

MEMORANDUM
RM-5550-ARPA
MAY 1968

REAL TIME RECOGNITION
OF HANDPRINTED TEXT:
PROGRAM DOCUMENTATION

G. F. Groner

PREPARED FOR:
ADVANCED RESEARCH PROJECTS AGENCY

The **RAND** *Corporation*
SANTA MONICA • CALIFORNIA

MEMORANDUM

RM-5550-ARPA

MAY 1968

REAL TIME RECOGNITION
OF HANDPRINTED TEXT:
PROGRAM DOCUMENTATION

G. F. Groner

This research is supported by the Advanced Research Projects Agency under Contract No. DAHC15 67 C 0141. RAND Memoranda are subject to critical review procedures at the research department and corporate levels. Views and conclusions expressed herein are nevertheless the primary responsibility of the author, and should not be interpreted as representing the official opinion or policy of ARPA or of The RAND Corporation.

DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

PREFACE

This Memorandum documents a computer program for the recognition of symbols handprinted on a RAND Tablet or similar device used in conjunction with a CRT display. This documentation describes the program in sufficient detail to facilitate its use, maintenance, and/or recoding in another computer language. Since the program is written in IBM-360 assembly language, understanding of the documentation requires familiarity with this language. The study resulting in this program is but one facet of an overall search for techniques to increase the facility of the man-computer interface.

SUMMARY

This Memorandum documents a computer program that permits an on-line computer user to print text naturally and have it recognized accurately. The program recognizes handprinted letters, numbers, punctuation marks, and geometric figures; it separates characters written in quick succession and in close proximity. The program is written as a re-entrant process in IBM-360 assembly language; it requires about thirty-seven hundred 32-bit words of storage. The user must provide programs that 1) communicate with an input device such as the RAND Tablet to supply a sequence of writing-instrument coordinates to the recognition program; 2) select options in real-time based on context; and 3) use the recognition program's outputs for displaying and editing information on a CRT display device.

This documentation describes the program at two levels. The most general description lists the symbols recognized and discusses feature extraction, character separation, symbol recognition, and user options. The second level provides a computer listing of the assembly-language program. This listing includes descriptions of the logical functions, calling sequences, and input/output parameters of each of the major processes comprising the program, and outlines the information processing and flow of control. The Appendix briefly describes processes and macros that perform functions required by the recognition program.

CONTENTS

PREFACE	iii
SUMMARY	v
Section	
I. INTRODUCTION	1
The Program	2
The Documentation	3
Glossary	5
II. GENERAL DESCRIPTION OF THE PROGRAM	11
The Symbols Recognized	11
Feature Extraction	11
Character Separation	13
Character Identification	13
User Options	17
Controls	17
Vector Length	18
Character Size	18
Between-Character Time Delay	18
III. FUNCTIONAL AND PROCEDURAL DESCRIPTIONS OF THE PROCESSES AND RCS'S	20
CHAREC	20
CHAREC Function	20
CHAREC Call	21
CHAREC Inputs	21
CHAREC Outputs	22
CHAREC Exits	22
CHAREC Parameters	22
CHAREC Read-Only Constants	24
CHAREC Sequence of Information Processing	24
CHAREC Program Listing	29
CHAREC RCS'S	51
ANG4	51
CHECK	55
CORNER	61
DOT	64
DELTAS	65
FN56	67
HYST	68
MXMNC	69
MXMNS	71
QMM	72
RAZE	74
RELM	75
SMOOTH	78

TCRNR	79
THIN	80
TURNER	81
TRAVEC	83
REC	86
REC Function	86
REC Call	86
REC Sequence of Information Processing	87
REC Program Listing	88
INTERP	99
INTERP Function	99
INTERP Call	99
INTERP Sequence of Information Processing	100
INTERP Program Listing	101
REC RCS'S	121
AHSTR1	121
BFI	124
BHTE	125
BSDP	126
BSMNW	128
BSRPRM	130
BSSM	132
BSVM	134
BTEST1	136
BTEST3	137
KNYTST	139
KNYIT	141
KVXYT	143
MWT	145
PSTEST	147
SYMT	148
TILDT	155
TPXY	157
VERTST	159

Appendix

The OS/360 Operating System--2250 Display

Recognition Program	161
CRT Display Character Codes	162
Register Assignment	163
Processes	163
CHAR	163
CLOCK	164
Macros	165
BEXIT	165
BOX	165
CLEAR	166
EPLOG (Epilogue)	166
INST (Instance)	167

PARL (Parallel)	168
PAWS (Pause)	169
PROCS (Process)	170
PROLG (Prologue)	170
RCS (Remote Code Sequence)	171
REGS (Registers)	172
SET	173
SVCS	173
TABLE	174
WATE (Wait)	175
REFERENCES	177

I. INTRODUCTION

This Memorandum documents a symbol-recognition program[†] that is part of an experimental software system called GRAIL (GRAPhical Input Language) [2] under development at The RAND Corporation (and supported by the Advanced Research Projects Agency). The objective of GRAIL is to investigate methods by which a user may deal directly, naturally, and easily with his problem. As one means of eliminating distracting operational mechanics from problem solving, the system features the ability to communicate with a computer via a single pen-like instrument moved over a two-dimensional surface in conjunction with a *CRT display*.^{††} Communication is enhanced by incorporating a program that interprets freehand motions and provides immediate feedback.^{†††}

This *symbol*-recognition program allows an on-line computer user to print or draw symbols naturally, and have them recognized accurately and quickly, even though it recognizes a large set of symbols. Designed to work for many users, the program imposes few constraints on style, speed, or position of writing; it is not intended to be modified for individual printing styles. It makes use of size and position information to differentiate among symbols not distinguishable by shape alone. Preliminary experiments [1] indicate that recognition accuracy (not including lower-case letters and geometric symbols) is about 90 percent for inexperienced

[†] A general description of this program together with a discussion of user interaction, a performance evaluation, and references to related work appear in Ref. 1.

^{††} Italicized words are defined in the Glossary at the end of this section (pp. 5-10).

^{†††} An immediate, continuous track on the display corresponds to the writing instrument position. A completed track is replaced by a symbol after a few milliseconds for recognition plus a time delay for symbol separation.

users. This error rate is tolerable because of the quick response and the GRAIL editing facilities.

The recognition program has been used daily, as part of the GRAIL system, while developing means for creating, editing, and executing computer code and flowcharts. The GRAIL system is being developed on an IBM System/360 Model 40 and is written in 360 assembly language.

The recognition program within the GRAIL system is written to operate under a nonstandard GRAIL supervisor and in conjunction with a nonstandard CRT display; a modified version has been written that operates under the IBM OS/360 operating system and in conjunction with an IBM 2250 display unit. The differences between the GRAIL recognition program documented here and the OS program are summarized in the Appendix. The OS program also has a number of users at RAND (its use is described in Ref. 3).

THE PROGRAM

The user must provide programs that: 1) communicate with an input device such as the RAND Tablet [4] in order to provide a sequence of x, y coordinates to the recognition program; 2) select options in real-time based on the context of the input; and 3) use the recognition program's outputs for displaying and editing information on a CRT display based on context. When the recognition program has been provided with a time-ordered set of x, y coordinates (describing the motion of a writing *stylus*) and a set of control bits, it normally places *vector strings* (which approximate the stylus motion) directly into a display *buffer* as it receives the inputs; upon completion of each symbol, the program returns a *character code* (its interpretation of the input) along with some geometrical properties of the symbol.

The recognition program is written as a *reentrant process* in 360 assembly language. It requires about thirty-seven hundred 32-bit words of storage. Each logical *instance*

of this *process* requires 26 words for *data* and *context*; the remaining storage is for the *read-only* code, which is required only once.

The user program *calls* the process CHAREC, which in turn calls the processes REC and CLOCK and a set of *remote code sequences* (processes with general-purpose register input/output operating in the environment of the calling process context) referred to herein as *RCS's*. CHAREC and its *RCS's* perform "inking" (generation of the vector strings), *feature* extraction, and character separation. CLOCK is used as a real-time clock for separating characters by timing. REC, together with its *RCS's*, identifies characters by testing the features computed by CHAREC. Most of the tests are performed in INTERP, an *RCS* comprised of decision tables. Figure 1 outlines the input/output parameters and logical functions of the two processes CHAREC and REC. The processes and *RCS's* called by CHAREC and REC are indicated by asterisks. The figure was drawn using the GRAIL system (but does not illustrate this system's scope or symbology).

THE DOCUMENTATION

The following documentation describes the program at two levels. The most general description lists the symbols recognized and discusses feature extraction, character separation, character identification, and user options.

The second level provides a computer listing of the assembly-language program. This listing includes descriptions of the logical functions, *calling sequences*, and input/output *parameters* of each of the processes and *RCS's* (except CLOCK); and outlines the sequence of information processing in CHAREC, REC, and INTERP. *Entry points* in these outlines are labeled (e.g., ****ENTRY****) identically to the corresponding entry points in assembly-language program listings. Also described are the program's parameters, features, and indicators used by CHAREC, REC, and the *RCS's*.

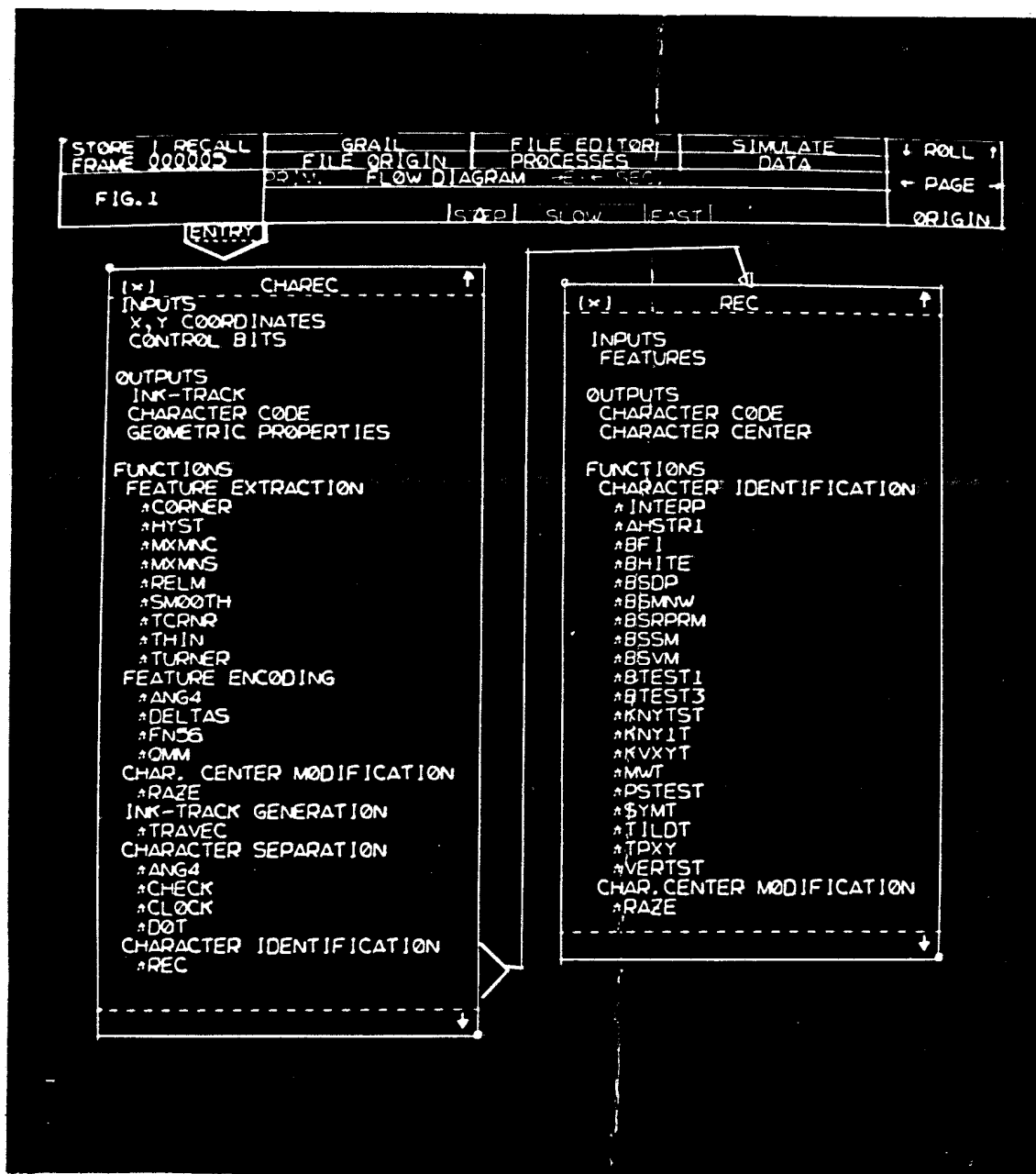


Fig. 1—CHAREC and REC outlines

In addition to summarizing the difference between the GRAIL and OS programs, the Appendix lists the CRT display character codes and briefly describes CLOCK, CHAR (the GRAIL process that allows the user's application program to interact with the Tablet by providing a convenient interface), and the GRAIL macros as used by the recognition program.

GLOSSARY[†]

A(NAME)	The address of NAME.
ANAME	In a <i>call</i> to <i>process</i> NAME, this is a linkage between the calling process <i>context</i> and NAME's context; the label "ANAME" is user determined.
aspect ratio	A <i>character's</i> height divided by its width.
buffer	A number of <i>bytes</i> used for transmitting <i>x,y</i> coordinates to the recognition program or <i>vector</i> strings from the program.
byte	Eight bits; referred to as 0 to 7, left to right.
call	Transfer of flow of control to another <i>process</i> .
calling sequence	The sequence of information and commands required to <i>call</i> a <i>process</i> .
cannot interpret	A sequence of input coordinates not interpretable as one of the allowable <i>symbols</i> ; same as " <i>no character</i> ."
CRT	Cathode ray tube.

[†]In addition to those italicized above, other words and phrases used throughout the text are also defined.

character	A sequence of input coordinates encoded as an entity by this program; same as " <i>symbol</i> " (see The Symbols Recognized, Sec. II).
character code	A 1-byte encoding of a <i>character</i> (see CRT Display Character Codes, Appendix, p. 162).
context	1) a continuous storage block consisting of linkages between <i>parent</i> (<i>calling</i>) and <i>daughter</i> (<i>called</i>) processes, <i>formal parameters</i> , and other information; 2) the environment used to interpret the meaning of an action or inputs.
data	1) <i>x,y</i> coordinates; 2) indicators or computed quantities used by the program.
daughter process	A <i>process</i> called by a <i>parent process</i> .
display	A programmed output device that presents an image.
display stream	The sequence of instructions controlling the <i>display</i> .
EEXIT	Appears in a <i>call</i> to a <i>process</i> or RCS; EXIT is a re-entry point in the calling (<i>parent</i>) process corresponding to a return from the called (<i>daughter</i>) process or RCS; the label "EXIT" is user determined.
ending point	The <i>x,y</i> position at which the writing <i>stylus</i> micro switch is opened when terminating a <i>stroke</i> .
entry point	The place at which control resumes.

F	1) full computer word (32 bits); 2) <i>formal</i> (input/output) <i>parameter</i> .
feature	A computed attribute of a <i>symbol</i> which is used for identification.
formal parameter	An input/output <i>data</i> location provided a <i>process</i> by its <i>parent</i> .
FPARAM	In a <i>call</i> to a <i>process</i> , refers to the <i>formal</i> (input/output) <i>parameter</i> PARAM of the calling (<i>parent</i>) <i>process</i> ; the label "PARAM" is user determined.
geometric corner	A detected sharp change (90° or more) in the direction of the writing <i>stylus</i> motion.
GPARAM	A reference to the <i>parameter</i> PARAM in a <i>call</i> to a <i>process</i> . $G = F$ for a <i>formal</i> (input or output) <i>parameter</i> of the calling <i>process</i> ; $G = I$ for an <i>informal</i> (local) <i>parameter</i> ; the label "PARAM" is user determined.
H	Computer halfword (16 bits).
informal parameter	Temporary or constant <i>data</i> defined within a <i>process</i> .
ink	1) same as " <i>ink track</i> "; 2) the action of generating an <i>ink track</i> .
ink track	A displayed string of <i>vectors</i> that approximates the writing <i>stylus</i> motion.
instance	The appearance of a <i>calling sequence</i> to a <i>process</i> in the program.
IPARAM	In a <i>call</i> to a <i>process</i> , refers to the <i>informal</i> (local) <i>parameter</i> PARAM of

	the calling (<i>parent</i>) process; the label "PARAM" is user determined.
NAMEA	In a <i>call</i> to process NAME, a read-only link to NAME; the label "NAMEA" is user determined.
no character	A sequence of input coordinates not interpretable as one of the allowable symbols; same as " <i>cannot interpret</i> ."
parameter	Temporary or constant <i>data</i> .
parallel task	An instruction sequence initiating two lines of control within the program.
parent process	The <i>process</i> that <i>called</i> a <i>daughter process</i> .
pen	The writing instrument that is moved on the <i>Tablet</i> writing surface; same as " <i>stylus</i> ."
pendown	Closure of the writing <i>stylus</i> micro switch due to a downward force.
penup	Opening of the writing <i>stylus</i> micro switch by release of a downward force.
PSG	Program Status Group, a GRAIL conceptual entity used for parallel task synchronization.
process	A computer program segment, somewhat akin to a subroutine, accessed by a formal call (see " <i>reentrant process</i> ").
raster unit	1/1024 of the <i>Tablet</i> or <i>display</i> surface dimension--0.01 in. in the case of a standard 10.24 by 10.24-in. Tablet.

raw data point	A writing stylus coordinate pair as received from the input device.
read-only	Computer storage that is read (and executed if code) but not modified.
reentrant process	A <i>process</i> requiring separate linkage and data storage blocks for each usage, but only a single storage block of <i>read-only</i> code. When executed, the code is not modified and therefore may be re-used even if the process has been suspended before completion.
RCS	<i>Remote code sequence.</i>
remote code sequence	A process with general-purpose register input/output operating in the environment of the <i>calling (parent) process context</i> ; has no context but is <i>reentrant</i> .
starting point	The <i>x,y</i> position at which the writing <i>stylus</i> micro switch is closed when initiating a <i>stroke</i> .
stroke	The sequence of <i>x,y</i> coordinates between closing and opening the writing <i>stylus</i> micro switch.
stylus	The writing instrument that is moved on the <i>tablet</i> writing surface; same as " <i>pen</i> ."
subcharacter	A set of <i>x,y</i> coordinates encoded internally by the program, but which may not be a complete <i>character</i> and has not been outputted by the program.

symbol	A sequence of input coordinates encoded as an entity by this program; same as " <i>character</i> " (see The Symbols Recognized, in Sec. II).
tablet	An input device comprising a pen-like writing instrument and a writing surface [4]; as the <i>stylus</i> is moved over the surface its <i>x,y</i> coordinates are sent to the computer for processing.
task	A sequence of instructions initiating lines of control (see " <i>parallel task</i> ").
time-pause corner	A detected deceleration-acceleration of the writing <i>stylus</i> motion.
track	1) same as " <i>ink track</i> "; 2) the action of generating an <i>ink track</i> .
vector	A line segment described by its length (2, 4, 6, or 8 <i>raster units</i>) and direction (1 of 16 in 22.5° increments).
x	The writing surface horizontal coordinate.
X (or any other non-blank character in column 72)	A continuation indicator.
y	The writing surface vertical coordinate.

II. GENERAL DESCRIPTION OF THE PROGRAM

THE SYMBOLS RECOGNIZED

Upper-case Latin alphabet.

Numbers: 0 through 9.

Lower-case (script) Latin alphabet: these characters are not recognized very accurately in the present program. A lower-case character output code may be changed to the corresponding upper-case output code by a one instruction change in CHAREC.

Punctuation marks: + - = / () * \$. , ' #

Left bracket, right bracket, less than, greater than, karat, tilda (tilda is not fully implemented--see TILDT, p. 155).

Geometric symbols (must be single stroke and larger in one dimension than twice the normally expected character height): Rectangle, circle, triangle (one side horizontal, the other two of approximately equal length), ellipse, diamond, trapezoid.

Erasure (scrubbing action).

Cannot interpret.

FEATURE EXTRACTION

The on-line nature of this program enables processing of the data point-by-point as the stylus is moved across the writing surface. In order to minimize time and storage requirements, therefore, CHAREC (together with its RCS's) extracts features as the data arrive. These features are:

The sequence of directions (right, left, up, or down) of stylus motion.

The number and relative (to character extents) positions of geometrically determined corners.

The number of pause-in-time determined corners.

The number and relative positions of relative maxima and minima in y (the vertical direction).

The number and relative positions of stroke starting and ending points.

The absolute size of the character in raster units (1 raster unit = 0.01 inch).

The ratio of height to width of the character.

The absolute position of the center of the character on the writing surface.

The first process in feature extraction is data reduction (thinning). When a data point arrives, its position is compared with that of the most recently accepted data point. It is accepted (used in further analysis) if these two points are sufficiently far apart; otherwise it is rejected. When this thinning distance is set to 0.02 in., data are reduced by a factor of about seven without losing any significant information about a 1/4-in.-high handprinted character. (The number of raw data points between thinned data points is required, however, for detecting pause-in-time corners.) Upon the acceptance of each new data point, tests are made for stylus direction, corners, and relative maxima and minima.

CHAREC is called into action when the stylus is placed on the writing surface (micro switch closed), and is notified (via an indicator) when it is lifted (micro switch opened). CHAREC is thus informed about the starting and ending of each stroke. When a stroke is completed, tests are made to determine if it is part of the same character as the previous stroke set (previous subcharacter). If so, the character extents are updated, the positions of various features are computed relative to these character extents, and this subcharacter is identified. Otherwise, the

previous subcharacter is outputted as a character, this stroke treated as a new subcharacter, relative positions computed, and the stroke identified.

CHARACTER SEPARATION

CHAREC groups sets of strokes into characters by considering timing, and the geometric extents and identifications of the strokes. If a prespecified time elapses following the end of the most recent stroke, a character is considered completed regardless of what follows. This between-character time delay must be greater than the maximum expected delay between two strokes belonging to the same character--0.3 sec has proven optimum for experienced users. A set of strokes is considered to be a completed character if it cannot be combined with the following stroke to form an allowable character. Some stroke sets (e.g., those that form 8, Q, A, and E) cannot be combined with any other stroke to form an allowable character. Some other stroke sets (e.g., 0, 2, 3, T, and F) can be combined with some strokes but not with others. Strokes written in quick succession, which can be combined to form an allowable character, are tested for overlapping or adjacency--thus separating groups of strokes too far apart to form a character of the normally expected size.

CHARACTER IDENTIFICATION

REC (together with INTERP and RCS's) uses the set of features generated by CHAREC (and its RCS's) to decide what character was written. Individual strokes are identified, as they are drawn, via a data-dependent sequence of tests. The first test groups stroke descriptions according to the first four stylus directions. This test reduces the number of stroke possibilities--typically, to one or two. Any further test depends on the set of possible stroke

identifications, and on previously tested features. The program thus has a tree structure as outlined in Fig. 2.

The recognition of a multiple-stroke symbol is based on the identities of the constituent strokes and on their relative positions--it is independent of stroke order. In most cases, each constituent stroke requires only a general, rather than a precise, identification (which is a code in P or PAD). For example, a stroke recognized as a 1,), (, or / if standing alone, need only be considered as a vertical (P=1) if part of a multiple-stroke symbol. This simplifies decision making.

REC performs a few simple tests, but mostly acts as a link between CHAREC and the testing procedures (INTERP and the RCS's), or between INTERP and the RCS's. INTERP performs sequences of tests on encoded 1-byte parameters, thereby including nearly all of the decision-making tree structure. Most of the RCS's perform complicated tests to discriminate among a particular set of characters.

The following comments may be useful when adding or deleting a character description. To add a description, write the character, observe its description (set of features calculated by CHAREC) either visually or in computer memory, and note the character code(s) outputted by the decision-making routines. If multiple characters are outputted, or if a single character with fewer strokes than the written character is outputted, then either this particular stroke combination is not allowed and must now be added to CHECK, or a new PAD code and a new PAD table (see INTERP, p. 116) entry must be added. If this problem does not occur, find the direction sequence (as encoded by ANG4) entry into INTERP; then follow through the tests, eventually reaching the test resulting in the outputted character. At this place, enter a feature test that will consistently distinguish between the written character and the outputted character. If no such feature (or set of features) exists,

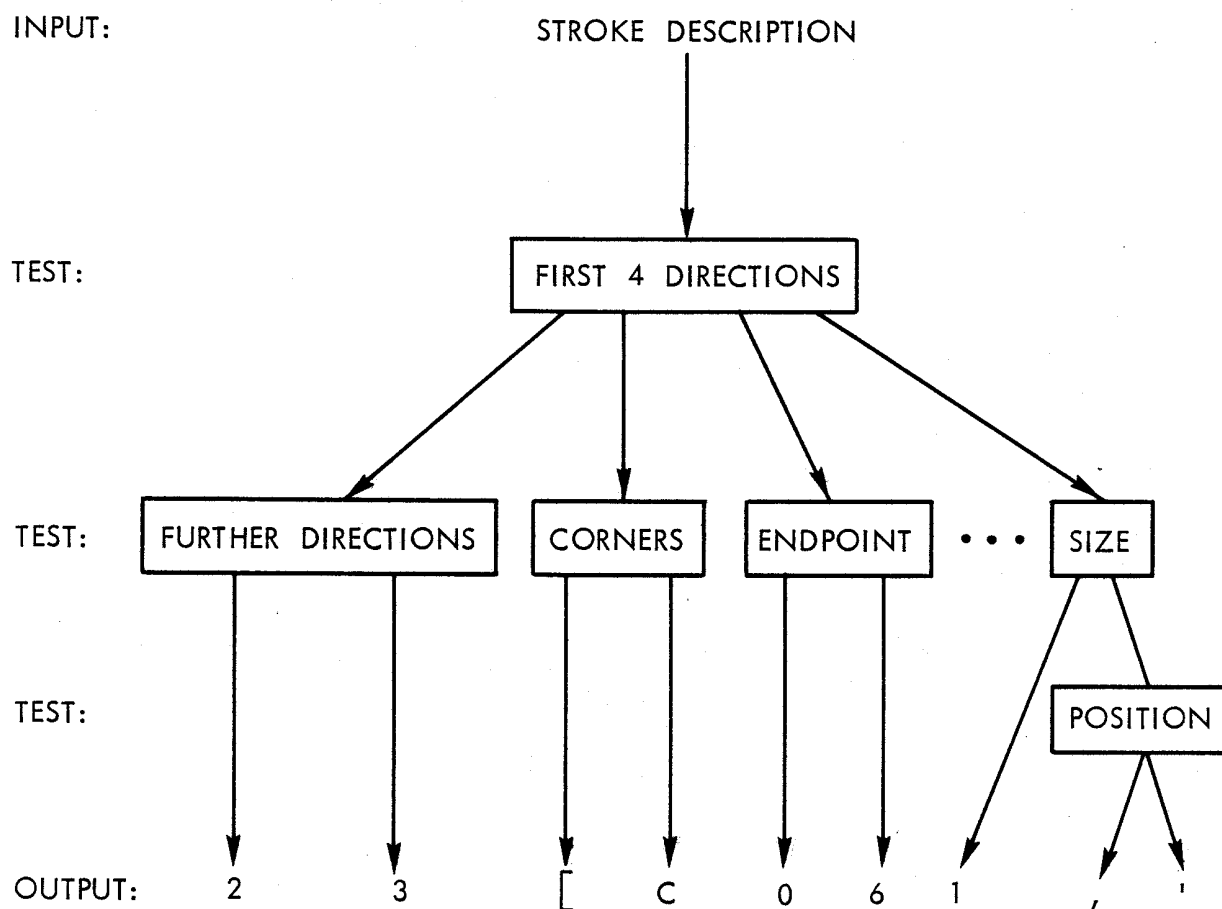


Fig.2—Outline of tree structure for character identification

it will be necessary to add a new CHAREC RCS to extract some new feature from the raw data. If this decision point occurs in the middle of a sequence of tests, it may be necessary to introduce a new PAD code and table entry. If strokes may be added to this character to generate new multi-stroke character descriptions, it must be added to CHECK. To delete a character description, follow through the tests as above, but delete the test(s) that result in this character. There may also be corresponding deletions from CHECK and the PAD codes and table entries.

A modification of the recognition program has been written that recognizes the mathematical symbols square root, infinity, integral, summation, and diagonal (upper-left to lower-right) in addition to the current symbols (except apostrophe and the geometric symbols). In order to allow any symbol to be written any size and at any position, the section of CHAREC that separates characters according to size and position (see CHAREC, p. 41) and the call (in REC, p. 93) to SYMT (which recognizes large single-stroke symbols as geometric symbols) were deleted. The tests for apostrophe were deleted from PSTEST so that a comma can be recognized when written in any position. The only new multi-stroke symbol--infinity comprised of the same strokes (2 0-like strokes) as a description of the number 8--did not require a change in CHECK or a new PAD code. The new symbols were added, however, to certain places in CHECK so that they can be combined with additional strokes to form multi-stroke symbols--e.g., if diagonal were not added to the vertical stroke section of CHECK, the letter x could not be written as a diagonal followed by a vertical. Since one of the first-4-direction descriptions (right-down-up-right) was previously a unique description (recognized as a script v), but could now also be a description of square root, a new code in ANG4 and a corresponding new entry into INTERP were added. All other changes--either feature tests

or setting character codes--were made in INTERP. For example, a stroke with the direction sequence up-down-up--starting point not in the lower quarter of the stroke, and ending point in the lower half of the stroke--was recognized as the number 2; but now, in addition, it could be the symbol integral. At the place where these tests result in a branch to set the character code to 2 (see SNLC1 in INTERP, p. 111), this branch was replaced by a 2 versus integral test. This new test results in a branch to set character code to 2 if the stroke starting point is in the left half of the stroke; otherwise, it results in a branch to set character code to integral.

USER OPTIONS

Controls

CHAREC normally provides an ink track (constructed of vectors of user-specified length), and outputs character codes along with some character size and position information. The ink track for a handprinted character is deleted upon recognition of that character. The user may control the operation of CHAREC by specifying no-track and/or no-recognize, or halt with each group of data points (including during mid-stroke).

No-Track. CHAREC continues to process the data normally and recognize characters, but does not store an ink track. Any existing ink track is deleted.

No-Recognize. CHAREC continues to process the data normally and generate an ink track, but waits for more data when it would usually (with the recognize option) take a character or no-character (cannot interpret) exit.

Halt. CHAREC deletes any existing ink track and takes the halt exit. This allows the user to ignore the character recognizer when taking a control action not involving printing.

Vector Length

The user specifies the vector length to be 2, 4, 6, or 8 raster units, where 1 raster unit = 0.01 in. CHAREC generates (and stores in an ink buffer) a string of vectors of this length to approximate the raw data-point track--this is the ink track. The thinning distance used for data reduction is set equal to the vector length. If the vector length is 8 raster units, the between-character time delay is set to zero. The vector codes generated by CHAREC are for a particular CRT display and are not generally compatible with other displays.

Character Size

The user specifies the normally expected character height and width. This information is used for distinguishing between large and small symbols (e.g., geometric symbol versus not-geometric,) versus ', upper-case c versus lower-case c, etc.), and for character separation. Character separation by position is based on the distance (relative to the normally expected character width) between strokes, and on the positions of strokes within character spaces. Comma and apostrophe are distinguished by the position of the stroke within a character space. CHAREC assumes that the writing surface is divided into a grid of character spaces the size of a normal character. Each such character space's left (or bottom) edge is an integer number of character widths (or heights) from the writing surface's left (or bottom) edge.

Between-Character Time Delay

The user cannot set this delay which is used for separating characters. It is presently a CHAREC parameter (see CHAREC Read-Only Constants, p. 24). However, this time should become a user option by adding it to the list of

CHAREC inputs and changing CHAREC accordingly. This change does not alter the call for CHAREC, but does alter the parent routine's block of data for CHAREC.

III. FUNCTIONAL AND PROCEDURAL DESCRIPTIONS OF
THE PROCESSES AND RCS'S

CHAREC

CHAREC Function

*CHAREC IS GIVEN THE TIME-SEQUENCE OF PEN-DOWNS, STYLUS COORDINATES,
* AND PEN-UPS. IT PERFORMS THREE PRIMARY FUNCTIONS.

*

*1. GENERATE A VECTOR-INK TRACK (SPECIFIED VECTOR SIZE).

*

*2. CALCULATE A SET OF FEATURES FROM THE STYLUS COORDINATE SEQUENCE.

* THESE FEATURES ARE PRESENTED TO THE ROUTINE 'REC' EACH TIME A
* STROKE IS COMPLETED, AND 'REC' TRANSLATES THEM INTO A SUBCHARACTER
* CODE.

* THE FEATURES ARE:

* FOR THE CURRENT STROKE:

* STYLUS DIRECTION SEQUENCE (QUANTIZED TO EAST, NORTH, WEST,
* SOUTH FOR CHARACTERS. QUANTIZED TO 16 DIRECTIONS FOR INK AND
* GEOMETRIC FIGURES).

* THE NO. AND POSITION OF GEOMETRIC CORNERS.

* THE NO. OF TIME-PAUSE CORNERS.

* THE NO. AND POSITIONS OF RELATIVE MAXIMA AND MINIMA IN Y.

* FOR EACH STROKE

* THE POSITIONS OF THE PENDOWN(STARTING) AND PENUP(ENDING) PTS.

* FOR THE CHARACTER

* THE BOUNDS

* THE NO. OF STROKES.

* QUANTIZATION OF DIRECTIONS TO 1 OF 4 PREVENTS THE GENERATION OF
* TOO MANY DESCRIPTIONS OF THE SAME CHARACTER WHILE, WITH THE OTHER
* FEATURES, IS SUFFICIENT FOR DISCRIMINATION.

* MOST FEATURES ARE REPRESENTED AS 1-BYTE NUMBERS TO EASE TESTING.

* FEATURE POSITIONS ARE INDEPENDENT OF WHERE THE CHARACTER IS DRAWN
* ON THE TABLET BECAUSE THEY ARE CALCULATED RELATIVE TO CHARACTER
* BOUNDS.

*

*3. DETERMINE WHEN A CHARACTER IS COMPLETE AND SEND THE CURRENT SUB-
* CHARACTER CODE (ALONG WITH SOME GEOMETRIC INFORMATION--SEE OUTPUTS
* LIST) TO THE USER.

*CHAREC HAS NO INK-TRACK, NO RECOGNIZE, HALT, AND SUPPRESS TABLET
*OPTIONS. NO TRACK, AND NO RECOGNIZE ARE INDEPENDENT.

CHAREC Call

* INST ACHRC,CHRCA,GDATA,GCHPSG,GINDEX,EFINX,ENCHARX,ECHARX,EXTX
* TN,EXTC

*WHERE ALL THE LABELS ARE SELECTED BY THE USER

* ACHRC IS A LINKAGE BETWEEN THE CALLING PROCESS CONTEXT AND CHAREC'S
* CONTEXT

* CHRCA IS A LINK TO CHAREC

* DATA IS THE ADDRESS OF THE INPUTS-OUTPUTS DATA BANK (SEE 'CHAREC
* INPUTS, OUTPUTS')

* CHPSG IS CHAREC'S PSG, 3F

* INDEX IS THE DATA/TIME EXPIRATION INDEX (0 = DATA, 1 = TIME), 1F

* EXITS FINX, NCHARX, CHARX, XTN, XTC ARE DESCRIBED UNDER 'CHAREC
* EXITS'

CHAREC Inputs

*ICP A(INK CCW), NO. OF BYTES DISPLAYED IS IN POSITION 6

*MCH A(MATCH DATA), NOT USED

*KEYB A(KEYBOARD DATA), EQU MCH, NOT USED

*PENU A(PEN UP DATA), EQU MCH, NOT USED

*INPB A(INPUT BUFFER), TIME SEQUENCE OF 12-BIT X, 12-BIT Y

* WHEN EACH IS 10-BIT NO. OF RASTER UNITS, THEN THE 2 LEAST SIG-
* NIFICANT BITS ARE 00. THE NO. OF COORDINATE PAIRS IS VARIABLE
* IT IS GIVEN IN 'INPL'.

*INKB A(INK BUFFER), INK DESCRIPTION IS PLACED HERE WITH BYTE SEQ-

* UENCE 00,LX,X,LYJ,Y,4S,V1,V2,V3,....,00 WHERE EACH SYMBOL

* BETWEEN COMMAS IS 1 BYTE, (LX,X) IS LOAD X, (LYJ,Y) IS LOAD Y

* AND JUMP TO NEW (X,Y), 4S IS ENTER VECTOR MODE WITH VECTOR

* LENGTH CODE S (SEE 'IND') AND THE VI'S ARE VECTOR DIRECTION
* CODES.

