September 1982

Plaster Source Code

Peter Karp
University of Pennsylvania

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-82-149.

Plaster Source Code

Abstract

This document contains the source code for program Plaster. Before attempting to read this one should become familiar with the *Plaster User's Guide*. The first routine listed below is the main program, which contains an overview of the entire system, many internal and installation notes, and an index of the routines which follow. The remaining user routines called by Plaster follow in alphabetical order. After these come all the Include files referenced by the routines.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-82-149.

Plaster Source Code

Peter Karp

September 1982

Introduction

This document contains the source code for program Plaster. Before attempting to read this one should become familiar with the Plaster User's Guide. The first routine listed below is the main program, which contains an overview of the entire system, many internal and installation notes, and an index of the routines which follow. The remaining user routines called by Plaster follow in alphabetical order. After these come all the Include files referenced by the routines.

```
      Program Plaster

C  ----------------   Commenting conventions  -----------------

C  A specific convention was used in commenting the parameters to the various
C  subroutines in this program.  Each subroutine contains a paragraph
C  describing its function.  If the function of each parameter is not fully
C  described in this paragraph (as is the case for only the simplest of
C  routines), then each parameter is listed, along with certain other
C  information, in the following form:

C  <Param Name> = <type> <useage> - <description>

C  Where:

C  <Param Name> is the name of the variable.

C  <type> describes the data type of the variable.  It can be one of:
C       i  = Default integer
C       i2 = Integer *2
C       i4 = Integer *4
C       r  = Default real
C       r4 = Real *4
C       r8 = Real *8
C       l1 = Logical *1
C       b  = Byte
C       cn = Character variable of length n
C       c* = Variable length character

C  <useage> describes how the parameter is used.  See the book Combinatorial
C       Algorithms by Wilf and Nijenhuis for a precise definition of useage
C       codes.  Simply, the codes are:
C       i  = Input parameter (routine does not alter starting value)
C       o  = Ouput parameter (routine returns a value in this parameter;
C                             its initial value is irrelevant)
C       io = Input/Output parameter (a value is returned, but the initial value
C                                    is relevant to the function of the
C                                    routine)
C       b  = Bookkeeping (routine alters its value, but the calling routine
C                         should never alter its value, which must be
C                         maintained across calls)
C       w  = Work (routine will alter its value, but the calling routine should
C                  not be concerned with the exit value, and may alter the
C                  value)

C  <description> is a verbal description of what the parameter is used for.

C  To save (my) energy, the parameter Exit_Code, which is present in most
C  routines, is never described, since it always has the same function.  This
C  function is to return an exit code from the routine, where the values of
C  Exit_Code have the following meanings:
```

```
C       Exit_Code < 0   means   The routine encountered an error condition.
C                               Exit_Code contains an error system code (see
C                               routine Error_Message for more info).
C       Exit_Code = 0   means   The routine is exiting normally.
C       Exit_Code > 0   means   The routine is exiting with a special, but non-
C                               error condition.  See the descriptions of the
C                               few routines that use this case for specific
C                               info.

C  Note that when a word or phrase is *surrounded* by asterisks, the intention
C  is to draw attention to that word (as in underlining).

C  ----------------   Logical Unit Useage Summary:   ----------------
C
C  Unit          Location of     Current    Function
C  Name          Definition      Value
C  ----          ----------      -------    --------
C  T_In          Termunits.Inc     5        Reads input from users terminal.
C  T_Out               "           6        Sends messages to users terminal.
C  P_Input_Unit  Plotinterp.Inc     9        Reads from Plot10 input file.
C  D_Printer_Unit  Devices.Inc     10        Writes data to printer.
C  Tu1     Next_Vec_In_Band_Alt    11        Reads/writes temporary vector file.
C  Tu2             "               12                  "
C  D_Raster_Unit Devices.Inc       13        Reads/writes raster data file.

C  ----------------   Program Structure   ----------------

C  This diagram shows the nesting of subroutines within Plaster.  A given
C  routine calls those routines directly to its right and those immediately
C  below this one to the right.  For example, Plaster calls Gen_Bands, which
C  itself calls Make_Band, which calls Next_Vec, which calls Char_Type. Only
C  major routines are included.

C Plaster
C       Get_Options     Lexan           Lx_Chartype
C                       Cparse
C                       Get_Numb
C                       Ident_String    Cparse
C
C
C       Init_Io         Init_Io_Vms
C                       Toprinter_Vms
C
C
C       Gen_Bands       Calc_Band_Params
C                       Zero_Band
C                       Make_Band       Next_Vec_In_Band_Alternating        Plot_Init
C                                       Vector_Gen_Asym
C                                       Next_Vec        Next_Char
C                                                       Char_Type
C                                                       Data_Type
```

```
C                                                   Swchar_Vec     Char_Check
C                                                                  Space1
C                                                                  Space2
C                                                                  Space3
C                                                                  Space4
C                                                                  Stick
C                                                                  Script
C                                                                  Triplx Trip
C                                                                  Gothic Goth
C
C                     Output_Band   Make_Line     Track
C                                   Output_Line
C                                   Track
C                     Save_Band
C
C
C       Read_Bands    Calc_Band_Params
C                     Output_Band
C
C
C       End_Io        Toprinter_Vms
C                     End_Io_Vms
C
C
C       Error_Message Wr_Mess

C  ----------------   Index Of Subroutines   ----------------

C  This is a list of all user subroutines called by Plaster and their
C  functions.

C  ASIGN - Returns: -1 for X<0; 0 for X=0, 1 for X>0, for real X
C  BIC - Clears a bit field in a word
C  BIS - Sets a bit field in a word
C  BY_TOPRINTER - Sends a string of bytes to the printer
C  CALC_BAND_PARAMS - Computes various quantities used in printing
C  CEILING - The ceiling function (smallest integer not smaller than X)
C  CGEN_POP - Maintains a stack used in the character generator
C  CGEN_PUSH - Maintains a stack used in the character generator
C  CHAR_CHECK - Determines if a given character set can draw a given character
C  CHAR_TYPE - Divides the ascii chars into a set of types for Pl0interp
C  CHTOBY - Converts a character string to a byte string
C  CH_TOPRINTER - Sends a string of characters to the printer
C  COMMON_INIT - Initializes the common areas for Plaster
C  CPARSE - Determines if a string is an abreviation for a command from a table
C  DATA_TYPE - Determines if a Plot10 coord is LoX, HiX, LoY, or HiY data.
C  END_IO - Terminates the printer I/O system
C  END_IO_VMS - Terminates the VMS stream I/O system
C  ERROR_MESSAGE - Issues most error messages in Plaster
C  ESL_NUMB - Codes a number into the form used by the ESL
C  EXCHANGE - Exchanges the values in two integer variables
```

```
C  FLUSH_PBUF - Flushes the buffer used in Output_Line to the printer
C  GCD - Computes the greatest common denomonator of two numbers
C  GEN_BANDS - Drives the band creation, printing, and save process.
C  GET_NUMB - Converts a string typed by the user into a parameter value
C  GET_OPTIONS - Drives the user interface - the parameter value assignments
C  GOTH - Returns data for the gothic character set generator
C  GOTHIC - Generates gothic characters
C  IDENT_STRING - Identifies string values for parameters in the user interface
C  INIT_IO - Initializes the printer I/O system
C  INIT_IO_VMS - Initializes the printer stream I/O system
C  IROUNDX - Rounds off a real number
C  LEXAN - Lexical analyzer which breaks a string typed by the user into tokens
C  LX_CHARTYPE - Divides the ascii char set into types for the lexical analyzer
C  MAKE_BAND - Fills a band with graphic elements
C  MAKE_LINE - Creates a print head image from a band
C  MAKE_LINE_VMS - VMS optimized version of MAKE_LINE
C  MAKE_SPACE_STRING - Converts a white distance into a graphics spacing string
C  MERGE - Updates the current position in PlOinterp by coord data from PlotlO
C  MYSIGN - Returns -1 if X<0; 0 if X=0; 1 if X>0, for integer X
C  NEXT_CHAR - Returns the next character from the PlotlO input file
C  NEXT_VEC - Returns the next vector from the PlotlO input file
C  NEXT_VEC_IN_BAND_ALTERNATING - Returns next PlotlO vector in current band
C  NEXT_VEC_IN_BAND_MULTIPASS - Returns next PlotlO vector in current band
C                                          (two different methods used)
C  OUTPUT_BAND - Drives the band printing process
C  OUTPUT_LINE - Converts a print head image into a printer command stream
C  PlO_INIT - Initizlizes
C  PCL - A routine with many entry points that creates many printer commands
C  READ_BANDS - Drives the printing of a picture from a raster data file
C  SAVE_BAND - Saves the current band in a raster data file
C  SCRIPT - Generates the script character set
C  SET_PIX - Sets a pixel in the current band
C  SKIP_SPACES - Skips over spaces in an input line typed by the user
C  SPACE1 - Contains spacing information for the stick character set
C  SPACE2 - Contains spacing information for the script character set
C  SPACE3 - Contains spacing information for the roman triplex character set
C  SPACE4 - Contains spacing information for the gothic character set
C  STICK - Generates the stick character set
C  SWCHAR_VEC - Drives the character generation process
C  TOPRINTER_VMS - Sends a string of bytes to the printer via VMS stream I/O
C  TOSTRING - Converts a binary number into a character string
C  TOUPPER - Converts a character to upper case
C  TRACK - Tracks the print head position the current band during printing
C  TRIP - Returns some data for the triplex character set
C  TRIPLX - Generates the triplex character set
C  VECTOR_GEN_ASYM - Rasterizes a vector using the asymmetrical method
C  VECTOR_GEN_SYM - Rasterizes a vector using the symmetrical method
C  WR_MESS - Actually writes an error message to the terminal
C  ZERO_BAND - Zeros out the band buffer

C  ----------------   Program Overview   ----------------
```

```
C
C  A "verbal run" of the program might go as follows:

C  Initialize all parameters to their default values;
C  Read a command string from the user and modify parameter values;
C  Initialize the selected I/O system;
C  Send commands to the printer to ready it to print graphics;
C  IF we aren't reading a raster data file:
C       Compute the size of the raster we will generate;
C       Divide the raster into a set of horizontal bands;
C       Figure out how to print these bands at the desired resolution;
C       FOR each band:
C               Fill the band with vectors from the Plot10 input file;
C               Extract a print head sweep from the band;
C               Send the sweep in a compressed form to the printer;
C               Write the band to a raster data file if the user so desires;
C       END-FOR;
C  ELSE:
C       Read header info from the raster file;
C       Figure out how to print the raster bands at the desired resolution;
C       FOR each band:
C               Read the band from the file;
C               Extract a print head sweep from the band;
C               Send the sweep in a compressed form to the printer;
C       END-FOR;
C  END-IF;
C  Send the printer back to alphanumeric printing mode;
C  Close the printer I/O system.
C

C  The picture is drawn one band at a time, with the highest band being drawn
C  first. Bands are stored in a byte array with each pixel in the band
C  represented by one bit.  They are stored in two different ways: row major
C  order for compatibility for other row raster representations, and column
C  major order for better processing speed.  The column major order is a more
C  natural mode of storage since we later want to extract bits in vertical (on
C  the page) chunks to be printed in head sweeps.

C  This byte array is indexed backwards in that its coordinate origin is the
C  top of a band.  That is, the first bit (bit 0) in the band represents the
C  pixel at the upper left hand corner of the band.  For this reason the
C  coordinates of a point are flipped about the vertical center of the band
C  when they are used to set a bit within the band.  The reason for this is to
C  have the sequence of low-to-high order bits within a byte correspond to a
C  top-to- bottom traversal of the band.  This enables us to transfer a chunk
C  of bits directly from the byte array to a character to be sent to the
C  printer, and maintain the correct ordering for the pin firing.  Dealing
C  with the bits in chunks of six like this rather than individually decreases
C  the running time of the program by at least a factor of three (probably
C  four or five) in the most computationally bound part of the program
C  (producing a very noticeable difference in printing speed in high
```

```
C  resolution plots).

C  Plaster deals with three different coordinate systems:
C  1)  The Plot10 coordinate system, an integer space with a range defined
C       in the include file Devices.Inc because this range may vary depending
C       on the limits of the user's Plot10 display device.  (Expected range is:
C       0-1023x, 0-1023y)
C  2)  The Sanders coordinate system, which is used to address points within
C       the raster we generate.  The index origin is 0 for X and Y, and the
C       X and Y maxima are determined from the requested image size and
C       resolution.
C  3)  The Band coordinate system is used to address points within the
C       current band.  Since a band is merely a window in the raster, the X
C       range of the Band and Sanders systems are the same, while the Band Y
C       range is much smaller - from 0 to the band height minus 1.  During the
C       rasterization process, vectors are extracted from the Plot10 file,
C       mapped from Plot10 to Sanders space and rasterized, and then rasterized
C       points falling outside the Band coordinate system are trashed and the
C       rest are used to set points within the band.
C

C  ----------------   Notes On Include File Useage   ----------------

C  Include files are used both to define common areas within routines and to
C  define often-used global parameters (as in the Fortran PARAMETER
C  statement).  This makes it easy to change parameter values or common area
C  definitions, as long as one is careful to re-compile all source referencing
C  an altered include file.

C  Variables defined in include files have names of the form:
C  <ident-char>_<words>
C  where <ident-char> is a (hopefully) unique character the same for all
C  variables within one include file.  This scheme should provide some
C  organization for groups of variables with similar functions, and also give
C  a reader of the source code some idea where all the variables he's seeing
C  are coming from.

C  ----------------  Notes On Transporting This Program  ----------------

C  With respect to transportability, this program was designed with several
C  thoughts in mind:
C  a) it should be written in standard Fortran '77 for easy transportability
C  b) it should be easy to overlay for minicomputer applications
C  c) it should be made clear where a user can hook in machine-dependent
C     optimization routines

C  Goal (a) has been largely fulfilled, though some statements in this program
C  are probably still specific to VAX Fortran '77 - particularly Fortran I/O
C  statements (OPEN, CLOSE, Q-format, variable length reads and writes).

C  Goal (b) has been fulfilled - simply note the tree structure shown above.
```

```
C  Admittedly it is an unbalanced tree, but the root need not be terribly long
C  and at each level there are quite a few routines to be overlayed.  It
C  should be possible to save quite a lot of space this way.

C  Goal (c) has also been fulfilled, and several VMS specific routines have
C  been included to do stream I/O and bit addressing operations.  If you are
C  not running VMS simply comment out the references to these routines
C  in modules: By_Toprinter, Ch_Toprinter, Init_Io, and End_Io, and change
C  the call to Make_Line_Vms in Output_Band to a call to Make_Line.  If
C  you are running VMS leave this code intact.

C  If you wish to simulate the above VMS routines on another machine, you
C  must do the following.  Make_Line_Vms should be read and understood and
C  you should try to substitute calls to your own bit-addressing library
C  functions for the calls to Lib$Extzv.  To simulate the other routines you
C  must design a stream I/O system.  By stream I/O I simply mean a way of
C  outputting strings of characters with no Cr-LF or any other junk appended
C  by the operating system or by Fortran.  In VMS this is accomplished through
C  the use of the QIO (Queue I/O) routine, which can queue I/O requests
C  without associated carriage control.  If this can be done in some way on
C  your machine (probably a DEC machine) fine, if not you can't use stream
C  I/O and hence ESL.  To simulate the VMS system it would be a good idea
C  to read through the routines listed in the paragraph above and see what
C  they do.  Note that the stream I/O system must send its output directly
C  to the printer, which could be hung on the terminal or attached to a
C  seperate line.  If the printer is on a seperate line you also must come
C  up with a way of letting a user spool pictures which Plaster has written
C  to files to the printer (on Dec machines one would allocate the appropriate
C  TT: device and then copy the file to it).

C  Note that before compiling this system on another machine, you should be
C  sure to look at the code in routines Init_Io and End_Io relating to the
C  initialization and exit_strings.  You should also check over the machine
C  dependent parameters in Devices.Inc .  You should also adjust the size of
C  the band buffer in Band.Inc .  Make the band as big as possible for most
C  efficient processing.  Also check the user terminal I/O logical unit
C  assignments in Termunits.Inc .

C  When transporting this software to another machine the following routines
C  are likely to require modification of some sort:
C  Next_Char
C  Next_Vec_In_Band_Alternating
C  Pl0_Init
C  By_Toprinter
C  Ch_Toprinter
C  End_Io
C  Gen_Bands
C  Init_Io
C  Read_Bands
C  Wr_Mess
```

```
C  One problem often encountered in transporting Fortran programs is the
C  descrepancies in word sizes among different machines (particularly 16 vs 32
C  bits).  That is, if the programmer's coded INTEGER*4 and this is
C  inefficient and unavailable on your machine, what do you do?  I have
C  declared all integer variables as simply INTEGER, hence when compiled all
C  variables assume the default compiler integer size.  It is possible that a
C  overflows may occur on 16 bit machines, though I believe this to be
C  unlikely.  If they do occur simply change the required variables to
C  INTEGER*4.

C  This entire program is quite dependent upon the ascii character set.  If it
C  is transported to an Ebcdic machine all the code must be gone over with a
C  fine tooth comb to find the numerous changes that must be made.

C  ------------------------------------------------------------------------


C  This is the main program, Plaster.  We are first sure to include all
C  common areas in this module so they are in the root segment if the
C  program is overlaid.

       Implicit Integer (A-Z)

       Include 'Devices.Inc'
       Include 'Symvecmem.Inc'
       Include 'Band.Inc'
       Include 'Options.Inc'
       Include 'Plointerp.Inc'
       Include 'Nvibstate.Inc'

       Character *4 Font_Id, Newdel *1

       Logical *1 Option_Error

C  Let the user modify any parameters he wishes to.  Let him try again on
C  syntax or semantic errors.

20     Exit_Code = 0
       Option_Error = .True.
       Call Get_Options(Exit_Code)
       If (Exit_Code .Lt. 0) Goto 800

       Option_Error = .False.

C  The ! is used as a graphic character, so under RCL we must change the
C  default delimiter to the character below.

       Newdel = '''

C  Assign the printing font based on the horizontal dot spacing selected.
```

```
        If (O_Horizontal_Dot_Spacing .Lt. 16) Then
                Font_Id = '0817'
                Font_X_Spacing = 8
        Else

C   1617 is the low speed 16 hep font.

                Font_Id = '1607'
                Font_X_Spacing = 16
        Endif

C   Compute the sine and cosine of the character rotation angle.

        G_Sin_Crot = Sin(3.14159265/180. * O_Char_Rotation)
        G_Cos_Crot = Cos(3.14159265/180. * O_Char_Rotation)

        Stat1 = Lib$Init_Timer()

C   Initialize the I/O system and ready the printer for graphics if we are
C   indeed to print.

        If (O_Print) Then
                Call Init_Io(Exit_Code)
                If (Exit_Code .Lt. 0) Goto 800

                Call Pcl_Fion
                Call Pcl_Cd(Newdel)
                Call Pcl_Af(31,Font_Id)
                Call Pcl_Sf(31)
        Endif

C   Either read bands from a raster file or generate them from a Plot10 file.

        If (O_Read_Raster) Then
                Call Read_Bands(Font_X_Spacing,Exit_Code)
        Else
                Call Gen_Bands(Font_X_Spacing,Exit_Code)
        Endif
        If (Exit_Code .Lt. 0) Goto 800

C   Ready the printer for non-graphic printing with logical font 0 and close
C   the printer I/O system (if we've been printing).

        If (O_Print) Then
                Call Pcl_Sf(0)
                Call Pcl_Fioff
                Call Pcl_Defdel

                If (O_Print) Call End_Io(Exit_Code)
                If (Exit_Code .Lt. 0) Goto 800
        Endif
```

```
        Stat2 = Lib$Show_Timer(,2,,,)
        Stat2 = Lib$Show_Timer(,1,,,)

        Goto 9999

C  Describe an error condition which has been detected in the bowels of the
C  program and exit.

800     Type *,'Exit_Code Is',Exit_Code
        Call Error_Message(Exit_Code)

        If (Option_Error) Goto 20

9999    Call Exit
        End
```

```
      Real Function Asign(X)

C  Computes my Sign function on a real number, i.e., returns
C                                          0  if X = 0
C                                  x / abs(x)  if X # 0

      If (X .Ne. 0.) Then
            Asign = X / Abs(X)
      Else
            Asign = 0.
      Endif

      Return
      End
```

```
      Integer Function Bic(Word,Mask)

C  BIt Clear.  For each bit set in MASK, the corresponding bit in WORD
C  is cleared.

C  Params:

C  WORD - I [IO]
C  MASK - I [I]

      Implicit Integer (A-Z)

      Bic = Word .And. (.Not. Mask)

      Return
      End
```

```
      Integer Function Bis(Word,Mask)
C  BIt Set.  For each bit set in MASK, the corresponding bit in WORD is set.
C  Params:
C  WORD - I [IO]
C  MASK - I [I]
      Implicit Integer (A-Z)
      Bis = Word .Or. Mask
      Return
      End
```

```
        Subroutine By_Toprinter(Bstring,Slen)

C  This routine sends a byte string to the printer.

C  Parameters:

C  Bstring(...) - b [i] = The string to send to the printer.
C  Slen - i [i] = The number of bytes in Bstring to send.

        Implicit Integer (A-Z)

        Byte Bstring(Slen)

        Include 'Devices.Inc'
        Include 'Options.Inc'

C  Send it through the I/O system we're currently using.

        If (O_Rcl_Mode .And. (.Not. D_Stream_Io .Or.
        1       .Not. O_Use_Default_Io_System)) Then
                Write (D_Printer_Unit,900) (Bstring(N),N=1,Slen)
900             Format (X,<Slen>A1)
        Else
                Call Toprinter_Vms(Bstring,Slen)
        Endif

999     Return
        End
```

```
      Subroutine Calc_Band_Params(Font_X_Spacing,Font_Y_Spacing,X_Ptos,
     1  Y_Ptos,Max_Band_Height,Nbands,Raster_Ymax,X_Passes,Y_Passes,
     2  Xtra_Space,Exit_Code)

C  Error system code is 500

C  This routine computes certain parameters of the printing and rasterization
C  process.

C  Parameters:

C  Font_X_Spacing - i [i] = The X resolution of the font used for printing.
C  Font_Y_Spacing - i [i] = The Y resolution of the font used for printing.
C  X_Ptos, Y_Ptos - r [o] = Factors used to scale coordinates from the Plot10
C       to the Sanders coordinate system.
C  Max_Band_Height - i [o] = The largest allowable height for a band in the
C       raster given the raster width and the band buffer available.
C  Nbands - i [o] = The number of bands of size Max_Band_Height which the
C       raster divides into.
C  Raster_Ymax - i [o] = The maximum Y coord in the raster as a whole (origin
C       is 0).
C  X_Passes, Y_Passes - i [o] = The number of passes the print head must make
C       in the X and Y directions within one interlace group to print an image
C       of the desired resolution with the current font.
C  Xtra_Space - i [i] = The number of heps of extra space that must be inserted
C       between every position along the X axis *in each pass*.

      Implicit Integer (A-Z)

      Real X_Ptos, Y_Ptos

      Include 'Devices.Inc'
      Include 'Band.Inc'
      Include 'Options.Inc'

      Real Desired_X_Density, Desired_Y_Density

C  If we're reading the raster from a file we already know the band height
C  and width, and we don't care about converting from Plot10 to a raster.

      If (.Not. O_Read_Raster) Then

C  Image rotation is accomplished by exchanging the X and Y coords found
C  in the Plot10 file.

            If (O_Image_Rotation .Ne. 0)
     1           Call Exchange(D_Plot10_Xmax,D_Plot10_Ymax)

C  Compute the dimensions of the raster, with an index origin of 0.  Take
C  the page aspect ratio into account, but if it's one the image must be
C  proportioned the same as the Plot10 coordinate system.
```

```
              Desired_X_Density = Float(D_Heps_Per_Inch) /
     1                Float(O_Horizontal_Dot_Spacing)
              Desired_Y_Density = Float(D_Veps_Per_Inch) /
     1                Float(O_Vertical_Dot_Spacing)

              Raster_Xmax = O_Width * Desired_X_Density - 1
              B_Band_Width = Raster_Xmax + 1

              Raster_Ymax = O_Width * O_Page_Aspect_Ratio *
     1          (Float(D_Plot10_Ymax)/Float(D_Plot10_Xmax)) *
     2          Desired_Y_Density - 1

C  Compute scaling factors.

              X_Ptos = Float(Raster_Xmax) / Float(D_Plot10_Xmax)
              Y_Ptos = Float(Raster_Ymax) / Float(D_Plot10_Ymax)
         Endif

C  Compute the number of passes in each direction.  See the discussion in
C  routine Output_Band for information.

         If (O_Horizontal_Dot_Spacing .Le. Font_X_Spacing) Then
              X_Passes = Ceiling(Float(Font_X_Spacing) /
     1                Float(O_Horizontal_Dot_Spacing) )
              Xtra_Space = X_Passes*O_Horizontal_Dot_Spacing - Font_X_Spacing
         Else
              X_Passes = 1
              Xtra_Space = O_Horizontal_Dot_Spacing - Font_X_Spacing
         Endif

         Y_Passes = Font_Y_Spacing / Gcd(Font_Y_Spacing,O_Vertical_Dot_Spacing)

C  If we're not reading a raster compute the maximum band height and the
C  number of bands in the raster.  If we're reading we can check the bands
C  for size limitations as we read them, and we already know the number of
C  bands in the raster.

         If (.Not. O_Read_Raster) Then
              Band_Bits = B_Band_Bytes * D_Bits_Per_Byte
              Max_Band_Height = Min( Raster_Ymax+1, Band_Bits/B_Band_Width )
              Band_Height_Unit = D_Print_Head_Pins * Y_Passes
              Max_Band_Height = Max_Band_Height / Band_Height_Unit *
     1                  Band_Height_Unit

C  If the band height doesn't divide into the raster height evenly there will
C  be one small band at the top.

              Nbands = Raster_Ymax / Max_Band_Height
              If (Nbands*Max_Band_Height .Ne. Raster_Ymax) Nbands = Nbands+1
```

```
C  Make sure the buffer is large enough to accomodate a minimum size band.
                If (Max_Band_Height .Lt. Band_Height_Unit) Then
                        Exit_Code = -500
                        Goto 999
                Endif
        Endif

999     Return
        End
```

```
      Integer Function Ceiling(X)
C  Returns the smallest integer not smaller than X.
      Implicit Integer (A-Z)
      Real X
      If (X .Eq. Float(Ifix(X))) Then
            Ceiling = X
      Else
            Ceiling = X + 1
      Endif
      Return
      End
```

```
      Subroutine Cgen_Pop(Index1,Index2)
C     ***************************************************
C     *                                                 *
C     *                                                 *
C     *      THIS ROUTINE POPS CHARACTER GENERATOR      *
C     *      LOOP VARIABLES OFF OF A STACK              *
C     *                                                 *
C     *                                                 *
C     ***************************************************

      Include 'Chargen.Inc'

      Index2=Istack(Iptr)
      Iptr=Iptr-1
      Index1=Istack(Iptr)
      Iptr=Iptr-1
      Return
      End
```

```
      Subroutine Cgen_Push(Index1,Index2)
C     ***********************************************************
C     *                                                         *
C     *                                                         *
C     *      THIS ROUTINE PUSHES CHARACTER GENERATOR            *
C     *      LOOP VARIABLES ONTO A STACK                        *
C     *                                                         *
C     *                                                         *
C     ***********************************************************

      Include 'Chargen.Inc'

      Iptr=Iptr+1
      Istack(Iptr)=Index1
      Iptr=Iptr+1
      Istack(Iptr)=Index2
      Return
      End
```

```
      Subroutine Char_Check(Nchar,Ifont,Valid)
C     ******************************************
C     *                                        *
C     *    THIS ROUTINE CHECKS TO SEE IF THE   *
C     *    SPECIFIED CHARACTER IN THE SELECTED *
C     *    FONT IS AVAILABLE, OR IF THE DESIRED*
C     *    CHARACTER IS A <SPACE>.             *
C     *                                        *
C     *    IF THE CHARACTER CANNOT BE GENERATED*
C     *    OR THE CHARACTER IS A <SPACE>, THEN *
C     *    VALID = .FALSE.                     *
C     *                                        *
C     ******************************************
      Logical Valid
      Valid=.True.
      If (Ifont.Eq.51) Go To 10
      If (Ifont.Eq.52) Go To 20
C
C TRIPLEX ROMAN AND GOTHIC VALIDITY TESTS
C
      If (Nchar .Lt. 33) Go To 30
      If (Nchar .Eq. 35) Go To 30
      If (Nchar .Eq. 37) Go To 30
      If (Nchar .Eq. 60) Go To 30
      If (Nchar .Eq. 62) Go To 30
      If (Nchar .Eq. 64) Go To 30
      If (Nchar.Ge.93.And.Nchar.Le.96) Go To 30
      If (Nchar.Ge.123.And.Nchar.Le.127) Go To 30
      Go To 40
C
C ENHANCED STICK VALIDITY TESTS
C
10    If (Nchar .Lt. 33) Go To 30
      If (Nchar.Ge.91.And.Nchar.Le.96) Go To 30
      If (Nchar .Eq. 123) Go To 30
      If (Nchar.Ge.125.And.Nchar.Le.127) Go To 30
      Go To 40
C
C SCRIPT VALIDITY TESTS
C
20    If (Nchar .Lt. 65) Go To 30
      If (Nchar.Ge.91.And.Nchar.Le.96) Go To 30
      If (Nchar.Ge.123.And.Nchar.Le.127) Go To 30
      Go To 40
30    Valid = .False.
40    Return
      End
```

```
      Subroutine Char_Type(Type)

C  Compute the type of the current character in the P10INTERP system.
