

SYSTEMS EFFECTIVENESS - NAVY

for presentation at

Summer Program for Industry
1964 Reliability and Industrial
Statistics Course

University of California at Los Angeles
10 August 1964

by

Keith N. Sargent
Head, Systems Effectiveness Branch
Office of Naval Material

AD696069

United States
Naval Postgraduate School



THESIS

A PROPOSED STUDENT OFFICER INFORMATION
RETRIEVAL SYSTEM FOR THE NAVAL POSTGRADUATE SCHOOL

by

Robert Lee Henry

and

Lawrence Lee Massa

June 1969

*This document has been approved for public re-
lease and sale; its distribution is unlimited.*

PERMISSION TO
CLEARINGHOUSE
OF FEDERAL GOVERNMENT & TECHNICAL
INFORMATION CENTER

NOV 10 1969
146

A Proposed Student Officer Information
Retrieval System for the Naval Postgraduate School

by

Robert Lee Henry
Lieutenant Commander, United States Navy
B.S., San Francisco State College, 1959

and

Lawrence Lee Massa
Lieutenant Commander, United States Navy
B.S., Naval Postgraduate School, 1964

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1969

Signature of Authors

Lawrence L. Massa

Robert L. Henry

Approved by

Richard K. Todding

Thesis Advisor

Brooks J. Lockhart
Chairman, Committee for Computer Science

R. F. Richart
Academic Dean

ABSTRACT

Pertinent information on past and present Naval Postgraduate School students is now maintained, stored and processed in bulk files by Curriculum Officers. Information desired for management studies or analysis requires manual sorting of an ever increasing number of individual student records. This is an inadequate and inefficient system.

The foregoing problem could be resolved by the implementation of the Student Officer Information Retrieval System (SOIRS), which is a narrow scope retrieval system specifically designed to be responsive to the Curriculum Officer's needs with respect to student information. SOIRS evolved through a series of logical system design steps, identified as follows: (1) Problem Analysis; (2) Design of Records, Files and Reports; (3) Software Design; (4) Test of Entire System.

SOIRS is a complete system, establishing required files, updating files, and retrieving stored information.

TABLE OF CONTENTS

I.	INTRODUCTION -----	11
II.	PROBLEM ANALYSIS -----	14
III.	SYSTEM DESIGN -----	17
	A. LOGICAL RECORD DESIGN -----	19
	B. REPORT DESIGN FOR RETRIEVAL OUTPUT -----	21
	C. DESIGN OF FILES -----	23
	1. File Organization -----	25
	2. File Residence -----	27
	D. FILE MAINTENANCE -----	30
	1. Creation of Historical File -----	30
	2. File Transactions -----	31
	a. Input Record Verification -----	31
	b. File Updates and Transfers -----	32
	3. Quarterly File Maintenance -----	34
IV.	SOFTWARE DESIGN -----	36
	A. BUPERS HISTORICAL FILE INITIALIZATION (BUPSXFR) -----	36
	B. INPUT CARD VERIFICATION (CRDCHK1) -----	37
	C. CURRENT FILE UPDATE AND TRANSFER (UPDATE) -----	41
	D. CURRENT AND HISTORICAL FILE INFORMATION RETRIEVAL (FILESRCH) -----	43
V.	CONCLUSION AND RECOMMENDATIONS -----	49
	APPENDIX A NPGS STUDENT OFFICER INFORMATION RETRIEVAL SYSTEM USER'S MANUAL -----	53

APPENDIX B	PROGRAM DOCUMENTATION -----	90
	FLOW CHARTS -----	90
	CRDCHK 1 -----	90
	BUPSXFR -----	91
	UPDATE -----	92
	FILESROH -----	93
	PROGRAM LISTINGS -----	96
	CARDCHK 1 -----	96
	BUPSXFR -----	112
	UPDATE -----	119
	FILESROH -----	125
APPENDIX C	STATUS OF FILES AND PROGRAMS -----	145
	INITIAL DISTRIBUTION LIST -----	146
	FORM DD 1473 -----	149

LIST OF FIGURES

FIGURE	PAGE
1. TASKS AND SUBTASKS IDENTIFIED IN THE INITIAL PLANNING PHASE	13
2. SYSTEM DESIGN SUBTASKS	18
3. SOIRS LOGICAL RECORD	20
4. SAMPLE BUPERS OUTPUT	24
5. BUPSXFR BINARY SEARCH OF TABLE CURRTAB	33
6. DECISION ALGORITHM FOR DETERMINATION OF SEARCH-KEY PARAMETERS	46
7. INPUT CARD VERIFICATION PROGRAM OUTPUT LISTING WITH ERRORS (TWO PAGES)	65
8. INPUT CARD VERIFICATION PROGRAM OUTPUT LISTING WITHOUT ERRORS	67
9. INSTRUCTIONS TO INCOMING STUDENTS FOR COMPLETING REQUIRED "SOIRS" DATA FORM (FIVE PAGES)	68
10. CURRENT FILE UPDATE AND TRANSFER INPUT DATA CARD FORM	77
11. CURRENT FILE UPDATE AND TRANSFER PRINTED OUTPUT LISTING	78
12. CURRENT FILE SEARCH PRINTED OUTPUT LISTING NUMBER 1	84
13. CURRENT FILE SEARCH PRINTED OUTPUT LISTING NUMBER 2	85
14. CURRENT FILE SEARCH PRINTED OUTPUT LISTING NUMBER 3	86
15. CURRENT FILE SEARCH PRINTED OUTPUT LISTING NUMBER 4	87
16. HISTORICAL FILE SEARCH PRINTED OUTPUT LISTING NUMBER 1	88
17. HISTORICAL FILE SEARCH PRINTED OUTPUT LISTING NUMBER 2	89

LIST OF DEFINITIONS

1. Data Element - A cohesive unit of information within a record (e.g., the data representing the sex of an individual in a logical record would be considered a data element).
2. Record (Logical Record) - A collection of related data elements treated as a unit. (E.g., in an inventory control application, one line of an invoice is a data element and the complete invoice is a record.)
3. File - A collection of related records treated as a unit.
4. Entity - A person, place, or thing.
5. Information Attribute - A characteristic of an entity such as a person's name, curriculum, service number, etc.
6. Master File - A recording of the information about one set of entities of concern to the System (e.g., Current Master File is a file composed of records for each entity (Officer Student) currently attending the NPGS). There is one record for each entity.
7. Data Base - The total collection of operational data (data elements) in a system.
8. Batch Processing - A method of operation where transaction records are collected and system resources scheduled in such a way that the transaction file is processed against Master Files (in the case of SOIRS the master files would be the Historical and Current Files).
9. Binary Search - A nonsequential search of a table or list which is divided into two parts. A probe into such a list will yield one bit of information (i.e., the probe was either directed to the correct part of the list or it was not). Binary searches may have other binary searches as subsets, thus forming a binary tree search structure. Normally, the tree structure is developed to a level where short sequential searches of attributes can be conducted.
10. Bit - A binary digit representing the smallest manageable unit within a machine.
11. Byte - A sequence of eight adjacent bits; the most elementary addressable unit and is used to store one character (A,2,%,etc.,). In the system/360 a byte is reflected by base 16 notation within the machine.

the number $1_{10} = 11110001_2 = F_{16}$

the letter A = $11000001_2 = C_{16}$

the left four bits of a byte are called the Zone Bits.

12. Boolean Masking - A bit or byte area of storage established at decision points within a processing program to compare conditions and alternatives based on boolean arithmetic. Logical "AND" and "OR" operations are typically used in masking schemes.

13. File-Activity Ratio - Defined as the ratio of the number of records used in one processing run to the number of records scanned in one run. In SOIRS this ratio could vary from a small number to 100% on a retrieval run, and will be 100% for all update runs. A large file-activity ratio implies the use of a high speed magnetic tape device. Tape will accomplish the processing in about the same time as using a direct-access device, and the former is a cheaper storage medium.

14. File-Volatility - Refers to the number of record additions and deletions to a file. In a static file environment, direct-access devices can be used with excellent results, however as the file-volatility increases these devices become quite inefficient due to a chaining process required to tie the file together. At this point a tape oriented file will provide a higher degree of efficiency.

15. Hashing Function - An algorithm which maps a set of keys into a set of integers, where the integers point to the specific storage location of the key related data.

16. Information Retrieval System - A process developed to recover specific information from a data bank.

17. Key-Searches - Searches conducted to retrieve information in accordance with a specific data input parameter (or set of parameters) called a key (search-key).

18. Key Transformation - See Hashing Function

19. NIBBLE - The righthand four bits of a byte. A NIBBLE is used in a great number of logical operations when the zone bits are not required.

20. Random Processing - A method of processing a file where the records of that file are not necessarily ordered in the same manner as the transaction records. A key transformation is utilized to direct the processing to the desired storage location.

21. Sequential Organization - A file organized in such a way that each record in the file is assumed to be placed in a series based upon an ascending sequence of some key field. In the case of SOIRS, this key field is the last name of students. The method of processing such

a file is termed Sequential Batch Processing (Sequential Processing), and is characterized by the condition that insertions and deletions of records to a file require rewriting the entire file. The transaction file is ordered in the same manner as the master file. Processing of the master file results in a serial transfer of records from the old master file to a new master file; inserting records on the new master and omitting the transfer of deleted records to the new master as directed by the transaction file. The obvious justification for such a file is a high file-activity ratio and a moderate (or higher) number of record additions and deletions.

22. Serial Search - An item by item key-search of a list (or file) until the search condition is satisfied, or the end of the list (file) is encountered.

I. INTRODUCTION

The maintenance and storage of pertinent information with regard to Officer Students presently attending the Naval Postgraduate School, Monterey, as well as those who previously attended, is currently the responsibility of the curriculum officer for each respective curricula. The bulk of this information is resident in one form or another in a manual storage media such as file folders, file cabinets or desk drawers. Over the years, the number of records accumulated is, in some cases, quite large, and consequently any retrieval of information required for one time reports, analysis projects, or recurring reports is a laborious undertaking. Additionally, the manual processing required to retrieve information might inhibit the initiative of an inspired individual who would otherwise attempt to use this data base for worthwhile analysis studies.

The authors have undertaken the project of developing a flexible and accessible automated student information retrieval system for the specific purpose of aiding the proper management authority in the execution of certain planning and control functions. The proposed system would be magnetic tape oriented and designed for implementation on the Naval Postgraduate School IBM/360-67 Computer. It would additionally be upward compatible with the IBM/360 family of computers from the Model 30 on.

The system as conceived could be utilized as a stand alone application which at a later date could be easily expanded to encompass the larger data base of a total personnel accounting sub-system, or it

could be integrated, with equal ease, into a total management information system. The goal of the design is to develop an information retrieval system which will assist the Curriculum Officer (or other designated managers) to carry out certain assigned responsibilities by providing needed information stored in a data bank, which could be retrieved easily when required.

The system to accomplish this is defined as one which would collect the data, update appropriate files in the data bank, and have the capability of retrieving any or all information from the data base.

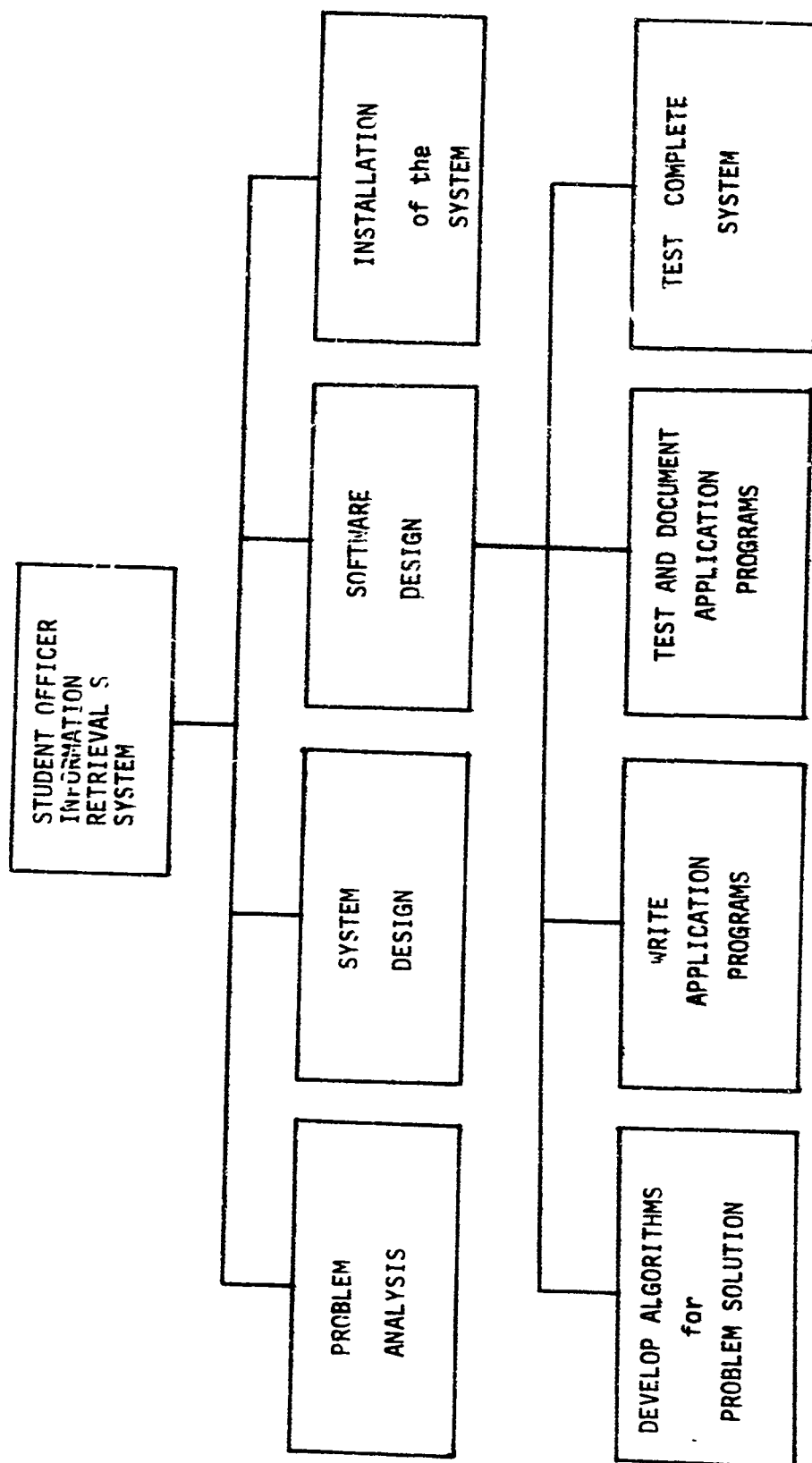
In the conceptual stage of planning, the major tasks involved in the successful development of the system were identified:

1. Problem Analysis (define the problem and the problem solution from the Curriculum Officers point of view).
2. System Design (develop the logic of the problem solution (algorithms), design the data base, and determine most appropriate method of file storage).
3. Software Design (write application programs for required algorithms, determine system programs to be utilized, and determine the programming language to be used for application programs).
4. Installation of the System (initialize the data base, develop user procedures, and document file status).

The foregoing tasks are illustrated in figure 1 together with a level of subtasks for Software design.

The remainder of this paper describes the evolution of the system development in steps corresponding to aforementioned tasks.

FIGURE 1
TASKS AND SURTASKS
IDENTIFIED IN THE
INITIAL PLANNING PHASE



II. PROBLEM ANALYSIS

In the design of any information system, safeguards must be developed to insure that the system will be an effective management tool and responsive to the needs of as many managers as possible without overburdening the data base with information of marginal importance. The goal is to provide only that information which is necessary or highly desirable for management planning, operating, and control. Interviews with curricular officers were conducted to:

1. Insure the integrity of the proposed system;
2. Determine the most effective operational data for inclusion in the data base;
3. Ascertain the type of report information desired from the system.

Initially all Curriculum Officers were contacted and received a brief description of the proposed Information Retrieval System. At that time a list of tentative data to be included in the data base was submitted to the curriculum officers, and interviews were arranged for future dates.

The interviews generated the following list of necessary (or highly desirable) data which would be included in the data base:

1. File number
2. Name
3. Grade
4. Designator
5. Year group
6. Source code
7. Date of rank
8. P-code
9. Date of birth

10. Sex
11. Social security number
12. Expected retirement year
13. Curriculum number
14. Level of education
15. Designator change history
16. Date of arrival at NPGS
17. Graduate Quality Point Rating (QPR)
18. Total QPR
19. Date of graduation from NPGS
20. Degree area
21. Country
22. Service
23. Past duty station history

Interviews additionally indicated that it would be desirable to develop retrieval methods which would be responsive to information inquiries keyed to the following elements (or combinations thereof) of a logical record within the data base:

1. Designator
2. Curriculum
3. Grade
4. Date of birth
5. Sex
6. Level of education
7. Date of arrival (departure) to (from) the NPGS
8. Degree area
9. Country
10. Service
11. Graduate QPR
12. Total QPR

Based upon the Curriculum Officers expected use of an information system, it was determined that the foregoing requirements developed

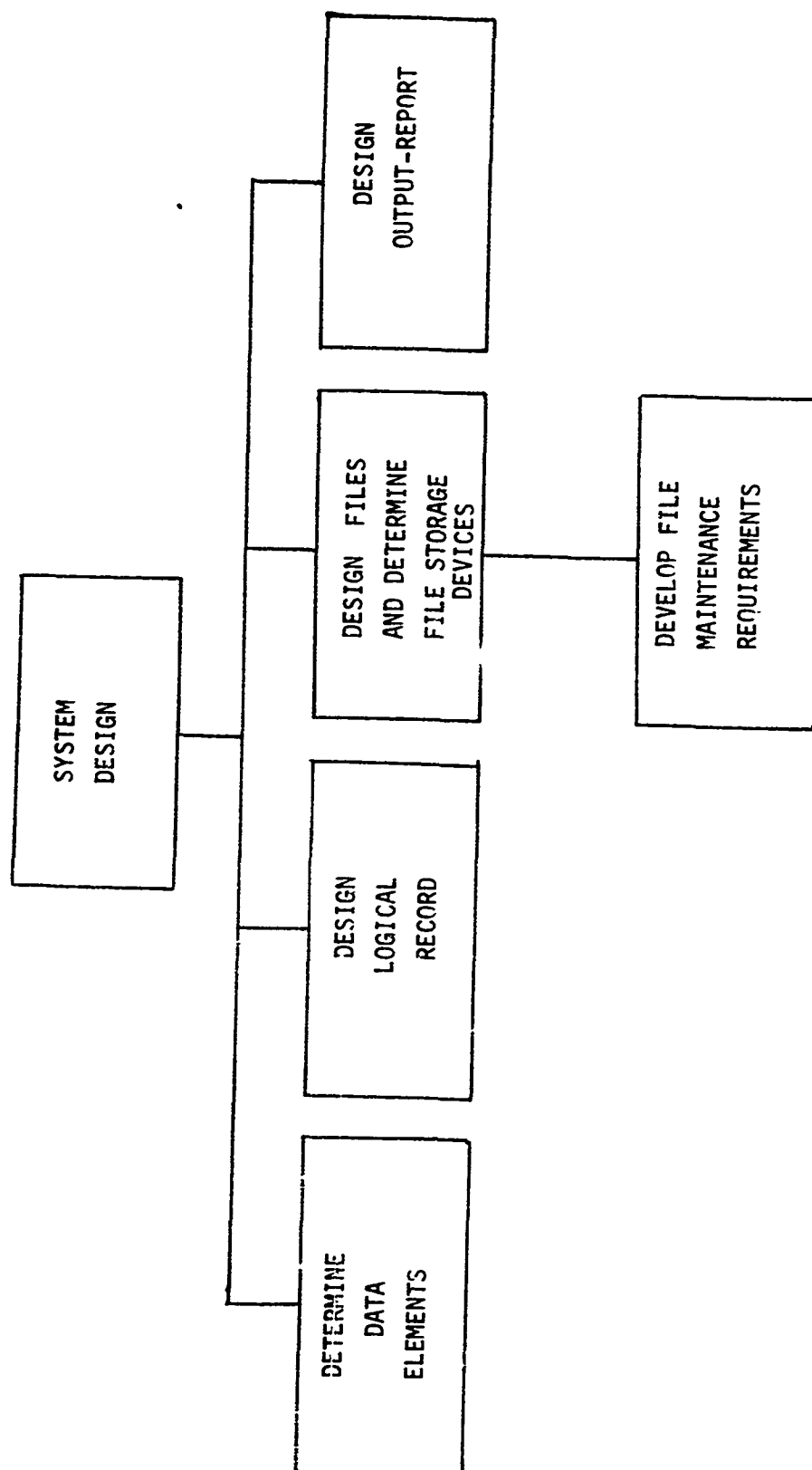
during the problem analysis phase were realistic, and in most cases justified. Consequently, the design of the Student Officer Information Retrieval System (SOIRS) was initiated.