*INPL INPUT BUFFER LENGTH, THE NUMBER OF STYLUS COORDINATE PAIRS

* A GROUP OF 7 DATA POINTS ARRIVING IN 30 MS HAS BEEN FOUND CON-
* VENIENT. HALF WORD

*INKL INK BUFFER LENGTH, THE MAXIMUM ALLOWABLE NO. OF BYTES IN THE
* INK DESCRIPTION

* HALF WORD

*IND INDICATORS. A 1 IN THE FOLLOWING BIT POSITIONS INDICATES POS-
* ITIVE ACTIONS. 0=TRACK, 1=RECOGNIZE, 2=PENUP, 3=HALT, 4 AND 5=

* CCDE FOR SIZE OF INK VECTORS (00=2 RASTERS, 01=4 RASTERS, 10=6
* RASTERS, 11=8 RASTERS), 6, 7=NOT ASSIGNED
*BOX EXPECTED CHARACTER WIDTH,HEIGHT: 12-BIT DX, 12-BIT DY
* WHEN EACH IS 10-BIT NO. OF RASTER UNITS, THEN THE 2 LEAST SIG-
* NIFICANT BITS ARE 00.

CHAREC Outputs (Set in CHAREC or REC)

*EP ENDPOINTS, THE PEN DOWN AND PEN UP LOCATIONS OF THE FIRST
* STROKE IN THE CHARACTER. 12-BIT X, 12-BIT Y, 12-BIT X, 12-BIT
* Y. (END OF CHAREC)
*CET GEOMETRIC CENTER OF THE CHARACTER: 12-BIT X, 12-BIT Y
* (END OF CHAREC)
*SIZE ACTUAL CHARACTER WIDTH , HEIGHT: 12-BIT DX, 12-BIT DY
* (END OF CHAREC)
*CHARA CHARACTER CODE--SEE 'RAND CHARACTER CODES' (REC OR CHAREC)
*AR 1-BYTE NO. OF GEOMETRIC CORNERS, 1-BYTE ASPECT RATIO =
* 4 HEIGHT/WIDTH. (END OF CHAREC)

CHAREC Exits

*FINX HALT EXIT
*NCHARX NO CHARACTER EXIT, MORE DATA PENDING (PARALLEL TASK)
*CHARX CHARACTER EXIT, MORE DATA PENDING (PARALLEL TASK)
*XTN TERMINAL NO CHAR EXIT, NO MORE DATA
*XTC TERMINAL CHAR EXIT, NO MORE DATA

CHAREC Parameters

*EACH X OR Y COORDINATE IS A 12-BIT NO. RIGHT JUSTIFIED IN A HALF-WORD
*
*ALL PARAMETERS ARE REFERENCED IN CHAREC. OTHER REFERENCES ARE GIVEN
*IN PARENTHESES. (REC) REFERS TO A REFERENCE IN ANY REC RCS (EXCEPT
*INTERP) IN ADDITION TO REC ITSELF. (ANGLE) REFERS TO THE IN-LINE CODE
*SECTION OF CHAREC CALLED ANGLE.
*
*
*I1 TOP OF DATA BANK, ALSO TRANSLATION OF 'CODE' (ANG4,CHECK)
*PAD CONTAINS THE ADDRESS OF A PLACE IN 'INTERP' (REC,INTERP)
*CODE SEQUENCE OF STYLUS DIRECTIONS--EACH 2 BITS IS A DIRECTION
* 00=E, 01=N, 10=W, 11=S (ANGLE,FN56,ANG4,REC,INTERP)
*XS,YS X,Y COORDINATES OF A SMOOTHED DATA POINT
*XT,YT X,Y COORDINATES OF A THINNED DATA POINT (MXMNS,RELM)

*DX,DY X,Y DISTANCES BETWEEN 2 PTS IN A THINNED TRACK (RELM)
 *MDX,MDY ABSOLUTE VALUES OF DX,DY
 *PANG CODE(SEE CODE) FOR PREVIOUS DIRECTION IN THE TRACK (ANGLE, TURNER,RELM)
 *
 *PACANG CODE (SEE CODE) FOR PREV. ACCEPTED DIRECTION. (ANGLE,TURNER)
 *N NO. DIRECTIONS IN THE LAST STROKE (ANGLE,FN56,ANG4,REC,INTERP)
 *SN TOTAL NO. OF STROKES (CHECK,DELTAS,REC,INTERP)
 *PUP CHAREC INDICATOR. BYTE 0 NOT USED. 1 IN THE FOLLOWING BIT POSITIONS OF BYTE 1 INDICATE POSITIVE ACTIONS: 0,1=NOT USED
 * 2=REQUEST FOR REC, 3=2 CHARACTERS, 4=PEN-UP-DELAY HAS HAPPENED
 * 5=CLOCK HAS BEEN CALLED, 6=TAKE HALT EXIT, 7=NOT FIRST PENDOWN
 *INKIND NO. BYTES OF INK
 *PQUAD CODE(NE=00,NW=01,SW=10,SE=11) FOR QUADRANT OF PREVIOUS DIRECTION (ANGLE)
 *
 *BR56 INDEX BASED ON DIRECTIONS 5 AND 6, VALUES 0-16 (FN56,INTERP)
 *DXC,DYC X,Y EXTENTS OF CHARACTER (MXMNC,REC)
 *XRC,XLC RIGHT,LEFT EXTREMES OF CHARACTER (DELTAS,MXMNC,REC)
 *YTC,YBC TOP,BOTTOM EXTREMES OF CHAR. (DELTAS,QMM,MXMNC,BSRPRM,BHITE)
 *ASPR ASPECT RATIO = 4*DYC/DXC (INTERP)
 *NT NO. OF THINNED POINTS (TCNR)
 *NTC NT AT WHICH LAST TIME-CORNER OCCURRED (TCNR)
 *INKC NO. OF BYTES OF INK IN THE FIRST CHARACTER
 *XYE,XYS CODED(SEE BELOW) SEQUENCE OF POSITIONS OF END,START POINTS OF STROKES--1/2 WORD FOR EACH STROKE ENDPT,STARTPT.(DELTAS,REC, INTERP)
 *
 *
 *

			YTC		
		3	2	1	0
	XLC	7	6	5	4
		11	10	9	8
		15	14	13	12
					YBC

*WIDTH,HEIGHT EXPECTED NORMAL CHARACTER WIDTH, HEIGHT--SEE BOX IN INPUTS LIST (PSTEST)
 *
 *YCENT Y COORDINATE OF CENTER OF PREVIOUSLY OUTPUTTED CHARACTER
 *PCHAR CODE FOR PREVIOUSLY OUTPUTTED CHARACTER
 *CUSP TEMPORARY STORAGE (TCNR,REC,INTERP)
 *NCUSP NO. GEOMETRIC CORNERS (INTERP)
 *NPTS NO. RAW DATA PTS. SINCE LAST THINNED PT. (TCNR)
 *DEL MINIMUM X OR Y DISTANCE BETWEEN THINNED POINTS (DERIVED FROM INC--SEE INPUTS). (RELM)
 *
 *P CODE INDICATING TYPE OF PREVIOUS STROKE OR STROKES. 1=DOWN VERT, 2=HORIZ, 3=7-LIKE, 4=V-LIKE, 5=C-LIKE, 6=O-LIKE, 7=U-LIKE, 8=2 HORIZS., 9=UP VERT, 10=1 VERT AND 1 HORIZ, 11=2 VERTS. (REC,INTERP)
 *
 *CHAR CHARACTER CODE(SEE CHARA IN LIST OF OUTPUTS) (REC,DCT,INTERP)
 *TEMP TEMPORARY STORAGE (REC,INTERP)
 *TINK NOT USED
 *XSP,YSP X,Y COORDINATES OF SEQUENCE OF STARTING PTS. OF STROKES--1/2 WORD EACH (DELTAS,BSVM)
 *
 *XEP,YEP X,Y COORDINATES OF SEQUENCE OF ENDING PTS. OF STROKES--1/2 WORD EACH. (DELTAS,REC)
 *

```

*ALXYJ  7 BYTES CONTAINING CO,LX,X,LYJ,Y,ENTER VECTOR MODE,00.  GOES
*        INTO INK BUFFER(SEE INKB IN LIST OF INPUTS)
*XL,YL  RAW DATA POINT COORDINATES
*XLO,YLO XL,YL USED BY TRAVEC (CORNER)
*AX,AX1,AX2,AX3  16-DIRECTIONS USED FOR GEOMETRIC CORNERS (CORNER)
*AX01,AX02,AX12,AX23
*        DIFFERENCES BETWEEN 16-DIRECTIONS (CORNER)
*NC      NO. GEOMETRIC CORNERS (CORNER)
*C       INTERNAL CORNER PARAMETER (CORNER)
*DYM     3/2 EXPECTED NORMAL CHARACTER HEIGHT--SEE BOX IN INPUTS LIST
*        (BHTE,PSTEST,TILDT)
*DXS,DYS X,Y EXTENTS OF CURRENT STROKE (MXMNS)
*XRS,XLS RIGHT,LEFT EXTREMES OF CURRENT STROKE (MXMNC,MXMNS)
*YTS,YBS TOP,BOTTOM EXTREMES OF CURRENT STROKE (MXMNC,MXMNS)
*CENT    X CENTER,Y CENTER--SEE CET IN OUTPUT LIST (RAZE,PSTEST)
*MVC     ADJUSTABLE MVC INSTRUCTION
*TTURN   CODE(SEE CODE) FOR A SINGLE DIRECTION TURN (TURNER)
*TURN    CODED(SEE CODE) SEQUENCE OF SINGLE DIRECTION TURNS (INTERP)
*XC,YC   SEQUENCE OF X,Y COORDINATES OF GEOMETRIC CORNERS (CORNER)
*        (XC=BSSM, YC=BSRPRM)
*DO THRU D15  NO. OF OCCURANCES OF DIRECTIONS 0 THRU 15 (SYMT)
*DN       SUM OF DO THRU D15 (SYMT)
*NTCUSP   NO. OF TIME CORNERS (TCRNR,REC,INTERP)
*PNPTS    PREVIOUS NPTS (TCRNR)
*PYMAX,PYMIN  Y LOCATION OF PREVIOUS RELATIVE Y MAX, MIN (RELM)
*NYMAX,NYMIN  NO. OF RELATIVE Y MAX,MIN (RELM,REC,INTERP)
*YMAX,YMIN    SEQUENCE OF Y LOCATIONS OF RELATIVE Y MAX,MIN FOR THE
*              CURRENT STROKE--1/2 WORD EACH (QMM,RELM,INTERP)
*QYMAX,QYMIN  SEQUENCE OF CODED(YTC,CO,01,02,03,YBC) QUANTIZED YMAX,
*              YMIN--1 BYTE EACH (QMM,REC,INTERP)
*              ALSO USED AS AN INDICATOR(RELM)
*PYMXX,PYMNX  X LOCATION OF PREVIOUS RELATIVE Y MAX,MIN (RELM)
*YMAXX,YMINX  SEQUENCE OF X LOCATIONS OF RELATIVE Y MAX,MIN FOR THE
*              CURRENT STROKE--1/2 WORD EACH (RELM,INTERP,BSMNW,BTEST3)

```

CHAREC Read-Only Constants

```

*TIME    PEN-UP-DELAY TIME FOR CLOCK, F'0100' = 0.1 SECOND
*LXYJ    LOAD X, 00, LOAD AND JUMP TO Y ,00
*CDOT    THE CHARACTER CODE FOR A POINT
*HEX10   THE DECIMAL EQUIVALENT OF HEX 10
*HEX90   THE DECIMAL EQUIVALENT OF HEX 90

```

CHAREC Sequence of Information Processing

```

****START****
*

```

```
*GO TO NEW CHARACTER ENTRY , THEN CONTINUE
*
****NEW CHARACTER ENTRY****
*
*INITIALIZE
*RETURN
*
*
****NEW DATA POINT ENTRY****
*
*IF HALT DESIRED, GO TO FINISH ENTRY 1
*IF PEN UP, GO TO PEN UP SIGNAL ENTRY
*IF NOT FIRST PEN DOWN, GO TO MIDSTROKE NEW DATA POINT ENTRY
*
****NEW STROKE ENTRY****
*
*INITIALIZE
*SET NOT FIRST PEN DOWN INDICATOR
*SET UP STARTING POINT AND INK-VECTOR SIZE IN INK BUFFER
*SET UP THINNING DISTANCE
*IF INK DESIRED, SET DISPLAY COUNT
*
****MIDSTROKE NEW DATA POINT ENTRY****
*
*'THIN' DETERMINES IF THE CURRENT DATA PT. IS SUFFICIENTLY FAR FROM THE
* PREVIOUS THINNED PT.
* NO, GO TO ANGLE SECTION-END
*'TCRNR' DETERMINES IF A TIME-PAUSE CORNER HAS OCCURRED
*CALCULATE INCREMENT BETWEEN NEW AND OLD THINNED POINTS
*'TRAVEC' CALCULATES 16-DIRECTION FOR INK
*IF NO INK-TRACK DESIRED, ZERO (SET TO 2) DISPLAYED INK COUNT, THEN
* SKIP TO 'CORNER' CALL
*STORE INK IF NEW THINNED PT. IS SUFFICIENTLY FAR FROM THE LAST PT. IN
* THE INK TRACK.
*'CORNER' DETERMINES IF A GEOMETRIC CORNER HAS OCCURRED AND CALCULATES
* ITS POSITION.
*'MXMNS' UPDATES STROKE BOUNDS
*'RELM' UPDATES RELATIVE MAXIMA AND MINIMA
*
****ANGLE SECTION-START****
*
*DETERMINE QUADRANT OF DIRECTION
*'HYST' MODIFIES DIRECTION FOR HYSTERESIS ZONE
*DETERMINE WHETHER EAST, NORTH, WEST, OR SOUTH
*IF NOT THE SAME AS THE PREVIOUS DIRECTION, 'TURNER' DETERMINES IF THIS
* WAS A 180 DEGREE TURN, THEN GO TO WAIT FOR NEXT DATA POINT
*IF THE SAME, PLACE IN DIRECTION SEQUENCE
*
****ANGLE SECTION-END****
*
*UPDATE THE DATA POINT COUNTER
```

```
*IF ALL DATA POINTS IN THE INPUT BUFFER HAVE NOT BEEN EXAMINED, GO TO
* MIDSTROKE NEW DATA POINT ENTRY
*OTHERWISE WAIT FOR NEXT DATA POINT GROUP
*
****WAIT FOR NEXT DATA POINT GROUP****
*
*(WAITING FOR A DATA POINT GROUP DOES NOT TIE UP THE CPU)
*WHEN NEW DATA POINT GROUP ARRIVES, THEN
*IF HALT DESIRED, GO TO FINISH ENTRY 3
*SET UP INK-VECTOR SIZE AND THINNING DISTANCE
*NEGATE REC REQUEST, 2 CHARACTERS, AND PEN-UP-DELAY INDICATORS
*GO TO NEW DATA POINT ENTRY
*
****PEN UP SIGNAL ENTRY****
*
*NEGATE NOT FIRST PENDOWN INDICATOR
*IF STROKE IS A DOT, 'DOT' CHECKS FOR POSSIBLE SCRIPT I OR J
* IF YES, GO TO MULTI-STROKES ENTRY
*IF THIS IS THE ONLY STROKE, GO TO MULTI-STROKES ENTRY
*IF THE PREVIOUS SUBCHARACTER CANNOT BE COMBINED WITH ANY STROKE, GO TO
* THE MULTI-STROKES ENTRY
*'ANG4' AND 'CHECK' DETERMINE IF THE PREVIOUS SUBCHARACTER CAN BE
* COMBINED WITH THIS STROKE
* IF NOT, GO TO THE MULTI-CHARACTERS ENTRY
*IF CURRENT STROKE IS A COMMA, GO TO MULTI-CHARACTERS ENTRY
*IF CURRENT STROKE AND PREVIOUS SUBCHARACTER ARE NOT GEOMETRICALLY
* CLOSE ENOUGH TO BE COMBINED AS A CHARACTER, GO TO MULTI-CHARACTERS
* ENTRY. (IF IT IS NOT DESIRED TO SEPARATE CHARACTERS BASED ON THEIR
* POSITIONS, REPLACE 'PTEST LA R7,1 WITH 'PTEST EQU *' AND DELETE ALL
* THE FOLLOWING CODE UP TO, BUT NOT INCLUDING, THE LINE LABELLED
* 'CASE1').
*
****MULTI-STROKES ENTRY****
*
*'MXMNC' UPDATES CHARACTER BOUNDS
*
****NEW CHARACTER PARAMETERS ENTRY****
*
*SET FIRST CHARACTERS INK COUNT TO TOTAL INK COUNT
*'DELTAS' QUANTIZES STARTING PT. AND ENDING PT. LOCATIONS
*'QMM' QUANTIZES RELATIVE Y MAX AND Y MIN LOCATIONS
*'ANG4' TRANSLATES FIRST 4 DIRECTIONS TO A 1-BYTE CODE CORRESPONDING TO
* A SET OF CHARACTERS
*'FN56' TRANSLATES DIRECTIONS 5 AND 6 TO A 4-BIT CODE
*COMPUTE ASPECT RATIO
*STORE NO. GEOM-CORNERS, AND NO. TIME-CORNERS
*COMPUTE CENTER
*IF NO. OF STROKES IS NOT 2, SKIP AROUND TESTS FOR SCRIPT I AND J
*IF PREV. SUBCHARACTER IS SCRIPT I, GO TO REC EXIT
*IF PREV. SUBCHARACTER IS SCRIPT J, 'RAZE' INCREASES Y CENTER
*IF NO. DIRECTIONS GTR 15, CHAR IS SCRUB, GO TO REC EXIT
```

```
*IF NO. DIRECTIONS NOT GTR 8 GO TO REC CALL
*IF NO. DIRECTIONS GTR 12, OR CHARACTER IS LARGE, SET CHAR=SCRUB, GO TO
* REC EXIT
*
****REC CALL****
*
*CALL REC, THEN GO TO REC EXIT
*
****MULTI-CHARACTERS ENTRY****
*
*IF FIRST CHARACTER INK COUNT=TOTAL INK COUNT, I.E. IF THERE IS ONLY 1
* CHARACTER PENDING, GO TO RESTORE INK COUNT ENTRY
*SET INK COUNT TO INK COUNT LESS FIRST CHARACTER INK COUNT, I.E. TO 2ND
* CHARACTER INK COUNT
*ZERO (SET TO 2) DISPLAYED INK COUNT, AND SAVE PREVIOUS DISPLAYED INK
* COUNT.
*MOVE 2ND CHARACTER INK TO THE HEAD OF THE INK BUFFER.
*IF NO TRACK DESIRED, GO TO ZERO INK COUNT ENTRY
* CHARACTER INK COUNT, I.E. TO 2ND CHARACTER INK COUNT.
*
****ZERO INK COUNT ENTRY****
*
*SET TOTAL INK COUNT TO ZERO
*
****DON'T RESTORE ENTRY****
*
*SET REC REQUEST AND 2 CHARACTERS INDICATORS
*
****2 CHARACTERS ENTRY****
*
*IF NO RECOGNITION IS DESIRED, GO TO WAIT FOR NEXT DATA POINT GROUP.
*IF CHARACTER IS NOT RECOGNIZABLE, GO TO NO CHARACTER ENTRY
*IF REC HAS NOT BEEN REQUESTED, GO TO TERMINAL CHARACTER ENTRY
*INITIATE PARALLEL PROCESS. HIGH PRIORITY TAKES CHARACTER EXIT. LOW
* PRIORITY GOES TO RESET FOR NEW CHARACTER ENTRY
*
****RESET FOR NEW CHARACTER ENTRY****
*
*RESET CHAR SIZE, STARTING AND ENDING POINT LOCATIONS, CENTER, ETC.
*GO TO NEW CHARACTER PARAMETERS ENTRY
*
****REC EXIT****
*
*IF NO. DIRECTIONS GTR 8, AND CHARACTER IS NOT SCRIPT, SET CHAR=SCRUB
*NEGATE REC REQUEST AND 2 CHARACTERS INDICATORS
*SET TIME/DATA EXPIRATION INDEX TO TIME
*IF DESIRED INK-VECTOR SIZE IS 8 RASTERS, GO TO CLOCK EXPIRED ENTRY
*SET CLOCK HAS BEEN CALLED INDICATOR
*CALL CLOCK, THEN GO TO CLOCK TURNED OFF OR CLOCK EXPIRED
*
****CLOCK EXPIRED (DUE TO RUNNING LONGER THAN 'TIME') ENTRY****
```

```
*
*TURN OFF CLOCK (SET)
*PAUSE, THEN GO TO CLOCK TURNED OFF ENTRY
*
****CLOCK TURNED OFF (DUE TO PENDOWN) ENTRY****
*
*IF HALT DESIRED, GO TO FINISH ENTRY 3
*NEGATE CLOCK CALLED INDICATOR
*IF TAKE FINISH EXIT INDICATOR IS SET, GO TO FINISH ENTRY 2
*GO TO SET UP OUTPUTS ENTRY, THEN RETURN HERE
*IF 2 CHARACTERS INDICATOR IS SET, GO TO 2 CHARACTERS ENTRY
*IF TIME/DATA EXPIRATION INDEX IS SET TO DATA, GO TO NEW DATA PT. ENTRY
*RESET ALL INTERNAL INDICATORS
*GO TO 2 CHARACTERS ENTRY
*
****FINISH ENTRY 1****
*
*IF CLOCK HAS BEEN CALLED, GO TO FINISH ENTRY 2
*SET TAKE FINISH EXIT INDICATOR
*GO TO CLOCK EXPIRED ENTRY
*
****SET UP OUTPUTS ENTRY****
*
*MOVE APPROPRIATE INTERNAL VALUES TO OUTPUTS
*RETURN
*
****NO CHARACTER ENTRY****
*
*IF REC HAS NOT BEEN REQUESTED, GO TO TERMINAL NO CHAR ENTRY
*INITIATE PARALLEL PROCESS. HIGH PRIORITY TAKES NO CHAR EXIT. LOW
* PRIORITY GOES TO RESET FOR NEW CHARACTER ENTRY
*
****FINISH ENTRY 2****
*
*GO TO SET UP OUTPUTS ENTRY, THEN RETURN HERE
*
****FINISH ENTRY 3****
*
*GO TO SET UP INK ENTRY, THEN RETURN HERE
*TAKE HALT EXIT
*
****TERMINAL CHARACTER ENTRY****
*
*GO TO SET UP INK ENTRY, THEN RETURN HERE
*TAKE TERMINAL CHARACTER EXIT
*
****TERMINAL NO CHAR ENTRY****
*
*GO TO SET UP INK ENTRY, THEN RETURN HERE
*TAKE TERMINAL NO CHAR EXIT
*
```

****SET UP INK ENTRY****

*

*IF DESIRED INK-VECTOR SIZE IS 8 RASTERS, RETURN

*ZERO (SET TO 2) DISPLAYED INK COUNT

*RETURN

*

****END OF CHAREC****

CHAREC Program Listing

```
        USING XR1,R1
        USING XR3,R3
        USING XR4,R4
        SVCS
        REGS
CD1      DSECT
XR1      DS      3F
AREC     DS      1F
CLK1     DS      1F
DATA     DS      1F
WAITBX   DS      1F
INDEX    DS      1F
FINX     EQU     0
NCHARX   EQU     4
CHARX     EQU     8
XTN      EQU     12
XTC      EQU     16
CD4      DSECT
XR4      DS      0F
ICP      DS      1F
MCH      DS      1F
KEYB     EQU     MCH
PENU     EQU     MCH
INPB     DS      1F
INKB     DS      1F
INPL     DS      1H
INKL     DS      1H
EP       DS      2F
CET      DS      1F
SIZE     DS      1F
IND      DS      1C
CHARA    DS      1C
AR       DS      1H
BCX      CS      1F.
CD3      DSECT
XR3      DS      0F
IL       DS      1F
PAD      DS      1F
CODE     CS      1F
XS       DS      1H
YS       DS      1H
XT       CS      1H
```

TIME/DATA EXPIRATION INDEX

TERMINAL NO CHAR EXIT
TERMINAL CHAR EXIT

A(INK CCW)
A(MATCH DATA)
A(KEYBOARD DATA)
A(PEN UP DATA)
A(INPUT BUFFER)
A(INK BUFFER)
INPUT BUFFER LENGTH
INK BUFFER LENGTH
END POINTS
CENTER
ACTUAL CHARACTER SIZE
INDICATORS
CHARACTER
CORNERS, ASPECT RATIO
MAX CHARACTER SIZE

YT	DS	1H
DX	DS	1H
DY	DS	1H
MDX	DS	1H
MDY	DS	1H
PANG	DS	1H
PACANG	DS	1H
N	CS	1H
SN	DS	1H
PUP	DS	1H
INKIND	DS	1H
PQUAD	DS	1H
BR56	CS	1H
DXC	DS	1H
DYC	DS	1H
XRC	DS	1H
XLC	DS	1H
YTC	DS	1H
YBC	DS	1H
ASPR	DS	1H
NT	DS	1H
NTC	DS	1H
INKC	DS	1H
XYE	DS	10C
XYS	DS	10C
WIDTH	DS	1H
HEIGHT	DS	1H
	DS	2F
YCENT	DS	1H
PCHAR	DS	1C
CUSP	DS	1C
NCUSP	DS	1H
NPTS	DS	1H
DEL	DS	1H
P	DS	1C
CHAR	DS	1C
TEMP	CS	1C
TINK	CS	5C
XSP	DS	10C
YSP	DS	10C
XEP	DS	10C
YEP	DS	10C
ALXYJ	DS	8C
XL	DS	1H
YL	DS	1H
XLO	DS	1H
YLO	DS	1H
AX3	DS	1H
AX2	DS	1H
AX1	CS	1H
AX	DS	1H

AX23	DS	1H
AX12	DS	1H
AX01	DS	1H
AX02	DS	1H
NC	DS	1H
C	DS	1H
DYM	DS	1H
DXS	DS	1H
DYS	DS	1H
XRS	DS	1H
XLS	DS	1H
YTS	DS	1H
YBS	DS	1H
CENT	DS	1F
MVC	DS	6C
TTURN	DS	1H
TURN	DS	1F
XC	DS	10C
YC	DS	10C
D0	DS	1H
D1	DS	1H
D2	DS	1H
D3	DS	1H
D4	DS	1H
D5	DS	1H
D6	DS	1H
D7	DS	1H
D8	DS	1H
D9	DS	1H
D10	DS	1H
D11	DS	1H
D12	DS	1H
D13	DS	1H
D14	DS	1H
D15	DS	1H
CN	DS	1H
NTCUSP	DS	1H
PNPTS	DS	1H
PYMAX	DS	1H
PYMIN	DS	1H
NYMAX	DS	1H
NYMIN	DS	1H
YMAX	DS	10H
YMIN	EQU	YMAX+10
QYMAX	DS	10C
QYMIN	EQU	QYMAX+5
PYMXX	DS	1H
PYMNX	DS	1H
YMAXX	DS	10H
YMINX	EQU	YMAXX+10
CHAREC	PROCS	CLEAR=5,CNTX=9,AUTO=86,PROLG=XCHRX,ID=9000021F

```

TIME      DC      F'0300'
LXYJ      DC      X'54006000'
CDDT      DC      X'80'
HEX10     DC      F'16'
HEX90     DC      F'144'
MOVER     MVC      0(C,R6),0(R7)
ANG56     DC      V(FN56)
ANG4A     DC      V(ANG4)
DELT      DC      V(DELTAS)
RECA      DC      V(REC)
          DC      X'8000021C'
SMTH      DC      V(SMOOTH)
THINN     DC      V(THIN)
MAXMNS    DC      V(MXMNS)
HYSTR     DC      V(HYST)
CLK2      DC      V(CLOCK)
          DC      X'80000000'
MAXMNC    DC      V(MXMNC)
TRAVC     DC      V(TRAVEC)
CORNR     DC      V(CORNER)
TURN      DC      V(TURNER)
CHECKA    DC      V(CHECK)
RELMA     DC      V(RELM)
QMA       DC      V(QMM)
DOTA      DC      V(DOT)
TCRNRA    DC      V(TCRNR)
RAZEA     DC      V(RAZE)
XCHRX     PROLG

```

*
*
*
*
*

****START****

```

      CLEAR PSG=WAITBX,CNTX=F
      MVI   ALXYJ,X'00'
      MVI   ALXYJ+5,X'40'
      MVI   ALXYJ+6,X'00'
      MVC   MVC(6),MOVER
      BAL   R15,TOP
      B     WAITZ

```

*
*
*
*
*

****NEW CHARACTER ENTRY****

```

TOP      XC      I1(4),I1
          XC      PAD(4),PAD
          XC      SN(2),SN
          XC      INKIND(2),INKIND
          XC      INKC(2),INKC
          XC      PUP(2),PUP

```

STROKES
INK INDEX

```

XC      XRC(2),XRC
XC      YTC(2),YTC
XC      CHAR(1),CHAR
XC      P(1),P
XC      TTURN(2),TTURN
XC      DO(32),DO
XC      DN(2),DN
LA      R6,1024
SLL     R6,2
STH     R6,XLC
STH     R6,YBC
L       R4,DATA
L       R7,CET
STH     R7,YCENT
L       R4,DATA
MVC     PCHAR(1),CHARA
L       R4,DATA
L       R7,BOX
LR      R8,R7
STH     R8,HEIGHT
SRL     R8,16
STH     R8,WIDTH
LR      R8,R7
SRL     R8,1
AR      R7,R8
STH     R7,DYM
BR      R15

```

STORE PREV CENT Y IN YCENT

STORE PREV CHAR IN PCHAR

STORE MAX ALLOW DY IN DYM

1 1/2 CHARACTER HEIGHT

```

*
*
*****NEW DATA POINT*****
*
*

```

```

WAITZ   L      R4,DATA
        TM     IND,X'10'
        BC     1,FIN
        TM     IND,X'20'
        BC     1,IND2
        LH     R12,INKIND
        TM     PUP+1,X'01'
        BC     1,PENDWN

```

ENTRY FROM WAIT BOX

B TO FIN IF HALT

B TO IND2 IF P.U.
R12=INK BUFF IND

```

*
*
*****NEW STROKE*****
*
*

```

```

        LH     R6,SN
        LA     R8,5
        CR     R6,R8
        BC     4,SNLSS5
        SR     R6,R6
SNLSS5  LR     R8,R6

```

STROKES CVERFLOW TEST

```

SLL    R8,1
LA     R6,1(R6)
STH    R6,SN
*INITIALIZATION
XC     NC(4),NC
LA     R6,16
STH    R6,AX1
STH    R6,AX
MVC    AX3(4),AX1
XC     NTCUSP(2),NTCUSP
XC     NPTS(2),NPTS
XC     NT(2),NT
XC     NTC(2),NTC
LA     R7,20
STH    R7,PNPTS
XC     NYMAX(2),NYMAX
XC     NYMIN(2),NYMIN
XC     N(2),N
XC     C(2),C
MVI    QYMAX,X'01'
MVI    QYMIN,X'01'
LA     R6,4
STH    R6,PANG
STH    R6,PACANG
STH    R6,PQUAD
XC     XRS(2),XRS
XC     YTS(2),YTS
LA     R6,1024
SLL    R6,2
STH    R6,XLS
STH    R6,YBS
OI     PUP+1,X'01'
L      R4,DATA
L      R10,INPB
LH     R7,0(R10)
STH    R7,XS
STH    R7,XT
STH    R7,XSP(R8)
STH    R7,XL
STH    R7,PYMAX
STH    R7,PYMIN
LH     R7,2(R10)
STH    R7,YS
STH    R7,YT
STH    R7,YSP(R8)
STH    R7,YL
STH    R7,PYMAX
STH    R7,PYMIN
LA     R13,4
B      REBUFF
CLNBUF L      R4,DATA

```

INITIALIZATION

```

# TIME CORNERS
# PTS BET. THIN PTS

```

```

PREV NPTS
# REL MAX
# REL MIN
# ANGLES
CORNER INDEX

```

```

PREV ANG =4
PREV ACC ANG =4
PREV QUADRANT =4

```

```

1ST XRAW
1ST SMOOTH X
1ST X THIN
X STARTING POINT

```

```

X CF POT. YMAX
X CF POT YMIN
1ST Y RAW
1ST Y SMOOTH
1ST Y THIN
Y STARTING POINT

```

```

POTENTIAL Y MAX
POTENTIAL Y MIN

```

```

      L      R7,ICP
      LA     R8,2
      STH   R8,6(R7)
REBUFF L      R4,DATA
      LH     R15,INKL
      LA     R6,7(R12)
      CR     R6,R15
      BC     4,INKLOK
      SR     R12,R12
      B      CLNBUF
INKLOK LH     R6,XL
      SRL    R6,2
      LH     R7,YL
      SRL    R7,2
      SLL    R6,16
      OR     R6,R7
      C      R6,LXYJ
      ST     R6,TEMP
      MVC    ALXYJ+1(4),TEMP
      L      R4,DATA
      IC     R15,INC
      LA     R14,12
      NR     R15,R14
      LA     R14,64
      LR     R6,R15
      LA     R6,4(R6)
      SLL    R6,1
      STH    R6,DEL
      CR     R15,R14
      STC    R15,ALXYJ+5
      L      R15,INKB
      LA     R15,0(R12,R15)
      MVC    0(7,R15),ALXYJ
      SR     R6,R6
      STC    R6,7(R15)
      L      R4,DATA
*TEST FOR NO INKING
      TM     IND,X*80'
      BC     8,ENTER1
      L      R7,ICP
      LH     R15,6(R7)
      LA     R15,6(R15)
      LA     R12,6(R12)
      LA     R6,2(R12)
      CR     R15,R6
      BC     2,R15GTR
      LA     R15,1(R15)
R15GTR STH    R15,6(R7)
      B      ENTER

```

SET VECTOR SIZE

SET DEL, MIN THIN DIFF

*
*

****MIDSTRCKE NEW DATA POINT****

*

*

PENDWN	SR	R13,R13	
ENTER	SR	R6,R6	SMOOTH TRACK
	CR	R12,R6	
	BC	8,CLNBUF	
ENTER1	L	R4,DATA	
	L	R10,INPB	
	LH	R6,0(R13,R10)	
	STH	R6,XS	
	LA	R13,2(R13)	
	LH	R6,0(R13,R10)	
	STH	R6,YS	
	LH	R8,NPTS	
	LA	R8,1(C,R8)	
	STH	R8,NPTS	
	LH	R7,YT	Y THIN THIN TRACK
	LH	R8,DEL	MIN THIN DIFF
	RCS	THINN,E**4	
	CH	R7,YT	
	BC	8,YSMALL	
	B	OK	
YSMALL	LH	R6,XS	X SMOOTH
	LH	R7,XT	X THIN
	LH	R8,DEL	MIN THIN DIFF
	RCS	THINN,E**4	
	CH	R7,XT	
	BC	8,SMALL	
*HERE IF NEW POINT ACCEPTED IN THIN TRACK			
OK	EQU	*	
	RCS	TCRNRA,I11,E**4	
	LH	R11,YS	
	LH	R10,YT	
	STH	R11,YT	
	SR	R11,R10	
	STH	R11,DY	ST DELTA Y
	LPR	R11,R11	
	STH	R11,MDY	ST MAG(DELTA Y)
	LH	R11,XS	
	LH	R7,XT	
	STH	R11,XT	
	SR	R11,R7	
	STH	R11,DX	ST DELTA X
	LPR	R11,R11	
	STH	R11,MDX	ST MAG(DELTA X)
NUINK	LH	R7,XT	STORE NEW INK
	LH	R9,YT	
	LH	R10,XL	
	LH	R11,YL	
	STH	R10,XLO	