C  This consists of converting an ascii code to a type code.

      Implicit Integer (A-Z)

      Include 'P10interp.Inc'

C  The special ascii codes we recognize:

      Byte Rub_Code, Gs_Code, Us_Code, Bel_Code, Si_Code, So_Code
      Byte Esc_Code, Sub_Code, Enq_Code, Etb_Code, Ff_Code

      Data Rub_Code/127/, Gs_Code/29/, Us_Code/31/, Bel_Code/7/
      Data Si_Code/15/, So_Code/14/, Esc_Code/27/, Sub_Code/26/
      Data Enq_Code/5/, Etb_Code/23/, Ff_Code/12/, Ctlv_Code/22/

C  Decide if we have a printable character or not.

      If (P_Curr_Char .Lt. 32 .Or. P_Curr_Char .Eq. Rub_Code) Then
              Type = P_Nonprint
      Else
              Type = P_Printable
              Goto 999
      Endif

C  Check for the special characters.  They are all non-printable.

      If (P_Curr_Char .Eq. Gs_Code) Type = P_Gs

      If (P_Curr_Char .Eq. Us_Code) Type = P_Crus

      If (P_Curr_Char .Eq. Bel_Code) Type = P_Bel

      If (P_Curr_Char .Eq. Ctlv_Code) Type = P_Ctlv

      If (P_Curr_Char .Eq. Si_Code .Or. P_Curr_Char .Eq. So_Code)
     1        Type = P_Siso

      If (P_Curr_Char .Eq. Esc_Code) Type = P_Esc

      If (P_Curr_Char .Eq. Sub_Code) Type = P_Sub

      If (P_Curr_Char .Eq. Enq_Code .Or.
     1    P_Curr_Char .Eq. Etb_Code .Or.
     2    P_Curr_Char .Eq. Ff_Code      ) Type = P_Enqetbff

999   Return
      End
```

```
      Subroutine Chtoby(Cstring,Slen,Bstring)

C  Converts a character string to a byte string.

C  Parameters:

C  Cstring - c* [i] = The character string.
C  Slen - i [i] = The number of characters in Cstring to convert.
C  Bstring - b [o] = The byte stringe we create.

      Implicit Integer (A-Z)

      Character *(*) Cstring
      Byte Bstring(Slen)

      Do 10 N = 1, Slen
10    Bstring(N) = Ichar(Cstring(N:1))

999   Return
      End
```

```
        Subroutine Ch_Toprinter(Cstring,Slen)

C  This routine sends a character string to the printer.

C  Parameters:

C  Cstring - C* [i] = The variable containing the string to send.
C  Slen - i [i] = The length of the substring we actually want to send.

        Implicit Integer (A-Z)

        Character *(*) Cstring

        Parameter ( Bblen = 50 )
        Byte Byte_Buff(Bblen)

        Include 'Devices.Inc'
        Include 'Options.Inc'

C  If we're using the Fortran I/O system we just write out the string.  If
C  we're using stream I/O we have to convert the character string to a byte
C  string since the stream I/O routine only takes byte strings.

C  We don't pass the error message below up through the error system because
C  there are just too many routines that call this one that would have to be
C  modified.

        If (O_Rcl_Mode .And. (.Not. D_Stream_Io .Or.
        1       .Not. O_Use_Default_Io_System)) Then
                Write (D_Printer_Unit,900) (Cstring(1:Slen))
900             Format (X,A<Slen>)
        Else
                If (Slen .Gt. Bblen) Then
                        Stop 'Ch_Toprinter-I-Character String Too Long To
        1 Convert To Byte Data.'

                Endif

C  Convert to a byte string and output.

                Call Chtoby(Cstring,Slen,Byte_Buff)
                Call Toprinter_Vms(Byte_Buff,Slen)
        Endif

999     Return
        End
        Block Data Common_Init

C  Initializes certain common variables for the entire Plaster system.

        Include 'Pl0interp.Inc'
```

```
      Include 'Devices.Inc'
      Include 'Options.Inc'

C  The current delimiter for the S700 RCL.

      Data D_Delimiter/'!'/

C  PlOinterp system is uninitialized.

      Data P_Done_PlOinit/.False./

C  Limits of the PlotlO coordinate system.

      Data D_PlotlO_Xmax/ 1023 /, D_PlotlO_Ymax/ 767 /

      End
```

```
      Subroutine Cparse(Cstring,Slen,Ctable,Code_Table,Tablen,Exit_Code)

C   This routine determines if a string represents a command from a
C  table of valid commands. If so it returns the index of that command
C  (also stored in the table). If not it returns an error code.

C  Entries in the command table (prepared with routine CTAB) are of the
C  form XXXX*YYY N where XXXXYYY is the full name of the command and N
C  is the associated index. XXXX is the smallest legal abreviation of
C  the command. All, some, or none of the string YYY may be present in
C  the user's string and will not affect the match. If the user attempts
C  to abreviate a command by less than XXXX, an "ambiguous command"
C  error will be returned. If no valid match is found an "illegal
C  command" error will be returned. If, for exampe, the command is
C  "read", it might be defined in CTAB as RE*AD. Thus if the user's
C  command is:
C
C       RE or REA or READ - the command is accepted;
C       R       -         an ambiguous command code is returned
C                         (assuming there is no other command defined
C                         as R_xxx).
C         anything else   - an illegal command code is returned.
C
C  Params:
C
C  Cstring - c* [i] = The command we're to look for in the table.
C  Slen - i [i] = The length of the command in Cstring.
C  Ctable(Tablen) - c* [i] = The table of commands.
C  Code_table(Tablen) - i [i] = The table of command codes, where each entry
C       in this array corresponds to a command in Ctable.
C  Tablen - i [i] = The number of entries in the command table.
C  Exit_code - i [o] = If positive, the code for the table entry which Cstring
C       matched.  Else, -1 if Cstring didn't match any entries in Ctable, and
C       -2 if it ambiguously matched several entries.

      Implicit Integer (A-Z)

      Integer Code_Table(Tablen)
      Character *(*) Cstring, Ctable(Tablen)

      Logical *1 Seen_Ambig, Curr_Ambig

      Character *1 Toupper

      Seen_Ambig = .False.

C  For each entry in the table...

      Do 10 Tabndx = 1, Tablen

C  Assume Cstring matches this entry ambiguously.
```

```
        Curr_Ambig = .True.
        Entry_Ndx = 1

C  For each char in Cstring...

        Do 20 Sndx = 1, Slen

C  See if we've reached the end of the table entry.

        If (Entry_Ndx .Gt. Len(Ctable(1))) Goto 10

25      Continue

C  See if the current characters in the table entry and Cstring match.

        If (Toupper(Cstring(Sndx:Sndx)) .Eq.
        1   Toupper(Ctable(Tabndx)(Entry_Ndx:Entry_Ndx)))  Then
                Entry_Ndx = Entry_Ndx + 1
                Goto 20
        Else

C  If not, we may have gotten to the * in the table entry.  If we did, we'll
C  go on comparing, noting that we can't have an ambiguous match anymore.
C  If it's not a *, the strings simply don't match.

                If (Ctable(Tabndx)(Entry_Ndx:Entry_Ndx) .Eq. '*') Then
                        Curr_Ambig = .False.
                        Entry_Ndx = Entry_Ndx + 1
                        Goto 25
                Else
                        Goto 10
                Endif
        Endif

20      Continue

C  We're at the end of Cstring but not the end of the current table entry,
C  and both strings correspond so far.  If the next char in the table entry
C  is a *, we've got the minimum abreviation for this entry.

        If (Entry_Ndx .Le. Len(Ctable(1))) Then
                If (Ctable(Tabndx)(Entry_Ndx:Entry_Ndx) .Eq. '*')
        1          Curr_Ambig = .False.
        Endif

C  We've either got an ambiguous match or a non-ambiguous match; in the latter
C  case we're done.

        If (Curr_Ambig) Then
                Seen_Ambig = .True.
```

```
        Else
                Exit_Code = Code_Table(Tabndx)
                Goto 999
        Endif

10      Continue

C  We've either got an ambigous string or an illegal one.

        If (Seen_Ambig) Then
                Exit_Code = -2
        Else
                Exit_Code = -1
        Endif

999     Return
        End
```

```
      Subroutine Data_Type(Type)

C  Compute the data type of the current P10interp character - we interpret
C  the character as a Plot-10 encoded coordinate.  See page 1-7 of the
C  4010/4010-1 Maintenance Manual.

C  1 = Hi X / Hi Y
C  2 = Low X
C  3 = Low Y
C  4 = Illegal coordinate (ascii value < 32)

      Implicit Integer (A-Z)

      Include 'P10interp.Inc'

      Type = P_Curr_Char / 32
      If (Type .Eq. 0) Type = 4

      Return
      End
```

```
      Subroutine End_Io(Exit_Code)

C  Terminates the printer I/O system.

      Implicit Integer (A-Z)

      Parameter ( Eslen = 4 )
      Byte Exit_String(Eslen)

      Include 'Devices.Inc'
      Include 'Options.Inc'

C  The Exit_String is a string which is sent to the printer just before we
C  close the stream I/O system (if that is the I/O system in use).  The
C  purpose of this string is to reset the user's terminal if it was changed by
C  Init_Io. I used the exit string because my Concept terminal would intercept
C  escape sequences meant for the printer and try to interpret them as
C  commands.  In Init_Io I changed the Concept's message delimiter from an
C  escape to a ^D; here I must change it back.  Thus I instruct the Concept to
C  change its escape message delimiter back from the ctl-D we set it to in
C  Init_Io to an escape character.  The command is: ^Do <esc> .  You may
C  either comment out the whole exit string business, or substitute an exit
C  string specific to your hardware.

      Data Exit_String/ 4, 111, 32, 27 /

C  Close out the I/O system in use.

      If (O_Rcl_Mode .And. (.Not. D_Stream_Io .Or.
     1       .Not. O_Use_Default_Io_System)) Then
             Close (Unit = D_Printer_Unit)
      Else
             If (D_Stream_Io)  Call Toprinter_Vms(Exit_String,Eslen)
             Call End_Io_Vms(Exit_Code)
      Endif

      Return
      End
```

```
        Subroutine End_Io_Vms(Exit_Code)

C  Error system code is 600

C  Machine dependent stream I/O routine.

C  This routine closes out the VMS stream I/O system.  See Init_Io_Vms for a
C  thorough description of what we're doing.

        Implicit Integer (A-Z)

        Parameter ( Zero_Length = 0 )

        Include 'Vmsio.Inc'

C  Issue a zero length QIO associated with event flag 2.

        If ( V_Success .Ne. Sys$Qio(%Val(V_Event_Flag2),%Val(V_Channel),
        1  %Val(V_Io$Writevblk),,,,Dummy,%Val(Zero_Length),,
        2  %Val(V_No_Carriage_Control),,) ) Then
                Exit_Code = -600
                Goto 999
        Endif

C  Wait for event flag 2 to be set - and hence all previous outstanding I/O to
C  finish.

        Call Sys$Waitfr(%Val(V_Event_Flag2))

C  De-assign the channel from the device.

        If (V_Success .Ne. Sys$Dassgn(%Val(V_Channel)) ) Then
                Exit_Code = -601
                Goto 999
        Endif

999     Return
        End
```

```
      Subroutine Error_Message(Error_Code)

C  This routine is the core of the error reporting system.  The error
C  system functions as follows.  Each subroutine is given a unique *error
C  system code*.  These error system codes are multiples of sequential
C  integers by 100, e.g., 100, 200, 300, etc.  Each different error
C  condition within a subroutine is assigned an *error message code*
C  formed by adding the numbers 0, 1, 2, etc. to the routines error
C  system code.  Thus, if the routine with error system code 800 has
C  three different possible error conditions, their error message codes are
C  800, 801 and 802.  Finally, these numbers are negated so calling
C  routines can test the value of the parameter Exit_Code (the communication
C  channel for error message codes), and abort their execution to print
C  the message if the value is negative.  See the source for Plaster for
C  more information on the values of Exit_Code.

C  This routine takes an error message code (Error_Code) and prints the
C  corresponding error message.  As stated in the User's Guide, error
C  messages have three components, and are of the form:

C  xxxxx-Y-zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz

C  where xxxxx is usually the name of the routine where the error condition
C  was detected, Y is an error severity code, and zzzz is a phrase describing
C  the error.  Y can be one of: F (fatal), W (warning), and I (internal).
C  Fatal messages were probably caused by the user and can be avoided by
C  having the user change his selection of parameters or his input file
C  (or occasionally by having more powerful version of the program created).
C  Internal errors should never occur.  They are the result of either bugs
C  in the program or improper program installation.  Warning messages inform
C  the user that the program is perfectly able to create a picture from its
C  input, but that it may not appear as the user expected for some reason.
C  Note that the routine names in the message should make it relatively
C  easy to tell exactly what conditions are causing these errors.

C  Errors are reported in this manner rather than by simply issuing them
C  in the routine encountering them for several reasons.  First, it
C  provides more organization - messages are not simply spread out over
C  fifty different modules.  Second, it allows all the messages to be
C  overlaid against the main program if this system is installed on a
C  minicomputer.  This should save a good deal of space.  Third, it
C  puts many of the terminal writing statements in the same place in
C  case the I/O mechanism has to be changed.

      Implicit Integer (A-Z)

C  We store a table of error message codes.  The parameter is converted into
C  an index into a table of labels to branch to.

      Parameter ( Nmessages = 39 )
      Integer Labels(Nmessages)
```

```
      Character *160 Message

      Data Labels / 100,101,102,200,201,202,300,301,400,401,402,403,404,
     1       405,406,407,408,409,410,500,600,601,700,800,801,802,900,
     2       1000,1001,1002,1003,1004,1005,1100,1200,1300,1400,1401,1402 /

C  Find the index of the message into the table.

      Do 10 Lndx = 1, Nmessages
10    If (Labels(Lndx) .Eq. - Error_Code) Goto 20

C  An error system error message...

      Message = 'Error_Message-I-Unknown Error Code.'
      Call Wr_Mess(Message)
      Stop

C  Output the appropriate message.

20    Goto (100,101,102,200,201,202,300,301,400,401,402,403,404,405,406,
     1       407,408,409,410,500,600,601,700,800,801,802,900,1000,1001,
     2       1002,1003,1004,1005,1100,1200,1300,1400,1401,1402),
     1 Lndx

C  The messages...

100   Message = 'Getnumb-F-This option type is incorrect for this
     1 parameter.'
      Goto 10000

101   Message = 'Getnumb-F-Value of option is less than lower limit for
     1 parameter.'
      Goto 10000

102   Message = 'Getnumb-F-Value of option is greater than upper limit for
     1 parameter.'
      Goto 10000

200   Message = 'Identstri-F-Parameter requires a string option.'
      Goto 10000

201   Message = 'Identstri-F-This option is not legal for this parameter.'
      Goto 10000

202   Message = 'Identstri-F-Option abreviation is ambiguous.'
      Goto 10000

300   Message = 'Getoption-F-Parameter name is unknown.'
      Goto 10000
```

```
301     Message = 'Getoption-F-Parameter name abreviation is ambiguous.'
        Goto 10000

400     Message = 'Lexan-F-Underscore cannot begin keyword.'
        Goto 10000

401     Message = 'Lexan-F-Illegal character in keyword.'
        Goto 10000

402     Message = 'Lexan-F-Null keyword.'
        Goto 10000

403     Message = 'Lexan-F-Equals sign expected.'
        Goto 10000

404     Message = 'Lexan-F-Illegal character in option.'
        Goto 10000

405     Message = 'Lexan-F-Null option.'
        Goto 10000

406     Message = 'Lexan-F-Illegal character in real.'
        Goto 10000

407     Message = 'Lexan-F-Second decimal point in real.'
        Goto 10000

408     Message = 'Lexan-F-Illegal character in number.'
        Goto 10000

409     Message = 'Lexan-F-Minus sign imbedded in number.'
        Goto 10000

410     Message = 'Lexan-F-Comma/Newline expected.'
        Goto 10000

500     Message = 'Calcbandparams-F-The band buffer is not large enough to
        1 contain the smallest band needed to print a page image of this size
        2 and resolution.'
        Goto 10000

C  Stream i/o error message

600     Message = 'Endiovms-I-Error issuing Qio (stream I/O).'
        Goto 10000

C  Stream i/o error message

601     Message = 'Endiovms-I-Error encountered when attempting to deassign
        1 the stream I/O channel from device TT:'
        Goto 10000
```

```
C  Stream i/o error message.

700     Message = 'Initiovms-I-Error encountered attempting to assign a
     1  a stream I/O channel to device TT:'
        Goto 10000

800     Message = 'Initio-F-Error opening printer logical output file.'
        Goto 10000

801     Message = 'Initio-F-This program has not been generated with stream
     1  I/O capability, so you cannot use ESL.'
        Goto 10000

802     Message = 'Initio-F-This program has not been generated with stream
     1  I/O capability, so you must use the default I/O system.'
        Goto 10000

900     Message = 'Outputban-I-Output buffer will not hold one printing
     1  line.'
        Goto 10000

1000    Message = 'Plointerp-I-Initialization not performed.'
        Goto 10000

1001    Message = 'Plointerp-F-Input file contains raw control chars in
     1  alpha mode.'
        Goto 10000

1002    Message = 'Plointerp-F-Unexpected end of input file.'
        Goto 10000

1003    Message = 'Plointerp-F-Illegal coordinate encountered in input file
     1  while in graph mode.'
        Goto 10000

1004    Message = 'Plointerp-F-Input file has illegal coordinate data
     1  sequence.'
        Goto 10000

1005    Message = 'Plointerp-F-Input file contains illegal escape
     1  sequence.'
        Goto 10000

1100    Message = 'Nextvecaltern-F-Error opening temporary work files.'
        Goto 10000

1200    Message = 'Ploinit-F-Error opening Plot10 input file.'
        Goto 10000

1300    Message = 'Genbands-F-Error creating raster data file.'
```

```
        Goto 10000

1400    Message = 'Readbands-F-Error opening raster data file.'
        Goto 10000

1401    Message = 'Readbands-F-Band stored in raster file is too large to
        1 fit in band buffer.'
        Goto 10000

1402    Message = 'Readbands-F-Fortran I/O error encountered reading raster
        1 data file (file is wrong format ?).'
        Goto 10000

10000   Call Wr_Mess(Message)
        Goto 999

999     Return
        End
```

```
      Subroutine Esl_Numb(Number,Nlen,String)

C  Converts a number to a string in the format used by the printer's
C  Escape Sequence Language - a base 64 system.  See Santec "Printer
C  Command Language" manual for details.

C  Parameters:

C  NUMBER - I [I] = The number to convert.
C  NLEN - I [I] = The number of "digits" to output.  Each printer command
C       requires a fixed number of digits for a given parameter, so we
C       sometimes must generate leading zeros.  1 <= NLEN <= 3
C  STRING - c* [o] = The encoded string we produce.

      Implicit Integer (A-Z)

      Character *(*) String

      Parameter ( At = 64 )

      Character *1 Char

      Numb = Number

C  For each digit...

      Do 10 Digit = Nlen, 1, -1

C  Determine its base 64 value.

      Radix = 64**(Digit-1)

      Value = Numb / Radix

C  This error message isn't routed through the normal error system because
C  it's not worth changing all the routines that call it and all the
C  routines that call them.

      If (Value .Gt. 63) Stop 'Esl_Numb-I-Incompatible Arguments.'

      String(Nlen+1-Digit:Nlen+1-Digit) = Char( At + Value )

      Numb = Mod(Numb,Radix)

10    Continue

999   Return
      End
```

```
      Subroutine Exchange(I1,I2)

C  Exchanges its two integer arguments.

      Implicit Integer (A-Z)

      Temp = I1
      I1 = I2
      I2 = Temp

      Return
      End
```

```
      Subroutine Flush_Pbuf(Pbuf,Pbuf_Dim,Plen)

C  Flushes the buffer used by Output_Line to the printer.

C  Parameters:

C  Pbuf(Pbuf_Dim) - b [i] = The buffer to flush.
C  Plen - i [i] = The actual number of bytes we're to send.

      Implicit Integer (A-Z)

      Byte Pbuf(Pbuf_Dim)

      Include 'Devices.Inc'
      Include 'Options.Inc'

C  If we're in RCL mode and we're not using stream i/o, whatever string we
C  send is going to be terminated by a CR-LF by the operating system or
C  Fortran or whoever.  This will cause the print head to move right one space
C  because we will have fill mode on.  We don't want this, so we append a
C  backspace command onto the buffer to move the print head left one space.
C  This requires that there be some extra space in Pbuf.  With stream i/o
C  there is no CR-LF generated.

      If (O_Rcl_Mode .And. (.Not. D_Stream_Io .Or.
     1      .Not. O_Use_Default_Io_System)) Then

            If (Plen+4 .Gt. Pbuf_Dim) Then
                  Stop 'Flushpbuf-I-Insufficient Room For Bs Command In
     1 Pbuf.'
            Endif

            Pbuf(Plen+1) = Ichar(D_Delimiter)
            Pbuf(Plen+2) = 'B'
            Pbuf(Plen+3) = 'S'
            Pbuf(Plen+4) = Ichar(D_Delimiter)
            Plen = Plen + 4
      Endif

C  Do the I/O.

      Call By_Toprinter(Pbuf,Plen)

      Plen = 0

      Return
      End
```

```
      Integer Function Gcd(A1,B1)
C  Returns the Greatest Common Divisor of two integers, A1 and B1.
      Implicit Integer (A-Z)

      A = A1
      B = B1

10    Temp = A
      A = B
      B = Mod(Temp,B)
      If (B .Ne. 0) Goto 10

      Gcd = A

999   Return
      End
```

```
      Subroutine Gen_Bands(Font_X_Spacing,Exit_Code)

C  Error system code is 1300

C  This routine generates raster bands from a Plot10 input file and sends them
C  to the printer and/or a raster file.

C  Parameters:

C  Font_X_Spacing - i [i] = The X resolution of the font we're printing with.

      Implicit Integer (A-Z)

      Include 'Devices.Inc'
      Include 'Band.Inc'
      Include 'P10interp.Inc'
      Include 'Options.Inc'

      Parameter ( Font_Y_Spacing = 4 )

      Real X_Ptos, Y_Ptos

      Integer Tymax, Tymin

      Logical *1 White_Band

      Data Tymax/-1/, Tymin/32000/

C  Figure out how big the bands and raster are and how to print it.

      Call Calc_Band_Params(Font_X_Spacing,Font_Y_Spacing,X_Ptos,Y_Ptos,
     1  S_Band_Height,Nbands,Raster_Ymax,X_Passes,Y_Passes,Xtra_Space,
     2  Exit_Code)
      If (Exit_Code .Lt. 0) Goto 999

C  Initialize raster file if we're to write one.  We write zero records to be
C  skipped, and a description of the raster.

      If (O_Save_Raster) Then
            Open (Unit = D_Raster_Unit, Type = 'New', Form='Unformatted',
     1            Access='Sequential', Err=800)
            Write (D_Raster_Unit) 0
            Write (D_Raster_Unit) Nbands, B_Band_Width
      Endif

C  Loop through each band in the raster.

      Do 100 Iband = Nbands, 1, -1

C  Compute the Sanders coordinates of the top and bottom of the band in the
C  raster.  Note that if the raster height is not an integer multiple of the
```

```
C  band height we make the top band will be smaller than the others.  We
C  proceed from the top of the raster down.

        S_Band_Top = Min(Raster_Ymax,Iband*S_Band_Height - 1)
        S_Band_Bot = (Iband-1) * S_Band_Height
        B_Band_Height = S_Band_Top - S_Band_Bot + 1

C  Map the Sanders coordinates of the top and bottom of the band to Plot10
C  coordinates.

        T_Band_Top = .5 + (S_Band_Top + 1) / Y_Ptos - 1
        T_Band_Bot = .5 + S_Band_Bot / Y_Ptos

C  If this band is outside the range of graphic coordinates in the Plot10 file
C  (which we may not have computed yet) we know it's all white space.

        White_Band =  ( (Iband .Ne. Nbands) .And.
        1             (T_Band_Bot .Gt. Tymax .Or. T_Band_Top .Lt. Tymin) )
        If (White_Band) Goto 110

C  Clear the band buffer.

        Call Zero_Band

C  Fill the band with graphic elements from the Plot10 input file.

        Call Make_Band(S_Band_Top,S_Band_Bot,X_Ptos,Y_Ptos,Tymin,Tymax,
        1  White_Band,Exit_Code)
        If (Exit_Code .Lt. 0) Goto 999

C  Send the band to the printer if the user so desires.

110     If (O_Print)
        1  Call Output_Band(White_Band,X_Passes,Y_Passes,Xtra_Space,
        2  Font_X_Spacing,Exit_Code)
        If (Exit_Code .Lt. 0) Goto 999

C  Write this band to the raster file if we're supposed to.

        If (O_Save_Raster) Call Save_Band

100     Continue

        If (O_Save_Raster) Close (Unit = D_Raster_Unit)
        Close (Unit = P_Input_Unit)
        Goto 999

C  Error creating raster file.

800     Exit_Code = -1300
        Goto 999
```

```
999     Return
        End
```

```
      Logical *1 Function Get_Numb(Toktyp,Number,Expect_Toktyp,Lower_Limit,
     1       Upper_Limit,Exit_Code)

C  Error system code = -100

C  This routine converts a number in the current token string to a binary
C  number.  It makes sure that the type of the token (the string) is as
C  expected for this parameter (real or integer), and makes sure that the
C  value of the number lies within a specified range. Basically, we accept a
C  numerical value of a parameter.  We return TRUE if we encounter an error.

C  Parameters:

C  Toktyp - i [i] = The type of the token we are to decode.
C  Number - i [o] = The number we create.
C  Expect_Toktyp - i [i] = The expected token type.
C  Lower_Limit, Upper_Limit - r [i] = The minimum and maximum allowable values
C        for the number.

      Implicit Integer (A-Z)

      Real Number, Lower_Limit, Upper_Limit

      Include 'Lexan.Inc'

      Logical *1 Minus

C  Make sure we have a legal token type.  We must have a number type.  If an
C  integer is expected we must have an integer token.  If a real is expected
C  we can have either a real or integer token.

      If (Toktyp .Ne. Expect_Toktyp .And.
     1  .Not. (Expect_Toktyp .Eq. L_Tt_Real .And. Toktyp .Eq. L_Tt_Int))
     2  Goto 800

C  Initialize...

      Minus = .False.
      Number = 0.
      Dot_Ndx = L_Tlen

C  Loop through the token (which is in common).  Look for minus signs, decimal
C  points, plus signs, and digits, whose value we accumulate.  This may not
C  be 100% precise, but I don't think users will be concerned with entering
C  parameters with 6 digits of accuracy.

      Do 10 Ndx = 1, L_Tlen

      If (L_Token(Ndx:Ndx) .Eq. '-') Then
            Minus = .True.
            Goto 10
```

```
        Endif

        If (L_Token(Ndx:Ndx) .Eq. '+') Goto 10

        If (L_Token(Ndx:Ndx) .Eq. '.') Then
              Dot_Ndx = Ndx
              Goto 10
        Endif

        Number = Number * 10. + Ichar(L_Token(Ndx:Ndx)) - Ichar('0')

10      Continue

C  Adjust the exponent of the number of a decmial point.

        Number = Number / 10.**(L_Tlen-Dot_Ndx)
        If (Minus) Number = - Number

C  Make sure the value of the number is within the required range.

        If (Number .Lt. Lower_Limit) Goto 801
        If (Number .Gt. Upper_Limit) Goto 802

        Get_Numb = .False.
        Goto 999

C  Type of option incorrect for parameter

800     Get_Numb = .True.
        Exit_Code = -100
        Goto 999

C  Number < Lower_Limit

801     Get_Numb = .True.
        Exit_Code = -101
        Goto 999

C  Number > Upper_Limit

802     Get_Numb = .True.
        Exit_Code = -102
        Goto 999

999     Return
        End
```

```
      Subroutine Get_Options(Exit_Code)

C  Error system code = -300

C  This routine controls the user interface, where we accept command lines
C  from the user in which he changes the values of Plaster's parameters.

      Implicit Integer (A-Z)

      Parameter ( Ctable_Length = 18 )
      Character *23 Ctable(Ctable_Length)

      Parameter ( Ncset_Names = 4, Npass_Names = 2 )
      Character *7 Cset_Names(Ncset_Names), Pass_Names(Npass_Names)*12
      Character *4 Clang_Names(2), Yes_No(2)*4

      Integer Code_Table(Ctable_Length), Index_Table(5)

      Logical *1 Get_Numb, Ident_String

      Real Number

      Parameter ( Cm_To_Inches = 2.54 )

      Include 'Termunits.Inc'
      Include'Lexan.Inc'
      Include 'Options.Inc'

C  The table of parameter (or command) names.

      Data Ctable/
     1       'Wi*Dth',
     1       'Pag*E_Aspect_Ratio',
     1       'Ho*Rizontal_Dot_Spacing',
     1       'Ve*Rtical_Dot_Spacing',
     1       'Char_Sc*Ale',
     1       'Char_A*Spect_Ratio',
     1       'Char_Se*T',
     1       'Char_R*Otation',
     1       'Im*Age_Rotation',
     1       'Li*Ne_Thickness',
     1       'In*Ches',
     1       'Pr*Int',
     1       'Sa*Ve_Raster',
     1       'Re*Ad_Raster',
     1       'Ro*W_Major',
     1       'Pas*S_Algorithm',
     1       'Co*Mmand_Language',
     1       'Us*E_Default_Io_System'       /

C  Tables of string parameter values.
