REAL TIME RECOGNITION
OF HANDPRINTED TEXT:
PROGRAM DOCUMENTATION

G. F. Groner

PREPARED FOR:
ADVANCED RESEARCH PROJECTS AGENCY
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.
-iii-

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
    MMXNC ............................................................ 69
    MMNMS ........................................................... 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            87
  Processing ................................
REC Program Listing ....................... 88
INTERP ................................ .... 99
INTERP Function ................................99
INTERP Call ................................ 99
INTERP Sequence of Information         100
  Processing ................................
INTERP Program Listing .................... 101
REC RCS'S ................................... 121
AHSTR1 ....................................... 121
BFI .......................................... 124
BWHITE ....................................... 125
BSDP ......................................... 126
BSMNW ........................................ 128
BSRPRM ...................................... 130
BSSM ......................................... 132
B SVM ......................................... 134
BTEST1 ....................................... 136
BTEST3 ....................................... 137
KNYTST ....................................... 139
KNYT ......................................... 141
KVXYT ....................................... 143
MWT .......................................... 145
PTEST ........................................ 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
<table>
<thead>
<tr>
<th>Term</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARL (Parallel)</td>
<td>168</td>
</tr>
<tr>
<td>PAWS (Pause)</td>
<td>169</td>
</tr>
<tr>
<td>PROCS (Process)</td>
<td>170</td>
</tr>
<tr>
<td>PROLG (Prologue)</td>
<td>170</td>
</tr>
<tr>
<td>RCS (Remote Code Sequence)</td>
<td>171</td>
</tr>
<tr>
<td>REGS (Registers)</td>
<td>172</td>
</tr>
<tr>
<td>SET</td>
<td>173</td>
</tr>
<tr>
<td>SVCS</td>
<td>173</td>
</tr>
<tr>
<td>TABLE</td>
<td>174</td>
</tr>
<tr>
<td>WATE (Wait)</td>
<td>175</td>
</tr>
</tbody>
</table>

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 oper- tional 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 com- puter 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 indi- vidual printing styles. It makes use of size and position information to differentiate among symbols not distinguish- able 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 CHARREC, 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. CHARREC 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 CHARREC. 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 CHARREC and REC. The processes and RCS's called by CHARREC 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 CHARREC, 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 CHARREC, 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†

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

†In addition to those italicized above, other words and phrases used throughout the text are also defined.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>character</td>
<td>A sequence of input coordinates encoded as an entity by this program; same as &quot;symbol&quot; (see The Symbols Recognized, Sec. II).</td>
</tr>
<tr>
<td>character code</td>
<td>A 1-byte encoding of a character (see CRT Display Character Codes, Appendix, p. 162).</td>
</tr>
<tr>
<td>context</td>
<td>1) a continuous storage block consisting of linkages between parent (calling) and daughter (called) processes, formal parameters, and other information; 2) the environment used to interpret the meaning of an action or inputs.</td>
</tr>
<tr>
<td>data</td>
<td>1) (x, y) coordinates; 2) indicators or computed quantities used by the program.</td>
</tr>
<tr>
<td>daughter process</td>
<td>A process called by a parent process.</td>
</tr>
<tr>
<td>display</td>
<td>A programmed output device that presents an image.</td>
</tr>
<tr>
<td>display stream</td>
<td>The sequence of instructions controlling the display.</td>
</tr>
<tr>
<td>EEXIT</td>
<td>Appears in a call to a process or RCS; EXIT is a re-entry point in the calling (parent) process corresponding to a return from the called (daughter) process or RCS; the label &quot;EXIT&quot; is user determined.</td>
</tr>
<tr>
<td>ending point</td>
<td>The (x, y) position at which the writing stylus micro switch is opened when terminating a stroke.</td>
</tr>
<tr>
<td>entry point</td>
<td>The place at which control resumes.</td>
</tr>
</tbody>
</table>
| **F**   | 1) full computer word (32 bits);  
| feature | 2) formal (input/output) parameter. |
| formal parameter | A computed attribute of a symbol which is used for identification. |
| FPARAM | In a call to a process, refers to the formal (input/output) parameter PARAM of the calling (parent) process; the label "PARAM" is user determined. |
| geometric corner | A detected sharp change (90° or more) in the direction of the writing stylus motion. |
| GPARAM | A reference to the parameter PARAM in a call to a process. G = F for a formal (input or output) parameter of the calling process; G = I for an informal (local) parameter; the label "PARAM" is user determined. |
| **H**   | Computer halfword (16 bits). |
| informal parameter | Temporary or constant data defined within a process. |
| ink     | 1) same as "ink track"; 2) the action of generating an ink track. |
| ink track | A displayed string of vectors that approximates the writing stylus motion. |
| instance | The appearance of a calling sequence to a process in the program. |
| IPARAM | In a call to a process, refers to the informal (local) parameter PARAM of |
the calling (parent) process; the label "PARAM" is user determined.

NAMEA

In a call 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 "cannot interpret."

parameter

Temporary or constant data.

parallel task

An instruction sequence initiating two lines of control within the program.

parent process

The process that called a daughter process.

pen

The writing instrument that is moved on the Tablet writing surface; same as "stylus."

pendown

Closure of the writing stylus micro switch due to a downward force.

penup

Opening of the writing stylus 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 "reentrant process").

raster unit

1/1024 of the Tablet or display surface dimension--0.01 in. in the case of a standard 10.24 by 10.24-in. Tablet.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw data point</td>
<td>A writing stylus coordinate pair as received from the input device.</td>
</tr>
<tr>
<td>read-only</td>
<td>Computer storage that is read (and executed if code) but not modified.</td>
</tr>
<tr>
<td>reentrant process</td>
<td>A process requiring separate linkage and data storage blocks for each usage, but only a single storage block of read-only code. When executed, the code is not modified and therefore may be re-used even if the process has been suspended before completion.</td>
</tr>
<tr>
<td>RCS</td>
<td>Remote code sequence.</td>
</tr>
<tr>
<td>remote code sequence</td>
<td>A process with general-purpose register input/output operating in the environment of the calling (parent) process context; has no context but is reentrant.</td>
</tr>
<tr>
<td>starting point</td>
<td>The (x,y) position at which the writing stylus micro switch is closed when initiating a stroke.</td>
</tr>
<tr>
<td>stroke</td>
<td>The sequence of (x,y) coordinates between closing and opening the writing stylus micro switch.</td>
</tr>
<tr>
<td>stylus</td>
<td>The writing instrument that is moved on the tablet writing surface; same as &quot;pen.&quot;</td>
</tr>
<tr>
<td>subcharacter</td>
<td>A set of (x,y) coordinates encoded internally by the program, but which may not be a complete character and has not been outputted by the program.</td>
</tr>
</tbody>
</table>
symbol  A sequence of input coordinates encoded as an entity by this program; same as "character" (see The Symbols Recognized, in Sec. II).

tablet  An input device comprising a pen-like writing instrument and a writing surface [4]; as the stylus is moved over the surface its $x,y$ coordinates are sent to the computer for processing.

task  A sequence of instructions initiating lines of control (see "parallel task").

time-pause corner  A detected deceleration-acceleration of the writing stylus motion.

track  1) same as "ink track"; 2) the action of generating an ink track.

vector  A line segment described by its length (2, 4, 6, or 8 raster units) and direction (1 of 16 in $22.5^\circ$ 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 CHARC.

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, CHARC (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 re-
duction (thinning). When a data point arrives, its posi-
tion 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 noti-
fied (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

CHARREC 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 B, 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 CHARREC (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 SNLCl 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 ACHRCC,CHRCA,GDATA,GCHPSG,GINDEX,EFIXN,ENCHARX,ECHARX,EXTX
* TN,EXTC
*
*WHERE ALL THE LABELS ARE SELECTED BY THE USER
* ACHRCC 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')
* CHPSSG 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
*PNU A(PEN UP DATA), EQU MCH, NOT USED
*INBP 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,LY,Y,4S,V1,V2,V3,...,00 WHERE EACH SYMBOL
* BETWEEN COMMAS IS 1 BYTE; (LX,X) IS LOAD X, (LY,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)

* STROKE IN THE CHARACTER. 12-BIT X, 12-BIT Y, 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 CCDE) 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=CLKCH HAS BEEN CALLED, 6=TAKE HALT EXIT, 7=NOT FIRST PENDOWN
*INKIND  NO. BYTES OF INK
*PGUAD  CODE (NE=00, NW=01, SW=10, SE=11) FOR QUADRANT OF PREVIOUS DIREC-
*       TION (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, QMP, MXMNC, BSRPRM, BHITE)
*ASPR  ASPECT RATIO = 4*DCY/DCX (INTERP)
*NT  NO. CF THINNED POINTS (TCRNR)
*NTP  NT AT WHICH LAST TIME-CORNER OCCURRED (TCRN)
*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)
*          
*          3  2  1  0
*          
*          XLC   7  6  5  4  XRC
*          11  10  9  8
*          15  14  13  12
*          YBC

*WIDTH, HEIGHT  EXPECTED NORMAL CHARACTER WIDTH, HEIGHT--SEE BCX IN
*               INPUTS LIST (PSTEST)
*YCENT  Y COORDINATE OF CENTER OF PREVIOUSLY OUTPUTED CHARACTER
*PCHAR  CODE FOR PREVIOUSLY OUTPUTED CHARACTER
*CUSP  TEMPORARY STORAGE (TCRNR, REC, INTERP)
*NCUSP  NO. GEOMETRIC CORNERS (INTERP)
*NPTS  NO. RAW DATA PTS. SINCE LAST THINNED PT. (TCRNR)
*DEL  MINIMUM X OR Y DISTANCE BETWEEN THINNED POINTS (DERIVED FROM
*      INC--SEE INPLIS). (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, CDT, 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, BSMV)
*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,CO. GOES
* INTO INK BUFFER(SEE INKB IN LIST OF INPUTS)
*XL,YL  RAW DATA POINT COORDINATES
*XLC,YLC X,YL USED BY TRAVEC (CORNER)
*AX,AX1,AX2,AX3  16-DIRECTIONS USED FOR GEOMETRIC CORNERS (CORNER)
*AX01,AXO2,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,PTEST,TILD)
*DXS,DYS X,Y EXTENTS OF CURRENT STROKE (XMNS)
*XRS,XLS RIGHT,LEFT EXTREMES OF CURRENT STROKE (XMNC,XMNS)
*YTS,YBS TOP,BOTTOM EXTREMES OF CURRENT STROKE (XMNC,XMNS)
*CENT  X CENTER,Y CENTER--SEE CET IN OUTPUT LIST (RAZE,PTEST)
*MVC  ADJUSTABLE MVC INSTRUCTION
*TURN  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=BSRPXM)
*DO THRU D15 NO. OF OCCURREANCES OF DIRECTIONS 0 THRU 15 (SYMT)
*DN  SUM OF CO 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)
*NXMAX,NXMIN 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 (QM,R,REC,INTERP)
*CQYMAX,CQYMIN SEQUENCE OF CODE(YC,CO,01,02,03,YBC) QUANTIZED YMAX,
  YMIN--1 BYTE EACH (QM,R,REC,INTERP)
  ALSO USED AS AN INDICATOR(RELM)
*PYMX,PYNX X LOCATION OF PREVIOUS RELATIVE Y MAX,MIN (RELM)
*YMAX,YMINX SEQUENCE OF X LOCATIONS OF RELATIVE Y MAX,MIN FOR THE
  CURRENT STROKE--1/2 WORD EACH (RELM,INTERP,BSMNX,BTEST3)

**CHAREC** Read-Only Constants

*TIME  PEN-UP-DELAY TIME FOR CLOCK, F'0100' = 0.1 SECOND
*LXYJ  LOAD X, CO, LOAD AND JUMP TO Y,CO
*CODT  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 CISPLOAY 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
**TCRNK** 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.
**MXMS** 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 HIALT 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 STRCKE 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 STRCKE, GC 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****
*
**MVMNC* 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
*STCRE 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 NCT GTR 8 GO TO REC CALL
*IF NO. DIRECTIONS GTR 12, OR CHARACTER IS LARGE, SET CHAR=SCRUB, GC 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, GC 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, GC 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, GC TO WAIT FOR NEXT DATA POINT GROUP.
*IF CHARACTER IS NOT RECOGNIZABLE, GC TO NO CHARACTER ENTRY
*IF REC HAS NOT BEEN REQUESTED, GC 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, GC 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, GC 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
SVGS
REGS
CD1 DSECT
XR1 CS 3F
AREC CS 1F
CLK1 CS 1F
DATA CS 1F
WAITBX CS 1F
INDEX CS 1F
FINX EQU 0
NCHAX EQU 4
CHARX EQU 8
XTN EQU 12
XTC EQU 16
CD4 DSECT
XR4 CS 0F
ICP CS 1F
MCH CS 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 CS 1F
IND CS 1C
CHARA CS 1C
AR DS 1H
BCX CS 1F
CC3 DSECT
XR3 CS 0F
I1 CS 1F
PAD CS 1F
CODE CS 1F
XS CS 1H
YS CS 1H
XT CS 1H
```

**TIME/DATA EXPIRATION INDEX**

TIME/Terminal NC CHAR EXIT
DATA/Terminal CHAR EXIT

**INK DATA**

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
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>YT</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>DX</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>DY</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>MDX</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>MDY</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>PANG</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>PACANG</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>N</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>SN</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>PUP</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>INKIND</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>PQUAD</td>
<td></td>
<td>1H</td>
</tr>
<tr>
<td>BR56</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>DXC</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>DYC</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>XRC</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>XLC</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>YTC</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>YBC</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>ASPR</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>NT</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>NTC</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>INKC</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>XYE</td>
<td>CS</td>
<td>10C</td>
</tr>
<tr>
<td>XYS</td>
<td>DS</td>
<td>10C</td>
</tr>
<tr>
<td>WIDTH</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>YCENT</td>
<td>CS</td>
<td>2F</td>
</tr>
<tr>
<td>PCHAR</td>
<td>CS</td>
<td>1C</td>
</tr>
<tr>
<td>CUSP</td>
<td>CS</td>
<td>1C</td>
</tr>
<tr>
<td>NCUSP</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>NPTS</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>CEL</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>P</td>
<td>CS</td>
<td>1C</td>
</tr>
<tr>
<td>CHAR</td>
<td>DS</td>
<td>1C</td>
</tr>
<tr>
<td>TEMP</td>
<td>CS</td>
<td>1C</td>
</tr>
<tr>
<td>TINK</td>
<td>CS</td>
<td>5C</td>
</tr>
<tr>
<td>XSP</td>
<td>DS</td>
<td>10C</td>
</tr>
<tr>
<td>YSP</td>
<td>CS</td>
<td>10C</td>
</tr>
<tr>
<td>XEP</td>
<td>CS</td>
<td>10C</td>
</tr>
<tr>
<td>YEP</td>
<td>CS</td>
<td>10C</td>
</tr>
<tr>
<td>ALXYJ</td>
<td>CS</td>
<td>8C</td>
</tr>
<tr>
<td>XL</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>YL</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>XLO</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>YLC</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>AX3</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>AX2</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>AX1</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>AX</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>AX23</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>AX12</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>AX01</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>AX02</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>NC</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>C</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>CYM</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>DXS</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>CYS</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>XRS</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>XLS</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>YTS</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>YBS</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>CENT</td>
<td>DS</td>
<td>1F</td>
</tr>
<tr>
<td>MVC</td>
<td>CS</td>
<td>6C</td>
</tr>
<tr>
<td>TTURN</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>TURN</td>
<td>DS</td>
<td>1F</td>
</tr>
<tr>
<td>XC</td>
<td>CS</td>
<td>10C</td>
</tr>
<tr>
<td>YC</td>
<td>CS</td>
<td>10C</td>
</tr>
<tr>
<td>D0</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>D1</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>D2</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>D3</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>D4</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>D5</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>D6</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>D7</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>D8</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>D9</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>D10</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>D11</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>D12</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>D13</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>D14</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>D15</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>CN</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>NTCUSP</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>PNPTS</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>PYMAX</td>
<td>CS</td>
<td>1H</td>
</tr>
<tr>
<td>PYMIN</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>NYMAX</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>NYMIN</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>YMAX</td>
<td>DS</td>
<td>10H</td>
</tr>
<tr>
<td>YMIN</td>
<td>EQU  YMAX+1C</td>
<td></td>
</tr>
<tr>
<td>CYMAX</td>
<td>CS</td>
<td>10C</td>
</tr>
<tr>
<td>QYMIN</td>
<td>EQU  QYMAX+5</td>
<td></td>
</tr>
<tr>
<td>PYMXX</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>PYMNX</td>
<td>DS</td>
<td>1H</td>
</tr>
<tr>
<td>YMXX</td>
<td>DS</td>
<td>10H</td>
</tr>
<tr>
<td>YMINX</td>
<td>EQU  YMXX+10</td>
<td></td>
</tr>
<tr>
<td>CHAREC</td>
<td>PROC S CLEAR=5,CNTX=9,AUTO=86,PRCLG=XCHRX,1D=9000021F</td>
<td></td>
</tr>
</tbody>
</table>
TIME   CC       F'0300'
LXYJ  DC       X'54000600'
CDOT  CC       X'80'
HEX10 CC       F'16'
HEX90 CC       F'144'
MOVER MVC O(C,R6),C(R7)
ANG56 CC       V(FN56)
ANG4A DC       V(AV4)
DELT CC       V(CELTAS)
RECA CC       V(REC)
       DC       X'8000C21C'
SMTH CC       V(SMOOTH)
THINN CC       V(TIN)
MAXMNS DC       V(MAXMNS)
HYSTR CC       V(HYST)
CLK2 DC       V(CLK2)
       DC       X'800000C0'
MAXMNC DC       V(MAXMNC)
TRAVC CC       V(TRAVC)
CORNR  DC       V(CORNER)
TURNA DC       V(TURNER)
CHECKA DC       V(CHECK)
RELMA CC       V(RELM)
QMA DC       V(QMM)
COTA DC       V(COT)
TCRNRDC DC       V(TCRNR)
RAZEA CC       V(RAZE)
XCHRX  PROCLG
*

****START****
*
*
CLEAR PSG=WAITBX,CNTX=F
MVI ALXYJ,X'CO'
MVI ALXYJ+5,X'40'
MVI ALXYJ+6,X'00'
MVC MVC(6)+MOVER
BAL R15,TOP
B WAITZ
*
*
****NEW CHARACTER ENTRY****
*
*
TOP  XC       IL(4),IL
     XC       PAD(4),PAD
     XC       SN(2),SN
# STRCKES
     XC       INKIND(2),INKIND
     XC       INKC(2),INKC
     XC       PUP(2),PUP
```
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,BCX
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