	STH	R11,YLO	
	L	R4,DATA	
	L	R15,INC	
	SRL	R15,26	
	LA	R14,3	
	NR	R15,R14	
	LA	R15,1(R15)	
	RCS	TRAVC,E*+4	
	C	R0,HEX10	
	BC	8,INKST	
	LR	R8,R0	INCREMENT CNT FOR THIS DIRECTION
	SLL	R8,1	
	LH	R7,DO(R8)	
	LA	R7,1(R7)	
	STH	R7,DO(R8)	
	LH	R7,DN	INCREMENT TOTAL COUNT
	LA	R7,1(R7)	
	STH	R7,DN	
	STH	R0,AX	
	STH	R10,XL	
	STH	R11,YL	
	L	R4,DATA	
	LH	R15,INKL	
	BCT	R15,A1	
A1	L	R4,DATA	TEST FOR NO INKING
	TM	INC,X'80'	
	BC	1,STOINK	
	L	R4,DATA	
	L	R7,ICP	
	LA	R8,2	
	STH	R8,6(R7)	
	SR	R12,R12	
	B	NOSTO	CLEAR INK COUNT
STOINK	EQU	*	
	L	R4,DATA	
	L	R7,ICP	
	LH	R8,6(R7)	
	LA	R8,1(R8)	
	STH	R8,6(R7)	
	L	R4,DATA	
	O	R0,HEX90	
	L	R6,INKB	
	STC	R0,0(R12,R6)	
	SRL	R0,8	
	LA	R12,1(R12)	
	STC	R0,1(R6,R12)	
NOSTO	EQU	*	
*GEOMETRIC CORNER DETECTOR			
	LH	R7,AX	
	SH	R7,AX1	
	LPR	R7,R7	

```

      STH   R7,AX01
      CLI   AX3+1,X'10'
      BC    8,SHIFT
      RCS   CORNR,I11,E**4
SHIFT  MVC   AX3(12),AX2
      CR    R12,R15
      BC    4,NUINK
      SR    R12,R12
*UPDATE STROKE BOX SIZE AND LOCATION
INKST  RCS   MAXMNS,I11,E**4
*UPDATE RELATIVE MAX AND MINS
      RCS   RELMA,I11,E**4
*
*
****ANGLE SECTION START****
*
*
*   DETERMINE QUADRANT
*
      LH    R6,DX
      LTR   R6,R6
      BC    4,DXNEG
      LH    R6,DY
      LTR   R6,R6
      BC    4,DYNEG
*DX,DY POS, QUAD=0
      SR    R6,R6
      B     QTEST
*DX POS, DY NEG, QUAD=3
DYNEG  LA    R6,3
      B     QTEST
DXNEG  LH    R6,DY
      LTR   R6,R6
      BC    4,DYNEGG
      LA    R6,1
      B     QTEST
DYNEGG LA    R6,2
      B     QTEST
*
*   DETERMINE DIRECTION
*   AND CHECK FOR 2 EQUAL SUCCESSIVE ANGLES
*
QTEST  CH    R6,PQUAD
      BC    8,QQPQ
      STH   R6,PQUAD
      LH    R6,MDX
      CH    R6,MDY
      BC    4,ODCANG
EVANG  LH    R6,DX
      LTR   R6,R6
      B     IF QUAD=PREVQ
      SET   PQUAD=QUAD
      B     IF MDX LESS THAN MDY
      ANG  EVEN, TEST SIGN(DX)

```

	BC	4,ANG2		
	SR	R6,R6		DX POS, ANG=0 RIGHT
	B	PRVANG		
ANG2	LA	R6,2		DX NEG, ANG=2 LEFT
	B	PRVANG		
ODDANG	LH	R6,DY		ANG ODD, TEST SIGN(DY)
	LTR	R6,R6		
	BC	4,ANG3		
	LA	R6,1		
	B	PRVANG		
ANG3	LA	R6,3		DY NEG, ANG=3 DOWN
	B	PRVANG		
QEQPQ	STH	R6,PQUAD		QUAD=PREV QUAD, HYSTERESIS
	LH	R6,PANG		
	LA	R7,1		
	NR	R6,R7		AND PANG WITH 1
	BC	8,EVPANG		B IF PREV ANG EVEN
*PREV ANGLE ODD				
	LH	R6,MDX		
	LH	R7,MDY		
	RCS	HYSTR,E**4		
	BC	2,EVANG		B IF 3/4 MDX GTR MDY
	B	PRVTST		
EVPANG	LH	R6,MDY		PREV ANGLE EVEN
	LH	R7,MDX		
	RCS	HYSTR,E**4		
	BC	2,ODDANG		B IF 3/4 MDY GTR MDX
	B	PRVTST		
PRVANG	CH	R6,PANG		DCES ANG=PREV ANG
	BC	8,PRVTST		B IF ANG=PANG
	LR	R9,R6		
	RCS	TURN,III,E**4		
	LR	R6,R9		
	STH	R6,PANG		SET PANG=ANG
	B	SMALL		
PRVTST	EQU	*		
	LH	R6,PANG		ANG=PREVANG
	CH	R6,PACANG		
	BC	8,SMALL		
	STH	R6,PACANG		
** ANGLES OVERFLOW TEST				
	LH	R7,N		# OF ANGLES
	LA	R8,15		
	CR	R7,R8		
	BC	4,NLOW		N LESS THAN 15
	MVI	CHAR,X'72'		CHAR IS A SCRUB
	SR	R7,R7		
NLOW	L	R10,CODE		
	LH	R9,TURN		
	L	R11,TURN		
	LA	R8,16		

```

SHFT      SR      R8,R7
          SRL      R10,2
          SRL      R11,2
          BCT      R8,SHFT
          SLL      R10,2
          SLL      R11,2
          LH       R6,PANG
          CR       R10,R6
          CR       R11,R9
          LA       R8,15
          SR       R8,R7
SHFT1     SLL      R10,2
          SLL      R11,2
          BCT      R8,SHFT1
          ST       R10,CODE
          ST       R11,TURN
          MVI      TTURN+1,X'00'
          LA       R7,1(R7)
          STH      R7,N
          B        SMALL

```

INC N

```

*
*
****ANGLE SECTION END****

```

```

SMALL     LA       R13,2(C,R13)
          L        R4,DATA
          LH       R10,INPL
          CR       R13,R10
          BC       4,ENTER
          STH      R12,INKIND

```

HERE PROCESSING OF NEW RAW COMP

GET NEXT POINT
KEEP INK BUFF IND

```

*
*
****WAIT FOR NEXT DATA POINT GROUP****

```

```

WATR1     WATE     PSG=WAITBX,CNTX=F
          L        R4,DATA
          TM       IND,X'10'
          BC       1,GOFINX
          L        R6,INKB
          IC       R15,IND
          LA       R14,X'0C'
          NR       R15,R14
          LA       R14,64
          CR       R14,R15
          STC      R14,5(R6)
          LA       R15,4(R15)
          SLL      R15,1
          STH      R15,DEL
NOFIN     NI       PUP+1,X'C7'

```

B WAITZ

```

*
*
****PEN UP SIGNAL****
*
*
IND2      NI      PUP+1,X'FE'      HERE CN PU TRAP
        LH      R7,DN      TEST FOR INK
        LTR     R7,R7
        BC      8,NOSANG      NO INK
        CLI     N+1,X'CO'      TEST FOR PERIOD
        BC      6,PTEST      NCT PERIOD
*TEST FOR SINGLE ANGLE
        LA      R7,4
        CH      R7,PANG
        BC      8,NOSANG
        LA      R7,1
        STH     R7,N
        LH      R10,PANG
        LH      R11,PANG
        LA      R8,3
SHFT2     SLL     R10,2
        AR      R10,R11
        BCT     R8,SHFT2
        SLL     R10,8
        ST      R10,CODE
        BC      15,PTEST
*UPDATE STROKE SIZE TO PREPARE FOR CENTER, ETC.
NCSANG    LH      R6,XS
        STH     R6,XLS
        STH     R6,XRS
        LH      R6,YS
        STH     R6,YTS
        STH     R6,YBS
*STROKE IS A DOT
*IS THIS THE 2ND STROKE OF A SCRIPT I OR J
        CLI     SN+1,X'02'
        BC      6,PTEST
        RCS     DOTA,I11,ECASE1,EPTST
*MAKE POSITION DECISION HERE, 1ST CHECK FOR SINGLE STROKE
PTEST     LA      R7,1
        CH      R7,SN
        BC      10,CASE1      ONLY 1 STROKE
*CAN OLD CHAR BE COMBINED WITH ANY STROKE, I.E. IS THERE A P AND/OR PAD
        L       R7,PAD
        LTR     R7,R7
        BC      6,NOTDK
        CLI     P,X'CO'
        BC      8,TOBIG
*CAN OLD CHAR BE COMBINED WITH STROKE
NOTDK     RCS     ANG4A,I11,E*+4

```

```

RCS    CHECKA,III,ETOBIG,ECOMBOX
*OLD CHAR CAN BE COMBINED WITH THIS STROKE, TEST FOR COMMA
COMBOX CLI    N+1,X'01'
      BC      6,COMOK1      NOT 1 ANG
      TM      CODE,X'CO'
      BC      8,COMOK1 RIGHT HORIZ
      BC      4,COMOK2      UP OR LEFT HORIZ.
      LH      R15,DYM
      SRL     R15,2
      LH      R7,YTS
      SH      R7,YBS
      CR      R7,R15
      BC      10,COMCK1     NOT SHORT
*SHORT VERTICAL, IS IT AT THE BOTTOM
      LH      R7,YBS
      CH      R7,YBC
      BC      2,COMOK1      NO
*YES, DOES IT SLANT TO THE LEFT, I.E. IS ENDP T TO LEFT OF STARTPT
      LH      R8,SN
      BCT     R8,SLFT
SLFT   SLL     R8,1
      LH      R7,XSP(R8)
      CH      R7,XT
      BC      4,COMOK1
*SPECIAL TEST FOR T, IS THE FIRST STROKE A MINUS?
      CLI     P,X'02'
      BC      8,COMOK1
      B       TOBIG
*TEST FOR HORIZ. COMMA
COMOK2 TM      CODE,X'80'
      BC      8,COMOK1 UP
*LEFT HORIZ., IS IT AT THE BOTTOM
      LH      R7,YBS
      CH      R7,YBC
      BC      2,COMOK1      NO
      BC      12,TOBIG      YES,COMMA
*STROKE NOT A COMMA
COMOK1 LH      R7,XRC
      CH      R7,XLS
      BC      4,CASE1C
      CH      R7,XRS
      BC      4,CASE1B
      LH      R7,XLC
      CH      R7,XRS
      BC      4,CASE1
CASE1C CLI     SN+1,X'02'      NO, IS OLD CHAR VERTICAL
      BC      6,TSTS2
      CLI     P,X'01'
      BC      8,CASE1A
TSTS2  CLI     N+1,X'01'      NO, IS NEW STROKE VERTICAL
      BC      6,TOBIG

```

```

      TM      CODE,X'CO'
      BC      12,TOBIG
      CLI     CHAR,X'CE'      YES, IS OLD CHR A PLUS
      BC      8,CASE1E
      CLI     CHAR,X'D2'      NO, IS IT A K
      BC      8,CASE1E
      BC      6,TOBIG
*1ST STROKE IS A VERTICAL, IS IT A 1
CASE1A  CLI   CHAR,X'F1'
      BC      8,CASE1B
      CLI     CHAR,X'E1'      NO, IS IT A SLASH
      BC      8,CASE1B
      CLI     CHAR,X'CD'      NO, IS IT A R. PAREN.
      BC      8,CASE1B      YES
      CLI     CHAR,X'CD'      NO, IS IT A L. PAREN.
      BC      6,TOBIG
*TEST FOR SHORT VERT SECOND STROKE
CASE1B  CLI   N+1,X'01'
      BC      6,CASE1D      NOT SINGLE ANGLE
      TM      CODE,X'CO'
      BC      12,CASE1D      NOT VERT
CASE1E  LH     R15,DYM      2ND VERT, IS IT SHORT
      SRL     R15,2
      LH      R7,YTS
      SH      R7,YBS
      CR      R7,R15
      BC      4,TOBIG
*NO, IS DIFF BETWEEN CENTERS GTR R RASTERS
*GET HERE WHEN
*FIRST STROKE VERT, SECOND NOT
*AND FIRST STROKE RIGHT CLOSE TO SECOND LEFT
*OR BOTH STROKES VERTICAL
CASE1F  EQU    *
      LH      R15,WIDTH
      LR      R10,R15
      SRL     R10,1
      AR      R15,R10      R = 3/4 WIDTH
      LH      R7,XRC
      AH      R7,XLC      2 OLD CENTER
      LH      R8,XRS
      AH      R8,XLS      2 NEW CENT
      LR      R9,R8
      SR      R8,R7
      LPR     R8,R8
      CR      R8,R15
      BC      12,CASE1
*YES, IS DIFF GTR R1(R1 GTR R) RASTERS
      LH      R15,WIDTH
      SLL     R15,1      R1 = WIDTH
      CR      R8,R15
      BC      2,TOBIG

```

*IS NEW XCENT IN LEFTMOST 1/4 OF A GRID POS?

LH	R15,WIDTH	
SRL	R15,3	CHAR WIDTH IN RASTERS
SR	R8,R8	
SRL	R9,3	NEW XCENT IN RASTERS
DR	R8,R15	NEW XCENT MOD(WIDTH)
LR	R10,R15	
SRL	R10,2	1/4 WIDTH
CR	R8,R10	REMAINDER IN R8
BC	4,CASE1	

*IS OLD X CENT IN RIGHTMOST 1/4 OF A GRID POS

SR	R6,R6	
SRL	R7,3	OLD XCENT IN RASTERS
DR	R6,R15	OLD XCENT MOD(WIDTH)
SR	R15,R10	
CR	R6,R15	3/4 WIDTH
LA	R10,11	
CR	R6,R10	REMAINDER IN R6
BC	2,CASE1	
BC	12,TOBIG	

*2ND STROKE IS NOT VERTICAL

*IS DIFF BETWEEN 2ND STROKE LEFT AND 1ST STROKE RIGHT GTR R RASTERS

CASE1D	EQU	*	
LH	R15,WIDTH		
SRL	R15,2		
LR	R10,R15		
SRL	R10,1		
AR	R15,R10	R = 3/8 WIDTH	
LH	R7,XLS		
SR	R7,R15		
CH	R7,XRC		
BC	2,TOBIG		
BC	12,CASE1F		

*

*

*****MULTI-STROKES*****

*

*

CASE1 RCS MAXMNC,I11,E*+4

*

*

*****NEW CHARACTER PARAMETERS*****

*

*

REINK	LH	R7,INKIND	
	STH	R7,INKC	
MORCHR	LH	R8,SN	ENDPOINTS
	BCT	R8,REDR8	
REDR8	SLL	R8,1	
	LH	R6,XT	
	STH	R6,XEP(R8)	

```

      LH      R6,YT
      STH     R6,YEP(R8)
      RCS     DELT,I11,E**4
*QUANTIZE REL MAX AND MINS
      RCS     QMMA,I11,E**4
*SET UP I1 AS A TRANSLATION OF CODE
      RCS     ANG4A,I11,E**4
      CLI     I1+3,X'EF'
      BC      6,ANG56X
EF13    MVI     I1+3,X'13'
ANG56X  RCS     ANG56,I11,E**4
      LH      R7,DXC
      LTR     R7,R7
      BC      8,ASPR3
      SR      R8,R8
      LH      R9,DYC
      SLL     R9,2
      DR      R8,R7
      LR      R7,R9
      B       ASPR2
ASPR3   LA      R7,4095
      SLL     R7,4
ASPR2   STH     R7,ASPR
      LH      R8,NC
      NTC1    STH     R8,NCUSP
      NTX     EQU     *
*NO. OF TIME CORNERS
      LH      R8,NT
      BCT     R8,TNT1
TNT1    CH      R8,NTC
      BC      2,TNTX
      LH      R8,NTCUSP
      BCT     R8,TNTC1
TNTC1   STH     R8,NTCUSP
TNTX    EQU     *
      LH      R7,YTC
      AH      R7,YBC
      SRL     R7,1
      LH      R8,XRC
      AH      R8,XLC
      SRL     R8,1
      SLL     R8,16
      AR      R7,R8
      ST      R7,CENT
*TEST FOR SPECIAL CHARACTERS
      CLI     SN+1,X'02'
      BC      6,TSTSCB
      CLI     CHAR,X'89'
      BC      8,RECRIN
      CLI     CHAR,X'91'
      BC      6,TSTSCB

```

CORNERS

SCRIPT I

*SCRIPT J

```

RCS    RAZEA,III,ERECRTN
TSTSCB EQU    *
      CLI    CHAR,X'72'          TEST FOR SCRUB (N GTR 15)
      BC     8,RECRTN
      CLI    N+1,X'08'
      BC     12,CALREC

```

*N GTR 8, CHARACTER IS A POTENTIAL SCRUB

*IF N GTR 12, , OR CHARACTER IS LARGE, SET CHAR=SCRUB

*CTHERWISE ALLOW FOR A POSSIBLE SCRIPT CHARACTER

```

      CLI    N+1,X'0C'
      BC     2,SCBX
      LH     R8,DYM
      CH     R8,DYC
      BC     4,SCBX              DYC GTR DYM
      CH     R8,DXC
      BC     10,CALREC          DXC LSS DYM
SCBX   EQU    *
      MVI    CHAR,X'72'
      B      RECRTN

```

*

*

****REC CALL****

*

*

CALREC INST AREC,RECA,III,III,ERECRTN

*

*

****MULTI-CHARACTERS****

*

*

```

TOBIG  LH     R7,INKC
      LH     R8,INKIND
      CR     R8,R7
      BNH    OVR2
      SR     R8,R7
      STH    R8,INKIND
      L      R4,DATA              MOVE INK
      L      R9,ICP
      LH     R10,6(R9)
      SR     R10,R7
      LA     R11,2
      STH    R11,6(R9)
      L      R6,INKB

```

```

MOVINK STC    R8,MVC+1
      LA     R7,0(R7,R6)
      EX     0,MVC
      L      R4,DATA
      TM     IND,X'80'
      BE     OVR21
      STH    R10,6(R9)

```

DON'T UPDATE CCW COUNT IF NO INK

```

      B      OVR2
*
*
****ZERO INK COUNT****
*
*
OVR21      XC      INKIND(2),INKIND
*
*
****DON'T RESTORE****
*
*
OVR2        OI      PUP+1,X'30'
*
*
****2 CHARACTERS****
*
*
ALPHA      L        R4,DATA                      TEST FOR NO RECOGNITION
           TM        IND,X'40'
           BC        8,WATR1
           CLI       CHAR,X'EF'                  CK IF CHAR OR NO CHAR
           BC        8,NCEXT                      NO CHAR
           TM        PUP+1,X'20'
           BC        8,TCE                        NO MORE DATA TAKE TERMINAL
           PARL      CNTX=F,LOW=PHI,HIGH=CHARX
*
*
****RESET FOR NEW CHARACTER****
*
*
*RESET CHAR SIZE, LOCATION, ETC.
PHI         MVC      DXC(12),DXS
           LH        R6,SN
           BCT       R6,DECR6
DECR6      SLL       R6,1
           LH        R7,XSP(R6)
           STH       R7,XSP
           LH        R7,YSP(R6)
           STH       R7,YSP
           LA        R6,1
           STH       R6,SN
           XC        P(1),P
           XC        PAD(4),PAD
           XC        CHAR(1),CHAR
           L         R4,DATA
           L         R7,CET
           STH       R7,YCENT
           MVC       PCHAR(1),CHARA
           B         REINK
*

```

```

*
****REC EXIT****
*
*
*REC EXITS TO HERE
RCRTN EQU *
*IF N GTR 8 AND CHAR IS NOT A SCRIPT CHARACTER, SET CHAR=SCRUB
    CLI CHAR,X'A9'
    BC 2,SCBX2
    CLI CHAR,X'81'
    BC 10,SCRIPT
SCBX2 EQU *
    CLI N+1,X'08'
    BC 12,RCRTN1
    MVI CHAR,X'72'
    B RCRTN1
SCRIPT EQU *
*THIS IS A SCRIPT CHARACTER
*
*THE FOLLOWING CODE CONVERTS A LOWER CASE CHARACTER TO THE SAME
*UPPER CASE CHARACTER
    OI CHAR,X'40'
RCRTN1 EQU *
    NI PUP+1,X'CF'
    L R4,INDEX PRESET EXPIRATION = TIME
    MVI 3(R4),X'01'
*SKIP AROUND THE CLOCK IF CHAR IS A GEOMETRIC SYMBOL, I.E. INK VECTOR
* SIZE IS 8 RASTERS
    L R4,DATA
    TM IND,X'0C'
    BC 1,CLEXP
NOSKIP EQU *
    OI PUP+1,X'04' IND CLOCK RUNNING
RECX INST CLK1,CLK2,FWAITBX,ITIME,ECLEXP,ECLEXF
*
*
****CLOCK EXPIRED (DUE TO RUNNING LONGER THEN TIME)****
*
*
CLEXP EQU *
SETCK EQU *
    SET PSG=WAITBX,CNTX=F TURN OFF CLOCK
    PAWS
*
*
****CLOCK TURNED OFF (DUE TO PENDOWN)****
*
*
CLEXF EQU *
    L R4,DATA
    TM IND,X'10'

```

```

BC      1,GOFINX
NI      PUP+1,X'FB'
TM      PUP+1,X'02'
BC      1,FINSH
BAL     R15,OUTPTS
TM      PUP+1,X'10'
BC      1,ALPHA
L       R4,INDEX
CLI     3(R4),X'00'
BC      8,WAITZ
MVI     PUP+1,X'00'
B       ALPHA

```

HALT CLOCK EXIT

CHECK IF PENDING CHAR EXIT

WHY CLOCK EXPIRED
MORE DATA
RESET ALL INIICATIONS

```

*
*
****FINISH ENTRY 1****

```

```

FIN      TM      PUP+1,X'04'
          BC      8,FINSH
          OI      PUP+1,X'02'
          BC      15,SETCK

```

CALLER INDICATES TO FINISH

SET BOX FOR CLOCK

```

*
*
****SET UP OUTPUTS****

```

```

OUTPTS   L       R4,DATA
          MVC     EP(2),XSP
          MVC     EP+2(2),YSP
          MVC     EP+4(2),XEP
          MVC     EP+6(2),YEP
          L       R4,DATA
          MVC     CHARA(1),CHAR
          L       R7,CENT
          ST      R7,CET
          MVC     AR(1),NCUSP+1
          MVC     AR+1(1),ASPR+1
          MVC     SIZE(2),DXC
          MVC     SIZE+2(2),DYC
          BR      R15

```

OUTPUT CHAR

OUTPUT CENTER

AR, CHAR SIZE

```

*
*
*****NO CHARACTER*****

```

```

NCEXT    EQU     *
          TM      PUP+1,X'20'
          BC      8,TNE
          PARL    CNTX=F,LOW=PHI,HIGH=NCHARX

```

TERMINAL EXIT NO MORE DATA

****FINISH ENTRY 2****

*

*

FINSH BAL R15,OUTPTS

*

*

****FINISH ENTRY 3****

*

*

GOFINX EQU *
BAL R15,OUTINK
EPLOG FINX

*

*

****TERMINAL CHARACTER****

*

*

TCE EQU *
BAL R15,OUTINK
EPLOG XTC

*

*

****TERMINAL NO CHAR****

*

*

TNE EQU *
BAL R15,OUTINK
EPLOG XTN

*

*

****SET UP INK****

*

*

CUTINK EQU *
L R4,DATA
TM IND,X'CC'
BC 1,OUTSKP
L R7,ICP
LA R8,2
STH R8,6(R7)
OUTSKP EQU *
BR R15

*

*

****END OF CHAREC****

*

*

END

CHAREC RCS'S

ANG4

*FUNCTION

*

*TRANSLATES THE FIRST FOUR STYLUS DIRECTIONS (IN CODE) TO A 1-BYTE

*INDEX (IN I1+3) CORRESPONDING TO A SET OF POTENTIAL STROKES.

*F0=NOT ALLOWABLE, 13=DON'T KNOW

*

*

*

*CALL

* RCS ANG4A,I11,EEXIT

*WHERE I1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST

*

*

*

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R7,R8,R10

*

*

USING XR6,R6

REGS

EX0 EQU 0

D6 DSECT

XR6 DS OF

I1 DS 1F

DS 1F

CODE DS 1F

DS 5F

N DS 1F

ANG4 BOX

LA R10,8

SH R10,N

BC 12,ANGOUT

LH R7,CODE

SRGT SRL R7,2

BCT R10,SRGT

LA R8,3

NR R8,R7

LA R10,8

SLFT	SH	R10,N	
	SLL	R7,2	
	CR	R7,R8	
	BCT	R10,SLFT	
	STH	R7,CODE	
ANGOUT	MVC	11+3(1),CODE	
	TR	11+3(1),THET4	
ANGE	BEXIT	EXO	
THET4	DS	OH	
	DC	X'0C'	0000 SBARM
	DC	15C'0'	ILLEGAL
	DC	2X'13'	0100-0101 DK
	DC	X'40'	0102 RSC
	DC	X'3F'	0103 SCRPT
	DC	C'0'	0110 ILLEGAL
	DC	X'13'	0111 DK
	DC	2C'0'	ILLEGAL
	DC	X'13'	0120 DK
	DC	X'21'	0121 S5
	DC	X'13'	0122 DK
	DC	X'3E'	0123 S09M
	DC	X'3D'	0130 S9LC1
	DC	X'41'	0131 SCPFP
	DC	X'43'	0132 RSS
	DC	X'42'	0133 SCPEL
	DC	2X'0D'	0200,0201 S2MRZ
	DC	X'2A'	0202 S3SCR8
	DC	2X'0F'	0203,0210 S3MBR
	DC	3X'0D'	0211-0213 S2MRZ
	DC	2C'0'	ILLEGAL
	DC	X'2D'	0222 SLKRTM
	DC	C'0'	
	DC	X'02'	0230 S23MB
	DC	X'0D'	0231 S2MRZ
	DC	X'0E'	0232 S3MB
	DC	X'14'	0233 S7MGK
	DC	X'0D'	0300 S2MRZ
	DC	X'32'	0301 S8
	DC	X'0E'	0302 S3MB
	DC	X'0F'	0303 S3MBR
	DC	X'44'	0310 RSV
	DC	X'49'	0311 S8LCV
	DC	X'40'	0312 RSC
	DC	X'01'	0313 STPM
	DC	X'02'	0320 S23MB
	DC	X'3B'	0321 S023MB
	DC	X'12'	0322 SRPRM
	DC	X'0D'	0323 S2MRZ
	DC	3C'0'	
	DC	X'14'	0333 S7MGK
	DC	X'1E'	1000 SFE

DC	3C'0'	
DC	X'13'	1010 DK
DC	X'13'	1011 DK, POSSIBLY TILDA
DC	X'13'	1012 DK
DC	X'13'	1013 DK
DC	X'02'	1020 S23MB
DC	X'CE'	1021 S3MB
DC	X'13'	1022 DK
DC	X'02'	1023 S23MB
DC	X'0F'	1030 S3MBR
DC	X'45'	1031 SCPNRZ
DC	X'03'	1032 S23MBP
DC	X'38'	1033 SA7
DC	5C'0'	
DC	X'16'	1111 S1MAK
DC	10C'0'	
DC	2X'13'	1200,1201 DK
DC	2X'18'	1202,1203 SSM
DC	4X'13'	1210-1213 DK
DC	2C'0'	
DC	X'13'	1222 DK
DC	C'0'	
DC	X'04'	1230 SMC
DC	X'32'	1231 S8
DC	X'05'	1232 SS8M
DC	X'15'	1233 STPA
DC	X'11'	1300 S24
DC	X'17'	1301 SNMA
DC	2X'35'	1302,1303 S3
DC	X'0D'	1310 S2MRZ
DC	X'17'	1311 SNMA
DC	X'36'	1312 SASTAR
DC	X'18'	1313 SMLC
DC	X'36'	1320 SASTAR
DC	X'0D'	1321 S2MRZ
DC	X'13'	1322 DK
DC	X'46'	1323 RSZ
DC	3C'0'	
DC	X'19'	1333 SCOMAM
DC	X'1A'	2000 SBARMK
DC	3C'0'	
DC	X'33'	2010 SG
DC	X'06'	2011 SG069M
DC	X'08'	2012 SG06M
DC	X'34'	2013 S9
DC	X'29'	2020 SGSCR8
DC	X'18'	2021 SSM
DC	X'30'	2022 SGS06M
DC	X'47'	2023 SE
DC	X'4A'	2030 SEQ
DC	X'48'	2031 SCPGQ

DC	X'05'	2032 SS8M
DC	X'1C'	2033 S9MK
DC	3X'13'	2100-2102 DK
DC	X'3C'	2103 S09
DC	C'0'	
DC	X'13'	2111 DK
DC	2C'0'	
DC	8X'13'	2120-2133 DK
DC	10C'0'	
DC	X'0C'	2222 SBARM
DC	5C'0'	
DC	X'1D'	2300 SCMEG
DC	X'06'	2301 SG069M
DC	X'2F'	2302 SGS
DC	X'07'	2303 SS589M
DC	X'22'	2310 STP5
DC	X'32'	2311 S8
DC	X'33'	2312 SG
DC	X'07'	2313 SS589M
DC	2X'32'	2320,2321 S8
DC	X'1B'	2322 SSM
DC	X'47'	2323 SE
DC	3C'0'	
DC	X'1E'	2333 SFE
DC	X'1F'	3000 SLMEK4
DC	3C'0'	
DC	X'3A'	3010 SG81
DC	X'20'	3011 SUMJU
DC	X'08'	3012 SGC6M
DC	X'27'	3013 SUMAM
DC	2X'22'	3020,3021 STP5
DC	X'10'	3022 STP6
DC	3X'22'	3023-3031 STP5
DC	X'22'	3032 STP5
DC	X'23'	3033 SK5
DC	X'24'	3100 STPH
DC	X'13'	3101 DK
DC	X'2C'	3102 SBDPR1
DC	X'09'	3103 SBDPR
DC	C'0'	
DC	X'25'	3111 SVM
DC	2C'0'	
DC	2X'24'	3120,3121 STPH
DC	X'26'	3122 SDMH
DC	X'39'	3123 SCG
DC	X'28'	3130 BR
DC	X'0A'	3131 SMNW
DC	X'2C'	3132 SBDPR1
DC	X'27'	3133 SUMAM
DC	4X'13'	3200-3203 DK
DC	X'3C'	3210 S09

```

DC      X'28'          3211 STPJ
DC      2X'34'         3212,3213 S9
DC      2C'0'
DC      X'2E'          3222 SRPRMJ
DC      C'0'
DC      X'37'          3230 SCC
DC      X'32'          3231 S8
DC      X'35'          3232 S3
DC      X'13'          3233 DK
DC      15C'0'
DC      X'0B'          3333 SM1M
END

```

CHECK

*FUNCTION

*
 *CHECKS TO SEE IF THE PREVIOUS SUBCHARACTER (PREV. 'REC' OUTPUT) CAN BE
 *COMBINED WITH THE CURRENT STROKE (AS ENCODED FROM THE FIRST FOUR
 *DIRECTIONS BY 'ANG4') TO FORM ONE OF THE ALLOWABLE CHARACTERS.

*
 *
 *
 *CALL
 * RCS CHECKA,I11,END,EYES
 *WHERE I1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
 * I1+3 CONTAINS THE STROKE CODE
 * EXIT NO WHEN STROKE AND SUBCHARACTER CANNOT BE COMBINED
 * EXIT YES WHEN STROKE AND SUBCHARACTER CAN BE COMBINED

*
 *
 *
 *INPUT REGISTER. R6

*
 *INTERNAL REGISTERS. R7 THRU R10

```

*
*      USING XR6,R6
EX0      EQU      0
EX4      EQU      4
          REGS
D6        DSECT
XR6       DS       0F
I1        DS       1F
          DS       2F
          DS       11H

```

SN	DS	1H	
	DS	14H	
	DS	20C	
	DS	3F	
	DS	1H	
	DS	2C	
	DS	3H	
P	DS	1C	
CHAR	DS	1C	
CHECK	BOX		
	SR	R8,R8	
	IC	R8,I1+3	
	BCT	R8,MULT	
MULT	SLL	R8,2	4 TIMES (I1-1)
	EX	0,CHKTAB(R8)	
CK	SR	R8,R8	
	IC	R8,CHAR	
*ALL VE	RTICALS	TREATED THE SAME	
	CLI	SN+1,X'02'	
	BC	6,CK2	
	CLI	P,X'01'	
	BC	8,CK1	
	CLI	P,X'09'	
	BC	6,CK2	
*OLD CH	AR IS	VERT	
*CHANGE	CHAR	CODE TO 1	
CK1	LA	R8,1	
CK2	SR	R9,R9	
	SR	R10,R10	
CK3	IC	R10,0(R7)	
	CR	R10,R9	
	BC	8,CKX	
	CR	R10,R8	
	BC	8,CKOK	
	LA	R7,1(R7)	
	BC	15,CK3	
*END OF	POSSIBLE	OLD-CHAR LIST	
CKOK	BEXIT	EX4	
CKX	BEXIT	EX0	
CHKTAB	DS	OF	
	LA	R7,S1	B1
	LA	R7,S2	B2
	LA	R7,S2	B3
	LA	R7,S10	B4
	LA	R7,S4	B5
	LA	R7,S10	B6
	LA	R7,S4	B7
	LA	R7,S10	B8
	LA	R7,S3	B9
	LA	R7,S4	B10
	LA	R7,S12	B11

LA	R7,S13	B12	
LA	R7,S3	B13	
LA	R7,S1	B14	
LA	R7,S1	B15	
LA	R7,S2	B16	
LA	R7,S7		B17(11)
LA	R7,S3	B18	
LA	R7,S4	B19	
LA	R7,S5	B20	
LA	R7,S4	B21	
LA	R7,S6	B22	
LA	R7,S7	B23	
LA	R7,S4	B24	
LA	R7,S8	B25	
LA	R7,S3	B26	
LA	R7,S4	B27	
LA	R7,S1	B28	
LA	R7,S1	B29	
LA	R7,S14	B30	
LA	R7,S9	B31	
LA	R7,S10	B32	
LA	R7,S4	B33	
LA	R7,S4	B34	
LA	R7,S1	B35	
LA	R7,S1	B36	
LA	R7,S11	B37	
LA	R7,S1	B38	
LA	R7,S1	B39	
LA	R7,S4	B40	
LA	R7,S4	B41	
LA	R7,S1	B42	
LA	R7,S4	B43	
LA	R7,S7	B44	
LA	R7,S1	B45	
LA	R7,S15		B46
LA	R7,S10	B47	
LA	R7,S10	B48	
LA	R7,S10	B49	
LA	R7,S4	B50	
LA	R7,S4	B51	
LA	R7,S4	B52	
LA	R7,S4	B53	
LA	R7,S4	B54	
LA	R7,S4	B55	
LA	R7,S4	B56	
LA	R7,S4	B57	
LA	R7,S4		B58(3A)
LA	R7,S2		B59(3B)
LA	R7,S2		B60(3C)
LA	R7,S4		B61(3D)
LA	R7,S10		B62(3E)

	LA	R7,S4		B63(3F)
	LA	R7,S4		B64(40)
	LA	R7,S4		B65(41)
	LA	R7,S4		B66(42)
	LA	R7,S4		B67(43)
	LA	R7,S4		B68(44)
	LA	R7,S4		B69(45)
	LA	R7,S4		B70(46)
	LA	R7,S4		B71(47)
	LA	R7,S4		B72(48)
	LA	R7,S4		B73(49)
	LA	R7,S4		B74(4A)
	EJECT			
S1	DS	OH		
	DC	X'01'	1	
	DC	X'00'		
S2	DS	OH		
	DC	X'01'	1	
	DC	X'E0'	-	
	DC	X'F0'	0	
	DC	X'E4'	U	
	DC	X'C3'	C	
	DC	X'F6'	6	
	DC	X'C7'	G	
	DC	X'E2'	S	
	DC	X'F5'	5	
	DC	X'F8'	8	
	DC	X'00'		
S3	DS	OH		
	DC	X'01'	1	
	DC	X'E0'	-	
	DC	X'00'		
S4	DS	OH		
	DC	X'00'		
S5	DS	OH		
	DC	X'01'	1	
	DC	X'E0'	-	
	DC	X'F0'	0	
	CC	X'F6'	6	
	DC	X'C3'	C	
	DC	X'C7'	G	
	DC	X'D2'	K	
	DC	X'E3'	T	
	DC	X'E5'	V	
	DC	X'E7'	X	
	DC	X'E8'	Y	
	DC	X'CE'	+	
	DC	X'CD'	(	
	CC	X'E1'	/	
	CC	X'00'		
S6	CS	OH		

	DC	X'01'	1
	DC	X'E0'	-
	DC	X'E7'	X
	DC	X'E8'	Y
	DC	X'E5'	V
	DC	X'D2'	K
	DC	X'F7'	7
	DC	X'DD'	)
	DC	X'00'	
S7	DS	OH	
	DC	X'E0'	-
	DC	X'00'	
S8	DS	OH	
	DC	X'01'	1
	DC	X'E0'	-
	DC	X'CE'	+
	DC	X'00'	
S9	DS	OH	
	DC	X'01'	1
	DC	X'E0'	-
	DC	X'D2'	K
	DC	X'FE'	=
	DC	X'F0'	0
	DC	X'F6'	6
	DC	X'C7'	G
	DC	X'E7'	X
	DC	X'E8'	Y
	DC	X'E5'	V
	DC	X'00'	
S10	DS	OH	
	DC	X'F0'	0
	DC	X'E4'	U
	DC	X'C3'	C
	DC	X'F6'	6
	DC	X'C7'	G
	DC	X'00'	
S11	DS	OH	
	DC	X'01'	1
	DC	X'E5'	V
	DC	X'D2'	K
	DC	X'E7'	X
	DC	X'E8'	Y
	DC	X'F0'	0
	DC	X'E4'	U
	DC	X'C3'	C
	DC	X'F6'	6
	DC	X'C7'	G
	DC	X'00'	
*VERTICAL STROKE			
S12	DS	OH	
	DC	X'01'	1

DC	X'E0'	-
DC	X'D2'	K
DC	X'E5'	V
DC	X'E7'	X
DC	X'E8'	Y
DC	X'CE'	+
DC	X'C9'	I
DC	X'C6'	F
DC	X'F0'	0
DC	X'F6'	6
DC	X'E4'	U
DC	X'D3'	L
DC	X'F7'	7
DC	X'CD'	(
DC	X'E1'	/
DC	X'CC'	LSS
DC	X'FE'	=
DC	X'D5'	N
DC	X'DC'	*
DC	X'C1'	A
DC	X'E2'	S
DC	X'C3'	C
DC	X'D7'	P
DC	X'F8'	8
DC	X'C7'	G
DC	X'DB'	\$
DC	X'F5'	
DC	X'F2'	
DC	X'85'	
DC	X'89'	
DC	X'99'	
DC	X'A5'	
DC	X'86'	
DC	X'00'	

5
2
LC E
LC I
LC R
LC V
LC F

*HORIZONTAL STROKE

S13

DS	OH	
DC	X'01'	1
DC	X'E0'	-
DC	X'D2'	K
DC	X'E5'	V
DC	X'E7'	X
DC	X'E8'	Y
DC	X'E3'	T
DC	X'CE'	+
DC	X'C9'	I
DC	X'C6'	F
DC	X'F5'	5
DC	X'E2'	S
DC	X'F2'	2
DC	X'C3'	C
DC	X'CF'	LBRAC

	DC	X'70'	KARAT	
	DC	X'F7'	7	
	DC	X'D3'	L	
	DC	X'CD'	(	
	DC	X'E1'	/	
	DC	X'D1'	J	
	DC	X'DD'	)	
	DC	X'EE'	GTR	
	DC	X'F0'	0	
	DC	X'F6'	6	
	DC	X'E4'	U	
	DC	X'C2'	B	
	DC	X'C4'	D	
	DC	X'D7'	P	
	DC	X'D9'	R	
	DC	X'F8'		8
	DC	X'F9'		9
	DC	X'C1'		A
	DC	X'85'		LC E
	DC	X'86'		LC F
	DC	X'89'		LC I
	DC	X'93'		LC L
	DC	X'88'		LC H
	DC	X'C8'		H
	DC	X'DC'		*
	DC	X'00'		
S14	DS	OH		
	DC	X'01'	1	
	DC	X'E0'	-	
	DC	X'F7'	7	
	DC	X'00'		
S15	DS	OH		
	DC	X'E0'		-
	DC	X'F0'		0
	DC	X'00'		
	END			

CORNER

*FUNCTION

*

*DETECTS CORNERS BASED ON SHARP CHANGES IN DIRECTION, AND UPDATES NC.