```

```
      Data Cset_Names/ 'St*Ick', 'Sc*Ript', 'R*Oman', 'G*Othic' /
      Data Pass_Names/ 'A*Lternating', 'M*Ultipass' /
      Data Clang_Names/ 'R*Cl', 'E*Sl' /
      Data Yes_No/ 'Y*Es', 'N*O' /

C  Tables of codes for parameter names and string values.

      Data Code_Table/ 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18 /
      Data Index_Table / 1,2,3,4,5 /

C  Default values for the parameters.

      O_Width = 6.
      O_Page_Aspect_Ratio = 1.
      O_Horizontal_Dot_Spacing = 8
      O_Vertical_Dot_Spacing = 4
      O_Char_Set = 1
      O_Char_Scale = 1.
      O_Char_Aspect_Ratio = 1.
      O_Char_Rotation = 0.
      O_Image_Rotation = 0
      O_Inch_Mode = .True.
      O_Print = .True.
      O_Save_Raster = .False.
      O_Read_Raster = .False.
      O_Row_Major = .False.
      O_Pass_Algorithm = 2
      O_Rcl_Mode = .True.
      O_Use_Default_Io_System = .True.

      L_Iptr = 0

C  Come here to receive the next assignment of the form: <keyword>=<value>

10    Continue

C  Get the keyword.  If we're at an end of line we exit.

      Call Lexan(L_Ex_Keywd,Toktyp,Exit_Code)
      If (Exit_Code .Lt. 0) Goto 899
      If (Toktyp .Eq. L_Tt_Eol) Then
            Exit_Code = 0
            Goto 999
      Endif

C  Parse the keyword to convert it to a parameter name code.

      Call Cparse(L_Token,L_Tlen,Ctable,Code_Table,Ctable_Length,Keyword)
      If (Keyword .Lt. 0) Goto 800
```

```
C  Pass the equals sign in the string.

        Call Lexan(L_Ex_Equal,Toktyp,Exit_Code)
        If (Exit_Code .Lt. 0) Goto 899

C  Get the value the users assigned to the keyword.

        Call Lexan(L_Ex_Option,Toktyp,Exit_Code)
        If (Exit_Code .Lt. 0) Goto 899

C  Now perform the semantics of the users assignment: change the value of
C  a parameter.

        Goto (1000,2000,3000,4000,5000,6000,7000,8000,9000,10000,11000,12000,
        1  13000,14000,15000,16000,17000,18000), Keyword

C  Pass over a comma or end of line after the assignment so we can go on
C  to the next.

100     Call Lexan(L_Ex_Comma,Toktyp,Exit_Code)
        If (Exit_Code .Lt. 0) Goto 899
        Goto 10

C  _________________   Assignment processing   _________________

C  The way we do these assignments is by obtaining a final value for the
C  parameter - a binary number of a string identification code, checking
C  that this value is within the legal range, and then changing the value
C  of an O_ variable to effect a parameter value change.

C  Width

1000    If (Get_Numb(Toktyp,Number,L_Tt_Real,1.,12.,Exit_Code))  Goto 899
        O_Width = Number
        If (.Not. O_Inch_Mode) O_Width = O_Width / Cm_To_Inches
        Goto 100

C  Page aspect ratio

2000    If (Get_Numb(Toktyp,Number,L_Tt_Real,.2,5.,Exit_Code))  Goto 899
        O_Page_Aspect_Ratio = Number
        Goto 100

C  Horizontal dot spacing

3000    If (Get_Numb(Toktyp,Number,L_Tt_Int,1.,100.,Exit_Code))  Goto 899
        O_Horizontal_Dot_Spacing = Number
        Goto 100

C  Vertical dot spacing
```

```
4000    If (Get_Numb(Toktyp,Number,L_Tt_Int,1.,50.,Exit_Code))  Goto 899
        O_Vertical_Dot_Spacing = Number
        Goto 100

C  Character scale

5000    If (Get_Numb(Toktyp,Number,L_Tt_Real,.1,10.,Exit_Code))  Goto 899
        O_Char_Scale = Number
        Goto 100

C  Character aspect ratio

6000    If (Get_Numb(Toktyp,Number,L_Tt_Real,.2,5.,Exit_Code))  Goto 899
        O_Char_Aspect_Ratio = Number
        Goto 100

C  Character set

7000    If (Ident_String(Toktyp,Sindex,Cset_Names,Ncset_Names,Index_Table,
        1       Exit_Code))  Goto 899
        O_Char_Set = Sindex
        Goto 100

C  Character rotation

8000    If (Get_Numb(Toktyp,Number,L_Tt_Real,-90.,90.,Exit_Code))  Goto 899
        O_Char_Rotation = Number
        Goto 100

C  Image rotation

9000    If (Get_Numb(Toktyp,Number,L_Tt_Int,-1.,1.,Exit_Code))  Goto 899
        O_Image_Rotation = Number
        Goto 100

C  Line thickness; a line thickening algorithm has not been implimented yet.

10000   If (Get_Numb(Toktyp,Number,L_Tt_Int,0.,5.,Exit_Code))  Goto 899
        O_Line_Thickness = Numb
        Goto 100

C  Inch mode

11000   If (Ident_String(Toktyp,Sindex,Yes_No,2,Index_Table,Exit_Code))
        1       Goto 899
        O_Inch_Mode = Sindex .Eq. 1
        Goto 100

C  Print mode

12000   If (Ident_String(Toktyp,Sindex,Yes_No,2,Index_Table,Exit_Code))
```

```
     1       Goto 899
     O_Print = Sindex .Eq. 1
     Goto 100

C  Raster save mode

13000  If (Ident_String(Toktyp,Sindex,Yes_No,2,Index_Table,Exit_Code))
     1       Goto 899
     O_Save_Raster = Sindex .Eq. 1
     Goto 100

C  Raster read mode

14000  If (Ident_String(Toktyp,Sindex,Yes_No,2,Index_Table,Exit_Code))
     1       Goto 899
     O_Read_Raster = Sindex .Eq. 1
     Goto 100

C  Row major mode

15000  If (Ident_String(Toktyp,Sindex,Yes_No,2,Index_Table,Exit_Code))
     1       Goto 899
     O_Row_Major = Sindex .Eq. 1
     Goto 100

C  Pass algorithm specification

16000  If (Ident_String(Toktyp,Sindex,Pass_Names,Npass_Names,Index_Table,
     1       Exit_Code))  Goto 899
     O_Pass_Algorithm = Sindex
     Goto 100

C  RCL mode

17000  If (Ident_String(Toktyp,Sindex,Clang_Names,2,Index_Table,
     1       Exit_Code))  Goto 899
     O_Rcl_Mode = Sindex .Eq. 1
     Goto 100

C  Use default I/O system mode

18000  If (Ident_String(Toktyp,Sindex,Yes_No,2,Index_Table,Exit_Code))
     1       Goto 899
     O_Use_Default_Io_System = Sindex .Eq. 1
     Goto 100

C  Illegal / Ambiguous parameter name

800    If (Keyword .Eq. -1) Then
            Exit_Code = -300
     Else
```

```
                Exit_Code = -301
        Endif
        Goto 899

C  For syntactic errors in the users command string we put a little carret
C  under the cause of the error.

899     Write (T_Out,900)
900     Format (X,<L_Iptr-2>X,'^')

999     Return
        End
```

```
      Subroutine Goth(Ipair,Nx,Ny)
      Dimension Ie(812)
C
C     ********************************************************
C     *      BASIC ELEMENTS TABLE FOR GOTHIC CHAR SET        *
C     ********************************************************
C
      Data Ie( 1   )/ 9  /,Ie( 2   )/ 7  /,Ie( 3   )/ 1  /
      Data Ie( 4   )/ 11 /,Ie( 5   )/ 2  /,Ie( 6   )/ 1  /
      Data Ie( 7   )/-2  /,Ie( 8   )/ 1  /,Ie( 9   )/-1  /
      Data Ie( 10  )/ 1  /,Ie( 11  )/-1  /,Ie( 12  )/-1  /
      Data Ie( 13  )/-2  /,Ie( 14  )/-1  /,Ie( 15  )/ 2  /
      Data Ie( 16  )/-1  /,Ie( 17  )/ 1  /,Ie( 18  )/-11 /
      Data Ie( 19  )/ 0  /,Ie( 20  )/ 11 /,Ie( 21  )/ 1  /
      Data Ie( 22  )/ 1  /,Ie( 23  )/ 1  /,Ie( 24  )/-1  /
      Data Ie( 25  )/ 0  /,Ie( 26  )/-15 /,Ie( 27  )/-2  /
      Data Ie( 28  )/-2  /,Ie( 29  )/-2  /,Ie( 30  )/ 2  /
      Data Ie( 31  )/ 0  /,Ie( 32  )/-1  /,Ie( 33  )/ 2  /
      Data Ie( 34  )/ 0  /,Ie( 35  )/ 4  /,Ie( 36  )/ 14 /
      Data Ie( 37  )/ 2  /,Ie( 38  )/ 6  /,Ie( 39  )/ 0  /
      Data Ie( 40  )/-6  /,Ie( 41  )/ 1  /,Ie( 42  )/ 6  /
      Data Ie( 43  )/ 8  /,Ie( 44  )/-6  /,Ie( 45  )/ 7  /
      Data Ie( 46  )/ 25 /,Ie( 47  )/ 0  /,Ie( 48  )/-29 /
      Data Ie( 49  )/ 4  /,Ie( 50  )/ 29 /,Ie( 51  )/ 0  /
      Data Ie( 52  )/ 25 /,Ie( 53  )/ 1  /,Ie( 54  )/-2  /
      Data Ie( 55  )/ 0  /,Ie( 56  )/-2  /,Ie( 57  )/-1  /
      Data Ie( 58  )/ 2  /,Ie( 59  )/-3  /,Ie( 60  )/ 1  /
      Data Ie( 61  )/-4  /,Ie( 62  )/ 0  /,Ie( 63  )/-3  /
      Data Ie( 64  )/-1  /,Ie( 65  )/ 0  /,Ie( 66  )/-3  /
      Data Ie( 67  )/ 3  /,Ie( 68  )/-2  /,Ie( 69  )/ 6  /
      Data Ie( 70  )/-2  /,Ie( 71  )/-1  /,Ie( 72  )/-2  /
      Data Ie( 73  )/ 1  /,Ie( 74  )/ 16 /,Ie( 75  )/-11 /
      Data Ie( 76  )/-4  /,Ie( 77  )/-11 /,Ie( 78  )/-5  /
      Data Ie( 79  )/ 10 /,Ie( 80  )/ 7  /,Ie( 81  )/ 6  /
      Data Ie( 82  )/ 6  /,Ie( 83  )/ 1  /,Ie( 84  )/ 0  /
      Data Ie( 85  )/-1  /,Ie( 86  )/ 0  /,Ie( 87  )/ 0  /
      Data Ie( 88  )/ 1  /,Ie( 89  )/-9  /,Ie( 90  )/-5  /
      Data Ie( 91  )/-6  /,Ie( 92  )/-6  /,Ie( 93  )/-4  /
      Data Ie( 94  )/ 5  /,Ie( 95  )/ 4  /,Ie( 96  )/ 4  /
      Data Ie( 97  )/-5  /,Ie( 98  )/-6  /,Ie( 99  )/ 6  /
      Data Ie( 100 )/-6  /,Ie( 101 )/ 4  /,Ie( 102 )/-6  /
      Data Ie( 103 )/ 2  /,Ie( 104 )/-2  /,Ie( 105 )/ 1  /
      Data Ie( 106 )/ 2  /,Ie( 107 )/-3  /,Ie( 108 )/ 0  /
      Data Ie( 109 )/-4  /,Ie( 110 )/ 6  /,Ie( 111 )/-6  /
      Data Ie( 112 )/ 6  /,Ie( 113 )/ 1  /,Ie( 114 )/-15 /
      Data Ie( 115 )/-3  /,Ie( 116 )/ 4  /,Ie( 117 )/ 9  /
      Data Ie( 118 )/ 11 /,Ie( 119 )/-4  /,Ie( 120 )/-5  /
      Data Ie( 121 )/ 5  /,Ie( 122 )/-5  /,Ie( 123 )/ 3  /
      Data Ie( 124 )/ 0  /,Ie( 125 )/-14 /,Ie( 126 )/ 0  /
      Data Ie( 127 )/ 9  /,Ie( 128 )/ 9  /,Ie( 129 )/ 9  /
      Data Ie( 130 )/ 14 /,Ie( 131 )/ 12 /,Ie( 132 )/ 25 /
```

```
Data Ie( 133 )/-2  /,Ie( 134 )/-3  /,Ie( 135 )/-2  /
Data Ie( 136 )/-4  /,Ie( 137 )/-1  /,Ie( 138 )/-5  /
Data Ie( 139 )/ 0  /,Ie( 140 )/-4  /,Ie( 141 )/ 1  /
Data Ie( 142 )/-5  /,Ie( 143 )/ 2  /,Ie( 144 )/-4  /
Data Ie( 145 )/ 2  /,Ie( 146 )/-3  /,Ie( 147 )/-4  /
Data Ie( 148 )/ 26 /,Ie( 149 )/-1  /,Ie( 150 )/-3  /
Data Ie( 151 )/-1  /,Ie( 152 )/-4  /,Ie( 153 )/ 1  /
Data Ie( 154 )/-4  /,Ie( 155 )/ 1  /,Ie( 156 )/-3  /
Data Ie( 157 )/ 2  /,Ie( 158 )/ 24 /,Ie( 159 )/-1  /
Data Ie( 160 )/-6  /,Ie( 161 )/ 1  /,Ie( 162 )/-6  /
Data Ie( 163 )/ 6  /,Ie( 164 )/ 25 /,Ie( 165 )/ 4  /
Data Ie( 166 )/ 26 /,Ie( 167 )/-2  /,Ie( 168 )/ 24 /
Data Ie( 169 )/ 0  /,Ie( 170 )/ 12 /,Ie( 171 )/ 2  /
Data Ie( 172 )/-10 /,Ie( 173 )/ 2  /,Ie( 174 )/ 10 /
Data Ie( 175 )/ 5  /,Ie( 176 )/-9  /,Ie( 177 )/-10 /
Data Ie( 178 )/ 6  /,Ie( 179 )/-10 /,Ie( 180 )/ 4  /
Data Ie( 181 )/ 10 /,Ie( 182 )/ 6  /,Ie( 183 )/-8  /
Data Ie( 184 )/-6  /,Ie( 185 )/ 10 /,Ie( 186 )/ 4  /
Data Ie( 187 )/ 10 /,Ie( 188 )/ 1  /,Ie( 189 )/ 0  /
Data Ie( 190 )/ 17 /,Ie( 191 )/ 0  /,Ie( 192 )/-17 /
Data Ie( 193 )/ 8  /,Ie( 194 )/ 8  /,Ie( 195 )/-17 /
Data Ie( 196 )/ 0  /,Ie( 197 )/ 17 /,Ie( 198 )/ 0  /
Data Ie( 199 )/ 9  /,Ie( 200 )/-3  /,Ie( 201 )/ 0  /
Data Ie( 202 )/ 2  /,Ie( 203 )/ 2  /,Ie( 204 )/ 2  /
Data Ie( 205 )/ 18 /,Ie( 206 )/ 9  /,Ie( 207 )/ 9  /
Data Ie( 208 )/ 3  /,Ie( 209 )/ 1  /,Ie( 210 )/-7  /
Data Ie( 211 )/ 18 /,Ie( 212 )/ 32 /,Ie( 213 )/-18 /
Data Ie( 214 )/-32 /,Ie( 215 )/ 15 /,Ie( 216 )/ 2  /
Data Ie( 217 )/-2  /,Ie( 218 )/ 0  /,Ie( 219 )/-5  /
Data Ie( 220 )/-1  /,Ie( 221 )/ 0  /,Ie( 222 )/ 16 /
Data Ie( 223 )/ 5  /,Ie( 224 )/ 1  /,Ie( 225 )/ 0  /
Data Ie( 226 )/-16 /,Ie( 227 )/-11 /,Ie( 228 )/ 16 /
Data Ie( 229 )/-2  /,Ie( 230 )/ 18 /,Ie( 231 )/ 2  /
Data Ie( 232 )/ 19 /,Ie( 233 )/ 13 /,Ie( 234 )/ 2  /
Data Ie( 235 )/-3  /,Ie( 236 )/-2  /,Ie( 237 )/ 0  /
Data Ie( 238 )/-13 /,Ie( 239 )/-5  /,Ie( 240 )/ 16 /
Data Ie( 241 )/ 10 /,Ie( 242 )/ 20 /,Ie( 243 )/ 3  /
Data Ie( 244 )/-1  /,Ie( 245 )/ 0  /,Ie( 246 )/-8  /
Data Ie( 247 )/-7  /,Ie( 248 )/ 0  /,Ie( 249 )/ 0  /
Data Ie( 250 )/-5  /,Ie( 251 )/ 4  /,Ie( 252 )/ 2  /
Data Ie( 253 )/ 4  /,Ie( 254 )/ 1  /,Ie( 255 )/ 4  /
Data Ie( 256 )/-1  /,Ie( 257 )/-5  /,Ie( 258 )/ 0  /
Data Ie( 259 )/ 5  /,Ie( 260 )/ 19 /,Ie( 261 )/ 1  /
Data Ie( 262 )/ 7  /,Ie( 263 )/-9  /,Ie( 264 )/-11 /
Data Ie( 265 )/ 3  /,Ie( 266 )/ 1  /,Ie( 267 )/ 5  /
Data Ie( 268 )/ 0  /,Ie( 269 )/ 9  /,Ie( 270 )/ 20 /
Data Ie( 271 )/ 0  /,Ie( 272 )/-7  /,Ie( 273 )/-2  /
Data Ie( 274 )/ 9  /,Ie( 275 )/ 5  /,Ie( 276 )/-1  /
Data Ie( 277 )/-6  /,Ie( 278 )/-2  /,Ie( 279 )/-4  /
Data Ie( 280 )/-1  /,Ie( 281 )/ 5  /,Ie( 282 )/ 7  /
Data Ie( 283 )/-1  /,Ie( 284 )/ 6  /,Ie( 285 )/ 11 /
```

```
Data Ie( 286 )/ 6  /,Ie( 287 )/-9  /,Ie( 288 )/ 0  /
Data Ie( 289 )/ 0  /,Ie( 290 )/ 5  /,Ie( 291 )/ 10 /
Data Ie( 292 )/ 10 /,Ie( 293 )/ 0  /,Ie( 294 )/-14 /
Data Ie( 295 )/ 2  /,Ie( 296 )/-14 /,Ie( 297 )/ 4  /
Data Ie( 298 )/ 0  /,Ie( 299 )/-14 /,Ie( 300 )/ 5  /
Data Ie( 301 )/ 8  /,Ie( 302 )/ 11 /,Ie( 303 )/ 0  /
Data Ie( 304 )/-18 /,Ie( 305 )/ 3  /,Ie( 306 )/ 19 /
Data Ie( 307 )/ 9  /,Ie( 308 )/ 0  /,Ie( 309 )/-12 /
Data Ie( 310 )/ 0  /,Ie( 311 )/ 0  /,Ie( 312 )/-9  /
Data Ie( 313 )/ 0  /,Ie( 314 )/-10 /,Ie( 315 )/-5  /
Data Ie( 316 )/ 19 /,Ie( 317 )/-3  /,Ie( 318 )/-7  /
Data Ie( 319 )/-1  /,Ie( 320 )/ 12 /,Ie( 321 )/-8  /
Data Ie( 322 )/-1  /,Ie( 323 )/ 0  /,Ie( 324 )/ 9  /
Data Ie( 325 )/-1  /,Ie( 326 )/ 7  /,Ie( 327 )/-2  /
Data Ie( 328 )/-7  /,Ie( 329 )/-1  /,Ie( 330 )/ 11 /
Data Ie( 331 )/ 17 /,Ie( 332 )/ 21 /,Ie( 333 )/-4  /
Data Ie( 334 )/-2  /,Ie( 335 )/-4  /,Ie( 336 )/ 1  /
Data Ie( 337 )/-4  /,Ie( 338 )/-4  /,Ie( 339 )/-1  /
Data Ie( 340 )/ 3  /,Ie( 341 )/ 0  /,Ie( 342 )/ 3  /
Data Ie( 343 )/ 1  /,Ie( 344 )/ 3  /,Ie( 345 )/ 2  /
Data Ie( 346 )/ 3  /,Ie( 347 )/-9  /,Ie( 348 )/ 6  /
Data Ie( 349 )/-5  /,Ie( 350 )/-10 /,Ie( 351 )/ 0  /
Data Ie( 352 )/ 7  /,Ie( 353 )/-8  /,Ie( 354 )/ 3  /
Data Ie( 355 )/ 0  /,Ie( 356 )/ 6  /,Ie( 357 )/-8  /
Data Ie( 358 )/-3  /,Ie( 359 )/ 3  /,Ie( 360 )/ 15 /
Data Ie( 361 )/ 1  /,Ie( 362 )/ 5  /,Ie( 363 )/-1  /
Data Ie( 364 )/ 8  /,Ie( 365 )/-10 /,Ie( 366 )/-5  /
Data Ie( 367 )/-2  /,Ie( 368 )/ 8  /,Ie( 369 )/ 5  /
Data Ie( 370 )/ 8  /,Ie( 371 )/ 13 /,Ie( 372 )/ 10 /
Data Ie( 373 )/-5  /,Ie( 374 )/ 17 /,Ie( 375 )/ 2  /
Data Ie( 376 )/ 11 /,Ie( 377 )/ 18 /,Ie( 378 )/ 13 /
Data Ie( 379 )/ 8  /,Ie( 380 )/ 21 /,Ie( 381 )/ 0  /
Data Ie( 382 )/ 4  /,Ie( 383 )/-6  /,Ie( 384 )/-4  /
Data Ie( 385 )/ 8  /,Ie( 386 )/ 0  /,Ie( 387 )/-4  /
Data Ie( 388 )/-3  /,Ie( 389 )/ 3  /,Ie( 390 )/ 18 /
Data Ie( 391 )/ 7  /,Ie( 392 )/-16 /,Ie( 393 )/-7  /
Data Ie( 394 )/ 16 /,Ie( 395 )/-13 /,Ie( 396 )/ 11 /
Data Ie( 397 )/ 0  /,Ie( 398 )/-11 /,Ie( 399 )/-6  /
Data Ie( 400 )/ 0  /,Ie( 401 )/ 3  /,Ie( 402 )/ 16 /
Data Ie( 403 )/-6  /,Ie( 404 )/-14 /,Ie( 405 )/ 2  /
Data Ie( 406 )/ 5  /,Ie( 407 )/-6  /,Ie( 408 )/-1  /
Data Ie( 409 )/-3  /,Ie( 410 )/-16 /,Ie( 411 )/ 1  /
Data Ie( 412 )/ 10 /,Ie( 413 )/-3  /,Ie( 414 )/ 5  /
Data Ie( 415 )/ 4  /,Ie( 416 )/-5  /,Ie( 417 )/ 6  /
Data Ie( 418 )/ 3  /,Ie( 419 )/ 3  /,Ie( 420 )/ 17 /
Data Ie( 421 )/-4  /,Ie( 422 )/ 4  /,Ie( 423 )/ 3  /
Data Ie( 424 )/ 2  /,Ie( 425 )/ 2  /,Ie( 426 )/ 9  /
Data Ie( 427 )/-3  /,Ie( 428 )/ 9  /,Ie( 429 )/-13 /
Data Ie( 430 )/-2  /,Ie( 431 )/ 18 /,Ie( 432 )/ 6  /
Data Ie( 433 )/-3  /,Ie( 434 )/ 2  /,Ie( 435 )/ 3  /
Data Ie( 436 )/-3  /,Ie( 437 )/-17 /,Ie( 438 )/ 10 /
```

```
Data Ie( 439 )/ 3  /,Ie( 440 )/ 3  /,Ie( 441 )/ 3  /
Data Ie( 442 )/ 13 /,Ie( 443 )/ 15 /,Ie( 444 )/ 3  /
Data Ie( 445 )/ 0  /,Ie( 446 )/ 15 /,Ie( 447 )/-11 /
Data Ie( 448 )/ 0  /,Ie( 449 )/ 14 /,Ie( 450 )/ 0  /
Data Ie( 451 )/-11 /,Ie( 452 )/ 18 /,Ie( 453 )/ 12 /
Data Ie( 454 )/ 0  /,Ie( 455 )/-12 /,Ie( 456 )/ 1  /
Data Ie( 457 )/ 1  /,Ie( 458 )/ 9  /,Ie( 459 )/-3  /
Data Ie( 460 )/-10 /,Ie( 461 )/-3  /,Ie( 462 )/ 18 /
Data Ie( 463 )/-2  /,Ie( 464 )/-16 /,Ie( 465 )/ 4  /
Data Ie( 466 )/-8  /,Ie( 467 )/-9  /,Ie( 468 )/-13 /
Data Ie( 469 )/ 7  /,Ie( 470 )/ 0  /,Ie( 471 )/-14 /
Data Ie( 472 )/-1  /,Ie( 473 )/ 10 /,Ie( 474 )/ 21 /
Data Ie( 475 )/-2  /,Ie( 476 )/-15 /,Ie( 477 )/ 9  /
Data Ie( 478 )/-1  /,Ie( 479 )/ 4  /,Ie( 480 )/-2  /
Data Ie( 481 )/ 0  /,Ie( 482 )/ 8  /,Ie( 483 )/-8  /
Data Ie( 484 )/-10 /,Ie( 485 )/ 11 /,Ie( 486 )/ 16 /
Data Ie( 487 )/ 4  /,Ie( 488 )/ 7  /,Ie( 489 )/-15 /
Data Ie( 490 )/-3  /,Ie( 491 )/ 15 /,Ie( 492 )/-2  /
Data Ie( 493 )/-7  /,Ie( 494 )/ 8  /,Ie( 495 )/ 0  /
Data Ie( 496 )/ 13 /,Ie( 497 )/ 3  /,Ie( 498 )/ 14 /
Data Ie( 499 )/-3  /,Ie( 500 )/-11 /,Ie( 501 )/ 9  /
Data Ie( 502 )/-9  /,Ie( 503 )/ 0  /,Ie( 504 )/ 10 /
Data Ie( 505 )/-11 /,Ie( 506 )/-12 /,Ie( 507 )/ 13 /
Data Ie( 508 )/ 21 /,Ie( 509 )/-7  /,Ie( 510 )/-1  /
Data Ie( 511 )/ 0  /,Ie( 512 )/-12 /,Ie( 513 )/ 14 /
Data Ie( 514 )/ 16 /,Ie( 515 )/-5  /,Ie( 516 )/-3  /
Data Ie( 517 )/ 3  /,Ie( 518 )/-10 /,Ie( 519 )/ 12 /
Data Ie( 520 )/ 12 /,Ie( 521 )/-13 /,Ie( 522 )/ 1  /
Data Ie( 523 )/-11 /,Ie( 524 )/-13 /,Ie( 525 )/ 2  /
Data Ie( 526 )/-6  /,Ie( 527 )/ 7  /,Ie( 528 )/-5  /
Data Ie( 529 )/-1  /,Ie( 530 )/ 4  /,Ie( 531 )/ 9  /
Data Ie( 532 )/ 21 /,Ie( 533 )/ 14 /,Ie( 534 )/-9  /
Data Ie( 535 )/-9  /,Ie( 536 )/ 5  /,Ie( 537 )/-3  /
Data Ie( 538 )/ 3  /,Ie( 539 )/-7  /,Ie( 540 )/-17 /
Data Ie( 541 )/-1  /,Ie( 542 )/ 16 /,Ie( 543 )/ 4  /
Data Ie( 544 )/-4  /,Ie( 545 )/-6  /,Ie( 546 )/ 18 /
Data Ie( 547 )/-3  /,Ie( 548 )/ 8  /,Ie( 549 )/-5  /
Data Ie( 550 )/ 8  /,Ie( 551 )/-5  /,Ie( 552 )/ 10 /
Data Ie( 553 )/ 5  /,Ie( 554 )/-11 /,Ie( 555 )/-12 /
Data Ie( 556 )/ 18 /,Ie( 557 )/ 6  /,Ie( 558 )/-12 /
Data Ie( 559 )/-16 /,Ie( 560 )/ 19 /,Ie( 561 )/ 0  /
Data Ie( 562 )/ 18 /,Ie( 563 )/ 4  /,Ie( 564 )/ 3  /
Data Ie( 565 )/-16 /,Ie( 566 )/-7  /,Ie( 567 )/-4  /
Data Ie( 568 )/ 19 /,Ie( 569 )/ 7  /,Ie( 570 )/ 12 /
Data Ie( 571 )/-14 /,Ie( 572 )/-7  /,Ie( 573 )/ 16 /
Data Ie( 574 )/ 0  /,Ie( 575 )/-2  /,Ie( 576 )/ 4  /
Data Ie( 577 )/-2  /,Ie( 578 )/-18 /,Ie( 579 )/-2  /
Data Ie( 580 )/ 3  /,Ie( 581 )/ 0  /,Ie( 582 )/-19 /
Data Ie( 583 )/ 6  /,Ie( 584 )/-5  /,Ie( 585 )/ 0  /
Data Ie( 586 )/-22 /,Ie( 587 )/-3  /,Ie( 588 )/ 16 /
Data Ie( 589 )/ 4  /,Ie( 590 )/ 19 /,Ie( 591 )/-10 /
```

```
Data Ie( 592 )/-8  /,Ie( 593 )/-3  /,Ie( 594 )/ 6  /
Data Ie( 595 )/-9  /,Ie( 596 )/-1  /,Ie( 597 )/ 5  /
Data Ie( 598 )/ 11 /,Ie( 599 )/-3  /,Ie( 600 )/ 7  /
Data Ie( 601 )/ 3  /,Ie( 602 )/-5  /,Ie( 603 )/-2  /
Data Ie( 604 )/ 6  /,Ie( 605 )/ 12 /,Ie( 606 )/ 18 /
Data Ie( 607 )/ 7  /,Ie( 608 )/-2  /,Ie( 609 )/ 5  /
Data Ie( 610 )/-14 /,Ie( 611 )/-8  /,Ie( 612 )/ 0  /
Data Ie( 613 )/-8  /,Ie( 614 )/ 1  /,Ie( 615 )/-15 /
Data Ie( 616 )/ 11 /,Ie( 617 )/ 10 /,Ie( 618 )/ 0  /
Data Ie( 619 )/-3  /,Ie( 620 )/-4  /,Ie( 621 )/-4  /
Data Ie( 622 )/-6  /,Ie( 623 )/-11 /,Ie( 624 )/ 4  /
Data Ie( 625 )/-10 /,Ie( 626 )/-7  /,Ie( 627 )/-3  /
Data Ie( 628 )/ 12 /,Ie( 629 )/-18 /,Ie( 630 )/ 2  /
Data Ie( 631 )/-6  /,Ie( 632 )/ 1  /,Ie( 633 )/-11 /
Data Ie( 634 )/-1  /,Ie( 635 )/ 6  /,Ie( 636 )/ 11 /
Data Ie( 637 )/ 1  /,Ie( 638 )/ 4  /,Ie( 639 )/ 0  /
Data Ie( 640 )/ 14 /,Ie( 641 )/-8  /,Ie( 642 )/-9  /
Data Ie( 643 )/-3  /,Ie( 644 )/ 15 /,Ie( 645 )/-6  /
Data Ie( 646 )/ 17 /,Ie( 647 )/ 12 /,Ie( 648 )/ 2  /
Data Ie( 649 )/-6  /,Ie( 650 )/ 16 /,Ie( 651 )/ 1  /
Data Ie( 652 )/-9  /,Ie( 653 )/ 6  /,Ie( 654 )/ 17 /
Data Ie( 655 )/-2  /,Ie( 656 )/ 15 /,Ie( 657 )/-6  /
Data Ie( 658 )/ 11 /,Ie( 659 )/ 16 /,Ie( 660 )/ 2  /
Data Ie( 661 )/-15 /,Ie( 662 )/ 17 /,Ie( 663 )/-12 /
Data Ie( 664 )/ 16 /,Ie( 665 )/ 8  /,Ie( 666 )/ 16 /
Data Ie( 667 )/-8  /,Ie( 668 )/ 10 /,Ie( 669 )/ 5  /
Data Ie( 670 )/ 4  /,Ie( 671 )/-1  /,Ie( 672 )/ 18 /
Data Ie( 673 )/ 8  /,Ie( 674 )/-18 /,Ie( 675 )/-13 /
Data Ie( 676 )/ 19 /,Ie( 677 )/ 8  /,Ie( 678 )/-17 /
Data Ie( 679 )/-9  /,Ie( 680 )/ 9  /,Ie( 681 )/ 4  /
Data Ie( 682 )/ 9  /,Ie( 683 )/-12 /,Ie( 684 )/-17 /
Data Ie( 685 )/-7  /,Ie( 686 )/-7  /,Ie( 687 )/-1  /
Data Ie( 688 )/ 10 /,Ie( 689 )/ 0  /,Ie( 690 )/-21 /
Data Ie( 691 )/ 6  /,Ie( 692 )/ 0  /,Ie( 693 )/ 0  /
Data Ie( 694 )/ 23 /,Ie( 695 )/-2  /,Ie( 696 )/ 17 /
Data Ie( 697 )/-2  /,Ie( 698 )/-6  /,Ie( 699 )/-1  /
Data Ie( 700 )/ 19 /,Ie( 701 )/ 3  /,Ie( 702 )/ 4  /
Data Ie( 703 )/ 2  /,Ie( 704 )/ 4  /,Ie( 705 )/ 5  /
Data Ie( 706 )/ 6  /,Ie( 707 )/ 10 /,Ie( 708 )/ 15 /
Data Ie( 709 )/-8  /,Ie( 710 )/-11 /,Ie( 711 )/-10 /
Data Ie( 712 )/-19 /,Ie( 713 )/-12 /,Ie( 714 )/ 10 /
Data Ie( 715 )/ 11 /,Ie( 716 )/ 21 /,Ie( 717 )/ 9  /
Data Ie( 718 )/ 15 /,Ie( 719 )/ 7  /,Ie( 720 )/ 9  /
Data Ie( 721 )/-8  /,Ie( 722 )/ 2  /,Ie( 723 )/-4  /
Data Ie( 724 )/ 9  /,Ie( 725 )/ 9  /,Ie( 726 )/ 1  /
Data Ie( 727 )/ 3  /,Ie( 728 )/ 12 /,Ie( 729 )/ 2  /
Data Ie( 730 )/ 12 /,Ie( 731 )/ 8  /,Ie( 732 )/-8  /
Data Ie( 733 )/-7  /,Ie( 734 )/ 10 /,Ie( 735 )/-1  /
Data Ie( 736 )/ 17 /,Ie( 737 )/ 7  /,Ie( 738 )/ 6  /
Data Ie( 739 )/ 5  /,Ie( 740 )/ 3  /,Ie( 741 )/ 3  /
Data Ie( 742 )/-4  /,Ie( 743 )/-6  /,Ie( 744 )/ 3  /
```

```
Data Ie( 745 )/ 7  /,Ie( 746 )/ 19 /,Ie( 747 )/ 1  /
Data Ie( 748 )/ 19 /,Ie( 749 )/ 12 /,Ie( 750 )/ 3  /
Data Ie( 751 )/-3  /,Ie( 752 )/-5  /,Ie( 753 )/ 2  /
Data Ie( 754 )/ 18 /,Ie( 755 )/ 3  /,Ie( 756 )/-18 /
Data Ie( 757 )/ 2  /,Ie( 758 )/-22 /,Ie( 759 )/ 2  /
Data Ie( 760 )/-5  /,Ie( 761 )/-1  /,Ie( 762 )/ 5  /
Data Ie( 763 )/-2  /,Ie( 764 )/ 13 /,Ie( 765 )/ 14 /
Data Ie( 766 )/ 2  /,Ie( 767 )/-2  /,Ie( 768 )/ 11 /
Data Ie( 769 )/ 1  /,Ie( 770 )/ 12 /,Ie( 771 )/-4  /
Data Ie( 772 )/ 10 /,Ie( 773 )/-3  /,Ie( 774 )/ 13 /
Data Ie( 775 )/ 5  /,Ie( 776 )/ 5  /,Ie( 777 )/-5  /
Data Ie( 778 )/ 5  /,Ie( 779 )/ 8  /,Ie( 780 )/-2  /
Data Ie( 781 )/-6  /,Ie( 782 )/ 13 /,Ie( 783 )/ 9  /
Data Ie( 784 )/ 2  /,Ie( 785 )/-9  /,Ie( 786 )/ 10 /
Data Ie( 787 )/-5  /,Ie( 788 )/ 11 /,Ie( 789 )/ 8  /
Data Ie( 790 )/ 3  /,Ie( 791 )/-4  /,Ie( 792 )/ 8  /
Data Ie( 793 )/-11 /,Ie( 794 )/ 11 /,Ie( 795 )/ 5  /
Data Ie( 796 )/-10 /,Ie( 797 )/-4  /,Ie( 798 )/ 7  /
Data Ie( 799 )/ 3  /,Ie( 800 )/ 6  /,Ie( 801 )/-3  /
Data Ie( 802 )/-6  /,Ie( 803 )/ 0  /,Ie( 804 )/ 19 /
Data Ie( 805 )/ 15 /,Ie( 806 )/ 14 /,Ie( 807 )/-12 /
Data Ie( 808 )/-14 /,Ie( 809 )/ 1  /,Ie( 810 )/ 13 /
Data Ie( 811 )/-2  /,Ie( 812 )/-10 /
Nx=Ie(Ipair*2-3)
Ny=Ie(Ipair*2+1-3)
Return
End
```

```
      Subroutine Gothic(Char,Xpos,Ypos,Exit_Code)
C
C  This routine returns the next relative move/draw to generate the current
C  character in a gothic font.