III. SYSTEM DESIGN

System design within the context of this paper will refer primarily to the development of logical record structure, file structure, file maintenance requirements, and report design. (Figure 2 presents the subtasks involved in this system design effort.)

The information content of an information processing system is given by its data base. Because of the extreme importance of information content the data base together with data files are planned, then the system is built around them (it is recommended that the reader refer to the LIST OF DEFINITIONS and read the entries "Data Element" through "Data Base" in the order presented). The determination of operational data for the data base was a two step procedure: (1) the first being locally generated at the NPGS as described in the section "Problem Analysis" and; (2) the second being the type of data which could be provided by the Bureau of Naval Personnel (BUPERS) to be used primarily in the development of a historical file of past NPGS students. BUPERS data relates only to officers in the U.S. Navy, whereas the NPGS generated additional data requirements due to the attendance of foreign students, allied officer students (other U.S. Services), and civilians. Unfortunately, an excessive amount of time was required in obtaining a BUPERS file, and the intersection of locally generated data elements and BUPERS data elements was not as large as desired. However, with the exception of two information attributes (undergraduate education; secondary P-code) all BUPERS data was utilized. The resulting information attributes for an entity within this system are denicted

FIGURE 2
SYSTEM DESIGN SUBTAS,



in Figure 3, p. 20 (BUPERS did not provide information concerning date of arrival at NPGS, quality point rating (graduate or total), country, service, or history of past duty stations). All of the information attributes with the exception of Past Duty Station History are obviously useful within the concept of an information processing system. The inclusion of the current and seven previous duty stations was encouraged by the majority of people interviewed and justified by the desire to obtain information on the utilization of officer graduate education in duty station assignments, particularly in the technical areas. The record was designed before the realization that BUPERS did not provide the desired information; however, if the system is implemented, subsequent Bureau updates could be used to locally generate a duty station history.

A. LOGICAL RECORD DESIGN

After determining the information attributes per entity, there was no difficulty encountered in assigning data elements to a logical record. The only specific requirements being: (1) insure that all data elements which might be utilized for key searches be contiguous and located at the beginning of the record in order to maintain a minimum length test mask, and minimize both the length of the data string and number of string manipulations required in search routines; and (2) provide space for a one byte test character expected to be utilized in update routines.

The logical record is presented in figure 3. Since all input to the system will be 80 byte card images (after initialization of the Historical File), the record naturally takes a form resembling 3 card

FIGURE 3

SOIRS LOGICAL RECORD

A		B		C		D		E		F		G		H	
0	5	6				35		38		42		45		48	
I		J		K		L		M		N		O		P	
54	58					65		74	76	80					
Q		R		S		T		U		V		W		X	
104						120									
Y		Z		AA		AB		AC		AD		AE		AF	
152						168									
AG		AH		AI		AJ		AK		AL		AM		AN	
200						216									

- A. SERVICE NUMBER

B. NAME

C. SPECIAL KEY

D. RANK

E. DESIGNATOR

F. YEAR GROUP

G. SOURCE CODE

H. DATE OF RANK
- I. P-CODE

J. DATE OF BIRTH

K. SEX

L. SOCIAL SECURITY NUMBER

M. RETIREMENT YEAR

N. CURRICULUM NUMBER

O. EDUCATION LEVEL

P. DESIGNATOR CHANGE HISTORY
- Q. DATE OF ARRIVAL

R. DATE OF GRADUATION

S. OPR(GRADUATE)

T. OPR(TOTAL)

U. DEGREE AREA

V. COUNTRY/SERVICE

W. NUMBER OF DUTY STATIONS LISTED IN X,Y

X. CURRENT DUTY STATION

Y. PAST DUTY STATIONS-listed in reverse chronological order

images with the last 8 bytes of the second image discarded. This latter condition exists to preclude the splitting of a data element on any card input record and is expected to reduce input errors. Data within all fields is left justified and trailing blanks are allowed in the name and station name fields. Some fields within the record may be blank, and these are described in detail in Appendix A, "Users Manual."

B. REPORT DESIGN FOR RETRIEVAL OUTPUT

As with the concept of an Information Retrieval System as a whole, the output report format was designed with the user in mind, adhering to the following guide lines as closely as possible:

1. A report should consolidate all information relevant to its intended use.
2. A report should be readable, uncluttered and attractive.
3. A report should be easy to interpret by the user.
4. A report should be helpful to the user in locating needed entries rapidly, with a minimum of confusion and searching.

Since all of the information attributes contained in a logical record were determined to be necessary, a report was designed to display all 232 bytes of information on standard printer output which permits the printing of only 132 characters per line. (A sample output format is illustrated in figure 7. Appendix A.) With the limitation imposed on the horizontal structuring of report information, a decision was made to use a column oriented output structure for the information attributes, stacking related attributes for each entity in columns when feasible. For example in figure 7., refer to the third column from the right hand edge of the report which is a stacking of two information attributes: DEGR (degree level, e.g., PhD) and AREA (the area in which

the degree was obtained). Note that these attributes are separated by a slash (/) which for the purpose of this report implies stacked output. Two other forms of stacked attributes occur in columns one and four. Column one information attributes are formatted to correspond exactly to the column label appearing at the top of every page (those items in parenthesis appear only for selected entities for obvious reasons). Column four demonstrates the third and final method in which stacking is indicated: DESIG (current designator) appearing over (CHG), the latter referring to designator change history if applicable to an entity. The data printed for a designator change will appear in two stacked parenthesized expressions below the current designator; the first a four digit field indicating the year and month of the change, and the second indicates the old designator. Utilizing the foregoing methods of stacking, thirty information attributes can be efficiently contained in the fourteen column spread of the report output. The horizontal organization of the report as illustrated in figure 7 is basically column oriented with respect to individual or stacked information attributes. Even though abbreviations and pneumonics are freely used, each of the column headings are self explanatory and will pose no particular difficulty for report users (Curriculum Officers). In all cases, the headings appear directly over the data fields to which they refer and are printed at the top of each page of the report. Proceeding down the output report it is noted that double spaces separate the information printed (excluding any error notification) for individual entities in order to provide good visual effect and readability. Additionally, measures have been taken to ensure that page ejection will not occur once the printing of information for an entity has been

initiated, precluding the separation of related matter on adjacent pages. An additional feature of this report is the descriptive output for the information attributes RANK, DEGR, and AREA. In contrast with this output, refer to the Bureau output in figure 4 which contains these same information attributes buried as a subset in the second data field which is one continuous string of alphabetical and numeric characters. In this latter report the user would have to resort to tables or lists in order to decode the printed information. This is not the case with the SOIRS report. In certain cases, for U.S. Naval Officers only, reference to the Register of Officers might be required for designator or source code decoding; curriculum numbers are readily identified in the Naval Postgraduate School Catalog. An index to P-Codes is contained in current BUPERSINST 1210.13 series if this information is desired.

In the event an error is detected in the retrieval input data deck, the program will respond with an error message indicating that an error occurred in the data and that the data deck should be corrected and resubmitted. This error print-out should not appear on the report which is submitted to the report requester, but should be corrected by the individual conducting the file transaction. The events which could generate this error message are discussed in the sections "File Maintenance" and "Users Manual."

C. DESIGN OF FILES

There are two classes of entities included in the data base, namely those students currently attending the Naval Postgraduate School, and those who have attended in the past. This condition implies that a logical arrangement of two distinct files within the data base would

SAMPLE BUPURS OUTPUT

KRUEGER RUDOLPH VINCE	647814457643262J110063004C	S PG MONTEREY	401019	U TEXAS	63 64300
KUHLER JAMES JOHN	647709215386251J110041001C	S PG MONTEREY	390827	USNA	61 64100
KUHNENDALL HERBERT GREGT	6041244285241111100560011	WAR COL NEWPORT	310412	GWU	68 88700
LABAY DENNIS RAYMOND	648620304380147J131041004C	S PG MONTEREY	391225	MARQUETTE	61 65600
LABYAK PETER STEPHEN	650217038265950J1100620010	S PG MONTEREY	390919	USNA	62 64100
LATR DONALD JOHN	674454502446885J1320440380	S PG MONTEREY	410804	6700132565C61355CONCDIA MH-	63 64400
LAVE RUDER ALLAN	710227534464512110040004C	S PG MONTEREY	460821	U KANSAS	68 87700
LAWLESS LENNIE LARRY	641855516141930J1100500010	S PG MONTEREY	370716	USNA	50 64100
LAWRY ROBERT	688491267643708K1100650251	S PG MONTEREY	420105	U LOUISVIL	65E64400
LAWGILL JUSTIN EDWARD III	4476142516446191100450010	3 PG MONTEREY	420105	RENSELAER1259N85400	
LAWICOLA JOHN JAY	7317323233881311100400010	940208803PHQ NAVMATCONMD	DC230204	USNA	68 64200
LAWSON RONALD RAY	64381447742112111107570041	S PG MONTEREY	460711	AMERICAN U	62 76700
LASH FRANKLIN BRUCE	452595385347472J110062004C	S PG MONTEREY	320821	U MINNESOT	61 68200
LASSITER REX DON	695002429705419J1320450382	S PG MONTEREY	381001	64C61105	
LASSWELL JAMES RYAN	640240567523988J1100430010	S PG MONTEREY	390224	67101125670013955ARKANS SU	64 64300
LATTA WILLIAM ALLEN JR	578325450640222J1100440010	S PG MONTEREY	410925	USNA	63 64100
LAUTERMILCH PAUL ANTHONY JR	543031177201213H1100510011	S PG MONTEREY	420113	USNA	64 84100
LAWRENCE GREGORY EDSON JR	508017570268845H1310500311	980199202PDRG 20 GMDRUGH271111		GWU	64 84700
LAWSON HIRSH ALLEN	65651240445458J110042004C	INT STF JCS WASH280815	390330	AMERICAN U	69 76939
LEAHY JOHN FRANCIS III	64185553300023J1100600010	S PG MONTEREY	360525	U LOUISVIL	62 64400
LEARDI PAUL LOUIS	64952922345421J110041004C	S PG MONTEREY	380804	USNA	60 64100
LEARY RICHARD KENT	733891004440497L110548006C	S PG MONTEREY	450927	U SO CAROL	61 68100
LEBRON GEORGE ROYMER JR	65042855529355K140546004C	S PG MONTEREY	420718	BALPINGTON	67 64300
LEPROY FLOYD GARLAND	661500427108380J1400630201	S PG MONTEREY	300827	FRESNO S	65 67200
LEE RONALD ALVIN	6703192C3507437J140063006C	S PG MONTEREY	400427	U MISSISSI	62E6430C
LENNIX RICHARD JOHN	66247130284970J1100430011	S PG MONTEREY	380413	DUKE	62 64300
LESSMANN RONALD FREDERICK	671044558507513J140043006C	S PG MONTEREY	390704	USNA	63 64100
LETTINGTON THOMAS FLOYD	637637478383476J110040004C	S PG MONTEREY	371216	CAL POLY	62 60000
LEVIN HELEN RICKEL	650802355222444H310055009C	RIPERS PER5 N	280304M	IOWA STATE	60 67400
LEWIS JOHN HOWARD	6364520943029371110060004C	S PG MONTEREY	360816	AMERICAN U	65 86900
LIERFMAN JAY EDWARD	732844173368071L1310049004C	S PG MONTEREY	460408	MANHATTAN	59 64300
LITON SAMUEL JOSEPH JR	63065657952850013100500010	S PG MONTEREY	370814	U MICHIGAN	68 64600
LILLY DAVID EDWARD	648659510327635J1100440271	S PG MONTEREY	310531	USNA	59 64100
LINDER ISHAM WILHELM	485352224452740201310450010	S PG MONTEREY	240214	U WASHINGTON	63F63600
		7 CLEVEL		U CALIFORNIA	2461K44301

FIGURE 4

provide a reasonable file base for the system. The file was designed in this manner and the two classes of files are referred to as the Historical File, and the Current File. There are several advantages in maintaining these files as separate entities; the more obvious are listed as follows:

1. After initial creation of the Current File, and subsequent to quarterly updates, each Curriculum Officer should be presented with a current listing of all students under his cognizance. This would preclude, except in rare instances, any further requirement for information retrieval from the Current File during that quarter.

2. The retrieval of information will normally be mutually exclusive with respect to the Current and Historical Files (i.e., keyed to either the Current File or the Historical File but not both simultaneously). For this reason the two file structure would be more efficient because fewer records would have to be scanned in each case. For example, the Historical File presently contains 4876 records, and the Current File will contain approximately 1500 records. The advantage obtained in searching only 5000 or 1500 records as opposed to 6500 records is obvious, especially when searching for current entities.

1. File Organization

Due to the absence of any real data regarding utilization of the information retrieval process, the criteria upon which the file organization is based are:

1. The expected number of record additions to and deletions from the files;
2. The size of files,
3. The number of records processed during update runs on the file.

The quarterly update of the Current File will involve a minimum of one information attribute (OPR) per entity to be updated, yielding a File-Activity Ratio (number of records processed divided by the number of records scanned) of 100%.

Addition and deletion of records with respect to the Current File are to be batch processed with the quarterly update, and are expected to number between two-hundred and four-hundred records

(the total number of arriving and departing students). Based on this, the Current File could in no way be considered static, but instead would be moderately volatile, justifying the rewriting of the Current File each quarter.

It is important to point out at this time, that there is presently no interface with Bureau of Naval Personnel Files which would permit updates of information attributes in the Historical File; this is a temporary condition pending implementation of SOIRS and is further discussed in Section V. Therefore, the term update is used with reference to the Current File only. However, records are updated and added to the Historical File every quarter when the Current File is purged of departing students.

The foregoing characteristics of expected Current File activity (high File-Activity Ratio and moderate file volatility), and the sequential nature of information retrieved from both the Current and Historical files indicate that a sequential file organization (refer to LIST OF DEFINITIONS) would be the most efficient for SOIRS, and was therefore utilized.

Normally, it would be inappropriate to initiate the information retrieval process with the specific intent of searching for one particular entity. On the other hand it would be quite appropriate to key a search to one or several parameters which would yield a list of entities within the intersection of search parameters. The latter use is the one for which the file was designed (in response to user desires), and with highly flexible retrieval techniques, the probability of obtaining information on any particular entity is uniformly distributed over the

length of the entire file. With the extensive range of search parameters provided it is necessary to scan each record of the file. Consequently it is a process quite compatible with sequential file organization. The alternate file organization would be one designed for random processing (see LIST OF DEFINITIONS) with searches established through the use of hashing functions. This would be extremely inefficient and time consuming due to the complexity of key transformations and an increased number of logical operations necessary to acquire the desired degree of flexibility.

2. File Residence

The decision resulting in a sequential file organization yielded two alternative devices upon which the data files could be stored and manipulated: (1) IBM-2311 disk pack or, (2) magnetic tape. The criteria considered in choosing magnetic tape as the appropriate device are listed as follows:

1. Cost
2. File protection
3. File security
4. File compatibility
5. Device speed

Cost in this context refers to the cost of data residence on a particular storage device. The cost for magnetic tape is approximately .01¢ per bit and the disk cost is approximately .1¢ per bit.¹ This latter figure is deceiving since, for best operation,

¹ Professor M.L. Cotton during a course of instruction concerning Information Structures, March 1968.

the entire disk pack should be dedicated to the subject files, and for the cost of one disk pack, several magnetic tapes could be purchased (these tapes would exceed the disk storage capacity many times over).

File protection refers to the vulnerability of the storage media to inadvertent destruction during file maintenance routines or testing. Both tape and disk provide for the unique labeling of a file to discourage unauthorized use, however, of greater concern is the desirability to maintain processing control over the stored data.

This latter control is best illustrated by an example: Suppose that during a file maintenance or testing routine, which required a file to be read, the completion of the data control block was in error due to incorrect use of OPEN MACROS, DCB MACROS or Data Definition statements which would result in a write operation. This could generate processing which would destroy a file unintentionally. In the case of magnetic tape, positive control over a read/write operation can be exercised by removing (or replacing) the "WRITE RING" from (in) the tape reel. When the ring is out the tape cannot be written on; only by inserting the ring can a write operation occur. This should significantly decrease the potential vulnerability of a file and preclude a lengthy process of file re-creation. The disk storage media does not provide this feature of protection.

File security refers to the physical security of a storage media which might or might not contain sensitive information. Magnetic tape is highly portable, moderately durable, and small enough so that several reels could be stored with ease in a safe or small vault. If required, tape reels could be transported to a computer

center, remain under observation during a processing run, and upon completion of processing returned to the designated storage area. The disk pack although portable, is fragile and bulky by comparison; additionally, disk packs are not normally removed from the processing center.

File compatibility refers to a decision to use either magnetic tape or disk storage or both for different files. The Grandfather, Father, Son concept of file back-up is utilized for both Historical and Current files to permit recreation of any file which might be destroyed in error. The most practical method of storage for these generations of files is magnetic tape for reasons mentioned in preceding paragraphs. In order to promote compatibility in file devices and to minimize the data management problem for the programmers who will maintain the file, magnetic tape was used for all files. (Additionally, magnetic tape is a universal storage media which is compatible with most computer installations; this is not currently the case with disk files.)