****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

****NEW STROKE****

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

STORE PREV CENT Y IN YCENT
STORE PREV CHAR IN PCHAR
STORE MAX ALLOW DY IN DYM
1 1/2 CHARACTER HEIGHT
ENTRY FROM WAIT BOX
B TO FIN IF HALT
B TO IND2 IF P.U.
R12=INK BUFF IND
# STROKES OVERFLOW TEST
```
SLL R8,1
LA R6,1(R6)
STH R6,SN

*INITIALIZATION
XC NG(4),NC
LA R6,16
STH R6,A1
STH R6,AX
MVC AX3(4),A1
XC NTCUSP(2),NTCUSP
XC NPTS(2),NPTS
XC NT(2),NT
XC NTC(2),NTC
LA R7,20
STH R7,NPTS
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,Y8S
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,XY
STH R7,PYMX
STH R7,PYMX
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
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
CR 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 017,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

*

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

*PENDWN SR R13, R13

ENTER SR R6, R6
CR R12, R6
BC 8, CLNBUF

ENTER1 L R4, DATA
L R10, INPB
LM R6, 0(R13, R10)
STH R6, XS
LA R13, 2(R13)
LM R6, 0(R13, R10)
STH R6, YS
LM R8, NPTS
LA R8, 1(C, R8)
STH R8, NPTS
LM R7, YT
YM THIN THIN TRACK
LM R8, DEL
YM MIN THIN DIFF
RCS THINN, E*+4
CH R7, YT
BC 8, YSMALL
B OK

YSMALL LM R6, XS
LM R7, XT
LM R8, DEL
LM MIN THIN DIFF
RCS THINN, E*+4
CH R7, XT
BC 8, SMALL

*HERE IF NEW POINT ACCEPTED IN THIN TRACK
CK EQU *
RCS TCRNRA, III, E*+4
LM R11, YS
LM R10, YT
STH R11, YT
SR R11, R10
STH R11, DY
LM ST DELTA Y
LPR R11, R11
STH R11, MDY
LM ST MAG(DELTA Y)
LM R11, XS
LM R7, XT
STH R11, XT
SR R11, R7
STH R11, DX
LM ST DELTA X
LPR R11, R11
STH R11, MDX
LM ST MAG(DELTA X)

NUINK LM R7, XT
LM R9, YT
LM R10, XL
LM R11, YL
STH R10, XLO
LM STORE NEW INK
STH R11, YLC
L R4, DATA
L R15, INC
SRL R15, 26
LA R14, 3
NR R15, R14
LA R15, 1(R15)
RKS TRAHC, E++4
C R0, HEX10
BC 8, INKST
LR R8, RC
SLL R8, 1
LH R7, D0(R8)
LA R7, 1(R7)
STH R7, DC(R8)
LH R7, DN
LA R7, 1(R7)
STH R7, DN
STH R0, AX
STH R10, XL
STH R11, YL
L R4, DATA
LH R15, INK1
BCT R15, A1

A1
L R4, DATA
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

STOINK EQU *
L R4, DATA
L R7, ICP
LH R8, 6(R7)
LA R8, 1(R8)
STH R8, 6(R7)
L R4, DATA
C R0, HEX9C
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,AXC1
CLI AX3+1,X*10'
BC 8,SHIFT
RCS CORNR,III,1,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,III,E*+4
*UPDATE relative max and mins
RCS RELMA,III,E*+4
*
*
****ANGLE SECTION START****
*
*
* DETERMINE QUADRANT *
*
LH R6,DX
LTR R6,R6
BC 4,DXNEG
LH R6,DY
LTR R6,R6
BC 4,CYNEG

*DX,DY POS, QUAD=0
SR R6,R6
B QTEST

*DX POS, DY NEG, QUAD=3
CYNEG LA R6,3
B QTEST
DXNEG LH R6,DY
LTR R6,R6
BC 4,CYNEG
LA R6,1
B QTEST

CYNEG LA R6,2
B QTEST

*
* DETERMINE DIRECTION *
* AND CHECK FOR 2 EQUAL SUCCESSIVE ANGLES *
*
QTEST CH R6,PQUAD
BC 8,CEQPG
STH R6,PQUAD
SET PQUAD=QUAD
LH R6,MCX
CH R6,MDY
BC 4,ODCANG
B IF MDX LESS THAN MDY
EVANG LH R6,DX
LTR R6,R6
ANG EVEN, TEST SIGN(DX)
BC 4, ANG2
SR R6, R6
B PRVANG

ANG2
LA R6, 2
B PRVANG

ODDANG
LH R6, DY
LTR R6, R6
BC 4, ANG3
LA R6, 1
B PRVANG

ANG3
LA R6, 3
B PRVANG

QEQPG
STH R6, PQUAD
LH R6, PANG
LA R7, 1
NR R6, R7
BC 8, EVPANG

*PREV ANGLE ODD
LH R6, MDX
LH R7, MDY
RCS HYSTR, E++4
BC 2, EVPANG
B PRVTST

EVPANG
LH R6, MDY
LH R7, MDY
RCS HYSTR, E++4
BC 2, ODDANG
B PRVTST

PRVANG
CH R6, PANG
BC 8, PRVTST
LR R9, R6
RCS TURN, III, E++4
LR R6, R9
STH R6, PANG
B SMALL

PRVTST
EQU *
LH R6, PANG
CH R6, PACANG
BC 8, SMALL
STH R6, PACANG

** ANGLES OVERFLOW TEST
LH R7, N
LA R8, 15
CR R7, R8
BC 4, NLOW
MVI CHAR, X'72'
SR R7, R7

NLOW
L R10, CODE
LH R9, TTURN
L R11, TTURN
LA R8, 16
SR  R8,R7
SHFT
SRL R10,2
SRL R11,2
BCR R8,SHFT
SLL R10,2
SLL R11,2
LR  R6,PANG
CR  R10,R6
CR  R11,R9
LA  R8,15
SR  R8,R7
SHFT1
SLL R10,2
SLL R11,2
BCR R8,SHFT1
ST  R10,CODE
ST  R11,TURN
MVI TTUR+1,X'00'
LA  R7,1(R7)
STH R7,N
B  SMALL

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

SMALL

LA  R13,2(C,R13)
HERE PROCESSING OF NEW RAW COMP
L  R4,DATA
LH R10,INPL
CR  R13,R10
BC  4,ENTER
GET NEXT POINT
STH R12,INKIND
KEEP INK BUFF IND

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

WATRL

WATE PSG=WAIT8X,CNTX=F
L  R4,DATA
TM  IND,X'10'
BC  1,GOFINX
L  R6,INKB
IC  R15,INC
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,CEL
NFIN  NI  PUP+1,X'C7'

-40-
B     WAITZ

****PEN UP SIGNAL****

IND2       NI  PUP+1,X*FE*  HERE CN PU TRAP
           LH  R7,ON
           LTR R7,R7
           BC  8,NOSANG
           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,CCDE
                  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*C2*
           BC  6,PTEST
           RCS DOTA,ILL,ECASE1,EPTEST

*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*CC*
          BC  8,TBIG

*CAN OLD CHAR BE COMBINED WITH STROKE
           NCTCK RCS ANG4A,ILL,E**4
RCS  CHECKALL, III, ETCBIG, ECOMBK
*OLD CHAR CAN BE COMBINED WITH THIS STROKE, TEST FOR COMMA
COMBK  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, COMOK1 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 ENDPT 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, TSTST2
CLI  P, X'01'
BC  8, CASE1A
TSTST2  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'
      BC   8,CASE1B
      CLI   CHAR,X'CD'
      BC   8,CASE1B
      CLI   CHAR,X'CD'
      BC   6,TOBIG

*TEST FOR SHORT VERT SECOND STROKE
CASE1B CLI   N+1,X'01'
      BC   6,CASE1D
      TM   CODE,X'CO'
      BC   12,CASE1D
      NOT SINGLE ANGLE
      NOT VERT

CASE1E LH   R15,0YM
      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,R1C
      R =3/4 WIDTH
      LH   R7,XRC
      AH   R7,XLC
      2 OLD CENTER
      LH   R8,XRS
      AH   R8,XLS
      2 NEW CNT
      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
      CR   R8,R15
      R1 = WIDTH
      BC   2,TOBIG
*IS NEW XCENT IN LEFTMOST 1/4 OF A GRID POS?
LH R15,WIDTH
SRL R15,3
SR R8,R8
SRL R9,3
DR R8,R15
LR R10,R15
SRL R10,2
CR R8,R10
BC 4,CASE1
CHAR WIDTH IN RASTERS
NEW XCENT IN RASTERS
NEW XCENT MOD(WIDTH)
1/4 WIDTH
REMAINDER IN R8

*IS OLD XCENT IN RIGHTMOST 1/4 OF A GRID POS
SR R6,R6
SRL R7,3
DR R6,R15
SR R15,R10
CR R6,R15
LA R10,11
CR R6,R10
BC 2,CASE1
BC 12,TOBIG
OLD XCENT IN RASTERS
OLD XCENT MOD(WIDTH)
3/4 WIDTH
REMAINDER IN R6

*2ND STROKE IS NOT VERTICAL
*IS DIFF BETWEEN 2ND STROKE LEFT AND 1ST STROKE RIGHT GTR R RASTERS
CASE1ID 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-STROKE****
* *
CASE1 RCS MAXMNC,III,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  # CORNERS
NTC1  STH R8, NCUSP
NTX   EQU *
*NO. OF TIME CORNERS
LH    R8, NT
BCT   R8, NT1
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, RECRNT
CLI   CHAR, X*91*
BC    6, TSTSCB
```
*SCRIPT J
  RCS RAZEA,III,ERE C RTN
  TST SCB EQU *
  CLI CHAR,X'72'
  TEST FOR SCRUB (N GTR 15)
  BC 8,REC RTN
  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
  OTHERWISE ALLOW FOR A POSSIBLE SCRIPT CHARACTER
  CLI N+1,X'0C'
  BC 2,SCBX
  LH R8,DYM
  CH R8,DC
  BC 4,SCBX DYM
  CH R8,DC
  BC 10,CALREC
  SCBX EQU *
  MVI CHAR,X'72'
  B RE C RTN
  *
  ****REC CALL****
  *
  CALREC INST AREC,RECA,III,III,ERE C RTN
  *
  ****MULTI-CHARACTERS****
  *
  TOBIG LH R7,INKC
  LH R8,IN K IND
  CR R8,R7
  BNH OVR2
  SR R8,R7
  STH R8,INK INC
  L R4,D AT A
  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,D AT A
  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  CI  PUP+1,X'30'

****2 CHARACTERS****

ALPHA  L  R4,DATA  TEST FOR NO RECOGNITION
TM  IND,X'40'
BC  8,WATRI
CLI  CHAR,X'EF'  CK IF CHAR OR NO CHAR
BC  8,NEXIT  NO CHAR
TM  PUP+1,X'20'
BC  8,TCE  NO MORE DATA TAKE TERMINAL
PARL  CNTX=F,LCW=PHI,HIGH=CHARX

****RESET FOR NEW CHARACTER****

RESET CHAR SIZE, LOCATION, ETC.

PHI  MVC  DXC(12),DXS
LH  R6,SN
BCT  R6,DECX6
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  
RECRTN EQU *  
*IF N GTR 8 AND CHAR IS NOT A SCRIPT CHARACTER, SET CHAR=SCRUB  
CLI CHAR,X'AO'  
BC 2,SCBX2  
CLI CHAR,X'81'  
BC 10,SCRPT  
SCBX2 EQU *  
CLI N+1,X'08'  
BC 12,RCRTN1  
MVI CHAR,X'72'  
B RCRTN1  
SCRPT 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,CLEXF  
NOSKIP EQU *  
GI 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, FINISH
BAL R15, OUTPTS
TM PUP+1, X'10'
BC 1, ALPHA
L R4, INDEX
CLI 3(R4), X'00'
BC 8, WAIITZ
MVI PUP+1, X'00'
8 ALPHA

****FINISH ENTRY 1****

FIN
TM PUP+1, X'04'
BC 8, FINISH
CI PUP+1, X'02'
BC 15, SETCK

****SET UP OUTPUTS****

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

****NO CHARACTER****

NCEXT EQU *
TM PUP+1, X'20'
BC 8, TNE
PARL CNTX=F, LGW=PHI, HIGH=NCHARX
****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,III,EEXIT
*WHERE II IS AT THE TCP OF CHAREC'S INTERNAL PARAMETER LIST
*
*
*INPUT REGISTER. R6
*
*INTERNAL REGISTERS. R7,R8,R10
*
*
USING XR6,R6
REGS
EXO EQU 0
C6 DSECT
XR6 CS OF
I1 DS 1F
DS 1F
CODE DS 1F
DS 5F
N CS 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: I1+3(1),CODE
- TR: I1+3(1),THET4

### ANGE
- BEXIT: EX0

### THET4
- CS: OH
- CC: X*CC CCCC SBARM
- CC: 15C*C ILLEGAL
- CC: 2X*13 0100-0101 DK
- CC: X*40 0102 RSC
- CC: X*3F 0103 SCRPT
- CC: C*0 0110 ILLEGAL
- CC: X*13 0111 DK
- CC: 2C*0 ILLEGAL
- CC: X*13 0120 DK
- CC: X*21 0121 S5
- CC: X*13 0122 DK
- CC: X*3E 0123 S09M
- CC: X*30 0130 S9LC1
- CC: X*41 0131 SCPFP
- CC: X*43 0132 RSV
- CC: X*42 0133 SCPEL
- CC: 2X*0D 0200,0201 S2MRZ
- CC: X*2A 0202 S3SCR B
- CC: 2X*0F 0203,0210 S3MRB
- CC: 3X*0D 0211-0213 S2MRZ
- CC: 2C*0 ILLEGAL
- CC: X*2D 0222 SLKRTM
- CC: C*0
- CC: X*02 0230 S23MB
- CC: X*0D 0231 S2MRZ
- CC: X*0E 0232 S3MB
- CC: X*14 0233 S7MGK
- CC: X*0D 0300 S2MRZ
- CC: X*0E 0301 S8
- CC: X*0F 0302 S3MB
- CC: X*44 0303 S3MBR
- CC: X*44 0310 RSV
- CC: X*49 0311 S8LCV
- CC: X*40 0312 RSC
- CC: X*01 0313 STPM
- CC: X*02 0320 S23MB
- CC: X*3B 0321 S023MB
- CC: X*12 C322 SRPRM
- CC: X*0D 0323 S2MRZ
- CC: 3C*0
- CC: X*14 0333 S7MGK
- CC: X*1E 1000 SFE
-53-

CC 3C'0' 1010 DK
CC X'13' 1011 DK, POSSIBLY TILDA
CC X'13' 1012 DK
CC X'13' 1013 DK
CC X'02' 1020 S23MB
CC X'CE' 1021 S3MB
CC X'13' 1022 DK
CC X'02' 1023 S23MB
CC X'0F' 1030 S3MBR
CC X'45' 1031 SCPNRZ
CC X'03' 1032 S23MBP
CC X'38' 1033 SA7
CC 5C'O' 1111 S1MAK
CC 1OC'O' 1200,1201 DK
CC 2X'13' 1202,1203 SSM
CC 4X'13' 1210-1213 DK
CC 2C'O' 1222 DK
CC C'0' 1230 SMC
CC X'04' 1231 S8
CC X'32' 1232 SS8M
CC X'05' 1233 STPA
CC X'15' 1300 S24
CC X'11' 1301 SNMA
CC 2X'17' 1302,1303 S3
CC X'0D' 1310 S2MRZ
CC X'17' 1311 SNMA
CC X'36' 1312 SASTAR
CC X'18' 1313 SMLC
CC X'36' 1320 SASTAR
CC X'0D' 1321 S2MRZ
CC X'13' 1322 DK
CC X'46' 1323 RSZ
CC 3C'O' 1333 SCQMMAM
CC X'1A' 2000 SBARMK
CC 3C'O' 2010 SG
CC X'33' 2011 SG069M
CC X'06' 2012 SG06M
CC X'08' 2013 S9
CC X'34' 2020 SGSCRB
CC X'29' 2021 SSM
CC X'30' 2022 SG06M
CC X'47' 2023 SE
CC X'4A' 2030 SEQ
CC X'48' 2031 SCPGQ
| CC  | X'05' | 2032 SS8M |
| CC  | X'1C' | 2033 S9MK |
| CC  | 3X'13' | 2100-2102 DK |
| CC  | X'3C' | 2103 S09 |
| CC  | C'C' | 2111 DK |
| CC  | 2C'O' | 2111 DK |
| CC  | 8X'13' | 2120-2133 DK |
| CC  | 10C'C' | 2222 SBARM |
| CC  | 5C'O' | 2300 SCMEG |
| CC  | X'06' | 2301 SG069M |
| CC  | X'2F' | 2302 SGS |
| CC  | X'07' | 2303 SS89M |
| CC  | X'22' | 2310 STP5 |
| CC  | X'32' | 2311 S8 |
| CC  | X'33' | 2312 SG |
| CC  | X'07' | 2313 SS89M |
| CC  | 2X'32' | 2320, 2321 S8 |
| CC  | X'18' | 2322 SSM |
| CC  | X'47' | 2323 SE |
| CC  | 3C'O' | 2333 SFE |
| CC  | X'1E' | 3000 SLMEK4 |
| CC  | 3C'O' | 3010 SG81 |
| CC  | X'20' | 3011 SUMJU |
| CC  | X'08' | 3012 SG06M |
| CC  | X'27' | 3013 SUMAM |
| CC  | 2X'22' | 3020, 3021 STP5 |
| CC  | X'10' | 3022 STP6 |
| CC  | 3X'22' | 3023-3031 STP5 |
| CC  | X'22' | 3032 STP5 |
| CC  | X'23' | 3033 SK5 |
| CC  | X'24' | 3100 STPH |
| CC  | X'13' | 3101 DK |
| CC  | X'2C' | 3102 SBDPR1 |
| CC  | X'09' | 3103 SBDPR |
| CC  | C'C' | 3111 SVM |
| CC  | X'25' | 3120, 3121 STPH |
| CC  | 2C'O' | 3122 SDMNH |
| CC  | X'39' | 3123 SCG |
| CC  | X'2B' | 3130 BR |
| CC  | X'0A' | 3131 SMNW |
| CC  | X'2C' | 3132 SBDPR1 |
| CC  | X'27' | 3133 SUMAM |
| CC  | 4X'13' | 3200-3203 DK |
| CC  | X'3C' | 3210 S09 |
CC X'28' 3211 STPJ
CC 2X'34' 3212, 3213 S9
CC 2C'O' 3222 SRPRMJ
CC C'O' 3230 SCC
CC X'37' 3231 S8
CC X'32' 3232 S3
CC X'35' 3233 DK
CC 15C'O' 3333 SMBM
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, III, END, EYES
*WHERE III IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
*III+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 CS 1F
DS 2F
DS 11H
SN  CS  1H
CS  14H
CS  20C
CS  3F
CS  1H
DS  2C
DS  3H
P  
CS  1C
CHAR  CS  1C
CHECK  BX8
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 CHAR IS VERT
*CHANGE CHAR CODE TO 1
CK1  LA  R8,1
CK2  SR  R9,R9
SR  R1G,R1G
CK3  IC  R10,O(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
<table>
<thead>
<tr>
<th>LA</th>
<th>R7, S13</th>
<th>B12</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA</td>
<td>R7, S3</td>
<td>B13</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S1</td>
<td>B14</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S1</td>
<td>B15</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S2</td>
<td>B16</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S7</td>
<td></td>
</tr>
<tr>
<td>LA</td>
<td>R7, S3</td>
<td>B18</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B19</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S5</td>
<td>B20</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B21</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S6</td>
<td>B22</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S7</td>
<td>B23</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B24</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S8</td>
<td>B25</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S3</td>
<td>B26</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B27</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S1</td>
<td>B28</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S1</td>
<td>B29</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S14</td>
<td>B30</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S9</td>
<td>B31</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S10</td>
<td>B32</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B33</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B34</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S1</td>
<td>B35</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S1</td>
<td>B36</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S11</td>
<td>B37</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S1</td>
<td>B38</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S1</td>
<td>B39</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B40</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B41</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S1</td>
<td>B42</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B43</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S7</td>
<td>B44</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S1</td>
<td>B45</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S15</td>
<td>B46</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S10</td>
<td>B47</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S10</td>
<td>B48</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S10</td>
<td>B49</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B50</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B51</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B52</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B53</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B54</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B55</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B56</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B57</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B58 (3A)</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S2</td>
<td>B59 (3B)</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S2</td>
<td>B60 (3C)</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S4</td>
<td>B61 (3D)</td>
</tr>
<tr>
<td>LA</td>
<td>R7, S10</td>
<td>B62 (3E)</td>
</tr>
<tr>
<td>S1</td>
<td>DS</td>
<td>OH</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'01'</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'00'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S2</th>
<th>DS</th>
<th>OH</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CC</td>
<td>X'01'</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'00'</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'E0'</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'F0'</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'E4'</td>
<td>U</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'C3'</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'F6'</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'C7'</td>
<td>G</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'E2'</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'F5'</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'F8'</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S3</th>
<th>DS</th>
<th>OH</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CC</td>
<td>X'01'</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'E0'</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'00'</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S4</th>
<th>DS</th>
<th>OH</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CC</td>
<td>X'00'</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S5</th>
<th>DS</th>
<th>OH</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CC</td>
<td>X'01'</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'E0'</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'F0'</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'F6'</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'C3'</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'C7'</td>
<td>G</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'D2'</td>
<td>K</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'E3'</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'E5'</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'E7'</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'E8'</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'CE'</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'CD'</td>
<td>(</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'E1'</td>
<td>/</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>X'00'</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S6</th>
<th>DS</th>
<th>OH</th>
<th></th>
</tr>
</thead>
</table>
CC X'01* 1
CC X'E0* -
CC X'E7* X
CC X'EB* Y
CC X'E5* V
CC X'D2* K
CC X'F7* 7
CC X'CD* )
CC X'00*
S7 DS OH
CC X'E0* -
CC X'CD*
S8 DS OH
CC X'01* 1
CC X'E0* -
CC X'D2* K
CC X'FE* =
CC X'F0* C
CC X'F6* 6
CC X'F7* G
CC X'E7* X
CC X'E8* Y
CC X'E5* V
CC X'00*
S10 DS OH
CC X'F0* C
CC X'E4* U
CC X'C3* C
CC X'F6* 6
CC X'C7* G
S11 DS OH
CC X'01* 1
CC X'E5* V
CC X'D2* K
CC X'E7* X
CC X'E8* Y
CC X'F0* C
CC X'E4* L
CC X'C3* C
CC X'F6* 6
CC X'C7* G
CC X'00*
*VERTICAL STROKE
S12 DS OH
DC X'01* 1
DC X'70'  KARAT
CC X'F7'  7
CC X'D3'  L
CC X'D0'  (n
CC X'F1'  /
CC X'D1'  J
CC X'DD'  )
CC X'EE'  GTR
DC X'F0'  C
DC X'F6'  6
DC X'E4'  U
CC X'C2'  B
CC X'C4'  D
CC X'C7'  P
CC X'D9'  R
DC X'F8'  8
DC X'F9'  9
CC X'C1'  A
CC X'85'  LC E
DC X'86'  LC F
CC X'89'  LC I
CC X'93'  LC L
DC X'88'  LC H
CC X'C8'  H
CC X'CC'  *
CC X'00'  *
S14 DS 0H
DC X'01'  1
DC X'EO'  -
CC X'F7'  7
DC X'00'  *
S15 DS 0H
DC X'EO'  -
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 CORMERA,111,EEXIT
*WHERE 11 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
AX2    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 AXO1+1,X*04*
BC 4,AXP2
CLI AXO1+1,X*0C*
BC 2,AXP2