*OF GEOMETRIC CORNERS (NC) AND THE ARRAY OF POSITIONS OF GEOMETRIC

*CORNERS (XC,YC).

*USES 16-DIRECTION SEQUENCE AX THRU AX3 AND DIFFERENCES.

*INDEX C=0 IS WAIT FOR CORNER, C=1 IS POTENTIAL CORNER, C=2 IS JUST GOT

*CORNER.

*
*
*

*CALL

* RCS CORNERA,I11,EEXIT

*WHERE I1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST

*
*
*

*INPUT REGISTER. R6

*
*INTERNAL REGISTERS. R7 THRU R9

*
*
*

	USING XR6,R6
EX0	EQU 0
	REGS
D6	DSECT
XR6	DS 0F
	DS 3F
	DS 26H
	DS 20C
	DS 3F
	DS 1H
	DS 2C
	DS 3H
	DS 56C
	DS 2H
XLO	DS 1H
YLO	DS 1H
	DS 1H
AX2	DS 1H
	DS 1H
AX	DS 1H
AX23	DS 1H
AX12	DS 1H
AX01	DS 1H
AX02	DS 1H
NC	DS 1H
C	DS 1H
	DS 7H
	DS 1F
	DS 6C
	DS 1H
	DS 1F
XC	DS 10C
YC	DS 10C
CORNER	BOX
*CORNER	DETECTOR

```

      CLI    C,X'02'
      BC     8,COUT
      CLI    C,X'01'
      BC     8,CEQ1
      CLI    AX01+1,X'04'
      BC     4,AXP2
      CLI    AX01+1,X'0C'
      BC     2,AXP2
AX1EQ2  LH     R7,AX12
      STH    R7,AX02
      BC     15,IEQJ
AXP2    LH     R7,AX
      SH     R7,AX2
      LPR    R7,R7
      STH    R7,AX02
      CLI    AX02+1,X'04'
      BC     4,COUT
      CLI    AX02+1,X'0C'
      BC     2,COUT
      LH     R7,AX23
      STH    R7,AX02
IEQJ    CLI    AX02+1,X'01'
      BC     12,SETC1
      CLI    AX02+1,X'0F'
      BC     6,COUT
SETC1   CLI    C,X'01'
      BC     8,INCNC
      MVI    C,X'01'
**STORE  POSITION OF POTENTIAL CUSP
      LH     R9,NC
      LA     R8,5
      CR     R9,R8
      BC     4,NCLSS5
      SR     R9,R9
NCLSS5  SLL    R9,1
      LH     R8,XLO
      STH    R8,XC(R9)
      LH     R8,YLO
      STH    R8,YC(R9)
      BC     15,CEXIT
CEQ1    LH     R7,AX01
      STH    R7,AX02
      BC     15,IEQJ
INCNC   LH     R7,NC
      LA     R7,1(C,R7)
      STH    R7,NC
      MVI    C,X'02'
      BC     15,CEXIT
COUT    MVI    C,X'00'
CEXIT   BEXIT  EX0
      END

```

DOT

*FUNCTION

*

*USED WHEN THE SECOND STROKE IS A DOT.

*DETERMINES IF THE FIRST STROKE RESULTS IN A SCRIPT I OR J.

*

*

*

*CALL

* RCS DOTA, I11, EYES, END

*WHERE I1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST

* EXIT YES WHEN SCRIPT I OR J

* EXIT NO WHEN NOT SCRIPT I OR J

*

*

*

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R7 THRU R10

*

*

*

USING XR6, R6

REGS

EX0 EQU 0

EX4 EQU 0

D6 DSECT

XR6 DS 0F

DS 3F

DS 26H

DS 20C

DS 3F

DS 1H

DS 2C

DS 3H

DS 1C

CHAR DS 1C

DOT BOX

*2ND STROKE IS A DOT

*DOES 1ST STROKE RESULT IN A SCRIPT I OR J

*IF YES, TAKE EX0, OTHERWISE EX4

SR R8, R8

IC R8, CHAR

SR R9, R9

	SR	R10,R10	
	LA	R7,ILIST	
CKLIST	EQU	*	
	IC	R10,0(R7)	
	CR	R10,R9	
	BC	8,NOX	NOT I OR J
	CR	R10,R8	
	BC	8,IJX	
	LA	R7,1(R7)	
	B	CKLIST	
IJX	LA	R8,JLIST	
	CR	R7,R8	
	BC	10,JX	
IX	MVI	CHAR,X'89'	
	B	YESX	
JX	MVI	CHAR,X'91'	
YESX	BEXIT	EX0	
NOX	BEXIT	EX4	
ILIST	DS	OF	
	DC	X'89'	I
	DC	X'85'	E
	DC	X'A5'	V
	DC	C'L'	L
	DC	C'2'	2
	DC	X'82'	B
	DC	X'70'	KARAT
JLIST	EQU	*	
	DC	X'86'	F
	DC	X'91'	J
	DC	X'F8'	8
	DC	X'E5'	V
	DC	X'DD'	RIGHT PAREN
	DC	X'00'	END OF LISTS
	END		

DELTAS

*FUNCTION

*

*QUANTIZES THE STARTING POINT AND ENDING POINT LOCATIONS OF EACH STROKE
 *BY CONSIDERING THE CHARACTER REGION AS A 4 X 4 GRID CODED AS

*

			YTC	
		3	2	1
		7	6	5
	XLC	11	10	9
		15	14	13
				12
			YBC	
				XRC

```

*
*
*
*CALL
*      RCS      DELTAS,I11,EEXIT
*WHERE I1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
*
*
*
*INPUT REGISTER.  R6
*
*INTERNAL REGISTERS.  R7 THRU R14
*
*
      USING XR6,R6
      REGS
EX0      EQU      0
D6       DSECT
XR6      DS       0F
          DS       3F
          DS       11H
SN        DS       1H
          DS       6H
XRC       CS       1H
XLC       DS       1H
YTC       DS       1H
YBC       DS       1H
          DS       4H
XYE       DS       10C
XYS       DS       10C
          DS       3F
          DS       1H
          DS       2C
          DS       3H
          DS       8C
XSP       DS       10C
YSP       DS       10C
XEP       DS       10C
YEP       DS       10C
DELTAS    BOX
          SR       R7,R7
          LA       R8,2
          LH       R9,SN
          SLL      R9,1
          BCT      R9,DEL1
DEL1      LH       R10,XRC
          SH       R10,XLC
          SRL      R10,2
          LH       R11,YTC
          SH       R11,YBC

```

```

DEL10  SRL  R11,2
        LA   R13,3
        LR   R14,R13
        LH   R12,XLC
DEL3    AR   R12,R10
        CH   R12,XSP(R7)
        BC   2,DEL2
        BCT  R13,DEL3
DEL2    LH   R12,YBC
DEL5    AR   R12,R11
        CH   R12,YSP(R7)
        BC   2,DEL4
        BCT  R14,DEL5
DEL4    SLL  R14,2
        OR   R13,R14
        STH  R13,XYS(R7)
        LA   R13,3
        LR   R14,R13
        LH   R12,XLC
DEL6    AR   R12,R10
        CH   R12,XEP(R7)
        BC   2,DEL7
        BCT  R13,DEL6
DEL7    LH   R12,YBC
DEL8    AR   R12,R11
        CH   R12,YEP(R7)
        BC   2,DEL9
        BCT  R14,DEL8
DEL9    SLL  R14,2
        OR   R13,R14
        STH  R13,XYE(R7)
        BXLE R7,R8,DEL10
        BEXIT EXO
        END

```

FN56

*FUNCTION

*

*PRODUCES INDEX IN BR56 BASED ON NO. OF DIRECTIONS (N) AND DIRECTIONS
*5 AND 6.

*N=4 GIVES BR56 = 16, OTHERWISE BR56 GETS BITS 8 THRU 11 OF CODE.

*

*

*

*CALL

* RCS FN56A,I11,EEXIT
 *WHERE I1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST

*
 *
 *
 *INPUT REGISTER. R6
 *
 *INTERNAL REGISTERS. R7
 *
 *

	USING XR6,R6
	REGS
EX0	EQU 0
D6	DSECT
XR6	DS 0F
	DS 2F
CCODE	DS 1F
	DS 10H
N	DS 1H
	DS 4H
BR56	DS 1H
FN56	BOX
	LA R7,16
	STH R7,BR56
	CLI N+1,X'05'
	BC 4,FN56E
	SR R7,R7
	IC R7,CCODE+1
	SRL R7,4
	STH R7,BR56
FN56E	BEXIT EX0
	END

HYST

*FUNCTION
 *
 *TRANSFORMS STYLUS INCREMENTAL DISTANCE TO PROVIDE HYSTERESIS ZONES
 *WHEN COMPUTING STYLUS DIRECTION.
 *
 *
 *
 *CALL
 * RCS HYSTA,EEXIT
 *
 *

```
*
*INPUT REGISTERS
*
*C(R6) = SMALLER (EITHER X OR Y) INCREMENT
*C(R7) = LARGER (EITHER Y OR X) INCREMENT
*
*
*
*OUTPUT REGISTERS
*
*C(R8) = 3/4 LARGER INCREMENT - SMALLER INCREMENT
*
*
*INTERNAL REGISTERS. NONE OTHER THAN THE ABOVE
*
*
```

```
                USING XR6,R6
                REGS
EX0             EQU    0
D6             DSECT
XR6            DS      0F
HYST           BOX
                LR      R8,R6
                SRA     R6,2
                SR      R8,R6
                SR      R8,R7
                BEXIT EX0
                END
```

MXMNC

```
*FUNCTION
*
*UPDATES THE X BOUNDS (XLC,XRC) AND Y BOUNDS (YTC,YBC) OF THE CHARACTER
*
*
*CALL
*      RCS    MXMNCA,I11,EEXIT
*WHERE I1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
*
*
*
*INPUT REGISTER. R6
*
```

*INTERNAL REGISTERS. R7

*
*

	USING XR6,R6	
	REGS	
EX0	EQU	0
D6	DSECT	
XR6	DS	0F
	DS	3F
	DS	16H
DXC	DS	1H
DYC	DS	1H
XRC	DS	1H
XLC	DS	1H
YTC	DS	1H
YBC	DS	1H
	DS	4H
	DS	20C
	DS	3F
	DS	1H
	DS	2C
	DS	3H
	DS	56C
	DS	17H
XRS	DS	1H
XLS	DS	1H
YTS	DS	1H
YBS	DS	1H
MXMNC	BOX	
	LH	R7,XRS
	CH	R7,XRC
	BC	12,MAX1
	STH	R7,XRC
MAX1	LH	R7,XLS
	CH	R7,XLC
	BC	10,MAX2
	STH	R7,XLC
MAX2	LH	R7,YTS
	CH	R7,YTC
	BC	12,MAX3
	STH	R7,YTC
MAX3	LH	R7,YBS
	CH	R7,YBC
	BC	10,MAX4
	STH	R7,YBC
MAX4	LH	R7,YTC
	SH	R7,YBC
	STH	R7,DYC
	LH	R7,XRC
	SH	R7,XLC
	STH	R7,DXC

BEXIT EXO
END

MXMNS

*FUNCTION

*

*UPDATES THE X BOUNDS (XLS,XRS) AND Y BOUNDS (YTS,YBS) OF THE CURRENT
*STROKE

*

*

*

*CALL

* RCS MXMNSA,I11,EEXIT

*WHERE I1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST

*

*

*

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R7

*

*

USING XR6,R6

REGS

EX0 EQU 0

D6 DSECT

XR6 DS 0F

DS 3F

DS 2H

XT DS 1H

YT DS 1H

DS 22H

DS 20C

DS 3F

DS 1H

DS 2C

DS 3H

DS 56C

DS 15H

CXS DS 1H

DYS DS 1H

XRS DS 1H

XLS DS 1H

YTS DS 1H

YBS DS 1H

```
MXMNS      BOX
           LH      R7,XT
           CH      R7,XRS
           BC      12,MAX11
           STH     R7,XRS
MAX11      CH      R7,XLS
           BC      10,MAX22
           STH     R7,XLS
MAX22      LH      R7,YT
           CH      R7,YTS
           BC      12,MAX33
           STH     R7,YTS
MAX33      CH      R7,YBS
           BC      10,MAX44
           STH     R7,YBS
MAX44      LH      R7,YTS
           SH      R7,YBS
           STH     R7,DYS
           LH      R7,XRS
           SH      R7,XLS
           STH     R7,DXS
           BEXIT   EXO
           END
```

QMM

```
*FUNCTION
*
*QUANTIZES YMAX (THE Y COORDINATE OF A RELATIVE MAXIMUM) ARRAY TO QYMAX
*ARRAY, AND QUANTIZES YMIN TO QYMIN.  THE QUANTIZATION INTERVAL IS 1/4
*CHARACTER HEIGHT WITH QYMAX = 0 IN THE TOP 1/4 OF THE CHARACTER, ETC.
*
*
*CALL
*      RCS      QMMA,I11,EEXIT
*WHERE I1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
*
*
*INPUT REGISTER.  R6
*
*INTERNAL REGISTERS.  R7 THRU R14
*
*
      USING XR6,R6
```

```

REGS
EX0 EQU 0
D6 DSECT
XR6 DS 0F
DS 3F
DS 20H
YTC DS 1H
YBC DS 1H
DS 4H
DS 20C
DS 3F
DS 1H
DS 2C
DS 3H
DS 56C
DS 21H
DS 1F
DS 6C
DS 1H
DS 1F
DS 20C
DS 23H
YMAX DS 10H
QYMAX DS 10C
QMM BOX

```

*NOTE THAT YMIN=YMAX+10, QYMIN=QYMAX+5

```

SR R7,R7
LA R8,2
LA R9,20
LH R13,YTC
SH R13,YBC
SRL R13,2
LH R10,YBC
AR R10,R13
LR R11,R10
AR R11,R13
LR R12,R11
AR R12,R13
ALF EQU *
LR R14,R7
SRL R14,1
LA R14,QYMAX(R14)
CH R11,YMAX(R7)
BC 4,Q01
CH R10,YMAX(R7)
BC 4,Q2
Q3 MVI 0(R14),X'03'
B BXLE
Q2 MVI 0(R14),X'02'
B BXLE
Q01 CH R12,YMAX(R7)

```

2(5 MIN + 5 MAX)

D= 1/4 CHAR HEIGHT

YBC + D

YBC + 2D

YBC + 3D

```
BC      4,Q0
Q1      MVI  0(R14),X'01'
        B    BXLE
Q0      MVI  0(R14),X'00'
BXLE    BXLE  R7,R8,ALF
        BEXIT EX0
        END
```

RAZE

*FUNCTION

*
*INCREASES THE Y COORDINATE OF THE CHARACTER CENTER BY (NORMAL CHARACTER HEIGHT/2) RASTERS SO THAT A CHARACTER WHICH STRADDLES A LINE
*WILL BE DISPLAYED IN THE PROPER POSITION.
*

*
*

*CALL

* RCS RAZE,II1,EEXIT
*WHERE II IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST

*
*

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R8,R15

*
*

```
        USING XR6,R6
EX0      EQU  0
        REGS
D6       DSECT
XR6      DS    0F
        DS    3F
        DS    26H
        DS    20C
        DS    1H
HEIGHT   DS    1H
        DS    2F
        DS    1H
        DS    2C
        DS    3H
        DS    56C
        DS    21H
CENT     DS    1F
```

```
RAZE      BCX
          BEXIT EX0
          L      R8,CENT
          LH     R15,HEIGHT          NORMAL CHAR HEIGHT
          SRL    R15,1
          AR     R8,R15
          ST     R8,CENT
          BEXIT EX0
          END
```

RELM

*FUNCTION

*
*UPDATES THE NO. AND POSITION OF RELATIVE Y MAXIMA AND Y MINIMA.
*A STARTING POINT CAN BE A MAX OR MIN, AN ENDING POINT CANNOT

*
*
*

*CALL

* RCS RELMA,I11,EEXIT
*WHERE I1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST

*
*
*

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R7,R8

*

*

USING XR6,R6

REGS

EX0 EQU 0

D6 DSECT

XR6 DS 0F

DS 3F

DS 2H

XT DS 1H

YT DS 1H

CX DS 1H

DY DS 1H

DS 3H

DS 17H

DS 20C

DS 3F

DS 1H

```

      DS      2C
      DS      2H
DEL    DS      1H
      DS      56C
      DS      21H
      DS      1F
      DS      6C
      DS      1H
      DS      1F
      DS      20C
      DS      19H
PYMAX  DS      1H
PYMIN  DS      1H
NYMAX  DS      1H
NYMIN  DS      1H
YMAX   DS      10H
YMIN   EQU     YMAX+10
QYMAX  DS      10C
QYMIN  EQU     QYMAX+5
PYMXX  DS      1H
PYMNX  DS      1H
YMAXX  DS      10H
YMINX  EQU     YMAXX+10
RELM   BOX
      LH      R7,YT
      CH      R7,PYMAX
      BNH     NO
      STH     R7,PYMAX
      MVC     PYMXX(2),XT
*UPWARD STYLUS MOTION
PMIN   CLI     QYMIN,X'01'
      BNE     EXIT
*A MAX HAS OCCURRED PREVIOUSLY
      LH      R7,YT
      LH      R8,PYMIN
      SR      R7,R8
      LPR     R7,R7
      LH      R8,DEL
      SLL     R8,1
      CR      R7,R8
      BNH     EXIT
      MAG(YT-PYMIN)
      2*THINNING DISTANCE
*A MINIMUM DETECTED
      MVI     QYMIN,X'00'
      MVI     QYMAX,X'01'
      MVC     PYMAX(2),YT
      MVC     PYMXX(2),XT
      LH      R7,NYMIN
      LA      R7,1(R7)
      LA      R8,5
      CR      R7,R8
      BNH     NXOK

```

```

NXOK      SR      R7,R7
          STH      R7,NYMIN
          BCTR     R7,0
          SLL      R7,1
          LH       R8,PYMIN
          STH      R8,YMIN(R7)
          LH       R8,PYMNX
          STH      R8,YMINX(R7)
          B        EXIT
NC        CH      R7,PYMIN
          BNL      PMXN
          STH      R7,PYMIN
          MVC      PYMNX(2),XT
          B        PMAX
PMXN      LH      R7,DY
          LTR      R7,R7
          BP       PMIN
*DOWNWARD STYLUS MOTION
PMAX      CLI     QYMAX,X'01'
          BNE      EXIT
*A MIN    HAS OCCURRED PREVIOUSLY
          LH      R7,YT
          LH      R8,PYMAX
          SR      R7,R8
          LPR      R7,R7
          LH      R8,DEL
          SLL      R8,1
          CR      R7,R8
          BNH      EXIT
*A MAXIMUM DETECTED
          MVI     QYMAX,X'00'
          MVI     QYMIN,X'01'
          MVC     PYMIN(2),YT
          MVC     PYMNX(2),XT
          LH      R7,NYMAX
          LA      R7,1(R7)
          LA      R8,5
          CR      R7,R8
          BNH     NNOK
          SR      R7,R7
NNOK      STH      R7,NYMAX
          BCTR     R7,0
          SLL      R7,1
          LH      R8,PYMAX
          STH      R8,YMAX(R7)
          LH      R8,PYMAXX
          STH      R8,YMAXX(R7)
EXIT      BEXIT   EXO
          END

```

MAG(YT-PYMAX)

2*THINNING DISTANCE

SMOOTH

```
*FUNCTION
*
*COMPUTES NEW AVERAGED DATA PT. X OR Y COORDINATE FROM NEW RAW DATA PT.
*COORD. AND PREV. AVERAGED DATA PT. COORD.
*NEW = 3/4 PREV + 1/4 RAW
*
*
*CALL
*      RCS    SMOOTHA,EEXIT
*
*
*INPUT REGISTERS
*
*C(R6) = PREV AVERAGED X OR Y COORD.
*C(R7) = NEW RAW X OR Y COORD.
*
*
*OUTPUT REGISTERS
*
*C(R6) = NEW AVERAGED X OR Y COORD.
*
*
*INTERNAL REGISTERS.  R8
*
*
*      USING XR6,R6
*      REGS
EXO    EQU    0
D6     DSECT
XR6    DS     OF
SMOOTH BOX
      LR      R8,R6
      SRA     R8,2
      SR      R6,R8
      SRA     R7,2
      AR      R6,R7
      BEXIT   EXO
      END
```

TCRNR

*FUNCTION

*

*DETECTS TIME-PAUSE CORNERS BASED ON NPTS, THE NO. OF RAW DATA POINTS

*WHICH HAVE OCCURRED SINCE THE LAST THINNED DATA POINT, AND UPDATES

*NTCUSP, THE NO. OF SUCH CORNERS

*INDEX CUSP=1 IS TIME-CORNER HAS JUST OCCURRED, OTHERWISE CUSP=0.

*

*

*

*CALL

* RCS TCRNRA, I11, EEXIT

*WHERE I1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST

*

*

*

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R8

*

*

USING XR6, R6
REGS

EX0 EQU 0

D6 DSECT

XR6 DS 0F

DS 3F

DS 23H

NT DS 1H

NTC DS 1H

DS 1H

DS 20C

DS 3F

DS 1H

DS 1C

CUSP DS 1C

DS 1H

NPTS DS 1H

DS 1H

DS 56C

DS 21H

DS 1F

DS 6C

DS 1H

DS 1F

DS 20C

DS 17H

```
NTCUSP   DS      1H
PNPTS    DS      1H
TCRNR    BOX
*TIME CORNER DETECTOR
      LH      R8,NT
      LA      R8,1(C,R8)
      STH     R8,NT
      CLI     NT+1,X'02'
      BC      12,CUSPID
      LH      R8,PNPTS
      SLL     R8,2
      AH      R8,PNPTS
      AH      R8,PNPTS
      CH      R8,NPTS
      BC      10,CUSPID
      CLI     CUSP,X'00'
      BC      6,NOCUSP
      R8=6*PNPTS
*CUSP=0
      LH      R8,NTCUSP
      LA      R8,1(C,R8)
      STH     R8,NTCUSP
      MVI     CUSP,X'01'
      LH      R8,NT
      STH     R8,NTC
      B       NOCUSP
CUSPID    MVI     CUSP,X'00'
NCCUSP    LH      R8,NPTS
      CH      R8,PNPTS
      BC      10,NPTS0
      STH     R8,PNPTS
NPTS0     XC      NPTS(2),NPTS
      BEXIT   EX0
      END
```

THIN

*FUNCTION

*

*DETERMINES IF THE CURRENT DATA PT. X OR Y COORDINATE IS FARTHER FROM
*THE PREV. THINNED DATA PT. X OR Y COORD. THAN A DISTANCE DELTA.

*

*

*

*CALL

*

RCS THINA,EEXIT

*

```

*
*
*INPUT REGISTERS
*
*C(R6) = CURRENT COORD
*C(R7) = PREV THINNED COORD
*C(R8) = DELTA
*
*
*
*OUTPUT REGISTERS
*
*C(R7) = NEW THINNED COORD = CURRENT DATA PT COORD, IF SUFFICIENTLY FAR
*C(R7) = PREV THINNED COORD IF NOT FAR
*
*
*INTERNAL REGISTERS.  R8,R9
*
*
      USING XR6,R6
      REGS
EXO    EQU    0
D6     DSECT
XR6    DS     0F
THIN    BOX
      LR      R9,R6
      SR      R9,R7      DIFF
      LPR     R9,R9      MDIFF
      SR      R9,R8
      BC      12,THIN1    EXIT IF MDIFF <= DEL
      LR      R7,R6      T(J)=S(I) IF > DEL
THIN1  BEXIT EXO
      END

```

TURNER

```

*FUNCTION
*
*DETECTS 180 DEGREE CHANGE IN STYLUS DIRECTION THAT OCCURS AFTER A
*SINGLE THINNING DISTANCE
*IF SUCH A TURN IS DETECTED, TTURN+1 = 1 FOR CLOCKWISE TURN, TTURN+1 =2
*FOR COUNTERCLOCKWISE TURN, OTHERWISE TTURN+1 = 0.
*
*
*

```

```
*CALL
*      RCS   TURNER, I11, EEXIT
*WHERE I1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
*
*
*INPUT REGISTERS
*
*C(R6) = I1
*C(R9) = ANGLE (AS ENCODED BY 'CHAREC')
*
*
*INTERNAL REGISTERS.  R7, R8
*
*
      USING XR6, R6
EX0    EQU    0
      REGS
D6     DSECT
XR6    DS     0F
      DS     3F
      DS     8H
PANG   DS     1H
PACANG DS     1H
      DS     16H
      DS     20C
      DS     3F
      DS     1H
      DS     2C
      DS     3H
      DS     56C
      DS     21H
      DS     1F
      DS     6C
TTURN  DS     1H
TURNER BOX
*ANG DD ES NCT EQ PREV ANG
*TEST F OR 180 DEG TURN
*DCES P REV ANG=PREV ACCEPTED ANG?
      LH     R7, PACANG
      CH     R7, PANG
      BC     8, NOTURN
*NO
*DO ANG ? PACANG DIFFER BY 2?
      SR     R7, R9
      LPR    R7, R7
      LA     R8, 2
      CR     R7, R8
      BC     6, NOTURN
*YES
```

```

*IS DIR    ECTION OF TURN CLKWISE?
*OR COU    NTERCLOCKWISE?
          LR    R7,R9
          SH    R7,PANG
          LTR   R7,R7
          BC    2,CCTURN
*POSSIB    LY CLKWISE
*DOES P    ANG EQ 0?
          SR    R7,R7
          CH    R7,PANG
          BC    8,CCT1
CTURN      MVI   TTURN+1,X'01'
          BC    15,TURNX
*POSSIB    LY CCLKWISE
*DOES A    NG EQ 0?
CCTURN     SR    R7,R7
          CR    R7,R9
          BC    8,CTURN
CCT1       MVI   TTURN+1,X'02'
          BC    15,TURNX
*NOT A     180 DEG WITH SINGLE ANGLE
NOTURN     MVI   TTURN+1,X'00'
*EXIT
TURNX      BEXIT EX0
          END

```

TRAVEC

```

*FUNCTION
*
*COMPUTES VECTOR DIRECTION ( 1 OF 16) IF STYLUS HAS MOVED A DISTANCE
*GREATER THAN DELTA (2, 4, 6, OR 8 RASTERS).
*
*
*CALL
*      RCS    TRAVECA,EEXIT
*
*
*INPUT REGISTERS
*
*C(R7) = X COORD OF NEW DATA PT.
*C(R9) = Y COORD OF NEW DATA PT.
*C(R10) = X COORD OF END PT. OF CURRENT VECTOR TRACK
*C(R11) = Y COORD OF END PT. OF CURRENT VECTOR TRACK

```

*C(R15) = 1/2 DELTA

*
*
*

*CUTPUT REGISTERS

*

*IF STYLUS HAS MOVED X OR Y DISTANCE GREATER THAN DELTA

* C(R0) = DIRECTION CODE (X'0'--X'F')

* C(R10) = X END PT. OF UPDATED VECTOR TRACK

* C(R11) = Y END PT. OF UPDATED VECTOR TRACK

*OTHERWISE

* C(R0) = X'0'

* C(R10), C(R11) NOT UPDATED

*
*
*

*INTERNAL REGISTERS R6, R8, R14

*
*

	USING XR6,R6	
EX0	EQU 0	
	REGS	
D6	DSECT	
XR6	DS 0F	
TRAVEC	BOX	
	LA R0,16	RAST/DIR CONSTANT
	SR R14,R14	QUADRANT CODE
	SR R7,R10	X(I) - X(L)
	BC 10,TRAV1	
	LA R14,4(0,R14)	QUAD 2 OR 3
	LPR R7,R7	ABS DX
TRAV1	SR R9,R11	Y(I) - Y(L)
	BC 10,TRAV2	
	LA R14,8(0,R14)	QUAD 3 OR 4
	LPR R9,R9	ABS DY
TRAV2	LR R8,R7	
	SR R6,R6	
	LH R7,TRAST	
	MR R6,R15	
	LR R6,R7	
	LR R7,R8	
	CR R7,R6	
	BC 11,TRAV3	
	CR R9,R6	
	BC 4,TRAV8	
TRAV3	CR R7,R9	ABS DX AND DY
	BC 8,TRAV4	EQUAL
	BC 4,TRAV5	DY > DX
	SLL R9,2	DX > DY
	SR R8,R8	4(ABS DY)
	DR R8,R7	4(ABS DY) / ABS DX

	LA	R9,1(0,R9)	1/2 ROUND
	SRL	R9,1	RESULT/2
	BC	15,TRAV6	
TRAV5	SLL	R7,2	4(ABS DX)
	SR	R6,R6	
	DR	R6,R9	4(ABS DX) / ABS DY
	LA	R7,1(0,R7)	1/2 ROUND
	SRL	R7,1	RESULT /2
	LNR	R9,R7	
	A	R9,TRAVK4	
	BC	15,TRAV6	
TRAV4	LA	R9,2	
TRAV6	A	R9,TQUAD(R14)	
	LPR	R14,R9	
	CR	R14,R0	
	BC	4,TRAV7	
	SR	R14,R14	
TRAV7	LR	R0,R14	
	SLL	R14,1	
	SR	R6,R6	
	LH	R7,TXIN(R14)	
	MR	R6,R15	
	AR	R10,R7	
	SR	R6,R6	
	LH	R7,TYIN(R14)	
	MR	R6,R15	
	AR	R11,R7	
TRAV8	BEXIT	EXO	
TRAVK4	DC	F'4'	
TQUAD	DC	F'0'	
	DC	F'-8'	
	DC	F'-16'	
	DC	F'8'	
TXIN	DS	OH	TABLE FOR 2 RAST VEC
	DC	H'8'	
	DC	H'8'	
	DC	H'8'	
	DC	H'4'	
	DC	H'0'	
	DC	H'-4'	
	DC	H'-8'	
	DC	H'-8'	
	DC	H'-8'	
	DC	H'-8'	
	DC	H'-8'	
	DC	H'-4'	
	DC	H'0'	
	DC	H'4'	
	DC	H'8'	
	DC	H'8'	
TYIN	DS	OH	TABLE FOR 2 RAST VECTORS

```

      CC      H'0'
      CC      H'4'
      CC      H'8'
      CC      H'8'
      CC      H'8'
      CC      H'8'
      CC      H'8'
      CC      H'8'
      CC      H'4'
      CC      H'0'
      CC      H'-4'
      CC      H'-8'
      CC      H'-8'
      CC      H'-8'
      CC      H'-8'
      CC      H'-8'
      CC      H'-4'
TRAST  CC      H'8'      RAST SIZE OF 2 RAST
      END

```

REC

REC Function

*'REC' PERFORMS A FEW SIMPLE TESTS, BUT MOSTLY ACTS AS A LINK BETWEEN
 *'CHAREC' (WHICH CALCULATES A SET OF FEATURES) AND THE PROCEDURES
 *('INTERP' AND OTHER RCS'S) WHICH TEST THESE FEATURES, OR BETWEEN
 *'INTERP' (WHICH PERFORMS MOST OF THE TESTS) AND THE OTHER RCS'S.
 *'REC' HAS AN ORDERED LIST OF THE FEATURES, AND IS GIVEN THE RELATIVE
 *ADDRESS OF THE HEAD OF THE LIST. IT RETURNS A CHARACTER CODE TO
 *'CHAREC'. THE ONLY PARAMETERS MODIFIED BY 'REC' AND ITS RCS'S ARE 'P'
 *'PAD', AND (ONLY FOR COMMA AND SOME SCRIPT CHARACTERS) THE Y
 *COORDINATE OF THE CHARACTER CENTER.

REC Call

```

*      INST  AREC,RECA,I11,I11,EEXIT
*WHERE AREC IS A LINKAGE BETWEEN CHAREC'S CONTEXT AND REC'S CONTEXT
*  RECA IS A LINK TO REC
*  I1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
*  EXIT EXIT IS THE ONLY EXIT
*
*
```

REC Sequence of Information Processing

*****TABLE RE-ENTRY*****

*

*LIST OF 'INTERP' LABELS EQU'D TO CODES
*USED FOR ENTERING 'INTERP' RCS

*

*****RETURNS*****

*

*LIST OF BRANCHES TO 'REC' LABELS
*USED FOR RETURNING TO 'REC' FROM 'INTERP'

*

*****INITIAL CODE*****

*

*INITIALIZE

*IF PERIOD, SET R8, GO TO CALL INTERP

*IF NOT SINGLE STROKE, GO TO SET-UP TABLE RE-ENTRY

*IF CHAR IS NOT LARGE, CALL 'TILDT' TO TEST FOR TILDA

* IF NOT TILDA, GO TO SET-UP TABLE RE-ENTRY

* IF TILDA, GO TO EXIT

*IF CHAR IS LARGE, CALL 'SYMT' TO TEST FOR AND RECOGNIZE GEOMETRIC SYM.

* IF NOT GEOMETRICAL SYMBOL, GO TO SET-UP TABLE RE-ENTRY

* IF GEOMETRICAL SYMBOL, GO TO EXIT

*

*****COMPUTATIONAL SUBROUTINES*****

*

*CALL ON AN RCS TO MAKE A TEST

* RETURN TO EXIT WITH A CHARACTER

* OR TO IN-LINE CODE

* OR TO SET-UP TABLE RE-ENTRY

*

*****SET-UP TABLE RE-ENTRY*****

*

*SET R8 TO ADDRESS OF TABLE RE-ENTRY LABEL

*GO TO CALL INTERP

*

*****IN-LINE CODE*****

*

*MAKE TESTS

*GO TO SOMEPLACE IN 'REC'

*

*****CALL INTERP*****

*

*STORE R8 IN CUSP

*SET UP R14,R15

*CALL 'INTERP'

* IF VALID EXIT, ENTER RETURNS TABLE BASED ON INDEX = R8

* IF ERROR EXIT, GO TO CALL INTERP WITH R8 = ADDRESS OF 'DON'T KNOW'
* LABEL.