Device speed in this case relates only to the transfer rates of the respective devices; IBM-2311 has a transfer rate of 156K bytes/second and magnetic tape devices have a transfer rate of 90K bytes/second. Clearly a distinct advantage in favor of disks; however, with the advent of multiprocessing with a variable number of tasks (MVT) presently available on the Naval Postgraduate School computer. This transfer rate was not considered significant since the additional input/output requirements expected with tape would not adversely affect the utilization of the central processing unit.

D. FILE MAINTENANCE

For the purpose of this discussion, File Maintenance will include a brief description of Historical File creation, file transactions involving input data which result in changes to files, and a proposed method for insuring the integrity of files.

1. Creation of Historical File

The following methodology describing the generation of a Historical File is applicable to the development of files which are based (whole or part) on data received from an outside activity, and would be appropriate in subsequent dealings with the Bureau of Naval Personnel in the event that SOIRS is implemented.

Extreme caution must be exercised in dealing with a tape received from outside activities since incorrect processing could result in the loss of tape data, requiring additional transactions, time in shipment, and computer time at both the sending and receiving activities. A request to BUPERS resulted in the receipt of a magnetic tape file which contained records on all past and present Navy students of the NPGS. The tape as requested was a labeled tape with a track density of 800 b.p.i. Upon receipt the tape label was verified by utilizing the IBM System Utility "IEBTPCH" to read and print the label. Subsequent to label verification the entire content of the tape was printed out, employing the same utility, providing visual assurance that the transmitted file contained desired information. Immediately thereafter a working copy of the BUREAU tape was generated, with appropriated blocking factors for economical buffer utilization,

through the use of the System Utility "IEBGENER." It was previously decided that SOIRS files should be arranged in alphabetical order (first 5 letters of the last name) for batch processing, and to additionally facilitate easy reading of the output report. In order to achieve this ordering, the System Procedure "SORT/MERGE" was utilized. The results of the foregoing processing yielded a sorted working copy of the Bureau file ready for processing with the SOIRS application program "BUPERS TRANSFER" (BUPSXFR). The BUPSXFR program essentially takes a record from this working copy, abstracts desired information attributes, performs any data conversions required, and places these attributes in a SOIRS logical record format. As the record for each entity is transformed, it is written onto a tape which, when completed, will be the Historical File. This processing program does not convert or place on the Historical File any Bureau record which does not reflect past attendance at the Naval Postgraduate School. Instead, the program causes the first 133 bytes of these rejected Bureau records to be printed out. Upon completion of processing, the Historical File has been generated and a listing of rejected records is printed out (see figure 4 for sample rejects).

2. File Transactions

There are two other SOIRS application programs which will be regularly used in file transactions. These are the CRDCHK1 and UPDATE programs, and are discussed in the following paragraphs.

a. Input Record Verification

Upon initial creation of the Current File and for subsequent insertion of logical records for new entities, the CRDCHK1 program will be utilized to insure the correctness of card input records

prior to placing the subject records on the Current File. The following attributes of each input record are checked, as appropriate, for correct numeric and alphabetical strings, incorrect use of blanks or blank fields and special characters, and for correct ranges of both numeric and alphabetical fields:

1. Name
2. Number of duty stations
3. Designator
4. Special Character (*)-required in column 37
5. School Start Date
6. Curriculum
7. Degree
8. Graduate and Total QPR
9. Rank
10. Service Number
11. Sex
12. Expected Graduation Date
13. Area
14. Social Security Number
15. Service and/or Country

Appendix A contains a sample CRDCHK1 output with a list identifying the types of errors which will be reported.

b. File Updates and Transfers

It was determined that all file updates and transfer transactions could best be accomplished using batch processing techniques on a quarterly basis. These transactions are:

1. Updating records in the Current File
2. Inserting new record in the Current File
3. Deletion of records from the Current File
4. Transferring the updated records of those students who have completed their tours at the Postgraduate School from the Current File to the Historical File.

The foregoing transactions are incorporated into one processing program, UPDATE, which performs these functions in a single processing run conducting updates and record dispositions as directed by input data cards. The update parameters and special character keys required in the input data are described in detail in Appendix A.

The Current File should at all times be maintained in alphabetical order (first five letters of last name), and since batch processing is the designated mode of file transacting when using UPDATE, the input data cards (less the record additions) should also be sorted in a like manner. An out of sequence data card will not be processed, but will generate an error message which identifies the card in error. The program will recover and proceed to process the next record. Other types of errors in input data are handled in a like manner. A sample listing of UPDATE output, which illustrates the four basic types of program response, is contained in Appendix A (figure 11). and is described as follows (the entities in the following examples may be located on the output listing by proceeding alphabetically down the name field):

1. The first transaction involves the updating of the entity "ACADEMY, JOE FRESHCAUGHT." The first line is a notation of the logical record prior to any changes, and the second is the updated record as it will appear in the new file. The transaction is identified by the suffix * UPDATE* appearing after the updated entry.

2. The second entry, "ARMYTYPE, ALPHONSE KNOTHEAD" has a suffix of * DELETE *, indicating that the record for this student has been deleted and will not appear on the new Historical or Current File.

3. The output for "FLEGLE, OOGLE EYE" has a suffix of * XFER *, which implies two conditions: (1) the record was transferred from the Current to the Historical File; and (2) prior to the transfer any required updates to the subject record were completed. This output is similar to that obtained with * UPDATE *, demonstrating the before and after conditions of the record.

4. The final transaction, "LUCKY, GO HAPPY", illustrates the last type of UPDATE output, and is identified by the trailing * NEW * in the one line output. This output constitutes the first 122 bytes of the new record. (Input data cards for new records are previously checked for accuracy by processing with the CRDCHK1 program).

The UPDATE printed output additionally becomes a useful medium for conducting spot checks on desired transactions.

3. Quarterly File Maintenance

The most important consideration in any information system structured on a dynamic data base is the protection of that data base from accidental loss or destruction. In the case of SOIRS, the two basic files, Historical and Current, should be maintained as described in following paragraphs in order to preclude the total loss of the data base, and to provide for reconstruction of both files to their most current form.

Subsequent to SOIRS implementation, the file content will consist of one copy each of the two basic files. Backup copies of each file should be generated as soon as possible by employing the System Utility "IEBGENER" (a back-up copy of the Historical file exists at the present time), yielding two copies of each file which are in turn sorted in alphabetical order. One each of the Historical and Current File copies will become working files, while the others are to be designated the "Current Father" and "Historical Father" files respectively. These fathers therefore are the backups for the working copies.

Quarterly, the UPDATE program is used to process record updates, deletions, additions, and transfers to the Historical File. All of these transactions are accomplished in a sequential manner which does not affect the working file, but instead performs the designated

operation and writes the new record on a new Current File. In the case of transfers, the record is written on a new Historical File. Upon completion of UPDATE processing, the new Current File contains new record additions, updated records for all entities which were resident on the Current Father and were not transferred to the Historical File, and does not contain records which were designated as DELETE from the Current Father. This new Current File is then sorted and becomes the Current Son, a new generation of current entity records. The UPDATE input data records should be retained for one quarter so that the Son could be regenerated from the Father, by another UPDATE processing, if required. The new Historical File is generated through UPDATE transfer of designated records from the Current Father to the new Historical File. After UPDATE processing, the Historical Father is copied onto the new Historical File (using IEBGENER with a disposition of MOD, KEEP). The new Historical File contains all records transferred from the Current Father and all records which reside on the Historical Father. Sorting of this new file yields the Historical Son; a new generation of historical entities. The creation of subsequent Current and Historical generations are carried out in the same manner as described above. The retention of files one generation old, and the retention of quarterly transition data for the UPDATE program provides for adequate protection and restoration of the data base in the event of destruction of the Current and/or Historical Sons.

IV. SOFTWARE DESIGN

The program documentation for each of the application programs is presented in Appendix B in the form of flow charts and program listings. Each line of instruction in the listings contains, in the comment field, a descriptive phrase concerning the logic of the operation, and constitutes a highly descriptive mode of documentation which will be extremely valuable to application programmer assigned responsibility for program maintenance. The following paragraphs will briefly describe in less detail certain characteristics of specific algorithms used in each of the problem programs.

A. BUPSXFR PROGRAM

This program was developed to:

1. Read the BUPERS logical records.
2. Test each entity for validity with respect to Historical File criteria.
3. Perform data conversion on certain attributes.
4. Transform required attributes to SOIRS logical record configuration.
5. Write appropriate records on the Historical File.
6. Print out a listing of those records which were not placed in the Historical File.

With the exception of 2,3, and 6, above, all programming techniques are elementary in execution and require no special knowledge on the part of the reader with regard to the BUREAU logical record. These exceptions (also employing only basic programming techniques) are listed and described as follows:

1. The criteria for entity placement into the Historical File is that a student must have attended the NPGS in the past. This condition is

ascertained by comparing the character constant "MONTEREY" with the college name attribute of the BUPERS record field "Education 1." An equal compare implies, that this record should be placed in the Historical File.

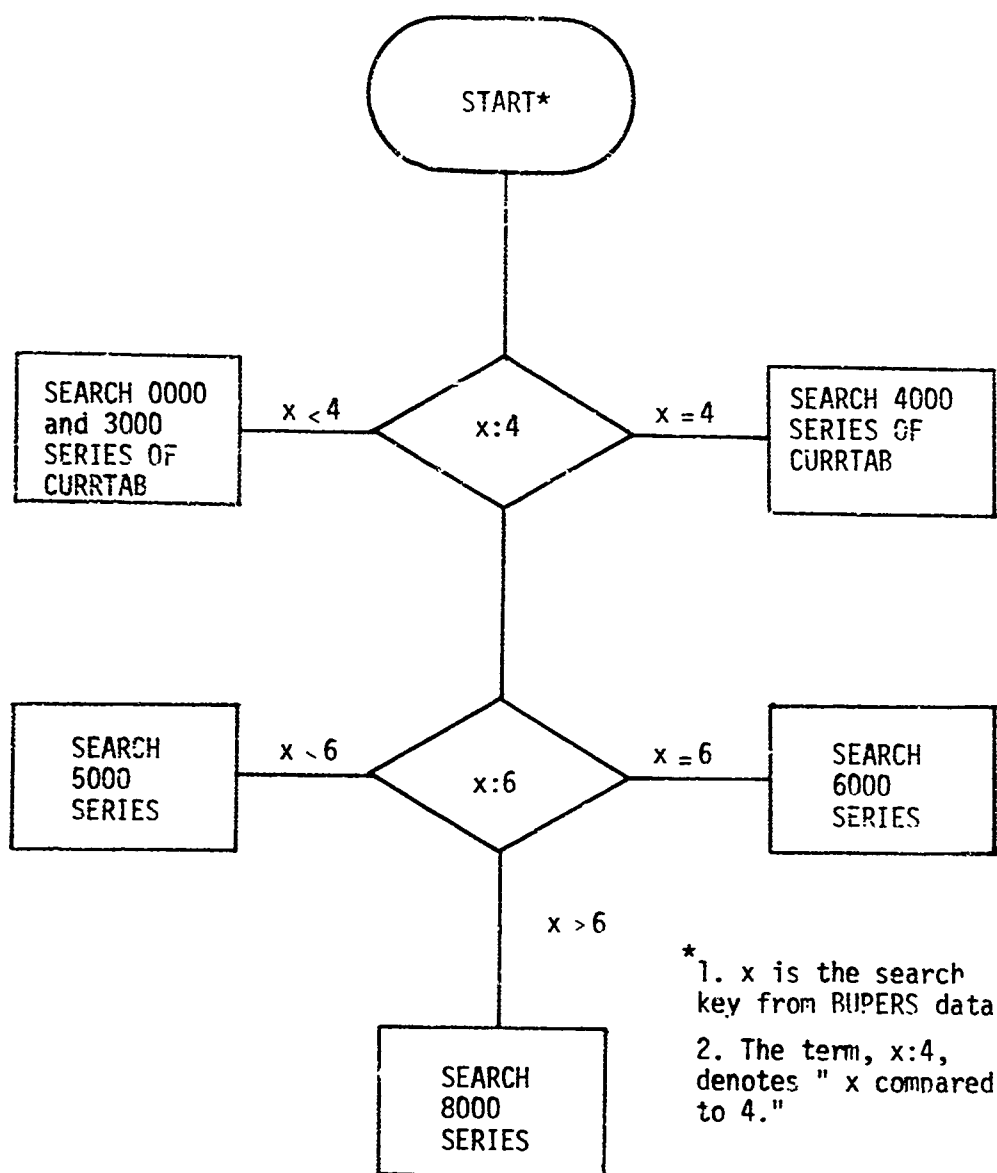
2. Upon receipt of the BUPERS File, it was found that information concerning course curriculum was not included in the file and information regarding degree area was encoded in accordance with the "Naval Officer Education, Language and Service Schools Code (NAVPERS 15824 series)." The latter being a rather extensive numeric listing of specialty codes which exceeded the requirements of SOIRS (These requirements are indicated as a list of twenty attributes in TABLED of the Information Retrieval Program, Appendix B.). In order to convert the Bureau data into meaningful SOIRS attributes, a table was constructed which mapped the four byte Bureau data concerning education onto curriculum numbers and degree area. This table is labeled as CURRTAB in the BUPSEXFR listing, Appendix B. All of the 79 attributes of CURRTAB are organized in the same manner. The first four numbers of each CURRTAB attribute indicates a Bureau education code. The next three numbers indicate the curriculum onto which the Bureau code is mapped, and the letter indicates the degree area, which is the final attribute in the range of the mapping. As an education code is read from the Bureau tape a binary search of CURRTAB is conducted for a match on the first digit of the code. When an equal compare occurs, the pertinent subset of CURRTAB is serially searched until a match on the four digit code is found; then the data is converted yielding a curriculum number and degree area for each entity. This binary search is relatively effective, requiring at the maximum a serial search of a 26 item list (from a total list consisting of 79 items) to obtain the correct entry in the table. This maximum occurs only when the last item in the 5000 series of CURRTAB is required (see figure 5).

3. If the criteria in item one above is not satisfied, or if the binary search of item 2 does not yield a correct response the program will reject the record and cause its contents to be printed on the output printer (i.e., the records printed are not included in the Historical File).

B. CRDCHK1 PROGRAM

This program was designed to verify the correctness of SOIRS input data cards (for new entities) destined for processing with the UPDATE program (the general program objectives have been previously described in FILE MAINTENANCE section III (D)(2)). The program specifically checks the following information attributes (from the SOIRS logical record) for the errors indicated:

FIGURE 5
BUPSEXFP. BINARY SEARCH OF TABLE CURRTAB



1. Name - The first byte is checked for a blank, and an equal compare results in an error. The first ten bytes of the name field are checked for correct alphabetical characters and if numeric or special characters are detected, an error exit occurs.

2. Duty Station Number - This byte is checked for a numeric value in the interval (0,8), any other character will result in an error condition. Up to three duty stations can be indicated on the first two data cards; if an entity had 3 or less duty stations and an incorrect number (greater than 3) of duty stations indicated, the program would automatically assign a duty station number of 3 to correct the error. During processing, a zero (0) is inserted in this attribute for each foreign entity to serve as a potential key in future transactions.

3. Designator - Each byte of this four-byte field is checked for numerics on the interval (0,9), any other characters result in an error exit.

4. Byte 37 of the input record is tested for a *, which must be present in order to convey condition codes to the UPDATE program, otherwise an error results.

5. School start (end) date, a three byte field, is checked for numerics on the interval (0,9), any other character will result in an error.

6. Curriculum Number - a three byte field indicating the student's curriculum is checked for numerics on the interval (0,9). Other characters result in errors.

7. Educational Level, Graduate and Total OPR, Service Number, and Social Security number are checked for valid numerics, on the interval (0,9), for each byte of the respective attribute.

8. Rank, and Degree Area are one byte fields tested for the alphabetical characters on the interval (A,T). Any other character results in an error condition.

9. Country/Service - A two byte attribute which indicates the branch of service for U.S. students, and indicates the country of a foreign student. The first byte must be numeric on the range (0,4) and the second byte numeric on the range (0,9). Both bytes collectively have a maximum permissible value of 41. Deviations from the foregoing result in error exits.

10. The attribute SEX is checked for one of the following valid characters F,M, or a blank. The latter two indicating "male." All other characters result in errors.

Errors generated from the above processes are descriptively printed out identifying not only the attribute name, but the character which caused the error. A complete description of all errors is contained in Appendix A.

1. CRDCHK1 Algorithm for Table Output Processing

The CRDCHK1 program contains five tables (C,D,G,L, and S) which assign meaningful output parameters to the record attributes which exist as one or two byte table keys in the logical record. For example, TABLE L of the CRDCHK1 program lists the ten possible levels of education which are considered in SOIRS. Any one of these levels may be indicated in the logical record image by inserting a number ranging from 0 to 9 in byte 101 of the record. When this byte is processed by CRDCHK1, this numeral must be transformed to a meaningful mnemonic or abbreviation which will appear in the output report. The manner by which this transformation occurs is described in the following example, using TABLE L for illustration.

address (b_{10})	TABLE L	Table key
0000	GSCH	0
0004	HSCH	1
0008	1YRC	2
0012	2YRC	3
0016	3YRC	4
0020	-PG-	5
0024	BACH	6
0028	CERT	7
0032	-MS-	8
0036	PHD-	9

Hexidecimal and binary formats are used for representing all characters internal to the machine, but for purposes of this example will be supplemented by base 10 notation.

The algorithm takes the hexadecimal representation of the Table Key and logically "ANDS" the Key with the hex constant "0F." This results in a pure binary number which is then multiplied by four (4). Adding this result to the address of the first attribute in TABLE L yields the address of the desired attribute. For purposes of illustration

let the table key be equal to 2_{10} ($F2_{16} = 11110010_2$). Submitting this value to the logical "AND" operation with $0F_{16} = 00001111_2$:

$$\begin{array}{r} 11110010_2 \\ 00001111_2 \\ \hline \text{result of "AND"} \quad 00000010_2 \end{array}$$

multiplication by 4 yields: 00001000_2

adding this value to the address of the first attribute of TABLE L:

$$\begin{array}{r} 00000000_2 \\ 00001000_2 \\ \hline 00001000_2 = 08_{10} \end{array}$$

where the address 08_{10} is pointing to the desired attribute 1YRC (one year of college)

In the example above, the address of the first attribute of TABLE L was assigned a value of 00 only for convenience in computation. During actual program execution, this address can be located at any authorized absolute address in core storage. The advantage of this method of table retrieval, is that the desired attribute is obtained in one table reference as opposed to a time consuming serial search of the table, potentially involving all attributes. The attributes of the four other tables in CRDCHK1 are obtained in a similar manner using the same basic algorithm.

C. UPDATE PROGRAM

The General Characteristics of the UPDATE Program are described in section III. D. 2.b; the basic functions of the program are reiterated as follows:

1. Add new records to the Current File

2. Update records in the Current File
3. Update and transfer records from the Current File to the Historical File.
4. Delete records.

The UPDATE Program permits updating of the following information attributes only:

1. Designator
2. Designator change history
3. Rank
4. Date of rank
5. Curriculum number
6. Degree area
7. Degree level
8. QPR (graduate and total)
9. Graduation date.