AXIEQ2
LH R7,AX12
STH R7,AX02
BC 15,IEQJ

AXP2
LH R7,AX
SH R7,AX2
LPR R7,R7
STH R7,AXC2
CLI AXC2+1,X*04*
BC 4,COUT
CLI AXO2+1,X*0C*
BC 2,COUT
LH R7,AX23
STH R7,AX02

IEQJ
CLI AXO2+1,X*01*
BC 12,SETC1
CLI AXO2+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,CEQ1

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,CEQ1

COUT
MVI C,X*00*

CEEXIT
BEXIT EX0
END
*FUNCTION
*
*USED WHEN THE SECOND STROKE IS A DOT.
*DETERMINES IF THE FIRST STROKE RESULTS IN A SCRIPT I OR J.
*
*
*CALL
*
RCS (DOTA, III, EYES, ENO
*WHERE II 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 OF
   DS 3F
   DS 26H
   DS 20C
   DS 3F
   DS 1H
   DS 2C
   DS 3H
   DS 1C
CHAR DS 1C
DOT BCX
*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,R1C
LA R7,ILIST

CKLIST EQU *
IC R10,0(R7)
CR R10,R9
BC 8,NOX
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'
DC X'85'
DC X'A5'
DC C'L'
DC C'2'
DC X'82'
DC X'70'

JLIST EQU *
DC X'86'
DC X'91'
DC X'F8'
DC X'E5'
DC X'DD'
DC X'00'

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  0*
*XLC  7  6  5  4  XRC*
*  11 10  9  8*
*  15 14 13 12*
*YBC*
CALL RCS DELTASA,III,EXIT
*WHERE II IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
*
*
*INPUT REGISTER. R6
*
*INTERNAL REGISTERS. R7 THRU R14
*
*
USING XR6,R6
REGS
EXO EQU 0
D6 DSECT
XR6 CS 0F
DS 3F
CS 11H
SN CS 1H
DS 6H
XRC CS 1H
XLC CS 1H
YTC DS 1H
YBC DS 1H
DS 4H
XYE CS 10C
XYS CS 10C
DS 3F
CS 1H
CS 2C
CS 3H
CS 8C
XSP CS 10C
YSP CS 10C
XEP CS 10C
YEP CS 10C
DELTAS BOX
SR R7,R7
LA R8,2
LH R9,SN
SLL R9,1
BCT R9,DELI
CEL1 LH R10,XRC
SH R10,XLC
SRL R10,2
LH R11,YTC
SH R11,YBC
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 EX0
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,III,EEXIT
*WHERE II IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
*
*
*INPUT REGISTER. R6
*
*INTERNAL REGISTERS. R7
*
*
USING XR6,R6
REGS
EXO EQU 0
D6 DSECT
XR6 CS 0F
    DS 2F
CODE DS 1F
    DS 10H
N    DS 1H
    DS 4H
BR56 CS 1H
FN56 BOX
    LA R7,16
    STH R7,BR56
    CLI N+1,X'05'
    BC 4,FN56E
    SR R7,R7
    IC R7,CODE+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
RECS
EXO EQU 0
D6 DSECT
XR6 DS OF
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,II1,EXIT
*WHERE II1 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
CYC 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,II1,EEXIT
* WHERE II1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
*
*
* INPUT REGISTER. R6
*
* INTERNAL REGISTERS. R7
*
*
 USING XR6,R6
REGS
EXO EQU 0
C6 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
CYS DS 1H
XRS DS 1H
XLS DS 1H
YTS DS 1H
YBS DS 1H
**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,II1,EEXIT**
  * **WHERE II1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST**

**INPUT REGISTER.** R6

**INTERNAL REGISTERS.** R7 THRU R14

**USING XR6,R6**
REGS

EXO 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+1C, QYM=QYMAX+5

SR R7,R7
LA R8,2
LA R9,20
LH R13,YTC
SH R13,YBC
SRL R13,2
LH R16,YBC
AR R10,R13
YBC + D
LR R11,R10
AR R11,R13
YBC + 2D
LR R12,R11
AR R12,R13
YBC + 3D

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

G3 MVI 0(R14),X'03'
   B BXLE
G2 MVI 0(R14),X'02'
   B BXLE
Q01 CH R12,YMAX(R7)
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 RCE RAZEA,III,EXIT
*WHERE III IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST .
*
*
*INPUT REGISTER. R6
*
*INTERNAL REGISTERS. R8,R15
*
*
USING XR6, R6
EXO EQU 0
REGS
D6 DSECT
XR6 DS 0F
  DS 2F
  DS 1F
HEIGHT DS 1H
  DS 2H
  DS 1H
  DS 2H
DS 56C
DS 21H
CENT DS 1F
RAZE

BCX
BEXIT EX0
L R8,CENT
LH R15,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,111,EEXIT
*WHERE 11 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
*
*
*INPUT REGISTER. R6
*
*INTERNAL REGISTERS. R7,R8
*
*
USING XR6,R6
REGS
EX0 EQU 0
C6 DSECT
XR6 DS 0F DS 3F DS 2H
XT DS 1H
YT DS 1H
CX DS 1H
cy DS 1H DS 3H DS 17H DS 20C DS 3F DS 1H
CS 2C
CS 2H
CEL DS 1H
DS 56C
DS 21H
CS 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+1C
GYMAX DS 10C
GYMIN EQU GYMAX+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 GYMIN,X*01*
BNE EXIT
*A MAX HAS OCCURRED PREVIOUSLY
LH R7,YT
LH R8,PYMIN
SR R7,R8
LPR R7,R7
MAG(YT-PYMIN)
2*THINNING DISTANCE
LH R8,DEL
SLL R8,1
CR R7,R8
BNH EXIT
*A MINIMUM DETECTED
MVI GYMIN,X*00*
MVI GYMAX,X*01*
MVC PYMXX(2),YT
MVC PYMXX(2),XT
LH R7,NYMIN
LA R7,1(R7)
LA R8,5
CR R7,R8
BNH NX0K
SR R7,R7
N O CK
STH R7,NYMIN
BC TR R7,0
S LL R7,1
L H R8,PYMIN
STH R8,YMIN(R7)
L H R8,PYMNX
STH R8,YMINX(R7)
B EXIT
N C
CH R7,PYMIN
B NL PMXN
STH R7,PYMIN
M VC PMXN(2),XT
B PMAX
P MXN
L H R7,DY
L TR R7,R7
B P M IN
*DOW N WD STYL US MOTI ON
P MAX
CLI QYMAX,X'01'
BNE EXIT
*A MIN HAS OCCURRED PREVIOUSLY
L H R7,YT
L H R8,PYMAX
S R R7,R8
L PR R7,R7
L H R8,DEL
S LL R8,1
C R R7,R8
BNH EXIT
*A MAXIMUM DETECTED
M VI QYMAX,X'00'
M VI QYMIN,X'01'
M VC PMIN(2),YT
M VC PMXN(2),XT
L H R7,NYMAX
L A R7,1(R7)
L A R8,5
C R R7,R8
BNH NNOK
S R R7,R7
N NOK
STH R7,NYMAX
BCTR R7,0
S LL R7,1
L H R8,PYMAX
STH R8,YMAX(R7)
L H R8,PYMX
STH R8,YMAXX(R7)
EXIT BEXIT EX0
END
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 SMOOTH, 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
EX0 EQU 0
D6 DSECT
XR6 DS 0F
SMOOTH BOX
LR R8, R6
SRA R8, 2
SR R6, R8
SRA R7, 2
AR R6, R7
BEXIT EX0
END
TCRN

*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,Rll,EEXIT
*WHERE Rl IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
*
*
*INPUT REGISTER. R6
*
*INTERNAL REGISTERS. R8
*
* USING XR6,R6
REGS
EXO EQU 0
D6 DSECT
XR6 DS OF
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
NTCUSB CS 1H
PNPTS CS 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
R8=6*PNPTS
CH R8,NPTS
BC 10,CUSPID
CLI CUSP,X'CC'*
BC 6,NOCUSP
*CUSP=0
LH R8,NT
LA R8,1(C,R8)
STH R8,NT
MVI CUSP,X'01'*
LH R8,NT
STH R8,NTC
B NOCUSP
CUSPID MVI CUSP,X'CO'*
NCCUSP LH R8,NPTS
CH R8,PNPTS
BC 10,NPTS0
STH R8,PNPTS
NPTS0 XC NPTS(2),NPTS
BEXIT EXO
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 EX0 EQU 0 C6 DSECT XR6 DS OF 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 EX0 END

TURNER

* FUNCTION
* 
* DETECTS 180 DEGREE CHANGE IN STYLIUS DIRECTION THAT OCCURS AFTER A
* SINGLE THINNING DISTANCE
* IF SUCH A TURN IS DETECTED, TTUREN+1 = 1 FOR CLOCKWISE TURN, TTUREN+1 =2
* FOR COUNTERCLOCKWISE TURN, OTHERWISE TTUREN+1 = 0.
* 
*
*CALL
* RCS TURNERA,III,EXIT
*WHERE II IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
*
*
*INPUT REGISTERS
*
*C(R6) = II
*C(R9) = ANGLE (AS ENCODED BY *CHAREC*)
*
*
*INTERNAL REGISTERS.  R7, R8
*
*
  USING XR6,R6
EX0 EQU 0
REGS
D6 DSECT
XR6 CS OF
DS 3F
DS 8H
PANG CS 1H
PACANG DS 1H
DS 16H
DS 20C
DS 3F
DS 1H
CS 2C
CS 3H
CS 56C
DS 21H
CS 1F
DS 6C
ITURN DS 1H
TURNER BX
*ANG DC ES NCT EQ PREV ANG
*TEST F CR 180 DEG TLRN
*DCEP 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?
*CRO COU NTERCLOCKWISE?
  LR    R7,R9
  SH    R7,PANG
  LTR   R7,R7
  BC    2,CCTURN

*POSSIBLY CLKWISE
*DOES P ANG EQ 0?
  SR    R7,R7
  CH    R7,PANG
  BC    8,CCT1
  CTURN MVI TTURN+1,X'01'
  BC    15,TURNX

*POSSIBLY CCLKWISE
*DOES A NG EQ 0?
  CTURN SR    R7,R7
   CR    R7,R9
   BC    8,CCTURN
  CCT1 MVI TTURN+1,X'02'
  BC    15,TURNX

*NOT A 180 DEG WITH SINGLE ANGLE
NOC TURNS 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 ENC. PT. OF UPDATED VECTOR TRACK
* C(R11) = Y ENC. PT. OF UPDATED VECTOR TRACK
*OTHERWISE
* C(R0) = X*0'
* C(R10), C(R11) NOT UPDATED
*
*
*INTERNAL REGISTERS R6, R8, R14
*
*
USING XR6, R6
EXO
EQU 0
REGS
C6 CS OF
CSECT
XR6 OF

LA R0, 16 RAST/DIR CONSTANT
SR R14, R14 QUADRANT CODE
SR R7, R10 X(I) - X(L)
BC 10, TRAV1
LA R14, 4(G,R14) QUAD 2 OR 3
LPR R7, R7 ABS DX

TRAV1
SR R9, R11 Y(I) - Y(L)
BC 10, TRAV2
LA R14, 81(G,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)
CR 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
CR 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,RC
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 EX0
TRAVK4 CC F*4*
TQUAD CC F*0*
CC F*-8*
CC F*-16*
CC F*8*
TXIN CS OH TABLE FOR 2 RAST VEC
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*-8*
CC H*-4*
CC H*0*
CC H*4*
CC H*8*
CC H*8*
TYIN CS OH TABLE FOR 2 RAST VECTORS
**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,II1,III,EEXIT
* WHERE AREC IS A LINKAGE BETWEEN CHAREC'S CONTEXT AND REC'S CONTEXT
* RECA IS A LINK TO REC
* II1 IS AT THE TOP OF CHAREC'S INTERNAL PARAMETER LIST
* EXIT EEXIT IS THE ONLY EXIT
*
**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 'TILD'TO TEST FOR TILDA
*IF NOT TILDA, GO TO SET-UP TABLE RE-ENTRY
*IF TILCA, 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 CN 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 'CAN'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   OF
REGS      DS   3F
BANK      DS   1F
INDEX     DS   1F
          DS   1F
EXIT      ECU  0
DSECT3    DSECT
XR3       CS   OF
SCRTCH    CS   1F
CCNC      CS   1F
CAT             SCRATCH
XRX       CS   0F
II        CS   1F
PAD       CS   1F
CGDE      CS   1F
XS        CS   1H
YS        CS   1H
XT        CS   1H
YT        CS   1H
CX        CS   1H
CY        CS   1H
MDX       CS   1H
MDY       CS   1H
PANG      CS   1H
PACANG    CS   1H
N         CS   1H
SN        CS   1H
PUP       CS   1H
INKIND    CS   1H
PGUAD     CS   1H
BR56      CS   1H
DXC       CS   1H
CYC       CS   1H
XRC       CS   1H
XLC       CS   1H
YTC       CS   1H
YBC       CS   1H
ASPR      CS   1H
NT        CS   1H
NTC       CS   1H
INKC CS 1H
XYE CS 10C
XYS CS 10C
WIDTH DS 1H
HEIGHT DS 1H
DS 2F
YCENT DS 1H
PCHAR DS 1C
CUSP DS 1C
NCUSP DS 1H
NPTS CS 1H
DEL CS 1H
P CS 1C
CHAR CS 1C
TEMP CS 1C
TINK DS 5C
XSP CS 10C
YSP CS 10C
XEP CS 10C
YEP CS 10C
ALXYJ DS 8C
XL CS 1H
YL CS 1H
XLO CS 1H
YLC CS 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 CS 1H
AX02 DS 1H
NC DS 1H
C DS 1H
CYM CS 1H
CXS CS 1H
CYS DS 1H
XRS CS 1H
XLS CS 1H
YTS CS 1H
YBS DS 1H
CENT CS 1F
MVC DS 6C
TTURN CS 1H
TURN DS 1F
XC CS 10C
YC CS 10C
CO CS 1H
C1 CS 1H
C2  DS  1H
C3  DS  1H
C4  CS  1H
C5  DS  1H
C6  CS  1H
C7  DS  1H

*BEWARE, DATA BELOW D7 CANNOT BE REFERRED TO BY TABLE MACROS

C8  DS  1H
C9  DS  1H
C10 CS  1H
C11 DS  1H
C12 CS  1H
C13 DS  1H
C14 CS  1H
C15 CS  1H
CN  DS  1H

NTCUSP EQU AX3
NTCSPI DS  1H
PNPTS CS  1H
PYMAX CS  1H
PYMIN DS  1H
NYMAX EQU AX2
NYMX1 DS  1H
NYMIN EQU AX1
NYMN1 CS  1H
YMAX DS  10H
YMIN EQU YMAX+10
QYMAX EQU AX
QYMIN EQU QYMAX+5
QYMX1 DS  10C
QYMN1 EQU QYMX1+5
PYMXX CS  1H
PYMNX DS  1H
YMAXX DS  10H
YMNX 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****
*
*
SPEQ 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
SML EQU 11
SXHSTR EQU 12
SOMCQ8 EQU 13
SCPNU EQU 14
RBS 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
SB EQU 41
SSTAR EQU 42
SSCRUB EQU 43
DK EQU 44
RSA EQU 45
SE EQU 46
REC PRCCS CLEAR=3,CNTX=6,AUTO=2,PROLG=SINS,ID=8000021C
AHTEST DC V(AHSTR1)
KNYT DC V(KNYTST)
KNTTA DC V(KNTYT)
KVXYTA DC V(KVXYYT)
MWT A DC V(MWT)
PTST DC V(PSTTest)
SYMTA DC V(SYMT)
TILCTA DC V(TILCT)
TPXYA DC V(TPXY)
VERT DC V(VERTST)
VFI DC V(BFI)
VSDP DC V(BSDP)
VSMNW CC V(BSMNW)
VSRPRM DC V(BSRPRM)
VSSM DC V(BSSM)
VSVM CC V(BSVM)
VTEST1 DC V(BTEST1)
VTEST3 DC V(BTEST3)
VTERP CC V(INTERP)
WHITE DC V(BHITE)
VRAZE DC V(RAZE)