C  Parameters:

C  CHAR - b [i] = The character to generate.
C  XPOS,YPOS - R [o] = A relative move/draw coordinate.
C  EXIT_CODE - i [o] = 2 for a move
C                      3 for a draw
C                     -1 for no more graphic elements in character.

C  This routine was originally part of the Hewlett-Packard Plot21 system.
C  Its logic, but not its data areas, have been substantially re-written.

C  For documentation of the program logic, see the body of the virtually
C  identical routine TRIPLX.

      Implicit Integer (A-Z)

      Integer Ic(2526),In(230)

      Byte Char

      Real Xm, Ym , Xpos, Ypos

      Include 'Chargen.Inc'
      Include 'Symvecmem.Inc'
      Include 'Options.Inc'

C     ********************************************************
C     *                                                      *
C     *                                                      *
C     *  THIS MODULE GENERATES GOTHIC CHARACTERS             *
C     *                                                      *
C     *                                                      *
C     ********************************************************
C
C
C
C     ********************************************************
C     *          CHARACTER TABLE                             *
C     ********************************************************
C------<< 33 ! >>
      Data Ic( 1    )/ 2   /,Ic( 2    )/ 3   /
      Data Ic( 3    )/ 4   /,Ic( 4    )/ 518 /
      Data Ic( 5    )/ 408 /,Ic( 6    )/ 9   /
      Data Ic( 7    )/ 10  /,Ic( 8    )/ 11  /
      Data Ic( 9    )/ 6   /,Ic( 10   )/ 12  /
      Data Ic( 11   )/ 13  /,Ic( 12   )/ 7   /
```

```
      Data Ic( 13   )/ 1   /,Ic( 14   )/ 14  /
      Data Ic( 15   )/ 410 /
C-----<< 34 " >>
      Data Ic( 16   )/ 19  /,Ic( 17   )/ 411 /
      Data Ic( 18   )/ 1   /,Ic( 19   )/ 23  /
      Data Ic( 20   )/ 411 /
C-----<< 36 $ >>
      Data Ic( 21   )/ 24  /,Ic( 22   )/ 25  /
      Data Ic( 23   )/ 1   /,Ic( 24   )/ 26  /
      Data Ic( 25   )/ 25  /,Ic( 26   )/ 1   /
      Data Ic( 27   )/ 27  /,Ic( 28   )/ 412 /
      Data Ic( 29   )/ 29  /,Ic( 30   )/ 4   /
      Data Ic( 31   )/ 413 /,Ic( 32   )/ 31  /
      Data Ic( 33   )/ 32  /,Ic( 34   )/ 33  /
      Data Ic( 35   )/ 15  /,Ic( 36   )/ 34  /
      Data Ic( 37   )/ 28  /,Ic( 38   )/ 35  /
      Data Ic( 39   )/ 36  /,Ic( 40   )/ 412 /
      Data Ic( 41   )/ 414 /,Ic( 42   )/ 1   /
      Data Ic( 43   )/ 38  /,Ic( 44   )/ 30  /
      Data Ic( 45   )/ 1   /,Ic( 46   )/ 39  /
      Data Ic( 47   )/ 415 /,Ic( 48   )/ 36  /
      Data Ic( 49   )/ 412 /,Ic( 50   )/ 1   /
      Data Ic( 51   )/ 40  /,Ic( 52   )/ 30  /
      Data Ic( 53   )/ 1   /,Ic( 54   )/ 38  /
      Data Ic( 55   )/ 416 /,Ic( 56   )/ 415 /
      Data Ic( 57   )/ 36  /,Ic( 58   )/ 35  /
      Data Ic( 59   )/ 28  /,Ic( 60   )/ 414 /
      Data Ic( 61   )/ 7   /,Ic( 62   )/ 33  /
      Data Ic( 63   )/ 32  /,Ic( 64   )/ 31  /
      Data Ic( 65   )/ 6   /,Ic( 66   )/ 30  /
      Data Ic( 67   )/ 4   /,Ic( 68   )/ 29  /
      Data Ic( 69   )/ 415 /
C-----<< 38 & >>
      Data Ic( 70   )/ 41  /,Ic( 71   )/ 42  /
      Data Ic( 72   )/ 417 /,Ic( 73   )/ 12  /
      Data Ic( 74   )/ 418 /,Ic( 75   )/ 44  /
      Data Ic( 76   )/ 419 /,Ic( 77   )/ 45  /
      Data Ic( 78   )/ 13  /,Ic( 79   )/ 420 /
      Data Ic( 80   )/ 46  /,Ic( 81   )/ 47  /
      Data Ic( 82   )/ 48  /,Ic( 83   )/ 42  /
      Data Ic( 84   )/ 1   /,Ic( 85   )/ 12  /
      Data Ic( 86   )/ 49  /,Ic( 87   )/ 48  /
      Data Ic( 88   )/ 50  /,Ic( 89   )/ 51  /
      Data Ic( 90   )/ 52  /,Ic( 91   )/ 53  /
      Data Ic( 92   )/ 9   /,Ic( 93   )/ 421 /
      Data Ic( 94   )/ 54  /,Ic( 95   )/ 7   /
      Data Ic( 96   )/ 55  /,Ic( 97   )/ 5   /
      Data Ic( 98   )/ 6   /,Ic( 99   )/ 56  /
      Data Ic( 100  )/ 57  /,Ic( 101  )/ 1   /
      Data Ic( 102  )/ 58  /,Ic( 103  )/ 59  /
      Data Ic( 104  )/ 1   /,Ic( 105  )/ 60  /
```

```
      Data Ic( 106  )/ 59  /,Ic( 107  )/ 1   /
      Data Ic( 108  )/ 61  /,Ic( 109  )/ 62  /
      Data Ic( 110  )/ 52  /,Ic( 111  )/ 53  /
      Data Ic( 112  )/ 9   /,Ic( 113  )/ 422 /
      Data Ic( 114  )/ 64  /,Ic( 115  )/ 48  /
      Data Ic( 116  )/ 1   /,Ic( 117  )/ 65  /
      Data Ic( 118  )/ 48  /
C------<< 39 ' >>
      Data Ic( 119  )/ 66  /,Ic( 120  )/ 411 /
C------<< 40 ( >>
      Data Ic( 121  )/ 67  /,Ic( 122  )/ 15  /
      Data Ic( 123  )/ 68  /,Ic( 124  )/ 69  /
      Data Ic( 125  )/ 70  /,Ic( 126  )/ 71  /
      Data Ic( 127  )/ 72  /,Ic( 128  )/ 73  /
      Data Ic( 129  )/ 74  /,Ic( 130  )/ 53  /
      Data Ic( 131  )/ 1   /,Ic( 132  )/ 75  /
      Data Ic( 133  )/ 76  /,Ic( 134  )/ 77  /
      Data Ic( 135  )/ 21  /,Ic( 136  )/ 78  /
      Data Ic( 137  )/ 423 /,Ic( 138  )/ 80  /
      Data Ic( 139  )/ 424 /,Ic( 140  )/ 81  /
      Data Ic( 141  )/ 21  /,Ic( 142  )/ 82  /
      Data Ic( 143  )/ 79  /,Ic( 144  )/ 28  /
C------<< 41 ) >>
      Data Ic( 145  )/ 83  /,Ic( 146  )/ 53  /
      Data Ic( 147  )/ 74  /,Ic( 148  )/ 73  /
      Data Ic( 149  )/ 72  /,Ic( 150  )/ 71  /
      Data Ic( 151  )/ 70  /,Ic( 152  )/ 69  /
      Data Ic( 153  )/ 68  /,Ic( 154  )/ 15  /
      Data Ic( 155  )/ 1   /,Ic( 156  )/ 84  /
      Data Ic( 157  )/ 79  /,Ic( 158  )/ 78  /
      Data Ic( 159  )/ 21  /,Ic( 160  )/ 77  /
      Data Ic( 161  )/ 76  /,Ic( 162  )/ 1   /
      Data Ic( 163  )/ 85  /,Ic( 164  )/ 28  /
      Data Ic( 165  )/ 79  /,Ic( 166  )/ 82  /
      Data Ic( 167  )/ 21  /,Ic( 168  )/ 81  /
      Data Ic( 169  )/ 76  /,Ic( 170  )/ 37  /
C------<< 42 * >>
      Data Ic( 171  )/ 65  /,Ic( 172  )/ 86  /
      Data Ic( 173  )/ 7   /,Ic( 174  )/ 87  /
      Data Ic( 175  )/ 7   /,Ic( 176  )/ 6   /
      Data Ic( 177  )/ 88  /,Ic( 178  )/ 419 /
      Data Ic( 179  )/ 89  /,Ic( 180  )/ 90  /
      Data Ic( 181  )/ 43  /,Ic( 182  )/ 23  /
      Data Ic( 183  )/ 43  /,Ic( 184  )/ 45  /
      Data Ic( 185  )/ 91  /,Ic( 186  )/ 45  /
      Data Ic( 187  )/ 1   /,Ic( 188  )/ 21  /
      Data Ic( 189  )/ 92  /,Ic( 190  )/ 44  /
      Data Ic( 191  )/ 93  /,Ic( 192  )/ 44  /
      Data Ic( 193  )/ 45  /,Ic( 194  )/ 94  /
      Data Ic( 195  )/ 45  /
C------<< 43 + >>
```

```
      Data Ic( 196  )/ 95  /,Ic( 197  )/ 44  /
      Data Ic( 198  )/ 96  /,Ic( 199  )/ 43  /
      Data Ic( 200  )/ 97  /,Ic( 201  )/ 1   /
      Data Ic( 202  )/ 98  /,Ic( 203  )/ 425 /
C------<< 44 , >>
      Data Ic( 204  )/ 101 /,Ic( 205  )/ 426 /
C------<< 45 - >>
      Data Ic( 206  )/ 104 /,Ic( 207  )/ 425 /
C------<< 46 . >>
      Data Ic( 208  )/ 105 /,Ic( 209  )/ 410 /
C------<< 47/ >>
      Data Ic( 210  )/ 106 /,Ic( 211  )/ 44  /
      Data Ic( 212  )/ 107 /,Ic( 213  )/ 43  /
      Data Ic( 214  )/ 108 /
C------<< 48 0 >>
      Data Ic( 215  )/ 109 /,Ic( 216  )/ 110 /
      Data Ic( 217  )/ 111 /,Ic( 218  )/ 427 /
      Data Ic( 219  )/ 112 /,Ic( 220  )/ 428 /
      Data Ic( 221  )/ 535 /,Ic( 222  )/ 114 /
      Data Ic( 223  )/ 1   /,Ic( 224  )/ 115 /
      Data Ic( 225  )/ 429 /,Ic( 226  )/ 430 /
C------<< 49 1 >>
      Data Ic( 227  )/ 118 /,Ic( 228  )/ 431 /
      Data Ic( 229  )/ 112 /,Ic( 230  )/ 30  /
      Data Ic( 231  )/ 119 /,Ic( 232  )/ 432 /
      Data Ic( 233  )/ 120 /,Ic( 234  )/ 433 /
      Data Ic( 235  )/ 434 /,Ic( 236  )/ 6   /
      Data Ic( 237  )/ 12  /,Ic( 238  )/ 28  /
      Data Ic( 239  )/ 114 /,Ic( 240  )/ 9   /
C------<< 50 2 >>
      Data Ic( 241  )/ 122 /,Ic( 242  )/ 408 /
      Data Ic( 243  )/ 435 /,Ic( 244  )/ 436 /
      Data Ic( 245  )/ 12  /,Ic( 246  )/ 9   /
      Data Ic( 247  )/ 437 /,Ic( 248  )/ 124 /
      Data Ic( 249  )/ 125 /,Ic( 250  )/ 33  /
      Data Ic( 251  )/ 15  /,Ic( 252  )/ 76  /
      Data Ic( 253  )/ 126 /,Ic( 254  )/ 127 /
      Data Ic( 255  )/ 128 /,Ic( 256  )/ 63  /
      Data Ic( 257  )/ 129 /,Ic( 258  )/ 15  /
      Data Ic( 259  )/ 31  /,Ic( 260  )/ 130 /
      Data Ic( 261  )/ 111 /,Ic( 262  )/ 1   /
      Data Ic( 263  )/ 131 /,Ic( 264  )/ 4   /
      Data Ic( 265  )/ 1   /,Ic( 266  )/ 129 /
      Data Ic( 267  )/ 438 /,Ic( 268  )/ 132 /
      Data Ic( 269  )/ 439 /,Ic( 270  )/ 133 /
      Data Ic( 271  )/ 134 /,Ic( 272  )/ 135 /
      Data Ic( 273  )/ 123 /
C------<< 51 3 >>
      Data Ic( 274  )/ 136 /,Ic( 275  )/ 15  /
      Data Ic( 276  )/ 435 /,Ic( 277  )/ 43  /
      Data Ic( 278  )/ 4   /,Ic( 279  )/ 12  /
```

```
      Data Ic( 280  )/ 9   /,Ic( 281  )/ 129 /
      Data Ic( 282  )/ 18  /,Ic( 283  )/ 137 /
      Data Ic( 284  )/ 110 /,Ic( 285  )/ 33  /
      Data Ic( 286  )/ 519 /,Ic( 287  )/ 138 /
      Data Ic( 288  )/ 4   /,Ic( 289  )/ 1   /
      Data Ic( 290  )/ 139 /,Ic( 291  )/ 440 /
      Data Ic( 292  )/ 22  /,Ic( 293  )/ 126 /
      Data Ic( 294  )/ 1   /,Ic( 295  )/ 140 /
      Data Ic( 296  )/ 9   /,Ic( 297  )/ 437 /
      Data Ic( 298  )/ 137 /,Ic( 299  )/ 441 /
      Data Ic( 300  )/ 142 /,Ic( 301  )/ 126 /
      Data Ic( 302  )/ 1   /,Ic( 303  )/ 143 /
      Data Ic( 304  )/ 440 /,Ic( 305  )/ 125 /
      Data Ic( 306  )/ 9   /
C------<< 52 4 >>
      Data Ic( 307  )/ 144 /,Ic( 308  )/ 145 /
      Data Ic( 309  )/ 146 /,Ic( 310  )/ 147 /
      Data Ic( 311  )/ 53  /,Ic( 312  )/ 37  /
      Data Ic( 313  )/ 148 /,Ic( 314  )/ 443 /
      Data Ic( 315  )/ 528 /,Ic( 316  )/ 96  /
      Data Ic( 317  )/ 1   /,Ic( 318  )/ 149 /
      Data Ic( 319  )/ 150 /,Ic( 320  )/ 13  /
      Data Ic( 321  )/ 102 /,Ic( 322  )/ 444 /
      Data Ic( 323  )/ 151 /,Ic( 324  )/ 445 /
      Data Ic( 325  )/ 22  /,Ic( 326  )/ 440 /
      Data Ic( 327  )/ 152 /,Ic( 328  )/ 54  /
      Data Ic( 329  )/ 6   /,Ic( 330  )/ 446 /
C------<< 53 5 >>
      Data Ic( 331  )/ 154 /,Ic( 332  )/ 155 /
      Data Ic( 333  )/ 447 /,Ic( 334  )/ 156 /
      Data Ic( 335  )/ 157 /,Ic( 336  )/ 63  /
      Data Ic( 337  )/ 128 /,Ic( 338  )/ 447 /
      Data Ic( 339  )/ 542 /,Ic( 340  )/ 158 /
      Data Ic( 341  )/ 441 /,Ic( 342  )/ 159 /
      Data Ic( 343  )/ 155 /,Ic( 344  )/ 1   /
      Data Ic( 345  )/ 160 /,Ic( 346  )/ 526 /
      Data Ic( 347  )/ 158 /,Ic( 348  )/ 1   /
      Data Ic( 349  )/ 161 /,Ic( 350  )/ 9   /
      Data Ic( 351  )/ 158 /,Ic( 352  )/ 1   /
      Data Ic( 353  )/ 162 /,Ic( 354  )/ 9   /
C------<< 54 6 >>
      Data Ic( 355  )/ 136 /,Ic( 356  )/ 53  /
      Data Ic( 357  )/ 436 /,Ic( 358  )/ 431 /
      Data Ic( 359  )/ 6   /,Ic( 360  )/ 448 /
      Data Ic( 361  )/ 110 /,Ic( 362  )/ 114 /
      Data Ic( 363  )/ 449 /,Ic( 364  )/ 113 /
      Data Ic( 365  )/ 18  /,Ic( 366  )/ 450 /
      Data Ic( 367  )/ 141 /,Ic( 368  )/ 451 /
      Data Ic( 369  )/ 164 /,Ic( 370  )/ 429 /
      Data Ic( 371  )/ 452 /,Ic( 372  )/ 165 /
      Data Ic( 373  )/ 453 /,Ic( 374  )/ 1   /
```

```
      Data Ic( 375  )/ 166 /,Ic( 376  )/ 53  /
      Data Ic( 377  )/ 124 /
C-----<< 55 7 >>
      Data Ic( 378  )/ 167 /,Ic( 379  )/ 168 /
      Data Ic( 380  )/ 141 /,Ic( 381  )/ 55  /
      Data Ic( 382  )/ 169 /,Ic( 383  )/ 103 /
      Data Ic( 384  )/ 123 /,Ic( 385  )/ 135 /
      Data Ic( 386  )/ 113 /,Ic( 387  )/ 37  /
      Data Ic( 388  )/ 68  /,Ic( 389  )/ 170 /
      Data Ic( 390  )/ 68  /,Ic( 391  )/ 76  /
      Data Ic( 392  )/ 34  /,Ic( 393  )/ 79  /
      Data Ic( 394  )/ 103 /,Ic( 395  )/ 171 /
      Data Ic( 396  )/ 172 /,Ic( 397  )/ 173 /
      Data Ic( 398  )/ 174 /,Ic( 399  )/ 1   /
      Data Ic( 400  )/ 175 /,Ic( 401  )/ 123 /
      Data Ic( 402  )/ 135 /,Ic( 403  )/ 454 /
      Data Ic( 404  )/ 176 /,Ic( 405  )/ 76  /
      Data Ic( 406  )/ 34  /,Ic( 407  )/ 79  /
C-----<< 56 8 >>
      Data Ic( 408  )/ 54  /,Ic( 409  )/ 527 /
      Data Ic( 410  )/ 13  /,Ic( 411  )/ 4   /
      Data Ic( 412  )/ 113 /,Ic( 413  )/ 18  /
      Data Ic( 414  )/ 177 /,Ic( 415  )/ 110 /
      Data Ic( 416  )/ 178 /,Ic( 417  )/ 110 /
      Data Ic( 418  )/ 179 /,Ic( 419  )/ 428 /
      Data Ic( 420  )/ 12  /,Ic( 421  )/ 542 /
      Data Ic( 422  )/ 21  /,Ic( 423  )/ 110 /
      Data Ic( 424  )/ 180 /,Ic( 425  )/ 110 /
      Data Ic( 426  )/ 21  /,Ic( 427  )/ 519 /
      Data Ic( 428  )/ 181 /,Ic( 429  )/ 445 /
      Data Ic( 430  )/ 182 /,Ic( 431  )/ 439 /
      Data Ic( 432  )/ 142 /,Ic( 433  )/ 526 /
      Data Ic( 434  )/ 439 /,Ic( 435  )/ 183 /
      Data Ic( 436  )/ 9   /,Ic( 437  )/ 439 /
      Data Ic( 438  )/ 184 /,Ic( 439  )/ 126 /
      Data Ic( 440  )/ 523 /,Ic( 441  )/ 185 /
      Data Ic( 442  )/ 21  /,Ic( 443  )/ 455 /
      Data Ic( 444  )/ 1   /,Ic( 445  )/ 186 /
      Data Ic( 446  )/ 440 /,Ic( 447  )/ 22  /
      Data Ic( 448  )/ 126 /
C-----<< 57 9 >>
      Data Ic( 449  )/ 187 /,Ic( 450  )/ 44  /
      Data Ic( 451  )/ 141 /,Ic( 452  )/ 8   /
      Data Ic( 453  )/ 528 /,Ic( 454  )/ 163 /
      Data Ic( 455  )/ 428 /,Ic( 456  )/ 535 /
      Data Ic( 457  )/ 114 /,Ic( 458  )/ 441 /
      Data Ic( 459  )/ 188 /,Ic( 460  )/ 456 /
      Data Ic( 461  )/ 189 /,Ic( 462  )/ 430 /
      Data Ic( 463  )/ 1   /,Ic( 464  )/ 162 /
      Data Ic( 465  )/ 9   /
C-----<< 58 : >>
```

```
      Data Ic( 466  )/ 66  /,Ic( 467  )/ 529 /
      Data Ic( 468  )/ 158 /,Ic( 469  )/ 410 /
C-----<< 59 ; >>
      Data Ic( 470  )/ 66  /,Ic( 471  )/ 529 /
      Data Ic( 472  )/ 114 /,Ic( 473  )/ 426 /
C-----<< 61 = >>
      Data Ic( 474  )/ 190 /,Ic( 475  )/ 425 /
      Data Ic( 476  )/ 1   /,Ic( 477  )/ 124 /
      Data Ic( 478  )/ 425 /
C-----<< 63 ? >>
      Data Ic( 479  )/ 457 /,Ic( 480  )/ 416 /
      Data Ic( 481  )/ 5   /,Ic( 482  )/ 458 /
      Data Ic( 483  )/ 134 /,Ic( 484  )/ 18  /
      Data Ic( 485  )/ 123 /,Ic( 486  )/ 432 /
      Data Ic( 487  )/ 29  /,Ic( 488  )/ 37  /
      Data Ic( 489  )/ 459 /,Ic( 490  )/ 460 /
      Data Ic( 491  )/ 192 /,Ic( 492  )/ 30  /
      Data Ic( 493  )/ 461 /,Ic( 494  )/ 193 /
      Data Ic( 495  )/ 54  /,Ic( 496  )/ 1   /
      Data Ic( 497  )/ 194 /,Ic( 498  )/ 13  /
      Data Ic( 499  )/ 71  /,Ic( 500  )/ 444 /
      Data Ic( 501  )/ 195 /,Ic( 502  )/ 34  /
      Data Ic( 503  )/ 173 /,Ic( 504  )/ 110 /
      Data Ic( 505  )/ 423 /,Ic( 506  )/ 71  /
      Data Ic( 507  )/ 410 /
C-----<< 65 A >>
      Data Ic( 508  )/ 196 /,Ic( 509  )/ 463 /
      Data Ic( 510  )/ 13  /,Ic( 511  )/ 197 /
      Data Ic( 512  )/ 464 /,Ic( 513  )/ 68  /
      Data Ic( 514  )/ 44  /,Ic( 515  )/ 6   /
      Data Ic( 516  )/ 30  /,Ic( 517  )/ 198 /
      Data Ic( 518  )/ 465 /,Ic( 519  )/ 466 /
      Data Ic( 520  )/ 13  /,Ic( 521  )/ 197 /
      Data Ic( 522  )/ 28  /,Ic( 523  )/ 12  /
      Data Ic( 524  )/ 461 /,Ic( 525  )/ 199 /
      Data Ic( 526  )/ 12  /,Ic( 527  )/ 4   /
      Data Ic( 528  )/ 467 /,Ic( 529  )/ 1   /
      Data Ic( 530  )/ 44  /,Ic( 531  )/ 17  /
      Data Ic( 532  )/ 1   /,Ic( 533  )/ 31  /
      Data Ic( 534  )/ 18  /,Ic( 535  )/ 28  /
      Data Ic( 536  )/ 1   /,Ic( 537  )/ 200 /
      Data Ic( 538  )/ 469 /,Ic( 539  )/ 150 /
      Data Ic( 540  )/ 13  /,Ic( 541  )/ 1   /
      Data Ic( 542  )/ 202 /,Ic( 543  )/ 203 /
      Data Ic( 544  )/ 1   /,Ic( 545  )/ 204 /
      Data Ic( 546  )/ 194 /
C-----<< 66 B >>
      Data Ic( 547  )/ 470 /,Ic( 548  )/ 209 /
      Data Ic( 549  )/ 471 /,Ic( 550  )/ 210 /
      Data Ic( 551  )/ 412 /,Ic( 552  )/ 29  /
      Data Ic( 553  )/ 15  /,Ic( 554  )/ 168 /
```

```
      Data Ic( 555  )/ 472 /,Ic( 556  )/ 211 /
      Data Ic( 557  )/ 53  /,Ic( 558  )/ 29  /
      Data Ic( 559  )/ 1   /,Ic( 560  )/ 212 /
      Data Ic( 561  )/ 455 /,Ic( 562  )/ 34  /
      Data Ic( 563  )/ 15  /,Ic( 564  )/ 35  /
      Data Ic( 565  )/ 28  /,Ic( 566  )/ 21  /
      Data Ic( 567  )/ 110 /,Ic( 568  )/ 473 /
      Data Ic( 569  )/ 5   /,Ic( 570  )/ 545 /
      Data Ic( 571  )/ 213 /,Ic( 572  )/ 546 /
      Data Ic( 573  )/ 214 /,Ic( 574  )/ 28  /
      Data Ic( 575  )/ 126 /,Ic( 576  )/ 1   /
      Data Ic( 577  )/ 215 /,Ic( 578  )/ 432 /
      Data Ic( 579  )/ 440 /,Ic( 580  )/ 216 /
      Data Ic( 581  )/ 134 /,Ic( 582  )/ 135 /
      Data Ic( 583  )/ 452 /,Ic( 584  )/ 183 /
      Data Ic( 585  )/ 475 /
C-----<< 67 C >>
      Data Ic( 586  )/ 217 /,Ic( 587  )/ 37  /
      Data Ic( 588  )/ 459 /,Ic( 589  )/ 544 /
      Data Ic( 590  )/ 218 /,Ic( 591  )/ 30  /
      Data Ic( 592  )/ 171 /,Ic( 593  )/ 192 /
      Data Ic( 594  )/ 533 /,Ic( 595  )/ 462 /
      Data Ic( 596  )/ 15  /,Ic( 597  )/ 424 /
      Data Ic( 598  )/ 71  /,Ic( 599  )/ 79  /
      Data Ic( 600  )/ 219 /,Ic( 601  )/ 123 /
      Data Ic( 602  )/ 476 /,Ic( 603  )/ 4   /
      Data Ic( 604  )/ 530 /,Ic( 605  )/ 220 /
      Data Ic( 606  )/ 76  /,Ic( 607  )/ 126 /
      Data Ic( 608  )/ 73  /,Ic( 609  )/ 35  /
      Data Ic( 610  )/ 123 /,Ic( 611  )/ 476 /
      Data Ic( 612  )/ 434 /,Ic( 613  )/ 158 /
      Data Ic( 614  )/ 418 /,Ic( 615  )/ 86  /
      Data Ic( 616  )/ 477 /,Ic( 617  )/ 158 /
      Data Ic( 618  )/ 1   /,Ic( 619  )/ 222 /
      Data Ic( 620  )/ 9   /,Ic( 621  )/ 420 /
      Data Ic( 622  )/ 7   /,Ic( 623  )/ 14  /
C-----<< 68 D >>
      Data Ic( 624  )/ 223 /,Ic( 625  )/ 224 /
      Data Ic( 626  )/ 6   /,Ic( 627  )/ 225 /
      Data Ic( 628  )/ 518 /,Ic( 629  )/ 226 /
      Data Ic( 630  )/ 412 /,Ic( 631  )/ 14  /
      Data Ic( 632  )/ 110 /,Ic( 633  )/ 473 /
      Data Ic( 634  )/ 5   /,Ic( 635  )/ 545 /
      Data Ic( 636  )/ 213 /,Ic( 637  )/ 546 /
      Data Ic( 638  )/ 227 /,Ic( 639  )/ 228 /
      Data Ic( 640  )/ 53  /,Ic( 641  )/ 478 /
      Data Ic( 642  )/ 229 /,Ic( 643  )/ 479 /
      Data Ic( 644  )/ 231 /,Ic( 645  )/ 134 /
      Data Ic( 646  )/ 135 /,Ic( 647  )/ 452 /
      Data Ic( 648  )/ 232 /,Ic( 649  )/ 525 /
      Data Ic( 650  )/ 11  /,Ic( 651  )/ 480 /
```

```
C------<< 69 E >>
      Data Ic( 652 )/ 457 /,Ic( 653 )/ 481 /
      Data Ic( 654 )/ 531 /,Ic( 655 )/ 9   /
      Data Ic( 656 )/ 18  /,Ic( 657 )/ 530 /
      Data Ic( 658 )/ 111 /,Ic( 659 )/ 420 /
      Data Ic( 660 )/ 233 /,Ic( 661 )/ 532 /
      Data Ic( 662 )/ 234 /,Ic( 663 )/ 11  /
      Data Ic( 664 )/ 173 /,Ic( 665 )/ 458 /
      Data Ic( 666 )/ 4   /,Ic( 667 )/ 18  /
      Data Ic( 668 )/ 123 /,Ic( 669 )/ 15  /
      Data Ic( 670 )/ 481 /,Ic( 671 )/ 8   /
      Data Ic( 672 )/ 444 /,Ic( 673 )/ 463 /
      Data Ic( 674 )/ 452 /,Ic( 675 )/ 235 /
      Data Ic( 676 )/ 173 /,Ic( 677 )/ 458 /
