number of atoms comprising the total in I=3.	/BLK3	/(3901)	CRINGS	I	INTE	NOK	(10,10)
					FIRSTR	I	INTE	NOK	(10,10)
					FUSION	O	INTE	NOK	(10,10)
					HEXGON	I	INTE	NOK	(10,10)
					NRINGS	I	INTE	NOK	(10,10)
NOKOMB	(I)	Contents of array. I=1 number of fused ring systems in molecule. I>1 number of rings in fused ring system I-1.	/BLK3	/(6380)	CORCIG	M	INTE	NOKOMB	(11)
					CRINGS	I	INTE	NOKOMB	(11)
					FIRSTR	I	INTE	NOKOMB	(11)
					FUSION	M	INTE	NOKOMB	(11)
					HEXGON	I	INTE	NOKOMB	(11)
					NRINGS	I	INTE	NOKOMB	(11)

17 APR 73 8.02-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
NOMESO		Function type =0 no asymmetric atom or meso structures present. =1 asymmetric atom present. =2 meso structures present.	/CORCIG/(*	)	CORCIG	I	INTE	NOMESO	
NOMESO		Function type =0 no asymmetric atom or meso structures present. =1 asymmetric atom present. =2 meso structures present. Are being compared. Are being compared.	/ENTSYM/(*	)	ENTSYM	I	INTE	NOMESO	
NOMESO		Function type. =0 no asymmetric atom or meso structures present. =1 asymmetric atom present. =2 meso structures present.	/EXTROT/(*	)	EXTROT	O	INTE	NOMESO	
NONARO	(I)	Array in which non-aromatic ring units (I) are set equal to 1 and aromatic units (I) are set equal to 0.	/BLK3	/(3602)	CORCIG HEXGON SORNGI	O O I	INTE INTE INTE	NONARO(40) NONARO(40) NONARO(40)	
NONFUS		Total number of rings in molecule which are not part of any fused ring set.	/BLK4	/(4322)	FUSION	M	INTE	NONFUS	
NONIT		Denotes whether the nitrogen atom is the first or second central atom of the gauche interaction structure.	/GAUCHE/(*	)	GAUCHE	M	INTE	NONIT	
NOPTS		Total number of enantiomeric and meso forms computed from input data.	/ENTSYM/(*	)	ENTSYM	M	INTE	NOPTS	
NOPTSC		Computed value for the total number of enantiomers and meso structures present.	/ENTSYM/(*	)	ENTSYM	M	INTE	NOPTSC	
NOS		Number of elements plus radicals used in the formation of the chemical groups.	/BLK1	/(2)	IDENT TGAP	I O	INTE INTE	NOS NOS	
NOSAME	(I,J)	Multi-variable array I=1 number of entries in array KCSAME for core atom J. I=2 number of different types of identical ligands in set. I>2 number of each type.	/BLK1	/(2678)	EXTROT SAME SETUP SYMRNG	I M O I	INTE INTE INTE INTE	NOSAME(8,100) NOSAME(8,100) NOSAME(8,100) NOSAME(8,100)	
NOSIX		Total number of six-membered rings containing at least two double bonds.	/BLK3	/(5611)	CORCIG CYCORR HEXGON	O M I	INTE INTE INTE	NOSIX NOSIX NOSIX	
NOSM		Input value for external symmetry number of molecule (optional).	/CORCIG/(*	)	CORCIG	I	INTE	NOSM	

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE SUBR	U	TYPE	USAGE VAR	DIM
NOSN		Input value for external symmetry number of molecule (optional).	/ENTSYM/(	)	ENTSYM	I	INTE	NOSN	
NOSN		Input value for external symmetry number of molecule (optional).	/TGAP /(	)	TGAP	W	INTE	NOSN	
NOSNC		Computed value for the external rotation symmetry number.	/CORCIG/(	)	CORCIG	I	INTE	NOSNC	
NOSNC		Computed value for the external rotation symmetry number.	/ENTSYM/(	)	ENTSYM	I	INTE	NOSNC	
NOSNC		Computed value for the external rotation symmetry number.	/EXTROT/(	)	EXTROT	M	INTE	NOSNC	
NOSNC		Computed value for the external rotation symmetry number.	/SYMRNG/(	)	SYMRNG	O	INTE	NOSNC	