*

****EXIT****
*
*EXIT TO 'CHAREC'

REC Program Listing

USING XR1,R1
USING XR3,R3
USING XRX,R6
REGS
SVCS

DSECT1 DSECT
XR1 DS 0F
REGS DS 3F
BANK DS 1F
INDEX DS 1F
DS 1F
EXIT EQU 0
DSECT3 DSECT
XR3 DS 0F
SCRATCH DS 1F
COND DS 1F
DATA DSECT
XRX DS 0F
I1 DS 1F
PAD DS 1F
CCODE DS 1F
XS DS 1H
YS DS 1H
XT DS 1H
YT DS 1H
CX DS 1H
CY DS 1H
MDX DS 1H
MDY DS 1H
PANG DS 1H
PACANG DS 1H
N DS 1H
SN DS 1H
PUP DS 1H
INKIND DS 1H
PQUAD DS 1H
BR56 DS 1H
DXC DS 1H
CYC DS 1H
XRC DS 1H
XLC DS 1H
YTC DS 1H
YBC DS 1H
ASPR DS 1H
NT DS 1H
NTC DS 1H

SCRATCH

INKC	DS	1H
XYE	DS	10C
XYS	DS	10C
WIDTH	DS	1H
HEIGHT	DS	1H
	DS	2F
YCENT	DS	1H
PCHAR	DS	1C
CUSP	DS	1C
NCUSP	DS	1H
NPTS	DS	1H
DEL	DS	1H
P	DS	1C
CHAR	DS	1C
TEMP	DS	1C
TINK	DS	5C
XSP	DS	10C
YSP	DS	10C
XEP	DS	10C
YEP	DS	10C
ALXYJ	DS	8C
XL	DS	1H
YL	DS	1H
XLC	DS	1H
YLC	DS	1H
*AX3 THRU AX02 ARE USED AS NTCUSP,NYMAX,NYMIN, QYMAX, AND QYMIN BY REC		
AX3	DS	1H
AX2	DS	1H
AX1	DS	1H
AX	DS	1H
AX23	DS	1H
AX12	DS	1H
AX01	DS	1H
AX02	DS	1H
NC	DS	1H
C	DS	1H
CYM	DS	1H
CXS	DS	1H
CYS	DS	1H
XRS	DS	1H
XLS	DS	1H
YTS	DS	1H
YBS	DS	1H
CENT	DS	1F
MVC	DS	6C
TURN	DS	1H
TURN	DS	1F
XC	DS	10C
YC	DS	10C
DO	DS	1H
DI	DS	1H

```
D2      DS      1H
D3      DS      1H
D4      DS      1H
D5      DS      1H
D6      DS      1H
D7      DS      1H
*BEWARE, DATA BELOW D7 CANNOT BE REFERRED TO BY TABLE MACROS
D8      DS      1H
D9      DS      1H
D10     DS      1H
D11     DS      1H
D12     DS      1H
D13     DS      1H
D14     DS      1H
D15     DS      1H
DN      DS      1H
NTCUSP  EQU     AX3
NTCSP1  DS      1H
PNPTS   DS      1H
PYMAX   DS      1H
PYMIN   DS      1H
NYMAX   EQU     AX2
NYMX1   DS      1H
NYMIN   EQU     AX1
NYMN1   DS      1H
YMAX    DS      10H
YMIN    EQU     YMAX+10
QYMAX   EQU     AX
QYMIN   EQU     QYMAX+5
QYMX1   DS      10C
QYMN1   EQU     QYMX1+5
PYMXX   DS      1H
PYMNX   DS      1H
YMAXX   DS      10H
YMINX   EQU     YMAXX+10
*BEWARE, DATA BELOW D7 CANNOT BE REFERRED TO BY TABLE MACROS
XYSP    EQU     XYS-DATA      XYS DSECT(R6)
XYEP    EQU     XYE-DATA      XYE DSECT(R6)
*
*
****TABLE RE-ENTRY****
*
*
SPER    EQU     0
SXHBL   EQU     1
S4      EQU     2
SK      EQU     3
AAAA    EQU     4
KVXYM   EQU     5
PADEX   EQU     6
TPLUSM  EQU     7
```

SBMS	EQU	8
SJMU	EQU	9
SUMJU1	EQU	10
SM1	EQU	11
SXMSTR	EQU	12
SOMCQ8	EQU	13
SCPNU	EQU	14
RSB	EQU	15
RSC	EQU	16
RSE	EQU	17
RSF	EQU	18
RSG	EQU	19
RSI	EQU	20
RSJ	EQU	21
RSL	EQU	22
RSM	EQU	23
RSN	EQU	24
RSD	EQU	25
RSR	EQU	26
RSS	EQU	27
RSU	EQU	28
RSV	EQU	29
RSW	EQU	30
RSY	EQU	31
RSZ	EQU	32
SA	EQU	33
SG	EQU	34
SM	EQU	35
SNN	EQU	36
SP	EQU	37
SR	EQU	38
SU	EQU	39
SO	EQU	40
S8	EQU	41
SSTAR	EQU	42
SSCRUB	EQU	43
DK	EQU	44
RSA	EQU	45
SE	EQU	46
REC	PROCS	CLEAR=3,CNTX=6,AUTO=2,PROLG=SINS,ID=8000021C
AHTEST	DC	V(AHSTR1)
KNYT	DC	V(KNYTST)
KNY1TA	DC	V(KNY1T)
KVXYTA	DC	V(KVXYT)
MWTA	DC	V(MWT)
PTST	DC	V(PSTEST)
SYMTA	DC	V(SYMT)
TILCTA	DC	V(TILDT)
TPXYA	DC	V(TPXY)
VERT	DC	V(VERTST)
VFI	DC	V(BFI)

VSDP	DC	V(BSDP)
VSMNW	DC	V(BSMNW)
VSRPRM	DC	V(BSRPRM)
VSSM	DC	V(BSSM)
VSVM	DC	V(BSVM)
VTEST1	DC	V(BTEST1)
VTEST3	DC	V(BTEST3)
VTERP	DC	V(INTERP)
VHITE	DC	V(BHITE)
VRAZE	DC	V(RAZE)

*

*

****RETURNS****

*

*

RETURNS	EQU	*
	BC	15, XAHSTR
	BC	15, XKNY
	BC	15, XFI
	BC	15, XKVXY
	BC	15, XMW
	BC	15, XMWIN
	BC	15, XMW1
	BC	15, XPOST
	BC	15, XRECD
	BC	15, XSDP
	BC	15, XSMNW
	BC	15, XSM1M
	BC	15, XSRPRM
	BC	15, XSSM
	BC	15, XSVM
	BC	15, XTEST1
	BC	15, XTEST3
	BC	15, XTPLUS
	BC	15, XSTRLC
	BC	15, XSALCS
	BC	15, XSG8LC
	BC	15, XSMLCN
	BC	15, XSCBOU
	BC	15, XSNLC
	BC	15, XSPLC
	BC	15, XSRLC
	BC	15, XS8LC
	BC	15, XSULC
	BC	15, XS4LC
	BC	15, XSCPEL
	BC	15, XSCPMW
	BC	15, XSCPYZ
	BC	15, XSCPBS
	BC	15, XSBVMN
	BC	15, XRAZE

```

BC      15,XS8LCV
BC      15,XSULC1
BC      15,XS4MK1
BC      15,XSELCA
*
*
****INITIAL CODE****
*
*
SINS      PROLG
          L      R6,BANK
*MOVE DATA USED BY TABLE MACROS ABOVE FF IN DATA BANK
          MVC     NTCUSP(2),NTCSP1
          MVC     NYMAX(2),NYMX1
          MVC     NYMIN(2),NYMN1
          MVC     QYMAX(10),QYMX1
          CLI     N+1,X'00'
          BC      7,SYMC
          LA      R8,SPER
          BC      15,START
SYMO      EQU     *
          CLI     SN+1,X'01'
          BC      7,REENTR
TILTST    EQU     *
          LH      R15,HEIGHT
          SLL     R15,1
          CH      R15,DYC
          BC      12,LARGE
          CH      R15,DXC
          BC      12,LARGE
*CHARAC   TER IS NOT LARGE
*TEST F   OR TILDA
          RCS     TILDTA,EREENTR,EXRECD
*POSSIB   LY A FLOW CHART SYMBOL
*IS N A   T LEAST 2?
LARGE     CLI     N+1,X'02'
          BC      4,REENTR
*RECOGN   IZE FLOW CHART SYMBCL
          RCS     SYMTA,EREENTR,EXRECD
*
*
****COMPUTATIONAL SUBROUTINES****
*
*
XAHSTR    RCS     AHTEST,EXRECD
XFI       EQU     *
          RCS     VFI,EXRECD
XKNY      RCS     KNYT,EXRECD
XKNY1     RCS     KNY1TA,EXRECD
XKVXY     RCS     KVXYTA,EXRECD
XMWIN     RCS     MWTA,EXRECD

```

XPOST	EQU	*
	RCS	PTST,EXRECD
XSDP	EQU	*
	RCS	VSDP,EXRECD
XSM1M	EQU	*
	RCS	VERT,EXSM1,EXKVXYM,EXPLUSM,EXPADEX
XSRPRM	EQU	*
	RCS	VSRPRM,EXRECD,EXSDP,EXPOST
XSSM	EQU	*
	RCS	VSSM,EXRECD
XSVM	EQU	*
	RCS	V SVM,EXRECD,EXSJMU,EXMW1,EXKNY1,EXSOMO,EXSMJUI
XTEST1	EQU	*
	RCS	VTEST1,EX8LCG,EXSSM
XTEST3	EQU	*
	RCS	VTEST3,EXRECD,EXSBM5
XTPLUS	RCS	TPXYA,EXRECD
XSCPEL	EQU	*
	RCS	VHITE,EXLEU,EXHLO
XSMLCN	EQU	*
	RCS	VHITE,EXSCPNU,EXSM
XSCPMW	EQU	*
	RCS	VHITE,EXLMW,EXSSCRB
XSALCS	EQU	*
	RCS	VHITE,EXRSS,EXSA
XSCPBS	EQU	*
	RCS	VHITE,EXRSS,EXRSB
XMLC	EQU	*
	RCS	VHITE,EXLMW,EXHMY
XSNLC	EQU	*
	RCS	VHITE,EXLOV,EXHBJN
XSMNW	EQU	*
	RCS	VSMNW,EXRECD,EXMLC,EXRAZE
XSRLC	EQU	*
	RCS	VHITE,EXRSN,EXSR
XSPLC	EQU	*
	RCS	VHITE,EXRSR,EXSPP
XSULC1	EQU	*
	RCS	VHITE,EXRSU,EXSU
XS4LC	EQU	*
	RCS	VHITE,EXRSE,EXRSL
XS8LC	EQU	*
	RCS	VHITE,EXLVO,EXS8
XSULC	EQU	*
	RCS	VHITE,EXRSN,EXHRU
XSTRLC	EQU	*
	RCS	VHITE,EXCS,EXSSTAR
XSBVMN	EQU	*
	RCS	VHITE,EXLMNV,EXHBL
XSCBOU	EQU	*
	RCS	VHITE,EXLOU,EXHBM

XSG8LC	EQU	*
	RCS	VHITE,EXRSO,EXSG8
XS8LCV	EQU	*
	RCS	VHITE,EXRSV,EXS8
XSELCA	EQU	*
	RCS	VHITE,EXRSA,EXSE
XRAZE	EQU	*
	RCS	VRAZE,EXRECD

*

*

*****SET-UP TABLE RE-ENTRY*****

*

*

REENTR	EQU	*
	L	R4,INDEX
	L	R7,0(R4)
	STC	R7,TEMP
	LA	R8,AAAA
	BC	15,START
XKVXYM	EQU	*
	LA	R8,KVXYM
	BC	15,START
XPADEX	EQU	*
	LA	R8,PADEX
	BC	15,START
XPLUSM	EQU	*
	LA	R8,TPLUSM
	BC	15,START
XSBM5	EQU	*
	LA	R8,SBM5
	B	START
XSJMU	EQU	*
	LA	R8,SJMU
	B	START
XSMJU1	EQU	*
	LA	R8,SUMJU1
	B	START
XSM1	EQU	*
	LA	R8,SM1
	BC	15,START
XSXMST	LA	R8,SXMSTR
	B	START
XSOMO	EQU	*
	LA	R8,SCMOQ8
	B	START
XHBL	LA	R8,SXHBL
	B	START
XSCPNU	LA	R8,SCPNU
	B	START
XRSA	LA	R8,RSA
	B	START

XRSB	LA	R8,RSB
	B	START
XRSC	LA	R8,RSC
	B	START
XRSE	LA	R8,RSE
	B	START
XRSF	LA	R8,RSF
	B	START
XRSG	LA	R8,RSG
	B	START
XRSI	LA	R8,RSI
	B	START
XRSJ	LA	R8,RSJ
	B	START
XRSL	LA	R8,RSL
	B	START
XRSM	LA	R8,RSM
	B	START
XRSN	LA	R8,RSN
	B	START
XRSC	LA	R8,RSO
	B	START
XRSR	LA	R8,RSR
	B	START
XRSS	LA	R8,RSS
	B	START
XRSU	LA	R8,RSU
	B	START
XRSV	LA	R8,RSV
	B	START
XRSW	LA	R8,RSW
	B	START
XRSY	LA	R8,RSY
	B	START
XRSZ	LA	R8,RSZ
	B	START
XSA	LA	R8,SA
	B	START
XSE	LA	R8,SE
	B	START
XSG	LA	R8,SG
	B	START
XSM	LA	R8,SM
	B	START
XSNN	LA	R8,SNN
	B	START
XSP	LA	R8,SP
	B	START
XSR	LA	R8,SR
	B	START
XSU	LA	R8,SU

```

      B      START
XS0   LA     R8,S0
      B      START
XS8   LA     R8,S8
      B      START
XSSTAR LA     R8,SSTAR
      B      START
XSSCRB LA     R8,SSCRUB
      B      START

```

*

*

*****IN-LINE CODE*****

*

*

```

XMW   EQU    *
      LA     R13,3
      B      XMWIN
XMW1  EQU    *
      LA     R13,2
      B      XMWIN
XS4MK1 SR     R15,R15
      LA     R13,2
K4    LA     R7,0(R6,R15)
      TM     XYEP+1(R7),X'03'
      BC     1,XSKX
      BXLE   R15,R13,K4
      LA     R8,S4
      B      START
XSKX  TM     XYSP+1(R7),X'0F'
      BC     8,XSXMST
      LA     R8,SK
      B      START
XHBM  EQU    *
      TM     XYE+1,X'0C'
      BC     1,XSM
      B      XRSB
XLMW  TM     XYE+1,X'0C'
      BC     8,XRSW
      B      XRSM
XHMY  TM     XYE+1,X'0C'
      BC     12,XRSY
      B      XSM
XLOV  LH     R7,XRC
      SH     R7,XLC
      SRL    R7,2      1/4 CHAR WIDTH
      LH     R8,XEP
      SH     R8,YMAXX
      LPR    R8,R8
      CR     R8,R7
      BC     4,XRSD
      B      XRSV

```

XHBJN	TM	XYS+1,X'0C'
	BC	12,XRSF
	TM	XYE+1,X'0C'
	BC	5,XRSB
	B	XSNN
XHLO	TM	XYE+1,X'03'
	BC	8,XRSL
	B	XS0
XHRU	EQU	*
*U IF 2ND MAX IN RIGHT 1/2, OTHERWISE R		
	LH	R8,DXC
	SRL	R8,1
	LH	R7,XRC
	SR	R7,R8
	CH	R7,YMAXX+2
	BC	2,XSR
	B	XSU
XSCPYZ	EQU	*
	LH	R7,YMINX+2
	CH	R7,YMAXX
	BC	4,XRSZ
	B	XRSY
XCS	EQU	*
*C IF 2ND OR 3RD ANGLE IS 0		
	TM	CODE,X'3C'
	BC	8,XRSC
	TM	CODE,X'0C'
	BC	8,XRSC
	B	XRSS
X8LCG	CLI	CODE,X'B7'
	BC	8,XRSG
	B	XRECD
XLEU	CLI	N+1,X'05'
	BC	2,XRSU
	B	XRSE
XLMNV	EQU	*
	CLI	N+1,X'05'
	BC	8,XNV
	BC	4,XRSE
	CLI	N+1,X'06'
	BC	2,XRSM
	B	XSCPNU
XNV	TM	XYE+1,X'08'
	BC	1,XSCPNU
	B	XRSV
XLQU	TM	XYE+1,X'08'
	BC	8,XRSO
	B	XRSU
XLVC	TM	CODE,X'08'
	BC	1,XRSO
	B	XRSV

```

XSG8      TM      XYE+1,X'0C'
          BC      8,XS8
          B       XSG

*
*
****CALL INTERP****
*
*
* THIS IS THE INTERPRETER          R8 IS THE 'INSTRUCTION COUNTER'
*                                COND CONTAINS THE 'CONDITION CODE'
*
START      EQU      *
          STC      R8,CUSP
          LA       R14,COND
          LA       R15,SCRCH
          RCS      VTERP,EEX0,EEX1          CALL THE INTERPRETER
EX0        EQU      *
          ST       R8,SCRCH
          SWTCH    INDEX=SCRCH,TABLE=RETURNS
EX1        EQU      *
          LA       R8,DK
          BC       15,START          SIGNAL NOT UNDERSTOOD

*
*
****EXIT****
*
*
XRECD      EQU      *
          EPLOG    EXIT
          END

```

INTERP

INTERP Function

*'INTERP' PERFORMS SEQUENCES OF TESTS ON ENCODED 1-BYTE FEATURES
 *THEREBY INCLUDING NEARLY ALL OF THE DECISION-MAKING TREE STRUCTURE.
 *'INTERP' IS ENTERED VIA 'REC' AND CALLS RCS'S (WHICH PERFORM THE MORE
 *COMPLICATED TESTS) VIA 'REC'. A 'TABLE' MACRO (DESCRIBED BELOW) IS
 *USED TO PERFORM THE TESTS.

INTERP Call

```
*
*      RCS      INTERPA,EVALID,EERROR
* WHERE INTERPA IS A LINK TO INTERP
*      EXIT VALID IS THE NORMAL EXIT
*      EXIT ERROR IS THE ERROR EXIT
```

INTERP Sequence of Information Processing

```
****INTERPRETER****
```

```
*
*INTERPRETER FOR 'TABLE' MACRO
```

```
****TABLE EXITS****
```

```
*
*LIST OF 'REC' LABELS EQU'D TO CODES
*USED FOR RETURNING TO 'REC' ROUTINE
```

```
****TABLE TESTS****
```

```
*
*CALLS ON THE 'TABLE' MACRO TO PERFORM SEQUENCES OF TESTS ON (OR MOD-
*IFICATIONS OF) ENCODED 1-BYTE FEATURES. THE CALL HAS THE FOLLOWING
*FORM:
```

```
*LABEL    TABLE /OPI,PI,CI/,C11,L11,C12,L12,...,C1K,L1K,/OP2,P2,C2/,C21,X
*              L21,C22,...
```

```
*WHERE CONTINUATION TO NEXT CARD IS INDICATED BY A NON-BLANK COLUMN 72
```

```
*      OPI IS AN ABBREVIATED OP CODE
```

```
*      TM = TEST UNDER MASK
```

```
*      MV = MOVE IMMEDIATE
```

```
*      NI = AND IMMEDIATE
```

```
*      CL = COMPARE LOGICAL IMMEDIATE
```

```
*      OI = OR IMMEDIATE
```

```
*      X2 = EXCLUSIVE OR IMMEDIATE
```

```
*      TR = TRANSLATE
```

```
*      SS = SWITCH
```

```
*      EX = EXIT FROM TABLE
```

```
*      IF OPI = TR
```

```
*          PI = THE TRANSLATION INDEX
```

```
*          CI = 00
```

```
*          CIJ = 0
```

```
*          LIJ = START OF A LIST OF DC'S
```

```
*      IF OPI = SS
```

```
*          PI = TEMP
```

```
*          CI = 00
```

```
*          CIJ = 0
```

```
*          LIJ = START OF LIST OF BRANCHES
```

```
*      IF OPI = EX
```

```
*          PI = A 'REC' LABEL
```

```
*          CI = 0
```

```
*          CIJ,LIJ ARE OMITTED
```

```

* OTHERWISE
*   PI = THE FEATURE TO BE TESTED OR MODIFIED (ONLY 'P', 'PAD', CR
*       'CHAR' MAY BE MODIFIED)
*   CI = THE 2 CHARACTER 1-BYTE NUMBER WHICH PI IS TESTED AGAINST OR
*       MODIFIED BY
*   CIJ = THE CONDITION CODE UNDER WHICH THE SEQUENCE OF CONTROL
*         BRANCHES TO LABEL LIJ
*
****SET-UP CHARACTER CODE****
*
*MOVE CHARACTER CODE INTO 'CHAR'
*BRANCH TO THE SET OF ESCAPES
*
****PAD TABLE****
*
*LIST OF BRANCHES TO 'INTERP' LABELS
*USED FOR ENTERING 'INTERP' BASED ON VALUE OF 'PAD'
*
****4 DIRECTION TABLE****
*
*LIST OF BRANCHES TO 'INTERP' LABELS
*USED FOR ENTERING 'INTERP' BASED ON THE VALUES ON THE FIRST FOUR
*DIRECTIONS IN THE DIRECTION SEQUENCE AS ENCODED BY 'ANG4'
*
****SET OF ESCAPES****
*
*EXITS FROM 'INTERP' TO 'REC'
*
****ENTRY SWITCH****
*
*LIST OF BRANCHES TO 'INTERP' LABELS
*USED FOR ENTERING 'INTERP' FROM 'REC'

```

INTERP Program Listing

	USING	XRX,R6	
	REGS		
TM	EQU	X'91'	TEST UNDER MASK
MV	EQU	X'92'	MOVE IMMEDIATE
NI	EQU	X'94'	AND IMMEDIATE
CL	EQU	X'95'	COMPARE LOGICAL IMMEDIATE
CI	EQU	X'96'	OR IMMEDIATE
X2	EQU	X'97'	EXCLUSIVE OR IMMEDIATE
TR	EQU	X'99'	TRANSLATE
SS	EQU	X'9A'	SWITCH
EX	EQU	X'9B'	EXIT THE TABLE
DATA	DSECT		
XRX	DS	OF	
I1	DS	1F	
PAD	DS	1F	
CCDE	DS	1F	
XS	DS	1H	
YS	DS	1H	

XT	DS	1H
YT	DS	1H
DX	DS	1H
DY	DS	1H
MDX	DS	1H
MDY	DS	1H
PANG	DS	1H
PACANG	DS	1H
N	DS	1H
SN	DS	1H
PUP	DS	1H
INKIND	DS	1H
PQUAD	DS	1H
BR56	DS	1H
DXC	DS	1H
DYC	DS	1H
XRC	DS	1H
XLC	DS	1H
YTC	DS	1H
YBC	DS	1H
ASPR	DS	1H
NT	DS	1H
NTC	DS	1H
INKE	DS	1H
XYE	DS	10C
XYS	DS	10C
WIDTH	DS	1H
HEIGHT	DS	1H
	DS	2F
YCENT	DS	1H
PCHAR	DS	1C
CUSP	DS	1C
NCUSP	DS	1H
NPTS	DS	1H
DEL	DS	1H
P	DS	1C
CHAR	DS	1C
TEMP	DS	1C
TINK	DS	5C
XSP	DS	10C
YSP	DS	10C
XEP	DS	10C
YEP	DS	10C
ALXYJ	DS	8C
XL	DS	1H
YL	DS	1H
XLO	DS	1H
YLO	DS	1H
*AX3 THRU AX02 ARE USED AS NTCUSP, ETC. BY REC		
AX3	DS	1H
AX2	DS	1H

AX1	DS	1H
AX	DS	1H
AX23	DS	1H
AX12	DS	1H
AX01	DS	1H
AX02	DS	1H
NC	DS	1H
C	DS	1H
DYM	DS	1H
DXS	DS	1H
DYS	DS	1H
XRS	DS	1H
XLS	DS	1H
YTS	DS	1H
YBS	DS	1H
CENT	DS	1F
MVC	DS	6C
TURN	DS	1H
TURN	DS	1F
XC	DS	10C
YC	DS	10C
D0	DS	1H
D1	DS	1H
D2	DS	1H
D3	DS	1H
D4	DS	1H
D5	DS	1H
D6	DS	1H
D7	DS	1H
*BEWARE, DATA BELOW D7 CANNOT BE REFERRED TO BY TABLE MACROS		
D8	DS	1H
D9	DS	1H
D10	DS	1H
D11	DS	1H
D12	DS	1H
D13	DS	1H
D14	DS	1H
D15	DS	1H
DN	DS	1H
NTCUSP	EQU	AX3
NTCSP1	DS	1H
PNPTS	DS	1H
PYMAX	DS	1H
PYMIN	DS	1H
NYMAX	EQU	AX2
NYMX1	DS	1H
NYMIN	EQU	AX1
NYMNI	DS	1H
YMAX	DS	10H
YMIN	EQU	YMAX+1C
QYMAX	EQU	AX

```

QYMX1    DS      10C
QYMIN    EQU     QYMAX+5
PYMXX    DS      1H
PYMNX    DS      1H
YMAXX    DS      10H
YMINX    EQU     YMAXX+10
*BEWARE, DATA BELOW D7 CANNOT BE REFERRED TO BY TABLE MACROS
XYSP     EQU     XYS-DATA                XYS DSECT(R6)
XYEP     EQU     XYE-DATA                XYE DSECT(R6)
EXO      EQU     0
EX1      EQU     4

```

```

*
*
****INTERPRETER****
*

```

INTERP	BOX		
START	EQU	*	ADVANCE
	LA	R7,BASE	
	LA	R8,GPSW	GENERAL PURPOSE SWITCH
	MVI	0(R14),X'00'	CLEAR THE CONDITION CODE
AGAIN	EQU	*	
	CLI	0(R8),X'90'	COMMAND
	BC	4,BRANCH	MUST BE A BRANCH
	CLI	0(R8),X'9F'	
	BC	2,BRANCH	
MAGIC	EQU	*	
	CLI	0(R8),X'98'	STANDARD COMMAND
	BC	2,T99	
COMM	EQU	*	
	MVC	0(2,R15),0(R8)	
	SR	R9,R9	
	IC	R9,2(R8)	
	LA	R9,DATA(R9)	GET THE DATA
	MVC	2(2,R15),OPER	
	EX	0,0(R15)	
	LA	R8,3(R8)	
	BAL	R10,COMM1	
COMM1	EQU	*	
	ST	R10,0(R14)	
	BC	15,AGAIN	
T99	EQU	*	
	CLI	0(R8),X'9F'	REALLY A BRANCH
	BC	2,BRANCH	
	CLI	0(R8),X'99'	
	BC	7,T9A	
	MVC	0(2,R15),3(R8)	MUST BE A TRANSLATE
	LH	R9,0(R15)	
	LA	R9,0(R7,R9)	FIND THE TABLE
	SR	R10,R10	
	IC	R10,2(R8)	

	LA R10,DATA(R10)	FIND THE DATA
	MVC TEMP(1),0(R10)	
	TR TEMP(1),0(R9)	
	LA R8,5(R8)	ADVANCE THE IC
	BC 15,AGAIN	
T9A	EQU *	
	CLI 0(R8),X'9A'	
	BC 7,T9B	
	SR R9,R9	MUST BE A SWITCH
	IC R9,2(R8)	
	LA R10,DATA(R9)	
	IC R9,0(R10)	
	SLL R9,1	
	MVC 0(2,R15),3(R8)	
	LH R10,0(R15)	
	LA R10,0(R7,R10)	GET TABLE
	LA R10,0(R9,R10)	GET T ABLE ENTRY
	MVC 2(2,R15),0(R10)	
	BC 15,AGREE	
T9B	EQU *	
	CLI 0(R8),X'9B'	
	BC 7,ERROR	
	MVI 0(R15),X'00'	
	MVC 1(1,R15),1(R8)	MUST BE AN EXIT
	LH R8,0(R15)	
	BEXIT EX0	
BRANCH	EQU *	MUST BE A BRANCH
	MVC 0(1,R15),0(R14)	
	NI 0(R15),X'30'	
	MVC 2(2,R15),0(R8)	
	MVC 1(1,R15),2(R15)	
	NI 1(R15),X'F0'	
	SR R10,R10	
	IC R10,0(R15)	
	SRL R10,2	
	LA R10,TESTM(R10)	
	EX 0,0(R10)	
	BC 1,AGREE	
	LA R8,2(R8)	
	BC 15,AGAIN	
AGREE	EQU *	
	NI 2(R15),X'0F'	
	LH R9,2(R15)	
	LA R8,0(R7,R9)	
	BC 15,AGAIN	
ERROR	EQU *	
	BEXIT EX1	
TESTM	EQU *	
	TM 1(R15),X'80'	
	TM 1(R15),X'40'	
	TM 1(R15),X'20'	

OPER TM 1(R15),X'10'
 DC X'9000'

*

*

****TABLE EXITS****

*

*

* TABLE EXITS

XAHSTR	EQU	0
XKNY	EQU	1
XFI	EQU	2
XKVXY	EQU	3
XMW	EQU	4
XMWIN	EQU	5
XMW1	EQU	6
XPOST	EQU	7
XRECD	EQU	8
XSDP	EQU	9
XSMNW	EQU	10
XSM1M	EQU	11
XSRPRM	EQU	12
XSSM	EQU	13
XSVM	EQU	14
XTEST1	EQU	15
XTEST3	EQU	16
XTPLUS	EQU	17
XSTRLC	EQU	18
XSALCS	EQU	19
XSG8LC	EQU	20
XSMLCN	EQU	21
XSCBDU	EQU	22
XSNLC	EQU	23
XSPLC	EQU	24
XSRLC	EQU	25
XS8LC	EQU	26
XSULC	EQU	27
XS4LC	EQU	28
XSCPEL	EQU	29
XSCPMW	EQU	30
XSCPYZ	EQU	31
XSCPBS	EQU	32
XS8VMN	EQU	33
XRAZE	EQU	34
XS8LCV	EQU	35
XSULC1	EQU	36
XS4MK1	EQU	37
XSELCA	EQU	38

*

*

****TABLE TESTS****

*

```

*
BASE      EQU      *
AHSTR     TABLE  /MV,PAD+3,3A/,15,AHSTRX
BR        TABLE  /TM,CODE+1,80/,1,TEST3,/TM,CODE+1,10/,8,TEST3,/CL,N+1,05X
          /,8,SW,15,SM
FIME      TABLE  /MV,P,00/,15,FIME1
FIME1     TABLE  /CL,P,02/,8,SE,/CL,P,01/,8,SPOUND,/MV,PAD+3,23/,15,FI
G6ETST    TABLE  /TR,P,00/,0,PBB,/SS,TEMP,00/,0,PBBX
PBB       DS      0H
          DC      X'03'
          DC      2X'0C'
          DC      2X'03'
          DC      3X'02'
          DC      4X'03'
PBBX      TABLE  15,OQ
          TABLE  15,SQ
          TABLE  15,S8
          TABLE  15,SG6X
KNYM      TABLE  /MV,P,00/
KNYM1     TABLE  /CL,P,01/,8,MW,/MV,PAD+3,24/,15,KNY
KVXYM     TABLE  /CL,P,01/,8,KNYM,/CL,P,02/,8,AHSTR,/MV,PAD+3,06/,/MV,P,0X
          B/,15,KVXY
LPRSLA    TABLE  /CL,XYE+1,0F/,8,SSLASH,15,SLPAR
MK        TABLE  /MV,P,00/,/TM,XYS+3,02/,1,SM,15,SK
*P IF SECOND STROKE IS NOT SINGLE ANGLE OR DOUBLE ANGLE
CQ        TABLE  /CL,N+1,02/,2,SP,/TM,XYS+3,0C/,5,SQ,/TM,XYS+3,03/,1,SQ,1X
          5,SO
PADEX     TABLE  /SS,PAD+3,00/,0,PADT
PARSLA    TABLE  /CL,XYS+1,00/,8,LPRSLA,/TM,XYS+1,02/,8,SLPAR,1,SRPAR
PGTR2     TABLE  /MV,P,C2/,15,PADEX
TEST5     TABLE  /MV,PAD+3,1E/,15,SVM
TPLUSM    TABLE  /CL,P,C1/,8,AHSTR,/CL,P,02/,8,FIME,/MV,PAD+3,36/,/MV,P,0X
          A/,15,TPLUS                                     36
XMK       TABLE  /MV,P,00/,15,XMK1
XMK1      TABLE  /CL,P,C1/,8,SK,/CL,P,02/,8,AHSTR,/MV,PAD+3,18/,/TM,XYS+1X
          ,02/,8,KVXY,/TM,XYS+3,02/,8,KVXY,/TM,XYE+3,02/,8,SR,/TM,X
          XYE+3,0C/,1,SD,15,SP
          EJECT
SASTAR    TABLE  /TM,XYE+1,0C/,12,SALC,/EX,XSTRLC,0/
SALC      TABLE  /TM,XYS+1,0C/,12,RSF,/TM,XYE+1,0C/,8,RSV,/TM,CCODE,08/,1,X
          SALCS,/TM,CCODE,04/,8,SCPBS,/TM,CCODE,02/,1,SALCS,15,SCPBS
SALCF     TABLE  /TM,XYS+1,0C/,1,SA,/MV,CHAR,86/,15,RECD
SALCS     TABLE  /EX,XSALCS,0/
SAMSTR    TABLE  /MV,P,00/,15,SAMST1
SAMST1    TABLE  /CL,P,C1/,8,SSTAR,/MV,PAD+3,15/,15,SA                                     15
SA7       TABLE  /CL,P,02/,8,SA,/MV,PAD+3,37/,15,S7
SBARM     TABLE  /CL,P,02/,8,SETEQM,2,PGTR2,/CL,P,01/,8,STTPLS,/CL,SN+1,0X
          1/,2,PGTR2,/MV,P,02/,/CL,CCODE,00/,8,SMINUS,/CL,ASPR,02/,X
          10,SSLASH,5,SMINUS
SBARMK    TABLE  /CL,P,C1/,8,S4MK,/CL,P,02/,8,SETEQM,2,PGTR2,/MV,PAD+3,0DX
          /,15,SCRKRT

```

SBDPR	TABLE /TR,BR56+1,00/,0,II,/SS,TEMP,00/,0,III
II	CS OH
	DC X'00'
	DC 4X'00'
	DC X'01'
	DC 4X'00'
	DC X'01'
	DC 4X'00'
	DC X'01'
	DC X'02'
SBDPR1	TABLE /TR,BR56+1,00/,0,KK,/SS,TEMP,00/,0,KKK
KK	CS OH
	DC X'03'
	DC X'01'
	DC X'02'
	DC 2X'04'
	DC X'00'
	DC X'01'
	DC X'04'
	DC 4X'00'
	DC X'03'
	DC X'01'
	DC X'02'
	DC X'03'
	DC X'00'
KKK	TABLE 15,SDP
	TABLE 15,DK
	TABLE 15,SB
	TABLE 15,TEST3
	TABLE 15,SBR1
III	TABLE 15,TEST3
	TABLE 15,SDP
	TABLE 15,SPRMA
SBM5	TABLE /MV,P,00/,15,SBM51
SBM51	TABLE /CL,P,02/,8,S5,/MV,PAD+3,26/,15,SB
SBR1	TABLE /CL,N+1,06/,12,SR,/TM,CODE+1,00/,12,SB,/CL,N+1,07/,8,SR,X
	15,SB
SCG	TABLE /CL,N+1,05/,2,SG,12,SCLC
SCLC	TABLE /CL,QYMIN,03/,8,RSD,15,SCC
SCLBRC	TABLE /CL,NTCUSP+1,01/,10,SLBRAC,/CL,ASPR+1,05/,2,SLBRAC,15,SCX
	C
SCMEG	TABLE /CL,P,01/,8,SK,/CL,P,02/,8,SE,/CL,P,03/,8,SG,/MV,PAD+3,1X
	6/,/MV,P,05/,/CL,NCUSP+1,01/,2,SLBRAC,8,SCLBRC,/CL,ASPR+X
	1,08/,2,SLPAR,15,SCC
SCMG	TABLE /TR,P,00/,0,PAA,/SS,TEMP,00/,0,PAAA
PAA	CS OH
	DC 2X'02'
	DC 2X'00'
	DC X'02'
	DC 3X'01'
	DC 4X'02'