      Data Ic( 678 )/ 527 /,Ic( 679 )/ 15  /
      Data Ic( 680 )/ 465 /,Ic( 681 )/ 444 /
      Data Ic( 682 )/ 103 /,Ic( 683 )/ 482 /
      Data Ic( 684 )/ 1   /,Ic( 685 )/ 234 /
      Data Ic( 686 )/ 483 /
C------<< 70 F >>
      Data Ic( 687 )/ 238 /,Ic( 688 )/ 8   /
      Data Ic( 689 )/ 537 /,Ic( 690 )/ 515 /
      Data Ic( 691 )/ 239 /,Ic( 692 )/ 479 /
      Data Ic( 693 )/ 240 /,Ic( 694 )/ 55  /
      Data Ic( 695 )/ 21  /,Ic( 696 )/ 37  /
      Data Ic( 697 )/ 168 /,Ic( 698 )/ 435 /
      Data Ic( 699 )/ 110 /,Ic( 700 )/ 33  /
      Data Ic( 701 )/ 463 /,Ic( 702 )/ 241 /
      Data Ic( 703 )/ 43  /,Ic( 704 )/ 12  /
      Data Ic( 705 )/ 224 /,Ic( 706 )/ 477 /
      Data Ic( 707 )/ 478 /,Ic( 708 )/ 242 /
      Data Ic( 709 )/ 474 /,Ic( 710 )/ 243 /
      Data Ic( 711 )/ 431 /,Ic( 712 )/ 55  /
      Data Ic( 713 )/ 1   /,Ic( 714 )/ 244 /
      Data Ic( 715 )/ 21  /,Ic( 716 )/ 422 /
      Data Ic( 717 )/ 183 /,Ic( 718 )/ 9   /
      Data Ic( 719 )/ 420 /,Ic( 720 )/ 33  /
      Data Ic( 721 )/ 148 /
C------<< 71 G >>
      Data Ic( 722 )/ 113 /,Ic( 723 )/ 16  /
      Data Ic( 724 )/ 30  /,Ic( 725 )/ 171 /
      Data Ic( 726 )/ 146 /,Ic( 727 )/ 533 /
      Data Ic( 728 )/ 103 /,Ic( 729 )/ 473 /
      Data Ic( 730 )/ 424 /,Ic( 731 )/ 34  /
      Data Ic( 732 )/ 484 /,Ic( 733 )/ 9   /
      Data Ic( 734 )/ 123 /,Ic( 735 )/ 150 /
      Data Ic( 736 )/ 134 /,Ic( 737 )/ 460 /
      Data Ic( 738 )/ 172 /,Ic( 739 )/ 413 /
      Data Ic( 740 )/ 481 /,Ic( 741 )/ 9   /
      Data Ic( 742 )/ 432 /,Ic( 743 )/ 71  /
      Data Ic( 744 )/ 37  /,Ic( 745 )/ 408 /
```

```
      Data Ic( 746  )/ 86  /,Ic( 747  )/ 245 /
      Data Ic( 748  )/ 1   /,Ic( 749  )/ 246 /
      Data Ic( 750  )/ 76  /,Ic( 751  )/ 126 /
      Data Ic( 752  )/ 423 /,Ic( 753  )/ 247 /
      Data Ic( 754  )/ 12  /,Ic( 755  )/ 192 /
      Data Ic( 756  )/ 30  /,Ic( 757  )/ 1   /
      Data Ic( 758  )/ 248 /,Ic( 759  )/ 200 /
      Data Ic( 760  )/ 418 /,Ic( 761  )/ 249 /
      Data Ic( 762  )/ 477 /,Ic( 763  )/ 200 /
      Data Ic( 764  )/ 1   /,Ic( 765  )/ 250 /
      Data Ic( 766  )/ 9   /,Ic( 767  )/ 420 /
      Data Ic( 768  )/ 251 /,Ic( 769  )/ 475 /
C-----<< 72 H >>
      Data Ic( 770  )/ 485 /,Ic( 771  )/ 4   /
      Data Ic( 772  )/ 103 /,Ic( 773  )/ 13  /
      Data Ic( 774  )/ 79  /,Ic( 775  )/ 71  /
      Data Ic( 776  )/ 77  /,Ic( 777  )/ 68  /
      Data Ic( 778  )/ 22  /,Ic( 779  )/ 146 /
      Data Ic( 780  )/ 30  /,Ic( 781  )/ 451 /
      Data Ic( 782  )/ 254 /,Ic( 783  )/ 539 /
      Data Ic( 784  )/ 486 /,Ic( 785  )/ 513 /
      Data Ic( 786  )/ 70  /,Ic( 787  )/ 432 /
      Data Ic( 788  )/ 126 /,Ic( 789  )/ 76  /
      Data Ic( 790  )/ 1   /,Ic( 791  )/ 56  /
      Data Ic( 792  )/ 475 /
C-----<< 73 I >>
      Data Ic( 793  )/ 487 /,Ic( 794  )/ 133 /
      Data Ic( 795  )/ 463 /,Ic( 796  )/ 241 /
      Data Ic( 797  )/ 18  /,Ic( 798  )/ 488 /
      Data Ic( 799  )/ 459 /,Ic( 800  )/ 55  /
      Data Ic( 801  )/ 5   /,Ic( 802  )/ 55  /
      Data Ic( 803  )/ 519 /,Ic( 804  )/ 258 /
      Data Ic( 805  )/ 257 /,Ic( 806  )/ 1   /
      Data Ic( 807  )/ 259 /,Ic( 808  )/ 431 /
      Data Ic( 809  )/ 55  /
C-----<< 74 J >>
      Data Ic( 810  )/ 487 /,Ic( 811  )/ 260 /
      Data Ic( 812  )/ 488 /,Ic( 813  )/ 15  /
      Data Ic( 814  )/ 544 /,Ic( 815  )/ 16  /
      Data Ic( 816  )/ 102 /,Ic( 817  )/ 12  /
      Data Ic( 818  )/ 43  /,Ic( 819  )/ 13  /
      Data Ic( 820  )/ 7   /,Ic( 821  )/ 451 /
      Data Ic( 822  )/ 261 /,Ic( 823  )/ 257 /
      Data Ic( 824  )/ 1   /,Ic( 825  )/ 262 /
      Data Ic( 826  )/ 63  /
C-----<< 75 K >>
      Data Ic( 827  )/ 485 /,Ic( 828  )/ 221 /
      Data Ic( 829  )/ 522 /,Ic( 830  )/ 435 /
      Data Ic( 831  )/ 1   /,Ic( 832  )/ 263 /
      Data Ic( 833  )/ 4   /,Ic( 834  )/ 150 /
      Data Ic( 835  )/ 486 /,Ic( 836  )/ 513 /
```

```
      Data Ic( 837  )/ 264 /,Ic( 838  )/ 68  /
      Data Ic( 839  )/ 15  /,Ic( 840  )/ 15  /
      Data Ic( 841  )/ 1   /,Ic( 842  )/ 265 /
      Data Ic( 843  )/ 44  /,Ic( 844  )/ 30  /
      Data Ic( 845  )/ 266 /,Ic( 846  )/ 5   /
      Data Ic( 847  )/ 13  /,Ic( 848  )/ 82  /
      Data Ic( 849  )/ 417 /,Ic( 850  )/ 12  /
      Data Ic( 851  )/ 1   /,Ic( 852  )/ 59  /
      Data Ic( 853  )/ 78  /
C------<< 76 L >>
      Data Ic( 854  )/ 267 /,Ic( 855  )/ 8   /
      Data Ic( 856  )/ 537 /,Ic( 857  )/ 515 /
      Data Ic( 858  )/ 206 /,Ic( 859  )/ 532 /
      Data Ic( 860  )/ 268 /,Ic( 861  )/ 483 /
      Data Ic( 862  )/ 1   /,Ic( 863  )/ 269 /
      Data Ic( 864  )/ 471 /,Ic( 865  )/ 513 /
      Data Ic( 866  )/ 31  /,Ic( 867  )/ 97  /
C------<< 77 M >>
      Data Ic( 868  )/ 22  /,Ic( 869  )/ 192 /
      Data Ic( 870  )/ 538 /,Ic( 871  )/ 102 /
      Data Ic( 872  )/ 514 /,Ic( 873  )/ 270 /
      Data Ic( 874  )/ 422 /,Ic( 875  )/ 234 /
      Data Ic( 876  )/ 418 /,Ic( 877  )/ 44  /
      Data Ic( 878  )/ 468 /,Ic( 879  )/ 8   /
      Data Ic( 880  )/ 463 /,Ic( 881  )/ 442 /
      Data Ic( 882  )/ 12  /,Ic( 883  )/ 112 /
      Data Ic( 884  )/ 1   /,Ic( 885  )/ 271 /
      Data Ic( 886  )/ 489 /,Ic( 887  )/ 272 /
      Data Ic( 888  )/ 490 /,Ic( 889  )/ 1   /
      Data Ic( 890  )/ 274 /,Ic( 891  )/ 491 /
      Data Ic( 892  )/ 520 /,Ic( 893  )/ 12  /
      Data Ic( 894  )/ 419 /,Ic( 895  )/ 275 /
      Data Ic( 896  )/ 422 /,Ic( 897  )/ 492 /
      Data Ic( 898  )/ 1   /,Ic( 899  )/ 29  /
      Data Ic( 900  )/ 63  /,Ic( 901  )/ 137 /
      Data Ic( 902  )/ 7   /,Ic( 903  )/ 53  /
      Data Ic( 904  )/ 213 /,Ic( 905  )/ 12  /
      Data Ic( 906  )/ 112 /,Ic( 907  )/ 1   /
      Data Ic( 908  )/ 119 /,Ic( 909  )/ 490 /
      Data Ic( 910  )/ 15  /,Ic( 911  )/ 16  /
      Data Ic( 912  )/ 242 /,Ic( 913  )/ 55  /
      Data Ic( 914  )/ 1   /,Ic( 915  )/ 207 /
      Data Ic( 916  )/ 491 /,Ic( 917  )/ 53  /
      Data Ic( 918  )/ 1   /,Ic( 919  )/ 277 /
      Data Ic( 920  )/ 422 /,Ic( 921  )/ 492 /
C------<< 78 N >>
      Data Ic( 922  )/ 116 /,Ic( 923  )/ 462 /
      Data Ic( 924  )/ 493 /,Ic( 925  )/ 524 /
      Data Ic( 926  )/ 279 /,Ic( 927  )/ 53  /
      Data Ic( 928  )/ 28  /,Ic( 929  )/ 280 /
      Data Ic( 930  )/ 219 /,Ic( 931  )/ 1   /
```

```
      Data Ic( 932  )/ 281 /,Ic( 933  )/ 493 /
      Data Ic( 934  )/ 522 /,Ic( 935  )/ 282 /
      Data Ic( 936  )/ 1   /,Ic( 937  )/ 283 /
      Data Ic( 938  )/ 534 /,Ic( 939  )/ 435 /
      Data Ic( 940  )/ 103 /,Ic( 941  )/ 9   /
      Data Ic( 942  )/ 421 /,Ic( 943  )/ 1   /
      Data Ic( 944  )/ 205 /,Ic( 945  )/ 9   /
      Data Ic( 946  )/ 420 /,Ic( 947  )/ 284 /
      Data Ic( 948  )/ 110 /,Ic( 949  )/ 7   /
      Data Ic( 950  )/ 29  /,Ic( 951  )/ 12  /
      Data Ic( 952  )/ 420 /,Ic( 953  )/ 5   /
      Data Ic( 954  )/ 420 /,Ic( 955  )/ 89  /
      Data Ic( 956  )/ 469 /,Ic( 957  )/ 489 /
      Data Ic( 958  )/ 285 /,Ic( 959  )/ 97  /
      Data Ic( 960  )/ 1   /,Ic( 961  )/ 286 /
      Data Ic( 962  )/ 522 /,Ic( 963  )/ 436 /
      Data Ic( 964  )/ 1   /,Ic( 965  )/ 287 /
      Data Ic( 966  )/ 4   /,Ic( 967  )/ 150 /
      Data Ic( 968  )/ 9   /
C-----<< 79 O >>
      Data Ic( 969  )/ 494 /,Ic( 970  )/ 484 /
      Data Ic( 971  )/ 9   /,Ic( 972  )/ 437 /
      Data Ic( 973  )/ 495 /,Ic( 974  )/ 496 /
C-----<< 80 P >>
      Data Ic( 975  )/ 293 /,Ic( 976  )/ 15  /
      Data Ic( 977  )/ 242 /,Ic( 978  )/ 33  /
      Data Ic( 979  )/ 213 /,Ic( 980  )/ 242 /
      Data Ic( 981  )/ 538 /,Ic( 982  )/ 179 /
      Data Ic( 983  )/ 413 /,Ic( 984  )/ 9   /
      Data Ic( 985  )/ 432 /,Ic( 986  )/ 294 /
      Data Ic( 987  )/ 1   /,Ic( 988  )/ 85  /
      Data Ic( 989  )/ 28  /,Ic( 990  )/ 294 /
      Data Ic( 991  )/ 1   /,Ic( 992  )/ 295 /
      Data Ic( 993  )/ 422 /,Ic( 994  )/ 182 /
      Data Ic( 995  )/ 213 /,Ic( 996  )/ 283 /
      Data Ic( 997  )/ 273 /,Ic( 998  )/ 148 /
      Data Ic( 999  )/ 473 /,Ic( 1000 )/ 5   /
      Data Ic( 1001 )/ 110 /,Ic( 1002 )/ 1   /
      Data Ic( 1003 )/ 296 /,Ic( 1004 )/ 491 /
      Data Ic( 1005 )/ 434 /,Ic( 1006 )/ 43  /
      Data Ic( 1007 )/ 491 /,Ic( 1008 )/ 31  /
      Data Ic( 1009 )/ 55  /,Ic( 1010 )/ 1   /
      Data Ic( 1011 )/ 464 /,Ic( 1012 )/ 452 /
      Data Ic( 1013 )/ 188 /,Ic( 1014 )/ 294 /
      Data Ic( 1015 )/ 1   /,Ic( 1016 )/ 282 /
      Data Ic( 1017 )/ 480 /
C-----<< 81 Q >>
      Data Ic( 1018 )/ 494 /,Ic( 1019 )/ 484 /
      Data Ic( 1020 )/ 9   /,Ic( 1021 )/ 9   /
      Data Ic( 1022 )/ 150 /,Ic( 1023 )/ 4   /
      Data Ic( 1024 )/ 4   /,Ic( 1025 )/ 496 /
```

```
      Data Ic( 1026 )/ 1   /,Ic( 1027 )/ 297 /
      Data Ic( 1028 )/ 12  /,Ic( 1029 )/ 526 /
      Data Ic( 1030 )/ 209 /,Ic( 1031 )/ 442 /
      Data Ic( 1032 )/ 7   /,Ic( 1033 )/ 435 /
      Data Ic( 1034 )/ 298 /,Ic( 1035 )/ 419 /
      Data Ic( 1036 )/ 53  /,Ic( 1037 )/ 74  /
      Data Ic( 1038 )/ 53  /,Ic( 1039 )/ 43  /
C-----<< 82 R >>
      Data Ic( 1040 )/ 470 /,Ic( 1041 )/ 252 /
      Data Ic( 1042 )/ 31  /,Ic( 1043 )/ 110 /
      Data Ic( 1044 )/ 33  /,Ic( 1045 )/ 119 /
      Data Ic( 1046 )/ 128 /,Ic( 1047 )/ 63  /
      Data Ic( 1048 )/ 9   /,Ic( 1049 )/ 530 /
      Data Ic( 1050 )/ 299 /,Ic( 1051 )/ 4   /
      Data Ic( 1052 )/ 150 /,Ic( 1053 )/ 486 /
      Data Ic( 1054 )/ 213 /,Ic( 1055 )/ 4   /
      Data Ic( 1056 )/ 412 /,Ic( 1057 )/ 414 /
      Data Ic( 1058 )/ 7   /,Ic( 1059 )/ 168 /
      Data Ic( 1060 )/ 519 /,Ic( 1061 )/ 300 /
      Data Ic( 1062 )/ 43  /,Ic( 1063 )/ 28  /
      Data Ic( 1064 )/ 71  /,Ic( 1065 )/ 444 /
      Data Ic( 1066 )/ 301 /,Ic( 1067 )/ 412 /
      Data Ic( 1068 )/ 414 /,Ic( 1069 )/ 119 /
      Data Ic( 1070 )/ 455 /,Ic( 1071 )/ 302 /
      Data Ic( 1072 )/ 497 /,Ic( 1073 )/ 303 /
      Data Ic( 1074 )/ 16  /,Ic( 1075 )/ 1   /
      Data Ic( 1076 )/ 219 /,Ic( 1077 )/ 73  /
      Data Ic( 1078 )/ 53  /
C-----<< 83 S >>
      Data Ic( 1079 )/ 304 /,Ic( 1080 )/ 521 /
      Data Ic( 1081 )/ 31  /,Ic( 1082 )/ 1   /
      Data Ic( 1083 )/ 305 /,Ic( 1084 )/ 461 /
      Data Ic( 1085 )/ 306 /,Ic( 1086 )/ 173 /
      Data Ic( 1087 )/ 102 /,Ic( 1088 )/ 521 /
      Data Ic( 1089 )/ 307 /,Ic( 1090 )/ 31  /
      Data Ic( 1091 )/ 6   /,Ic( 1092 )/ 30  /
      Data Ic( 1093 )/ 458 /,Ic( 1094 )/ 134 /
      Data Ic( 1095 )/ 150 /,Ic( 1096 )/ 129 /
      Data Ic( 1097 )/ 15  /,Ic( 1098 )/ 68  /
      Data Ic( 1099 )/ 1   /,Ic( 1100 )/ 308 /
      Data Ic( 1101 )/ 13  /,Ic( 1102 )/ 123 /
      Data Ic( 1103 )/ 155 /,Ic( 1104 )/ 455 /
      Data Ic( 1105 )/ 29  /,Ic( 1106 )/ 37  /
      Data Ic( 1107 )/ 1   /,Ic( 1108 )/ 309 /
      Data Ic( 1109 )/ 29  /,Ic( 1110 )/ 13  /
      Data Ic( 1111 )/ 123 /,Ic( 1112 )/ 310 /
      Data Ic( 1113 )/ 412 /,Ic( 1114 )/ 29  /
      Data Ic( 1115 )/ 311 /,Ic( 1116 )/ 312 /
      Data Ic( 1117 )/ 31  /,Ic( 1118 )/ 545 /
      Data Ic( 1119 )/ 213 /,Ic( 1120 )/ 128 /
      Data Ic( 1121 )/ 63  /,Ic( 1122 )/ 123 /
```

```
      Data Ic( 1123 )/ 1   /,Ic( 1124 )/ 266 /
      Data Ic( 1125 )/ 54  /,Ic( 1126 )/ 45  /
      Data Ic( 1127 )/ 6   /,Ic( 1128 )/ 145 /
      Data Ic( 1129 )/ 518 /,Ic( 1130 )/ 17  /
      Data Ic( 1131 )/ 415 /,Ic( 1132 )/ 236 /
      Data Ic( 1133 )/ 9   /,Ic( 1134 )/ 17  /
      Data Ic( 1135 )/ 1   /,Ic( 1136 )/ 313 /
      Data Ic( 1137 )/ 9   /,Ic( 1138 )/ 194 /
      Data Ic( 1139 )/ 13  /,Ic( 1140 )/ 1   /
      Data Ic( 1141 )/ 314 /,Ic( 1142 )/ 546 /
      Data Ic( 1143 )/ 315 /,Ic( 1144 )/ 68  /
      Data Ic( 1145 )/ 1   /,Ic( 1146 )/ 37  /
      Data Ic( 1147 )/ 459 /,Ic( 1148 )/ 44  /
      Data Ic( 1149 )/ 45  /,Ic( 1150 )/ 13  /
C-----<< 84 T >>
      Data Ic( 1151 )/ 217 /,Ic( 1152 )/ 37  /
      Data Ic( 1153 )/ 459 /,Ic( 1154 )/ 544 /
      Data Ic( 1155 )/ 5   /,Ic( 1156 )/ 291 /
      Data Ic( 1157 )/ 171 /,Ic( 1158 )/ 192 /
      Data Ic( 1159 )/ 533 /,Ic( 1160 )/ 76  /
      Data Ic( 1161 )/ 71  /,Ic( 1162 )/ 79  /
      Data Ic( 1163 )/ 74  /,Ic( 1164 )/ 9   /
      Data Ic( 1165 )/ 123 /,Ic( 1166 )/ 476 /
      Data Ic( 1167 )/ 4   /,Ic( 1168 )/ 530 /
      Data Ic( 1169 )/ 316 /,Ic( 1170 )/ 79  /
      Data Ic( 1171 )/ 53  /,Ic( 1172 )/ 9   /
      Data Ic( 1173 )/ 123 /,Ic( 1174 )/ 63  /
      Data Ic( 1175 )/ 454 /,Ic( 1176 )/ 296 /
      Data Ic( 1177 )/ 8   /,Ic( 1178 )/ 32  /
      Data Ic( 1179 )/ 317 /,Ic( 1180 )/ 32  /
      Data Ic( 1181 )/ 8   /,Ic( 1182 )/ 37  /
      Data Ic( 1183 )/ 12  /,Ic( 1184 )/ 4   /
      Data Ic( 1185 )/ 63  /,Ic( 1186 )/ 36  /
      Data Ic( 1187 )/ 63  /,Ic( 1188 )/ 4   /
      Data Ic( 1189 )/ 530 /,Ic( 1190 )/ 318 /
      Data Ic( 1191 )/ 129 /,Ic( 1192 )/ 474 /
      Data Ic( 1193 )/ 111 /,Ic( 1194 )/ 408 /
      Data Ic( 1195 )/ 200 /,Ic( 1196 )/ 12  /
      Data Ic( 1197 )/ 54  /,Ic( 1198 )/ 163 /
      Data Ic( 1199 )/ 1   /,Ic( 1200 )/ 7   /
      Data Ic( 1201 )/ 472 /,Ic( 1202 )/ 319 /
      Data Ic( 1203 )/ 14  /
C-----<< 85 U >>
      Data Ic( 1204 )/ 457 /,Ic( 1205 )/ 31  /
      Data Ic( 1206 )/ 110 /,Ic( 1207 )/ 531 /
      Data Ic( 1208 )/ 437 /,Ic( 1209 )/ 515 /
      Data Ic( 1210 )/ 240 /,Ic( 1211 )/ 13  /
      Data Ic( 1212 )/ 14  /,Ic( 1213 )/ 33  /
      Data Ic( 1214 )/ 32  /,Ic( 1215 )/ 31  /
      Data Ic( 1216 )/ 16  /,Ic( 1217 )/ 30  /
      Data Ic( 1218 )/ 171 /,Ic( 1219 )/ 172 /
```

```
      Data Ic( 1220 )/ 320 /,Ic( 1221 )/ 424 /
      Data Ic( 1222 )/ 71  /,Ic( 1223 )/ 484 /
      Data Ic( 1224 )/ 123 /,Ic( 1225 )/ 476 /
      Data Ic( 1226 )/ 4   /,Ic( 1227 )/ 53  /
      Data Ic( 1228 )/ 103 /,Ic( 1229 )/ 44  /
      Data Ic( 1230 )/ 30  /,Ic( 1231 )/ 321 /
      Data Ic( 1232 )/ 4   /,Ic( 1233 )/ 518 /
      Data Ic( 1234 )/ 15  /,Ic( 1235 )/ 448 /
      Data Ic( 1236 )/ 37  /,Ic( 1237 )/ 158 /
      Data Ic( 1238 )/ 12  /,Ic( 1239 )/ 54  /
      Data Ic( 1240 )/ 163 /,Ic( 1241 )/ 1   /
      Data Ic( 1242 )/ 322 /,Ic( 1243 )/ 79  /
      Data Ic( 1244 )/ 219 /,Ic( 1245 )/ 123 /
      Data Ic( 1246 )/ 422 /,Ic( 1247 )/ 323 /
      Data Ic( 1248 )/ 472 /,Ic( 1249 )/ 152 /
      Data Ic( 1250 )/ 498 /,Ic( 1251 )/ 14  /
      Data Ic( 1252 )/ 53  /,Ic( 1253 )/ 1   /
      Data Ic( 1254 )/ 324 /,Ic( 1255 )/ 525 /
      Data Ic( 1256 )/ 11  /,Ic( 1257 )/ 422 /
      Data Ic( 1258 )/ 311 /,Ic( 1259 )/ 63  /
C-----<< 86 V >>
      Data Ic( 1260 )/ 499 /,Ic( 1261 )/ 500 /
      Data Ic( 1262 )/ 224 /,Ic( 1263 )/ 4   /
      Data Ic( 1264 )/ 518 /,Ic( 1265 )/ 459 /
      Data Ic( 1266 )/ 33  /,Ic( 1267 )/ 478 /
      Data Ic( 1268 )/ 326 /,Ic( 1269 )/ 501 /
      Data Ic( 1270 )/ 489 /,Ic( 1271 )/ 328 /
      Data Ic( 1272 )/ 498 /,Ic( 1273 )/ 478 /
      Data Ic( 1274 )/ 329 /,Ic( 1275 )/ 13  /
      Data Ic( 1276 )/ 14  /,Ic( 1277 )/ 1   /
      Data Ic( 1278 )/ 330 /,Ic( 1279 )/ 480 /
C-----<< 87 W >>
      Data Ic( 1280 )/ 331 /,Ic( 1281 )/ 468 /
      Data Ic( 1282 )/ 502 /,Ic( 1283 )/ 7   /
      Data Ic( 1284 )/ 468 /,Ic( 1285 )/ 516 /
      Data Ic( 1286 )/ 13  /,Ic( 1287 )/ 103 /
      Data Ic( 1288 )/ 134 /,Ic( 1289 )/ 412 /
      Data Ic( 1290 )/ 103 /,Ic( 1291 )/ 134 /
      Data Ic( 1292 )/ 502 /,Ic( 1293 )/ 444 /
      Data Ic( 1294 )/ 332 /,Ic( 1295 )/ 501 /
      Data Ic( 1296 )/ 18  /,Ic( 1297 )/ 452 /
      Data Ic( 1298 )/ 117 /,Ic( 1299 )/ 13  /
      Data Ic( 1300 )/ 525 /,Ic( 1301 )/ 129 /
      Data Ic( 1302 )/ 524 /,Ic( 1303 )/ 117 /
      Data Ic( 1304 )/ 13  /,Ic( 1305 )/ 525 /
      Data Ic( 1306 )/ 333 /,Ic( 1307 )/ 525 /
      Data Ic( 1308 )/ 334 /,Ic( 1309 )/ 525 /
      Data Ic( 1310 )/ 335 /,Ic( 1311 )/ 422 /
      Data Ic( 1312 )/ 311 /,Ic( 1313 )/ 422 /
      Data Ic( 1314 )/ 336 /,Ic( 1315 )/ 422 /
      Data Ic( 1316 )/ 311 /,Ic( 1317 )/ 63  /
```

```
C------<< 88 X >>
      Data Ic( 1318 )/ 337 /,Ic( 1319 )/ 463 /
      Data Ic( 1320 )/ 13  /,Ic( 1321 )/ 338 /
      Data Ic( 1322 )/ 13  /,Ic( 1323 )/ 420 /
      Data Ic( 1324 )/ 339 /,Ic( 1325 )/ 13  /
      Data Ic( 1326 )/ 340 /,Ic( 1327 )/ 524 /
      Data Ic( 1328 )/ 339 /,Ic( 1329 )/ 482 /
      Data Ic( 1330 )/ 338 /,Ic( 1331 )/ 13  /
      Data Ic( 1332 )/ 436 /,Ic( 1333 )/ 530 /
      Data Ic( 1334 )/ 341 /,Ic( 1335 )/ 342 /
      Data Ic( 1336 )/ 9   /,Ic( 1337 )/ 421 /
      Data Ic( 1338 )/ 534 /,Ic( 1339 )/ 110 /
      Data Ic( 1340 )/ 419 /,Ic( 1341 )/ 12  /
      Data Ic( 1342 )/ 13  /,Ic( 1343 )/ 420 /
      Data Ic( 1344 )/ 343 /,Ic( 1345 )/ 465 /
      Data Ic( 1346 )/ 8   /,Ic( 1347 )/ 37  /
      Data Ic( 1348 )/ 12  /,Ic( 1349 )/ 527 /
      Data Ic( 1350 )/ 342 /,Ic( 1351 )/ 1   /
      Data Ic( 1352 )/ 344 /,Ic( 1353 )/ 482 /
      Data Ic( 1354 )/ 1   /,Ic( 1355 )/ 345 /
      Data Ic( 1356 )/ 150 /,Ic( 1357 )/ 466 /
      Data Ic( 1358 )/ 135 /
C------<< 89 Y >>
      Data Ic( 1359 )/ 499 /,Ic( 1360 )/ 454 /
      Data Ic( 1361 )/ 333 /,Ic( 1362 )/ 501 /
      Data Ic( 1363 )/ 489 /,Ic( 1364 )/ 30  /
      Data Ic( 1365 )/ 321 /,Ic( 1366 )/ 134 /
      Data Ic( 1367 )/ 4   /,Ic( 1368 )/ 103 /
      Data Ic( 1369 )/ 522 /,Ic( 1370 )/ 8   /
      Data Ic( 1371 )/ 346 /,Ic( 1372 )/ 37  /
      Data Ic( 1373 )/ 15  /,Ic( 1374 )/ 5   /
      Data Ic( 1375 )/ 169 /,Ic( 1376 )/ 130 /
      Data Ic( 1377 )/ 128 /,Ic( 1378 )/ 347 /
      Data Ic( 1379 )/ 123 /,Ic( 1380 )/ 1   /
      Data Ic( 1381 )/ 348 /,Ic( 1382 )/ 498 /
      Data Ic( 1383 )/ 525 /,Ic( 1384 )/ 349 /
      Data Ic( 1385 )/ 13  /,Ic( 1386 )/ 14  /
      Data Ic( 1387 )/ 74  /,Ic( 1388 )/ 1   /
      Data Ic( 1389 )/ 350 /,Ic( 1390 )/ 468 /
      Data Ic( 1391 )/ 1   /,Ic( 1392 )/ 351 /
      Data Ic( 1393 )/ 480 /
C------<< 90 Z >>
      Data Ic( 1394 )/ 495 /,Ic( 1395 )/ 128 /
      Data Ic( 1396 )/ 150 /,Ic( 1397 )/ 129 /
      Data Ic( 1398 )/ 15  /,Ic( 1399 )/ 31  /
      Data Ic( 1400 )/ 201 /,Ic( 1401 )/ 111 /
      Data Ic( 1402 )/ 352 /,Ic( 1403 )/ 353 /
      Data Ic( 1404 )/ 352 /,Ic( 1405 )/ 354 /
      Data Ic( 1406 )/ 54  /,Ic( 1407 )/ 448 /
      Data Ic( 1408 )/ 32  /,Ic( 1409 )/ 169 /
      Data Ic( 1410 )/ 103 /,Ic( 1411 )/ 123 /
```

```
      Data Ic( 1412 )/ 347 /,Ic( 1413 )/ 113 /
      Data Ic( 1414 )/ 311 /,Ic( 1415 )/ 69  /
      Data Ic( 1416 )/ 311 /,Ic( 1417 )/ 50  /
      Data Ic( 1418 )/ 37  /,Ic( 1419 )/ 1   /
      Data Ic( 1420 )/ 355 /,Ic( 1421 )/ 356 /
      Data Ic( 1422 )/ 1   /,Ic( 1423 )/ 323 /