NOVAL	(I)	Chemical valence of each element and radical. In the group additivity tables nitrogen always has a valence of three.	/BLK1 /(	16)	STAND	I	INTE	NOVAL	(9)
					TGAP	O	INTE	NOVAL	(9)
NOXY		Number of CO groups found in ring K.	/OXYATM/(	)	OXYATM	W	INTE	NOXY	
NO1ALT	(I)	Temporary storage for original NOSAME data computed for KC1.	/BLK1 /(	3895)	EXTROT	M	INTE	NO1ALT	(8)
NO2ALT	(I)	Temporary storage for original NOSAME data computed for KC2.	/BLK1 /(	3903)	EXTROT	M	INTE	NO2ALT	(8)
NPSUDA		Pseudoasymmetric atom flag =0 pseudoasymmetric atom not present. =1 pseudoasymmetric atom present.	/CORCIG/(	)	CORCIG	I	INTE	NPSUDA	
NPSUDA		Pseudoasymmetric atom flag =0 pseudoasymmetric atom not present. =1 pseudoasymmetric atom present.	/ENTSYM/(	)	ENTSYM	I	INTE	NPSUDA	
NPSUDA		Pseudoasymmetric atom flag. =0 pseudoasymmetric atom not present. =1 pseudoasymmetric atom present.	/EXTROT/(	)	EXTROT	O	INTE	NPSUDA	
NRING	(I)	Identifies nitrogen containing rings. Location I is set to 1 if ring I contains nitrogen and to 0 if it does not.	/BLK1 /(	3729)	CORCIG	O	INTE	NRING	(40)
					CYCORR	O	INTE	NRING	(40)
					HEXGON	I	INTE	NRING	(40)
NRINGS		Searches for a particular fused nitrogen-containing-ring molecule and applies ring correction if present.	/NRINGS/(	0	) CORCIG	S		NRINGS	
					NRINGS	E		NRINGS	

17 APR 73 8.02.35

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
NRLOC	(I)	The number of the chain that contains ring I is stored in location I.	/BLK3	/(3503)	LESSEN	M	INTE	NRLOC	(40)
					RESETR	I	INTE	NRLOC	(40)
					RING	M	INTE	NRLOC	(40)
NS		Bond direction of second character.	/IDENT	/( )	IDENT	W	INTE	NS	
NTBENZ		Total number of benzene- and pyridine-type rings.	/BLK3	/(6396)	CORCIG	O	INTE	NTBENZ	
					CTWO	I	INTE	NTBENZ	
					HEXGON	M	INTE	NTBENZ	
NTITLE		Title printout flag =0 title of interaction data not yet output. =1 title has been output.	/CORCIG/( )		CORCIG	W	INTE	NTITLE	
NTITLE		Title printout flag =0 title of interaction data not yet output. =1 title has been output.	/PRINT2/( )		PRINT2	O	INTE	NTITLE	
NTOTAL	(I,J)	Number of identical atoms I of a particular type that are associated with core atom J.	/BLK1	/(3478)	EXTROT	M	INTE	NTOTAL	(3,100)
					INTROT	I	INTE	NTOTAL	(3,150)
					LINEAR	I	INTE	NTOTAL	(3,100)
					SAME	M	INTE	NTOTAL	(3,100)
					SETUP	O	INTE	NTOTAL	(3,100)
					SYMRNG	M	INTE	NTOTAL	(3,100)
					SYNTRY	O	INTE	NTOTAL	(3,100)
NT1ALT	(I)	Temporary storage for original NTOTAL data computed for KC1	/BLK1	/(3911)	EXTROT	M	INTE	NT1ALT	(3)
NT1X	(I)	Same as data in array NT1ALT except that here the contribution from atom KC2 is deleted if present.	/BLK1	/(3917)	EXTROT	M	INTE	NT1X	(3)
NT2ALT	(I)	Temporary storage for original NTOTAL data computed for KC2.	/BLK1	/(3914)	EXTROT	M	INTE	NT2ALT	(3)
NT2X	(I)	Same as data in array NT2ALT except that here the contribution from atom KC1 is deleted if present.	/BLK1	/(3920)	EXTROT	M	INTE	NT2X	(3)
NUM		Bond direction 1 = north 2 = northeast 3 = east 4 = southeast 5 = south 6 = southwest 7 = west 8 = northwest	/ASSIGN/( )		ASSIGN	I	INTE	NUM	
NUM		Number of ring components.	/CYCOMP/( )		CYCOMP	W	INTE	NUM	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