As each input data card is read, a test is performed to determine if the entity is a U.S. or Foreign Officer. A star (*) in column 1 indicates a Foreign Officer and the file is searched alphabetically until a match is encountered. If column 1 is blank a serial search on social security numbers is conducted until the desired U.S. Officer record is found. The following parameters on input data cards indicate record disposition:

1. A percent sign (%) indicates delete the subject record from the file.
2. A dollar sign (\$) indicates the record is to be undated and transferred to the historical file.
3. A star (*) indicates a new record addition.
4. The absence of any of the aforementioned special characters results in an update to the subject record and retention in the Current File.

Additional requirements for input data preparation are discussed in detail in Appendix A.

D. FILESRCH PROGRAM

The Information Retrieval Program (FILESRCH) is at the hierarchical apex of all SOIRS programs and is the essence of the system. FILESRCH is a management tool and will respond to information inquiries keyed to the following information attributes subject to the search-key parameters (equal (=), not equal (#), and an inclusive range (%) of two numbers (2 digits each)).

<u>Possible Search Attributes</u>	<u>Possible Search-Key Parameters</u>
Social Security Number (A)	=
Sex (B)	=
P-Code (C)	=, #
Designator (D)	=, #
Service/Country (E)	=, #
Year of Birth (F)	=, #, %
Year Group (G)	=, #, %
Rank (H)	=
Curriculum (I)	=
Degree Area (J)	=
Degree Level (K)	=
OPR (Graduate) (L)	%
OPR (Total) (M)	%
Graduation Year (N)	=, %

(The alphabetical characters in parentheses after each attribute, A-N, are keys used to refer to respective attributes during input data preparation.)

The result of FILESRCH program execution will yield a subset of the file searched corresponding to the logical "AND" of each search attribute (together with its search-key parameter) with all other search attributes desired. The following two elementary examples will illustrate this property.

1. A file search keyed to the following inputs: B=F, F%36,45, I=360, L%25,30 will yield a list of Female students with a year of birth ranging from 1936 to 1945, who were enrolled in curriculum 360

and possessed a graduate QPR ranging from 2.50 to 3.00. The entities for which all of the foregoing attributes are true would appear on the SOIRS output report.

2. A file search keyed to: A=549426211, D=1100, F#40 would yield at maximum one entity; the student who has a social security number equal to 549-42-6211, a designator equal to 1100 and who was not born in the year 1940. (Obviously, since a social security number is unique, the output list could contain no more than one student, and then if and only if all search attributes are true). This case is an example of a search which should not be initiated since the information on one individual is easily obtained from a quarterly list which should be made available to each Curriculum Officer.

During a given retrieval run it is possible to key on all search parameters (A-N), however, as indicated in example 2 above, certain combinations could result in absurd output listings. The limitations on searches are: (1) each search attribute can be used only once for each program run; and (2) the range search-key (%) can be used only twice during a program run. This latter restriction arises from a programming requirement to use only four bits to set condition codes. Two bits are required for = and #, and each % requires one additional bit; in this case a maximum of two range checks are allowed in order to contain all testing criteria in NIBBLE configuration. Even with the above restriction, there are 6.24×10^5 possible independent searches available resulting in a rather high degree of search flexibility.

1. The Basic FILESRCH Algorithms

FILESRCH will process desired information in accordance with the type of Search-Key Parameters (=, #, %) specified in the input data. To keep track of seven possible combinations of search-keys, a one byte storage area called FLAGMASK (FM) is established to record and subsequently direct the FILESRCH processing with respect to search-key requirements. FLAGMASK will have one of the following configurations after all input data has been processed:

FLAGMASKSearch-Key Processing Required

111²
110²
101
100
011
010
001

%,#,
%,#
%,=
%
#,

=

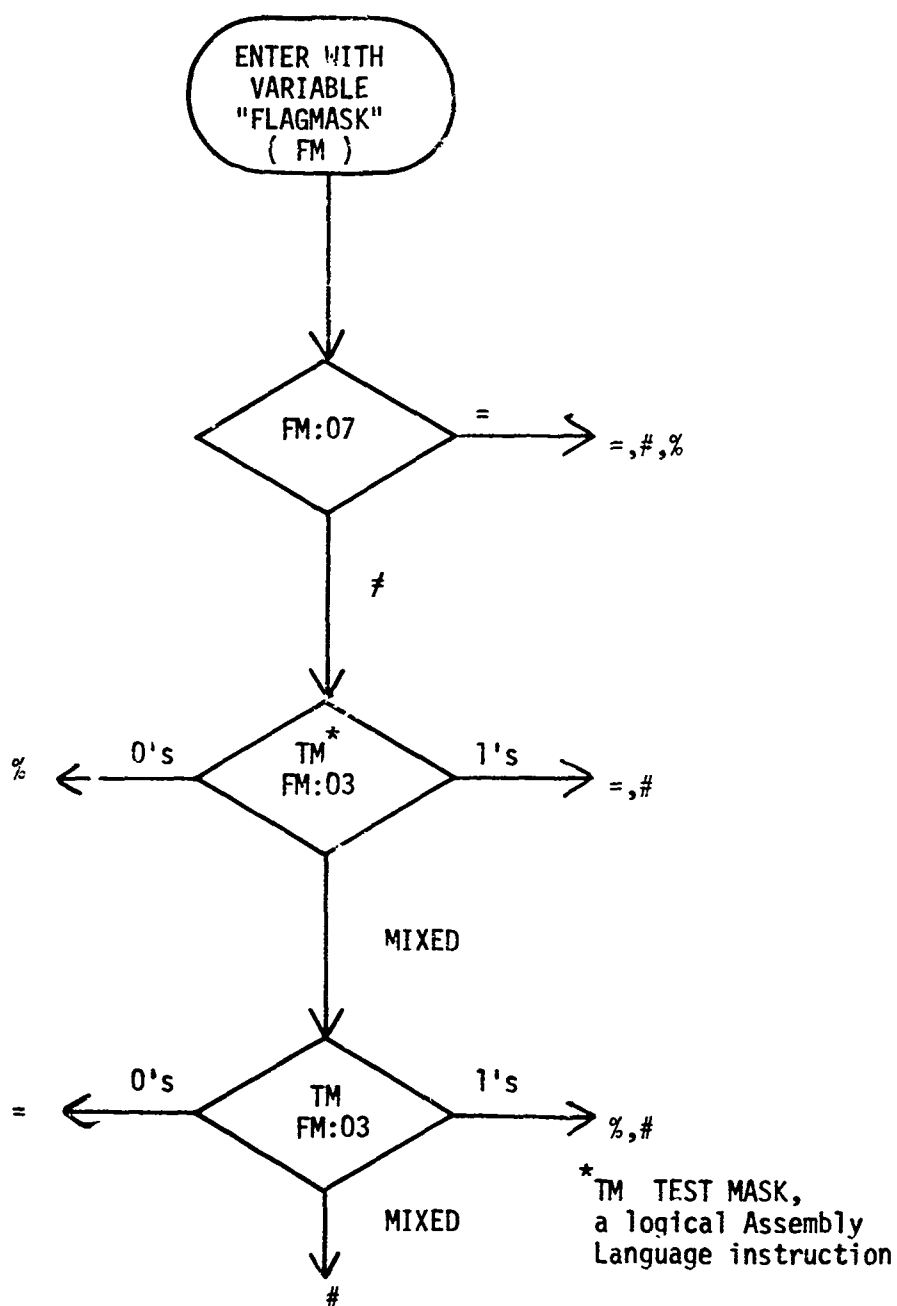
The algorithm which directs correct search-key processing is illustrated in figure 6 (normal use of Immediate Compare and Test Under Mask instructions are required).

Depending on the FLAGMASK setting, input data is compared with each record of the file and appropriate entities are listed in the output. The method of introducing desired information attributes to the processing program will be discussed in the following paragraphs.

All of the possible search attributes are contained in bytes 37-103 of SOIRS Logical Record. This results in a 67 byte field which might contain one or more sub-fields of interest. In order to extract the required attributes, a masking scheme was developed for each key-search parameter (%,=,#). For purpose of discussion, only the equal key parameter will be referenced.

Three 67 byte storage areas are utilized to produce a field which will be used in a serial search of the file. For purposes of illustration these three areas are to be called EOMASK1, EOMASK2, and SAVEQ. When an input data card indicates a key-search parameter of equal (=), the search attribute is placed in EOMASK1 (initially all zeros) in a position corresponding to the attribute's location in bytes 37-103 of the logical record. One's are inserted into the corresponding bytes of EOMASK2 (initially all zeros). This process

FIGURE 6
 DECISION ALGORITHM for DETERMINATION
 of SEARCH-KEY PARAMETERS



continues until all input data cards are processed. At this time bytes 37-103 of the first file record are placed in SAVEQ. Immediately a logical "AND" between SAVEQ and EQMASK2 is executed resulting in destruction of undesired attributes in SAVEQ. EQMASK1 and SAVEQ are then compared character by character. If an equal compare results, the subject record is desired and printed. For example, let the following be indicated on a data card as the only desired information: B = F, where F = 11000110₁₆. The user desires a list of all female students. The hexadecimal equivalent of "F" would be placed in byte 28 of EQMASK1 and ones would be inserted in byte 28 of EQMASK2:

EQMASK1		0	←	zeros		11000110		zeros →	66	
						byte 28				
EQMASK2		0	←	zeros		11111111		zeros →	66	
						byte 28				

Reading the first record from the file and moving the 67 bytes of possible search fields to SAVEQ, would yield in the case of a male entity:

SAVEQ		0	←	other data		11010100		other data →	66	
						byte 28				

"ANDING" SAVEQ with EQMASK2 yields:

SAVEQ		0	←	zeros		11010100		zeros →	66	
						byte 28				

A compare of EQMASK1 and SAVEQ will result in the rejection of the current record. This process would be continued until the file was exhausted. The output report will contain all female students.

Based on the configuration of FLAGMASK, the other six methods of processing would be conducted in a similar manner.

A complete description of FILESRCH input data preparation, together with several examples, is included in Appendix A.

There is one final characteristic of the FILESRCH program which will be valuable to users. This is the inherent ability of the program to dump the entire file without resorting to the System Utility IEBTPCH. This may be accomplished by using one of several search attributes and specifying a search-key parameter of unequal (#) with data which could not be included in any record. For example, search the file for entities who were not born in the year 99, or those who have a social security number not equal to 999-99-9999. The results would obviously yield a listing of the entire file.

E. CHOOSING A PROGRAMMING LANGUAGE

The most important criteria leading to the choice of Assembly Language (AL) as the programming language for SOIRS, was the desire to achieve maximum compatibility with the Bureau of Naval Personnel which employs AL for Officer Files. The only alternative languages considered were those from the List Processing Set (LISP, PL-1, etc.), and those were inferior to AL with respect to efficiency of machine execution. Additionally, AL is more commonly used (locally) than any list processing language, and it is therefore expected that program maintenance will be carried out more effectively.

V. CONCLUSION AND RECOMMENDATIONS

This paper has discussed an Information Retrieval System which was developed to fulfill certain specific requirements for a particular group of managers. As noted in section II, the desires of these managers resulted in designing a narrow scope retrieval system. However, adequate space has been provided in the logical record to expand the number of information attributes per entity if desired, and there is sufficient modularity within all program packages to permit the insertion of additional processing routines with a minimum of effort. In other words, the present system could easily be expanded into a broad base retrieval system which could satisfy the information requirements of a larger set of managers. Recommendations for future modification and expansion of SOIRS are discussed in the following paragraphs.

1. SOIRS should be implemented as soon as possible, thereby eliminating an additional Historical File creation based on BUREAU data which does not contain all of the desired information attributes. Curriculum Officers could effect implementation using the following steps:

- a. The collection of data on all students presently attending NPGS.
- b. Key punching the subject data.
- c. Processing the record additions onto the Current File with the CRDCHK1 and UPDATE programs.

The above steps would be carried out in accordance with section III. D. 2 and Appendix A.

2. Correspondence with the Bureau of Naval Personnel (PERS-N) should be initiated to develop a transaction file which would be used to update the following attributes on an annual basis:

- a. Rank
- b. Designator
- c. Date of rank
- d. P-code
- e. Retirement year
- f. Designator change history
- g. Current duty station

After the format of this transaction file has been agreed upon, a program should be written to carry out the subject update. The most complex area of the program would be the insertion of a new duty station, which is easily accomplished because all duty station name fields are of equal length. By moving the logical record into a core position of length 248 bytes and shifting all station names right 16 bytes, the first name field is available for a new current duty station. The right most 16 bytes of this core storage area should not be moved to the updated file.

3. After implementation of SOIRS, user surveys should be conducted to determine if changes to the system are required. Currently 112 bytes of the logical record are reserved for past duty stations, however these areas are blank since the data was not available in ADP form at the BUREAU. Perhaps a requirement for only two past duty stations could be justified, releasing 80 bytes of the record for additional data of a personnel accounting nature such as section assignment and transfer between curricula. The former being an additional

administrative assist for all Curriculum Officers, and the latter being particularly useful to the Engineering Science Curriculum Officer since a considerable number of Engineering Science students do transfer to other curricula areas.

4. Consideration should be given to increasing the size of the SOIRS logical record and inserting data which would support information attributes of the following nature:

- a. Refresher course requirements
- b. Military housing requirements
- c. Text book requirements
- d. Classroom scheduling
- e. Flight scheduling

These and other attributes could be useful in preparing resource load analysis. For example, the availability of projected class schedules, curricula schedules, and instructor assignments (all derived from a data base) would enhance certain areas of faculty administration, and promote the coordinated development of each separate curriculum.

Additional application programs or subroutines would be required to cope with the expanded system. If future changes to SOIRS are not required or are not feasible, the potential of the data base should be exploited through the use of special programs which would replace certain present manual processing chores (e.g., the quarterly computation of NPR is currently accomplished by use of EAM methods).

5. Investigation should be conducted, after sufficient file usage data is accumulated, to determine the feasibility of placing the system into a real time environment. This could be accomplished by placing application programs on the 2314 magnetic disk drives in load

module form, and supplying input data at an on-line terminal. If real time is desired, the most current copies of the files should also be disk resident.

6. Although far removed from actual machine language, list processing languages offer certain advantages over AL. Specifically, these languages by their nature are most effective in an information retrieval application, and should perhaps be considered for SOIRS at a future time.

7. There are two minor problem areas of SOIRS which require additional consideration subsequent to implementation:

a. It is not uncommon to encounter a student who has attended NPGS on two separate occasions. This situation will result in a double entry in the Historical File after completion of a student's second period of attendance at NPGS. This is an undesirable condition and can be corrected by establishing an additional education data field within the logical record. If this is done, it might be advantageous to record the undergraduate education of all officers in this additional field.

b. The printed output, resulting from execution of the UPDATE program, currently contains no descriptive labeling. This condition should be corrected if UPDATE is to be used by an individual other than a relatively experienced programmer.

APPENDIX A

NPGS STUDENT OFFICER INFORMATION RETRIEVAL SYSTEM (SOIRS) USER'S MANUAL

This manual provides a basic introduction to the use of the NPGS Student Officer Information Retrieval System (SOIRS). It is written for those individuals who have not had previous automated data retrieval system experience.

Examples are given throughout the manual to emphasize the basic features of SOIRS. A complete and detailed description of SOIRS may be found in a thesis on a Proposed Student Officer Retrieval System by LCDR's R. L. HENRY and L. L. MASSA.

The information contained in this manual should enable a reader to use SOIRS effectively within a short period of time.

TABLE OF CONTENTS

PART

1.	INTRODUCTION -----	56
2.	SYSTEM ORGANIZATION -----	57
3.	PUNCHED CARD RECORD INPUT FORMAT -----	59
4.	FILE UPDATE AND TRANSFER PROCEDURES -----	73
5.	FILE SEARCH TECHNIQUES -----	79

LIST OF FIGURES

FIGURE

7.	INPUT CARD VERIFICATION PROGRAM OUTPUT LISTING WITH ERRORS (TWO PAGES)	65
8.	INPUT CARD VERIFICATION PROGRAM OUTPUT LISTING WITHOUT ERRORS	67
9.	INSTRUCTIONS TO INCOMING STUDENTS FOR COMPLETING REQUIRED "SOIRS" DATA FORM (FIVE PAGES)	68
10.	CURRENT FILE UPDATE AND TRANSFER INPUT DATA CARD FORM	77
11.	CURRENT FILE UPDATE AND TRANSFER PRINTED OUTPUT LISTING	78
12.	CURRENT FILE SEARCH PRINTED OUTPUT LISTING NUMBER 1	84
13. f	CURRENT FILE SEARCH PRINTED OUTPUT LISTING NUMBER 2	85
14.	CURRENT FILE SEARCH PRINTED OUTPUT LISTING NUMBER 3	86
15.	CURRENT FILE SEARCH PRINTED OUTPUT LISTING NUMBER 4	87
16.	HISTORICAL FILE SEARCH PRINTED OUTPUT LISTING NUMBER 1	88
17.	HISTORICAL FILE SEARCH PRINTED OUTPUT LISTING NUMBER 2	89

PART 1. INTRODUCTION

The NPGS Student Officer Information Retrieval System (SOIRS) consists of four basic parts:

1. Card deck input error checking.
2. BUPERS Historical File initialization.
3. Current File update, and record transfer to Historical File.
4. Current and Historical File information retrieval.

Of these parts, only parts one, three, and four, will be of pertinent interest to the user.

This manual is organized into sections as follows:

1. INTRODUCTION
2. SYSTEM ORGANIZATION
3. PUNCHED CARD RECORD INPUT FORMAT
4. FILE UPDATE AND TRANSFER PROCEDURES
5. FILE SEARCH TECHNIQUES

Sections three, four, and five contain detailed instructions with examples of input/output data including error message's, their possible cause, and correction procedures.

This manual contains the information needed by curriculum officers to implement and use SOIRS effectively but, it is emphasized that it does not provide the information needed for SOIRS file maintenance. The Application Programmer assigned to maintain SOIRS will need the additional information contained in a thesis on a Proposed Student Officer Retrieval System by LCDR's R. L. HENRY and L. L. MASSA on file construction, record format and Job Control Language.

PART 2. SYSTEM ORGANIZATION

SOIRS data is contained on two magnetic tape reels designated the "Current File" and "Historical File". The Current File contains the following data items on all officer students presently attached to the Naval Postgraduate School at Monterey.

1. Name
2. Rank
3. File Number
4. Branch of Service or Foreign Country
5. Sex
6. "IGEP" Indicator
7. Social Security Number
8. Date of Birth
9. Current Designator and Previous Designator with Date of Change if applicable.
10. Date of Rank
11. Original Source Code
12. Year Group
13. Year first eligible to Retire
14. P-Code
15. Quarter and Year started school
16. Expected Quarter and Year of Graduation
17. School Curriculum
18. Expected Level of Education to be achieved (M.S., B.S., etc.)
19. Expected Area in which Degree will be awarded (E.E., O.R., etc.)
20. Current, Graduate and Total "OPR"
21. Chronological list of last eight Duty Stations.