* *
****RETURNS****
* *

RETURNS EQU *
BC 15,XAHSTR
BC 15,XKNY
BC 15,XFI
BC 15,XKVVY
BC 15,XMW
BC 15,XMWIN
BC 15,XMW1
BC 15,XPOST
BC 15,XRECD
BC 15,XSDP
BC 15,XSMNW
BC 15,XSMIM
BC 15,XSRRPRM
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,XSCBCOU
BC 15,XSNLC
BC 15,XSPLGC
BC 15,XSRLC
BC 15,XS8LC
BC 15,XSULC
BC 15,XS4LC
BC 15,XSCPEL
BC 15,XSCPMM
BC 15,XSCPYZ
BC 15,XSCPBS
BC 15,XS8VMN
BC 15,XRAZE
BC  15,XS8LCV
BC  15,XSULC1
BC  15,XS4MK1
BC  15,XSELCA

**
**
****INITIAL CODE****
**
**
SINS  PROCLG
     L  R6,BANK
*MOVE DATA USED BY TABLE MACROS ABOVE FF IN DATA BANK
MVC  NTCUSP(2),NTCSPI1
MVC  NYMAX(2),NYMX1
MVC  NYMIN(2),NYMN1
MVC  QYMAX(10),QYMX1
CLI  N+1,X'0C0'
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,DYCI
BC  12,LARGE
CH  R15,DXCI
BC  12,LARGE
*CHARAC T ER IS NOT LARGE
*TEST F OR TILDA
RC  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 SYMBOL
RC  SYMTA,EREENTR,EXRECD
*
*
****COMPUTATIONAL SUBROUTINES****
*
*
XAHSTR  RC  AHTEST,EXRECD
XFI  EQU *
     RC  VFI,EXRECD
XKNY  RC  KNYT,EXRECD
XKNY1  RC  KNYITA,EXRECD
XXVXY  RC  KVXYTA,EXRECD
XMWIN  RC  MWTA,EXRECD
XPOST EQU *
RCS PTST, EXRECD
XSDP EQU *
RCS VSDP, EXRECD
XSM1M EQU *
RCS VERT, EXSM1, EXKVXYM, EXPLUSM, EXPDAEX
XSRPRM EQU *
RCS VSRPRM, EXRECD, EXSDP, EXPOST
XSSM EQU *
RCS VSSM, EXRECD
XSVM EQU *
RCS VSVM, EXRECD, EXSJMU, EXMW1, EXKNY1, EXSOMG, EXSMJU1
XTEST1 EQU *
RCS VTEST1, EX8LCG, EXSSM
XTEST3 EQU *
RCS VTEST3, EXRECD, EXSBM5
XTPLUS EQU *
RCS TPXYA, EXRECD
XSCP EQU *
RCS VWHITE, EXLEU, EXHLO
XSMLCN EQU *
RCS VWHITE, EXSCPNU, EXSM
XSCPMW EQU *
RCS VWHITE, EXLMW, EXSSCR8
XSALCS EQU *
RCS VWHITE, EXRSS, EXSA
XSCP B3 S EQU *
RCS VWHITE, EXRSS, EXRSB
XMLC EQU *
RCS VWHITE, EXLMW, EXHMY
XSNLC EQU *
RCS VWHITE, EXLOV, EXHBJN
XSMMN W EQU *
RCS VSMNW, EXRECD, EXMLC, EXRAZE
XSRLC EQU *
RCS VWHITE, EXRSN, EXSR
XSPLC EQU *
RCS VWHITE, EXRSR, EXSSPP
XSULC1 EQU *
RCS VWHITE, EXRUS, EXSU
XS4LC EQU *
RCS VWHITE, EXRSE, EXRSL
XS8LC EQU *
RCS VWHITE, EXLVO, EXSB
XSULC EQU *
RCS VWHITE, EXRSN, EXHPU
XSRLC EQU *
RCS VWHITE, EXCS, EXSSTAR
XS8VMN EQU *
RCS VWHITE, EXLMNV, EXHBL
XSCB OU EQU *
RCS VWHITE, EXLOU, EXHBM
XSG8LC EQU *
RCS WHITE,EXRSO,EXSG8
XSBLCV EQU *
RCS WHITE,EXRSV,EXS8
XSELCA EQU *
RCS WHITE,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
XSMJUI EQU *
LA R8,SUMJUI
B START
XSM1 EQU *
LA R8,SM1
BC 15,START
XSXMSL LA R8,5XSXMSL
B START
XSOCO EQU *
LA R8,SCMQQ8
B START
XHBL LA R8,5XHBL
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
XRSO   LA R8,RSO
B      START
XRSR   LA R8,RSR
B      START
XRSS   LA R8,RSS
B      START
XRSS   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
XSPP   LA R8,SP
B      START
XSR    LA R8,SR
B      START
XSU    LA R8,SU

B     START
XS0   LA   R8,SC
B     START
XS8   LA   R8,S8
B     START
XSSTAR LA   R8,STAR
B     START
XSCCRB LA   R8,SCCRB
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*C3*
     BC  1,XXKX
     BXLE R15,R13,K4
     LA   R8,S4
     B   START
XSKX  TM   XYSR+1(R7),X*CF*
     BC  8,XXSSTM
     LA   R8,SK
     B   START
XHBM  EQU  *
     TM   XYE+1,X*OC*
     BC  1,XXM
     B   XRSB
XLMW  TM   XYE+1,X*OC*
     BC  8,XRSW
     B   XRSM
XHMY  TM   XYE+1,X*OC*
     BC  12,XRSY
     B   XSM
XLOW  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,XRSO
     B   XRSV
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XHBJN TM</td>
<td>XYS+1,X'0C'</td>
</tr>
<tr>
<td>BC</td>
<td>12,XRSF</td>
</tr>
<tr>
<td>TM</td>
<td>XYE+1,X'0C'</td>
</tr>
<tr>
<td>BC</td>
<td>5,XRSB</td>
</tr>
<tr>
<td>B</td>
<td>XSNN</td>
</tr>
<tr>
<td>XHLO TM</td>
<td>XYE+1,X'03'</td>
</tr>
<tr>
<td>BC</td>
<td>8,XRSL</td>
</tr>
<tr>
<td>B</td>
<td>XSO</td>
</tr>
<tr>
<td>XHRL EQU</td>
<td>*</td>
</tr>
<tr>
<td>XSCP YZ EQU</td>
<td>*</td>
</tr>
<tr>
<td>XCS EQU</td>
<td>*</td>
</tr>
<tr>
<td>X8LC G CLI</td>
<td>CODE,X'3C'</td>
</tr>
<tr>
<td>BC</td>
<td>8,XRSC</td>
</tr>
<tr>
<td>TM</td>
<td>CODE,X'0C'</td>
</tr>
<tr>
<td>BC</td>
<td>8,XRSC</td>
</tr>
<tr>
<td>B</td>
<td>XRSS</td>
</tr>
<tr>
<td>X8LC U CLI</td>
<td>N+1,X'C5'</td>
</tr>
<tr>
<td>BC</td>
<td>2,XRSE</td>
</tr>
<tr>
<td>XLM NV EQU</td>
<td>*</td>
</tr>
<tr>
<td>CLI</td>
<td>N+1,X'C5'</td>
</tr>
<tr>
<td>BC</td>
<td>8,XNV</td>
</tr>
<tr>
<td>BC</td>
<td>4,XRSE</td>
</tr>
<tr>
<td>CLI</td>
<td>N+1,X'C6'</td>
</tr>
<tr>
<td>BC</td>
<td>2,XRSM</td>
</tr>
<tr>
<td>B</td>
<td>XSCPNU</td>
</tr>
<tr>
<td>XNV TM</td>
<td>XYE+1,X'08'</td>
</tr>
<tr>
<td>BC</td>
<td>1,XSCPNU</td>
</tr>
<tr>
<td>B</td>
<td>XRSV</td>
</tr>
<tr>
<td>XLC U TM</td>
<td>XYE+1,X'08'</td>
</tr>
<tr>
<td>BC</td>
<td>8,XRSO</td>
</tr>
<tr>
<td>B</td>
<td>XRSU</td>
</tr>
<tr>
<td>XLVC TM</td>
<td>CODE,X'08'</td>
</tr>
<tr>
<td>BC</td>
<td>1,XRSO</td>
</tr>
<tr>
<td>B</td>
<td>XRSV</td>
</tr>
</tbody>
</table>
XSG8  TM  XYE+1,X'0C'
BC  8,XSG
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,SCRATCH
RCS  VTERP,EE0,EE1              CALL THE INTERPRETER
EXO  EQU  *
ST  R8,SCRATCH
SWITCH INDEX=SCRATCH,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 (DEPIRED BELOW) IS
* USED TO PERFORM THE TESTS.

INTERP Call
**RCS INTERPA, EVALIC, ERROR**
* WHERE INTERPA IS A LINK TO INTERP
* EXIT VALIC 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
*FMR:
*LABEL TABLE /OP1,P1,C1/,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
* CI = OR IMMEDIATE
* X2 = EXCLUSIVE OR IMMEDIATE
* TR = TRANSLATE
* SS = SWITCH
* EX = EXIT FROM TABLE
* IF CPI = TR
* PI = THE TRANSLATION INDEX
* CI = OC
* CIJ = C
* LIJ = START OF A LIST OF DC'S
* IF OPI = SS
* PI = TEMP
* CI = OC
* CIJ = C
* LIJ = START OF LIST OF BRANCHES
* IF CPI = EX
* PI = A 'REC' LABEL
* CI = 0
* CIJ,LIJ ARE OMITTED
OTHERWISE
* 
PI = THE FEATURE TO BE TESTED OR MODIFIED (ONLY 'P', 'PAD', OR
* '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'
*
****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'
MV EQU X'92'
NI EQU X'94'
CL EQU X'95'
CI EQU X'96'
X2 EQU X'97'
TR EQU X'99'
SS EQU X'9A'
EX EQU X'9B'

CATA CSECT
XRX DS 0F
II DS 1F
PAD DS 1F
CODE DS 1F
XS DS 1H
YS DS 1H
XT CS 1H
YT CS 1H
cx CS 1H
cy CS 1H
mdx CS 1H
mdy CS 1H
PANG CS 1H
PACANG CS 1H
n DS 1M
sn CS 1H
PUP CS 1H
INKIND CS 1H
PGUAD CS 1H
BR56 CS 1H
DXC CS 1H
dyc CS 1H
XRC CS 1H
XLC CS 1H
YTC CS 1H
YBC CS 1H
ASPR CS 1H
nt CS 1H
NTC CS 1H
INKC CS 1H
XYE CS 10C
XYS CS 10C
WIDTH CS 1H
HEIGHT CS 1H
YCENT CS 1H
PCHAR CS 1C
CUSP CS 1C
NCUSP CS 1H
NPTS CS 1H
CEL CS 1H
P CS 1C
CHAR CS 1C
TEMP CS 1C
TINK CS 5C
XSP CS 10C
YSP CS 10C
XEP CS 10C
YEP CS 10C
ALXYJ CS 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   CS  1H
AX23 CS  1H
AX12 CS  1H
AX01 CS  1H
AX02 CS  1H
NC   DS  1H
C    DS  1H
CYM  DS  1H
DXS  DS  1H
DYS  CS  1H
XRS  CS  1H
XLS  CS  1H
YTS  CS  1H
YBS  CS  1H
CENT CS  1F
MVC  DS  6C
TTURN DS  1H
TURN CS  1F
XC   CS  10C
YC   CS  10C
D0   DS  1H
C1   CS  1H
D2   DS  1H
D3   CS  1H
C4   CS  1H
D5   CS  1H
C6   CS  1H
C7   CS  1H

*BEWARE, DATA BELOW D7 CANNOT BE REFERRED TO BY TABLE MACROS
D8   DS  1H
D9   CS  1H
D10  CS  1H
D11  CS  1H
D12  CS  1H
D13  CS  1H
D14  DS  1H
D15  DS  1H
CN   CS  1H
NTCUSP EQU AX3
NTCSP1 CS  1H
PNPTS CS  1H
PYMAX CS  1H
PYMIN CS  1H
NYMAX EQU AX2
NYMX1 CS  1H
NYMIN EQU AX1
NYMN1 DS  1H
YMAX CS  10H
YMIN EQU YMAX+1C
QYMAX EQU AX
*BEWARE, DATA BELOW D7 CANNOT BE REFERRED TO BY TABLE MACROS*

**INTERPRETER**

```
START EQU *
LA R7, BASE
LA R8, GPSW
MVI O(R14), X'00'
AGAIN EQU *
CLI 0(R8), X'90'
BC 4, BRANCH
CLI 0(R8), X'9F'
BC 2, BRANCH
MAGIC EQU *
CLI 0(R8), X'98'
BC 2, T99
COMM EQU *
MVC 0(2,R15), 0(R8)
SR R9, R9
IC R9, 2(R8)
LA R9, DATA(R9)
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'
BC 2, BRANCH
CLI 0(R8), X'99'
BC 7, T99
MVC 0(2,R15), 3(R8)
LH R9, 0(R15)
LA R9, 0(R7, R9)
SR R10, R1C
IC R10, 2(R8)
```
LA R10, DATA(R10)  
MVC TEMP(1), O(R10)  
TR TEMP(1), O(R9)  
LA R8, 5(R8)  
BC 15, AGAIN  

T9A EQU *  
CLI 0(R8), X'9A'  
BC 7, T9B  
SR R9, R9  
IC R9, 2(R8)  
LA R10, DATA(R9)  
IC R9, 0(R10)  
SLL R9, 1  
MVC 0(12, R15), 3(R8)  
LH R10, C(R15)  
LA R10, 0(R7, R10)  
LA R10, 0(R9, R10)  
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)  
LH R8, 0(R15)  
BEXIT EX0

BRANCH EQU *  
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 R1C, 2  
LA R10, TESTM(R10)  
EX 0, 0(R1C)  
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'
**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
XTST1 EQU 15
XTST3 EQU 16
XTPLUS EQU 17
XSTRLC EQU 18
XSALCS EQU 19
XSGBLC EQU 20
XSMLCN EQU 21
XSGBOU EQU 22
XSNLCE EQU 23
XSPLC EQU 24
XSRLC EQU 25
XS8LC EQU 26
XSULC EQU 27
XS4LC EQU 28
XSCEL EQU 29
XSCEPMW EQU 30
XSCEPYZ EQU 31
XSCEPBS EQU 32
XSCEMVN EQU 33
XRAZE EQU 34
XS8LCV EQU 35
XSULCI EQU 36
XS4MKI EQU 37
XSELCA EQU 38

**TABLE TESTS**

*
<table>
<thead>
<tr>
<th>BASE</th>
<th>EQU  *</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHSTR</td>
<td>TABLE /MV,PAC+3,3A/,15,AHSTRX</td>
</tr>
<tr>
<td>BR</td>
<td>TABLE /TM,CODE+1,80/,1,TEST3,/TM,CODE+1,10/,8,TEST3,/CL,N+1,05X</td>
</tr>
<tr>
<td></td>
<td>/8,SW,15,SM</td>
</tr>
<tr>
<td>FIME</td>
<td>TABLE /MV,P,00/,15,FIME1</td>
</tr>
<tr>
<td>FIME1</td>
<td>TABLE /CL,P,02/,8,SE,CL,P,01/,8,SPOUND,/MV,PAD+3,23/,15,FI</td>
</tr>
<tr>
<td>G6ETST</td>
<td>TABLE /TR,P,00/,0,PBB,SS,TEMP,00/,0,PBBX</td>
</tr>
<tr>
<td>PBB</td>
<td>DS OH</td>
</tr>
<tr>
<td></td>
<td>DC X<em>03</em></td>
</tr>
<tr>
<td></td>
<td>DC 2X<em>C</em></td>
</tr>
<tr>
<td></td>
<td>CC 2X<em>C</em></td>
</tr>
<tr>
<td></td>
<td>CC 3X<em>02</em></td>
</tr>
<tr>
<td></td>
<td>DC 4X<em>03</em></td>
</tr>
<tr>
<td>PBBX</td>
<td>TABLE 15,QQ</td>
</tr>
<tr>
<td></td>
<td>TABLE 15,SQ</td>
</tr>
<tr>
<td></td>
<td>TABLE 15,S8</td>
</tr>
<tr>
<td></td>
<td>TABLE 15,SG6X</td>
</tr>
<tr>
<td>KNYM</td>
<td>TABLE /MV,P,00/</td>
</tr>
<tr>
<td>KNYM1</td>
<td>TABLE /CL,P,C1/,8,MW,/MV,PAD+3,24/,15,KNY</td>
</tr>
<tr>
<td>KVXYM</td>
<td>TABLE /CL,P,C1/,8,KNYM,/CL,P,02/,8,AHSTR,/MV,PAD+3,06/,/MV,P,0X</td>
</tr>
<tr>
<td></td>
<td>B',15,KVXY</td>
</tr>
<tr>
<td>LPRSLA</td>
<td>TABLE /CL,XYE+1,0F/,8,SSLASH,15,SLPAR</td>
</tr>
<tr>
<td>MK</td>
<td>TABLE /MV,P,C0/,/TM,XYS+3,02/,1,SM,15,SK</td>
</tr>
<tr>
<td>*P IF SECOND STROKE IS NOT SINGLE ANGLE OR DOUBLE ANGLE</td>
<td></td>
</tr>
<tr>
<td>CQ</td>
<td>TABLE /CL,N+1,02/,2,SP,/TM,XYS+3,0C/,5,SQ,/TM,XYS+3,03/,1,SQ,1X</td>
</tr>
<tr>
<td></td>
<td>5,S0</td>
</tr>
<tr>
<td>PADEX</td>
<td>TABLE /SS,PAC+3,00/,0,PADT</td>
</tr>
<tr>
<td>PARS LA</td>
<td>TABLE /CL,XYS+1,00/,8,LPRSLA,/TM,XYS+1,02/,8,SLPAR,1,SRPAR</td>
</tr>
<tr>
<td>PGTR2</td>
<td>TABLE /MV,P,C2/,15,PADEX</td>
</tr>
<tr>
<td>TEST5</td>
<td>TABLE /MV,PAD+3,1E/,15,SVM</td>
</tr>
<tr>
<td>TPLUSM</td>
<td>TABLE /CL,P,C1/,8,AHSTR,/CL,P,02/,8,FIME,/MV,PAD+3,36/,/MV,P,0X</td>
</tr>
<tr>
<td></td>
<td>A',15,TPLUS</td>
</tr>
<tr>
<td>XMK</td>
<td>TABLE /MV,P,00/,15,XMK1</td>
</tr>
<tr>
<td>XMK1</td>
<td>TABLE /CL,P,C1/,8,SK,/CL,P,02/,8,AHSTR,/MV,PAD+3,18/,/TM,XYS+1X</td>
</tr>
<tr>
<td></td>
<td>02/,8,KVXY,/TM,XYS+3,02/,8,KVXY,/TM,XYE+3,02/,8,SR,/TM,X</td>
</tr>
<tr>
<td></td>
<td>XYE+3,0C/,1,SC,15,SP</td>
</tr>
<tr>
<td>EJECT</td>
<td></td>
</tr>
<tr>
<td>SASTAR</td>
<td>TABLE /TM,XYE+1,0C/,12,SALC,EX,XSTRLC,0/</td>
</tr>
<tr>
<td>SALC</td>
<td>TABLE /TM,XYS+1,0C/,12,RSF,/TM,XYE+1,0C/,8,RSV,/TM,CODE,08/,1,X</td>
</tr>
<tr>
<td></td>
<td>SALCS,/TM,CODE,04/,8,SCPBS,/TM,CODE,02/,1,SALCS,15,SCPBS</td>
</tr>
<tr>
<td>SALCF</td>
<td>TABLE /TM,XYS+1,0C/,1,SA,/MV,CHAR,86/,15,RECO</td>
</tr>
<tr>
<td>SALCS</td>
<td>TABLE /EX,XSALCS,0/</td>
</tr>
<tr>
<td>SAMSTR</td>
<td>TABLE /MV,P,C0/,15,SAMST1</td>
</tr>
<tr>
<td>SAMST1</td>
<td>TABLE /CL,P,C1/,8,SSTAR,/MV,PAD+3,15/,15,SA</td>
</tr>
<tr>
<td>SAT</td>
<td>TABLE /CL,P,C0/,8,SA,/MV,PAD+3,37/,15,S7</td>
</tr>
<tr>
<td>SBARM</td>
<td>TABLE /CL,P,C0/,8,SETEQM,2,PGTR2,/CL,P,01/,8,STTPLS,/CL,SN+1,0X</td>
</tr>
<tr>
<td></td>
<td>1/,2,PGTR2,/MV,P,02/,/CL,CODE,00/,8,SMINS,/CL,ASPR,02/,X</td>
</tr>
<tr>
<td></td>
<td>10,SSLASH,5,SMINS</td>
</tr>
<tr>
<td>SBARMK</td>
<td>TABLE /CL,P,01/,8,54MK,/CL,P,02/,8,SETEQM,2,PGTR2,/MV,PAD+3,0DX</td>
</tr>
<tr>
<td></td>
<td>15,SCRKRT</td>
</tr>
</tbody>
</table>
**SBDR1**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>TR, BR56+1, 00/0, II, SS, TEMP, 00/0, III</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>OH</td>
</tr>
<tr>
<td>CC</td>
<td>X'00*</td>
</tr>
<tr>
<td>CC</td>
<td>4X'00*</td>
</tr>
<tr>
<td>CC</td>
<td>X'01*</td>
</tr>
<tr>
<td>CC</td>
<td>4X'01*</td>
</tr>
<tr>
<td>CC</td>
<td>X'02*</td>
</tr>
<tr>
<td>CC</td>
<td>4X'02*</td>
</tr>
<tr>
<td>DC</td>
<td>X'01*</td>
</tr>
<tr>
<td>DC</td>
<td>X'02*</td>
</tr>
</tbody>
</table>