NUM		Entry value: number used to compute initial bond scan direction. Exit value: bond direction from indexed atom to ligand core atom.	/FIND	/(•	) FIND	M	INTE	NUM	
NUM		Number of atoms which ring pair KSUB has in common.	/FUSION	/(•	) FUSION	M	INTE	NUM	
NUM		Bond direction from parent atom to indexed atom. Set to -3 initially prior to the identification of the first symbol of the molecule.	/IDENT	/(•	) IDENT	M	INTE	NUM	
NUM		Bond direction. Entry value: from parent core atom to indexed core atom. Computed value: from indexed core atom to parent core atom.	/NUMBER	/(•	) NUMBER	M	INTE	NUM	
NUM		Bond direction from parent atom to core atom.	/SCAN	/(•	) SCAN	I	INTE	NUM	
NUM		Initial setting for the bond direction variable.	/STAND	/(•	) STAND	M	INTE	NUM	
NUM		Number of sets of identical ligands on the ring.	/SYMRNG	/(•	) SYMRNG	M	INTE	NUM	
NUMATM	(I)	Contains elemental composition of molecule.	/BLK5	/(	2) IDENT	I	INTE	NUMATM(5)	
					PRINT1	I	INTE	NUMATM(5)	
					SCAN	I	INTE	NUMATM(5)	
					SCANBR	I	INTE	NUMATM(5)	
					SCANCH	I	INTE	NUMATM(5)	
					STAND	O	INTE	NUMATM(5)	
NUMATM	(I)	Contains elemental composition of molecule or ligand.	/SUMATM	/(•	) SUMATM	M	INTE	NUMATM(5)	
NUMBER		This subroutine determines the number of ligands (MP) attached to core atom 1 or the number of ligands - 1 of any core atom whose group number exceeds 1.	/NUMBER	/(0	) FIND	S		NUMBER	
					NUMBER	E		NUMBER	
					SCAN	S		NUMBER	
NUMC		Number of carbon atoms in ring backbone.	/BLK3	/(5776)	CYCORR	M	INTE	NUMC	
NUMCO		Number of CO atoms in ring backbone.	/BLK3	/(5779)	CYCORR	M	INTE	NUMCO	
NUMFRQ	(I)	Contains the subscript of array SYMBOL that identifies each element in a radical and the number of each element contained therein. Six locations per radical. Odd locations contain subscript, even locations the number.	/BLK6	/(1)	SUMATM	I	INTE	NUMFRQ(24)	
					TOAP	O	INTE	NUMFRQ(24)	
NUMN		Number of nitrogen atoms in ring backbone.	/BLK3	/(5778)	CYCORR	M	INTE	NUMN	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
NUMO		Number of oxygen atoms in ring backbone.	/BLK3	/(5777)	CYCORR	M	INTE	NUMO	
NV	(I)	Number of core atoms bonded to atom I. If I is a ring atom, value is reset in subroutine EQUALR to the number of non-ring core atoms bonded to I plus one.	/BLK1	/(1425)	EQUAL	I	INTE	NV	(100)
					EQUALR	M	INTE	NV	(100)
					SYMTRY	O	INTE	NV	(100)
NVR	(I)	Number of ring atoms bonded to ring atom I.	/BLK4	/(4162)	EQUALR	M	INTE	NVR	(100)
NW	(I)	Number of actual core atoms bonded to core atom i, this includes the oxygen atom of the CO radical.	/BLK3	/(5201)	DITERE	I	INTE	NW	(100)
					GAUCHE	I	INTE	NW	(100)
					STAND	M	INTE	NW	(100)
NX		Redefinition value for bond direction if limit is exceeded.	/CIS	/(*)	CIS	W	INTE	NX	
NX		Flag that controls the transfer to the various ring correction sections in subroutine CYCORR.	/CYCORR/	(*)	CYCORR	W	INTE	NX	
NX		Sum of asymmetric and pseudoasymmetric atoms.	/ENTSYM/	(*)	ENTSYM	W	INTE	NX	
NX		Transfer flag to computation areas for diverse structural types.	/EXTROT/	(*)	EXTROT	W	INTE	NX	
NX		Bond direction from indexed atom to number 1 (core) ligand.	/FIND	/(*)	FIND	I	INTE	NX	
NX		Current test bond direction.	/IDENT	/(*)	IDENT	W	INTE	NX	
NX		Bond direction from core atom.	/NUMBER/	(*)	NUMBER	W	INTE	NX	