The information contained in data items 3, 6, 9, 11, 12, 13, 14, and 21 is not included for foreign and non-Navy officers and data item 7 is additionally omitted for foreign officers. The Historical File contains identical information on all officer students who have

completed a course of study at NPGS Monterey with actual and final data substituted for the expected and current values contained in the Current File.

Current File initialization is accomplished by locally prepared punched three card deck input for each presently enrolled student officer. Initialization of the Historical File requires a magnetic tape input from the Bureau of Naval Personnel and therefore will only include data on U.S.N. officers currently on active duty. It is expected that when SOIRS is actually initiated by PGS Monterey the Historical File input from BuPers will be augmented with local records available in the Registrar's office to include previous officer students for foreign countries and other U.S. military services.

Current File updating (additions, deletions, and data record changes) and transfer of individual student officer records to the Historical File is accomplished locally using punched card input. Historical File updating can be accomplished from BuPers tape records for applicable data items (Rank, Duty Stations, etc.). It should be noted that all locally prepared three card individual student data decks are error checked by a SOIRS program prior to inclusion on a magnetic tape file.

Current or Historical File data may be retrieved by searching on various combinations of fourteen data record parameters. The search combinations are made up of equal, not-equal and inclusive numeric range comparisons. Each record extracted as a result of a Current or Historical File search is printed and the printed output contains all the information available on the specified file for the selected individual record.

All SOIRS data manipulation programs are coded in Assembly Language and are disk resident in an assembled load module form.

PART 3. PUNCHED CARD RECORD INPUT FORMAT

This part of the SOIRS user manual describes the procedures for making up input data card decks and eliminating data input errors by verifying sixteen key data fields. All three card input data decks, used for Current File initialization or as a Current File record addition with the UPDATE program must have the following individual student officer information keypunched in the columns indicated. A * appearing after the card column numbers signifies that the field is to be left blank for all foreign and non-Navy officers. A ** signifies that the field is to be left blank for foreign officers only.

CARD 1

<u>Column(s)</u>	<u>Information</u>
1 - 6 *	File Number
7 - 36	Last Name(b)first name(b)middle name(b)(b).....(b) (b) = one space, truncate letters in excess of thirty characters.
37	* : If record is to be added to Current Fi
38	Rank Code: from Table "A", Figure 9
39 - 42 *	Designator
43 - 44 *	Year Group
45 *	Year Group subdivision (blank if N.A.)
46 - 48 *	Original Source Code: As listed in the current "Officer Register"
49 - 54	Date of Rank Col. 49, 50 - Year 51, 52 - Month 53, 54 - Day
55 - 58 *	P-Code (blank if N.A.)

CARD 1 (continued)

<u>Column(s)</u>	<u>Information</u>
59 - 64	Date of Birth (Year; Month; Day)
65	Sex (blank if male)
66 - 74 **	Social Security Number
75 - 76 *	Year first eligible to Retire (blank if N.A.)
77 - 79	Curriculum Number
80	Expected level of education at graduation code: From Table "B", Figure 9

CARD 2

<u>Column(s)</u>	<u>Information</u>
1 - 4 *	Date of Designator change (blank if N.A.) (Year; Month)
5 - 8 *	Old Designator (blank if N.A.)
9	School start quarter
10 - 11	School start year
12	Expected graduation quarter
13 - 14	Expected graduation year
15 - 17	Current graduate OPR
18 - 20	Current total OPR
21	Expected Degree at graduation code: From Table "C", Figure 9
22 - 23	Branch of Service or Foreign Country: From Table "D", Figure 9
24	Number of past Duty Stations (including NPGS, Monterey) Note: U.S.N. Officers <u>must</u> have a minimum of 1 and a maximum of 8 in this column. All non-USN students <u>must</u> have a "0" in this column.
25 - 40 *	S(b)PG(b)Monterey(b)(b)(b) (b) = one blank space

CARD 2 (continued)

<u>Column(s)</u>	<u>Information</u>
41 - 56 *	Previous Duty Station (blank if N.A.)
57 - 72 *	Second previous Duty Station (blank if N.A.)
73 - 80	Blank

CARD 3

<u>Column(s)</u>	<u>Information</u>
1 - 16 *	Third previous Duty Station (blank if N.A.)
17 - 32 *	Fourth previous Duty Station (blank if N.A.)
33 - 48 *	Fifth previous Duty Station (blank if N.A.)
49 - 64 *	Sixth previous Duty Station (blank if N.A.)
65 - 80 *	Seventh previous Duty Station (blank if N.A.)

Prior to inclusion of new student officer data decks on Current File magnetic tape they must be processed by the SOIRS input card deck verification program to eliminate errors in key data record fields. Sixteen input card data fields are checked for errors. If any errors are detected an error message is printed that specifies the incorrect data field and prints the non-valid character(s) found. A list of sixteen possible error messages, with the indicators * or ** used in lieu of non-valid characters, is shown below:

1. WARNING CARD DECK DELETED DUE TO ERRONEOUS DATA.
2. INCORRECT AREA CODE = * .
3. INCORRECT CJRR CODE = * .
4. INCORRECT CTY CODE = ** .
5. INCORRECT DEGR CODE = * .
6. INCORRECT DESG CODE = * .
7. INCORRECT DUTY CODE = * .
8. INCORRECT EDTE CODE = * .
9. INCORRECT GOPR CODE = * .

10. INCORRECT RANK CODE = * .
11. INCORRECT SDTE CODE = * .
12. INCORRECT SRNR CODE = * .
13. INCORRECT SSNR CODE = * .
14. INCORRECT SVC CODE = ** .
15. INCORRECT TOPR CODE = * .
16. NEW RECORD IND MISSING .

A description of the type of error checking performed, on each of the sixteen data fields tested, with an example of each resulting error message from the sample output container in Figure 7, is shown below:

1. NAME FIELD: First ten columns checked for non-blank and alphabetic characters. Result of error(s) = card deck rejected and no printed output except the statement, " WARNING CARD DECK DELETED DUE TO ERRONEOUS DATA ". Shown in Figure 7.
2. DEGREE AREA CODE: Checked for valid alphabetic characters " A " through " T ". Result of error = error message printed as follows, " INCORRECT AREA CODE = X ". Shown in Figure 7.
3. CURRICULUM: Checked for non-numeric characters (including blanks). Result of error(s) = incorrect curriculum number and the following error message, " INCORRECT CURR CODE = A ", printed as shown in Figure 7.
4. FOREIGN COUNTRY CODE: Checked for valid numeric codes " 20 " through " 41 ". Result of error = error message printed as follows, " INCORRECT CTY CODE = 51 ". Shown in Figure 7.
5. EDUCATIONAL LEVEL CODE: Checked for non-numeric characters (including blanks). Result of error = error message printed as follows, " INCORRECT DEGR CODE = J ". Shown in Figure 7.
6. DESIGNATOR: Checked for non-numeric characters (including blanks). Result of error(s) = incorrect Designator and the following error message, " INCORRECT DESG CODE = / ", printed as shown in Figure 7.
7. NUMBER OF PAST DUTY STATIONS: Checked for non-numeric characters (including a blank) and maximum numeric value of eight. Result of error = only three duty stations and the following error message, " INCORRECT DUTY CODE = 9 ", printed as shown in Figure 7.
8. GRADUATION DATE: Checked for non-numeric characters (including blanks). Result of error(s) = incorrect Graduation quarter and year and the following error message, " INCORRECT EDTE CODE = / ", printed as shown in Figure 7.
9. GRADUATE "OPR": Checked for non-numeric characters (including blanks). Result of error(s) = incorrect Graduate OPR and the following error message, " INCORRECT GOPR CODE = B ", printed as shown in Figure 7.

10. RANK CODE: Checked for valid alphabetic characters " A " through " S ". Result of error = error message printed as follows, " INCORRECT RANK CODE = & ". Shown in Figure 7.

11. SCHOOL START DATE: Checked for non-numeric characters (including blanks). Result of error(s) = incorrect school start quarter and year and the following error message, " INCORRECT SDTE CODE + L ", printed as shown in Figure 7.

12. SERVICE NUMBER: Checked for non-numeric characters (including blanks). Result of error(s) = incorrect Service Number and the following error message, " INCORRECT SRNR CODE = I ", printed as shown in Figure 7.

13. SOCIAL SECURITY NUMBER: Checked for non-numeric characters (including blanks). Result of error(s) = incorrect Social Security Number and the following error message, " INCORRECT SSNR CODE = U ", printed as shown in Figure 7.

14. BRANCH OF SERVICE CODE: Checked for the seven printed valid numeric codes 00, 01, 02, 10, 11, 12, 13. Result of error = error message printed as follows, " INCORRECT SVC CODE = 04 ". Shown in Figure 7.

15. TOTAL OPR: Checked for non-numeric characters (including blanks). Result of error(s) = incorrect Total OPR and the following error messages, " INCORRECT TOPR CODE = I ". Shown in Figure 7.

16. NEW DATA DECK INDICATOR: The following statement will be printed if indicator is not present. " NEW RECORD IND MISSING ". Shown in Figure 7.

Figure 7 is sample output from the input card verification program and contains examples of each error message and statement described above. Additional entries are listed with multiple error messages because error messages will be generated for all errors found in each data card deck.

Figure 8, illustrates the result of correcting the errors contained in Figure 7 and re-verifying the input card decks. It is emphasized that an error free output as shown in this figure is mandatory before the input card decks can be incorporated in the Current File.

Figure 9 is a recommended hand-out for incoming student officers to use when providing curriculum officers with the data needed for initialization and updating SOIRS Current File. The hand-out contains

all required tables with complete and specific instructions for filling out the attached form which can be easily used by keypunching personnel in preparing the individual officer's three card data input deck.

INPUT CARD VERIFICATION PROGRAM OUTPUT LISTING WITH ERRORS

NAME	FILE NR	SVC	CITY	ST	ZIP	DATE	TIME	YR	MO	DAY	HR	MIN	SEC	AREA	DEGR	CUR	PG	STAS
ACADEMY JPS FRESHMOUNT	1	0945367	41122	01-0	60	03	1-68	MS	380	MAIS	221	USMA	3 PG MONTEREY					
AIRFORCE CHARLY FLYTBIRD	0	69469012	361010	651111	2-54	610	DACH	200	078	USMA	3 PG MONTEREY							
ALPES MAXIMUM MOST	0	631892123	321206	1100	66122	01-0	58	79	8801	MS	278	3 PG MONTEREY						
ARMYTYPE ALPHONSE KNOTHEAD	0	69468012	361010	451111	3-65	610	DACH	125	139	USMA	3 PG MONTEREY							
CIVIL SERVICE GENDYTHAR	0	269092123	390309	1400	69022	07-0	64	03	03	USMA	3 PG MONTEREY							
DOMINION WAY DOWN GO	0	234567891	490707	1300	69040	01-0	68	03	03	USMA	3 PG MONTEREY							
FLAGG COOLE EYE	0	123456789	390809	1510	671212	21-1	60	03	03	USMA	3 PG MONTEREY							
GEORGE LEROY LOOP	0	531320123	390309	1377	661215	07-0	61	89	8601	CERT	180	3 PG MONTEREY						
HATCH MATTER	0	69469012	361010	451111	3-65	610	DACH	125	139	USMA	3 PG MONTEREY							

WARNING CARD DECK DELETED DUE TO ERRONEOUS DATA

STAS: 001, 002, 003, 004, 005, 006, 007, 008, 009, 010, 011, 012, 013, 014, 015, 016, 017, 018, 019, 020, 021, 022, 023, 024, 025, 026, 027, 028, 029, 030, 031, 032, 033, 034, 035, 036, 037, 038, 039, 040, 041, 042, 043, 044, 045, 046, 047, 048, 049, 050, 051, 052, 053, 054, 055, 056, 057, 058, 059, 060, 061, 062, 063, 064, 065, 066, 067, 068, 069, 070, 071, 072, 073, 074, 075, 076, 077, 078, 079, 080, 081, 082, 083, 084, 085, 086, 087, 088, 089, 090, 091, 092, 093, 094, 095, 096, 097, 098, 099, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000.

FIGURE 7

[illegible]

FIGURE 8

67

FIGURE 9

**INSTRUCTIONS TO INCOMING STUDENTS FOR COMPLETING NPGS
STUDENT OFFICER INFORMATION RETRIEVAL SYSTEMS (SOIRS) DATA FORM**

INCOMING STUDENT OFFICER
CURRICULUM OFFICE DATA SHEET

The attached data sheet is divided into three IBM card images. The data sheet is to be filled in exactly as instructed below. Do not use lower case letters or substitute blanks for zeros. Each data field block must be completely filled in unless specifically stated otherwise. Foreign officers should not fill in items marked with * or **, non-Navy U.S. officers should not fill in items marked with *.

CARD 1

<u>Block(s)</u>	<u>Required Data</u>
1 - 6 *	File Number
7 - 36	Last name (b) first name (b) middle name (b)(b)...(b) (b) indicates blank spaces Omit portion of name that exceeds thirty letters.
38	Rank Code: From Table "A".
39 - 42 *	Designator
43 - 44 *	Year Group
45 *	Year Group Subdivision (blank if N.A.).
46 - 48 *	Original Source Code: As listed in "Officers Register".
49 - 54	Date of Rank (Year, Month, Day).
55 - 58 *	P-Code (blank if N.A.).
65	Sex: Insert "F" if Female, Male officers leave blank.
66 - 74 **	Social Security Number
75 - 76 **	Year first eligible to Retire (blank if N.A.).
77 - 79	Curriculum Number
80	Expected Level of Education at Graduation Code: From Table "B".

CARD 2

<u>Block(s)</u>		<u>Required Data</u>
1 -	*	Designator Change Data: Year and Month (blank if N.A.).
5 - 8	*	Old Designator (blank if N.A.).
9 - 11		School Start Date: Quarter and Year.
12 - 14		Expected Graduation Date: Quarter and Year.
15 - 17		Current Graduate QPR; 000 if none established.
18 - 20		Current Total QPR: 000 if none established.
21		Expected Degree at Graduation Code: From Table "C".
22 - 23		Branch of Service or Foreign Country Code: From Table "D".
24		Number of Duty Stations listed below. Note: USN officers <u>must</u> have a minimum of one and a maximum of eight in <u>this</u> block. All non-USN officer students <u>must</u> have a "0" in this column.
41 - 56	*	Previous Duty Station (blank if N.A.).
57 - 72	*	Second previous Duty Station (blank if N.A.).

CARD 3

<u>Block(s)</u>		<u>Required Data</u>
1 - 16	*	Third previous Duty Station (blank if N.A.)
17 - 32	*	Fourth previous Duty Station (blank if N.A.)
33 - 48	*	Fifth previous Duty Station (blank if N.A.)
49 - 64	*	Sixth previous Duty Station (blank if N.A.)
65 - 80	*	Seventh previous Duty Station (blank if N.A.)

TABLES

"A"	"B"	"C"	"D"
FADM.....A	LS THN HS.....0	NONE.....A	USN.....00
ADM.....B	HIGH SCHL.....1	A.E.....B	USN(IGEP).....01
VADM.....C	LS 2Y COL.....2	AERO ENG.....C	USNR.....02
RAUM(U).....D	2 YRS COL.....3	ENG SCI.....D	USA.....10
RAUM(L).....E	3 YRS COL.....4	INT REL.....E	USCG.....11
COMO.....F	PG NO DGR.....5	E.E.....F	USMC.....12
CAPT.....G	BACHELOR.....6	ELEC ENG.....G	USAF.....13
CDR.....H	CERTIFICATE.....7	METEOROLOGY.....H	BRAZIL.....20
LCDR.....I	MASTERS.....8	OCEANOGRAPHY.....I	CANADA.....21
LT.....J	DOCTORS.....9	O.R.....J	CEYLON.....22
LTJG.....K		COMP MGT.....K	CHILE.....23
ENS.....L		COMP SCI.....L	CHINA.....24
CIVILIAN.....M		MGT.....M	COLUMBIA.....25
2LT.....N		M.E.....N	EQUADOR.....26
1LT.....O		MECH ENG.....O	GERMANY.....27
MAJ.....P		MATH.....P	INDIA.....28
LCO.....Q		PHYSICS.....Q	IRAN.....29
COL.....R		CHEM.....R	ISRAEL.....30
GEN.....S		MAT SCI.....S	KOREA.....31
		C.E.....T	NORWAY.....32
			PERU.....33
			PHILIPPINES.....34
			PORTUGAL.....35
			THAILAND.....36
			TURKEY.....37
			URUGUAY.....38
			VENEZUELA.....39
			VIETNAM.....40
			OTHER.....41

[illegible][illegible][illegible][illegible][illegible][illegible]

PART 4. FILE UPDATE AND TRANSFER PROCEDURE

This part of the SOIRS user manual describes the procedures for updating the Current File (including additions and deletions) and transferring departed students records from the Current to the Historical File.

SOIRS file updating is accomplished by the Current File Update and Historical File Transfer Program (UPDATE) in one step. This program adds, deletes, and changes Current File records in addition to transferring records to the Historical File (records may be updated during the transfer process). The following student officer record data items can be changed using the UPDATE program:

1. Designator (including date of change)
2. Rank
3. Date of Rank
4. Curriculum
5. Degree Area
6. Educational Level
7. Graduate and Total QPR
8. Graduation Date (Quarter and Year)

The punched card deck format for records to be added to the Current File is described in detail in Part 3. Record deletion, change, and transfer requires a single punched input card which is described below. Any data item that is not to be changed may be left blank. The * and NAME fields for foreign officers and Social Security Number field for all other students must be filled in because these fields are used to identify the desired Current File record. Items marked with a * apply to foreign officers only and items marked with ** apply to U.S.N.

officers only.

1. RECORD DELETION CARD

<u>Column(s)</u>	<u>Information</u>
1 *	*: Foreign Officer Indicator
2 - 29 *	Name: Exactly as it appears in Current File tape records
1 - 19	Name: First nineteen characters only
20 - 28	Social Security Number
29 - 79	Blank
80	?: Record Deletion Indicator

2. RECORD UPDATE AND TRANSFER CARDS

<u>Column(s)</u>	<u>Information</u>
1 *	*: Foreign Officer Indicator
2 - 29 *	Name: Exactly as it appears in Current File record
1 - 19	Name: First nineteen characters only
20 - 28	Social Security Number
30 - 33 **	New Designator
34 - 39 **	Date of Designator change: Year and Month
40	Rank Code: From Table "A", Figure 9
41 - 46	Date of Rank: Year, Month and Day
50 - 52	Curriculum
54	Degree Area Code: From Table "C", Figure 9
56	Educational Level: From Table "B", Figure 9
60 - 62	Graduate QPR
63 - 65	Total QPR
70 - 72	Graduation Date: Quarter and Year
80	?: Record transfer indicator, leave blank if update only is desired

Examples of each type of UPDATE input card are contained in Figure 10 and a standard coding sheet should be used to provide keypunch information. It should be noted that all possible data changes are shown in the examples of Figure 10 but they may be left blank if a particular data item is unchanged.

The delete, update and transfer data input cards used in the UPDATE program must be in alphabetical order and precede the three card data input decks that are to be added to the Current File (these do not have to be in alphabetical order), although each three card deck must be in correct order.