PAAA	TABLE 15,SG	
	TABLE 15,S8	
	TABLE 15,SCMG1	
SCMG1	TABLE /MV,PAD+3,05/,/MV,P,05/,15,SCC	05
SCOM	TABLE /TM,XYE+1,0C/,8,SOMQ8,5,SCMG	
SCOMAM	TABLE /CL,PAD+3,15/,8,SAMST1,/CL,P,01/,8,SXMSTR,/CL,P,02/,8,SAX	
	MSTR,/CL,P,0A/,8,SSTAR,/MV,PAD+3,14/,15,SKARAT	14
SCRKRT	TABLE /CL,NTCUSP+1,01/,4,SCC,15,SRKRT	
SDMH	TABLE /CL,P,01/,8,SH,/MV,PAD+3,22/,15,SD	
SEGO6M	TABLE /CL,PAD+3,09/,8,SOMQ8,/CL,PAD+3,21/,8,OQ,/CL,PAD+3,0A/,X	
	8,G6ETST,/TM,XYE+1,08/,8,SOMQ8,15,G6ETST	09
SETEQ	TABLE /MV,PAD+3,07/,/MV,P,08/,15,SEQL	07
SETEQM	TABLE /CL,PAD+3,07/,8,SETXX,/MV,P,00/,15,SETXX	
SETXX	TABLE /CL,P,02/,8,STPE,/CL,P,01/,8,FIME,/TM,XYS+1,08/,1,SETEQ,X	
	/TM,XYE+1,0C/,12,SETEQ,/MV,PAD+3,06/,/MV,P,11/,15,SX	
SETIAK	TABLE /MV,PAD+3,19/,/MV,P,09/,15,SK	19
SEQ	TABLE /CL,N+1,04/,2,RSQ,15,SE	
SFE	TABLE /CL,PAD+3,20/,8,SFME1,/CL,PAD+3,1F/,8,SFME,/CL,P,08/,8,SX	
	E,/CL,P,07/,2,SFE1,/CL,P,05/,10,OQ,/CL,P,01/,8,STPK,/CL,X	
	P,02/,8,SFME,15,SFE1	
SFE1	TABLE /MV,PAD+3,1F/,15,LPRSLA	1F
SFME	TABLE /MV,P,CC/,15,SFME1	
SFME1	TABLE /CL,P,02/,8,SE,/MV,PAD+3,20/,/CL,XYE+1,0C/,8,SXMSTR,/CL,X	
	XYE+3,CC/,8,SXMSTR,/CL,XYE+1,0D/,8,SXMSTR,/CL,XYE+3,0D/,X	
	8,SXMSTR,/TM,XYE+3,02/,1,SY,15,SF	
SGS	TABLE /CL,N+1,04/,12,S65,/TM,CODE+1,C0/,8,SG8,4,SSM,/TM,CODE+1X	
	,30/,5,SEGO6M,/EX,XSELCA,0/	
SGSCR8	TABLE /CL,N+1,04/,2,SSCRUB,12,SG8	
SGS06M	TABLE /TM,TURN,08/,1,SEGO6M,12,SSM	
SG06M	TABLE /TR,8R56+1,00/,0,FF,/SS,TEMP,00/,0,FFF	
FF	DS 0H	
	DC 4X'02'	
	DC 4X'01'	
	DC 4X'00'	
	DC X'03'	
	DC 3X'00'	
	DC X'00'	
FFF	TABLE 15,SEGO6M	
	TABLE 15,S8	
	TABLE 15,SG8	
	TABLE 15,RSO	
SG069M	TABLE /TR,8R56+1,00/,0,EE,/SS,TEMP,00/,0,EEEX	
EE	DS 0H	
	DC 9X'02'	
	DC X'03'	
	DC 2X'04'	
	DC 4X'01'	
	DC X'00'	
EEEX	TABLE 15,SCOM	
	TABLE 15,S9MG	
	TABLE 15,SG8	

	TABLE 15,S8	
	TABLE 15,SEG06M	
SG6X	TABLE /MV,PAD+3,0A/,/MV,P,06/,/CL,NTCUSP+1,01/,10,SG,/TM,XYE+1X 02/,1,S6,/TM,XYE+1,01/,8,SG,/TM,XYE+1,0C/,1,S6,15,SG	
SG8	TABLE /TM,XYE+1,0C/,5,SG,/EX,XSG8LC,0/	
SG81	TABLE /TM,XYE+1,0C/,1,SG,15,S8	
SJMU	TABLE /MV,P,CO/,15,SUMXX	
SUMXX	TABLE /CL,P,C2/,8,SU,/MV,PAD+3,27/,15,SJ	
SK5	TABLE /CL,P,C1/,8,SK,/MV,PAD+3,28/,15,S5	
SLKRTM	TABLE /MV,PAD+3,10/,/CL,P,00/,8,SLKRT,2,SRPRM	
SLMEK4	TABLE /TR,P,CO/,0,PEE,/SS,TEMP,00/,0,PEEE	
PEE	DS OH	
	DC X'03'	
	DC X'00'	
	DC X'05'	
	DC 3X'03'	
	DC X'04'	
	DC X'03'	
	DC X'01'	
	DC 2X'03'	
	DC X'02'	
PEEE	TABLE 15,S4Y	
	TABLE 15,STPE	
	TABLE 15,SK	
	TABLE 15,SLX	
	TABLE 15,SQ	
	TABLE 15,SXME	
SLX	TABLE /MV,PAD+3,1A/,/CL,ASPR+1,08/,2,SLPAR,12,SL	1A
SMC	TABLE /TR,BR56+1,00/,0,DD,/SS,TEMP,00/,0,DDD	
DD	DS OH	
	DC 4X'02'	
	DC X'00'	
	DC X'01'	
	DC X'03'	
	DC X'04'	
	DC X'06'	
	DC X'02'	
	DC X'05'	
	DC X'06'	
	DC 3X'02'	
	DC X'04'	
	DC X'01'	
DDD	TABLE 15,SG	
	TABLE 15,SCMG	
	TABLE 15,SSM	
	TABLE 15,SEG06M	
	TABLE 15,S9MG	
	TABLE 15,S6S	
	TABLE 15,SE	
SMLC	TABLE /CL,N+1,05/,2,SCPMWK,/TM,XYS+1,08/,8,SCPPYZ,/CL,CYMAX+1,X 00/,2,SCPBH,/EX,XSMLCN,0/	

SM1	TABLE	/CL,ASPR+1,20/,2,S1,/CL,ASPR+1,0C/,12,PARSLA,/TM,XYS+1,0X 2/,1,S1RPAR,/TM,XYE+1,02/,1,S1,8,SLPAR,15,S1RPAR
SNMA	TABLE	/CL,P,02/,8,SA,/CL,P,01/,8,RSX,/MV,PAD+3,29/,/TM,CODE,0CX /,1,SNLC,4,SNLC1,/TM,XYS+1,0C/,12,S2LCY,/EX,XSCBCU,0/
SNLC	TABLE	/EX,XSNLC,0/
SNLC1	TABLE	/TM,CODE,08/,1,SNLC,/TM,XYS+1,0C/,1,SNLC,/TM,XYE+1,08/,1X S2,15,SNLC
SPRMA	TABLE	/MV,PAD+3,0C/,/CL,P,01/,8,SM,/CL,P,02/,8,SA,/TM,XYE+1,0CX /,12,SPLC,/TM,XYE+1,02/,1,SD,15,SRLC
SPLC	TABLE	/EX,XSPLC,0/
SRLC	TABLE	/CL,QYMAX+1,00/,8,SRLCX,/CL,N+1,05/,2,RSK,15,RSH
SRPRMJ	TABLE	/CL,P,02/,8,SJ,/CL,P,06/,8,SC,/MV,PAD+3,2A/,15,SRPAR
SS589M	TABLE	/TR,BR56+1,00/,0,GG,/SS,TEMP,00/,0,GGG
GG	DS	0H
	DC	X'05'
	DC	2X'06'
	DC	X'01'
	DC	X'07'
	DC	X'04'
	DC	2X'01'
	DC	X'03'
	DC	X'04'
	DC	X'02'
	DC	X'01'
	DC	X'03'
	DC	X'04'
	DC	X'01'
	DC	X'02'
	DC	X'00'
GGG	TABLE	15,S9MG
	TABLE	15,DK
	TABLE	15,SSM
	TABLE	15,S8
	TABLE	15,TEST1
	TABLE	15,SE
	TABLE	15,RSQ
	TABLE	15,RSQ
SS8M	TABLE	/TR,BR56+1,00/,0,HH,/SS,TEMP,00/,0,HHH
FH	DS	0H
	DC	5X'01'
	DC	X'03'
	DC	4X'01'
	DC	X'02'
	DC	4X'01'
	DC	X'02'
	DC	X'00'
FHH	TABLE	15,SSM
	TABLE	15,S8
	TABLE	15,DK
	TABLE	15,TEST1
STPA	TABLE	/MV,P,00/,/MV,PAD+3,28/,15,SALCF

STPE	TABLE /MV,P,CC/,/MV,PAD+3,2C/,15,SE	
STPH8	TABLE /CL,P,C1/,8,SH,/CL,P,02/,8,SF,/MV,PAD+3,2D/,15,S8LC	
S8LC	TABLE /EX,XS8LC,0/	
STPK	TABLE /CL,P,C1/,8,KVXY,15,STPK1	
STPK1	TABLE /MV,P,CO/,/MV,PAD+3,2E/,15,SK	
STPJ	TABLE /MV,PAD+3,2F/,15,SJ	
STPM	TABLE /CL,P,C1/,8,SM,/MV,PAD+3,30/,15,SYLC	
SYLC	TABLE /CL,N+1,05/,2,RSM,/CL,QYMIN,03/,8,RSN,15,SY	
STP5	TABLE /CL,PAD+3,03/,8,S5,/CL,P,02/,8,S5,/MV,PAD+3,03/,/TM,XYE+X 1,08/,8,S8,/CL,CODE,CB/,6,S5,/TM,TURN,04/,8,S6,15,S5	
STP6	TABLE /MV,PAD+3,02/,/CL,P,01/,8,SO,/CL,P,02/,8,S5,/CL,P,06/,8,X S8,/MV,P,06/,15,S6	
STTPLS	TABLE /MV,P,CO/,15,TPLUSM	
SUMAM	TABLE /MV,PAD+3,0F/,/CL,P,C1/,8,MK,/CL,P,02/,8,SA,/CL,N+1,05/,X 2,SW,/CL,CODE,C7/,8,SULC1,15,SULC	
SULC	TABLE /CL,QYMAX+1,0C/,2,RSH,/TM,XYE+1,0C/,12,RSR,/EX,XSULC,0/	
SULC1	TABLE /CL,QYMIN,03/,4,RSY,/EX,XSULC1,0/	
SUMJU	TABLE /CL,P,02/,8,SJMU,/CL,P,01/,8,SY,/CL,P,07/,8,S8,/MV,TEMP,X 00/,15,TEST5	
SUMJUL	TABLE /MV,P,C7/,/MV,PAD+3,31/,15,SU	
SXHBL	TABLE /CL,N+1,04/,8,RSL,/CL,NYMAX+1,01/,12,RSB,/CL,N+1,07/,2,RX SK,15,RSH	
SXME	TABLE /MV,PAD+3,1B/,/MV,P,CO/,/CL,CODE,00/,8,SLBRAC,15,SX	18
SXMSTR	TABLE /MV,P,CC/,15,SXMST1	
SXMST1	TABLE /CL,P,02/,8,SSTAR,/MV,PAD+3,32/,15,SX	32
	EJECT	
SOMQQ8	TABLE /TR,P,CO/,0,PBB,/SS,TEMP,00/,0,PBBB	
PBBB	TABLE 15,OQ	
	TABLE 15,SQ	
	TABLE 15,S8	
	TABLE 15,S0X	
SOX	TABLE /MV,PAD+3,09/,/MV,P,06/,/TM,CODE,C0/,1,S0D,15,S0	
S0D	TABLE /CL,NTCUSP+1,01/,10,SD,15,S0	
S023MB	TABLE /TM,XYE+1,08/,1,S23MB,15,SOMQQ8	
S09	TABLE /TM,XYE+1,0C/,1,S9,15,SOMQQ8	
S09M	TABLE /CL,N+1,05/,2,SOMQQ8,15,S9LC1	
S1MAK	TABLE /TR,P,CO/,0,PCC,/SS,TEMP,00/,0,PCCC	
PCC	DS OH	
	DC X'02'	
	DC X'00'	
	DC X'01'	
	DC X'03'	
	DC 7X'00'	
	DC X'03'	
PCCC	TABLE 15,SET1AK	
	TABLE 15,STPA	
	TABLE 15,S1MAKX	
	TABLE 15,STPK	
S1MAKX	TABLE /MV,PAD+3,19/,/MV,P,09/,15,S1	19
S1RPAR	TABLE /TM,XYE+1,02/,1,SRPAR,15,S1	
S2MRZ	TABLE /CL,P,02/,8,SZ,/CL,P,01/,8,TEST3,/TM,XYS+1,0C/,1,SASTAR,X	

```

/CL, CODE, 38/, 6, N0323, /CL, CCDE+1, AA/, 8, S3MB, /CL, CODE+1, 3AX
/, 8, S3MB, 15, N0323
N0323 TABLE /MV, PAD+3, 08/, 15, S2LC
S2LC TABLE /TM, XYE+1, 08/, 1, S2, /CL, N+1, 04/, 2, RSP, 15, RSF
S2LC1 TABLE /MV, PAD+3, 08/, /TM, XYS+1, 08/, 8, S2MRZ, /CL, CODE, 24/, 8, RSS, 1X
5, RSR
S2LCY TABLE /CL, N+1, 04/, 8, S2, 15, RSY
S2LCZ TABLE /CL, N+1, 04/, 8, S2, 15, RSZ
S23MB TABLE /CL, N+1, 04/, 2, S23MB1, /CL, CODE, 39/, 8, SOMOQ8, 15, S23MB1
S23MB1 TABLE /TR, BR56+1, 00/, 0, BB, /SS, TEMP, 00/, 0, BBB
BB DS OH
DC X'00'
DC 4X'02'
DC X'00'
DC 4X'02'
DC X'02'
DC 4X'02'
DC X'02'
DC 4X'02'
DC X'00'
S23MBP EQU *
TABLE /TM, XYS+1, 0C/, 1, TEST3, /CL, PAD+3, 0E/, 8, S3MB, /CL, N+1, 07/, 2X
, S3MB, /TR, BR56+1, 00/, 0, CC, /SS, TEMP, 00/, 0, BBB
CC DS OH
DC X'00'
DC 3X'02'
DC 10X'00'
DC X'02'
DC X'00'
DC X'01'
BBB TABLE 15, S2MRZ
TABLE 15, SOMOQ8
TABLE 15, S3MB
S24 TABLE /TM, XYS+1, 0C/, 1, S4LC, /MV, PAD+3, 08/, 15, S2
S4LC TABLE /TM, XYS+1, 03/, 12, S4, /MV, PAD+3, 38/, /EX, XS4LC, 0/
S3MB TABLE /CL, P, C1/, 8, TEST3, /MV, PAD+3, 0E/, 15, S3LC1
S3LC TABLE /TM, XYE+1, 08/, 8, RSZ, /TM, XYS+1, 0C/, 1, RSR, 15, S3
S3LC1 TABLE /TM, XYE+1, 08/, 8, RSZ, /TM, XYE+1, 02/, 1, S3, 15, RSA
S3MBR TABLE /CL, PAD+3, 04/, 8, S3MBR1, /TM, CODE, 03/, 1, S3MBR1, /CL, N+1, 04/X
, 8, S2LC1, 15, S3MBR1
S3MBR1 TABLE /CL, P, C1/, 8, BR, /MV, PAD+3, 04/, 15, S3LC
S3SCR8 TABLE /CL, N+1, 04/, 2, SSCRUB, 12, S3MB
S4MK TABLE /MV, P, 00/, 15, S4MK1
S4MK1 TABLE /CL, P, C1/, 8, SK, /MV, PAD+3, 33/, 15, S4MK1X
S4Y TABLE /CL, N+1, 01/, 12, S4Y1, 2, S4MK
*2ND STROKE HAS ONLY 1 ANGLE, 1ST STROKE IS L
*DOES THE 1ST STROKE L HAVE ITS ENDP T IN RIGHT 1/4
S4Y1 TABLE /TM, XYE+1, 03/, 8, S4MK, 15, SY
S6S TABLE /TM, TURN+1, 40/, 1, SSM, 15, STP6
S65 TABLE /TM, TURN, 01/, 1, SSM, 15, STP6
S7MGK TABLE /CL, PAD+3, 18/, 8, XMK1, /TR, P, 00/, 0, PDD, /SS, TEMP, 00/, 0, PDDD
PDD DS OH

```

```

DC      X'02'
DC      X'00'
DC      X'06'
DC      2X'02'
DC      X'05'
DC      X'04'
DC      2X'02'
DC      X'03'
DC      X'02'
DC      X'01'
PCCC    TABLE 15,XMK
        TABLE 15,SK
        TABLE 15,S7X
        TABLE 15,STPK
        TABLE 15,SQ
        TABLE 15,SG
        TABLE 15,SXMSTR
S7X      TABLE /MV,PAD+3,17/,/MV,P,03/,/CL,NCUSP+1,01/,10,S7,/CL,ASPR+1X
        ,08/,2,SRPAR,15,S7
S8LCV    TABLE /MV,PAD+3,39/,/TM,XYE+1,08/,1,S2,15,S8LCVX
S9MG      TABLE /CL,P,02/,8,SG,/MV,PAD+3,34/,15,S9LC
S9LC1     TABLE /TM,XYE+1,0C/,1,SCPEL,/CL,NYMIN+1,01/,12,S9,/TM,XYS+1,0CX
        /,1,RSB,15,RSF
S9LC      TABLE /CL,NYMIN+1,01/,2,S9LC2N,4,S9LCGQ,/CL,QYMIN,03/,8,SCPAD,X
        15,S9LCGQ
S9LC2N    TABLE /CL,QYMIN+1,03/,8,SCPAD,15,S9LCGQ
S9LCGQ    TABLE /TM,XYE+1,0C/,1,S9,15,SCPGQ
S9MK      TABLE /CL,P,01/,8,SK,/MV,PAD+3,35/,15,S9
*SCRIPT LETTERS
SCPAD     TABLE /TM,XYE+1,08/,8,SCPGQ,/CL,QYMAX,00/,8,RSA,/TM,XYS+1,0C/,X
        8,RSA,15,RSD
SCPGQ     TABLE 15,RSG
SCPEL     TABLE /MV,PAD+3,38/,/EX,XSCPEL,0/
SCPMW     TABLE /EX,XSCPMW,0/
SCPPYZ    TABLE /CL,QYMIN+1,03/,8,RSP,15,SCPYZ
SCPYZ     TABLE /EX,XSCPYZ,0/
SCPBH     TABLE /TM,XYE+1,0C/,1,RSH,15,RSB
SCPMWK    TABLE /CL,QYMAX+1,00/,2,RSK,15,SCPMW
SCPNU     TABLE 15,RSN
SCPBS     TABLE /EX,XSCPBS,0/
SCFPF     TABLE /TM,XYS+1,0C/,12,SCFPFY,/MV,PAD+3,39/,/EX,XSBVMN,0/
SCFPFY    TABLE /CL,N+1,05/,12,RSF,/CL,QYMIN+1,03/,8,RSP,15,RSY
SCPNRZ    TABLE /TM,XYS+1,0C/,12,S2LCZ,/CL,N+1,04/,8,RSR,/CL,N+1,07/,2,RX
        SM,/TM,CODE,C0/,4,RSC,15,RSN
SCRPT     TABLE /TM,XYS+1,0C/,12,SCPGQ,/CL,QYMAX,00/,2,RSD,/CL,N+1,05/,1X
        2,RSR,15,RSA
SCPTX     TABLE /CL,P,01/,8,RSX,15,RST
*
*
****SET-UP CHARACTER CODE****
*

```

*
SA TABLE /MV,CHAR,C1/,15,RECD
SB TABLE /MV,CHAR,C2/,15,RECD
SCC TABLE /MV,CHAR,C3/,15,RECD
SD TABLE /MV,CHAR,C4/,15,RECD
SE TABLE /MV,CHAR,C5/,15,RECD
SF TABLE /MV,CHAR,C6/,15,RECD
SG TABLE /MV,CHAR,C7/,15,RECD
SH TABLE /MV,CHAR,C8/,15,RECD
SJ TABLE /MV,CHAR,D1/,15,RECD
SK TABLE /MV,CHAR,D2/,15,RECD
SL TABLE /MV,CHAR,D3/,15,RECD
SM TABLE /MV,CHAR,D4/,15,RECD
SNN TABLE /MV,CHAR,D5/,15,RECD
SO TABLE /MV,CHAR,D6/,15,RECD
SP TABLE /MV,CHAR,D7/,15,RECD
SQ TABLE /MV,CHAR,D8/,15,RECD
SR TABLE /MV,CHAR,D9/,15,RECD
ST TABLE /MV,CHAR,E3/,15,RECD
SU TABLE /MV,CHAR,E4/,15,RECD
SV TABLE /MV,CHAR,E5/,15,RECD
SW TABLE /MV,CHAR,E6/,15,RECD
SX TABLE /MV,CHAR,E7/,15,RECD
SY TABLE /MV,CHAR,E8/,15,RECD
SZ TABLE /MV,CHAR,E9/,15,RECD
S0 TABLE /MV,CHAR,F0/,15,RECD
S1 TABLE /MV,CHAR,F1/,15,POSTST
S2 TABLE /MV,CHAR,F2/,15,RECD
S3 TABLE /MV,CHAR,F3/,15,RECD
S4 TABLE /MV,CHAR,F4/,15,RECD
S5 TABLE /MV,CHAR,F5/,15,RECD
S6 TABLE /MV,CHAR,F6/,15,RECD
S7 TABLE /MV,CHAR,F7/,15,RECD
S8 TABLE /MV,CHAR,F8/,15,RECD
S9 TABLE /MV,CHAR,F9/,15,RECD
SEQL TABLE /MV,CHAR,FE/,15,RECD
SKARAT TABLE /MV,CHAR,70/,15,RECD
SLBRAC TABLE /MV,CHAR,CF/,15,RECD
SLKRT TABLE /MV,CHAR,EE/,15,RECD
SLPAR TABLE /MV,CHAR,CD/,15,POSTST
SMINUS TABLE /MV,CHAR,E0/,15,RECD
SPER TABLE /MV,CHAR,CB/,15,RECD
SPLUS TABLE /MV,CHAR,CE/,15,RECD
SPOUND TABLE /MV,CHAR,FB/,15,RECD
SRBRAC TABLE /MV,CHAR,DF/,15,RECD
SRKRT TABLE /MV,CHAR,CC/,15,RECD
SRPAR TABLE /MV,CHAR,DD/,15,POSTST
SSCRUB TABLE /MV,CHAR,72/,15,RECD
SSLASH TABLE /MV,CHAR,E1/,15,POSTST
SSTAR TABLE /MV,CHAR,DC/,15,RECD
STILDA TABLE /MV,CHAR,DO/,15,RECD

DK TABLE /MV,CHAR,EF/,15,RECD
SBOX TABLE /MV,CHAR,73/,15,RECD
SCIRC TABLE /MV,CHAR,74/,15,RECD
SELIPS TABLE /MV,CHAR,76/,15,RECD
SPBOX TABLE /MV,CHAR,77/,15,RECD
STRAP TABLE /MV,CHAR,78/,15,RECD
STRI TABLE /MV,CHAR,75/,15,RECD
RSA TABLE /MV,CHAR,81/,15,RECD
RSB TABLE /MV,CHAR,82/,15,RECD
RSC TABLE /MV,CHAR,83/,15,RECD
RSD TABLE /MV,CHAR,84/,15,RECD
RSE TABLE /MV,CHAR,85/,15,RECD
RSF TABLE /MV,CHAR,86/,15,RAZER
RSG TABLE /MV,CHAR,87/,15,RAZER
RSH TABLE /MV,CHAR,88/,15,RECD
RSI TABLE /MV,CHAR,89/,15,RECD
RSJ TABLE /MV,CHAR,91/,15,RAZER
RSK TABLE /MV,CHAR,92/,15,RECD
RSL TABLE /MV,CHAR,93/,15,RECD
RSM TABLE /MV,CHAR,94/,15,RECD
RSN TABLE /MV,CHAR,95/,15,RECD
RSO TABLE /MV,CHAR,96/,15,RECD
RSP TABLE /MV,CHAR,97/,15,RAZER
RSQ TABLE /MV,CHAR,98/,15,RAZER
RSR TABLE /MV,CHAR,99/,15,RECD
RSS TABLE /MV,CHAR,A2/,15,RECD
RST TABLE /MV,CHAR,A3/,15,RECD
RSU TABLE /MV,CHAR,A4/,15,RECD
RSV TABLE /MV,CHAR,A5/,15,RECD
RSW TABLE /MV,CHAR,A6/,15,RECD
RSX TABLE /MV,CHAR,A7/,15,RECD
RSY TABLE /MV,CHAR,A8/,15,RAZER
RSZ TABLE /MV,CHAR,A9/,15,RAZER

*

*

****PAD TABLE****

*

*

PADT	EQU	*	
	TABLE	15,DK	
	TABLE	15,SDP	
	TABLE	15,STP6	02
	TABLE	15,STP5	
	TABLE	15,S3MBR1	04
	TABLE	15,SCMG	05
	TABLE	15,KVXYM	06
	TABLE	15,SETXX	07
	TABLE	15,SEGC6M	08
	TABLE	15,SOMQ8	09
	TABLE	15,G6ETST	0A
	TABLE	15,S2MRZ	0B

TABLE 15,SPRMA	0C
TABLE 15,SBARMK	0D
TABLE 15,S3MB	0E
TABLE 15,SUMAM	0F
TABLE 15,SRPRM	10
TABLE 15,SRPRM	
TABLE 15,SSM	12
TABLE 15,SSM	
TABLE 15,SCOMAM	14
TABLE 15,SAMST1	15
TABLE 15,SCMEG	16
TABLE 15,S7MGK	17
TABLE 15,XMK1	18
TABLE 15,S1MAK	19
TABLE 15,SLMEK4	1A
TABLE 15,STPE	1B
TABLE 15,SVM	1C
TABLE 15,SVM	
TABLE 15,TEST5	1E
TABLE 15,SFE	1F
TABLE 15,SFME1	20
TABLE 15,OQ	21
TABLE 15,SCMH	22
TABLE 15,FIME1	23
TABLE 15,KNYM1	
TABLE 15,AHSTR	25
TABLE 15,SBM51	26
TABLE 15,SUMXX	27
TABLE 15,SK5	28
TABLE 15,SNMA	29
TABLE 15,SRPRMJ	2A
TABLE 15,STPA	2B
TABLE 15,SE	2C
TABLE 15,STPH8	2D
TABLE 15,STPK1	2E
TABLE 15,SJ	2F
TABLE 15,STPM	30
TABLE 15,SUMJU	31
TABLE 15,SXMST1	32
TABLE 15,S4MK1	33
TABLE 15,S9MG	34
TABLE 15,S9MK	35
TABLE 15,TPLUSM	36
TABLE 15,SA7	
TABLE 15,SCPTX	38
TABLE 15,RSX	39
TABLE 15,SPOUND	3A

*

*

****4 DIRECTION TABLE****

*

*

AAAA

AAA

TABLE /SS,TEMP,00/,0,AAA

EQU *

TABLE 15,RECD

TABLE 15,STPM

TABLE 15,S23MB

TABLE 15,S23MBP

TABLE 15,SMC

TABLE 15,SS8M

TABLE 15,SG069M

TABLE 15,SS589M

TABLE 15,SG06M

TABLE 15,SB DPR

TABLE 15,SMNW

TABLE 15,SM1M

TABLE 15,SBARM

TABLE 15,S2MRZ

TABLE 15,S3MB

TABLE 15,S3MBR

TABLE 15,STP6

TABLE 15,S24

TABLE 15,SRPRM

TABLE 15,DK

TABLE 15,S7MGK

TABLE 15,STPA

TABLE 15,S1MAK

TABLE 15,SNMA

TABLE 15,SMLC

TABLE 15,SCOMAM

TABLE 15,SBARMK

TABLE 15,SSM

TABLE 15,S9MK

TABLE 15,SCMEG

TABLE 15,SFE

TABLE 15,SLMEK4

TABLE 15,SUMJU

TABLE 15,S5

TABLE 15,STP5

TABLE 15,SK5

TABLE 15,STPH8

TABLE 15,SVM

TABLE 15,SDMH

TABLE 15,SUMAM

TABLE 15,STPJ

TABLE 15,SGSCR8

TABLE 15,S3SCR8

TABLE 15,BR

TABLE 15,SB DPR1

TABLE 15,SLKRTM

TABLE 15,SRPRMJ

TABLE 15,SGS

2E

TABLE 15,SGSC6M	
TABLE 15,SOMQ8	
TABLE 15,S8	32
TABLE 15,SG	33
TABLE 15,S9	34
TABLE 15,S3	35
TABLE 15,SASTAR	36
TABLE 15,SCC	37
TABLE 15,SA7	38
TABLE 15,SCG	39
TABLE 15,SG81	3A
TABLE 15,S023MB	3B
TABLE 15,S09	3C
TABLE 15,S9LC1	
TABLE 15,S09M	
TABLE 15,SCRPT	
TABLE 15,RSC	
TABLE 15,SCPFP	
TABLE 15,SCPEL	
TABLE 15,RSS	
TABLE 15,RSV	
TABLE 15,SCPNRZ	
TABLE 15,RSZ	
TABLE 15,SE	
TABLE 15,SCPGQ	
TABLE 15,S8LCV	49
TABLE 15,SEQ	4A

*

*

*****SET OF ESCAPES*****

*

*

AHSTRX	TABLE /EX,XAHSTR,0/
KNY	TABLE /EX,XKNY,0/
FI	TABLE /EX,XFI,0/
KVXY	TABLE /EX,XKVXY,0/
MW	TABLE /EX,XMW,0/
MWIN	TABLE /EX,XMWIN,0/
MW1	TABLE /EX,XMW1,0/
POSTST	TABLE /EX,XPOST,0/
RAZER	TABLE /EX,XRAZE,0/
RECD	TABLE /EX,XRECD,0/
SDP	TABLE /EX,XSDP,0/
SMNW	TABLE /EX,XSMNW,0/
SM1M	TABLE /EX,XSM1M,0/
SRPRM	TABLE /EX,XSRPRM,0/
SSM	TABLE /EX,XSSM,0/
SVM	TABLE /EX,XSVM,0/
TEST1	TABLE /EX,XTEST1,0/
TEST3	TABLE /EX,XTEST3,0/
TPLUS	TABLE /EX,XTPLUS,0/

S8LCVX TABLE /EX,XS8LCV,0/
SRLCX TABLE /EX,XSRLC,0/
S4MK1X TABLE /EX,XS4MK1,0/

*

*

****ENTRY SWITCH****

*

*

GPSW TABLE /SS,CUSP,00/,0,GPSWTCH
GPSWTCH EQU *

TABLE 15,SPER
TABLE 15,SXHBL
TABLE 15,S4
TABLE 15,SK
TABLE 15,AAAA
TABLE 15,KVXYM
TABLE 15,PADEX
TABLE 15,TPLUSM
TABLE 15,SBM5
TABLE 15,SJMU
TABLE 15,SUMJU1
TABLE 15,SM1
TABLE 15,SXMSTR
TABLE 15,SCMCQ8
TABLE 15,SCPNU
TABLE 15,RSB
TABLE 15,RSC
TABLE 15,RSE
TABLE 15,RSF
TABLE 15,RSR
TABLE 15,RSI
TABLE 15,RSJ
TABLE 15,RSL
TABLE 15,RSM
TABLE 15,RSN
TABLE 15,RSO
TABLE 15,RSR
TABLE 15,RSS
TABLE 15,RSU
TABLE 15,RSV
TABLE 15,RSW
TABLE 15,RSY
TABLE 15,RSZ
TABLE 15,SA
TABLE 15,SG
TABLE 15,SM
TABLE 15,SNN
TABLE 15,SP
TABLE 15,SR
TABLE 15,SU
TABLE 15,SO

TABLE 15,S8
TABLE 15,SSTAR
TABLE 15,SSCRUB
TABLE 15,DK
TABLE 15,RSA
TABLE 15,SE

*
ENDMARK EQU *
END

REC RCS'S

AHSTR1

*FUNCTION
*
*DISTINGUISHES AMONG 3-STROKE A, H, K, AND * BASED ON POSITIONS OF
*STARTING AND ENDING POINTS
*
*
*
*CALL
* RCS AHSTR1A,ECHAR
*EXIT CHAR WHEN A CHARACTER IS RECOGNIZED
*
*
*
*INPUT REGISTER. R6
*
*INTERNAL REGISTERS. R7, R9-R14
*
*
USING XR6,R6
EX0 EQU 0
REGS
D6 DSECT
XR6 DS OF
XYEP EQU X'40'
XYSP EQU X'4A'
XSPI EQU X'72'
YSPI EQU X'7C'

```

XEPI      EQU      X'86'
YEPI      EQU      X'90'
          DS        3F
          DS        26H
          DS        20C
          DS        3F
          DS        1H
          DS        2C
          DS        3H
P          DS        1C
CHAR      DS        1C
AHSTR1    BOX
*RI0 IS FIRST VERTICAL STROKE REF
*RI1 IS SECOND VERTICAL STROKE REF
*RI2 IS HORIZONTAL STROKE REF
*IS THIRD STROKE HORIZONTAL?
          CLI      P,X'02'
          BC       8,H3
*NO, IS FIRST HORIZONTAL?
          LA       R11,4(R6)
          LH       R7,XEPI(R6)
          SH       R7,XSPI(R6)
          LPR      R7,R7
          LH       R9,YEPI(R6)
          SH       R9,YSPI(R6)
          LPR      R9,R9
          CR       R7,R9
          BC       2,H1
*NO, SECOND STROKE IS THE HORIZONTAL
          LA       R12,2(R6)
          LA       R10,0(R6)
          B        HDONE
*THIRD STROKE IS THE HORIZ
H3        LA       R12,4(R6)
          LA       R11,2(R6)
          LA       R10,0(R6)
          B        HDONE
*FIRST STROKE IS THE HORIZ
H1        LA       R12,0(R6)
          LA       R10,2(R6)
HDONE     EQU      *
*TEST FOR K
*ARE BOTH VERT ENDPTS AT THE LEFT
          TM       XYEP+1(R10),X'03'
          BC       12,NOTK
          TM       XYEP+1(R11),X'03'
          BC       1,SKX
          TM       XYSP+1(R10),X'03'
          BC       12,NOTK
*IS HORIZ START OR END POINT IN UPPER RIGHT?
          CLI      XYEP+1(R12),X'00'

```

```

BC      8,ETOP
CLI     XYEP+1(R12),X'01'
BC      8,ETOP
CLI     XYSP+1(R12),X'00'
BC      8,STOP
CLI     XYSP+1(R12),X'01'
BC      6,NOTK
*IS TOP,RIGHT PART OF HORIZ ABOVE TOP OF SECOND VERT?
STOP    EQU      *
        LH      R13,YSPI(R12)
        B       ETOP1
ETOP     EQU      *
        LH      R13,YEPI(R12)
ETOP1    EQU      *
        CH      R13,YSPI(R11)
        BC      2,SKX
NOTK     EQU      *
*NOT K, TEST FOR A,H, OR *
*ARE START PTS CLOSE COMPARED TO ENDPPTS
        LH      R13,XEPI(R10)
        SH      R13,XEPI(R11)
        LPR     R13,R13
        SRL     R13,2
                                1/4 MAG OF ENDPD DIFF
        LH      R14,XSPI(R10)
        SH      R14,XSPI(R11)
        LPR     R14,R14
        CR      R14,R13
        BC      4,SAX
*NO, DO VERTICAL STROKES CROSS?
        LH      R13,XEPI(R10)
        CH      R13,XEPI(R11)
        BC      2,END1R
        LH      R13,XSPI(R10)
        CH      R13,XSPI(R11)
        BC      2,SSTARX
        B       SHX
END1R    EQU      *
        LH      R13,XSPI(R10)
        CH      R13,XSPI(R11)
        BC      4,SSTARX
        B       SHX
SAX      EQU      *
        MVI     CHAR,C'A'
        BC      15,BEXIT1
SHX      EQU      *
        MVI     CHAR,C'H'
        BC      15,BEXIT1
SKX      EQU      *
        MVI     CHAR,C'K'
        BC      15,BEXIT1
SSTARX   EQU      *

```

```

BEXIT1  MVI  CHAR,X'DC'
        MVI  P,X'CC'
        BEXIT EXO
        END

```

BFI

*FUNCTION

*
 *DISTINGUISHES AMONG 3-STROKE F, I, AND * BASED ON POSITIONS OF START-
 *ING POINTS

*

*

*

*CALL

* RCS BFIA,ECHAR

*EXIT CHAR WHEN A CHARACTER IS RECOGNIZED

*

*

*

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R7, R12, R13, R15

*

*

USING XR6,R6

REGS

EXO EQU 0

D6 DSECT

XR6 DS 0F

XYSP EQU X'4A' XYSP (R6)

DS 3F

DS 26H

DS 20C

DS 3F

DS 1H

DS 2C

DS 3H

DS 1C

CHAR DS 1C

BFI BCX

FI SR R15,R15

LA R12,2

LA R13,4

FI1 LA R7,0(R6,R15)

TM XYSP+1(R7),X'03'

```

MVI  CHAR,X'DC'          ASTERISK
BC    8,FIX
MVI  CHAR,C'I'
TM    XYSP+1(R7),X'CC'
BC    1,FIX
BXLE  R15,R12,FI1
MVI  CHAR,C'F'
FIX   BEXIT EX0
      END

```

BHITE

*FUNCTION

*
 *DISTINGUISHES BETWEEN TALL AND SHORT CHARACTERS. A SHORT CHARACTER
 *IS ONE SHORTER THAN 3/4 OF THE NORMALLY EXPECTED CHARACTER HEIGHT
 *('CHAREC' SETS DYM = 3/2 NORMAL CHARACTER HEIGHT).