NX		Bond direction from core atom to test location.	/SCAN	/(*)	SCAN	W	INTE	NX	
NXX		Bond direction from indexed core atom to ligand one.	/NUMBER	(*)	NUMBER	O	INTE	NXX	
NX3		Number of entries in array JSTORE.	/NRINGS/	(*)	NRINGS	W	INTE	NX3	
NV		Value outside renge of bond direction.	/CIS	/(*)	CIS	W	INTE	NV	
NZ		Bond direction from number 1 (core) ligand to indexed atom.	/FIND	/(*)	FIND	W	INTE	NZ	
NZ		Initial test bond direction.	/IDENT	/(*)	IDENT	W	INTE	NZ	
NZ		Bond direction from core atom to parent atom. Inverse of NUM.	/SCAN	/(*)	SCAN	W	INTE	NZ	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	LOC	SUBROUTINE SUBR	U	TYPE	USAGE VAR	DIM
M1		Ring number.	/CRINGS/(	*	) CRINGS	M	INTE	M1	
N1		Location in IDX where group number of ligand of KC1P is stored.	/EQUAL /(	*	) EQUAL	M	INTE	N1	
N1B		Direction of bond adjacent to bond between atoms K1R and KC1PR.	/EQUALR/(	*	) EQUALR	M	INTE	N1B	
N1E		Number of non-ring core atoms bonded to an atom in ring A.	/EQUALR/(	*	) EQUALR	M	INTE	N1E	
N1F		Upper limit of loop counter.	/EQUAL /(	*	) EQUAL	M	INTE	N1F	
N1R		Number of other ring atoms bonded to an atom in ring A.	/EQUALR/(	*	) EQUALR	M	INTE	N1R	
N2		Ring size.	/CRINGS/(	*	) CRINGS	M	INTE	N2	
N2		Location in IDX where group number of ligand of KC2P is stored.	/EQUAL /(	*	) EQUAL	M	INTE	N2	
N2E		Number of non-ring core atoms bonded to an atom in ring B.	/EQUALR/(	*	) EQUALR	M	INTE	N2E	
N2R		Number of other ring atoms bonded to an atom in ring B.	/EQUALR/(	*	) EQUALR	M	INTE	N2R	
N3		Ring correction indicator.	/CRINGS/(	*	) CRINGS	M	INTE	N3	
OPTS		Floating point equivalent of the total number of enantiomeric and meso structures.	/ENTSYM/(	*	) ENTSYM	M	REAL	OPTS	
ORDER		Stores in array IA the order that the elements in array IB would have if rearranged in numerical order.	/ORDER /(	6	) CORCIB EQUALR LESSEN ORDER TGAP	S S S E S		ORDER ORDER ORDER ORDER ORDER	
ORDSUM	(I)	Contains order elements in array SUM would have if arranged in order of increasing magnitude.	/SEARCH/(	*	) SEARCH	I	INTE	ORDSUM(	100)
ORDSUM	(I)	Contains order elements in array SUM would have if arranged in order of increasing magnitude.	/TGAP /(	*	) TGAP	I	INTE	ORDSUM(	100)
OXYATM		Determines the number of CO atoms in a particular ring which are adjacent to another specific atom in the ring.	/OXYATM/(	6	) CYCORR OXYATM	S E		OXYATM OXYATM	

17 APR 73 8.08-36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DTA
PERLIG	(I,J)	Ordinal number I of ligand bonded to atom J.	/BLK4	/(3762)	EQUAL	I	INTE	PERLIG(4,100)	
					SCANBR	O	INTE	PERLIG(4,100)	
					SCANCH	O	INTE	PERLIG(4,100)	
					SETUP	I	INTE	PERLIG(4,100)	
					SYMANG	O	INTE	PERLIG(4,100)	
PERTKC	(I,J)	Group number of first atom in ligand I that is bonded to core atom J. If atom is non-core, the negative of the chemical symbol subscript of the atom is stored instead.	/BLK4	/(3362)	EQUAL	I	INTE	PERTKC(4,100)	
					SCANBR	O	INTE	PERTKC(4,100)	
					SCANCH	O	INTE	PERTKC(4,100)	
					SETUP	I	INTE	PERTKC(4,100)	
					SYMANG	I	INTE	PERTKC(4,100)	
					SYMTRY	O	INTE	PERTKC(4,100)	
PRINT1		Prints the atomic composition, the assigned weights, the structure and coordinate array IX, and (and if rings are present) the number of components in each ring and their group numbers.	/PRINT1/(6	)	CORCIG	S		PRINT1	