In addition to effecting the required Current and Historical File magnetic tape record changes the UPDATE program provides printed output to verify each data record affected. If a delete, update, or transfer data input card is not in correct alphabetical sequence or the * and NAME fields for foreign officers and Social Security Number field for all other students is incorrect, an error message will be printed for that input card as follows: " PROGRAM WILL NOT RUN. DATA UPDATE CARD FOR (student name) OUT OF SEQUENCE OR INCORRECT." If one or more error messages occur during execution of the UPDATE program the data cards containing errors must be corrected and the program re-executed for all input data cards. For successful execution of the UPDATE program no error messages may occur.

Figure 11 is a sample output listing from the UPDATE program containing several examples of record addition, deletion, updating and transfer. Record addition and deletion produces a single line of printed output containing the first one hundred and thirty-three (133) characters of the affected record with the appropriate indicator,

"* NEW *" or "*DELETE*" appearing at the right edge of the printed line. Record updating and transfer produces two lines of single spaced printed output. The first line contains the first one hundred and thirty-three (133) characters of the original record. The second line contains the same data but includes all changes effected and the appropriate indicator "* UPDATE*" or "* XFER *" at the right edge of the printed line.

[illegible][illegible][illegible][illegible][illegible]

PART 5. FILE SEARCH TECHNIQUES

This part of the SOIRS user manual describes the procedures for extracting required records from the Current or Historical File. Current and Historical File record retrieval is effected with one to twenty, free-form, punched input data cards containing valid combinations (each key used must be followed by at least one space) of fourteen search parameter keys. The Application Programmer must be told which file is to be searched as the keys are common to both the Current and Historical Files. These types of searches, EQUAL, NON-EQUAL or GREATER-THAN/LESS-THAN, may be conducted either singly or in combination. Each search compares all individual records in the specified file with the data specified on the input data card(s) and extracts only those records that are respectively equal, not-equal, or within the specified numeric range. Permissible record data field search parameters and their keys are listed below:

<u>DATA FIELD SEARCH PARAMETER</u>	<u>KEY</u>	<u>PERMISSIBLE SEARCH TYPE(S)</u>
1. Social Security Number	A	Equal
2. Sex	B	Equal
3. P-Code	C	Equal Not-equal
4. Designator	D	Equal Not-equal
5. Branch of Service or Foreign Country Code (from Table "D", Fig.9)	E	Equal Not-equal

NOTE: A secondary key of "N" for all USN officers, "F" for all Foreign officers and "U" for all non-Navy U.S. officers may be used in addition to the primary key. (E=N, E#E, E=U, etc.)

<u>DATA FIELD SEARCH PARAMETER</u>	<u>KEY</u>	<u>PERMISSIBLE SEARCH TYPE(S)</u>
6. Year of Birth	F	Equal Not-equal Greater-than/Less-than
7. Year Group	G	Equal Not-equal Greater-than/Less-than
8. Rank Code (from Table "A", Fig. 9)	H	Equal
9. Curriculum	I	Equal
10. Degree Area (from Table "C", Fig. 9)	J	Equal
11. Educational Level	K	Equal
12. Graduate QPR (To nearest 10th)	L	Equal
13. Total OPR (To nearest 10th)	M	Equal
14. Graduation Year	N	Equal Greater-than/Less-than

The character " = " follows the search parameter key when an " EQUAL " search is desired. An equal search will test the specified data field of each individual record contained in the designated file (Current or Historical) for an exact match with the character(s) following the equal search indicator and print the complete record if a match is found. For example, if a list of all previously graduated students who were enrolled in curriculum 367 was wanted, the Historical File would be searched using a data card containing I = 367 as the search parameter key.

The character " # " follows the search parameter key when a " NOT-EQUAL " search is desired. A not-equal search will test the specified data field of each individual record contained in the designated field

for an exact match with the character(s) following the not-equal search indicator and print the complete record if they do not match. For example, if a list of all USNR students presently enrolled in NPGS was wanted, the Current File would be searched using a data card containing E#02 as the search parameter key.

The character " %" follows the search parameter key when a " GREATER-THAN/LESS-THAN " search is desired (a comma must separate the numeric range values). A greater-than/less-than search will test the specified data field of each individual record contained in the designated file for a numerical value greater-than or equal to the second and largest range value. Any records found within the specified range would be retrieved from the magnetic tape file and printed. For example, if a list of all currently enrolled students with a graduate OPR of 2.0 through 2.5 was wanted, the Current File would be searched using a data card containing L%20,25 as the search parameter key.

Any combination of permissible search parameter keys may be used with the following two restrictions:

1. Only two greater-than/less-than search keys may be used at one time. For example, if a list of all students born between 1956 and 1960, who received a masters degree, did not have a 1100 designation and received a degree in the area of management was wanted, the Historical File would be searched using data card(s) containing the following search parameter keys:

F%34,40

N%56,60

K=8

D#1100

J=M

The input data card image needed to conduct the above search could be as follows;

.....F%34,40.....N%56,60.....K=8.....D#1100.....J=M.....

The periods denote blank spaces.

2. A maximum of twenty search parameter keys may be used in any one file search. The search parameter keys are free-form as mentioned previously and can occur anywhere on the input data card(s) as long as one or more blank spaces separate each search key.

The error message, " ERROR IN INPUT CARDS * PROGRAM WILL NOT RUN * RECHECK INPUT CARDS AND RESUBMIT ", will occur if more than two greater-than/less-than search keys are used, more than a total of twenty search keys are used and when non-valid search key characters are used.

The possible combinations of search parameter keys are too numerous to list and it is felt by the authors that the record retrieval flexibility provided by SOIRS will enable the user to perform any search required.

Printed output contains all individual record data as listed in PART 2 for each record retrieved. Figures 12 through 17 are illustrations of sample output resulting from the following Historical and Current File searches:

1. FIGURE 12 is a partial output listing of a sample Current File search for designators not-equal to 9999, which will result in a complete printout of the file as there is no such designator. The input data card would contain: D#9999.

2. FIGURE 13 is output from a sample Current File when searched for USNR officers who do not have a P-Code of 1510. The input data card(s) would contain: C#1510.

3. FIGURE 14 is output from a sample Current File when searched for all 1100 designated officers with a graduate OPR of 2.0 through 3.0 from year groups 1955 through 1965. The input data card(s) would contain: D=1100 L%20,30 G%55,65.

4. FIGURE 15 is a partial output listing from a sample Current File when searched for Navy officers (USN and USNR) who have a graduate OPR of 1.7 through 3.0 and total OPR of 2.0 through 3.0. The input data card(s) would contain: E-N L%17,30 M%20,30.

5. FIGURE 16 is the complete output from the Historical File when searched for past graduates who received PhD's. The input data card would contain: K=9.

6. FIGURE 17 is a partial output listing from the Historical File when searched for 1100 designated past officer graduates who received a Masters degree, were enrolled in curriculum 360 and graduated in 1960 through 1965. The input data card(s) would contain: K=8 D=1100 N%60,65 J=360.

The curriculum officer, when initiating a file search, need only specify the parameters he desires the file to be searched for, and the type of search desired (equal, not-equal or greater-than/less-than); to the Application Programmer responsible for SOIRS. The Application Programmer will insure the input data cards contain the required search parameter keys and are keypunched correctly.

FIGURE 13

FIGURE 15

HISTORICAL TITLE SEARCH PRINTED OUTPUT LISTING NUMBER 1

NAME, FILE NO., SVC(CITY), LOE, IF, REFERENCE, IF, IGEPI	SOCIAL NUMBER	BIRTH DATE	DESIG YR-ED	RANK YR-ED	SOURCE CODE	YR GP	IST YR	P CODE	NPGR STZED	SVR LUM	DEGR AREA	OPR STZ	PAST-DUTY, STAS
AMERSON, JAMES BRUCE CDR, 540434, USN	568353625	331016	1510		04-0	56		8106	-66	610	PHD A.E.		MISLSEN PT MUGU
ARTH, ROGER GRAYSON CDR, 568851, USN	381267495	290729	1310		01-0	52		8108	-67	530	PHV		VP 9
BUSH, THOMAS OLIVER LT, 530823, USN	441359461	380714	1709 {1107} {1102}		04-0	60			-68	521	PHV		DEF ATOM SUPPAGY
FEATHERSTON, FRANK HUNTER CDR, 532795, USN	224523892	290309	1510		01-0	50			-63	530	PHV		NASC AIR 01
GALL, CAINE, MIGUEL CDR, 491369, USN	504183312	290501	1510 {1106} {1100}		06-0	53			-65	530	PHV		NASC AIR 03
HARRIS, JACK, REDVER CDR, 514936, USN	367320053	301017	1510		04-0	53		8205	-63	590	PHD E.E.		NADC NAF JHNSVL
HART, DAVID, AUSTIN LCDR, 630664, USN	425686942	380224	1510 {1100} {1130}		06-0	60			-66	380	PHV		RVAM 120
MCFFMANN, JOHN, MELVILLE LCDR, 620079, USN	178264430	341205	1510 {1106} {1130}		03-0	58			-66	530	PHD CHEM		NAVWEAPEN CHILK
MILL, STEPHEN, FLOYD CDR, 530122, USN	132282225	370428	1400 {1106} {1100}		04-0	58			-65	590	PHD E.E.		NELECLABEN SDGO
MUCSON, RALPH, EDWARD LT, 630847, USN	559500334	361121	1510 {1100} {1130}		06-0	60			-66	530	PHD E.E.		VO 1
JAUNES, STEPHEN JR CDR, 532548, USN	562209287	250630	1510 {1106} {1100}		22-1	54			-62	590	PHD E.E.		NSA FT MEADE MD
KIMBLE, CHARLES, DONALD LCDR, 592322, USN	523467922	371012	1310 {1106} {1100}		04-0	59		8301	-68	530	PH CHE		CVA 34 ORISKANY
LEVIN, RICHARD, ROBERT LCDR, 530891, USN	073285793	341110	1510 {1106} {1130}		04-0	57			-66	530	PHV		NASPASYSACT LOSA
MILLER, ALOYSIUS, RUDOLPH LCDR, 579915, USN	146282455	360526	1400 {1106} {1100}		04-0	58			-65	570	PHD E.E.		NSVD CHARLESTN
PHHELPS, PHARO, ALFRED CDR, 533099, USN	561305700	280824	5100		01-0	50			-63	380	PHV		OICC NFECC SPAIN
POPE, JOHN, WILLIAM RIPPON JR CDR, 509604, USN	516442126	310515	1510 {1106} {1130}		07-0	54			-66	590	PHD E.E.		NAV SORCOMDHC DC
STURM, EDWARD, JOHN LT, 530885, USN	296303136	370610	1510 {1106} {1130}		04-0	60		8106	-68	610	PHD A.E.		NASC AIR 051
TUCKER, JAMES, EARL LCDR, 561119, USN	514206265	320609	1510 {1106} {1130}		04-0	56			-66	530	PHV		MISLSEN PT MUGU

FIGURE 16

NAME
 RANK, FL NR, SVC/CITY (S)
 IC-P, IE, LE, AL, E, I, IE, IC, EP, I
 AILES, P, ER, ER, ER
 L, P, R, 613310, USN
 RALPH IN, EDWIN MCLEEN
 LCDR, 24332, USN
 MERTY JAMES ROBERT III
 LCDR, 613316, USN
 RPNSTIN PAUL ALBERT
 LCDR, 464723, USN
 BRANCH ALVIN DEON
 CDR, 464245, USN
 COLLINS FERDINAND IGLEHART JR
 CDR, 344391, USN
 CONWAY JAMES MCNARNEY
 LCDR, 599988, USN
 CRANE JOHN WILLOUGHBY JR
 CAPT, 484053, USN
 CRAWFORD WILLIAM TRAVIS
 CDR, 584402, USN
 CUMMINGS PAUL ZACH II
 LCDR, 624443, USN
 DAVIS CHARLES EDWARD
 LCDR, 634227, USN
 GARDNER GREGORY LELAND
 LCDR, 599938, USN
 GREENWALD WILLIAM TAYLOR JR
 LCDR, 599993, USN
 HAGER CHARLES FRANKLIN
 CDR, 586780, USN
 HALL JOHN VALENTINE
 CDR, 528507, USN
 HODGSKISS WILLIAM LAWRENCE
 LCDR, 544236, USN
 HOWE JOHN EDWARD
 LCDR, 601496, USN
 HUGHES WAYNE PHILIP JR
 CDR, 594984, USN
 HUGHES THOMAS JOSEPH JR
 CAPT, 472573, USN
 JOHNSON JOHN MARK INTOSH
 LCDR, 570081, USN
 KEANE JAMES PHILIP
 CDR, 595003, USN
 KERNER JOSEPH JOHN JR
 LCDR, 596017, USN

SPECIAL
 SECURITY
 NUMBER
 224925011
 397327398
 179201450
 013265192
 224521335
 1363226601
 34266704
 420222070
 410642013
 07428719
 109306058
 375408074
 228424011
 184240050
 711544472
 516347647
 55482116
 339248803
 064130099
 482304270
 019222502
 257242216

BIRTH
 DATE
 YR-M-D
 391024
 361019
 350711
 340726
 280624
 320226
 320201
 250613
 310607
 360501
 371103
 330420
 340121
 321218
 321224
 330628
 340614
 300930
 261014
 311207
 260114
 330501

DESIG
 ICHQ1
 1100
 1100
 1100
 1100
 1100
 1100
 1100
 1100
 1100
 1100
 1100
 1100
 1100
 1100
 1100
 1100
 1100
 1100
 1100
 1100

RANK
 DATE
 YR-M-D
 01-0 57
 01-0 58
 01-0 57
 04-0 57
 01-0 50
 01-0 54
 01-0 55
 01-0 46
 01-0 54
 01-0 58
 02-0 60
 01-1 55
 01-0 55
 06-0 54
 04-0 54
 04-0 56
 06-0 57
 01-0 52
 21-1 46
 01-0 53
 01-0 52
 01-0 55

ORG
 CODE
 P
 9501
 8501
 8501
 8501
 8501
 8501
 8501
 8501
 8501
 8501
 8501
 8501
 8501
 8501
 8501
 8501
 8501
 8501
 8501
 8501

DEGR
 AREA
 MS
 MS
 MS
 MS
 MS
 MS
 MS
 MS
 MS
 MS
 MS
 MS
 MS
 MS
 MS
 MS
 MS
 MS
 MS

CUR
 PIC
 LUB
 360
 360
 360
 360
 360
 360
 360
 360
 360
 360
 360
 360
 360
 360
 360
 360
 360
 360
 360

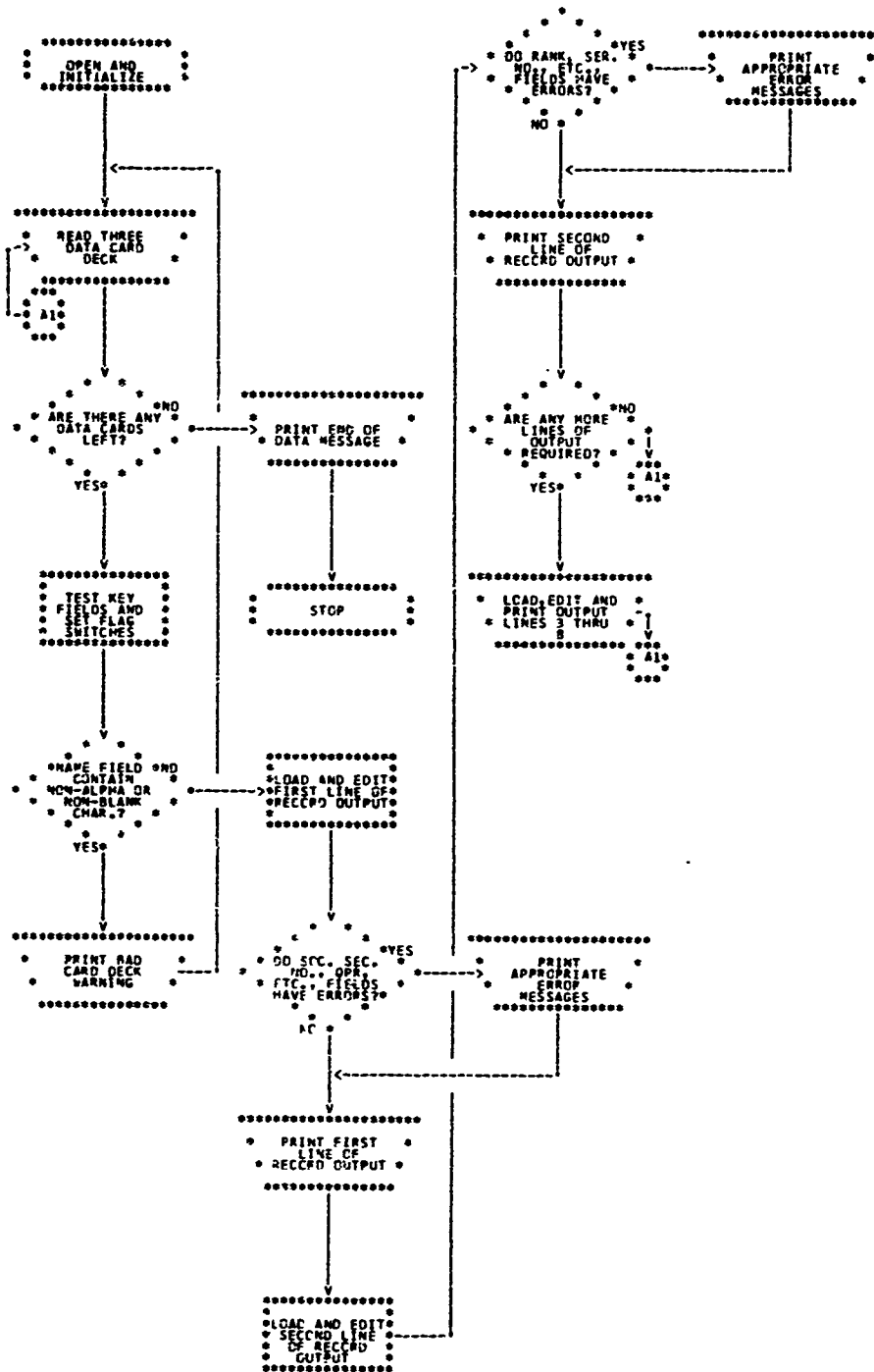
MPGS
 QTR-VP
 SIZE/END
 -63
 -64
 -65
 -64
 -63
 -61
 -63
 -60
 -61
 -64
 -65
 -63
 -64
 -64
 -63
 -65
 -65
 -64
 -62
 -64
 -60

PAST DUTY STAS
 DELOIS HAMBERG
 NPTU IDAMC FALLS
 SS 567 GUDGEON
 FLTCOMPROGCEMLT
 DDG 6 BARNEY
 OPNAV
 SS 350 DOGFISH
 COM PHIR GR 2
 DELOIS KRELOCH
 OPNAV
 NAVAL ACAD
 SUPERS PERS M
 OSD
 NOSC OPD 06
 OPNAV
 COMSERVPAC
 COM MSTs
 DO 948 MURTON
 AO 54 CHIKASALIA
 MAR COL NEWPORT
 SSBN 634 BLUE CR
 SS 365 HAPONEAD