*
 *
 *
 *CALL
 * RCS BHITEA,ESHORT,ETALL
 *EXIT SHORT WHEN THE CHARACTER IS SHORT
 *EXIT TALL WHEN THE CHARACTER IS TALL

*
 *
 *
 *INPUT REGISTER. R6
 *
 *INTERNAL REGISTERS. R7, R8
 *
 *

```

      USING XR6,R6
      REGS
EX0    EQU    0
EX4    EQU    4
D6     DSECT
XR6    DS     0F
      DS     3F
      DS     20H
YTC    DS     1H
YBC    DS     1H
      DS     4H
      DS     20C
      DS     3F
      DS     1H

```

	DS	2C	
	DS	3H	
	DS	56C	
	DS	14H	
DYM	DS	1H	
BHITE	BOX		
	LH	R7,YTC	
	SH	R7,YBC	
	LH	R8,DYM	3/2 NORM CHAR HITE
	SRL	R8,1	3/4 NORM CHAR HITE
	CR	R7,R8	
	BC	4,LOX	
HIX	BEXIT	EX4	
LCX	BEXIT	EX0	
	END		

BSDP

*FUNCTION

*
 *DISTINGUISHES AMONG 'D, P, 5, AND SCRIPT B BASED ON THE POSITION
 *OF THE LAST STROKE ENDPOINT, THE POSITION OF THE 2ND REL. Y MAX. IN
 *THIS STROKE, AND THE NO. OF STROKES

*
 *
 *

*CALL

* RCS BSDPA,ECHAR
 *EXIT CHAR WHEN A CHARACTER IS RECOGNIZED

*
 *
 *

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R7, R15

*
 *

	USING	XR6,R6	
	REGS		
EX0	EQU	0	
D6	DSECT		
XR6	DS	0F	
XYEP	EQU	X'40'	XYE (R6)
	DS	1F	
PAD	DS	1F	
	DS	1F	

SN	DS	11H	
	DS	1H	
	DS	14H	
	DS	20C	
	DS	3F	
	DS	1H	
	DS	2C	
	DS	3H	
P	DS	1C	
CHAR	DS	1C	
	DS	54C	
	DS	21H	
	DS	1F	
	DS	6C	
	DS	1H	
	DS	1F	
	DS	20C	
	DS	33H	
QYMAX	DS	10C	
BSDP	BOX		
	CLI	PAD+3,X'01'	
	BC	8,SDM51	
	MVI	PAD+3,X'01'	
SDP	LH	R15,SN	
	BCT	R15,SDP1	
SDP1	SLL	R15,1	
	LA	R7,0(R6,R15)	
	TM	XYEP+1(R7),X'0C'	
	MVI	CHAR,C'P'	
	BC	12,SDPX	
SDM5	MVI	P,X'00'	
	MVI	CHAR,C'D'	
	CLI	SN+1,X'02'	
	BC	8,SDSD	
	CLI	QYMAX+1,X'00'	
	BC	8,SDSD	
	MVI	CHAR,X'82'	LC B
	B	SDPX	
SDM51	CLI	P,X'02'	
	MVI	CHAR,C'5'	
	BC	8,SDPX	
	MVI	CHAR,C'D'	
SDSD	EQU	*	
SDPX	BEXIT	EXO	
	END		

BSMNW

*FUNCTION

*
*DISTINGUISHES AMONG SCRUB, N, W, SCRIPT Y, AND A CHARACTER GROUP (M,
*SCRIPT M, SCRIPT W, SCRIPT Y) BASED ON NO. OF DIRECTIONS, ASPECT RATIO
*AND THE POSITION OF THE FIRST REL. Y MIN.
*
*
*

*CALL

* RCS BSMNWA,ECHAR,EGROUP,ERAZE
*EXIT CHAR WHEN A CHARACTER IS RECOGNIZED
*EXIT GROUP WHEN THE CHARACTER IS M, OR SCRIPT M, W, OR Y
*EXIT RAZE WHEN THE CHAR. IS RECOGNIZED AS A SCRIPT Y, AND THE CHAR
* CENTER MUST BE RAISED
*
*
*

*INPUT REGISTER. R6

*INTERNAL REGISTERS. R7,R8
*
*

USING XR6,R6
REGS

EX0	EQU	0
EX4	EQU	4
EX8	EQU	8
D6	DSECT	
XR6	DS	0F
	DS	3F
	DS	10H
N	DS	1H
	DS	5H
CXC	DS	1H
	DS	5H
ASPR	DS	1H
	DS	3H
	DS	20C
	DS	3F
	DS	1H
	DS	2C
	DS	3H
	DS	1C
CHAR	DS	1C
	DS	6C
XSP	DS	10C
	DS	38C
	DS	21H
	DS	1F

	DS	6C	
	DS	1H	
	DS	1F	
	DS	20C	
	DS	33H	
QYMAX	DS	10C	
QYMIN	EQU	QYMAX+5	
	DS	2H	
YMAXX	DS	10H	
BSMNW	BOX		
SMNW	CLI	N+1,X'05'	
	BC	4,TEST4	
	BC	8,SMLCX	
	MVI	CHAR,X'72'	SCRUB
	B	SMNWX	
TEST4	CLI	ASPR+1,X'04'	
	MVI	CHAR,C'W'	
	BC	4,SMNWX	
*N IF A	SP	RATIO GTR THAN 2	
	CLI	ASPR+1,X'08'	
	BC	2,SNLCY	
*ARE THE SP AND 2ND MAX CLOSER THAN			
*3/8 CHARACTER WIDTH			
	LH	R7,DXC	
	SRL	R7,2 1/4 DELTA X	
	LR	R8,R7	
	SRL	R8,1	
	AR	R7,R8	3/8 DELTA X
	LH	R8,YMAXX+2	
	SH	R8,XSP	
	LPR	R8,R8	
	CR	R8,R7	
	BC	4,SNLCY	
	MVI	CHAR,C'W'	NO
	B	SMNWX	
SNLCY	MVI	CHAR,C'N'	
	CLI	QYMIN,X'03'	
	BC	8,SMNWX	
	MVI	CHAR,X'A8'	LC Y
	BEXIT	EX8	
SMNWX	BEXIT	EX0	
SMLCX	BEXIT	EX4	
	END		

BSRPRM

*FUNCTION

*
 *DISTINGUISHES AMONG R, 3, 5, RIGHT BRACKET, AND 2 GROUPS OF CHARACTERS
 *(D, P), (RIGHT PAREN., COMMA, APOSTROPHE) BASED ON THE IDENTITY OF THE
 *THE PREVIOUS SUBCHARACTER, THE NO. OF GEOM. CORNERS, AND THE POSITION
 *OF A CORNER

*
 *
 *
 *CALL
 * RCS BSRPRMA,ECHAR,EDP,EPAREN
 *EXIT CHAR WHEN A CHARACTER IS RECOGNIZED
 *EXIT DP WHEN THE CHARACTER IS A D OR P
 *EXIT PAREN WHEN THE CHARACTER IS A RIGHT PAREN., COMMA, OR APOSTROPHE,
 * TEST SIZE AND POSITION

*
 *
 *
 *INPUT REGISTER. R6

*
 *INTERNAL REGISTERS. R7, R8

*
 USING XR6,R6
 EX0 EQU 0
 EX4 EQU 4
 EX8 EQU 8
 REGS
 D6 CSECT
 XR6 DS 0F
 DS 1F
 PAD DS 1F
 DS 1F
 DS 20H
 YTC DS 1H
 YBC DS 1H
 DS 4H
 DS 20C
 DS 3F
 DS 1H
 DS 2C
 NCUSP DS 1H
 DS 2H
 P DS 1C
 CHAR DS 1C
 DS 54C
 DS 21H
 DS 1F
 DS 6C
 DS 1H
 DS 1F

	CS	10C	
YC	CS	10C	
BSRPRM	BCX		
	CLI	PAD+3,X'11'	
	BC	8,DPMR1	
SRPRM	CLI	P,X'01'	
	BC	8,DPMR	
	CLI	P,X'02'	
	MVI	CHAR,C'5'	
	BC	8,SRPRMX	
	CLI	P,X'0B'	
	MVI	CHAR,C'R'	
	BC	8,SRPRMX	
	MVI	PAD+3,X'10'	SRPRM
	CLI	NCUSP+1,X'03'	
	MVI	CHAR,C'3'	
	BC	10,SRPRMX	
	CLI	NCUSP+1,X'01'	
RBRAKX	MVI	CHAR,X'DF'	RIGHT BRACKET
	BC	2,SRPRMX	
	MVI	CHAR,X'DD'	RIGHT PARENTHESIS
	BC	4,SRPMX1	
*3 IF T	HE	CUSP IS IN THE MIDDLE	
	LH	R7,YTC	
	SH	R7,YBC	
	SRL	R7,1 1/2 DELTA Y	
	LR	R8,R7	
	SRL	R8,1 1/4 DELTA Y	
	AH	R8,YBC	
	AR	R7,R8	
	CH	R8,YC	
	BC	2,RBRAKX	
	CH	R7,YC	
	MVI	CHAR,C'3'	MIDDLE
	BC	10,SRPRMX	
	BC	4,RBRAKX	
DPMR	MVI	P,X'00'	
DPMR1	CLI	P,X'01'	
	MVI	CHAR,C'R'	
	BC	8,SRPRMX	
	MVI	PAD+3,X'11'	DPMR1
	BEXIT	EX4	
SRPRMX	BEXIT	EX0	
SRPMX1	BEXIT	EX8	
	END		

BSSM

*FUNCTION

*

*DISTINGUISHES AMONG S, 5, 8, 9, AND \$ BASED ON THE GENERAL IDENTITY OF
*THE PREVIOUS SUBCHARACTER, THE POSITION OF THE ENDPOINT, THE NO. OF
*DIRECTIONS, THE FIRST DIRECTION, AND THE NO. OF TIME-CORNERS

*

*

*

*CALL

* RCS BSSMA,ECHAR

*EXIT CHAR WHEN A CHARACTER IS RECOGNIZED

*

*

*

*INPUT REGISTER. R6

*

*INTERNAL REGISTER. R7

*

*

	USING XR6,R6
	REGS
EX0	EQU 0
D6	DSECT
XR6	DS 0F
	DS 1F
PAD	DS 1F
CCODE	DS 1F
	DS 10H
N	DS 1H
	DS 7H
XRC	DS 1H
XLC	DS 1H
	DS 6H
XYE	DS 10C
	DS 10C
	DS 3F
	DS 1H
	DS 2C
NCUSP	DS 1H
	DS 2H
P	DS 1C
CHAR	DS 1C
TEMP	DS 1C
	DS 53C
	DS 21H
	DS 1F
	DS 6C
	DS 1H
	DS 1F

```

XC      DS      10C
        DS      10C
        DS      17H
NTCUSP  DS      1H
BSSM    BOX
        CLI     PAD+3,X'13'
        BC      8,DOLDOL
SSM      MVC     TEMP(1),P
        TR      TEMP(1),LL
        SR      R7,R7
        IC      R7,TEMP
        EX      0,LLL(R7)
SSM1     MVI     P,X'05'
        MVI     PAD+3,X'12'
        TM      XYE+1,X'08'
        BC      8,S8S8
*NOT 5   IF 1ST ANGLE IS 1
        TM      CODE,X'80'
        BC      1,SSM2
        TM      CODE,X'40'
        BC      1,S9
*TEST FOR TIME CORNERS
SSM2     EQU     *
        CLI     NTCUSP+1,X'01'
        BC      2,S5S5
        BC      4,S5S5
*1 TIME CORNER, CHECK FOR GEOM CORNERS
        CLI     NCUSP+1,X'02'
        BC      2,S5S5
        BC      4,S5S5
        LH      R7,XRC
        SH      R7,XLC
        SRL     R7,1  1/2 DELTA X
        AH      R7,XLC
        CH      R7,XC+2
        BC      2,S5S5
        BC      12,S5S5
LL       DS      OH
        DC      X'00'
        DC      X'04'
        DC      X'10'
        DC      2X'00'
        DC      3X'08'
        DC      3X'0C'
        DC      X'0C'
LLL      DS      OH
        BC      15,SSM1
        BC      15,STPCOL
        BC      15,S8S8
        BC      15,DOLDOL
        BC      15,S5S5

```

SSM

2
NONE

LEFT
RIGHT

*POSSIBLE 9, TEST ANGLES, AND TIME CORNERS

```

S9      CLI    N+1,X'06'
        BC     4,SSSS
        CLI    NTCUSP+1,X'01'
        BC     4,SSSS
S9S9    MVI    CHAR,C'9'
        B      SSMX
STPDOL  MVI    P,X'00'
        MVI    PAD+3,X'13'
DCLDOL  MVI    CHAR,X'DB'          DCLDOL
        BC     15,SSMX          DCLLARS
SSSS    MVI    CHAR,C'5'
        BC     15,SSMX
SSSS    MVI    CHAR,C'S'
        BC     15,SSMX
S8S8    MVI    CHAR,C'8'
        BC     15,SSMX
SSMX    BEXIT  EXO
        END

```

BSVM

*FUNCTION

*
 *DISTINGUISHES AMONG V, W, AND 5 GROUPS OF CHARACTERS (J, U), (M, W),
 *(K, N, Y), (O, 8, Q, Q), (U, 8) BASED ON THE GENERAL IDENTITY OF THE
 *PREVIOUS SUBCHARACTER, THE ORIGIN OF THE CALL TO THIS ROUTINE (TEMP
 *HAS BEEN ENCODED AS C IN 'INTERP' IF THE CHAR CAN BE U), AND THE DIS-
 *TANCE BETWEEN THE STARTING AND ENDING POINTS

*
 *
 *

*CALL

* RCS BSVMA,ECHAR,EJU,EMW,EKNY,E080Q,EU8

*EXIT CHAR WHEN A CHARACTER IS RECOGNIZED

*EXIT JU WHEN THE CHAR IS J, POTENTIALLY U

*EXIT MW WHEN THE CHAR IS M OR W

*EXIT KNY WHEN THE CHAR IS K, N, OR Y

*EXIT 080Q WHEN THE CHAR IS O, POTENTIALLY 8, O, OR Q

*EXIT U8 WHEN THE CHAR IS U, POTENTIALLY 8

*
 *
 *

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R7,R8,R10

```

*
*
      USING XR6,R6
EX0   EQU    0
      REGS
EX4   EQU    4
EX8   EQU    8
EX12  EQU    12
EX16  EQU    16
EX20  EQU    20
C6    DSECT
XR6   DS     0F
      DS     1F
PAD   DS     1F
      DS     1F
      DS     18H
XRC   DS     1H
XLC   DS     1H
      DS     6H
      DS     20C
      DS     3F
      DS     1H
      DS     2C
      DS     3H
P     DS     1C
CHAR  DS     1C
TEMP  DS     1C
      DS     5C
XSP   DS     10C
      DS     10C
XEP   DS     10C
BSVM  BOX
      CLI    PAD+3,X'1D'
      BC     8,KNYXXX
      CLI    PAD+3,X'1E'
      BC     8,TEST5
SVM   CLI    P,X'01'
      BC     8,KNYMMW
      CLI    P,X'02'
      BC     8,SJMUXX
      CLI    P,X'0B'
      BC     8,MW1X
      CLI    P,X'04'
      MVI    CHAR,C'W'
      MVI    TEMP,X'C4'
      BC     15,TEST5
      BC     8,SVMX
SVM1  MVI    PAD+3,X'1C'
      MVI    P,X'C4'
      MVI    CHAR,C'V'
      BC     15,SVMX
      SVM

```

```

KNYMMW   MVI    P,X'CC'
KNYXXX   CLI    P,X'01'
          BC     8,MW1X
          MVI    PAD+3,X'1D'          KNYXXX
          BC     15,KNY1X
**0 VS   U,V TEST
*0 IF S   TARTPT AND ENDPT ARE CLOSER THAN
*1/2 CH   ARACTER WIDTH
*TEMP CO  NTAINS CODE FOR RETURN TO U OR V
TEST5    LH     R7,XSP
          SH     R7,XEP
          LPR    R7,R7
          LH     R8,XRC
          SH     R8,XLC
          LPR    R8,R8
          SRL    R8,1
          CR     R7,R8
          BC     4,SOMX
*ENDPT    IN LEFT OR RIGHT 1/4
          SR     R10,R10
          IC     R10,TEMP
          EX     0,T5SW(R10)
T5SW     DS     OF
          BC     15,SUJU1X
          BC     15,SVM1
SVMX     BEXIT  EX0
SJMXXX   BEXIT  EX4
MW1X     BEXIT  EX8
KNY1X    BEXIT  EX12
SOMX     BEXIT  EX16
SUJU1X   BEXIT  EX20
          END

```

BTEST1

*FUNCTION

*
 *DISTINGUISHES BETWEEN TWO CHARACTER GROUPS (8, SCRIPT G), (S-LIKE
 *CHARACTERS) BASED ON THE POSITION OF THE ENDPOINT

*

*

*CALL

* RCS BTEST1A,E8G,ESSM

*EXIT 8G WHEN CHAR IS 8 OR SCRIPT G, TEST DIRECTIONS

*EXIT SSM WHEN CHAR IS S-LIKE, TEST FURTHER WITH BSSM

*

```
*
*INPUT REGISTER.  R6
*
*INTERNAL REGISTERS.  R7, R15
*
*
```

```
        USING XR6,R6
        REGS
EX0      EQU    0
EX4      EQU    4
D6       DSECT
XR6      DS     0F
XYEP     EQU    X'40'           XYE
        DS     3F
        DS     11H
SN       DS     1H
        DS     14H
        DS     20C
        DS     3F
        DS     1H
        DS     2C
        DS     3H
        DS     1C
CHAR     DS     1C
BTEST1   BCX
TEST1    LH     R15,SN
        BCT     R15,TEST11
TEST11   SLL    R15,1
        LA     R7,0(R6,R15)
        TM     XYEP+1(R7),X'C8'
        BC     1,SSMXXX
        MVI    CHAR,C'8'
        BEXIT  EX0
SSMXXX   BEXIT  EX4
        END
```

BTEST3

```
*FUNCTION
```

```
*
*DISTINGUISHES AMONG B, R, U, SCRIPT K, SCRIPT X, AND A CHARACTER GROUP
*(5, B) BASED ON THE NO. OF STROKES, THE POSITIONS OF STARTING AND END-
*ING POINTS, THE DIRECTIONS, THE POSITIONS OF REL. Y MAXIMA
*
```

```
*
*
*CALL
```

* RCS BTEST3A,ECHAR,E5B
 *EXIT CHAR WHEN A CHARACTER IS RECOGNIZED
 *EXIT 5B WHEN CHAR IS 5, POTENTIALLY B

*
 *
 *

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R7, R8

*

*

USING XR6,R6

REGS

EX0	EQU	0	
EX4	EQU	4	
D6	DSECT		
XR6	DS	0F	
XYSP	EQU	X'4A'	XYSP
XYEP	EQU	X'40'	XYE
	DS	2F	
CODE	DS	1F	
	DS	11H	
SN	DS	1H	
	DS	4H	
DXC	DS	1H	
	DS	1H	
XRC	DS	1H	
	DS	7H	
XYE	DS	10C	
	DS	10C	
	DS	3F	
	DS	1H	
	DS	2C	
	DS	3H	
	DS	1C	
CHAR	DS	1C	
	DS	54C	
	DS	21H	
	DS	1F	
	DS	6C	
	DS	1H	
	DS	1F	
	DS	20C	
	DS	33H	
QYMAX	DS	10C	
	DS	2H	
YMAXX	DS	10H	
BTEST3	BOX		
TEST3	CLI	SN+1,X'01'	
	BC	8,TEST31	

*2 STROKE CHARACTERS

```

TEST32    LH      R8,SN
          BCT     R8,TEST32
          SLL     R8,1
          LA      R8,0(R8,R6)
          MVI     CHAR,X'A7'          LC X
          TM      XYSP+1(R8),X'02'
          BC      8,TEST3X
          TM      XYEP+1(R8),X'02'
          MVI     CHAR,C'B'
          BC      1,TEST3X          END IN LEFT HALF
          MVI     CHAR,C'R'
          BC      8,TEST3X          END IN RIGHT HALF
*SINGLE STROKE CHARACTERS
TEST31    TM      XYE+1,X'02'
          BC      1,SBM5X
          MVI     CHAR,C'R'          END IN RIGHT HALF
RLC       EQU     *
          CLI     CODE,X'DC'        3130
          BC      8,RU
          CLI     QYMAX+1,X'00'
          BC      8,TEST3X
          MVI     CHAR,X'92'        K
          B       TEST3X
RU        EQU     *
          LH      R8,DXC
          SRL     R8,1
          LH      R7,XRC
          SR      R7,R8
* IS MAX 2 IN RIGHT 1/2
          CH      R7,YMAXX+2
          BC      2,TEST3X          NO,R
          MVI     CHAR,C'U'
TEST3X    BEXIT EXC
SBM5X     BEXIT EX4
          END

```

KNYTST

*FUNCTION

*

*DISTINGUISHES AMONG 3-STROKE (ALL VERT) K, N, AND Y BASED ON THE POS-
*ITIONS OF THE STARTING AND ENDING POINTS

*

*

*

*CALL

* RCS KNYTSTA,ECHAR
 *EXIT CHAR WHEN A CHARACTER IS RECOGNIZED

*
 *
 *

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R7-R13, R15

*

*

	USING	XR6,R6
EX0	EQU	0
	REGS	
D6	DSECT	
XR6	DS	0F
XYSP	EQU	X'4A'
XYEP	EQU	X'40'
	DS	3F
	DS	26H
XYE	DS	10C
XYS	DS	10C
	DS	3F
	DS	1H
	DS	2C
	DS	3H
	DS	1C
CHAR	DS	1C
KNYTST	BOX	
	SR	R9,R9
	SR	R10,R10
	SR	R11,R11
	LA	R12,2
	LA	R13,4
	SR	R15,R15
KNYIN	LA	R8,0(R6,R15)
	TR	XYSP+1(1,R8),HHS
	LH	R7,XYS(R15)
	EX	0,HHHS(R7)
KNYSN	CR	R9,R12
	BC	8,SNX
	LR	R9,R12
KNYSKY	TR	XYEP+1(1,R8),HHE
	LH	R7,XYE(R15)
	EX	0,HHHE(R7)
KNYEY	CR	R10,R12
	BC	8,SYX
	LR	R10,R12
	BC	15,KNYI
KNYEN	CR	R11,R12
	BC	8,SNX
	LR	R11,R12

```

KNYI    BXLE  R15,R12,KNYIN
SKXX    EQU   *
        MVI   CHAR,C'K'
        BC    15,BEXIT2
SNX     EQU   *
        MVI   CHAR,C'N'
        BC    15,BEXIT2
SYX     EQU   *
        MVI   CHAR,C'Y'
BEXIT2  BEXIT  EX0
HHHS    DS    0F
        BC    15,KNYSKY
        BC    15,KNYSN
HHHE    DS    0F
        BC    15,KNYI
        BC    15,KNYEY
        BC    15,KNYEN
HHS     DS    0H
        DC    2X'00'
        DC    2X'04'
        DC    3X'00'
        DC    X'04'
        DC    8X'00'
HHE     DS    0H
        DC    5X'00'
        DC    2X'04'
        DC    X'00'
        DC    X'08'
        DC    2X'04'
        DC    X'00'
        DC    2X'08'
        DC    2X'00'
END

```

KNYIT

*FUNCTION

*
 *DISTINGUISHES AMONG 2-STROKE (1 VERT, 1 V-LIKE) K, N, AND Y BASED ON
 *THE POSITIONS OF THE STARTING AND ENDING POINTS

*
 *
 *

*CALL

* RCS KNYITA,ECHAR
 *EXIT CHAR WHEN A CHARACTER IS RECOGNIZED

```

*
*
*
*INPUT REGISTER.  R6
*
*INTERNAL REGISTERS.  R7, R8, R12, R13, R15
*
*
      USING XR6,R6
EXO      EQU      0
      REGS
D6        DSECT
XR6       DS       0F
XYEP      EQU      X'40'
XYSP      EQU      X'4A'
          DS       3F
          DS       26H
XYE       DS       10C
XYS       DS       10C
          DS       3F
          DS       1H
          DS       2C
          DS       3H
          DS       1C
CHAR      DS       1C
KNY1T     BOX
          SR       R15,R15
          LA       R13,2
          SR       R12,R12
KNY11     LA       R8,0(R6,R15)
          TR       XYEP+1(1,R8),FFE
          LH       R7,XYE(R15)
          EX       0,FFFE(R7)
KNY1NY    TR       XYSP+1(1,R8),FFS
          LH       R7,XYS(R15)
          EX       0,FFFS(R7)
KNY1J     CR       R12,R13
          BC       8,NKNY1
          LR       R12,R13
KNY12     BXLE     R15,R13,KNY11
YKNY1     EQU      *
          MVI      CHAR,C'Y'
          BC       15,BEXIT5
KKNY1     EQU      *
          MVI      CHAR,C'K'
          BC       15,BEXIT5
NKNY1     EQU      *
          MVI      CHAR,C'N'
BEXIT5    BEXIT    EXO
FFE       DS       0H
          DC       8X'00'

```

```

      DC      X'08'
      DC      2X'00'
      CC      X'C4'
      CC      X'08'
      CC      2X'00'
      DC      X'C4'
FFFE   DS      OF
      BC      15,KNY1NY
      BC      15,NKNY1
      BC      15,KKNY1
FFS    DS      OH
      DC      2X'00'
      DC      2X'04'
      DC      3X'00'
      CC      X'C4'
      DC      8X'00'
FFFS   DS      OF
      BC      15,KNY12
      BC      15,KNY1J
      END

```

KVXYT

*FUNCTION

*

*DISTINGUISHES AMONG 2-STROKE (ALL VERT) K, V, X, AND Y BASED ON THE
*POSITIONS OF THE STARTING AND ENDING POINTS

*

*

*

*CALL

* RCS KVXYTA,ECHAR

*EXIT CHAR WHEN A CHARACTER IS RECOGNIZED

*

*

*

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R7, R8, R12, R13, R15

*

*

```

      USING XR6,R6
EX0    EQU     0
      REGS
D6      DSECT
XR6     CS      OF

```

XYEP	EQU	X'40'
XYSP	EQU	X'4A'
	DS	3F
	DS	26H
XYE	DS	10C
	DS	10C
	DS	3F
	DS	1H
	DS	2C
	DS	3H
	DS	1C
CHAR	DS	1C
KVXYT	BOX	
	SR	R15,R15
	LA	R13,2
	SR	R12,R12
KVXY1	LA	R8,0(R6,R15)
	TR	XYEP+1(1,R8),EEE
	LH	R7,XYE(R15)
	EX	0,EEEE(R7)
KXY	TM	XYSP+1(R8),X'03'
	BC	1,KKVXY
	BC	12,KVXY2
KVXY3	CR	R12,R13
	BC	8,VKVXY
	CR	R15,R13
	BC	8,XKVXY
	LR	R12,R13 SET J=1
KVXY2	BXLE	R15,R13,KVXY1
XKVXY	EQU	*
	MVI	CHAR,C'X'
	BC	15,BEXIT6
KKVXY	EQU	*
	MVI	CHAR,C'K'
	BC	15,BEXIT6
VKVXY	EQU	*
	MVI	CHAR,C'V'
	BC	15,BEXIT6
YKVXY	EQU	*
	MVI	CHAR,C'Y'
BEXIT6	BEXIT	EX0
EEE	DS	0H
	DC	X'08'
	DC	X'08'
	DC	X'08'
	DC	X'00'
	DC	X'08'
	DC	X'08'
	DC	X'08'
	DC	X'00'
	DC	X'08'

```

DC      X'08'
DC      X'08'
DC      X'00'
DC      X'04'
DC      2X'0C'
DC      X'00'
EEEE    DS      OF
BC      15,KXY
BC      15,KVXY2
BC      15,YKVXY
BC      15,KVXY3
END

```

MWT

*FUNCTION

*
 *DISTINGUISHES BETWEEN 3-STROKE (2 VERTS, 1 V-LIKE) OR 4-STROKE (ALL
 *VERT) M AND W BASED ON THE POSITIONS OF THE ENDING POINTS
 *

*
 *

*CALL

* RCS MWT, ECHAR
 *EXIT CHAR WHEN A CHARACTER IS RECOGNIZED
 *

*
 *

*INPUT REGISTERS

*

* C(R6) = ADDRESS OF THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
 * (PASSED DOWN DIRECTLY FROM CHAREC , NOT SET SPECIFICALLY IN
 * REC)
 *

* C(R13) = NO. OF STROKES - 1
 *

*
 *

*INTERNAL REGISTERS. R8, R9, R11, R12, R15
 *

*
 *

```

EX0      USING XR6,R6
          EQU    0
          REGS
D6        DSECT
XR6       DS      OF
XYEP      EQU    X'40'

```

	DS	3F	
	DS	26H	
XYE	DS	10C	
	DS	10C	
	DS	3F	
	DS	1H	
	DS	2C	
	DS	3H	
	DS	1C	
CHAR	DS	1C	
MWT	BOX		
	SR	R15,R15	
	SR	R9,R9	J
	SR	R11,R11	K
	LA	R12,1	
MW11N	SLL	R15,1	
	LA	R8,0(R6,R15)	
	TR	XYEP+1(1,R8),GGE	
	LH	R8,XYE(R15)	
	EX	0,GGGE(R8)	
MW11Q	LA	R9,1(C,R9)	J=J+1
	BC	15,MW11	
MW13Q	LA	R11,1(C,R11)	
MW11	SRL	R15,1	
	BXLE	R15,R12,MW11N	
	CR	R9,R12	
	BC	6,MWW	
	CR	R11,R12	J=1
	BC	6,MWW	K NOT 1
MWM	EQU	*	
	MVI	CHAR,C'M'	
	BC	15,BEXIT4	
MWW	EQU	*	
	MVI	CHAR,C'W'	
BEXIT4	BEXIT	EXO	
GGE	DS	0H	
	DC	8X'04'	
	DC	X'08'	
	DC	2X'04'	
	DC	X'00'	
	DC	X'08'	
	DC	X'04'	
	DC	X'04'	
	DC	X'00'	
GGGE	DS	0F	
	BC	15,MW11Q	
	BC	15,MW11	
	BC	15,MW13Q	
	END		

PSTEST

*FUNCTION

*
*DISTINGUISHES AMONG COMMA, APOSTROPHE, AND NORMAL SIZE CHARACTERS.
*NORMAL SIZE IF ITS HEIGHT IS GREATER THAN 3/8 OF THE NORMALLY EXPECTED
*CHARACTER HEIGHT ('CHAREC' SETS DYM = 3/2 NORM CHAR HEIGHT). COMMA IF
*TOP OF CHARACTER IS IN THE LOWER 5/8 OF A CHARACTER SPACE, OTHERWISE
*APOSTROPHE. IF COMMA, CHARACTER CENTER IS SHIFTED UPWARD BY
*(NORMAL CHARACTER HEIGHT/4) RASTERS.

*
*
*

*CALL

* RCS PSTESTA,ECHAR
*EXIT CHAR WHEN A CHARACTER IS RECOGNIZED

*
*
*

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R8, R9, R15

*

*

	USING XR6,R6
EX0	EQU 0
	REGS
D6	DSECT
XR6	DS 0F
	DS 3F
	DS 17H
DYC	DS 1H
	DS 2H
YTC	DS 1H
	DS 5H
	DS 20C
WIDTH	DS 1H
HEIGHT	DS 1H
	DS 2F
	DS 1H
	DS 2C
	DS 3H
	DS 1C
CHAR	DS 1C
	DS 54C
	DS 14H
DYM	DS 1H

```

CENT      DS      6H
PSTEST    DS      1F
          BCX
          LH      R15,DYM
          SRL     R15,2  1/4 MAX DY
          CH      R15,DYC
          BC      4,PTX
          LH      R9,YTC
          SRL     R9,2  YTC IN RASTERS
          LH      R15,HEIGHT
          SRL     R15,2
                                HEIGHT IN RASTERS
          SR      R8,R8
          DR      R8,R15
                                R8=REM(YTC/HEIGHT)
          SRL     R15,1
                                1/2 HEIGHT
          LR      R9,R15
          SRL     R9,2
                                1/8 HEIGHT
          AR      R15,R9
                                5/8 HEIGHT
          CR      R8,R15
          BC      2,PTA
*REM(YTC/HEIGHT) LSS, EQ 5/8 HEIGHT
          MVI     CHAR,X'EB'
*SHIFT CENTER OF COMMA UP BY HEIGHT/4 RASTERS
          L       R8,CENT
          LH      R15,HEIGHT
          SRL     R15,2
          AR      R8,R15
          ST      R8,CENT
          BC      15,PTX
*REM(YTC/HEIGHT) GTR 5/8 HEIGHT
          PTA     MVI     CHAR,X'FD'
          PTX     BEXIT EXO
          END

```

SYMT

*FUNCTION

*
 *RECOGNIZES GEOMETRIC SYMBOLS BASED FIRSTLY ON THE NO. OF TIMES EACH
 *16-DIRECTION (THE SAME AS THE DIRECTIONS IN THE INK TRACK) OCCURS,
 *THEN ON NO. OF TIME-CORNERS, THE 4-DIRECTION SEQUENCE, SEPARATION BET-
 *WEEN STARTING AND ENDING POINTS, AND ASPECT RATIO.