      Data Ic( 1424 )/ 129 /,Ic( 1425 )/ 150 /
      Data Ic( 1426 )/ 128 /,Ic( 1427 )/ 1   /
      Data Ic( 1428 )/ 357 /,Ic( 1429 )/ 128 /
      Data Ic( 1430 )/ 150 /,Ic( 1431 )/ 129 /
      Data Ic( 1432 )/ 1   /,Ic( 1433 )/ 358 /
      Data Ic( 1434 )/ 150 /,Ic( 1435 )/ 466 /
      Data Ic( 1436 )/ 150 /
C-----<< 91 [ >>
      Data Ic( 1437 )/ 359 /,Ic( 1438 )/ 8   /
      Data Ic( 1439 )/ 416 /,Ic( 1440 )/ 28  /
      Data Ic( 1441 )/ 103 /,Ic( 1442 )/ 16  /
      Data Ic( 1443 )/ 102 /,Ic( 1444 )/ 1   /
      Data Ic( 1445 )/ 34  /,Ic( 1446 )/ 29  /
      Data Ic( 1447 )/ 12  /,Ic( 1448 )/ 6   /
C-----<< 92 \ >>
      Data Ic( 1449 )/ 360 /,Ic( 1450 )/ 426 /
C-----<< 97 A >>
      Data Ic( 1451 )/ 361 /,Ic( 1452 )/ 15  /
      Data Ic( 1453 )/ 416 /,Ic( 1454 )/ 415 /
      Data Ic( 1455 )/ 103 /,Ic( 1456 )/ 454 /
      Data Ic( 1457 )/ 362 /,Ic( 1458 )/ 28  /
      Data Ic( 1459 )/ 522 /,Ic( 1460 )/ 1    /
      Data Ic( 1461 )/ 298  /,Ic( 1462 )/ 29   /
      Data Ic( 1463 )/ 415  /,Ic( 1464 )/ 43   /
      Data Ic( 1465 )/ 1    /,Ic( 1466 )/ 363  /
      Data Ic( 1467 )/ 18   /,Ic( 1468 )/ 134  /
      Data Ic( 1469 )/ 447  /,Ic( 1470 )/ 53   /
      Data Ic( 1471 )/ 7    /,Ic( 1472 )/ 124  /
      Data Ic( 1473 )/ 497  /,Ic( 1474 )/ 163  /
      Data Ic( 1475 )/ 31   /,Ic( 1476 )/ 518  /
      Data Ic( 1477 )/ 15   /,Ic( 1478 )/ 62   /
      Data Ic( 1479 )/ 1    /,Ic( 1480 )/ 298  /
      Data Ic( 1481 )/ 7    /,Ic( 1482 )/ 422  /
      Data Ic( 1483 )/ 63   /,Ic( 1484 )/ 63   /
      Data Ic( 1485 )/ 6    /,Ic( 1486 )/ 200  /
      Data Ic( 1487 )/ 13   /
C-----<< 98 B >>
      Data Ic( 1488 )/ 364  /,Ic( 1489 )/ 521  /
      Data Ic( 1490 )/ 282  /,Ic( 1491 )/ 119  /
      Data Ic( 1492 )/ 28   /,Ic( 1493 )/ 148  /
      Data Ic( 1494 )/ 449  /,Ic( 1495 )/ 500  /
      Data Ic( 1496 )/ 450  /,Ic( 1497 )/ 33   /
      Data Ic( 1498 )/ 1    /,Ic( 1499 )/ 143  /
      Data Ic( 1500 )/ 30   /,Ic( 1501 )/ 12   /
      Data Ic( 1502 )/ 97   /,Ic( 1503 )/ 523  /
```

```
      Data Ic( 1504 )/ 517  /,Ic( 1505 )/ 472  /
      Data Ic( 1506 )/ 363  /,Ic( 1507 )/ 453  /
C------<< 99 C >>
      Data Ic( 1508 )/ 286  /,Ic( 1509 )/ 157  /
      Data Ic( 1510 )/ 443  /,Ic( 1511 )/ 528  /
      Data Ic( 1512 )/ 163  /,Ic( 1513 )/ 128  /
      Data Ic( 1514 )/ 4    /,Ic( 1515 )/ 542  /
      Data Ic( 1516 )/ 110  /,Ic( 1517 )/ 16   /
      Data Ic( 1518 )/ 519 /,Ic( 1519 )/ 44  /
      Data Ic( 1520 )/ 503 /,Ic( 1521 )/ 366 /
      Data Ic( 1522 )/ 464 /
C------<<100 D >>
      Data Ic( 1523 )/ 364 /,Ic( 1524 )/ 521 /
      Data Ic( 1525 )/ 163 /,Ic( 1526 )/ 213 /
      Data Ic( 1527 )/ 8   /,Ic( 1528 )/ 33  /
      Data Ic( 1529 )/ 157 /,Ic( 1530 )/ 449 /
      Data Ic( 1531 )/ 500 /,Ic( 1532 )/ 253 /
      Data Ic( 1533 )/ 57  /,Ic( 1534 )/ 171 /
      Data Ic( 1535 )/ 119 /,Ic( 1536 )/ 367 /
      Data Ic( 1537 )/ 472 /,Ic( 1538 )/ 368 /
      Data Ic( 1539 )/ 157 /,Ic( 1540 )/ 523 /
      Data Ic( 1541 )/ 369 /,Ic( 1542 )/ 5   /
      Data Ic( 1543 )/ 12  /,Ic( 1544 )/ 28  /
      Data Ic( 1545 )/ 51  /,Ic( 1546 )/ 157 /
C------<<101 E >>
      Data Ic( 1547 )/ 370 /,Ic( 1548 )/ 371 /
      Data Ic( 1549 )/ 4   /,Ic( 1550 )/ 59  /
      Data Ic( 1551 )/ 448 /,Ic( 1552 )/ 157 /
      Data Ic( 1553 )/ 433 /,Ic( 1554 )/ 431 /
      Data Ic( 1555 )/ 163 /,Ic( 1556 )/ 4   /
      Data Ic( 1557 )/ 372 /,Ic( 1558 )/ 1   /
      Data Ic( 1559 )/ 373 /,Ic( 1560 )/ 503 /
      Data Ic( 1561 )/ 366 /,Ic( 1562 )/ 219 /
C------<<102 F >>
      Data Ic( 1563 )/ 374 /,Ic( 1564 )/ 114 /
      Data Ic( 1565 )/ 9   /,Ic( 1566 )/ 43  /
      Data Ic( 1567 )/ 8   /,Ic( 1568 )/ 528 /
      Data Ic( 1569 )/ 112 /,Ic( 1570 )/ 495 /
      Data Ic( 1571 )/ 542 /,Ic( 1572 )/ 110 /
      Data Ic( 1573 )/ 16  /,Ic( 1574 )/ 444 /
      Data Ic( 1575 )/ 44  /,Ic( 1576 )/ 97  /
      Data Ic( 1577 )/ 452 /,Ic( 1578 )/ 375 /
      Data Ic( 1579 )/ 13  /,Ic( 1580 )/ 420 /
      Data Ic( 1581 )/ 184 /,Ic( 1582 )/ 422 /
      Data Ic( 1583 )/ 18  /,Ic( 1584 )/ 150 /
C------<<103 G >>
      Data Ic( 1585 )/ 504 /,Ic( 1586 )/ 505 /
      Data Ic( 1587 )/ 324 /,Ic( 1588 )/ 456 /
      Data Ic( 1589 )/ 3   /,Ic( 1590 )/ 35  /
      Data Ic( 1591 )/ 257 /,Ic( 1592 )/ 1   /
      Data Ic( 1593 )/ 377 /,Ic( 1594 )/ 5   /
```

```
      Data Ic( 1595 )/ 451 /,Ic( 1596 )/ 378 /
      Data Ic( 1597 )/ 522 /,Ic( 1598 )/ 158 /
      Data Ic( 1599 )/ 79  /,Ic( 1600 )/ 29  /
C------<<104 H >>
      Data Ic( 1601 )/ 506 /,Ic( 1602 )/ 134 /
      Data Ic( 1603 )/ 447 /,Ic( 1604 )/ 542 /
      Data Ic( 1605 )/ 157 /,Ic( 1606 )/ 15  /
      Data Ic( 1607 )/ 37  /,Ic( 1608 )/ 1   /
      Data Ic( 1609 )/ 224 /,Ic( 1610 )/ 9   /
      Data Ic( 1611 )/ 158 /,Ic( 1612 )/ 37  /
      Data Ic( 1613 )/ 1   /,Ic( 1614 )/ 315 /
      Data Ic( 1615 )/ 453 /,Ic( 1616 )/ 77  /
      Data Ic( 1617 )/ 34  /,Ic( 1618 )/ 28  /
      Data Ic( 1619 )/ 43  /,Ic( 1620 )/ 16  /
C------<<105 I >>
      Data Ic( 1621 )/ 267 /,Ic( 1622 )/ 529 /
      Data Ic( 1623 )/ 379 /,Ic( 1624 )/ 15  /
      Data Ic( 1625 )/ 16  /,Ic( 1626 )/ 507 /
      Data Ic( 1627 )/ 124 /,Ic( 1628 )/ 417 /
      Data Ic( 1629 )/ 1   /,Ic( 1630 )/ 215 /
      Data Ic( 1631 )/ 498 /,Ic( 1632 )/ 157 /
      Data Ic( 1633 )/ 13  /
C------<<106 J >>
      Data Ic( 1634 )/ 267 /,Ic( 1635 )/ 529 /
      Data Ic( 1636 )/ 380 /,Ic( 1637 )/ 30  /
      Data Ic( 1638 )/ 16  /,Ic( 1639 )/ 507 /
      Data Ic( 1640 )/ 157 /,Ic( 1641 )/ 78  /
      Data Ic( 1642 )/ 34  /,Ic( 1643 )/ 15  /
      Data Ic( 1644 )/ 110 /,Ic( 1645 )/ 45  /
      Data Ic( 1646 )/ 452 /,Ic( 1647 )/ 282 /
      Data Ic( 1648 )/ 498 /,Ic( 1649 )/ 157 /
      Data Ic( 1650 )/ 28  /
C------<<107 K >>
      Data Ic( 1651 )/ 506 /,Ic( 1652 )/ 213 /
      Data Ic( 1653 )/ 4   /,Ic( 1654 )/ 74  /
      Data Ic( 1655 )/ 119 /,Ic( 1656 )/ 195 /
      Data Ic( 1657 )/ 1   /,Ic( 1658 )/ 245 /
      Data Ic( 1659 )/ 53  /,Ic( 1660 )/ 1   /
      Data Ic( 1661 )/ 218 /,Ic( 1662 )/ 74  /
      Data Ic( 1663 )/ 1   /,Ic( 1664 )/ 520 /
      Data Ic( 1665 )/ 381 /,Ic( 1666 )/ 497 /
      Data Ic( 1667 )/ 382 /,Ic( 1668 )/ 419 /
      Data Ic( 1669 )/ 467 /,Ic( 1670 )/ 72  /
      Data Ic( 1671 )/ 13  /
C------<<108 L >>
      Data Ic( 1672 )/ 508 /
C------<<109 M >>
      Data Ic( 1673 )/ 18  /,Ic( 1674 )/ 541 /
      Data Ic( 1675 )/ 134 /,Ic( 1676 )/ 447 /
      Data Ic( 1677 )/ 53  /,Ic( 1678 )/ 157 /
      Data Ic( 1679 )/ 9   /,Ic( 1680 )/ 15  /
```

```
      Data Ic( 1681 )/ 521 /,Ic( 1682 )/ 12  /
      Data Ic( 1683 )/ 242 /,Ic( 1684 )/ 5   /
      Data Ic( 1685 )/ 451 /,Ic( 1686 )/ 4   /
      Data Ic( 1687 )/ 9   /,Ic( 1688 )/ 503 /
      Data Ic( 1689 )/ 345 /,Ic( 1690 )/ 509 /
C-----<<110 N >>
      Data Ic( 1691 )/ 347 /,Ic( 1692 )/ 541 /
      Data Ic( 1693 )/ 509 /
C-----<<111 O >>
      Data Ic( 1694 )/ 384 /,Ic( 1695 )/ 163 /
      Data Ic( 1696 )/ 4   /,Ic( 1697 )/ 435 /
      Data Ic( 1698 )/ 6   /,Ic( 1699 )/ 8   /
      Data Ic( 1700 )/ 33  /,Ic( 1701 )/ 110 /
      Data Ic( 1702 )/ 157 /,Ic( 1703 )/ 449 /
      Data Ic( 1704 )/ 134 /,Ic( 1705 )/ 420 /
      Data Ic( 1706 )/ 341 /,Ic( 1707 )/ 124 /
      Data Ic( 1708 )/ 523 /,Ic( 1709 )/ 385 /
      Data Ic( 1710 )/ 157 /,Ic( 1711 )/ 455 /
      Data Ic( 1712 )/ 1   /,Ic( 1713 )/ 386 /
      Data Ic( 1714 )/ 35  /,Ic( 1715 )/ 438 /
      Data Ic( 1716 )/ 387 /,Ic( 1717 )/ 522 /
      Data Ic( 1718 )/ 157 /
C-----<<112 P >>
      Data Ic( 1719 )/ 364 /,Ic( 1720 )/ 521 /
      Data Ic( 1721 )/ 163 /,Ic( 1722 )/ 521 /
      Data Ic( 1723 )/ 28  /,Ic( 1724 )/ 157 /
      Data Ic( 1725 )/ 8   /,Ic( 1726 )/ 18  /
      Data Ic( 1727 )/ 157 /,Ic( 1728 )/ 213 /
      Data Ic( 1729 )/ 30  /,Ic( 1730 )/ 192 /
      Data Ic( 1731 )/ 13  /,Ic( 1732 )/ 4   /
      Data Ic( 1733 )/ 500 /,Ic( 1734 )/ 450 /
      Data Ic( 1735 )/ 33  /,Ic( 1736 )/ 1   /
      Data Ic( 1737 )/ 16  /,Ic( 1738 )/ 13  /
      Data Ic( 1739 )/ 153 /,Ic( 1740 )/ 12  /
      Data Ic( 1741 )/ 30  /,Ic( 1742 )/ 1   /
      Data Ic( 1743 )/ 146 /,Ic( 1744 )/ 526 /
      Data Ic( 1745 )/ 1   /,Ic( 1746 )/ 517 /
      Data Ic( 1747 )/ 472 /,Ic( 1748 )/ 363 /
      Data Ic( 1749 )/ 453 /
C-----<<113 Q >>
      Data Ic( 1750 )/ 504 /,Ic( 1751 )/ 153 /
      Data Ic( 1752 )/ 218 /,Ic( 1753 )/ 54  /
      Data Ic( 1754 )/ 321 /,Ic( 1755 )/ 5   /
      Data Ic( 1756 )/ 419 /,Ic( 1757 )/ 168 /
      Data Ic( 1758 )/ 456 /,Ic( 1759 )/ 3   /
      Data Ic( 1760 )/ 35  /,Ic( 1761 )/ 97  /
      Data Ic( 1762 )/ 6   /,Ic( 1763 )/ 54  /
C-----<<114 R >>
      Data Ic( 1764 )/ 236 /,Ic( 1765 )/ 540 /
      Data Ic( 1766 )/ 443 /,Ic( 1767 )/ 444 /
      Data Ic( 1768 )/ 388 /,Ic( 1769 )/ 9   /
```

```
      Data Ic( 1770 )/ 503 /,Ic( 1771 )/ 166 /
      Data Ic( 1772 )/ 127 /,Ic( 1773 )/ 542 /
      Data Ic( 1774 )/ 110 /,Ic( 1775 )/ 16  /
      Data Ic( 1776 )/ 1   /,Ic( 1777 )/ 43  /
      Data Ic( 1778 )/ 464 /
C-----<<115 S >>
      Data Ic( 1779 )/ 63  /,Ic( 1780 )/ 352 /
      Data Ic( 1781 )/ 389 /,Ic( 1782 )/ 174 /
      Data Ic( 1783 )/ 54  /,Ic( 1784 )/ 7   /
      Data Ic( 1785 )/ 435 /,Ic( 1786 )/ 8   /
      Data Ic( 1787 )/ 33  /,Ic( 1788 )/ 71  /
      Data Ic( 1789 )/ 9   /,Ic( 1790 )/ 347 /
      Data Ic( 1791 )/ 9   /,Ic( 1792 )/ 71  /
      Data Ic( 1793 )/ 33  /,Ic( 1794 )/ 8   /
      Data Ic( 1795 )/ 481 /,Ic( 1796 )/ 8   /
      Data Ic( 1797 )/ 213 /,Ic( 1798 )/ 63  /
      Data Ic( 1799 )/ 9   /,Ic( 1800 )/ 12  /
      Data Ic( 1801 )/ 192 /,Ic( 1802 )/ 461 /
      Data Ic( 1803 )/ 390 /,Ic( 1804 )/ 445 /
      Data Ic( 1805 )/ 391 /,Ic( 1806 )/ 445 /
      Data Ic( 1807 )/ 390 /,Ic( 1808 )/ 5   /
      Data Ic( 1809 )/ 192 /,Ic( 1810 )/ 12  /
      Data Ic( 1811 )/ 9   /,Ic( 1812 )/ 436 /
      Data Ic( 1813 )/ 1   /,Ic( 1814 )/ 130 /
      Data Ic( 1815 )/ 420 /,Ic( 1816 )/ 257 /
      Data Ic( 1817 )/ 110 /
C-----<<116 T >>
      Data Ic( 1818 )/ 508 /,Ic( 1819 )/ 1   /
      Data Ic( 1820 )/ 392 /,Ic( 1821 )/ 422 /
      Data Ic( 1822 )/ 18  /,Ic( 1823 )/ 63  /
C-----<<117 U >>
      Data Ic( 1824 )/ 510 /,Ic( 1825 )/ 364 /
      Data Ic( 1826 )/ 44  /,Ic( 1827 )/ 6   /
      Data Ic( 1828 )/ 511 /,Ic( 1829 )/ 157 /
      Data Ic( 1830 )/ 53  /,Ic( 1831 )/ 530 /
      Data Ic( 1832 )/ 395 /,Ic( 1833 )/ 9   /
      Data Ic( 1834 )/ 158 /,Ic( 1835 )/ 13  /
C-----<<118 V >>
      Data Ic( 1836 )/ 393 /,Ic( 1837 )/ 518 /
      Data Ic( 1838 )/ 242 /,Ic( 1839 )/ 521 /
      Data Ic( 1840 )/ 28  /,Ic( 1841 )/ 157 /
      Data Ic( 1842 )/ 512 /,Ic( 1843 )/ 335 /
      Data Ic( 1844 )/ 13  /,Ic( 1845 )/ 157 /
      Data Ic( 1846 )/ 523 /,Ic( 1847 )/ 517 /
      Data Ic( 1848 )/ 124 /
C-----<<119 W >>
      Data Ic( 1849 )/ 396 /,Ic( 1850 )/ 33  /
      Data Ic( 1851 )/ 15  /,Ic( 1852 )/ 270 /
      Data Ic( 1853 )/ 163 /,Ic( 1854 )/ 30  /
      Data Ic( 1855 )/ 455 /,Ic( 1856 )/ 124 /
      Data Ic( 1857 )/ 522 /,Ic( 1858 )/ 1   /
```

```
      Data Ic( 1859 )/ 395 /,Ic( 1860 )/ 13  /
      Data Ic( 1861 )/ 157 /,Ic( 1862 )/ 523 /
      Data Ic( 1863 )/ 364 /,Ic( 1864 )/ 518 /
      Data Ic( 1865 )/ 177 /,Ic( 1866 )/ 12  /
      Data Ic( 1867 )/ 518 /,Ic( 1868 )/ 409 /
      Data Ic( 1869 )/ 124 /,Ic( 1870 )/ 512 /
      Data Ic( 1871 )/ 394 /,Ic( 1872 )/ 9   /
      Data Ic( 1873 )/ 157 /,Ic( 1874 )/ 523 /
      Data Ic( 1875 )/ 517 /,Ic( 1876 )/ 124 /
C-----<<120 X >>
      Data Ic( 1877 )/ 109 /,Ic( 1878 )/ 15  /
      Data Ic( 1879 )/ 518 /,Ic( 1880 )/ 397 /
      Data Ic( 1881 )/ 521 /,Ic( 1882 )/ 44  /
      Data Ic( 1883 )/ 103 /,Ic( 1884 )/ 455 /
      Data Ic( 1885 )/ 234 /,Ic( 1886 )/ 522 /
      Data Ic( 1887 )/ 43  /,Ic( 1888 )/ 1   /
      Data Ic( 1889 )/ 398 /,Ic( 1890 )/ 9   /
      Data Ic( 1891 )/ 399 /,Ic( 1892 )/ 452 /
      Data Ic( 1893 )/ 400 /,Ic( 1894 )/ 401 /
      Data Ic( 1895 )/ 13  /,Ic( 1896 )/ 421 /
      Data Ic( 1897 )/ 534 /,Ic( 1898 )/ 5   /
      Data Ic( 1899 )/ 1   /,Ic( 1900 )/ 12  /
      Data Ic( 1901 )/ 417 /,Ic( 1902 )/ 1   /
      Data Ic( 1903 )/ 47  /,Ic( 1904 )/ 402 /
      Data Ic( 1905 )/ 465 /,Ic( 1906 )/ 7   /
      Data Ic( 1907 )/ 54  /,Ic( 1908 )/ 4   /
      Data Ic( 1909 )/ 452 /,Ic( 1910 )/ 7   /
      Data Ic( 1911 )/ 6   /,Ic( 1912 )/ 451 /
      Data Ic( 1913 )/ 182 /,Ic( 1914 )/ 422 /
      Data Ic( 1915 )/ 63  /,Ic( 1916 )/ 63  /
C-----<<121 Y >>
      Data Ic( 1917 )/ 510 /,Ic( 1918 )/ 293 /
      Data Ic( 1919 )/ 102 /,Ic( 1920 )/ 171 /
      Data Ic( 1921 )/ 253 /,Ic( 1922 )/ 5   /
      Data Ic( 1923 )/ 103 /,Ic( 1924 )/ 542 /
      Data Ic( 1925 )/ 505 /,Ic( 1926 )/ 403 /
      Data Ic( 1927 )/ 9   /,Ic( 1928 )/ 120 /
      Data Ic( 1929 )/ 1   /,Ic( 1930 )/ 377 /
      Data Ic( 1931 )/ 5   /,Ic( 1932 )/ 44  /
C-----<<122 Z >>
      Data Ic( 1933 )/ 404 /,Ic( 1934 )/ 141 /
      Data Ic( 1935 )/ 32  /,Ic( 1936 )/ 5   /
      Data Ic( 1937 )/ 409 /,Ic( 1938 )/ 476 /
      Data Ic( 1939 )/ 127 /,Ic( 1940 )/ 405 /
      Data Ic( 1941 )/ 127 /,Ic( 1942 )/ 539 /
      Data Ic( 1943 )/ 9   /,Ic( 1944 )/ 15  /
      Data Ic( 1945 )/ 5   /,Ic( 1946 )/ 32  /
      Data Ic( 1947 )/ 141 /,Ic( 1948 )/ 1   /
      Data Ic( 1949 )/ 406 /,Ic( 1950 )/ 9   /
      Data Ic( 1951 )/ 474 /,Ic( 1952 )/ 407 /
      Data Ic( 1953 )/ 150 /,Ic( 1954 )/ 452 /
```

```
      Data Ic( 1955 )/ 175 /,Ic( 1956 )/ 194 /
C-----<<408   >>
      Data Ic( 1957 )/ 7   /,Ic( 1958 )/ 8   /
C-----<<409   >>
      Data Ic( 1959 )/ 15  /,Ic( 1960 )/ 9   /
C-----<<410   >>
      Data Ic( 1961 )/ 409 /,Ic( 1962 )/ 4   /
      Data Ic( 1963 )/ 16  /,Ic( 1964 )/ 1   /
      Data Ic( 1965 )/ 17  /,Ic( 1966 )/ 7   /
      Data Ic( 1967 )/ 18  /,Ic( 1968 )/ 6   /
C-----<<411   >>
      Data Ic( 1969 )/ 20  /,Ic( 1970 )/ 6   /
      Data Ic( 1971 )/ 7   /,Ic( 1972 )/ 21  /
      Data Ic( 1973 )/ 22  /
C-----<<412   >>
      Data Ic( 1974 )/ 9   /,Ic( 1975 )/ 28  /
C-----<<413   >>
      Data Ic( 1976 )/ 30  /,Ic( 1977 )/ 6   /
C-----<<414   >>
      Data Ic( 1978 )/ 34  /,Ic( 1979 )/ 37  /
C-----<<415   >>
      Data Ic( 1980 )/ 28  /,Ic( 1981 )/ 9   /
C-----<<416   >>
      Data Ic( 1982 )/ 37  /,Ic( 1983 )/ 29  /
C-----<<417   >>
      Data Ic( 1984 )/ 13  /,Ic( 1985 )/ 43  /
C-----<<418   >>
      Data Ic( 1986 )/ 37  /,Ic( 1987 )/ 7   /
C-----<<419   >>
      Data Ic( 1988 )/ 6   /,Ic( 1989 )/ 1   /
C-----<<420   >>
      Data Ic( 1990 )/ 18  /,Ic( 1991 )/ 1   /
C-----<<421   >>
      Data Ic( 1992 )/ 18  /,Ic( 1993 )/ 12  /
C-----<<422   >>
      Data Ic( 1994 )/ 63  /,Ic( 1995 )/ 1   /
C-----<<423   >>
      Data Ic( 1996 )/ 79  /,Ic( 1997 )/ 1   /
C-----<<424   >>
      Data Ic( 1998 )/ 37  /,Ic( 1999 )/ 76  /
C-----<<425   >>
      Data Ic( 2000 )/ 45  /,Ic( 2001 )/ 99  /
      Data Ic( 2002 )/ 17  /,Ic( 2003 )/ 100 /
C-----<<426   >>
      Data Ic( 2004 )/ 102 /,Ic( 2005 )/ 16  /
      Data Ic( 2006 )/ 103 /,Ic( 2007 )/ 28  /
      Data Ic( 2008 )/ 29  /,Ic( 2009 )/ 37  /
      Data Ic( 2010 )/ 519 /,Ic( 2011 )/ 20  /
      Data Ic( 2012 )/ 520 /,Ic( 2013 )/ 102 /
C-----<<427   >>
      Data Ic( 2014 )/ 8   /,Ic( 2015 )/ 521 /
```

```
      Data Ic( 2016 )/ 110 /,Ic( 2017 )/ 4   /
C-----<<428   >>
      Data Ic( 2018 )/ 18  /,Ic( 2019 )/ 113 /
      Data Ic( 2020 )/ 4   /
C-----<<429   >>
      Data Ic( 2021 )/ 14  /,Ic( 2022 )/ 523 /
      Data Ic( 2023 )/ 116 /,Ic( 2024 )/ 114 /
      Data Ic( 2025 )/ 9   /,Ic( 2026 )/ 524 /
      Data Ic( 2027 )/ 117 /
C-----<<430   >>
      Data Ic( 2028 )/ 522 /,Ic( 2029 )/ 525 /
      Data Ic( 2030 )/ 116 /,Ic( 2031 )/ 35  /
      Data Ic( 2032 )/ 14  /
C-----<<431   >>
      Data Ic( 2033 )/ 44  /,Ic( 2034 )/ 5   /
C-----<<432   >>
      Data Ic( 2035 )/ 13  /,Ic( 2036 )/ 28  /
C-----<<433   >>
      Data Ic( 2037 )/ 8   /,Ic( 2038 )/ 526 /
      Data Ic( 2039 )/ 13  /,Ic( 2040 )/ 12  /
      Data Ic( 2041 )/ 4   /
C-----<<434   >>
      Data Ic( 2042 )/ 1   /,Ic( 2043 )/ 121 /
C-----<<435   >>
      Data Ic( 2044 )/ 110 /,Ic( 2045 )/ 5   /
C-----<<436   >>
      Data Ic( 2046 )/ 18  /,Ic( 2047 )/ 4   /
C-----<<437   >>
      Data Ic( 2048 )/ 123 /,Ic( 2049 )/ 18  /
C-----<<438   >>
      Data Ic( 2050 )/ 124 /,Ic( 2051 )/ 1   /
C-----<<439   >>
      Data Ic( 2052 )/ 21  /,Ic( 2053 )/ 1   /
C-----<<440   >>
      Data Ic( 2054 )/ 137 /,Ic( 2055 )/ 1   /
C-----<<441   >>
      Data Ic( 2056 )/ 110 /,Ic( 2057 )/ 141 /
      Data Ic( 2058 )/ 427 /,Ic( 2059 )/ 527 /
      Data Ic( 2060 )/ 524 /
C-----<<442   >>
      Data Ic( 2061 )/ 9   /,Ic( 2062 )/ 43  /
C-----<<443   >>
      Data Ic( 2063 )/ 442 /,Ic( 2064 )/ 8   /
C-----<<444   >>
      Data Ic( 2065 )/ 7   /,Ic( 2066 )/ 1   /
C-----<<445   >>
      Data Ic( 2067 )/ 71  /,Ic( 2068 )/ 1   /
C-----<<446   >>
      Data Ic( 2069 )/ 153 /,Ic( 2070 )/ 9   /
C-----<<447   >>
      Data Ic( 2071 )/ 4   /,Ic( 2072 )/ 12  /
```

```
C------<<448   >>
       Data Ic( 2073 )/ 8   /,Ic( 2074 )/ 141 /
C------<<449   >>
       Data Ic( 2075 )/ 8   /,Ic( 2076 )/ 18  /
       Data Ic( 2077 )/ 9   /,Ic( 2078 )/ 13  /
       Data Ic( 2079 )/ 4   /
C------<<450   >>
       Data Ic( 2080 )/ 163 /,Ic( 2081 )/ 4   /
       Data Ic( 2082 )/ 431 /,Ic( 2083 )/ 6   /
       Data Ic( 2084 )/ 408 /
C------<<451   >>
       Data Ic( 2085 )/ 44  /,Ic( 2086 )/ 1   /
C------<<452   >>
       Data Ic( 2087 )/ 9   /,Ic( 2088 )/ 1   /
C------<<453   >>
       Data Ic( 2089 )/ 526 /,Ic( 2090 )/ 157 /
C------<<454   >>
       Data Ic( 2091 )/ 134 /,Ic( 2092 )/ 1   /
C------<<455   >>
       Data Ic( 2093 )/ 9   /,Ic( 2094 )/ 13  /
C------<<456   >>
       Data Ic( 2095 )/ 157 /,Ic( 2096 )/ 452 /
       Data Ic( 2097 )/ 166 /,Ic( 2098 )/ 157 /
       Data Ic( 2099 )/ 442 /,Ic( 2100 )/ 1   /
C------<<457   >>
       Data Ic( 2101 )/ 191 /,Ic( 2102 )/ 8   /
C------<<458   >>
       Data Ic( 2103 )/ 54  /,Ic( 2104 )/ 12  /
C------<<459   >>
       Data Ic( 2105 )/ 15  /,Ic( 2106 )/ 8   /
C------<<460   >>
       Data Ic( 2107 )/ 103 /,Ic( 2108 )/ 54  /
C------<<461   >>
       Data Ic( 2109 )/ 5   /,Ic( 2110 )/ 1   /
C------<<462   >>
       Data Ic( 2111 )/ 103 /,Ic( 2112 )/ 4   /
C------<<463   >>