**SBDR1**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>TR, BR56+1, 00/0, KK, SS, TEMP, 00/0, KKK</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS</td>
<td>OH</td>
</tr>
<tr>
<td>CC</td>
<td>X'03*</td>
</tr>
<tr>
<td>CC</td>
<td>X'01*</td>
</tr>
<tr>
<td>CC</td>
<td>X'02*</td>
</tr>
<tr>
<td>DC</td>
<td>2X'04*</td>
</tr>
<tr>
<td>CC</td>
<td>X'00*</td>
</tr>
<tr>
<td>CC</td>
<td>X'01*</td>
</tr>
<tr>
<td>CC</td>
<td>X'04*</td>
</tr>
<tr>
<td>CC</td>
<td>4X'06*</td>
</tr>
<tr>
<td>CC</td>
<td>X'03*</td>
</tr>
<tr>
<td>CC</td>
<td>X'01*</td>
</tr>
<tr>
<td>CC</td>
<td>X'02*</td>
</tr>
<tr>
<td>DC</td>
<td>X'03*</td>
</tr>
<tr>
<td>DC</td>
<td>X'00*</td>
</tr>
</tbody>
</table>

**KKK**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>15, SDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE</td>
<td>15, DK</td>
</tr>
<tr>
<td>TABLE</td>
<td>15, SB</td>
</tr>
<tr>
<td>TABLE</td>
<td>15, TEST3</td>
</tr>
<tr>
<td>TABLE</td>
<td>15, SBR1</td>
</tr>
</tbody>
</table>

**III**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>15, TEST3</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE</td>
<td>15, SDP</td>
</tr>
<tr>
<td>TABLE</td>
<td>15, SPRMA</td>
</tr>
</tbody>
</table>

**SBM5**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>MV, P, CO/15, SBM51</th>
</tr>
</thead>
</table>

**SBM51**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>CL, P, C2/8, SS, MV, PAD+3, 26/15, SB</th>
</tr>
</thead>
</table>

**SBR1**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>CL, N+1, 05/12, SR, TM, CODE+1, CC/12, SB, CL, N+1, 07/8, SR, X 15, SB</th>
</tr>
</thead>
</table>

**SCG**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>CL, N+1, 05/2, SG, 12, SCLC</th>
</tr>
</thead>
</table>

**SCLC**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>CL, CyMIN+03/8, RSD, 15, SCC</th>
</tr>
</thead>
</table>

**SCLB**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>CL, NTCUSP+1, 01/10, SLBRAC, CL, ASP+1, 05/2, SLBRAC, 15, SCX C</th>
</tr>
</thead>
</table>

**SCMEG**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>CL, P, CL/18, SK, CL, P, 02/8, SE, CL, P, 03/8, SG, MV, PAD+3, 1X 6/6, MV, P, 05/1, CL, NTCUSP+1, 01/2, SLBRAC, 8, SCLBRC, CL, ASP+X 1/08/2, SLPAR, 15, SCC</th>
</tr>
</thead>
</table>

**SCMG**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>TR, P, CO/0, PAA, SS, TEMP, OC/0, PAAA</th>
</tr>
</thead>
</table>

**PAA**

<p>| CS     | OH                                      |
| CC     | 2X'02*                                  |
| CC     | 2X'00*                                  |
| CC     | X'02*                                  |
| DC     | 3X'01*                                  |
| DC     | 4X'02*                                  |</p>
<table>
<thead>
<tr>
<th>TABLE</th>
<th>15, SG</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE</td>
<td>15, S8</td>
</tr>
<tr>
<td>TABLE</td>
<td>15, SCMG1</td>
</tr>
<tr>
<td>SCMG1</td>
<td>TABLE</td>
</tr>
<tr>
<td>SCGM</td>
<td>TABLE</td>
</tr>
<tr>
<td>SCGMAM</td>
<td>TABLE</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>SCRKT</td>
<td>TABLE</td>
</tr>
<tr>
<td>SDM</td>
<td>TABLE</td>
</tr>
<tr>
<td>SEGO06M</td>
<td>TABLE</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>SESEQ</td>
<td>TABLE</td>
</tr>
<tr>
<td>SETECM</td>
<td>TABLE</td>
</tr>
<tr>
<td>SETXX</td>
<td>TABLE</td>
</tr>
<tr>
<td>SET1AK</td>
<td>TABLE</td>
</tr>
<tr>
<td>SEQ</td>
<td>TABLE</td>
</tr>
<tr>
<td>SFE</td>
<td>TABLE</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>SFE1</td>
<td>TABLE</td>
</tr>
<tr>
<td>SFME</td>
<td>TABLE</td>
</tr>
<tr>
<td>SFME1</td>
<td>TABLE</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>SGS</td>
<td>TABLE</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>SGSCRB</td>
<td>TABLE</td>
</tr>
<tr>
<td>SGSO6M</td>
<td>TABLE</td>
</tr>
<tr>
<td>SG06M</td>
<td>TABLE</td>
</tr>
<tr>
<td>FF</td>
<td>CS</td>
</tr>
<tr>
<td></td>
<td>CC</td>
</tr>
<tr>
<td></td>
<td>CC</td>
</tr>
<tr>
<td></td>
<td>CC</td>
</tr>
<tr>
<td></td>
<td>CC</td>
</tr>
<tr>
<td></td>
<td>CC</td>
</tr>
<tr>
<td></td>
<td>CC</td>
</tr>
<tr>
<td>FFF</td>
<td>TABLE</td>
</tr>
<tr>
<td></td>
<td>TABLE</td>
</tr>
<tr>
<td></td>
<td>TABLE</td>
</tr>
<tr>
<td></td>
<td>TABLE</td>
</tr>
<tr>
<td>SG069M</td>
<td>TABLE</td>
</tr>
<tr>
<td>EE</td>
<td>DS</td>
</tr>
<tr>
<td></td>
<td>CC</td>
</tr>
<tr>
<td></td>
<td>CC</td>
</tr>
<tr>
<td></td>
<td>CC</td>
</tr>
<tr>
<td></td>
<td>CC</td>
</tr>
<tr>
<td></td>
<td>CC</td>
</tr>
<tr>
<td>EEEE</td>
<td>TABLE</td>
</tr>
<tr>
<td></td>
<td>TABLE</td>
</tr>
<tr>
<td></td>
<td>TABLE</td>
</tr>
<tr>
<td>TABLE 15, S8</td>
<td>TABLE 15, SEG06M</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------</td>
</tr>
<tr>
<td>SG6X</td>
<td>TABLE /MV, PAC+3, 0A/6, CL, NTCSUP+1, 01/6, 10, SG, TM, XYE+1X</td>
</tr>
<tr>
<td></td>
<td>02/6, 1, 15, SG</td>
</tr>
<tr>
<td>SG8</td>
<td>TABLE /TM, XYE+1, 0C/5, 15, SG, EX, XSGBLC, 0/</td>
</tr>
<tr>
<td>SG81</td>
<td>TABLE /TM, XYE+1, 0C/1, 1, SG, 15, S8</td>
</tr>
<tr>
<td>SJMU</td>
<td>TABLE /MV, P, CC/15, SUMXX</td>
</tr>
<tr>
<td>SUMXX</td>
<td>TABLE /CL, P, C2/8, SU, MV, PAD+3, 27/15, SJ</td>
</tr>
<tr>
<td>SK5</td>
<td>TABLE /CL, P, C1/8, SK, MV, PAD+3, 28/15, S5</td>
</tr>
<tr>
<td>SLKRTM</td>
<td>TABLE /MV, PAC+3, 10/8, CL, P, CC/8, SLKRT, 2, SRPRM</td>
</tr>
<tr>
<td>SLMEK4</td>
<td>TABLE /TR, P, CC/0, PEE, SS, TEMP, 00/0, PEEE</td>
</tr>
<tr>
<td>PEE</td>
<td>CS 0H</td>
</tr>
<tr>
<td></td>
<td>CC X<em>03</em></td>
</tr>
<tr>
<td></td>
<td>CC X<em>00</em></td>
</tr>
<tr>
<td></td>
<td>CC X<em>05</em></td>
</tr>
<tr>
<td></td>
<td>CC 3X<em>03</em></td>
</tr>
<tr>
<td></td>
<td>CC X<em>04</em></td>
</tr>
<tr>
<td></td>
<td>CC X<em>03</em></td>
</tr>
<tr>
<td></td>
<td>CC X<em>01</em></td>
</tr>
<tr>
<td></td>
<td>CC 2X<em>03</em></td>
</tr>
<tr>
<td></td>
<td>DC X<em>02</em></td>
</tr>
<tr>
<td>PEEE</td>
<td>TABLE 15, S4Y</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, STPE</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, SK</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, SLX</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, SG</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, SXME</td>
</tr>
<tr>
<td>SLX</td>
<td>TABLE /MV, PAC+3, 1A/6, CL, ASPR+1, 08/2, SLPAR, 12, SL</td>
</tr>
<tr>
<td>SMC</td>
<td>TABLE /TR, BR56+1, 00/0, DD, SS, TEMP, 00/0, DDD</td>
</tr>
<tr>
<td></td>
<td>CS 0H</td>
</tr>
<tr>
<td></td>
<td>CC 4X<em>02</em></td>
</tr>
<tr>
<td></td>
<td>CC X<em>00</em></td>
</tr>
<tr>
<td></td>
<td>DC X<em>01</em></td>
</tr>
<tr>
<td></td>
<td>CC X<em>03</em></td>
</tr>
<tr>
<td></td>
<td>CC X<em>04</em></td>
</tr>
<tr>
<td></td>
<td>CC X<em>06</em></td>
</tr>
<tr>
<td></td>
<td>CC X<em>02</em></td>
</tr>
<tr>
<td></td>
<td>DC X<em>06</em></td>
</tr>
<tr>
<td></td>
<td>CC 3X<em>02</em></td>
</tr>
<tr>
<td></td>
<td>CC X<em>04</em></td>
</tr>
<tr>
<td></td>
<td>CC X<em>01</em></td>
</tr>
<tr>
<td>CCC</td>
<td>TABLE 15, SG</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, SCMG</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, SSM</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, SEG06M</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, S9MG</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, S6S</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, SE</td>
</tr>
<tr>
<td>SMLC</td>
<td>TABLE /CL, N+1, 05/2, SCPMWK, TM, XY+1, 08/8, SCPPYZ, CL, CYMAX+1, X</td>
</tr>
<tr>
<td></td>
<td>00/2, SCPBH, EX, XSMLCN, 0/</td>
</tr>
<tr>
<td>Code</td>
<td>Table Code</td>
</tr>
<tr>
<td>-------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>SM1</td>
<td>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</td>
</tr>
<tr>
<td>SNMA</td>
<td>CL, P, C2/, 8, SA, CL, P, C1/, 8, RSX, /MV, PAD+3, 29/, TM, CODE, 0CX 1/, 1, SNLC+4, SNLCl, TM, XYS+1, 0C/, 12, S2LCY, /EX, XSCBCU, 0/</td>
</tr>
<tr>
<td>SNLC</td>
<td>EX, XSNLC, 0/</td>
</tr>
<tr>
<td>SNLCl</td>
<td>TM, CODE, 08/, 1, SNLC, /TM, XYS+1, 0C/, 1, SNLCl, TM, XYE+1, 08/, 1X S2, 15, SNLC</td>
</tr>
<tr>
<td>SPRMA</td>
<td>MV, PAD+3, 0C/, CL, P, C1/, 8, SM, CL, P, 02/, 8, SA, TM, XYE+1, 0CX 12, SPLIC, TM, XYE+1, 02/, 1, SD, 15, SRLC</td>
</tr>
<tr>
<td>SPLIC</td>
<td>EX, XSPLC, 0/</td>
</tr>
<tr>
<td>SRLC</td>
<td>CL, QYMAX+1, 00/, 8, SRLCX, CL, N+1, 05/, 2, RSK, 15, RSH</td>
</tr>
<tr>
<td>SRPRMJ</td>
<td>CL, P, C2/, 8, SJ, CL, P, 06/, 8, SC, /MV, PAD+3, 2A/, 15, SRPAR</td>
</tr>
<tr>
<td>SS589M</td>
<td>TR, BR56+1, 00/, 0, GG, /SS, TEMP, 00/, 0, GGG</td>
</tr>
<tr>
<td>GGG</td>
<td>15, 59S</td>
</tr>
<tr>
<td>GGG</td>
<td>15, D</td>
</tr>
<tr>
<td>GGG</td>
<td>150, S</td>
</tr>
<tr>
<td>GGG</td>
<td>15, TEST1</td>
</tr>
<tr>
<td>GGG</td>
<td>15, SE</td>
</tr>
<tr>
<td>GGG</td>
<td>15, RSQ</td>
</tr>
<tr>
<td>GGG</td>
<td>15, RSG</td>
</tr>
<tr>
<td>SS8M</td>
<td>TR, BR56+1, 00/, 0, HH, /SS, TEMP, 00/, 0, HHH</td>
</tr>
<tr>
<td>HHH</td>
<td>DS, CH</td>
</tr>
<tr>
<td>HHH</td>
<td>5X, 01</td>
</tr>
<tr>
<td>HHH</td>
<td>X, 03</td>
</tr>
<tr>
<td>HHH</td>
<td>4X, 01</td>
</tr>
<tr>
<td>HHH</td>
<td>X, 02</td>
</tr>
<tr>
<td>HHH</td>
<td>4X, 01</td>
</tr>
<tr>
<td>HHH</td>
<td>X, 02</td>
</tr>
<tr>
<td>HHH</td>
<td>X, 00</td>
</tr>
<tr>
<td>STPA</td>
<td>MV, P, C0/, /MV, PAD+3, 2B/, 15, SALCF</td>
</tr>
<tr>
<td>Command</td>
<td>Table Code</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>STPE</td>
<td>TABLE /MV,P,CC/  ,/MV,PA D+3, 2C/  ,15, SE</td>
</tr>
<tr>
<td>STPH8</td>
<td>TABLE /CL,P,C1/  ,8,SH/  ,/CL,P,02/ ,8, SF/  ,/MV,PA D+3, 2D/  ,15, S8LC</td>
</tr>
<tr>
<td>S8LC</td>
<td>TABLE /EX,XS8LC/  ,0/</td>
</tr>
<tr>
<td>STPK</td>
<td>TABLE /CL,P,C1/  ,8,KVXY/  ,15,STPK1</td>
</tr>
<tr>
<td>STPK1</td>
<td>TABLE /MV,P,CO/  ,/MV,PA D+3, 2E/  ,15, SK</td>
</tr>
<tr>
<td>STPJ</td>
<td>TABLE /MV,P,PA C+3, 2F/  ,15, SJ</td>
</tr>
<tr>
<td>STPM</td>
<td>TABLE /CL,P,C1/  ,8,SM/  ,/MV,PA D+3, 30/ ,15, SYLC</td>
</tr>
<tr>
<td>SYLC</td>
<td>TABLE /CL,N+1,05/  ,2,RSN/  ,/CL, QYMIN/  ,03/ ,8,RSN,15,SY</td>
</tr>
<tr>
<td>STP5</td>
<td>TABLE /CL,P,PA C+3, 03/  ,8, S5/  ,/CL,P,02/ ,8, S5/  ,/MV,PA D+3, 03/  ,/TM,YXE+X 1,08/ ,8, S8/  ,/CL, CODE/  ,CB/  ,6, S5/  ,/TM,TURN/  ,04/ ,8, S6,15, S5</td>
</tr>
<tr>
<td>STP6</td>
<td>TABLE /MV,P,PA C+3, 02/  ,/CL,P,01/ ,8, S0/  ,/CL,P,02/ ,8, S5/  ,/CL,P,06/ ,8, X  S8/  ,/MV,P,06/ ,15, S6</td>
</tr>
<tr>
<td>STTPLS</td>
<td>TABLE /MV,P,C0/  ,15, TPLUSM</td>
</tr>
<tr>
<td>SUMAM</td>
<td>TABLE /MV,P,PA D+3, 0F/  ,/CL,P,C1/ ,8, MK/  ,/CL,P,02/ ,8, SA/  ,/CL,N+1,05/  ,X 2,SN/  ,/CL, CODE/  ,C7/  ,8, SULC1,15, SULC</td>
</tr>
<tr>
<td>SULC</td>
<td>TABLE /CL, QYMAX+1, OC/  ,2, RSH/  ,/TM,YXE+1, OC/  ,12, RSR/  ,/EX,XSULC/  ,O/</td>
</tr>
<tr>
<td>SULC1</td>
<td>TABLE /CL,QYMIN/  ,03/  ,4, RSY/  ,/EX,XSULC1/  ,O/</td>
</tr>
<tr>
<td>SUMJU</td>
<td>TABLE /CL,P,C2/ ,8, SJMU/  ,/CL,P,01/ ,8, SY/  ,/CL,P,07/ ,8, S8/  ,/MV,TEMP/  ,X 00/ ,15, TEST5</td>
</tr>
<tr>
<td>SUMJUBL</td>
<td>TABLE /MV,P,C7/  ,/MV,PA D+3, 31/ ,15, SU</td>
</tr>
<tr>
<td>SXHBL</td>
<td>TABLE /CL,N+1,04/  ,8, RSL/  ,/CL,NYMAX+1, 01/ ,12, RSR/  ,/CL,N+1,07/ ,2, RX  SK,15,RSH</td>
</tr>
<tr>
<td>SXME</td>
<td>TABLE /MV,P,PA D+3, 1B/  ,/MV,P,CO/  ,/CL, CODE/  ,00/ ,8, SBLBAC,15, SX 18</td>
</tr>
<tr>
<td>SXMSTR</td>
<td>TABLE /MV,P,C/  ,15, SXMSTR1</td>
</tr>
<tr>
<td>SXMSTR1</td>
<td>TABLE /CL,P,C0/  ,8, SSTAR/  ,/MV,PA D+3, 32/ ,15, SX 32</td>
</tr>
<tr>
<td>EJECT</td>
<td>TABLE /TR,P,CO/  ,0,PBBB/  ,/SS,TEMP/  ,00/ ,0,PBBB</td>
</tr>
<tr>
<td>PBBB</td>
<td>TABLE 15,0Q</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, SQ</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, S8</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, SCX</td>
</tr>
<tr>
<td>SOX</td>
<td>TABLE /MV,P,PA D+3, 09/  ,/MV,P,06/ ,/TM, CODE/  ,CO/ ,1,SOD,15, SO</td>
</tr>
<tr>
<td>SOD</td>
<td>TABLE /CL, NICUSP+1, 01/ ,10, SD,15, SO</td>
</tr>
<tr>
<td>S023MB</td>
<td>TABLE /TM,YXE+1, 08/ ,1,S23MB,15, SCM0Q8</td>
</tr>
<tr>
<td>S09</td>
<td>TABLE /TM,YXE+1, OC/ ,1,S9,15, SCM0Q8</td>
</tr>
<tr>
<td>S09M</td>
<td>TABLE /CL,N+1,05/ ,2, SCM0Q8,15, S9 LC1</td>
</tr>
<tr>
<td>S1MAK</td>
<td>TABLE /TR,P,CO/ ,0,PCC/  ,/SS,TEMP/  ,00/ ,0,PCC</td>
</tr>
<tr>
<td>PCC</td>
<td>DS 0H</td>
</tr>
<tr>
<td></td>
<td>CC X'02*</td>
</tr>
<tr>
<td></td>
<td>CC X'00*</td>
</tr>
<tr>
<td></td>
<td>CC X'01*</td>
</tr>
<tr>
<td></td>
<td>CC X'03*</td>
</tr>
<tr>
<td></td>
<td>CC 7X'00*</td>
</tr>
<tr>
<td></td>
<td>CC X'03*</td>
</tr>
<tr>
<td>PCCC</td>
<td>TABLE 15, SET1AK</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, STPA</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, S1MAK</td>
</tr>
<tr>
<td></td>
<td>TABLE 15, STPK</td>
</tr>
<tr>
<td>S1MAKX</td>
<td>TABLE /MV,P,PA D+3, 19/  ,/MV,P,09/ ,15, S1</td>
</tr>
<tr>
<td>S1RPAR</td>
<td>TABLE /TM,YXE+1, 02/ ,1, SRPAR,15, S1</td>
</tr>
<tr>
<td>S2MRZ</td>
<td>TABLE /CL,P,C2/  ,8,SZ/  ,/CL,P,01/ ,8, TEST3/  ,/TM,XYS+1, OC/  ,1,SASTAR/  ,X</td>
</tr>
</tbody>
</table>
/CL,CCDE,3B/,6,N0323,/CL,CCDE+1,AA/,8,S3MB,/CL,CCDE+1,3AX
/8,S3MB,15,N0323
N0323 TABLE /MV,PAD+3,OB/,15,S2LC
S2LC TABLE /TM,YE+1,08/,1,52,/CL,N+1,11,4/,2,RSR,15,RSF
S2LC1 TABLE /MV,PAD+3,OB/,7,52,YS+1,08/,8,S2MRZ,/CL,CCDE,24/,8,RSR,1X
5,RSR
S2LCY TABLE /CL,N+1,11,4/,8,S2,15,RSY
S2LCZ TABLE /CL,N+1,04/,8,S2,15,RSZ
S23MB TABLE /CL,N+1,04/,2,S23MB1,/CL,CCDE,19/,8,SOMQ8,15,S23MB1
S23MB1 TABLE /TR,BR56+1,00/,0,BB/,SS,TEMP,00/,0,BBB
BB DS OH
CC X*00*
CC 4*X02*
CC X*00*
CC 4*X02*
CC X*02*
CC 4*X02*
CC X*02*
CC X*00*
S23MBP EQU *
TABLE /TM,XYE+1,00/,1,TEST3,/CL,PAD+3,0E/,8,S3MB,/CL,N+1,07/,2X
8,S3MB,/TR,BR56+1,00/,0,CC,/SS,TEMP,00/,0,BBB
CC DS OH
CC X*00*
CC 3*X02*
CC 10*X00*
CC X*02*
CC X*00*
CC X*01*
BBB TABLE 15,S2MRZ
TABLE 15,SOMQ8
TABLE 15,S3MB
S24 TABLE /TM,XYE+1,00/,1,S4LC,/MV,PAD+3,08/,15,S2
S4LC TABLE /TM,XYE+1,03/,12,S4,/MV,PAD+3,38/,/EX,XX4LC,0/
S3MC TABLE /CL,PC1,1,8,TEST3,/MV,PAD+3,0E/,15,S3LC
S3LC TABLE /TM,XYE+1,08/,8,RSZ,/TM,XYE+1,OC/,1,RSR,15,S3
S3LC1 TABLE /TM,XYE+1,08/,8,RSZ,/TM,XYE+1,02/,1,S3,15,RSR
S3MR TABLE /CL,PAD+3,04/,8,S3MBR1,/TM,CCDE,03/,1,S3MRB1,/CL,N+1,04/X
8,S2LC1,15,S3MRB1
S3MRB1 TABLE /CL,PC1,1,8,8R/,MV,PAD+3,04/,15,S3LC
S3SCR TABLE /CL,N+1,04/,2,SSCRUB,12,S3MB
S4MK TABLE /MV,PC/,15,S4MK1
S4MK2 TABLE /CL,PC1,1,8,SKV,5M,PAD+3,30/,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 HAVE ITS ENDPT IN RIGHT 1/4
S4Y1 TABLE /TM,XYE+1,03/,8,S4MK,15,SY
S6S TABLE /TM,TURN+1,40/,1,S5M,15,STP6
S65 TABLE /TM,TURN,01/,1,S5M,15,STP6
S7MK TABLE /CL,PAD+3,18/,8,XMK1,/TR,P,00/,0,PDD,/SS,TEMP,00/,0,PDD
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,STK
TABLE 15,SQ
TABLE 15,SIG