*

*

*

*CALL

* RCS SYMTA, ENOCHAR, ECHAR

*EXIT NOCHAR WHEN THE SYMBOL IS NOT ONE OF THE GEOMETRIC SYMBOLS
*EXIT CHAR WHEN A GEOMETRIC SYMBOL IS RECOGNIZED

*
*
*

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R7-R13, R15

*
*

	USING XR6,R6
EX0	EQU 0
EX4	EQU 4
	REGS
D6	DSECT
XR6	DS 0F
	DS 2F
CODE	DS 1F
	DS 10H
N	DS 1H
	DS 5H
DXC	DS 1H
DYC	DS 1H
	DS 8H
	DS 20C
	DS 3F
	DS 1H
	DS 2C
NCUSP	DS 1H
	DS 2H
	DS 1C
CHAR	DS 1C
	DS 6C
XSP	DS 10C
YSP	DS 10C
XEP	DS 10C
YEP	DS 10C
	DS 8C
	DS 21H
	DS 1F
	DS 6C
	DS 1H
	DS 1F
	DS 20C
D0	DS 1H
D1	DS 1H
	DS 2H
D4	DS 1H
	DS 2H
D7	DS 1H
D8	DS 1H

D9	DS	1H	
	DS	2H	
D12	DS	1H	
	DS	2H	
D15	DS	1H	
CN	DS	1H	
NTCUSP	DS	1H	
SYMT	BOX		
	SR	R8,R8	
	LH	R9,DN	
	LA	R15,5	
	CR	R8,R15	
	LR	R12,R9	
	LH	R13,CN	
	LR	R15,R13	
	SRL	R15,2	
	SR	R13,R15	
*C(R12)	=0.2(NO. OF DIRECTION OCCURANCES)		
*C(R13)	=0.75(NO. OF DIRECTION OCCURANCES)		
*IS NO.	OF HORIZ GTR 0.2 DN?		
	LH	R7,D0	
	AH	R7,D8	
	CR	R7,R12	
	BC	2,BOXTRI	YES
*NO, IS	NO. OF HORIZ IN 1 DIRECTION GTR		
*1/8 DN	?		
	LH	R8,DN	
	SRL	R8,3	
*NEARLY	RIGHT-DIRECTION		
	LH	R7,D15	
	CR	R7,R8	
	BC	2,RIGHT	
	LA	R10,2	
	SR	R9,R9	
	LA	R11,2	
NEARR	LH	R7,D0(R9)	
	CR	R7,R8	
	BC	2,RIGHT	
	BXLE	R9,R10,NEARR	
	LA	R11,18	
	CR	R9,R11	
	BC	10,NOTSQ	
*NEARLY	LEFT-DIRECTION		
	LA	R9,14	
	B	NEARR	
*NO			
*IS NO.	OF 4 MAIN DIRECTIONS LESS THAN 1/8 DN?		
NOTSQ	LH	R7,D0	
	AH	R7,D8	
	AH	R7,D4	
	AH	R7,D12	

```

LH      R9,DN
SRL     R9,3
CR      R7,R9
BC      4,PBOXX      YES
*NO, IS  IT GTR 0.2 DN?
CR      R7,R12
BC      2,ROUND      YES
LA      R10,256
BC      15,ROUND
*IS NO.  OF 4 MAIN DIRECTIONS AT LEAST
*3/4 DN  ?
BOXTRI  AH      R7,D4
        AH      R7,D12
        CR      R7,R13
        BC      10,BOXX      YES
*NO, IS  NC. OF VERTS GTR 1/4 DN?
LH      R9,DN
SRL     R9,2
LH      R7,D4
AH      R7,D12
CR      R7,R9
BC      2,BOXX
*NO
*IS NO.  IN 1 HORIZONTAL DIRECTION
*PLUS 2  OTHER DIRECTIONS AT LEAST 3/4 DN?
*FIRST  FIND HORIZ. DIRECTION
LH      R7,D0
CR      R7,R12
BC      10,RIGHT
LH      R7,D8
CR      R7,R12
BC      4,ROUND
*R7 HAS  NC. OF RIGHTS OR LEFTS
*FIND D  OWNWARD DIRECTION
RIGHT   LA      R9,20
        LA      R10,2
        LA      R11,30
DOWN    LH      R8,D0(R9)
        AH      R8,D0-2(R9)
        CR      R8,R12
        BC      10,DOWNX
        BXLE    R9,R10,DOWN
*NO SUC  H DOWNWARD DIRECTION
        BC      15,ROUND
*R9 CON  TAINS DOWNWARD DIRECTION CODE
*R8 CON  TAINS NO. OF DOWNWARDS
*FIND U  PWARD DIRECTION DIRECTION
DOWNX   AR      R7,R8
        LA      R15,24
        CR      R9,R15
        BC      2,DGTR12

```

```

BC      4,DLSS12
*DOWNWA RD DIRECTION IS 12
*UP DIR ECTION MUST BE 3,4, OR 5
      LA      R9,6
      LA      R11,10
      BC      15,UP
*DOWN D  IR. IS 10 OR 11
**UP DI  R. MUST BE 4,5,OR 6
CLSS12   LA      R9,8
      LA      R11,12
      BC      15,UP
*DOWN D  IR. IS 13 OR 14
*UP DIR  . MUST BE 2,3,OR4
DGTR12   LA      R9,4
      LA      R11,8
*FIND U  P DIRECTION
UP        LH      R8,D0(R9)
      AH      R8,D0+2(R9)
      CR      R8,R12
      BC      10,UPX
      BXLE    R9,R10,UP
*NO SUC  H UPWARD DIRECTION
      BC      15,ROUND
*R7 CON  TAINS NO. OF HORIZS. ? DOWNS
*R8 CON  TAINS NO. OF UPWARDS
*IS TOT  AL HORIZ, UPS, AND DOWNS
*GREATE  R THAN 3/4 DN?
UPX      AR      R7,R8
      CR      R7,R13
      BC      12,NOTSQ
*TRIANGLE, TRAPAZOID, OR ELLIPSE
*TRIANGLE IF HORIZ NOT GTR 0.375 DN
      LH      R7,D0
      AH      R7,D8
      LR      R15,R13
      SRL     R15,1
      CR      R7,R15
      BC      12,TRIX
*CHECK TIME CORNERS FOR TRAP
      CLI     NTCUSP+1,X'02'
      BC      2,TRAPXX
      B       ELPSX
*SYMBOL  NCT BOX OR TRIANGLE
*TEST F  OR CIRCLE OR ELLIPSE
*CR TRAPAZOID
*4-ANGL  E SEQUENCE MUST BE
*0-3-2-  1 OR 2-3-0-1
ROUND    CLI     CODE,X'B1'
      BC      8,OKSYM
      CLI     CODE,X'39'
      BC      8,OKSYM

```

0.375 DN

```

*3-2-1- 0
          CLI  CODE,X'E4'
          BC   8,CKSYM
*2-1-0-  3
          CLI  CODE,X'93'
          BC   8,OKSYM
*1-0-3-  2
          CLI  CODE,X'4E'
          BC   8,OKSYM
*3-0-1-  2
          CLI  CODE,X'C6'
          BC   8,OKSYM
*0-1-2-  3
          CLI  CODE,X'1B'
          BC   8,CKSYM
*1-2-3-  0
          CLI  CODE,X'6C'
          BC   8,OKSYM
*IS THI  S A POTENTIAL PBOX?
          LA   R9,256
          CR   R9,R10
          BC   8,PBOXX
*TEST FOR NARROW TRAPEZOID
*NO MORE THAN 4 ANGLES
*MOSTLY HORIZONTAL
*CLOSE ENDPOINTS
          CLI  N+1,X'04'
          BC   2,NOSYMX
          LH   R7,D0
          AH   R7,D1
          AH   R7,D15
          AH   R7,D8
          AH   R7,D7
          AH   R7,D9
          LH   R9,DN
          SRL  R9,1
          CR   R7,R9
          BC   4,NOSYMX
          CLI  NTCUSP+1,X'02'
          BC   12,XELPS
          LA   R10,128
          BC   15,PBOXX
*DECIDE IF ELLIPSE
*ARE ENDPOINTS SEPARATED VERTICALLY
XELPS    LH   R7,YSP
          SH   R7,YEP
          LPR  R7,R7
          SLL  R7,1
          CH   R7,DYC
          BC   4,ELPSX
NOSYMX   BEXIT EXO

```

*DECIDE BETWEEN CIRCLE AND ELLIPSE
*AND TRAPAZOID

```
CKSYM    CLI    NTCUSP+1,X'02'  
          BC     2,TRAPXX  
          LH     R7,DYC  
          SLL    R7,1  
          CH     R7,DXC  
          BC     2,CIRCX  
ELPSX    EQU    *  
          MVI    CHAR,X'76'  
          BC     15,BEXIT7  
BOXX     EQU    *  
          MVI    CHAR,X'73'  
          BC     15,BEXIT7  
CIRCX    EQU    *  
          MVI    CHAR,X'74'  
          BC     15,BEXIT7  
TRIX     EQU    *  
          MVI    CHAR,X'75'  
          BC     15,BEXIT7  
TRAPXX   EQU    *  
          MVI    CHAR,X'78'  
          BC     15,BEXIT7  
*TEST F  OR CLOSNESS OF ENDP  
PBOXX    LH     R7,XSP  
          SH     R7,XEP  
          LPR    R7,R7  
          SLL    R7,1  
          CH     R7,DXC  
          BC     2,NOSYMX  
          LH     R7,YSP  
          SH     R7,YEP  
          LPR    R7,R7  
          SLL    R7,1  
          CH     R7,DYC  
          BC     2,NOSYMX  
          LA     R9,128  
          CR     R9,R10  
          BC     8,TRAPXX  
XXPBOX   EQU    *  
          MVI    CHAR,X'77'  
BEXIT7   BEXIT  EX4  
          END
```

TILDT

*FUNCTION

*
 *RECOGNIZES TILDA BASED ON CHAR. HEIGHT, ASPECT RATIO, AND FIRST FOUR
 *DIRECTIONS.
 *ALTHOUGH THIS ROUTINE ENCORPORATES ALL THE CODE FOR RECOGNITION, IT
 *PRESENTLY ALWAYS TAKES THE NOT TILDA EXIT.

*
 *
 *

*CALL

* RCS TILDTA,ENOTTIL,ETIL
 *EXIT NOTTIL WHEN THE CHARACTER IS NOT A TILDA
 *EXIT TIL WHEN THE CHARACTER IS A TILDA

*
 *
 *

*INPUT REGISTER. R6

*

*INTERNAL REGISTER. R7

*
 *

	USING XR6,R6	
EX0	EQU 0	
EX4	EQU 4	
	REGS	
D6	DSECT	
XR6	DS 0F	
	DS 2F	
CODE	DS 1F	
	DS 17H	
DYC	DS 1H	
	DS 4H	
ASPR	DS 1H	
	DS 3H	
	DS 20C	
	DS 3F	
	DS 1H	
	DS 2C	
	DS 3H	
P	DS 1C	
CHAR	DS 1C	
	DS 54C	
	DS 14H	
DYM	DS 1H	
TILDT	BOX	
	LH R7,DYM	
	SRL R7,2	1/4 DYM
	CH R7,DYC	
	BC 4,NOTIL	
*DYC LE	SS THAN 1/4 DYM	
	CLI ASPR+1,X*02*	

```

BC      2,NOTIL
CLI     ASPR+1,X'01'
BC      4,NOTIL
* ASPEC  T RATIO IS BETWEEN 1/2 AND 1/4
*TEST F  OR ALLOWABLE SEQUENCES
*0-0-0-  0
          CLI     CODE,X'00'
          BC      8,TIL
*0-3-0-  0
          CLI     CODE,X'30'
          BC      8,TIL
*0-3-0-  1
          CLI     CODE,X'31'
          BC      8,TIL
*1-0-0-  0
          CLI     CODE,X'40'
          BC      8,TIL
*1-0-1-  1
          CLI     CODE,X'45'
          BC      8,TIL
*1-0-3-  0
          CLI     CODE,X'4C'
          BC      8,TIL
*1-0-3-  1
          CLI     CODE,X'4D'
          BC      8,TIL
*1-3-0-  0
          CLI     CODE,X'70'
          BC      8,TIL
*1-3-0-  1
          CLI     CODE,X'71'
          BC      8,TIL
*1-3-1-  1
          CLI     CODE,X'75'
          BC      8,TIL
NCTIL   BEXIT EX0
*TEMPOR ARILY KILL TILDA
TIL     BC      15,NOTIL
          MVI     P,X'02'
TILX    EQU     *
          MVI     CHAR,X'D0'
          BEXIT EX4
          END

```

TPXY

*FUNCTION

*

*DISTINGUISHES AMONG 2-STROKE (1 VERT, 1 HORIZ) T, X, Y, AND PLUS BASED
*ON THE POSITIONS OF STARTING AND ENDING POINTS

*

*

*

*CALL

* RCS TPXYA,ECHAR

*EXIT CHAR WHEN A CHARACTER IS RECOGNIZED

*

*

*

*INPUT REGISTER. R6

*

*INTERNAL REGISTERS. R7-R9, R12, R13, R15

*

*

EX0 USING XR6,R6

EQU 0

REGS

D6 DSECT

XR6 DS 0F

DS 3F

DS 26H

XYE DS 10C

DS 10C

DS 3F

DS 1H

DS 2C

DS 3H

P DS 1C

CHAR DS 1C

XYEP EQU X'40'

XYSP EQU X'4A'

TPXY BOX

*IS SECOND STROKE HORIZONTAL?

CLI P,X'02'

BC 8,YES

LA R9,2(R6)

VERT REF

B GO

YES LA R9,0(R6)

VERT REF

GO EQU *

SR R15,R15

SR R12,R12

LA R13,2

TPLUS1 LA R7,0(R6,R15)

LH R8,XYEP(R7)

STH R8,XYE+4

TR XYE+5(1),TTE

LH R8,XYE+4

```

EX      0,TTTE(R8)
TPLUS4  TM      XYSP+1(R7),X'CC'
BC      8,TPLS1  YS GTR 3/4 DELTA Y
BC      1,TPLUS2 YS LESS 1/4 DELTA Y
*START  IN MIDDLE Y
TM      XYSP+1(R7),X'08'
BC      1,TPLUS5
TM      XYEP+1(R7),X'0C'
BC      9,TPLUS2
BC      4,PTPXY
*START  IN LOWER MID Y
TPLUS5  TM      XYEP+1(R7),X'0C'
BC      1,TPLUS2
BC      12,PTPXY
*START  AT TCP
TPLS1    TM      XYEP+1(R7),X'0C'
BC      8,TTPXY
BC      4,TPLUS3
*END AT  BOTTOM
TM      XYEP+1(R7),X'03'
BC      1,TPLUSX LEFT
BC      8,XTPXY RIGHT
BC      4,TPLUS2 MIDDLE
*START  AT TCP, END IN MIDDLE Y
*IS END  IN RIGHT MID Y?
TPLUS3   TM      XYEP+1(R7),X'03'
BC      5,TPLUS2
*YES
TM      XYEP+1(R7),X'80'
BC      1,XTPXY
CR      R12,R13
BC      8,YTPXY
BC      6,TPLUS2
TPLUSX   LR      R12,R13
TPLUS2   BXLE    R15,R13,TPLUS1
CR      R12,R13
BC      8,XTPXY
TTPXY    EQU     *
*IS VERT START IN UPPER LEFT
CLI      XYSP+1(R9),X'CO'
BC      8,YTPXY
MVI      CHAR,C'T'
BC      15,BEXIT3
PTPXY    EQU     *
*IS VERT START IN UPPER LEFT
CLI      XYSP+1(R9),X'CO'
BC      8,YTPXY
MVI      CHAR,X'CE'
BC      15,BEXIT3
XTPXY    EQU     *
MVI      CHAR,C'X'

```

```

YTPXY      BC      15,BEXIT3
            EQU      *
            MVI      CHAR,C'Y'
BEXIT3     BEXIT    EX0
TTE         DS      0H
            DC      X'00'
            DC      2X'04'
            DC      X'00'
            DC      X'10'
            DC      2X'04'
            DC      X'08'
            DC      X'10'
            DC      2X'04'
            DC      X'10'
            DC      X'0C'
            DC      3X'10'
ITTE        DS      0F
            BC      15,TTPXY
            BC      15,YTPXY
            BC      15,PTPXY
            BC      15,XTPXY
            BC      15,TPLUS4
            END

```

VERTST

*FUNCTION

*
*DETERMINES THE SET OF STROKE TYPES WHEN THE MOST RECENT STROKE IS A
*VERTICAL. BASED ON 'P' AND THE NO. OF STROKES

*
*
*
*CALL
* RCS VERTSTA,EVI,EV2,EVIH1,EVINOT
*EXIT V1 WHEN THERE IS ONLY ONE VERTICAL STROKE (THE MOST RECENT)
*EXIT V2 WHEN THERE ARE 2 VERTICAL STROKES
*EXIT VIH1 WHEN THERE IS 1 VERT STROKE AND 1 HORIZ STROKE
*EXIT VINOT WHEN THERE IS 1 VERT STROKE AND THE PREVIOUS SUBCHARACTER
* IS NEITHER VERT OR HORIZ

*
*
*
*INPUT REGISTER. R6

*
*INTERNAL REGISTERS. NONE

*
*
*

```

      USING XR6,R6
EX0    EQU    0
EX12   EQU    12
EX4    EQU    4
EX8    EQU    8
      REGS
D6     DSECT
XR6    DS     0F
      DS     3F
      DS     11H
SN     DS     1H
      DS     14H
      DS     20C
      DS     3F
      DS     1H
      DS     2C
      DS     3H
P      DS     1C
VERTST BOX
      CLI     P,X'02'
      BC      8,EQ2
      BC      2,GTR2
LSS2   CLI     P,X'01'
      BC      8,EQ1
LSS1   CLI     SN+1,X'01'
      BC      2,GTR2      SN>1
      MVI     P,X'01'
      BEXIT   EX0
EQ1    MVI     P,X'00'
      BEXIT   EX4
EQ2    MVI     P,X'00'
      BEXIT   EX8
GTR2   MVI     P,X'01'
      BEXIT   EX12
      END
```

APPENDIX

THE OS/360 OPERATING SYSTEM--2250 DISPLAY RECOGNITION PROGRAM

In order to modify the GRAIL recognition program for operation under OS/360 and in conjunction with a 2250 display, only CHAREC and the macros need be changed:

The following changes must be made for the program to operate under OS/360:

- 1) Either the GRAIL macros (see MACROS below) must be modified so that they do not require the SVC (supervisor call) command, or the GRAIL SVC's must be built into OS/360. The GRAIL SVC's are used to initiate and terminate processes, synchronize parallel processes, go to the wait state, etc., and may be replaced by the equivalent code. The macros must be added to the macro library.
- 2) CHAREC must be modified to await the asynchronous event of either a pendown or the expiration of the real-time interval timer. This is done by first issuing a STIMER OS/360 macro and then a WAIT OS/360 macro for the Tablet pen. If the timer expires, the ECB (Event Control Block) for the WAIT is posted with a special code and control is returned to the system. When control is returned from the WAIT, the special code is checked to see if it was posted by the timer; if not, the timer is cancelled and the pendown is processed.

The following changes must be made for the program to operate in conjunction with a 2250 display:

- 1) CHAREC must be modified to do its inking on the 2250. This involves formatting the x,y coordinates and writing them into the 2250 buffer. The method of erasing the ink track must similarly be modified

- 2) The character codes (see CRT Display Character Codes below) must be converted to EBCDIC (Extended Binary-Coded-Decimal Interchange Code). This may be done either in CHAREC prior to outputting a code, or externally to the recognition program.

CRT DISPLAY CHARACTER CODES

Upper-Case		Lower-Case		Punctuation		Geometric	
Letter	Hex Code	Letter	Hex Code	Symbol	Hex Code	Symbol	Hex Code
A	C1	A	81	+	CE		73
B	C2	B	82	-	EO		74
C	C3	C	83	=	FE		75
D	C4	D	84	/	E1		76
E	C5	E	85	(	CD		77
F	C6	F	86	)	DD		78
G	C7	G	87	*	DC		
H	C8	H	88	\$	DB		
I	C9	I	89	.	CB		
J	D1	J	91	,	EB		
K	D2	K	92	!	FD		
L	D3	L	93	#	FB		
M	D4	M	94	[	CF		
N	D5	N	95	]	DF		
O	D6	O	96	<	CC		
P	D7	P	97	>	EE		
Q	D8	Q	98	^	70		
R	D9	R	99	~	DO		
S	E2	S	A2				
T	E3	T	A3				
U	E4	U	A4				
V	E5	V	A5				
W	E6	W	A6				
X	E7	X	A7				
Y	E8	Y	A8				
Z	E9	Z	A9				
				Special			
				Symbol	Hex Code		
				Erasure	72		
				Cannot Interpret	EF		
						Numbers	
						Number	Hex Code
						0	F0
						1	F1
						2	F2
						3	F3
						4	F4
						5	F5
						6	F6
						7	F7
						8	F8
						9	F9

REGISTER ASSIGNMENT

Registers are referred to as R0, R1, ..., R15, rather than as 0, 1, ..., 15. The equivalence is made by the macro REGS (see MACROS below).

R1 through R5 have special system assignments:

R1 is the contextual base-register.

R2 is the read-only code base-register.

R3 is the data base-register for data defined within a given context.

R4 is an address argument register, and is used in process calls.

R5 is used in macro and process calls, and as the address argument register for SS instructions with two formal parameters.

R6 has a special assignment in REC and the RCS's-- it is locally loaded by REC to reference DSECT type label descriptions of CHAREC's data.

PROCESSES

CHAR

CHAR is an interface process between a Tablet input device and the recognition program on one side, and an application program on the other. It allows its parent process (the application program) to interact with the Tablet by providing a convenient level of control. In addition to providing CHAREC outputs (see CHAREC outputs below), CHAR provides the raw Tablet data to the user. CHAR is a read-only reentrant process that uses two other read-only processes--CHAREC (see p. 20), a reentrant process, and TABLET, a serially re-usable process (i.e., each use must wait for the hardware device to be free), which communicates with the Tablet.

CHAR allows the following user controls:

- Permit/inhibit inking (stylus tracking) by CHAREC.
- Permit/inhibit character recognition.
- Permit/inhibit halting CHAREC.
- Permit/inhibit providing raw data to either CHAREC, or the user.
- Specify ink vector length.

CHAR has the following parallel task exits:

- Match (coincidence of the virtual tablet stylus and displayed data) detected--similar to a light pen strike.
- Keyboard character detected (for optional keyboard device).
- Penup detected.
- Raw data buffer filled.
- Character recognized.
- Character not recognized.

CHAR has the following terminal exits:

- Normal termination exit.
- Error exit (channel multiplex or device error).

CLOCK

Function. This process acts as a real-time clock that is turned off (takes the terminal turned-off exit) by CHAREC as a result of a pendown, or sets an alarm (takes the expired parallel task exit) if the 360 real-time clock runs longer than a prespecified time before a pendown occurs.

Call.

INST ACLK,CLKA,FWAITBX,ITIME,EEXP,ETOFF

ACLK is a linkage between CHAREC's context and
CLOCK's context.

CLKA is a link to CLOCK.

WAITBX is CHAREC's PSG.

TIME is the time at which CLOCK takes the expired exit.

Exit EXP is the expired (parallel) exit.

Exit TOFF is the turned-off (terminal) exit.

MACROS

BEXIT

*FUNCTION

*

*RETURN FROM A REMOTE CODE SEQUENCE

*

*

*

*MACRO DEFINITION

*

```
MACRO
&LABEL    BEXIT    &EXIT
&LABEL    L        R2,4(R1)
          EX        C,&EXIT.(R5)
          MEND
```

BOX

*FUNCTION

*

*INITIATES A REMOTE CODE SEQUENCE

*

*

*

*MACRO DEFINITION

*

```
MACRO
&LABEL    BOX
&LABEL    CSECT
          USING    *,R2
          MEND
```

*

CLEAR

*FUNCTION

*

*PARALLEL PROCESS SYNCHRONIZER. NULLIFIES THE ADVENT OF 'WATE' AND/OR
*'SET'

*

*

*

*MACRO DEFINITION

*

```
MACRO
&LABEL    CLEAR    &CNTX=I,&PSG=0
           AIF      ('&CNTX' EQ 'I').A
&LABEL    L        R5,&PSG
           TM       0(R5),X'01'
           BC       8,#+6
           SVC      CRW
           NI       0(R5),X'7E'
           MEXIT
.A
&LABEL    LA       R5,&PSG
           TM       0(R5),X'01'
           BC       8,#+6
           SVC      CRW
           NI       0(R5),X'7E'
           MEND
```

EPLOG (Epilogue)

*FUNCTION

*

*TERMINATES A PROCESS

*

*

*

*MACRO DEFINITION

*

```
MACRO
&LABEL    EPLOG    &EXIT,&STATE,&PSW,&ENTER
&LABEL    LA       R5,&EXIT
```

```

        AIF ('&STATE' EQ 'S').B
        SVC      RETURN
MEXIT
        ANOP
        LA      R6,&PSW
        LA      R7,&ENTER
        SVC      RETSUP
        MEND

```

INST (Instance)

*FUNCTION

*
*
*GENERATES THE CALLING SEQUENCE FOR A RE-ENTRANT PROCESS
*

*MACRO DEFINITION

```

        MACRO
&LABEL  INST      &CNTX,&LOCN,&A1,&A2,&A3,&A4,&A5,&A6,&A7,&A8,&A9,&A10X
          ,&A11,&A12,&A13,&A14,&A15,&A16,&A17,&A18,&A19,&A20,&A21,X
          &A22,&A23,&A24,&A25,&A26,&A27,&A28,&A29,&A30,&A31,&A32,&X
          A33,&A34,&A35,&A36,&A37,&A38,&A39,&A40,&A41,&A42,&A43,&AX
          44,&A45,&A46
        LCLA  &A1,&A2,&A3,&A4
        LCLC  &CG1,&CG2,&CG3
&LABEL  LA      R4,&CNTX
        LA      R5,&LOCN
        SVC      FORMAL
&CG3     SETC  ' '
&A1      SETA  2
&A2      SETA  6
&A3      SETA  1
        .A      ANOP
&A1      SETA  &A1+1
&A2      SETA  &A2+1
&CG1     SETC  '&SYSLIST(&A1)'(1,1)
&CG2     SETC  '&SYSLIST(&A1).&CG3'(2,8)
        AIF    ('&CG1' NE 'E').E
&A3      SETA  0
        AIF    (&A1 GT 3).G
        .F      ANOP
&A4      SETA  &A1-3
&A4      SETA  &A4*4

```

```

        LA      R7,GS&SYSNDX
        ST      R7,&AL4.(R6)
        ST      R4,&CNTX
        BR      R5
GS&SYSNDX      B      &CG2
.Y            ANCP
&AL1          SETA    &AL1+1
              AIF      ('&SYSLIST(&AL1)' EQ '').W
&CG2          SETC    '&SYSLIST(&AL1).&CG3'(2,8)
              B      &CG2
              AGCB     .Y
.W            ANCP
              MEXIT
.E            AIF      (&AL2 LE 15).B
.G            ANOP
&AL2          SETA    &AL2-1
&AL4          SETA    &AL1-3
&AL4          SETA    &AL4-&AL2+6
              AIF      (&AL1 GT 12).X
&AL4          SETA    0
.X            ANOP
&AL4          SETA    &AL4*4
              STM      R7,R&AL2,&AL4.(R6)
&AL2          SETA    7
              AIFB     (&AL3 EQ 0).F
.B            ANCP
              AIF      ('&CG1' EQ '1').C
              L        R&AL2,&CG2
              AGCB     .A
.C            ANOP
              LA      R&AL2,&CG2
              AGCB     .A
              MEND

```

PARL (Parallel)

*FUNCTION

*
 *INITIATES A PARALLEL PROCESS. THIS PROCESS FIRST TAKES THE HIGH
 *PRIORITY EXIT. WHEN THE HIGH PRIORITY TASK IS COMPLETED OR SUSPENDED,
 *THIS PROCESS TAKES THE LOW PRIORITY EXIT.
 *
 *
 *

*MACRO DEFINITION

*

```

MACRO
&LABEL    PARL    &CNTX=I,&LOW=0,&HIGH=0,&STATE=0,&PSW=0
          AIF      ('&CNTX' EQ 'F').A
&LABEL    SVC     PARIN
          B        &LOW
          B        &HIGH
          MEXIT
.A
&LABEL    ANOP
&LABEL    SVC     PARLEL
          B        &LOW
          LH       R5,10(R2)
          BCT      R5,#+4
          SLL      R5,2
          L        R5,0(R5,R1)
          AIF      ('&STATE' NE '0').B
          L        R1,0(R1)
          LM       R2,R3,4(R1)
          EX       0,&HIGH.(R5)
          MEXIT
.B
          LA       R5,&HIGH.(R5)
          ST       R5,&PSW+4
          LA       R5,&PSW
          L        R1,0(R1)
          LM       R2,R3,4(R1)
          LPSW     0(R5)
          MEND

```

PAWS (Pause)

*FUNCTION

*
*TERMINATES A FLOW OF CONTROL. RESULTS IN INITIATING THE NEXT TASK ON
*THE SUPERVISOR TASK LIST, WHICH, IF THE ONLY TASK, WILL BE THE WAIT
*STATE WITH TRAPS ENABLED.
*
*
*

*MACRO DEFINITION

```

MACRO
&LABEL    PAWS
&LABEL    SVC     PAUSE
          MEND

```

*
*

PROCS (Process)

*FUNCTION

*

*SETS UP THE PROCESS ENTRY POINT, ITS IDENTIFICATION NUMBER, AND ITS
*STORAGE REQUIREMENTS

*

*

*

*MACRO DEFINITION

*

```
MACRO
&LABEL    PROCS    &CLEAR=3,&CNTX=3,&AUTO=0,&ID=80000000,&PRCLG=0
&LABEL    CSECT
          USING    *,R2
          LM      R2,R3,4(R4)
          B       &PRCLG
          DC      H'&CLEAR'
          DC      H'&CNTX'
          DC      H'C'
          DC      H'&AUTO'
          DC      X'&ID'
          MEND
```

PROLG (Prologue)

*FUNCTION

*

*INITIATES A PROCESS--PRECONDITIONS CERTAIN VALUES

*

*

*

*MACRO DEFINITION

*

```
MACRO
&LABEL    PROLG    &AUTO=YES,&STATE=0,&PSG=0,&LINK=0
          AIF     ('&AUTO' EQ 'C').A
&LABEL    DS      0H
          LR      R1,R4
          AIF     ('&STATE' EQ '0').B
          LA      R4,&PSG
          LA      R5,&LINK
```

```

        SVC    SUPER
.B      MEXIT
.A      ANCP
&LABEL DS      OH
        LR     R1,R4
        AIF     ('&STATE' EQ '0').C
        SVC     SUPER
.C      MEXIT
        MEND

```

RCS (Remote Code Sequence)

*FUNCTION

*
*GENERATES THE CALLING SEQUENCE FOR A REMOTE CODE SEQUENCE--A PROCESS
*WITH ONLY REGISTER I/O WHICH OPERATES IN THE ENVIRONMENT OF THE PARENT
*(CALLING) CONTEXT

*
*
*
*MACRO DEFINITION

```

*
        MACRO
&NAME   RCS      &LABEL,&A6,&A7,&A8,&A9,&A10,&A11,&A12,&A13,&A14,&A15X
          ,&A0,&E1,&E2,&E3,&E4,&E5,&E6,&E7,&E8,&E9,&E10,&E11,&E12
        LCLA     &A1,&A2,&A3
        LCLC     &CG1,&CG2,&CG3
&NAME   DS      OH
&A1     SETA     1
&A3     SETA     0
&CG3    SETC     ' '
.D      ANCP
&A1     SETA     &A1+1
&A2     SETA     &A1+4
&A3     SETA     &A3+1
        AIF     ('&SYSLIST(&A1)' EQ '').A
&CG1    SETC     '&SYSLIST(&A1)'(1,1)
&CG2    SETC     '&SYSLIST(&A1).&CG3'(2,8)
        AIF     ('&CG1' EQ 'E').C
        AIF     ('&CG1' EQ 'I').B
        L       R&A2,&CG2
        AGO8     .D
.B      LA       R&A2,&CG2
        AGO8     .D
.A      AIF     (&A3 EQ 15).C
        AGO8     .D

```

```
.C      L      R2,&LABEL
        BALR    R5,R2
.Y      B      &CG2
&AL1    SETA    &AL1+1
        AIF     ('&SYSLIST(&AL1)' EQ '').W
&CG2    SETC    '&SYSLIST(&AL1).&CG3'(2,8)
        AGCB    .Y
.W      ANOP
        MEND
```

REGS (Registers)

*FUNCTION

*

*GENERATES THE CODE R0 EQU 0, R1 EQU 1, . . . , R15 EQU 15

*THE SYMBOLIC FORM IS USED BY THE OTHER MACROS

*

*

*

*MACRO DEFINITION

*

&NAME	MACRO	
R0	EQU	0
R1	EQU	1
R2	EQU	2
R3	EQU	3
R4	EQU	4
R5	EQU	5
R6	EQU	6
R7	EQU	7
R8	EQU	8
R9	EQU	9
R10	EQU	10
R11	EQU	11
R12	EQU	12
R13	EQU	13
R14	EQU	14
R15	EQU	15
	MEND	

SET

*FUNCTION

*
 *PARALLEL PROCESS SYNCHRONIZER--DENOTES AN EVENT HAS OCCURED
 *RESULTS IN SUPERVISOR STACKING A 'WAIT'ED TASK ON THE SUPERVISOR TASK
 *LIST IF IN THE WAIT STATE

*
 *
 *
 *MACRO DEFINITION
 *

	MACRO	
&LABEL	SET	&CNTX=I,&PSG=0
	AIF	('&CNTX' EQ 'I').B
&LABEL	L	R5,&PSG
	AGC	.A
.B	ANOP	
&LABEL	LA	R5,&PSG
.A	TM	C(R5),X'01'
	BO	GS&SYSNDX
	OI	C(R5),X'80'
	B	GS&SYSNDX+2
GS&SYSNDX	SVC	STACK
	MEND	

SVCS

*FUNCTION

*
 *DEFINES PARAMETERS FOR MACROS

*
 *
 *
 *MACRO DEFINITION
 *

	MACRO	
&NAME	SVCS	
STACK	EQU	5
WAIT	EQU	6
CCUPID	EQU	7
FORMAL	EQU	8
AUTO	EQU	9
CRW	EQU	20
RETURN	EQU	15
PARIN	EQU	21
PARLEL	EQU	16
PAUSE	EQU	17
	MEND	

TABLE

*FUNCTION

*

*PERFORMS SEQUENCES OF TESTS ON ENCODED 1-BYTE FEATURES

*

*

*

*

*MACRO DEFINITION

*

	MACRO
&LABEL	TABLE &A1,&A2,&A3,&A4,&A5,&A6,&A7,&A8,&A9,&A10,&A11,&A12,&A13,X &A14,&A15,&A16,&A17,&A18,&A19,&A20,&A21,&A22,&A23,&A24,&X A25,&A26,&A27,&A28,&A29,&A30,&A31,&A32,&A33,&A34,&A35,&AX 36,&A37,&A38,&A39,&A40,&A41,&A42,&A43,&A44,&A45,&A46,&A4X 7,&A48,&A49
	LCLA &AL1,&AL2
	LCLC &CG1,&CG2,&CG3,&CG4,&CG5
	AIF ('&LABEL' EQ '').D
&LABEL	EQU *
.D	ANOP
&AL1	SETA 0
.A	ANOP
&AL1	SETA &AL1+1
	AIF ('&SYSLIST(&AL1)' NE '').B
	MEXIT
.B	ANOP
&CG1	SETC '&SYSLIST(&AL1)''(1,1)
	AIF ('&CG1' GT 'Z').C
&CG2	SETC 'AL1('
&CG3	SETC ')'
&CG4	SETC '&SYSLIST(&AL1)''(2,2)
	AIF ('&CG4' EQ 'EX').E
	DC &CG2&CG4&CG3
&AL1	SETA &AL1+2
&CG2	SETC 'X'''
&CG3	SETC ''''
&CG4	SETC '&SYSLIST(&AL1)''(1,2)
	DC &CG2&CG4&CG3
&AL1	SETA &AL1-1
&CG2	SETC 'AL1('

```

&CG3      SETC      ') '
&CG4      SETC      '-DATA'
          DC          &CG2&SYSLIST(&AL1)&CG4&CG3
&AL1      SETA      &AL1+1
          AGCB       .A
.C         ANOP
&AL2      SETA      4096*&SYSLIST(&AL1)
&CG2      SETC      '+ '
&CG3      SETC      '-BASE'
&CG4      SETC      'AL2('
&CG5      SETC      ') '
&AL1      SETA      &AL1+1
          DC          &CG4&AL2&CG2&SYSLIST(&AL1)&CG3&CG5
          AGCB       .A
.E         ANOP
&CG5      SETC      'AL1('
&AL1      SETA      &AL1+1
          DC          &CG2&CG4&CG3
          DC          &CG5&SYSLIST(&AL1)&CG3
&AL1      SETA      &AL1+1
          AGCB       .A
          MEND

```

WATE (Wait)

*FUNCTION

*

*WAITS FOR AN EVENT TO OCCUR, THEN FLOW OF CONTROL CONTINUES.

*IF AN EVENT HAS ALREADY OCCURRED (SEE 'SET'), THEN THE FLOW OF

*CONTROL CONTINUES UNINTERRUPTED

*

*

*

*MACRO DEFINITION

*

```

          MACRO
&LABEL    WATE      &CNTX=I,&PSG=0
          AIF        ('&CNTX' EQ 'F').A
&LABEL    LA         R5,&PSG
          AGC        .B
.A         ANOP
&LABEL    L          R5,&PSG
.B         TM        0(R5),X'80'
          BZ         GW&SYSNDX
          NI         0(R5),X'7F'
          B          GW&SYSNDX+2
GW&SYSNDX SVC        WAIT
          MEND

```

*

REFERENCES

1. Groner, G. F., "Real-Time Recognition of Handprinted Text," *AFIPS Conference Proceedings* (1966 FJCC), Vol. 29, Spartan Books, Inc., Baltimore, Maryland, 1966, pp. 591-601. (Also, The RAND Corporation, RM-5016-ARPA, October 1966.)
2. Ellis, T. O., and W. L. Sibley, *On the Problem of Directness in Computer Graphics*, The RAND Corporation, P-3697, March 1968.

3. Brown, G. D., and C. H. Bush, *The Integrated Graphics System for the IBM 2250*, The RAND Corporation, RM-5531-ARPA (in process).
4. Davis, M. R., and T. O. Ellis, "The RAND Tablet: A Man-Machine Graphical Communication Device," *AFIPS Conference Proceedings* (1964 FJCC), Vol. 26, Part 1, Spartan Books, Inc., Baltimore, Maryland, 1964, pp. 325-331. (Also, The RAND Corporation, RM-4122-ARPA, August 1964.)