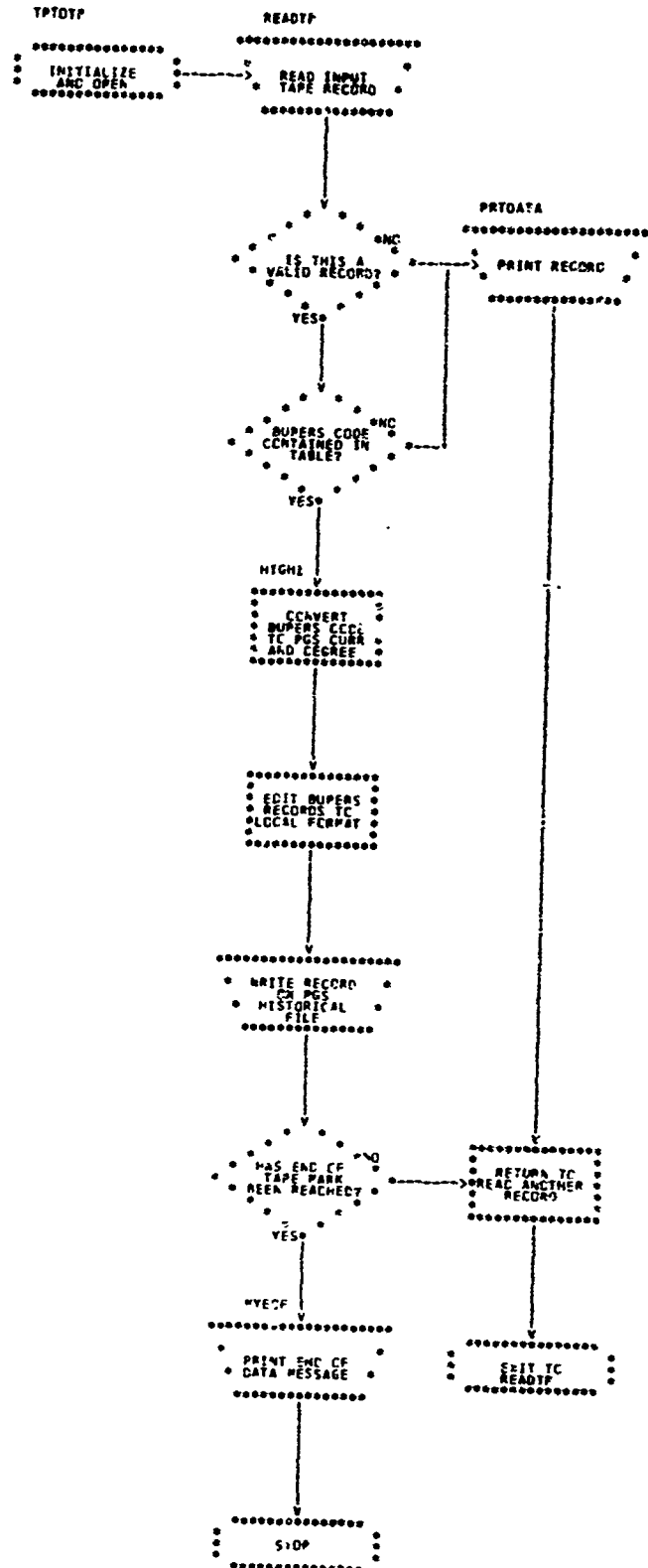
FIGURE 17

*** APPENDIX B -- FLOWCHART NUMBER 1 ***
DATA CARD DECK VERIFICATION PROGRAM

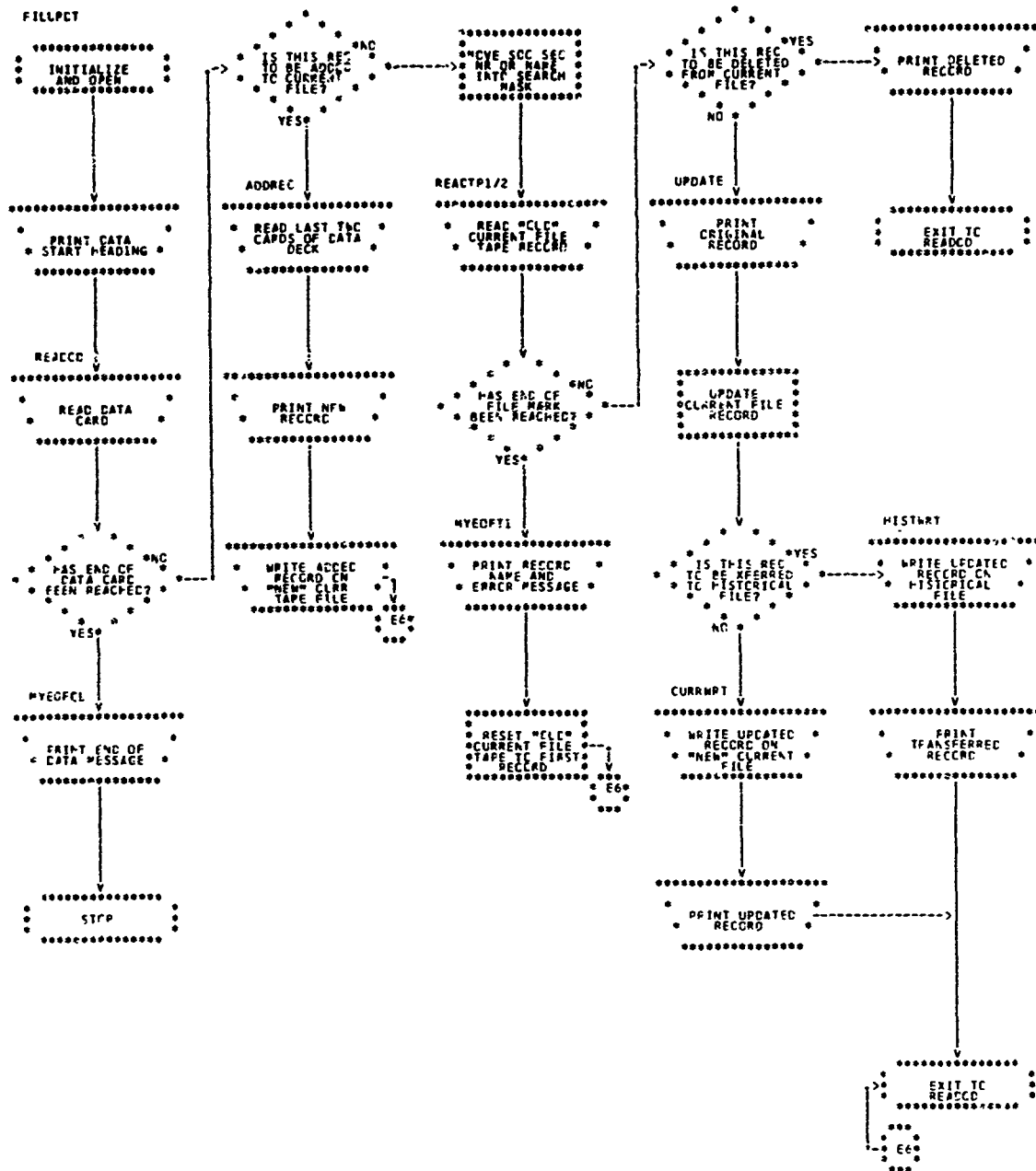
1



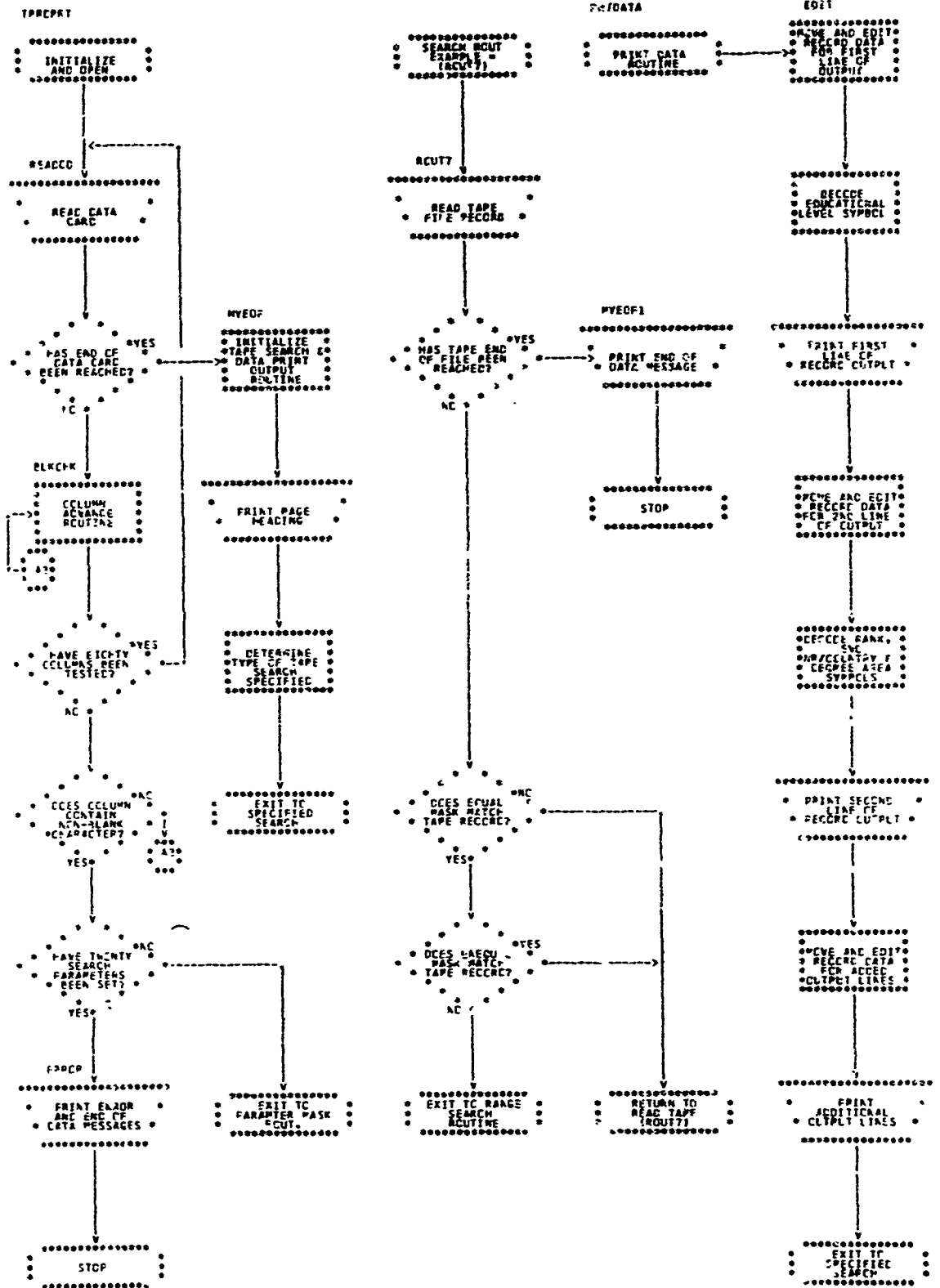
*** APPENDIX B -- FLOWCHART NUMBER 2 ***
SUPERS TAPE DATA TRANSFER PROGRAM



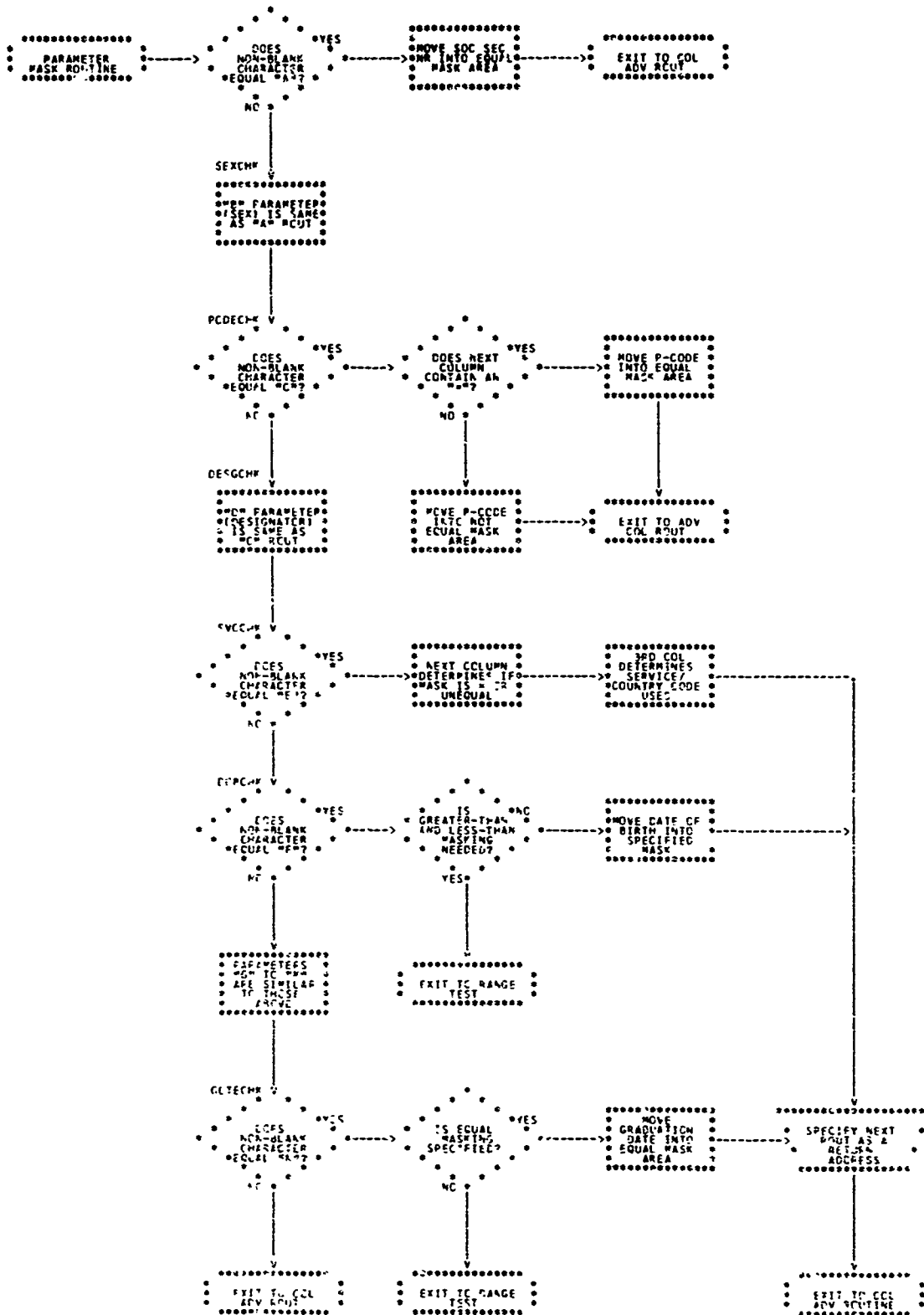
*** APPENDIX B -- FLOWCHART NUMBER 3 ***
CURRENT TAPE FILE UPDATE AND TRANSFER PROGRAM



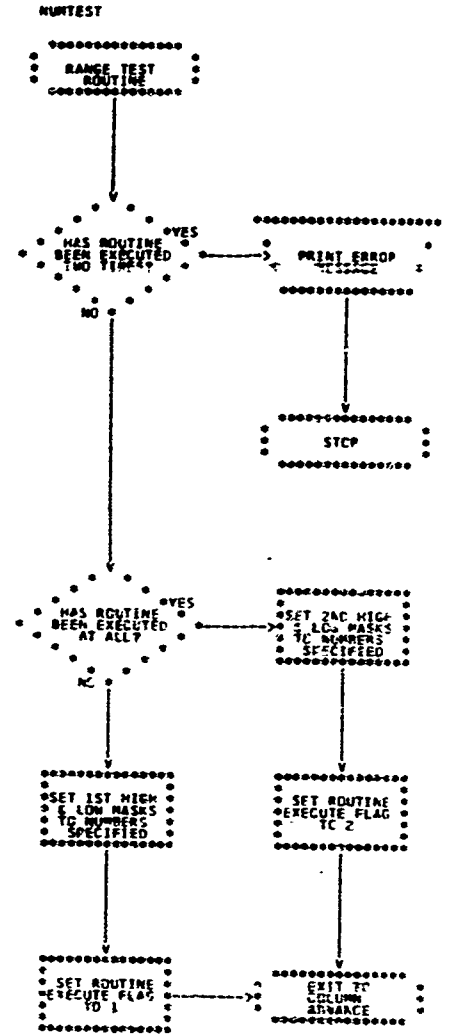
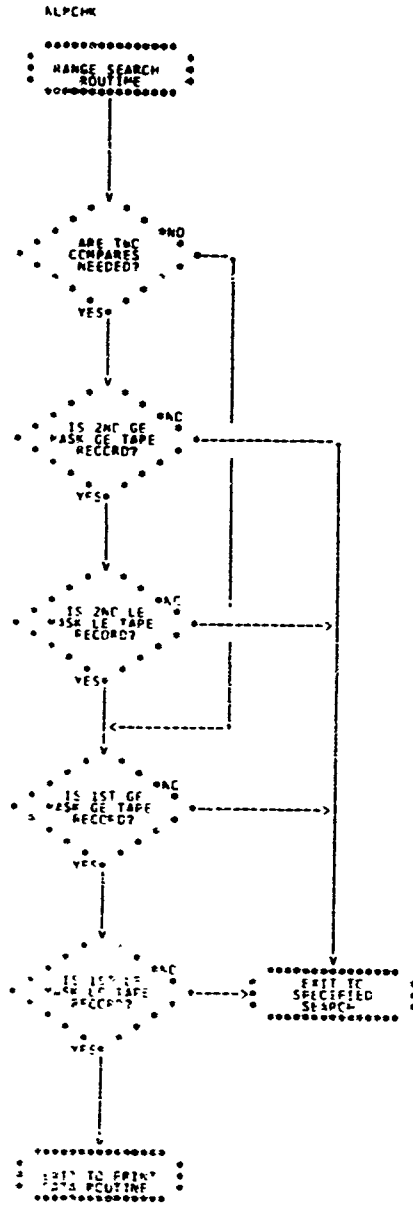
*** APPENDIX B -- FLOWCHART NUMBER 4A ***
DATA FILE SEARCH PROGRAM



*** APPENDIX B -- FLOWCHART NUMBER 48 ***
DATA FILE SEARCH PROGRAM



*** APPENDIX B -- FLOWCHART NUMBER 4C ***
DATA FILE SEARCH PROGRAM



CLC
BC
MVC
MVC
SR
PUT
MVI
B
NI
NI
LA
IC
LA
ACT
IC
SLI
MVC
B
LI
BC
MVC
SR
PUT
MVI
B
CLC
BC
MVI
NI
IC
SLI
BC
LA
LA
MVC
CLI
BC
MVI
CLI
BC
MVI
MVC