S7X TABLE /MV,PAC+3,17/,/MV,P,03/,/CL,NCUSP+1,01/,10,S7,/CL,ASPR+1X
,08/,2,SRPAR,15,S7

S8LCV TABLE /MV,PAC+3,39/,/TM,XYE+1,08/,1,S2,15,S8LCVX
S9MG TABLE /CL,P,C2/8,SG,/MV,PAD+3,34/,15,S9LC
S9LC1 TABLE /TM,XYE+1,OC/,1,SCPEL,/CL,NYMIN+1,01/,12,S9,/TM,XY+1,0C
,1,RSB,15,RSF
S9LC TABLE /CL,NYMIN+1,01/,2,S9LC2N,4,S9LCGGQ/,/CL,QYMIN,03/,8,SCP~D,X
15,S9LCGGQ
S9LC2N TABLE /CL,QYMIN+1,03/,8,SCP~D,15,S9LCGGQ
S9LCGGQ TABLE /TM,XYE+1,OC/,1,S9,15,SCPGQ
S9MK TABLE /CL,P,CL,8,SK,/MV,PAD+3,35/,15,S9

*SCRIPT LETTERS
SCP~D TABLE /TM,XYE+1,08/,8,SCP~GQ/,/CL,QYMAX,00/,8,RSA,15,SC/P~X+1,0C/,1
8,RSA,15,RSF

SCP~GQ 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,OC/,1,RSH,15,RSB
SCP~M~W TABLE /CL,QYMAX+1,00/,2,RSK,15,SCPMW
SCPNU TABLE 15,RSN
SCP~B~S TABLE /EX,XSCPBS,0/
SCPFP TABLE /TM,XY+1,0C/,12,SCPFPY/,/MV,PAD+3,39/,/EX,XSBVMN,0/
SCPFPY TABLE /CL,N+1,05/,12,RSF,15,SCPYZ
SCP~N~RZ TABLE /TM,XY+1,0C/,12,S2LCz/,/CL,N+1,04/,8,RSR,15,07/,2,RX
SM,15,SCP~N~RZ
SCRPT TABLE /TM,XY+1,0C/,12,SCPGQ,15,SCP~N~RZ,00/,2,RSD,15,05/,1X
2,RSR,15,RSR
SCPTX TABLE /CL,P,01/,8,RSX,15,RST

* *
****SET-UP CHARACTER CODE****
* *
<table>
<thead>
<tr>
<th></th>
<th>TABLE</th>
<th>MV, CHAR, C1, 15, RECD</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>TABLE</td>
<td>MV, CHAR, C2, 15, RECD</td>
</tr>
<tr>
<td>SB</td>
<td>TABLE</td>
<td>MV, CHAR, C3, 15, RECD</td>
</tr>
<tr>
<td>SCC</td>
<td>TABLE</td>
<td>MV, CHAR, C4, 15, RECD</td>
</tr>
<tr>
<td>SD</td>
<td>TABLE</td>
<td>MV, CHAR, C5, 15, RECD</td>
</tr>
<tr>
<td>SE</td>
<td>TABLE</td>
<td>MV, CHAR, C6, 15, RECD</td>
</tr>
<tr>
<td>SF</td>
<td>TABLE</td>
<td>MV, CHAR, C7, 15, RECD</td>
</tr>
<tr>
<td>SG</td>
<td>TABLE</td>
<td>MV, CHAR, C8, 15, RECD</td>
</tr>
<tr>
<td>SH</td>
<td>TABLE</td>
<td>MV, CHAR, D1, 15, RECD</td>
</tr>
<tr>
<td>SJ</td>
<td>TABLE</td>
<td>MV, CHAR, D2, 15, RECD</td>
</tr>
<tr>
<td>SK</td>
<td>TABLE</td>
<td>MV, CHAR, D3, 15, RECD</td>
</tr>
<tr>
<td>SL</td>
<td>TABLE</td>
<td>MV, CHAR, D4, 15, RECD</td>
</tr>
<tr>
<td>SM</td>
<td>TABLE</td>
<td>MV, CHAR, D5, 15, RECD</td>
</tr>
<tr>
<td>SNN</td>
<td>TABLE</td>
<td>MV, CHAR, D6, 15, RECD</td>
</tr>
<tr>
<td>SC</td>
<td>TABLE</td>
<td>MV, CHAR, D7, 15, RECD</td>
</tr>
<tr>
<td>SP</td>
<td>TABLE</td>
<td>MV, CHAR, D8, 15, RECD</td>
</tr>
<tr>
<td>SQ</td>
<td>TABLE</td>
<td>MV, CHAR, D9, 15, RECD</td>
</tr>
<tr>
<td>SR</td>
<td>TABLE</td>
<td>MV, CHAR, E1, 15, RECD</td>
</tr>
<tr>
<td>ST</td>
<td>TABLE</td>
<td>MV, CHAR, E2, 15, RECD</td>
</tr>
<tr>
<td>SU</td>
<td>TABLE</td>
<td>MV, CHAR, E3, 15, RECD</td>
</tr>
<tr>
<td>SV</td>
<td>TABLE</td>
<td>MV, CHAR, E4, 15, RECD</td>
</tr>
<tr>
<td>SW</td>
<td>TABLE</td>
<td>MV, CHAR, E5, 15, RECD</td>
</tr>
<tr>
<td>SX</td>
<td>TABLE</td>
<td>MV, CHAR, E6, 15, RECD</td>
</tr>
<tr>
<td>SY</td>
<td>TABLE</td>
<td>MV, CHAR, E7, 15, RECD</td>
</tr>
<tr>
<td>SZ</td>
<td>TABLE</td>
<td>MV, CHAR, E8, 15, RECD</td>
</tr>
<tr>
<td>S0</td>
<td>TABLE</td>
<td>MV, CHAR, E9, 15, RECD</td>
</tr>
<tr>
<td>S1</td>
<td>TABLE</td>
<td>MV, CHAR, F0, 15, RECD</td>
</tr>
<tr>
<td>S2</td>
<td>TABLE</td>
<td>MV, CHAR, F1, 15, POSTST</td>
</tr>
<tr>
<td>S3</td>
<td>TABLE</td>
<td>MV, CHAR, F2, 15, RECD</td>
</tr>
<tr>
<td>S4</td>
<td>TABLE</td>
<td>MV, CHAR, F3, 15, RECD</td>
</tr>
<tr>
<td>S5</td>
<td>TABLE</td>
<td>MV, CHAR, F4, 15, RECD</td>
</tr>
<tr>
<td>S6</td>
<td>TABLE</td>
<td>MV, CHAR, F5, 15, RECD</td>
</tr>
<tr>
<td>S7</td>
<td>TABLE</td>
<td>MV, CHAR, F6, 15, RECD</td>
</tr>
<tr>
<td>S8</td>
<td>TABLE</td>
<td>MV, CHAR, F7, 15, RECD</td>
</tr>
<tr>
<td>S9</td>
<td>TABLE</td>
<td>MV, CHAR, F8, 15, RECD</td>
</tr>
<tr>
<td>SEGL</td>
<td>TABLE</td>
<td>MV, CHAR, F9, 15, RECD</td>
</tr>
<tr>
<td>SKARAT</td>
<td>TABLE</td>
<td>MV, CHAR, FE, 15, RECD</td>
</tr>
<tr>
<td>SLBRAC</td>
<td>TABLE</td>
<td>MV, CHAR, EO, 15, POSTST</td>
</tr>
<tr>
<td>SLKRT</td>
<td>TABLE</td>
<td>MV, CHAR, EE, 15, RECD</td>
</tr>
<tr>
<td>SLPAR</td>
<td>TABLE</td>
<td>MV, CHAR, EB, 15, RECD</td>
</tr>
<tr>
<td>SMINUS</td>
<td>TABLE</td>
<td>MV, CHAR, CE, 15, RECD</td>
</tr>
<tr>
<td>SPER</td>
<td>TABLE</td>
<td>MV, CHAR, CB, 15, RECD</td>
</tr>
<tr>
<td>SPLUS</td>
<td>TABLE</td>
<td>MV, CHAR, CD, 15, RECD</td>
</tr>
<tr>
<td>SPOUND</td>
<td>TABLE</td>
<td>MV, CHAR, CC, 15, RECD</td>
</tr>
<tr>
<td>SRBRAC</td>
<td>TABLE</td>
<td>MV, CHAR, DD, 15, RECD</td>
</tr>
<tr>
<td>SRKRT</td>
<td>TABLE</td>
<td>MV, CHAR, DC, 15, RECD</td>
</tr>
<tr>
<td>SRPAR</td>
<td>TABLE</td>
<td>MV, CHAR, DD, 15, POSTST</td>
</tr>
<tr>
<td>SSCRUB</td>
<td>TABLE</td>
<td>MV, CHAR, CB, 15, RECD</td>
</tr>
<tr>
<td>SSLASH</td>
<td>TABLE</td>
<td>MV, CHAR, CD, 15, RECD</td>
</tr>
<tr>
<td>SSTAR</td>
<td>TABLE</td>
<td>MV, CHAR, DD, 15, RECD</td>
</tr>
<tr>
<td>STILCA</td>
<td>TABLE</td>
<td>MV, CHAR, DO, 15, RECD</td>
</tr>
<tr>
<td>Command</td>
<td>Address</td>
<td>Value</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>DK</td>
<td>15, MV, CHAR, EF, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>SBOX</td>
<td>15, MV, CHAR, 73, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>SCIRC</td>
<td>15, MV, CHAR, 74, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>SELIPS</td>
<td>15, MV, CHAR, 76, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>SPBCX</td>
<td>15, MV, CHAR, 77, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>STRAP</td>
<td>15, MV, CHAR, 78, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>STRI</td>
<td>15, MV, CHAR, 75, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSA</td>
<td>15, MV, CHAR, 81, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSB</td>
<td>15, MV, CHAR, 82, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSC</td>
<td>15, MV, CHAR, 83, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSD</td>
<td>15, MV, CHAR, 84, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSE</td>
<td>15, MV, CHAR, 85, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSF</td>
<td>15, MV, CHAR, 86, 15, RAZER</td>
<td></td>
</tr>
<tr>
<td>RSG</td>
<td>15, MV, CHAR, 87, 15, RAZER</td>
<td></td>
</tr>
<tr>
<td>RSH</td>
<td>15, MV, CHAR, 88, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSI</td>
<td>15, MV, CHAR, 89, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSJ</td>
<td>15, MV, CHAR, 91, 15, RAZER</td>
<td></td>
</tr>
<tr>
<td>RSK</td>
<td>15, MV, CHAR, 92, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSL</td>
<td>15, MV, CHAR, 93, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSM</td>
<td>15, MV, CHAR, 94, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSN</td>
<td>15, MV, CHAR, 95, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSG</td>
<td>15, MV, CHAR, 96, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSP</td>
<td>15, MV, CHAR, 97, 15, RAZER</td>
<td></td>
</tr>
<tr>
<td>RSQ</td>
<td>15, MV, CHAR, 98, 15, RAZER</td>
<td></td>
</tr>
<tr>
<td>RSR</td>
<td>15, MV, CHAR, 99, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSS</td>
<td>15, MV, CHAR, A2, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RST</td>
<td>15, MV, CHAR, A3, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSU</td>
<td>15, MV, CHAR, A4, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSV</td>
<td>15, MV, CHAR, A5, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSW</td>
<td>15, MV, CHAR, A6, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RXS</td>
<td>15, MV, CHAR, A7, 15, RECD</td>
<td></td>
</tr>
<tr>
<td>RSY</td>
<td>15, MV, CHAR, A8, 15, RAZER</td>
<td></td>
</tr>
<tr>
<td>RSZ</td>
<td>15, MV, CHAR, A9, 15, RAZER</td>
<td></td>
</tr>
</tbody>
</table>