					HEXGON	S		PRINT1	
					PRINT1	E		PRINT1	
					TGAP	S		PRINT1	
PRINT2		Prints the title preceding the printout of ring, second-order interaction, and internal rotational symmetry contributions.	/PPINT2/(6	)	CORCIG	S		PRINT2	
					PRINT2	E		PRINT2	
RDATA	(I,J)	Thermodynamic corrections for certain fused carbon-ring systems.	/CRINGS/(*	)	CRINGS	I	REAL	RDATA(6,7)	
RDATA	(I)	Thermodynamic corrections for a particular fused nitrogen-containing-ring molecule.	/NRINGS/(*	)	NRINGS	I	REAL	RDATA(6)	
RESETR		Sets the ring array IBC and adds to all the chain columns the residual non-branch ring atoms, if the first chain atom defined was a non-branch ring atom. These residuals were not included when the chains were first constructed.	/RESETR/(6	)	CHAINM	S		RESETR	
					RESETR	E		RESETR	
RING		Determines if similar ring was not previously identified. If not, the group numbers of the ring constituents are stored in IRING.	/RING /(6	)	CHAINM	S		RING	
					RING	E		RING	
RINGD	(I,J)	Array equivalenced to the entire set of arrays A1, A2 etc.	/CORCIG/(*	)	CORCIG	I	REAL	RINGD(6,50)	
RINGLI	(I)	Contains variable that designates the ring correction to be applied to ring I.	/BLK3	/(5695)	CORCIG	M	INTE	RINGLI(40)	
ROTINS		Total contribution to the entropy due to internal rotational symmetry.	/CORCIG/(*	)	CORCIG	I	REAL	ROTINS	
ROTINS		Total contribution to the entropy due to internal rotational symmetry.	/INTROT/(*	)	INTROT	M	REAL	ROTINS	
S		Entropy of molecule for 298°K. [cal-deg^{-1}-mole^{-1}]	/TGAP /(*	)	TGAP	M	REAL		

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	USAGE VAR	USAGE DIM
SAME		Stores all the pertinent identification variables for any two identical ligands. If one of these ligands has been thus identified previously, it is bypassed and only the variables for the other ligand are stored.	/SAME	/(3	) EQUAL SAME SETUP	S E S		SAME SAME SAME	
SCAN		Locates and identifies all atoms bonded to core atom KT of group KT which have not yet been scanned and stores pertinent data of ligand in IX. It also rearranges data of group KT in IX such that data of core species are stored first followed by data of non-core species.	/SCAN	/(8	) SCAN STAND	E S		SCAN SCAN	
SCANBR		Computes the atomic composition of all ligands bonded to a branch atom and establishes the similarities of said ligands providing the branch atom is not a ring atom.	/SCANBR/(8		) SCANBR SYMTRY	E S		SCANBR SCANBR	
SCANCH		Computes atomic composition of all ligands bonded to a chain atom and establishes the similarities of these ligands providing said chain atom is not a ring atom.	/SCANCH/(8		) SCANCH SYMTRY	E S		SCANCH SCANCH	
SCONS		Entropy constant.	/TGAP	/(*	) TGAP	W	REAL	SCONS	
SEARCH		Integer function flag. =0 a data match for computed group weight MOLWT(k] has been found. =1 no match found, computed weight below range of data group weights. =-1 no match found, computed weight above range of data group weights.	/SEARCH/(8		) SEARCH TGAP	O F		SEARCH SEARCH	