21(2,8),=CL2,41,
13(INE+26(2),21(8),
ELINE+16(4),=C,CTY,
6(10)
OUT,ELINE,
ELINE+26,C,
SEFX
21(8),X,OF,
22(8),X,OF,
4(16)TABLEC-160
2,21(8)
3,80
4,3
2,22(8)
2,22(8)
4,2
13(8,5),0(4)
SEFX
22(8),X,F6,
13(8),X,F6,
ELINE+26(2),21(8),
ELINE+16(4),=C,SVC,
6(10)
OUT,ELINE,
ELINE+26,C,
SEFX
21(2,8),=C,01,
7(8),X,C9,
30(5),X,C9,
22(8),X,OF,
2,22(8)
21(8),C,0,
8,12
3,12
3,12
3,12
13(4,5),0(3)
64(7),X,C6,
7,8
30(5),X,C6,
0(9),C,
8,10
53(5),C,1,
54(4,5),0(9)

SVC

SEFX

TEST CTY CODE FOR MAX VALUE
COND BRANCH AROUND CTR ERROR ROUT
MOVE BAD CTY CODE INTO ERROR MSG
INSERT CTY IN ERROR MSG
SUBTRACT 1 FROM LINE CTR
PRINT CTY CODE FROM ERROR MSG
CLEAR OLD CODE FROM ERROR MSG
BRANCH TO NEXT OUTPUT LINE ENTRY
CONVERT 1ST CTY CODE DIGIT TO BIN
SET R4 AS CTY CODE POINTER
INSERT 1ST CTY CODE DIGIT IN R2
SET R3 TO 80
ADD 10 LINES TO CTY POINTER
INCREASE POINTER VALUE IF REORD
INSERT 2ND CTY CODE DIGIT IN R2
MULTIPLY TABLE BY 3
SET CTY TABLE ADDR POINTER (R4)
MOVE CTY NAME INTO OUTPUT LINE
TEST SVC TO NEXT OUTPUT LINE
COND BRANCH AROUND SVC ERROR ROUT
MOVE BAD SVC CODE INTO ERROR MSG
INSERT SVC IN ERROR MSG
SUBTRACT 1 FROM LINE CTR
PRINT SVC ERROR MSG
CLEAR OLD CODE FROM ERROR MSG
BRANCH AROUND SVC DECODING ROUT
TEST FOR "IGEP" ENTRY
COND BRANCH AROUND IGEP ENTRY
MOVE 1 INTO OUTPUT LINE
CONVERT SVC CODE TO BINARY NR
INSERT NR IN R2
MULTIPLY NR BY 4
TEST FOR NON-US OFFICER
COND BRANCH TO USN TABLE
SET R3 TO NON-USN POINTER
BRANCH AROUND USN POINTER
SET R3 TO SVC TABLE BASE ADDR (R3)
SET SVC TABLE ADDR POINTER
MOVE SVC INTO OUTPUT LINE
COND BRANCH "ALF" SEX INDICATOR
MOVE BRANCH DESIGNATOR
COND BRANCH IN DESIGNATOR CHANGE ENTRY
TEST BRCH AROUND OLD DESIG DATA
INSERT OPENING PAREN
MOVE DESIG CHG DTE IN OUTPUT LINE

SCIENCE CENTER
FIRE DEPT. 100
COLUMBIA, MISSOURI 64108

UUUULLLUUUUUUUUUUU
CCCCCCCCC

PANK AND TITLE TABLE

[illegible][illegible]

TABLES

LA	6,CURRTAB+344	SET R6 TO TABLE SEARCH ADDR
CLC	126(4,2),0(6)	ADD ONE FOR A TABLE PTR (R6)
BCT	8,MAJOR	TEST FOR A TABLE PTR MATCH
B	7,PRDATA	COND BRANCH IF NEXT TABLE ENTRY
LA	7,26	BRANCH & TEST NEXT DATA ENTRIES
LA	6,CURRTAB+136	SET R6 TO 26500 SEARCH ADDR (R6)
CLC	126(4,2),0(6)	ADD ONE FOR A TABLE PTR MATCH
BCT	8,MAJOR	TEST FOR A TABLE PTR MATCH
B	7,PRDATA	COND BRANCH IF NEXT TABLE ENTRY
LA	7,10	BRANCH & TEST NEXT DATA ENTRIES
LA	6,CURRTAB+544	SET R6 TO 10000 SEARCH ADDR (R6)
CLC	126(4,2),0(6)	ADD ONE FOR A TABLE PTR MATCH
BCT	8,MAJOR	TEST FOR A TABLE PTR MATCH
B	7,PRDATA	COND BRANCH IF NEXT TABLE ENTRY

HIGH2

LOW2

PRDATA	MVC 1(130,5),0(2)	MOVE BAD DATA REC TO PRINT BUFF
	PUT PRDATA,LI	PRINT BAD DATA RECORD
	B READTP	BRANCH & READ NEXT RECORD

	EDIT BUPERS DATA RECORDS	
	TO LOCAL FORMAT	

MAJOR	MVC 76(3,3),4(6)	MOVE CURRICULUM NR TO NEW RECORD
	MVC 0(6),3(3),3(2)	MOVE SERVICE NR TO NEW RECORD
	MVC 37(3),3(3),0(2)	MOVE NAME TO NEW RECORD
	MVI 48(3),3(3),46(2)	MOVE RANK, DESIG, YR GRP & S-CODE
	MVC 49(5,3),48(3)	INSERT BLANK IN 1ST DOR COL
	MVC 54(4,3),61(2)	FILL DATE OF BIRTH FIELD WITH BLK
	MVC 58(6,3),87(2)	MOVE PCODE TO NEW RECORD

[illegible]

[illegible]

[illegible]

[illegible]


```

MOVE DESGNATOR INTO = MASK AREA
MOVE 1'S INTO = AND MASK
COND BRANCH AROUND UNEQUAL MASK
SET ROUT IND TO UNEQUAL MASK AREA
MOVE DESGNATOR INTO # MASK AREA
MOVE 1'S INTO # AND MASK
ADD 7 TO CARD COL PTR
BRANCH TO BLANK CARD COL TEST

```

```

EQMASK+1(4),2(7)
EQMASK+6A(4),ONES
*+20
FLGMASK,X'02'
NEFMASK+1(4),2(7)
NEFMASK+6A(4),ONES
2,7
BLKCHK

```

```

MVC
MVC
R I
MVC
MVC
LA
B

```

```

*****
***** BRANCH OF SERVICE AND/OR FOREIGN *****
***** COUNTRY INDICATOR (F) TEST ROUTINE *****
*****
*****
*****
*****

```

```

*
*
*
*
*
*

```

```

TEST CARD COL FOR SVC/CNTRY IND
COND BRANCH TO NEXT TEST IF NE
SAVE NEXT TEST COL PTR
ADD 4 TO CARD COL PTR
TEST FOR EQUAL COMPARE INDICATOR
COND BRANCH AROUND UNEQUAL MASK
SET ROUT IND TO UNEQUAL MASK
COND BRANCH AROUND UNEQUAL MASK
SET ROUT IND TO UNEQUAL MASK
SAVE UNEQUAL MASK ADDR IN R5
TEST FOR NAVY OFFICERS ONLY
COND BRANCH AROUND NAVY MASK
MOVE 1'S INTO NAVY MASK AREA
MOVE 1'S INTO AND MASK COL TEST
BRANCH TO BLANK CARD COL TEST
TEST FOR ALL US OFFICERS OFF MASK
COND BRANCH AROUND ALL OFF MASK
MOVE 1'S INTO AND MASK COL TEST
BRANCH TO BLANK CARD COL TEST
TEST FOR FOREIGN OFFICERS
COND BRANCH AROUND FOREIGN MASK
MOVE 1'S INTO AND MASK COL TEST
BRANCH TO BLANK CARD COL TEST
MOVE SVC/CNTRY CODE TO MASK AREA
MOVE 1'S INTO AND MASK
ADD 5 TO CARD COL PTR
BRANCH TO BLANK CARD COL TEST

```

```

O(7),C'E'
7,DOORCHK
3,DOORCHK
2,4
1,7,C'='
FLGMASK,X'01'
6,FLGMASK
*+12
FLGMASK,X'02'
6,FLGMASK
2,7,C'N'
7,7,C'0'
6,6,C'0'
131(6),X'FF'
BLKCHK,C'U'
2,7,C'1'
6,6,C'1'
131(6),X'FF'
BLKCHK,C'F'
2,7,C'F'
6,6,C'0'
133(6),X'FF'
BLKCHK
64(2,6),2(7)
131(2,6),ONES
2,5
BLKCHK

```

```

CLI
BC
LA
CLI
BC
OT
LA
R I
OT
LA
CLI
BC
MVI
MVI
B CLI
BC
MVI
MVI
B CLI
BC
MVI
MVI
B MVC
LA
B

```

```

SVCHK

```



```

*****
***** YEAR-GROUP INDICATOR (G) TEST ROUTINE *****
*****
*****
*****
*****

```

```

YGCHK
CLI 0(7),C'G'
RC 7,RANKCHK
LA 3,RANKCHK
LA 2,5
CLI 1(7),C'='
RC 7,++16
DI FIGMASK,X'01'
LA 6,EOMASK
RC RANKCHK-16
LA 1(7),C'#'
CLI 7,++16
RC FLGMASK,X'02'
DI 6,NFEMASK
LA B RANKCHK-16
LA 6,RDTAPE+42
B NUMTEST
MVC 5(2,6),2(7)
MVC 72(2,6),ONES
B BLKCHK

TEST CARD COL FOR YEAR GRP IND
COND BRANCH TO NEXT TEST IF NE
SAVE NEXT TEST ROUT ADDP IN R3
ADD 5 TO CARD COL PTR
TEST FOR EQUAL COMPARE INDICATOR
COND BRANCH AROUND EQUAL MASKING
SET ROUT INDICATOR TO = COMPARE
SAVE EQUAL MASK ADDR IN R6 TESTS
BRANCH AROUND OTHER YR GP TESTS
TEST FOR UNEQUAL COMPARE IND
COND BRANCH AROUND # MASK ROUT
SET ROUT IND TO UNEQUAL MASK
SAVE UNEQUAL MASK ADDR IN R6
BRANCH AROUND YR GP RANGE TEST
SAVE YRGP ADDR IN TAPE READ AREA
BRANCH TO < AND > MASK ROUTINE
MOVE YEAR GROUP INTO MASK AREA
MOVE 1'S INTO AND MASK
BRANCH TO BLANK CARD COL TEST

```

```

*****
***** RANK INDICATOR (H) TEST ROUTINE *****
*****
*****
*****

```

```

PANKCHK
CLI 0(7),C'H'
RC 7,CURCHK
LA 3,CURCHK
DI FLGMASK,X'01'
LA 2,4
MVC EOMASK(1),2(7)
MVI EOMASK+67,X'FF'
B BLKCHK

TEST CARD COL FOR RANK INDICATOR
COND BRANCH TO NEXT TEST IF NE
SAVE NEXT TEST ROUT ADDP IN R3
SET ROUT INDICATOR TO = COMPARE
ADD 4 TO CARD COL PTR
MOVE RANK CODE INTO = MASK AREA
MOVE 1'S INTO = AND MASK
BRANCH TO BLANK CARD COL TEST

```

```

*****
***** CURRICULUM INDICATOR (I) TEST ROUTINE *****
*****
*****
*****

```

```

CURCHK
CLI 0(7),C'I'
TEST CARD COL FOR CURRICULUM IND

```

```

BC 7,DEGACHK
LA 3,DEGACHK
OI FLGMASK,X'01'
LA 2,6
MVC EQMASK+39(3),2(7)
B EQMASK+104(3),DNFS
BLKCHK

```

```

COND BRANCH TO NEXT TEST IF NE
SAVE NEXT TEST ROUT ADDR IN R3
SET ROUT INDICATOR TO = COMPARE
ADD 6 TO CARD COL PTR
MOVE CURRICULUM INTO = MASK AREA
MOVE I'S INTO = AND MASK
BRANCH TO BLANK CARD COL TEST

```

```

*****
***** EDUCATIONAL LEVEL AND AREA INDICATORS *****
***** (J,K) TEST ROUTINES *****
*****

```

```

DEGACHK CL I 0(7),C'J'
BC 7,DEGLCHK
LA 3,DEGLCHK
OI FLGMASK,X'01'
LA 2,4
MVC EQMASK+63(1),2(7)
B EQMASK+130,X'FF'
BLKCHK
DEGLCHK CL I 0(7),C'K'
BC 7,OPRGCHK
LA 3,OPRGCHK
OI FLGMASK,X'01'
LA 2,4
MVC EQMASK+42(1),2(7)
B EQMASK+103,X'FF'
BLKCHK

```

```

TEST CARD COL FOR EDUCATION DEG
COND BRANCH TO NEXT TEST IF NE
SAVE NEXT TEST ROUT ADDR IN R3
SET ROUT INDICATOR TO = COMPARE
ADD 4 TO CARD COL PTR
MOVE DEC AREA ONE TO = MASK AREA
MOVE I'S INTO = AND MASK
BRANCH TO BLANK CARD COL TEST
COND CARD COL FOR EDUCATION LEVL
COND BRANCH TO NEXT TEST IF NE
SAVE NEXT TEST ROUT ADDR
SET ROUT INDICATOR TO = COMPARE
ADD 4 TO CARD COL PTR
MOVE DEG LEVEL CODE TO = MASK
MOVE I'S INTO = AND MASK
BRANCH TO BLANK CARD COL TEST

```

```

*****
***** GRADUATE AND TOTAL OPR INDICATORS *****
***** (L,M) TEST ROUTINES *****
*****

```

```

OPRGCHK CL I 0(7),C'L'
BC 7,OPRTCHK
LA 3,OPRTCHK
LA 6,RTAPE+94
B NUMTEST
OPRTCHK CL I 0(7),C'M'
BC 7,SDTECHK
LA 3,SDTECHK
LA 6,RTAPE+97

```

```

TEST CARD COL FOR GRAD OPR IND
COND BRANCH TO NEXT TEST IF NE
SAVE NEXT TEST ROUT ADDR READ AREA
SAVE OPRG TO <A,C> MASK ROUTINE
TEST CARD COL FOR TOY OPR IND
COND BRANCH TO NEXT TEST IF NE
SAVE OPRD ADDR IN TAPE READ AREA

```


DC	C	IRAN	:
DC	C	ISRAEL	:
DC	C	KOREA	:
DC	C	NORWAY	:
DC	C	PERU	:
DC	C	PHILL.	:
DC	C	PORT.	:
DC	C	THAILAND	:
DC	C	TURKEY	:
DC	C	URUGUAY	:
DC	C	VENE.	:
DC	C	VIETNAM	:
DC	C	FORIGN	:

DEGREE (EDUCATIONAL) AREA TABLE

TABLED

RANK AND TITLE TABLE

TABLE C


```

DC      C'USAF'
*****
***** DATA STATUS MESSAGES AND TITLE CONSTANTS *****
*****
TITL1   DC      C'1NAME      RANK      ORIG      1ST      SOCIAL      BIRTH DEGR
TITL2   DC      C' RANK,FILE NR,SVC/CTY(,*) RET      P SECURITY QTR-Y? DATE /
TITL3   DC      C' (OPR DATE NR,FILE NR,SVC/CTY(,*) RET      P SECURITY QTR-Y? DATE /
TITL4   DC      C' (CHG) YR-M-D CODE GP YR CODE ST/END NUMBER YR-M-D LUM AREA
C' +-----+-----+-----+-----+-----+-----+-----+-----+-----+
ERRLINE DC      C'0***** ERROR IN INPUT CARDS ***** PRO
GRAM WILL NOT RUN ***** RECHECK INPUT CARDS A
ND RESUMIT *****
END      DC      C'0***** END OF DATA ***** END OF
DATA ***** END OF DATA ***** END OF
*****
***** DEFINED STORAGE AREA *****
*****
DBCK     DC      X'00'
TEST     DC      X'00'
ONES     DC      10X'FF'
FLGMA SK DC      X'00'
NUMFLG   DC      134X'00'
FOMASK   DC      134X'00'
ENEMASK   DC      180X'00'
CDAREA   DC      232X'00'
RDTAPE   DC      133X'00'
LINEGT1  DS      CL2
SAVEGT1  DS      CL2
SAVEGT2  DS      CL2
TWC CARD DECK FL
SCRATCH TEST AREA
BINAPY ONES STORAGE
COMPA RE ROUTINE INDICATOR
DATA RANGE COMPARE INDICATOR
EQUAL COMPARE MASK
NOT EQUAL COMPARE MASK
CARD INPUT AREA
TAPE INPUT AREA
OUTPUT DATA AREA
1ST GREATER THAN SAVE AREA
1ST LESS THAN SAVE AREA
2ND GREATER THAN SAVE AREA

```

[illegible]

APPENDIX C

STATUS OF FILES AND PROGRAMS

The following information is provided to assist file and program maintenance personnel in carrying out assigned responsibilities.

SOIRS files are resident on labeled magnetic tape in fixed block record format with a blocking factor of four (4). Pertinent data on each file is listed as follows:

<u>FILE</u>	<u>DSNAME</u>	<u>VOLUMN NUMBER</u>
Bureau Original	HEN.SORT	NPS 209
Historical Father	HEN.MAS	NPS 138
Current Father	HEN.DONE	NPS 194
Historical Son	HEN.HSON	NPS 150
Current Son	HEN.CSON	NPS 155

SOIRS application programs are resident on disk pack (2311) FAC001 in load module form. The DSNAME of the library is HEN.-MISLIB. The member names of programs in this library correspond to the program names used in the main text of this paper and are listed as follows:

1. BUPSXFR
2. CRDCHK1
3. UPDATE
4. FILESRCH

DOCUMENT CONTROL DATA - R & D		
Security Classification <i>(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)</i>		
1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940		2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP
3. REPORT TITLE A Proposed Student Officer Information Retrieval System for the Naval Postgraduate School		
4. DESCRIPTIVE NOTES (Type of report and, inclusive dates) Master's Thesis: June 1969		
5. AUTHOR(S) (First name, middle initial, last name) Robert Lee Henry Lawrence Lee Massa		
6. REPORT DATE June 1969	7a. TOTAL NO. OF PAGES 147	7b. NO. OF REFS 0
8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
10. DISTRIBUTION STATEMENT Distribution of this document is unlimited		
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940	
13. ABSTRACT Pertinent information on past and present Naval Postgraduate School students is now maintained, stored and processed in bulk files by Curriculum Officers. Information desired for management studies or analysis requires manual sorting of an ever increasing number of individual student records. This is an inadequate and inefficient system. The foregoing problem could be resolved by the implementation of the Student Officer Information Retrieval System (SOIRS), which is a narrow scope retrieval system specifically designed to be responsive to the Curriculum Officer's needs with respect to student information. SOIRS evolved through a series of logical system design steps, identified as follows: (1) Problem Analysis; (2) Design of Records, Files and Reports; (3) Software Design; (4) Test of Entire System. SOIRS is a complete system, establishing required files, updating files, and retrieving stored information		