       Data Ic( 2113 )/ 462 /,Ic( 2114 )/ 18  /
C------<<464   >>
       Data Ic( 2115 )/ 13  /,Ic( 2116 )/ 18  /
C------<<465   >>
       Data Ic( 2117 )/ 6   /,Ic( 2118 )/ 110 /
C------<<466   >>
       Data Ic( 2119 )/ 1   /,Ic( 2120 )/ 63  /
C------<<467   >>
       Data Ic( 2121 )/ 43  /,Ic( 2122 )/ 13  /
C------<<468   >>
       Data Ic( 2123 )/ 5   /,Ic( 2124 )/ 55  /
C------<<469   >>
       Data Ic( 2125 )/ 468 /,Ic( 2126 )/ 8   /
       Data Ic( 2127 )/ 15  /,Ic( 2128 )/ 539 /
```

```
      Data Ic( 2129 )/ 13  /,Ic( 2130 )/ 530 /
      Data Ic( 2131 )/ 201 /
C-----<<470   >>
      Data Ic( 2132 )/ 457 /,Ic( 2133 )/ 537 /
      Data Ic( 2134 )/ 103 /,Ic( 2135 )/ 1   /
      Data Ic( 2136 )/ 205 /,Ic( 2137 )/ 422 /
      Data Ic( 2138 )/ 206 /,Ic( 2139 )/ 532 /
C-----<<471   >>
      Data Ic( 2140 )/ 163 /,Ic( 2141 )/ 536 /
      Data Ic( 2142 )/ 207 /
C-----<<472   >>
      Data Ic( 2143 )/ 157 /,Ic( 2144 )/ 1   /
C-----<<473   >>
      Data Ic( 2145 )/ 8   /,Ic( 2146 )/ 15  /
C-----<<474   >>
      Data Ic( 2147 )/ 150 /,Ic( 2148 )/ 1   /
C-----<<475   >>
      Data Ic( 2149 )/ 474 /,Ic( 2150 )/ 195 /
      Data Ic( 2151 )/ 150 /
C-----<<476   >>
      Data Ic( 2152 )/ 63  /,Ic( 2153 )/ 134 /
C-----<<477   >>
      Data Ic( 2154 )/ 4   /,Ic( 2155 )/ 221 /
      Data Ic( 2156 )/ 9   /,Ic( 2157 )/ 421 /
      Data Ic( 2158 )/ 534 /,Ic( 2159 )/ 435 /
      Data Ic( 2160 )/ 1   /,Ic( 2161 )/ 15  /
C-----<<478   >>
      Data Ic( 2162 )/ 148 /,Ic( 2163 )/ 1   /
C-----<<479   >>
      Data Ic( 2164 )/ 146 /,Ic( 2165 )/ 538 /
      Data Ic( 2166 )/ 45  /,Ic( 2167 )/ 514 /
      Data Ic( 2168 )/ 59  /,Ic( 2169 )/ 422 /
C-----<<480   >>
      Data Ic( 2170 )/ 9   /,Ic( 2171 )/ 436 /
      Data Ic( 2172 )/ 1   /,Ic( 2173 )/ 47  /
      Data Ic( 2174 )/ 4   /,Ic( 2175 )/ 527 /
C-----<<481   >>
      Data Ic( 2176 )/ 5   /,Ic( 2177 )/ 110 /
C-----<<482   >>
      Data Ic( 2178 )/ 18  /,Ic( 2179 )/ 13  /
C-----<<483   >>
      Data Ic( 2180 )/ 169 /,Ic( 2181 )/ 545 /
      Data Ic( 2182 )/ 119 /,Ic( 2183 )/ 128 /
      Data Ic( 2184 )/ 236  /,Ic( 2185 )/ 123 /
      Data Ic( 2186 )/ 530  /,Ic( 2187 )/ 237 /
      Data Ic( 2188 )/134   /,Ic( 2189 )/ 236 /
      Data Ic( 2190 )/ 123  /
C-----<<484  >>
      Data Ic( 2191 )/ 79   /,Ic( 2192 )/ 28  /
      Data Ic( 2193 )/ 53   /
C-----<<485  >>
```

```
      Data Ic( 2194 )/ 470  /,Ic( 2195 )/ 252 /
      Data Ic( 2196 )/ 15   /,Ic( 2197 )/ 468 /
      Data Ic( 2198 )/ 141  /,Ic( 2199 )/ 213 /
      Data Ic( 2200 )/ 539  /,Ic( 2201 )/ 9   /
      Data Ic( 2202 )/ 253 /
C-----<<486  >>
      Data Ic( 2203 )/ 452 /,Ic( 2204 )/ 48  /
      Data Ic( 2205 )/ 471 /
C-----<<487  >>
      Data Ic( 2206 )/ 255 /,Ic( 2207 )/ 8   /
      Data Ic( 2208 )/ 31  /,Ic( 2209 )/ 55  /
      Data Ic( 2210 )/ 531 /,Ic( 2211 )/ 123 /
      Data Ic( 2212 )/ 63  /,Ic( 2213 )/ 530 /
      Data Ic( 2214 )/ 256 /,Ic( 2215 )/ 474 /
      Data Ic( 2216 )/ 239 /,Ic( 2217 )/ 479 /
C-----<<488   >>
      Data Ic( 2218 )/ 54  /,Ic( 2219 )/ 11  /
      Data Ic( 2220 )/ 173 /,Ic( 2221 )/ 103 /
      Data Ic( 2222 )/ 37  /,Ic( 2223 )/ 257 /
      Data Ic( 2224 )/ 37  /
C-----<<489   >>
      Data Ic( 2225 )/ 63  /,Ic( 2226 )/ 452 /
C-----<<490   >>
      Data Ic( 2227 )/ 49  /,Ic( 2228 )/ 273 /
      Data Ic( 2229 )/ 120 /,Ic( 2230 )/ 28  /
      Data Ic( 2231 )/ 43  /
C-----<<491   >>
      Data Ic( 2232 )/ 219 /,Ic( 2233 )/ 148 /
C-----<<492   >>
      Data Ic( 2234 )/ 276 /,Ic( 2235 )/ 43  /
      Data Ic( 2236 )/ 219 /,Ic( 2237 )/ 71  /
      Data Ic( 2238 )/ 55  /
C-----<<493   >>
      Data Ic( 2239 )/ 527 /,Ic( 2240 )/ 74  /
      Data Ic( 2241 )/ 278 /,Ic( 2242 )/ 74  /
C-----<<494   >>
      Data Ic( 2243 )/ 113 /,Ic( 2244 )/ 16  /
      Data Ic( 2245 )/ 30  /,Ic( 2246 )/ 171 /
      Data Ic( 2247 )/ 179 /,Ic( 2248 )/ 533 /
      Data Ic( 2249 )/ 103 /,Ic( 2250 )/ 473 /
      Data Ic( 2251 )/ 424 /,Ic( 2252 )/ 71  /
C-----<<495   >>
      Data Ic( 2253 )/ 134 /,Ic( 2254 )/ 4   /
C-----<<496   >>
      Data Ic( 2255 )/ 460 /,Ic( 2256 )/ 173 /
      Data Ic( 2257 )/ 192 /,Ic( 2258 )/ 171 /
      Data Ic( 2259 )/ 30  /,Ic( 2260 )/ 16  /
      Data Ic( 2261 )/ 5   /,Ic( 2262 )/ 7   /
      Data Ic( 2263 )/ 119 /,Ic( 2264 )/ 33  /
      Data Ic( 2265 )/ 120 /,Ic( 2266 )/ 12  /
      Data Ic( 2267 )/ 54  /,Ic( 2268 )/ 253 /
```

```
      Data Ic( 2269 )/ 1   /,Ic( 2270 )/ 125 /
      Data Ic( 2271 )/ 76  /,Ic( 2272 )/ 21  /
      Data Ic( 2273 )/ 423 /,Ic( 2274 )/ 288 /
      Data Ic( 2275 )/ 173 /,Ic( 2276 )/ 179 /
      Data Ic( 2277 )/ 289 /,Ic( 2278 )/ 419 /
      Data Ic( 2279 )/ 290 /,Ic( 2280 )/ 460 /
      Data Ic( 2281 )/ 173 /,Ic( 2282 )/ 179 /
      Data Ic( 2283 )/ 30  /,Ic( 2284 )/ 291 /
      Data Ic( 2285 )/ 5   /,Ic( 2286 )/ 292 /
      Data Ic( 2287 )/ 434 /,Ic( 2288 )/ 200 /
      Data Ic( 2289 )/ 1   /,Ic( 2290 )/ 186 /
      Data Ic( 2291 )/ 442 /,Ic( 2292 )/ 4   /
      Data Ic( 2293 )/ 1   /,Ic( 2294 )/ 50  /
      Data Ic( 2295 )/ 4   /,Ic( 2296 )/ 526 /
C-----<<497   >>
      Data Ic( 2297 )/ 417 /,Ic( 2298 )/ 15  /
      Data Ic( 2299 )/ 16  /
C-----<<498   >>
      Data Ic( 2300 )/ 12  /,Ic( 2301 )/ 6   /
      Data Ic( 2302 )/ 520 /
C-----<<499   >>
      Data Ic( 2303 )/ 325 /,Ic( 2304 )/ 31  /
      Data Ic( 2305 )/ 32  /,Ic( 2306 )/ 516 /
      Data Ic( 2307 )/ 9   /,Ic( 2308 )/ 213 /
C-----<<500   >>
      Data Ic( 2309 )/ 134 /,Ic( 2310 )/ 18  /
C-----<<501   >>
      Data Ic( 2311 )/ 28  /,Ic( 2312 )/ 478 /
      Data Ic( 2313 )/ 275 /,Ic( 2314 )/ 422 /
      Data Ic( 2315 )/ 327 /
C-----<<502   >>
      Data Ic( 2316 )/ 112 /,Ic( 2317 )/ 16  /
      Data Ic( 2318 )/ 119 /,Ic( 2319 )/ 482 /
      Data Ic( 2320 )/ 14  /
C-----<<503   >>
      Data Ic( 2321 )/ 158 /,Ic( 2322 )/ 452 /
C-----<<504   >>
      Data Ic( 2323 )/ 376 /,Ic( 2324 )/ 33  /
      Data Ic( 2325 )/ 8   /,Ic( 2326 )/ 528 /
      Data Ic( 2327 )/ 163 /,Ic( 2328 )/ 18  /
      Data Ic( 2329 )/ 495 /,Ic( 2330 )/ 535 /
C-----<<505   >>
      Data Ic( 2331 )/ 257 /,Ic( 2332 )/ 76  /
      Data Ic( 2333 )/ 15  /,Ic( 2334 )/ 427 /
      Data Ic( 2335 )/ 527 /,Ic( 2336 )/ 13  /
      Data Ic( 2337 )/ 1   /
C-----<<506   >>
      Data Ic( 2338 )/ 347 /,Ic( 2339 )/ 543 /
      Data Ic( 2340 )/ 446 /,Ic( 2341 )/ 15  /
      Data Ic( 2342 )/ 1   /,Ic( 2343 )/ 369 /
      Data Ic( 2344 )/ 30  /,Ic( 2345 )/ 12  /
```

```
      Data Ic( 2346 )/ 446 /,Ic( 2347 )/ 1   /
      Data Ic( 2348 )/ 345 /
C-----<<507   >>
      Data Ic( 2349 )/ 163 /,Ic( 2350 )/ 5   /
      Data Ic( 2351 )/ 447 /,Ic( 2352 )/ 522 /
      Data Ic( 2353 )/ 8   /
C-----<<508   >>
      Data Ic( 2354 )/ 310 /,Ic( 2355 )/ 543 /
      Data Ic( 2356 )/ 153 /,Ic( 2357 )/ 443 /
      Data Ic( 2358 )/ 444 /,Ic( 2359 )/ 369 /
      Data Ic( 2360 )/ 30  /,Ic( 2361 )/ 12  /
      Data Ic( 2362 )/ 446 /
C-----<<509   >>
      Data Ic( 2363 )/ 134 /,Ic( 2364 )/ 447 /
      Data Ic( 2365 )/ 542 /,Ic( 2366 )/ 124 /
      Data Ic( 2367 )/ 497 /,Ic( 2368 )/ 163 /
      Data Ic( 2369 )/ 5   /,Ic( 2370 )/ 451 /
      Data Ic( 2371 )/ 4   /,Ic( 2372 )/ 9   /
      Data Ic( 2373 )/ 158 /,Ic( 2374 )/ 13  /
C-----<<510   >>
      Data Ic( 2375 )/ 393 /,Ic( 2376 )/ 431 /
      Data Ic( 2377 )/ 163 /,Ic( 2378 )/ 16  /
      Data Ic( 2379 )/ 15  /,Ic( 2380 )/ 467 /
      Data Ic( 2381 )/ 124 /,Ic( 2382 )/ 433 /
      Data Ic( 2383 )/ 454 /,Ic( 2384 )/ 394 /
      Data Ic( 2385 )/ 9   /,Ic( 2386 )/ 503 /
C-----<<511   >>
      Data Ic( 2387 )/ 242 /,Ic( 2388 )/ 4   /
      Data Ic( 2389 )/ 431 /,Ic( 2390 )/ 6   /
      Data Ic( 2391 )/ 409 /
C-----<<512   >>
      Data Ic( 2392 )/ 219 /,Ic( 2393 )/ 103 /
      Data Ic( 2394 )/ 500 /,Ic( 2395 )/ 511 /
      Data Ic( 2396 )/ 438 /
C-----<<513   >>
      Data Ic( 2397 )/ 462 /,Ic( 2398 )/ 527 /
      Data Ic( 2399 )/ 15  /,Ic( 2400 )/ 16  /
      Data Ic( 2401 )/ 110 /,Ic( 2402 )/ 466 /
      Data Ic( 2403 )/ 467 /,Ic( 2404 )/ 1   /
C-----<<514   >>
      Data Ic( 2405 )/ 458 /,Ic( 2406 )/ 124 /
      Data Ic( 2407 )/ 418 /,Ic( 2408 )/ 1   /
      Data Ic( 2409 )/ 230 /,Ic( 2410 )/ 440 /
C-----<<515   >>
      Data Ic( 2411 )/ 530 /,Ic( 2412 )/ 205 /
      Data Ic( 2413 )/ 422 /
C-----<<516   >>
      Data Ic( 2414 )/ 321 /,Ic( 2415 )/ 413 /
      Data Ic( 2416 )/ 5   /,Ic( 2417 )/ 432 /
      Data Ic( 2418 )/ 21  /,Ic( 2419 )/ 110 /
      Data Ic( 2420 )/ 7   /,Ic( 2421 )/ 29  /
```

```
      Data Ic( 2422 )/ 12  /,Ic( 2423 )/ 18  /
      Data Ic( 2424 )/ 137 /,Ic( 2425 )/ 8   /
      Data Ic( 2426 )/ 63  /,Ic( 2427 )/ 123 /
C-----<<517   >>
      Data Ic( 2428 )/ 365 /,Ic( 2429 )/ 9   /
C-----<<518   >>
      Data Ic( 2430 )/ 5   /,Ic( 2431 )/ 6   /
C-----<<519   >>
      Data Ic( 2432 )/ 8   /,Ic( 2433 )/ 1   /
C-----<<520   >>
      Data Ic( 2434 )/ 7   /,Ic( 2435 )/ 13  /
C-----<<521   >>
      Data Ic( 2436 )/ 6   /,Ic( 2437 )/ 5   /
C-----<<522   >>
      Data Ic( 2438 )/ 13  /,Ic( 2439 )/ 9   /
C-----<<523   >>
      Data Ic( 2440 )/ 35  /,Ic( 2441 )/ 1   /
C-----<<524   >>
      Data Ic( 2442 )/ 13  /,Ic( 2443 )/ 1   /
C-----<<525   >>
      Data Ic( 2444 )/ 114 /,Ic( 2445 )/ 1   /
C-----<<526   >>
      Data Ic( 2446 )/ 43  /,Ic( 2447 )/ 9   /
C-----<<527   >>
      Data Ic( 2448 )/ 18  /,Ic( 2449 )/ 9   /
C-----<<528   >>
      Data Ic( 2450 )/ 7   /,Ic( 2451 )/ 6   /
      Data Ic( 2452 )/ 5   /,Ic( 2453 )/ 44  /
      Data Ic( 2454 )/ 4   /
C-----<<529   >>
      Data Ic( 2455 )/ 410 /,Ic( 2456 )/ 1   /
C-----<<530   >>
      Data Ic( 2457 )/ 103 /,Ic( 2458 )/ 1   /
C-----<<531   >>
      Data Ic( 2459 )/ 15  /,Ic( 2460 )/ 4   /
C-----<<532   >>
      Data Ic( 2461 )/ 179 /,Ic( 2462 )/ 538 /
      Data Ic( 2463 )/ 45  /,Ic( 2464 )/ 536 /
      Data Ic( 2465 )/ 208 /,Ic( 2466 )/ 422 /
C-----<<533   >>
      Data Ic( 2467 )/ 173 /,Ic( 2468 )/ 54  /
C-----<<534   >>
      Data Ic( 2469 )/ 37  /,Ic( 2470 )/ 8   /
C-----<<535   >>
      Data Ic( 2471 )/ 522 /,Ic( 2472 )/ 18  /
      Data Ic( 2473 )/ 8   /
C-----<<536   >>
      Data Ic( 2474 )/ 458 /,Ic( 2475 )/ 157 /
      Data Ic( 2476 )/ 418 /,Ic( 2477 )/ 1   /
      Data Ic( 2478 )/ 207 /,Ic( 2479 )/ 438 /
C-----<<537   >>
```

```
        Data Ic( 2480 )/ 468 /,Ic( 2481 )/ 531 /
        Data Ic( 2482 )/ 9   /,Ic( 2483 )/ 63  /
C------<<538   >>
        Data Ic( 2484 )/ 110 /,Ic( 2485 )/ 7   /
        Data Ic( 2486 )/ 102 /,Ic( 2487 )/ 12  /
        Data Ic( 2488 )/ 18  /
C------<<539   >>
        Data Ic( 2489 )/ 134 /,Ic( 2490 )/ 63  /
C------<<540   >>
        Data Ic( 2491 )/ 521 /,Ic( 2492 )/ 44  /
        Data Ic( 2493 )/ 4   /,Ic( 2494 )/ 242 /
        Data Ic( 2495 )/ 6   /,Ic( 2496 )/ 44  /
        Data Ic( 2497 )/ 103 /,Ic( 2498 )/ 53  /
        Data Ic( 2499 )/ 157 /
C------<<541   >>
        Data Ic( 2500 )/ 540 /,Ic( 2501 )/ 9   /
        Data Ic( 2502 )/ 15  /,Ic( 2503 )/ 1   /
        Data Ic( 2504 )/ 383 /,Ic( 2505 )/ 13  /
        Data Ic( 2506 )/ 503 /,Ic( 2507 )/ 345 /
C------<<542   >>
        Data Ic( 2508 )/ 522 /,Ic( 2509 )/ 43  /
        Data Ic( 2510 )/ 8   /
C------<<543   >>
        Data Ic( 2511 )/ 521 /,Ic( 2512 )/ 44  /
        Data Ic( 2513 )/ 4   /,Ic( 2514 )/ 321 /
        Data Ic( 2515 )/ 30  /,Ic( 2516 )/ 213 /
C------<<544   >>
        Data Ic( 2517 )/ 33  /,Ic( 2518 )/ 55  /
        Data Ic( 2519 )/ 31  /
C------<<545   >>
        Data Ic( 2520 )/ 130 /,Ic( 2521 )/ 141 /
C------<<546   >>
        Data Ic( 2522 )/ 134 /,Ic( 2523 )/ 150 /
        Data Ic( 2524 )/ 123 /,Ic( 2525 )/ 1   /
        Data Ic( 2526 )/ 0/
C
C     ********************************************************
C     *          CHARACTER INDEX TABLE                       *
C     ********************************************************
C
        Data In( 1  )/ 0    /,In( 2   )/ 15   /
        Data In( 3   )/ 20   /,In( 4   )/ 20   /
        Data In( 5   )/ 69   /,In( 6   )/ 69   /
        Data In( 7   )/ 118  /,In( 8   )/ 120  /
        Data In( 9   )/ 144  /,In( 10  )/ 170  /
        Data In( 11  )/ 195  /,In( 12  )/ 203  /
        Data In( 13  )/ 205  /,In( 14  )/ 207  /
        Data In( 15  )/ 209  /,In( 16  )/ 214  /
        Data In( 17  )/ 226  /,In( 18  )/ 240  /
        Data In( 19  )/ 273  /,In( 20  )/ 306  /
        Data In( 21  )/ 330  /,In( 22  )/ 354  /
```

```
Data In( 23  )/ 377  /,In( 24  )/ 407  /
Data In( 25  )/ 448  /,In( 26  )/ 465  /
Data In( 27  )/ 469  /,In( 28  )/ 473  /
Data In( 29  )/ 473  /,In( 30  )/ 478  /
Data In( 31  )/ 478  /,In( 32  )/ 507  /
Data In( 33  )/ 507  /,In( 34  )/ 546  /
Data In( 35  )/ 585  /,In( 36  )/ 623  /
Data In( 37  )/ 651  /,In( 38  )/ 686  /
Data In( 39  )/ 721  /,In( 40  )/ 769  /
Data In( 41  )/ 792  /,In( 42  )/ 809  /
Data In( 43  )/ 826  /,In( 44  )/ 853  /
Data In( 45  )/ 867  /,In( 46  )/ 921  /
Data In( 47  )/ 968  /,In( 48  )/ 974  /
Data In( 49  )/ 1017 /,In( 50  )/ 1039 /
Data In( 51  )/ 1078 /,In( 52  )/ 1150 /
Data In( 53  )/ 1203 /,In( 54  )/ 1259 /
Data In( 55  )/ 1279 /,In( 56  )/ 1317 /
Data In( 57  )/ 1358 /,In( 58  )/ 1393 /
Data In( 59  )/ 1436 /,In( 60  )/ 1448 /
Data In( 61  )/ 1450 /,In( 62  )/ 1450 /
Data In( 63  )/ 1450 /,In( 64  )/ 1450 /
Data In( 65  )/ 1450 /,In( 66  )/ 1487 /
Data In( 67  )/ 1507 /,In( 68  )/ 1522 /
Data In( 69  )/ 1546 /,In( 70  )/ 1562 /
Data In( 71  )/ 1584 /,In( 72  )/ 1600 /
Data In( 73  )/ 1620 /,In( 74  )/ 1633 /
Data In( 75  )/ 1650 /,In( 76  )/ 1671 /
Data In( 77  )/ 1672 /,In( 78  )/ 1690 /
Data In( 79  )/ 1693 /,In( 80  )/ 1718 /
Data In( 81  )/ 1749 /,In( 82  )/ 1763 /
Data In( 83  )/ 1778 /,In( 84  )/ 1817 /
Data In( 85  )/ 1823 /,In( 86  )/ 1835 /
Data In( 87  )/ 1848 /,In( 88  )/ 1876 /
Data In( 89  )/ 1916 /,In( 90  )/ 1932 /
Data In( 91  )/ 1956 /,In( 92  )/ 1958 /
Data In( 93  )/ 1960 /,In( 94  )/ 1968 /
Data In( 95  )/ 1973 /,In( 96  )/ 1975 /
Data In( 97  )/ 1977 /,In( 98  )/ 1979 /
Data In( 99  )/ 1981 /,In( 100 )/ 1983 /
Data In( 101 )/ 1985 /,In( 102 )/ 1987 /
Data In( 103 )/ 1989 /,In( 104 )/ 1991 /
Data In( 105 )/ 1993 /,In( 106 )/ 1995 /
Data In( 107 )/ 1997 /,In( 108 )/ 1999 /
Data In( 109 )/ 2003 /,In( 110 )/ 2013 /
Data In( 111 )/ 2017 /,In( 112 )/ 2020 /
Data In( 113 )/ 2027 /,In( 114 )/ 2032 /
Data In( 115 )/ 2034 /,In( 116 )/ 2036 /
Data In( 117 )/ 2041 /,In( 118 )/ 2043 /
Data In( 119 )/ 2045 /,In( 120 )/ 2047 /
Data In( 121 )/ 2049 /,In( 122 )/ 2051 /
Data In( 123 )/ 2053 /,In( 124 )/ 2055 /
```

```
Data In( 125 )/ 2060 /,In( 126 )/ 2062 /,
Data In( 127 )/ 2064 /,In( 128 )/ 2066 /
Data In( 129 )/ 2068 /,In( 130 )/ 2070 /
Data In( 131 )/ 2072 /,In( 132 )/ 2074 /
Data In( 133 )/ 2079 /,In( 134 )/ 2084 /
Data In( 135 )/ 2086 /,In( 136 )/ 2088 /
Data In( 137 )/ 2090 /,In( 138 )/ 2092 /
Data In( 139 )/ 2094 /,In( 140 )/ 2100 /
Data In( 141 )/ 2102 /,In( 142 )/ 2104 /
Data In( 143 )/ 2106 /,In( 144 )/ 2108 /
Data In( 145 )/ 2110 /,In( 146 )/ 2112 /
Data In( 147 )/ 2114 /,In( 148 )/ 2116 /
Data In( 149 )/ 2118 /,In( 150 )/ 2120 /
Data In( 151 )/ 2122 /,In( 152 )/ 2124 /
Data In( 153 )/ 2131 /,In( 154 )/ 2139 /
Data In( 155 )/ 2142 /,In( 156 )/ 2144 /
Data In( 157 )/ 2146 /,In( 158 )/ 2148 /
Data In( 159 )/ 2151 /,In( 160 )/ 2153 /
Data In( 161 )/ 2161 /,In( 162 )/ 2163 /
Data In( 163 )/ 2169 /,In( 164 )/ 2175 /
Data In( 165 )/ 2177 /,In( 166 )/ 2179 /
Data In( 167 )/ 2190 /,In( 168 )/ 2193 /
Data In( 169 )/ 2202 /,In( 170 )/ 2205 /
Data In( 171 )/ 2217 /,In( 172 )/ 2224 /
Data In( 173 )/ 2226 /,In( 174 )/ 2231 /
Data In( 175 )/ 2233 /,In( 176 )/ 2238 /
Data In( 177 )/ 2242 /,In( 178 )/ 2252 /
Data In( 179 )/ 2254 /,In( 180 )/ 2296 /
Data In( 181 )/ 2299 /,In( 182 )/ 2302 /
Data In( 183 )/ 2308 /,In( 184 )/ 2310 /
Data In( 185 )/ 2315 /,In( 186 )/ 2320 /
Data In( 187 )/ 2322 /,In( 188 )/ 2330 /
Data In( 189 )/ 2337 /,In( 190 )/ 2348 /
Data In( 191 )/ 2353 /,In( 192 )/ 2362 /
Data In( 193 )/ 2374 /,In( 194 )/ 2386 /
Data In( 195 )/ 2391 /,In( 196 )/ 2396 /
Data In( 197 )/ 2404 /,In( 198 )/ 2410 /
Data In( 199 )/ 2413 /,In( 200 )/ 2427 /
Data In( 201 )/ 2429 /,In( 202 )/ 2431 /
Data In( 203 )/ 2433 /,In( 204 )/ 2435 /
Data In( 205 )/ 2437 /,In( 206 )/ 2439 /
Data In( 207 )/ 2441 /,In( 208 )/ 2443 /
Data In( 209 )/ 2445 /,In( 210 )/ 2447 /
Data In( 211 )/ 2449 /,In( 212 )/ 2454 /
Data In( 213 )/ 2456 /,In( 214 )/ 2458 /
Data In( 215 )/ 2460 /,In( 216 )/ 2466 /
Data In( 217 )/ 2468 /,In( 218 )/ 2470 /
Data In( 219 )/ 2473 /,In( 220 )/ 2479 /
Data In( 221 )/ 2483 /,In( 222 )/ 2488 /
Data In( 223 )/ 2490 /,In( 224 )/ 2499 /
Data In( 225 )/ 2507 /,In( 226 )/ 2510 /
```

```
      Data In( 227 )/ 2516 /,In( 228 )/ 2519 /
      Data In( 229 )/ 2521 /,In( 230 )/ 2525 /
C
C
      Exit_Code = 0
      Icomp=408

      If (G_Char_Gen_State .Eq. 0) Then
            Ipen=3
            Iptr=0
      Else
            Ipen = 2
            Goto 16
      Endif

      Isym=Char
5     If ((Isym .Ge. 33) .And. (Isym .Le. 122)) Go To 10
      Isym=Isym-285
C
C OBTAIN INDEXES (BEGIN/END) TO THE CHARACTER TABLE
C
10    Ifirst=In(Isym-32)+1
      Ilast=In(Isym-32+1)
C
C LOOP THRU CHARACTER TABLE
C
15    I = Ifirst
16    If (I .Gt. Ilast) Goto 25
      J=Ic(I)
C
C IS THE CHARACTER TABLE ELEMENT A COMPOUND ?
C
      If (J .Ge. Icomp) Go To 30
C
C IS THE CHARACTER TABLE ELEMENT A PEN UP COMMAND ?
C
      If (J .Eq. 1) Go To 18
C
C OBTAIN THE X,Y BASIC ELEMENT PAIR FROM THE BASIC ELEMENTS TABLE
C
      Call Goth(J,Nx,Ny)
      Ym=Float(Ny)
      Xm=Float(Nx)
C
      Ypos = O_Char_Scale/21.0 * ( Ym*G_Cos_Crot + Xm*G_Sin_Crot )
      Xpos = O_Char_Aspect_Ratio*O_Char_Scale/21.0 *
     1       ( Xm*G_Cos_Crot - Ym*G_Sin_Crot )
C
      G_Char_Gen_State = Ipen
      I = I + 1
      Goto 45
```

```
18      Ipen = 3
        I = I + 1
        Goto 16
C
C ANY LOOP VARIABLES ON THE STACK ?
C
25      If (Iptr .Gt. 0) Go To 40

        G_Char_Gen_State = -1
        Go To 45
C
C DO WE NEED TO SAVE LOOP VARIABLES ?
C
30      If ( I .Eq. Ilast) Go To 35
        Index1=I+1
        Index2=Ilast
C
C PUT THE LOOP VARIABLES ON THE STACK
C
        Call Cgen_Push(Index1,Index2)
35      Isym=J
        Go To 5
C
C RETRIEVE THE LOOP VARIABLES FROM THE STACK
C
40      Call Cgen_Pop(Index1,Index2)