---

**PAD Table**

---

<table>
<thead>
<tr>
<th>Command</th>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PADT</td>
<td>15, DK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15, SDP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15, STP6</td>
<td>02</td>
</tr>
<tr>
<td></td>
<td>15, STP5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>15, S3MBR1</td>
<td>04</td>
</tr>
<tr>
<td></td>
<td>15, SCMG</td>
<td>05</td>
</tr>
<tr>
<td></td>
<td>15, KVXYM</td>
<td>06</td>
</tr>
<tr>
<td></td>
<td>15, SETXX</td>
<td>07</td>
</tr>
<tr>
<td></td>
<td>15, SECG6M</td>
<td>08</td>
</tr>
<tr>
<td></td>
<td>15, SOMCQ8</td>
<td>09</td>
</tr>
<tr>
<td></td>
<td>15, G6ETST</td>
<td>0A</td>
</tr>
<tr>
<td></td>
<td>15, S2MRZ</td>
<td>0B</td>
</tr>
<tr>
<td>TABLE</td>
<td>Name</td>
<td>Value</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>15,SPRMA</td>
<td>OC</td>
<td></td>
</tr>
<tr>
<td>15,SBARMK</td>
<td>0D</td>
<td></td>
</tr>
<tr>
<td>15,S3MB</td>
<td>0E</td>
<td></td>
</tr>
<tr>
<td>15,SUMAM</td>
<td>0F</td>
<td></td>
</tr>
<tr>
<td>15,SRRM</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>15,SRRM</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>15,SSM</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15,SCOMAM</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>15,SAMST1</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>15,SCMEG</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>15,S7MGK</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>15,XMK1</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>15,SLMAK</td>
<td>1A</td>
<td></td>
</tr>
<tr>
<td>15,SLMEK4</td>
<td>1B</td>
<td></td>
</tr>
<tr>
<td>15,STPE</td>
<td>1C</td>
<td></td>
</tr>
<tr>
<td>15,SVM</td>
<td>1E</td>
<td></td>
</tr>
<tr>
<td>15,SVM</td>
<td>1F</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>15,TST5</td>
<td>37</td>
<td></td>
</tr>
</tbody>
</table>

* * *

****4 DIRECTION TABLE****  
* *
AAAA  TABLE /SS, TEMP, CO/0, C, AAA
AAA  EQU *
TABLE 15, RECC
TABLE 15, S2PM
TABLE 15, S23MB
TABLE 15, S23MBP
TABLE 15, SMC
TABLE 15, S8M
TABLE 15, SG069M
TABLE 15, SS589M
TABLE 15, SGC6M
TABLE 15, SBDPR
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, S1MA
TABLE 15, SMLC
TABLE 15, SCCMAM
TABLE 15, SBARMK
TABLE 15, SS
TABLE 15, S9MK
TABLE 15, SCM3G
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, SGSCRB
TABLE 15, S3SCR8
TABLE 15, BR
TABLE 15, SBDPRI
TABLE 15, SLKRTM
TABLE 15, SRPRMJ  2E
TABLE 15, SGS
| TABLE  15,SGSC6M |
| TABLE  15,SCMQ8 |
| 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,SC23MB 3B |
| TABLE  15,SO9   3C |
| TABLE  15,S9LC1 |
| TABLE  15,SO9M  |
| 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,XF1,0/    |
| KVVXY   | TABLE /EX,XKVVXY,0/ |
| MW      | TABLE /EX,XMW,0/    |
| MWIN    | TABLE /EX,XMWIN,0/  |
| MW1     | TABLE /EX,XMW1,0/   |
| PCSTST  | TABLE /EX,XPCST,0/  |
| RAZER   | TABLE /EX,XRAZE,0/  |
| RECD    | TABLE /EX,XRECD,0/  |
| SCP     | TABLE /EX,XSCP,0/   |
| SMNW    | TABLE /EX,XSMNW,0/  |
| SM1M    | TABLE /EX,XSM1M,0/  |
| SPRM    | TABLE /EX,XSPRM,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/ |
SBLCVX TABLE /EX, XSBLCV, 0/
SRLCX TABLE /EX, XSRLC, 0/
S4MK1X TABLE /EX, XS4MK1, 0/

****ENTRY SWITCH****

GPSW TABLE /SS, CUSP, CO, 0, GPSWITCH

GPSWITCH EQU *
TABLE 15, SPER
TABLE 15, SXBL
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, SGMCQ8
TABLE 15, SCPN8U
TABLE 15, RSB
TABLE 15, RSC
TABLE 15, RSE
TABLE 15, RSF
TABLE 15, RSG
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, 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, SG
RECORD RCS'S

AHSTRI

*FUNCTION
*
*DISTINGUISHES AMONG 3-STROKE A, H, K, AND * BASED ON POSITIONS OF
*STARTING AND ENDING POINTS
*
*
*CALL
* RCS AHSTRI A,ECHAR
*EXIT CHAR WHEN A CHARACTER IS RECOGNIZED
*
*
*INPUT REGISTER. R6
*
*INTERNAL REGISTERS. R7, R9-R14
*
*
USING XR6,R6
EXO EQU C
REGS
D6 DSECT
XR6 CS 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'
CS 3F
CS 26H
CS 20C
CS 3F
CS 1H
DS 2C
DS 3H
P DS 1C
CHAR CS 1C
AHSTRI BCX
*R10 IS FIRST VERTICAL STROKE REF
*R11 IS SECOND VERTICAL STROKE REF
*R12 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 R1C,O(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 R1C,2(R6)
HDONE EQU *
*TEST FOR K
*ARE BOTH VERT ENDPNTS 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 ENCPNT IN UPPER RIGHT?
CLI XYEP+1(R12),X'00'
BC  8,ETCP
CLI  XYPE+1(R12),X'01'
BC  8,ETCP
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  ETCP1
ETCP  EQU  *
     LH  R13,YEPI(R12)
ETCP1 EQU  *
     CH  R13,YSPI(R11)
     BC  2,SKX
NOTK EQU  *

*NOT K, TEST FOR A,H, OR *
*ARE START PTS CLOSE COMPARED TO ENDPTS
     LH  R13,XEPI(R10)
     SH  R13,XEPI(R11)
     LPR  R13,R13
     SRL  R13+2
     LH  R14,XSPI(R10)
     SH  R14,XSPI(R11)
     LPR  R14,R14
     CR  R14,R13
     BC  4,SAX

*NG, DO VERTICAL STROKES CROSS?
     LH  R13,XEPI(R10)
     CH  R13,XEPI(R11)
     BC  2,ENDIR
     LH  R13,XSPI(R10)
     CH  R13,XSPI(R11)
     BC  2,SSTARX
     B  SHX
ENDIR 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  *
MVI CHAR,X'0C'
MVI P,X'0C'
BEXIT EX0
ENC

BFI

*FUNCTION
*
*DISTINGUISHES AMONG 3-STROKE F, I, AND * BASED ON POSITIONS OF STARTING 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
06 DSECT
XR6 CS 0F
XYSP EQU X'4A'
XYS (R6)
DS 3F
DS 26H
DS 20C
DS 3F
DS 1H
DS 2C
DS 3H
DS 1C
CHAR CS 1C
BFI BX0
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'0DC'
BC 8,FIX
MVI CHAR,C'I'
TM XYP+1(R7),X'0CC'
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
*{CHARLOC SETS DYM = 3/2 NORMAL CHARACTER HEIGHT}.
*
*
*
*CALL
RCS BHITEX,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
C6 DSECT
XR6 CS 0F
DS 3F
DS 20H
YTC CS 1H
YBC DS 1H
DS 4H
DS 20C
DS 3F
DS 1H
BSDP

*FUNCTION
*DISTINGUISHES AMONG 'C, 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)
CS 1F
PAD CS 1F
DS 1F
SN	CS	11H
CS	1H
CS	14H
CS	20C
CS	3F
CS	1H
CS	2C
CS	3H
P	CS	1C
CHAR	CS	1C
CS	54C
CS	21H
CS	1F
CS	6C
CS	1H
CS	1F
CS	20C
CS	33H
GYMAX	CS	10C
BSDP	BCX
CLI	PAD+3,X*01'
BC	8,SDM51
MVI	PAD+3,X*01'
SCP	LH	R15,SN
BCT	R15,SDP1
SDP1	SLL	R15,1
LA	R7,0(R6,R15)
TM	XYPE+1(R7),X*0C'
MVI	CHAR,C*P'
BC	12,SDPX
SDM5	MVI	P,X*00'
MVI	CHAR,C*C'
CLI	SN+1,X*02'
BC	8,SDSD
CLI	GYMAX+1,X*00'
BC	8,SDSD
MVI	CHAR,X*82'
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  BSMNW, ECHAR, EGRCUP, 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
c6  CSECT
xr6  CS  0F
     CS  3F
     CS  10H
n  CS  1H
     CS  5H
cxc  CS  1H
     CS  5H
aspr  CS  1H
     DS  3H
     CS  20C
     DS  3F
     CS  1H
     CS  2C
     CS  3H
     DS  1C
char  CS  1C
     CS  6C
xsp  DS  10C
     DS  38C
     CS  21H
     DS  1F
CS 6C
CS 1H
CS 1F
CS 20C
CS 33H

GYMAX  DS 10C
GYMIN  EQU  QYMAX+5
DS 2H
YMAXX  DS 10H

BSMNW  BOX

SMNW  CLI  N+1,X*C5'
BC 4,TEST4
BC 8,SMLCX
MVI CHAR,X*72'
B  SMNW

TEST4  CLI  ASPR+1,X*04'
MVI CHAR,C*W'
BC 4,SMNW

*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
LH R8,YMAXX+2
SH R8,XSP
LPR R8,R8
CR R8,R7
BC 4,SNLCY
MVI CHAR,C*W'
B  SMNW

SNLCY  MVI CHAR,C*N'
CLI QYMIN,X*C3'
BC 8,SMNW
MVI CHAR,X*A8'
BEXIT EX8

SMNW  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
*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 CS 0F
      CS 1F
PAD DS 1F
      DS 1F
      DS 20H
YTC CS 1H
YBC CS 1H
      CS 4H
      CS 20C
      CS 3F
      CS 1H
      CS 2C
NCUSP DS 1H
      DS 2H
P DS 1C
CHAR CS 1C
      CS 54C
      CS 21H
      CS 1F
      DS 6C
      DS 1H
      DS 1F
CS 10C
CS 10C

BSRPRM
BOX
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*03'
MVI CHAR,C*R'
BC 8,SRPRMX
MVI PAD+3,X*10'
CLI NCUSP+1,X*03'
MVI CHAR,C*3'
BC 10,SRPRMX
CLI NCUSP+1,X*01'

RBRAXX
MVI CHAR,X*DF'
BC 2,SRPRMX
MVI CHAR,X*CD'
BC 4,SRPRMX1

*3 IF THE 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,RBRAXX
CH R7,YC
MVI CHAR,C*3'
BC 10,SRPRMX
BC 4,RBRAXX

DPMR
MVI P,X*00'

DPMR1
CLI P,X*01'
MVI CHAR,C*R'
BC 8,SRPRMX
MVI PAD+3,X*11'

BEXIT EX4

SRPRMX BEXIT EX0
SRPRMX1 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 BSMA, ECHAR
*EXIT CHAR WHEN A CHARACTER IS RECOGNIZED
*
*
*INPUT REGISTER. R6
*
*INTERNAL REGISTER. R7
*
*
USING XR6, R6
REGS
EXO EQU 0
C6 CSECT
XR6 CS 0F
   CS 1F
PAD CS 1F
CCDE CS 1F
   CS 10H
N CS 1H
   CS 7H
XRC CS 1H
XLX CS 1H
   CS 6H
XYE CS 10C
   CS 10C
   CS 3F
   CS 1H
   CS 2C
NCUSP CS 1H
   CS 2H
P CS 1C
CHAR CS 1C
TEMP CS 1C
   DS 53C
   DS 21H
   CS 1F
   CS 6C
   CS 1H
   CS 1F
XC          DS  10C  
CS          DS  10C  
CS          DS  17H  
NTCUSP      DS  1H  
BSSM        BOX  
            CLI    PAD+3,X*13'  
            BC     8,COLOCOL  
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'  
            SSM    CODE,X*80'  
            TM    XYE+1,X*08'  
            BC     8,S8S8  
            *NOT 5 IF 1ST ANGLE IS 1  
            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    NTCUSP+1,X*02'  
            BC     2,S5S5  
            BC     4,S5S5  
            LMA    R7,XRC  
            SH    R7,XLC  
            SRL    R7,1  1/2 DELTA X  
            AM    R7,XLC  
            CH    R7,XC+2  
            BC     2,S5S5  
            BC     12,S5S5  
            LL    DS  0H  
            CC    X*00'  
            DC    X*04'  
            CC    X*10'  
            DC    2X*00'  
            CC    3X*08'  
            CC    3X*CC'  
            CC    X*CC'  
            LLL    CS  0H  
            BC     15,SSM1  
            BC     15,STPCOL  
            BC     15,S8S8  
            BC     15,DOLCOL  
            BC     15,S5S5  

*POSSIBLE 9, TEST ANGLES, AND TIME CORNERS
S9
  CLI N+1,X*C6'
  BC 4, SSSS
  CLI NTCUSP+1,X*01'
  BC 4, SSSS
S9S9
  MVI CHAR,C*9'
  B SSMX
STPDOL
  MVI P,X*0C'
  MVI PAD+3,X*13'
DCLDCL
  MVI CHAR,X*DB'
  BC 15, SSMX
DCLLARS
S5S5
  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 EX0
END

BSVM

*FUNCTION
*
*DISTINGUISHES AMONG V, W, AND 5 GROUPS OF CHARACTERS (J, U), (M, W),
*(K, N, Y), (C, 8, O, 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, EM*, EKNY, E08OQ, 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 08OQ WHEN THE CHAR IS C, 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
CS 0F
CS 1F
PAD
CS 1F
DS 1F
CS 18H
XRC
CS 1H
XLC
DS 1H
DS 6H
CS 20C
CS 3F
DS 1H
CS 2C
CS 3H
P
DS 1C
CHAR
CS 1C
TEMP
CS 1C
DS 5C
XSP
DS 10C
CS 10C
XEP
CS 10C
BSVM
BOX
CLI PAC+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
**0 VS U,V TEST**
*0 IF STARTPT AND ENDCPT ARE CLOSER THAN*
*1/2 CHARACTER WIDTH*
*TEMP CONTAINS 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**
CS CF
BC 15, SUJU1X
BC 15, SVM1

SVMX BEXIT EX0
SJMUXX BEXIT EX4
MW1X BEXIT EX8
KNY1X BEXIT EX12
SOMX BEXIT EX16
SUJU1X BEXIT EX20
END

**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
C6 DSECT
XR6 CS 0F
XYEP EQU X'40'
XYE CS 3F
CS 11H
SN CS 1H
CS 14H
CS 20C
CS 3F
CS 1H
CS 2C
CS 3H
CS 1C
CHAR CS 1C
BTEST1 BCX
TEST1 LH R15, SN
BCT R15, TEST1
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, 8) based on the no. of strokes, the positions of starting and ending
*points, the directions, the positions of rel. y maxima
*
* CALL
* RCS BTEST3A, ECHAR, ESB
* 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   CS  0F
XYSP  EQU X'A
XYEP  EQU X'B
CODE  CS  1F
SN   CS  1H
CS   4H
CXC   CS  1H
CS   1H
XRC   CS  1H
CS   7H
XYE   CS  10C
CS   10C
CS   3F
CS   1H
CS   2C
CS   3H
CS   1C
CHAR  CS  1C
CS   54C
CS   21H
CS   1F
CS   6C
CS   1H
CS   1F
CS   20C
CS   33H
QYMAX CS  10C
CS   2H
YMAXX CS  1CH
BTEST3 BOX
TEST3 CLI SN+1, X'01'
BC 8, TEST31
*2 STROKE CHARACTERS
LH    R8,SN
BCT   R8,TEST32
TEST32 SLL  R8,1
LA    R8,U(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,Y*XCC*        3130
BC    8,RU
CLI   QYMAYX+1,X*00*
BC    8,TEST3X
MVI   CHAR,X*92*        K
B     TEST3X
RU    EQU  *
LH    R8,DXC
SRL   R8,1
LH    R7,XGC
SR    R7,R8
* IS MAX 2 IN RIGHT 1/2
CH    R7,QMAYX+2
BC    2,TEST3X        NO,R
MVI   CHAR,C*MU*
TEST3X BEXIT EXC
SBM5X BEXIT EX4
END

KNYHTST

*FUNCTION
*
*DISTINGUISHES AMONG 3-STROKE (ALL VERT) K, N, AND Y BASED ON THE POSITIONS 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
EXO EQU 0
REGS
D6 DSECT
XR6 CS CF
XYP EQU X'4A'
XYPE EQU X'40'
CS 3F
CS 26H
XYE CS 1C
XYS CS 1C
CS 3F
CS 1H
CS 2C
CS 3H
CS 1C
CHAR CS 1C
KNYTST BX
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 XYP+1(R1, R8), HHS
LH R7, XYS(R15)
EX 0, HHHE(R7)
KNSN CR R9, R12
BC 8, SNX
LR R9, R12
KNSKY TR XYE+1(R1, R8), HHE
LH R7, XYE(R15)
EX 0, HHHE(R7)
KNEY CR R10, R12
BC 8, SYX
LR R10, R12
BC 15, KNYI
KNEYEN CR R11, R12
BC 8, SNX
LR R11, R12
KNIYI   BXLE  R15,R12,KNIYIN
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 EXC
^HHS   DS  OF
       BC  15,KNYSKY
       BC  15,KNYSN
^HHE   DS  OF
       BC  15,KNIY
       BC  15,KNYEY
       BC  15,KNYEN
HHS    CS  0H
       CC  2X'0C'
       CC  2X'04'
       CC  3X'0C'
       CC  X'04'
       CC  8X'0C'
HHE    CS  0H
       CC  5X'00'
       CC  2X'04'
       CC  X'0C'
       CC  X'08'
       CC  2X'04'
       CC  X'00'
       CC  2X'08'
       CC  X'00'
END