SETUP		Processes the comparison of ligands bonded to the same or to different core atoms by first comparing their atomic compositions. If the compositions are the same and both are non-core ligands, they are identical but if the ligands are non-core atoms, their identity must be checked further.	/SETUP	/(8	) EXTHOT SCANBR SCANCH SETUP SYMRNG	S S S E S		SETUP SETUP SETUP SETUP SETUP	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	USAGE VAR	DIM
SHIFT		Data contained in column j=2 of IX(JC,j,k) is switched with that of core atom KC.	/SHIFT/(0	)	CORCIG EXTROT SHIFT STAND	S S E S		SHIFT SHIFT SHIFT SHIFT	
SN		Floating point equivalent for the symmetry number.	/ENTSYM/(*	)	ENTSYM	M	REAL	SN	
SOPTS		Total contribution to the entropy due to optical isomerism.	/CORCIG/(*	)	CORCIG	I	REAL	SOPTS	
SOPTS		Total contribution to the entropy due to optical isomerism.	/ENTSYM/(*	)	ENTSYM	O	REAL	SOPTS	
SOPTSC		Computed value for the contribution to the entropy due to optical isomerism.	/ENTSYM/(*	)	ENTSYM	M	REAL	SOPTSC	
SOPTSI		Entropy contribution due to optical isomerism computed from input data.	/ENTSYM/(*	)	ENTSYM	M	REAL	SOPTSI	
SORNGI		Controls the search for gauche and cis interactions in which one or both of the central atoms are non-aromatic ring atoms and for their contributions if these interactions occur.	/SORNGI/(0	)	CORCIG SORNGI	S E		SORNGI SORNGI	
SRING		Total of ring contributions to the entropy.	/CORCIG/(*	)	CORCIG	M	REAL	SRING	
SRING		Sum of ring contributions to the entropy.	/CRINGS/(*	)	CRINGS	M	REAL	SRING	
SRING		Sum of ring contributions to the entropy.	/HEXGON/(*	)	HEXGON	M	REAL	SRING	
SRING		Sum of ring contributions to the entropy.	/NRINGS/(*	)	NRINGS	M	REAL	SRING	
SSN		Total contribution to the entropy due to external rotational symmetry	/CORCIG/(*	)	CORCIG	I	REAL	SSN	
SSN		Total contribution to the entropy due to external rotational symmetry.	/ENTSYM/(*	)	ENTSYM	O	REAL	SSN	
SSNC		External rotational entropy contribution computed by program.	/ENTSYM/(*	)	ENTSYM	M	REAL	SSNC	
SSNI		External rotational entropy contribution computed from input data.	/ENTSYM/(*	)	ENTSYM	M	REAL	SSNI	
SSYM		Total contribution to the entropy from second-order interactions, ring corrections, symmetry and optical isomerism.	/CORCIG/(*	)	CORCIG	O	REAL	SSYM	

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
SSYM		Sum total of corrections to the entropy at 298°K derived from section 3 of the program. [cal-deg^{-1}-mole^{-1}]	/TGAP	/(*	) TGAP	I	REAL	SSYM	
ST	(I)	Entropy of molecule for temperatures specified in TARRAY. [cal-deg^{-1}-mole^{-1}]	/BLK3	/(1219)	YGAP	M	REAL	ST	(14)
STAND		Control routine for section 1 of the program.this section identifies the groups and group components of the molecule as well as the atomic coordinates, bond vectors and bond types. The group weights are also assigned. For some unsaturated ring compounds the latter is altered in section 3 of the program.	/STAND	/(8	) STAND TGAP	E S		STAND STAND	
SUM	(I)	Identification weight of each group in the data library.	/BLK7	/(1)	DATA1 GADATA TGAP	D I C	INTE INTE INTE	SUM SUM SUM	(180) (180) (180)
SUM		Identification weight of particular group contained in data library.	/SEARCH/	(*	) SEARCH	I	INTE	SUM	(180)