KNIYI

*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  KNIYITA,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 CSECT
XR6 DS 0F
XYEP EQU X'40'
XYSP EQU X'4A'
CS 3F
CS 26H
XYE DS 10C
XYS DS 10C
CS 3F
CS 1H
CS 2C
CS 3H
CS 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'
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
C6    DSECT
XR6    DS 0F
XYEP EQU X'40'
XYSP EQU X'4A'
CS 3F
DC 26H
XYE CS 10C
CS 10C
DC 3F
DC 1H
DC 2C
DC 3H
DC 1C
CHAR CS 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)
KKXY TM XYSP + 1(R8), X'C3'
BC 1, KKVXY
BC 12, KVXY2
KVXY3 CR R12, R13
BC 8, KKVXY
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 EXC
EEE DC 0H
DC X'08'
DC X'08'
CC X'08'
DC X'08'
CC X'08'
DC X'08'
CC X'08'
DC X'08'
CC X'08'
DC X'08'
CC X'08'
DC X'08'
CC X'08'
DC X'08'
CC X'08'
DC X'08'
DC X'08'
DC X'0C'
DC X'04'
DC 2X'OC'
DC X'00'

EEEE
CS 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 MWTA,ECHAR
*EXIT CHAP 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 STRCKES - 1
*
*
*INTERNAL REGISTERS R8, R9, R11, R12, R15
*
*
**USING XR6,R6
EX0 ECU 0
REGS
DC DSECT
XR6 DS OF
XYEP ECU X'40'
DS 3F
CS 26H
XYE CS 10C
DS 10C
DS 3F
DS 1H
DS 2C
DS 3H
DS 1C
CHAR DS 1C
MWT BX
SR R15,R15
SR R9,R9 J
SR R11,R11 K
LA R12,1
MW1IN SLL R15,1
LA R8,0(R6,R15)
TR XYEP+1(1,R8),GGE
LH R8,XYE(R15)
EX 0,GGE(R8)
MW1IQ LA R9,1(C,R9) J=J+1
BC 15,MW1I
MW13Q LA R11,1(C,R11)
MW1I SRL R15,1
BXLE R15,R12,MW1IN
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 BX*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,MW1IQ
BC 15,MW1I
BC 15,MW13Q
END
*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 NCRM 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
* (NORMA L 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
EXO EQU 0
REGS
C6 DSECT
XR6 DS OF
CS 3F
CS 17H
DYC CS 1H
CS 2H
YTC CS 1H
CS 5H
CS 20C
WIDTH CS 1H
HEIGH T CS 1H
DS 2F
CS 1H
CS 2C
CS 3H
CS 1C
CHAR CS 1C
CS 54C
CS 14H
DYM DS 1H
CS       6H
CENT      CS       1F
PTEST       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 EX0
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
CCDE 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
D4   DS  2H
D7   DS  1H
D8   DS  1H
C9 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,DN
    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,BOXTIRI 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,DC
CR R7,R12
BC 10,RIGHT
LH R7,D8
CR R7,R12
BC 4,ROUND

*R7 HAS NC. OF RIGHTS OR LEFTS

*FIND DOWNWARD DIRECTION
RIGHT LA R9,20
LA R10,2
DOWN LH R8,D0(R9)
LA R11,2C
AH R8,D0-2(R9)
CR R8,R12
BC 10,DOWNX
BXLE R9,RIC,DCWN

*NO SUC H DOWNSWARD DIRECTION
BC 15,ROUND

*R9 CON TAINS DOWNSWARD DIRECTION CODE

*R8 CON TAINS NO. OF DOWNSWARDS

*FIND UPWARD 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
*COWN 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
*COWN D  IR. IS 13 OR 14
*UP DIR  R. MUST BE 2, 3, OR 4
DGTR12  LA  R9, 4
   LA  R11, 8
*FIND UP DIRECTION
UP  LH  R8, DO(R9)
   AH  R8, DO(R9)
   CR  R8, R12
   BC  10, UPX
   BXLE R9, R10, UP
*NO SUCH H UPWARD DIRECTION
   BC  15, ROUND
*R7 CON TAINS NO. CF HORIZ. ? DOWNS
*R8 CON TAINS NO. OF UPWARDS
*IS TOT AL HORIZ, UPS, AND DOWNS
*GREATER R THAN 3/4 DN?
   UPX
   AR  R7, R8
   CR  R7, R13
   BC  12, NOTSQ
*TRIANGLE, TRAPEZOID, OR ELLIPSE
*TRIANGLE IF HORIZ NOT GTR 0.375 DN
   LH  R7, DO
   AH  R7, O8
   LR  R15, R13
   SRL R15, 1
   CR  R7, R15
   BC  12, TRIX
*CHECK TIME CORNERS FOR TRAP
   CLI  NTCSUP+1, X*O2'
   BC  2, TRAPXX
   B  ELPXS
*SYMBOL NCT BOX OR TRIANGLE
*TEST F OR CIRCLE OR ELLIPSE
*CRT TRAPEZOID
*4-ANGLE SEQUENCE MUST BE
*0-3-2- 1 OR 2-3-0-1
   ROUND  CLI  CODE, X*O1'
   BC  8, OKSYM
   CLI  CODE, X*39'
   BC  8, OKSYM
*3-2-1-  C
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,OKSYM
*1-2-3-  0
CLI  CODE,X'6C'
BC  8,OKSYM

*IS THIS 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'C4'
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'C2'
BC  12,XELPS
LA  R10,128
BC  15,PBCXX

*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 EX0
DECIDE BETWEEN CIRCLE AND ELLIPSE

*AND TRAPEZOCIC

CKSYM   CLI NTCSNP+1,X'02'
BC     2,TRAPXX
LH     R7,DC
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 FOR CLOSENESS OF ENDOPTS

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,DC
BC     2,NOSYMX
LA     R9,128
CR     R9,R10
BC     8,TRAPXX

XXPBOX  EQU *
MVI    CHAR,X'77'

BEXIT7  BEXIT EX4

ENC
TILDT

*FUNCTION
*
*RECOGNIZES TILDA BASED ON CHAR. HEIGHT, ASPECT RATIO, AND FIRST FOUR
*DIRECTIONS.
*ALTHOUGH THIS ROUTINE ENCOPRATES 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
C6 DSECT
XR6 DS 0F
DS 2F
CCDE DS 1F
DS 17H
CYC 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
CYM DS 1H
TILDT BOX
LH R7, DYM
SRL R7, 2 1/4 DYM
CH R7, DYC
BC 4, NOTTIL
*CYC LE SS THAN 1/4 CYM
CLI APR+1, X'02'
* ASPEC
  T RATIO IS BETWEEN 1/2 AND 1/4
*TEST F OR ALLOWABLE SEQUENCES
*0-0-0-  0
  CLI   CODE,X'30'
  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'30'
  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'CO'
BEXIT EX4
END
TPXY

*FUNCTION
*
*DISTINGUISHES AMONG 2-STROKE (1 VERT, 1 HORIZ) T, X, Y, AND PLUS BASED
*CN 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
*
*
USING XR6, R6
EXO EQU 0
REGS
C6 DSECT
XR6 DS OF
    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'0C2'
BC 8,YES
LA R9,2(R6) VERT REF
B GO
YES LA R9,0(R6) VERT REF
GC 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
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX 0,TTEE(R8)</td>
<td></td>
</tr>
<tr>
<td>TPLUS4</td>
<td>TM XYSPL+1(R7),X<em>CC</em></td>
</tr>
<tr>
<td>BC 8, TPLUS1</td>
<td>YS GTR 3/4 DELTA Y</td>
</tr>
<tr>
<td>BC 1, TPLUS2</td>
<td>YS LESS 1/4 DELTA Y</td>
</tr>
<tr>
<td>*START IN MIDDLE Y</td>
<td></td>
</tr>
<tr>
<td>TM XYSPL+1(R7),X<em>CB</em></td>
<td></td>
</tr>
<tr>
<td>BC 1, TPLUS5</td>
<td></td>
</tr>
<tr>
<td>TM XYSP+1(R7),X<em>CC</em></td>
<td></td>
</tr>
<tr>
<td>BC 9, TPLUS2</td>
<td></td>
</tr>
<tr>
<td>BC 4, PTPXY</td>
<td></td>
</tr>
<tr>
<td>*START IN LOWER MID Y</td>
<td></td>
</tr>
<tr>
<td>TPLUS5</td>
<td>TM XYSP+1(R7),X<em>OC</em></td>
</tr>
<tr>
<td>BC 1, TPLUS2</td>
<td></td>
</tr>
<tr>
<td>BC 12, PTPXY</td>
<td></td>
</tr>
<tr>
<td>*START AT TCP</td>
<td></td>
</tr>
<tr>
<td>TPLUS1</td>
<td>TM XYSP+1(R7),X<em>CC</em></td>
</tr>
<tr>
<td>BC 8, TTPXY</td>
<td></td>
</tr>
<tr>
<td>BC 4, TPLUS3</td>
<td></td>
</tr>
<tr>
<td>*END AT BOTTOM</td>
<td></td>
</tr>
<tr>
<td>TM XYSP+1(R7),X<em>03</em></td>
<td></td>
</tr>
<tr>
<td>BC 1, TPLUSX LEFT</td>
<td></td>
</tr>
<tr>
<td>BC 8, XTPXY RIGHT</td>
<td></td>
</tr>
<tr>
<td>BC 4, TPLUS2 MIDDLE</td>
<td></td>
</tr>
<tr>
<td>*START AT TCP, END IN MIDDLE Y</td>
<td></td>
</tr>
<tr>
<td>*IS END IN RIGHT MID Y?</td>
<td></td>
</tr>
<tr>
<td>TPLUS3</td>
<td>TM XYSP+1(R7),X<em>03</em></td>
</tr>
<tr>
<td>BC 5, TPLUS2</td>
<td></td>
</tr>
<tr>
<td>*YES</td>
<td></td>
</tr>
<tr>
<td>TM XYSP+1(R7),X<em>80</em></td>
<td></td>
</tr>
<tr>
<td>BC 1, XTPXY</td>
<td></td>
</tr>
<tr>
<td>CR R12, R13</td>
<td></td>
</tr>
<tr>
<td>BC 8, YTPXY</td>
<td></td>
</tr>
<tr>
<td>BC 6, TPLUS2</td>
<td></td>
</tr>
<tr>
<td>TPLUSX</td>
<td>LR R12, R13</td>
</tr>
<tr>
<td>TPLUS2</td>
<td>BXLE R15, R13, TPLUS1</td>
</tr>
<tr>
<td>CR R12, R13</td>
<td></td>
</tr>
<tr>
<td>BC 8, XTPXY</td>
<td></td>
</tr>
<tr>
<td>TTPXY EQU *</td>
<td></td>
</tr>
<tr>
<td>*IS VERT START IN UPPER LEFT</td>
<td></td>
</tr>
<tr>
<td>CLI XYSPL+1(R9),X<em>CO</em></td>
<td></td>
</tr>
<tr>
<td>BC 8, YTPXY</td>
<td></td>
</tr>
<tr>
<td>MVI CHAR, C<em>T</em></td>
<td></td>
</tr>
<tr>
<td>BC 15, BEXIT3</td>
<td></td>
</tr>
<tr>
<td>PTPXY EQU *</td>
<td></td>
</tr>
<tr>
<td>*IS VERT START IN UPPER LEFT</td>
<td></td>
</tr>
<tr>
<td>CLI XYSPL+1(R9),X<em>CO</em></td>
<td></td>
</tr>
<tr>
<td>BC 8, YTPXY</td>
<td></td>
</tr>
<tr>
<td>MVI CHAR, X<em>CE</em></td>
<td></td>
</tr>
<tr>
<td>BC 15, BEXIT3</td>
<td></td>
</tr>
<tr>
<td>XTPXY EQU *</td>
<td></td>
</tr>
<tr>
<td>MVI CHAR, C<em>X</em></td>
<td></td>
</tr>
</tbody>
</table>
YTPXY EQU *
       MVI CHAR,C'Y'
BEXIT3 BEXIT EX0
TTE  CS 0H
     CC X'00'
     CC X'08'
     CC X'10'
     CC X'04'
     CC X'0C'
     CC X'10'
     CC X'04'
     CC X'08'
     CC X'04'
     CC X'00'
     CC 3X'10'
TTTE CS 0F
     BC 15,YTPXY
     BC 15,YTPXY
     BC 15,YTPXY
     BC 15,XTPXY
     BC 15,TPPL54
     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,EV1,EV2,EV1H1,EV1NOT
*EXIT V1 WHEN THERE IS ONLY ONE VERTICAL STROKE (THE MOST RECENT)
*EXIT V2 WHEN THERE ARE 2 VERTICAL STROKES
*EXIT V1H1 WHEN THERE IS 1 VERT STROKE AND 1 HORIZ STROKE
*EXIT V1NOT 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 CS 0F
CS 3F
CS 11H

SN CS 1H
DS 14H
CS 20C
CS 3F
CS 1H
CS 2C
CS 3H

P CS 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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>81</td>
<td>+</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>82</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>83</td>
<td>=</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>84</td>
<td>/</td>
</tr>
<tr>
<td>E</td>
<td>E</td>
<td>85</td>
<td>)</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>86</td>
<td>*</td>
</tr>
<tr>
<td>G</td>
<td>G</td>
<td>87</td>
<td>$</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>88</td>
<td>.</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>89</td>
<td>'</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
<td>91</td>
<td>'</td>
</tr>
<tr>
<td>K</td>
<td>K</td>
<td>92</td>
<td>#</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>93</td>
<td>[</td>
</tr>
<tr>
<td>M</td>
<td>M</td>
<td>94</td>
<td>]</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
<td>95</td>
<td>&lt;</td>
</tr>
<tr>
<td>O</td>
<td>O</td>
<td>96</td>
<td>&gt;</td>
</tr>
<tr>
<td>P</td>
<td>P</td>
<td>97</td>
<td>^</td>
</tr>
<tr>
<td>Q</td>
<td>Q</td>
<td>98</td>
<td>_</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>A2</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>A3</td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>U</td>
<td>A4</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>V</td>
<td>A5</td>
<td></td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>A6</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>A7</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>Y</td>
<td>A8</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>Z</td>
<td>A9</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Numbers Hex Number Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Special Symbol Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Erasure 72</td>
</tr>
<tr>
<td>Cannot Interpret EF</td>
</tr>
</tbody>
</table>
REGISTER ASSIGNMENT

Registers are referred to as RO, 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 BCX
&LABEL CSECT
USING *,R2
MEND
*
CLEAR

*FUNCTION
*
*PARALLEL PROCESS SYNCHRONIZER. NULLIFIES THE ADVENT OF 'WATE' AND/CR
**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
ANCP
.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 ('ESTATE' EQ 'S') .B
SVC RETURN
MEXIT
.B
ANCP
LA R6, &PSW
LA R7, &ENTER
SVC RETSUP
MEND

INST (Instance)

*FUNCTION
*
*
*GENERATES THE CALLING SEQUENCE FOR A RE-ENTRANT PROCESS
*
*
*
*MACRO DEFINITION
*
MACRO

 &A11, &A12, &A13, &A14, &A15, &A16, &A17, &A18, &A19, &A20, &A21, &A22,
 &A23, &A24, &A25, &A26, &A27, &A28, &A29, &A30, &A31, &A32, &A33,
 &A34, &A35, &A36, &A37, &A38, &A39, &A40, &A41, &A42, &A43, &AX
 &A44, &A45, &A46
LCLA &AL1, &AL2, &AL3, &AL4
LCLC &CG1, &CG2, &CG3

&LABEL LA R4, &CNTX
LA R5, &LOCN
SVC FORMAL

&CG3 SETC .
&AL1 SETA 2
&AL2 SETA 6
&AL3 SETA 1
.A ANCP
&AL1 SETA &AL1+1
&AL2 SETA &AL2+1
&CG1 SETC 'SYSLIST(&AL1)((1,1))
&CG2 SETC 'SYSLIST(&AL1)*CG3*(2,8)
AIF ('&CG1' NE 'E') +E
&AL3 SETA C
AIF (&AL1 GT 3) .G
.F ANCP
&AL4 SETA &AL1-3
&AL4 SETA &AL4*4
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 SVC PARLEL
B &LOW
LH R5,10(R2)
BCT R5,**+4
SLL R5,+2
L R5,0(R5,R1)
AIF ('&STATE' NE 'O').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
*
&LABEL 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 PROCSS &CLEAR=3,&CNTX=3,&AUTO=0,&ID=80000000,&PRCLG=0
&LABEL CSECT
USING  *,R2
LM    R2,R3,4(R4)
B &PRCLG
CC    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 PROCSS &AUTO=YES,&STATE=0,&PSG=0,&LINK=0
AIF ('&AUTOC' EQ 'C').A
&LABEL CS  OH
LR    R1,R4
AIF ('&STATE' EQ 'O').B
LA    R4,&PSG
LA    R5,&LINK
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
&EQ 'O').C
SVC SUPER
&NAME MEXIT &NAME MEXIT &C MEXIT MEND

LCLA &AL1,&AL2,&AL3
LCLC &CG1,&CG2,&CG3
&NAME DS OH
&AL1 SETA 1
&AL3 SETA 0
&CG3 SETC *
&D ANCP
&AL1 SETA &AL1+1
&AL2 SETA &AL1+4
&AL3 SETA &AL3+1
AIF ('&ESYSLIST(&AL1)' EQ '').A
&CG1 SETC '&ESYSLIST(&AL1)'(1,1)
&CG2 SETC '&ESYSLIST(&AL1). &CG3'(2,8)
AIF ('&CG1' EQ 'E').C
AIF ('&CG1' EQ 'I').B
L &AL2.&CG2
AGOB .D
.B LA &AL2.&CG2
AGOB .D
.A AIF (&AL3 EQ 15).C
AGOB .D
.C     L    R2, &LABEL
       BALR   R5, R2  
.Y     B     &CG2
&AL1   SETA   &AL1+1
       AIF    ('ESYSLIST(&AL1)' EQ **).W
&CG2   SETC   'ESYSLIST(&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
*
MACRO

;NAME  REGS
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 OCCURRED
*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*
 80 GS&SYSNDX
 OI C(R5), X*00*
 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
 CCUPIC 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
*
MACRC

&LABEL TABLE EA1, EA2, EA3, EA4, EA5, EA6, EA7, EA8, EA9, EA10, EA11, EA12, EA13, X
 EA14, EA15, EA16, EA17, EA18, EA19, EA20, EA21, EA22, EA23, EA24, X
 EA25, EA26, EA27, EA28, EA29, EA30, EA31, EA32, EA33, EA34, EA35, &AX
 36, EA37, EA38, EA39, EA40, EA41, EA42, EA43, EA44, EA45, EA46, EA4X
 7, EA48, EA49
LCLA &AL1, &AL2
LCLC &CG1, &CG2, &CG3, &CG4, &CG5
AIF ('&LABEL' EQ '').D

&LABEL EQU *
.C 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(' &CG2 &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'
CC &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
CC &CG4&AL2&CG2&SYSLIST(&AL1)&CG3&CG5
AGCB .A
.E ANOP
&CG5 SETC 'AL1('
&AL1 SETA &AL1+1
CC &CG2&CG4&CG3
CC &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
&LABEL AIF (*&CNTX* EQ 'F').A
&LABEL LA R5,&PSG
&LABEL AGC .B
&LABEL ANOP
&LABEL L R5,&PSG
&LABEL TM 0(R5), X*80*
&LABEL BZ GW&SYSNDX
&LABEL NI 0(R5), X*7F*
&LABEL B GW&SYSNDX+2

GW&SYSNDX SVC WAIT "
MEND
REFERENCES