SUMATM		Adds or subtracts (when KP= 1 or -1 respectively) the number of each element present in the chemical symbol represented by M to the location assigned that particular atom in array umatm.	/SUMATM/	(8	) IDENT SCAN SCANBR SCANCH SUMATM SYMTRY	S S S S E S		SUMATM SUMATM SUMATM SUMATM SUMATM SUMATM	
SUM1		Sum of symbol code numbers of carbon atoms one and two.	/DITERE/	(*	) DITERE	W	INTE	SUM1	
SUM2		Sum total of core atoms bonded to carbon atoms one and two.	/DITERE/	(*	) DITERE	W	INTE	SUM2	
SYMBL	(I,J)	Contains character I of multi-character program library symbol J. The description and order of these symbols correspond to those of the single word, multi-character Hollerith program library symbols.	/MULTI	/(*	) MULTI	C	INTE	SYMBL	(3,9)
SYMBOL	(I)	Chemical symbol for elements and radicals contained in the program.	/BLK1	/(7)	IDENT PRINT1 TGAP	I I D	INTE INTE INTE	SYMBOL(9) SYMBOL(9) SYMBOL(9)	
SYMRNG		Computes the external rotation symmetry number of aromatic, monocyclic benzene-type structures.	/SYMRNG/	(8	) EXTROT SYMRNG	S E		SYMRNG SYMRNG	

17 APR 73 8.02.36

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE USAGE SUBR	U	TYPE	VAR	DIM
SYMTRY		Controls the operation which finds the atomic composition of each ligand of each core atom in the molecule. If the core atom is not a ring atom, the routine determines whether it possesses identical ligands.	/SYMTRY/(	)	CORCIG SYMTRY	S E		SYMTRY SYMTRY	
SYMX	(I)	Contains symbols for bond types and blank.	/BLK1 /(	3)	BOND IDENT NUMBER SCAN STAND TGAP	I I I I I D	INTE INTE INTE INTE INTE INTE	SYMX SYMX SYMX SYMX SYMX SYMX	(9) (9) (9) (9) (9) (9)
S298	(I)	Entropy at 298°K of each group in the data library. [cal-deg^{-1}-mole^{-1}]	/BLK7 /(	361)	DATA1 GADATA TGAP	D I I	REAL REAL REAL	S298 S298 S298	(180) (180) (180)
TARRAY	(I)	Temperature values used to compute and output thermochemical data.	/TGAP /(*	)	TGAP	C	REAL	TARRAY	(19)
TCIS		Floating point equivalent of variable ICIS.	/CORCIG/(*	)	CORCIG	W	REAL	TCIS	
TGAP		Main or control unit of the thermochemical group additivity program. It prints out data library (optional),reads input data, prints outstructural input data, transfers control to 3 main sections of the program, and computes and prints out thermochemical properties of molecule	/TGAP /(8	)	TGAP	E		TGAP	
TOUT	(I)	Output legend that specifies whether property printed was input, computed, or is not computable.	/ENTSYM/(*	)	ENTSYM	C	REAL	TOUT	(12)
TYPE	(I,J)	Contains legends that describe the type of second-order interaction to be printed.	/CORCIG/(*	)	CORCIG	C	REAL	TYPE	(4,8)
WEIGHT	(I)	Identification number assigned each element and radical.	/BLK2 /(	1)	DELTA1 TGAP	I D	INTE INTE	WEIGHT(9) WEIGHT(9)	
.UNOS.		Description not input	/.UNOS./(8	)	TGAP	I		.UNOS.	

17 APR 73 8.02-56

FORTRAN SYMBOL	MATH SYMBOL	DESCRIPTION	STORAGE BLOCK	STORAGE LOC	SUBROUTINE SUBR	SUBROUTINE U	SUBROUTINE TYPE	USAGE VAR	USAGE DIM
.UN06.		Description not input	/.UN06./	0	BOND	0		.UN06.	
					CHAINM	0		.UN06.	
					CORCIS	0		.UN06.	
					DITERE	0		.UN06.	
					ENTSY	0		.UN06.	
					EQUAI	0		.UN06.	
					FIND	0		.UN06.	
					FIRSTR	0		.UN06.	
					GADATA	0		.UN06.	
					IDENT	0		.UN06.	
					MAXCHN	0		.UN06.	
					PRINT1	0		.UN06.	
					PRINT2	0		.UN06.	
					SCAN	0		.UN06.	
					STAND	0		.UN06.	
					SYMTRY	0		.UN06.	
					TGAP	0		.UN06.	

17 APR 73 8.02-36