        Ifirst=Index1
        Ilast=Index2
        Go To 15
C
C RETURN INFO REGARDING THE FIRST MOVE ONLY
C
45      Exit_Code = G_Char_Gen_State

        Return
        End
```

```
      Logical *1 Function Ident_String(Toktyp,Sindex,Name_Table,Tablen,
     1       Index_Table,Exit_Code)

C Error system code = -200

C This routine takes a string token and identifies it as an abreviation of a
C string parameter value from a table of legal values for the current
C parameter.  We return TRUE if we encounter an error.

C Parameters:

C Toktyp - i [i] = The type of the current token.
C Sindex - i [o] = The index of the token in the table.
C Name_Table(Tablen) - c* [i] = The table of keyword names.
C Tablen - i [i] = The number of keyword names in the table.
C Index_Table(Tablen) - i [i] = A table of keyword numbers corresponding
C      to the keyword names.

      Implicit Integer (A-Z)

      Character *(*) Name_Table(Tablen)
      Integer Index_Table(Tablen)

      Include 'Lexan.Inc'

C Make sure we have a string token.

      If (Toktyp .Ne. L_Tt_String) Goto 800

C Do the actual identification.

      Call Cparse(L_Token,L_Tlen,Name_Table,Index_Table,Tablen,Sindex)

C Make sure we the token was in the table.

      If (Sindex .Lt. 0) Goto 801

C We're outa here...

      Ident_String = .False.
      Exit_Code = 0
      Goto 999

C String option required.

800   Ident_String = .True.
      Exit_Code = -200
      Goto 999

C Illegal / Ambiguous option given.
```

```
801     Ident_String = .True.
        If (Sindex .Eq. -1) Then
                Exit_Code = -201
        Else
                Exit_Code = -202
        Endif
        Goto 999

999     Return
        End
```

```
        Subroutine Init_Io(Exit_Code)

C   Error system code is 800

C   This routine initializes the printer I/O system.

        Implicit Integer (A-Z)

        Parameter ( Islen = 4 )
        Byte Init_String(Islen)

        Include 'Devices.Inc'
        Include 'Options.Inc'

C   If we are using the stream I/O system after initializing it we send an
C   "initialization string" to the printer.  This is necessary because the
C   Concept terminal which I work on would interpcept the escape sequences
C   meant for the printer as commands to itself.  Thus I issue a command to
C   change the Concept command character from an escape to a ctl-D. The command
C   is: <esc>o ^D .  On a different machine this could either be commented out
C   or changed to another command if a similar situation exists.

        Data Init_String/ 27, 111, 32, 4 /

C   Make sure stream I/O is available if we're asked not to use the default I/O
C   system (in which case Fortran is the default and there is no non-default
C   system).

        If (.Not. D_Stream_Io .And. .Not. O_Use_Default_Io_System) Goto 802

C   Either assign a Fortran logical unit to the printer or initialize stream
C   I/O.

        If (O_Rcl_Mode .And. (.Not. D_Stream_Io .Or.
     1      .Not. O_Use_Default_Io_System)) Then
            Open (Unit=D_Printer_Unit, Type='New',
     1              Recl=D_Max_Output_Line_Length, Err=800)
        Else
            If (.Not. D_Stream_Io) Goto 801

            Call Init_Io_Vms(Exit_Code)
            If (Exit_Code .Lt. 0) Goto 999

            If (D_Stream_Io) Call Toprinter_Vms(Init_String,Islen)
        Endif
        Goto 999

C   Error opening Fortran printer file.

800     Exit_Code = -800
        Goto 999
```

```
C  Attempt to use ESL without stream I/O capability.

801     Exit_Code = -801
        Goto 999

C  Default I/O system (Fortran) is the only one available.

802     Exit_Code = -802
        Goto 999

999     Return
        End
```

```
      Subroutine Init_Io_Vms(Exit_Code)

C  Error system code is 700

C  Machine dependent stream I/O routine.

C  This routine initializes the VMS stream I/O system.  The device name below
C  (TT:) must be assigned to a terminal device whose name must be given in the
C  assign command in the form: "_TTA0:".  Presumably "TT:" will be assigned to
C  the user's terminal by VMS.

      Implicit Integer (A-Z)

      Include 'Vmsio.Inc'

      Integer Sys$Assign

      Character *(*) Devnam
      Parameter ( Devnam = 'Tt' )

C  We use VMS no-wait queue I/O requests to send output to the printer. Before
C  we are able to issue QIOs we must associate an I/O channel number with a
C  device (terminal) through Sys$Assign.  When we're done we must free this
C  channel through a call to Sys$Dassgn.  However, since we're doing no-wait
C  QIOs we must have a way to tell when all our I/O is done when the program
C  is about to exit since the I/O requests will be canceled if the program is
C  no longer around.  This is done by using event flags, which are set when
C  I/O requests finish.  If we only used one constant event flag for all the
C  QIOs we did the flag might be set when an early QIO finished even though
C  there were more outstanding.  Thus we issue all the QIOs which send data to
C  the printer with event flag 1.  Then at the end of the program we issue a
C  zero-length QIO with event flag 2.  Since this QIO will not complete till
C  all the real ones issued before it do, we then need only wait for event
C  flag 2 to be set.

C  Clear the two event flags we'll be using.

      Call Sys$Clref(%Val(V_Event_Flag1))
      Call Sys$Clref(%Val(V_Event_Flag2))

C  Attempt to assign the channel to the device.

      If (Sys$Assign(Devnam,V_Channel,,) .Ne. V_Success) Then
            Exit_Code = -700
            Goto 999
      Endif

999   Return
      End
```

```
      Integer Function Iroundx(X)
C  Rounds off a real number to the nearest integer.
      Implicit Integer (A-Z)
      Real X
      If (X .Lt. 0.) Then
            Iroundx = X - .5
      Else
            Iroundx = X + .5
      Endif
999   Return
      End
```

```
      Subroutine Lexan(Expected_Ttyp,Toktyp,Exit_Code)

C  Error system code is -400

C  This is a lexical analyzer which divides a set of strings typed by the user
C  into a series of tokens for semantic analysis.

C  Params:

C  Expected_Ttyp - i [i] = The type of the token which the calling routine
C       expects (hopes) we will next encounter.
C  Toktyp - i [o] = The type of the token we actually did encounter.

C  BNF of a command string is:
C  <command-string>     -->   <command line> newline <command-string> | newline
C  <command-line>       -->     <option> | <option> , <command-line>
C  <option>             -->     <keyword> = <value>
C  <value>              -->     <string> | <integer> | <real> | <boolean>
C  <keyword>            -->     (a..z)+ (a..z_)*
C  <string>             -->     (a..z .[]:;$)+ (a..z -.[]:;$ 0..9)*
C  <integer>            -->     (0..9)+ | - (0..9)+
C  <real>               -->     (0..9)* . (0..9)* | - (0..9)* . (0..9)*
C  <boolean>            -->     YES | Y | NO | N


      Implicit Integer (A-Z)

      Integer * Lx_Chartype

      Include 'Lexan.Inc'
      Include 'Termunits.Inc'

      L_Tlen = 0

C  If the input buffer is empty prompt for a command line and read it in.

      If (L_Iptr .Eq. 0) Then
            Write (T_Out,901)
901         Format (' Enter A Command Line:')
            Read (T_In,900) L_Inplen, L_Inpline
900         Format (Q,A)
            L_Iptr = 1
      Else

C  Otherwise push the buffer pointer back one so we now encounter the
C  character which told us we had reached the end of the previous token.
C  We discover we're at the end of the token by realizing that we're
C  encountering a new type of token.

            L_Iptr = L_Iptr - 1
      Endif
```

```
C  Skip any spaces at the beginning of this token.

        Call Skip_Spaces

C  Branch to a segment of code determined by the token type we expect.

        Goto (100,200,300,400), Expected_Ttyp

C  The lexical analyzer is implimented by having a routine (Lx_Chartype)
C  which gets the next character in the input stream, and then determines
C  in which of several character classes (below) it exists.  We then branch
C  (change our state) depending on what kind of character was received.
C  In certain states the receipt of certain characters tell us that we are
C  at the end of the current token, so we exit.  Lx_Chartype puts all the
C  characters in the current token into the variable Token.

C  _________________   Keyword expected   _________________

C                  undrscor     dot       comma      equal     illegal
C            letter  | symbol  | digit   |  space   |  minus  |   EOL
C              |     |    |    |    |    |    |     |    |    |    |

100     Goto (110, 910, 911, 911, 911, 911, 100, 912, 911, 911, 500),
        1  Lx_Chartype()

110     Goto (110, 110, 911, 911, 911, 911, 120, 120, 911, 911, 120),
        1  Lx_Chartype()

120     Toktyp = L_Tt_Keywd
        Goto 990

C  _________________   Equal expected   _________________

200     Goto (920, 920, 920, 920, 920, 920, 200, 210, 920, 920, 920),
        1  Lx_Chartype()

C  This is a special type of token since it determines its own end.

210     L_Iptr = L_Iptr + 1
        L_Tlen = L_Tlen + 1
        Toktyp = L_Tt_Equal
        Goto 990

C  _________________   Option expected   _________________

300     Goto (310, 930, 310, 330, 350, 931, 300, 930, 350, 930, 931),
        1  Lx_Chartype()

310     Goto (310, 310, 310, 310, 310, 320, 320, 930, 310, 930, 320),
        1  Lx_Chartype()
```

```
320     Toktyp = L_Tt_String
        Goto 990

C  _________________   Number Expected   _________________

330     Goto (932, 932, 932, 933, 330, 340, 340, 932, 935, 932, 340),
        1  Lx_Chartype()

340     Toktyp = L_Tt_Real
        Goto 990

C  Real / Int.

350     Goto (934, 934, 934, 330, 350, 360, 360, 934, 935, 934, 360),
        1  Lx_Chartype()

360     Toktyp = L_Tt_Int
        Goto 990

C  _________________   Comma / EOL expected   _________________

400     Goto (940, 940, 940, 940, 940, 410, 400, 940, 940, 940, 420),
        1  Lx_Chartype()

410     L_Iptr = L_Iptr + 1
        L_Tlen = L_Tlen + 1
        Goto 430

C  EOL

420     L_Iptr = 0

430     Toktyp = L_Tt_Comma
        Goto 990

500     Toktyp = L_Tt_Eol
        Goto 990

C  Underscore begins keyword

910     Exit_Code = -400
        Goto 999

C  Illegal Character In Keyword

911     Exit_Code = -401
        Goto 999

C  Null Keyword
```

```
912     Exit_Code = -402
        Goto 999

C  Equals Sign Expected

920     Exit_Code = -403
        Goto 999

C  Illegal Character In Option

930     Exit_Code = -404
        Goto 999

C  Null Option

931     Exit_Code = -405
        Goto 999

C  Illegal Char In Real

932     Exit_Code = -406
        Goto 999

C  Second Decimal Point In Real

933     Exit_Code = -407
        Goto 999

C  Illegal Char In Number

934     Exit_Code = -408
        Goto 999

C  Minus sign must begin number

935     Exit_Code = -409
        Goto 999

C  Comma/Newline Expected

940     Exit_Code = -410
        Goto 999

C  Since we detect the end of a token with a character outside the current
C  token, the variable Token contains an extra character, so we decrease its
C  length count so that character doesn't show up.

990     L_Tlen = L_Tlen - 1
999     Return
        End
```

```
        Integer Function Lx_Chartype

C  Returns the type of the next character in the command string typed by the
C  user.

        Implicit Integer (A-Z)

        Include 'Lexan.Inc'

        Byte Char_Types(128)

C  A map from ascii value to character type.

        Data Char_Types / 32*L_Ct_Illeg, L_Ct_Space, 3*L_Ct_Illeg, L_Ct_Syms,
        1  7*L_Ct_Illeg, L_Ct_Comma, L_Ct_Minus, L_Ct_Dot, L_Ct_Illeg,
        2  10*L_Ct_Digit, 2*L_Ct_Syms, L_Ct_Illeg, L_Ct_Equal, 3*L_Ct_Illeg,
        3  26*L_Ct_Letter, L_Ct_Syms, L_Ct_Illeg, L_Ct_Syms, L_Ct_Illeg,
        4  L_Ct_Undscr, L_Ct_Illeg, 26*L_Ct_Letter, 5*L_Ct_Illeg /

C  If we're at the end of the buffer, return this type code.

        If (L_Iptr .Gt. L_Inplen) Goto 10

C  Get the type.

        Lx_Chartype = Char_Types(1 + Ichar(L_Inpline(L_Iptr:L_Iptr)))

C  Put this character into token.

        L_Tlen = L_Tlen + 1
        L_Token(L_Tlen:L_Tlen) = L_Inpline(L_Iptr:L_Iptr)

C  Increment input string pointer.

        L_Iptr = L_Iptr + 1
        Goto 999

C  We have reached the end of the user's string.

10      Lx_Chartype = L_Ct_Eol
        L_Iptr = L_Iptr + 1
        L_Tlen = L_Tlen + 1

999     Return
        End
```

```
      Subroutine Make_Band(S_Band_Top,S_Band_Bot,X_Ptos,Y_Ptos,Tymin,Tymax,
      1  White_Band,Exit_Code)

C  Fills a band with graphic elements which origininated in the Plot10
C  input file.

C  Parameters:

C  S_Band_Top, S_Band_Bot - i [i] = The Sanders coordinates in the raster of
C       the top and bottom of the current band.
C  X_Ptos, Y_Ptos - r [i] = Factors used to scale values from the Plot10 to
C       the Sanders coordinate system.
C  Tymin, Tymax - i [io] = The minimum and maximum coordinate values seen in
C       the Plot10 input file.
C  White_Band - L1 [o] = TRUE if no graphic elements mapped to this band.

      Implicit Integer (A-Z)

      Logical *1 White_Band

      Include 'Options.Inc'

      Parameter ( Alternating = 1, Multipass = 2 )

      White_Band = .True.

C  Retrieve the next vector from the Plot10 file using one of two algorithms.

10    If (O_Pass_Algorithm .Eq. Alternating)
      1  Call Next_Vec_In_Band_Alternating(S_Band_Top,S_Band_Bot,Sx1,Sy1,
      2  Sx2,Sy2,X_Ptos,Y_Ptos,Tymin,Tymax,Exit_Code)

      If (O_Pass_Algorithm .Eq. Multipass)
      1  Call Next_Vec_In_Band_Multipass(S_Band_Top,S_Band_Bot,Sx1,Sy1,
      2  Sx2,Sy2,X_Ptos,Y_Ptos,Tymin,Tymax,Exit_Code)

      If (Exit_Code) 999, 100, 200

100   Continue

C  Map the vector returned from Sanders to Band coordinates.

      By1 = Sy1 - S_Band_Bot
      By2 = Sy2 - S_Band_Bot

C  Rasterize the vector into the band.

      Call Vector_Gen_Asym(Sx1,By1,Sx2,By2)

C  Obviously the band isn't empty.
```

```
            White_Band = .False.

            Goto 10

C  No more vectors in Plot10 file.

200         Exit_Code = 0
            Goto 999

999         Return
            End
```

```
      Subroutine Make_Line(Outbuf,Obmax,Blen,Xpass,Ypass,X_Passes,Y_Passes,
     1  Band_Head_Pos,Next_Dot_Dis)

C  This routine takes a set of rows in the current band and maps them to a
C  string of characters which could be sent to the printer to form one
C  head-sweep on the page.  Note that these rows in the band need not be
C  contiguous since we must interlace with the print head at certain
C  resolutions.

C  Parameters:

C  Outbuf(Obmax) - b [o] = The string of characters we generate.
C  Blen - i [o] = The number of characters we've actually put in Outbuf.
C  Xpass, Ypass - i [i] = The passes we're on in the X and Y directions
C       within the current interlace group.  The first passes are numbered 0.
C  X_Passes, Y_Passes - i [i] = The total number of passes within the current
C       interlace group.
C  Band_Head_Pos - i [i] = The internal row at or below the highest page row
C       in the current interlace group.
C  Next_Dot_Dis - i [i] = The distance from the highest page row in the
C       current interlace group to internal row Band_Head_Pos.

C  You should understand the discussion in routine Output_Band before
C  attempting to understand this routine.

C  This routine contains three different algorithms for accomplishing its
C  task.  The first is totally general, and works for all the cases described
C  in Output_Band.  It's also pretty slow.  The second algorithm is more
C  specialized and faster.  It is used in cases when the vertical printing
C  resolution divides evenly into the print head pin seperation. The third
C  algorithm really cooks and is used for the case when the vertical print
C  resolution equals the print head pin seperation - i.e., there is a
C  one-to-one correspondence between pins and internal rows, and there is only
C  one vertical pass in each interlace group.  However, note that this
C  algorithm can only be used when the band is organized in column major
C  order, which puts points in contiguous columns in contiguous bits in the
C  buffer.    If the band is in row major order than we must use algorithm 2
C  for this case.

C  Outbuf will contain Blen characters, where Blen is the number of columns
C  in the current horizontal pass.  Each character represents a map of the
C  six pins in the print head.  Each of the six low bits in each character
C  specifies whether or not the corresponding pin (bit 0 = top pin) in the
C  print head should print or not (1 = print, 0 = don't print).  The
C  translation from these bit-maps to the actual character codes acceptable
C  as graphics instructions to the S700 is done by routine Output_Line.
C  If Blen is returned with a value of 0 it means the entire head sweep is
C  empty of printing dots.

C  Note that all the bit mapping done by this routine is the major bottleneck
C  in this damn program.
```

```
      Implicit Integer (A-Z)

      Byte Outbuf(Obmax)

      Include 'Band.Inc'
      Include 'Devices.Inc'
      Include 'Options.Inc'

      Integer Ihead
      Byte Bhead(4), Bhead1, Bhead2
      Equivalence (Ihead,Bhead)
      Equivalence (Bhead(1),Bhead1)
      Equivalence (Bhead(2),Bhead2)

      Integer Bit_Clear(6)

      Data Bit_Clear / 1, 2, 4, 8, 16, 32 /

C  Used for bit manipulations.

      Parameter ( Bytes_Per_Word = 2 )
      Byte Bdest(Bytes_Per_Word), Bsrc(Bytes_Per_Word)
      Integer Idest, Isrc
      Equivalence (Idest,Bdest)
      Equivalence (Isrc,Bsrc)

C  See if we've got either of the special cases.

      If (.Not. O_Row_Major .And.
     1  O_Vertical_Dot_Spacing .Eq. D_Print_Head_Pin_Spacing) Goto 200

      If (Mod(D_Print_Head_Pin_Spacing,O_Vertical_Dot_Spacing) .Eq. 0)
     1  Goto 100

C  ******    General Case

C  Bhp and Ndd will now contain their respective values for each pin in the
C  head.  First we bump them up to be accurate for the top pin given which
C  pass we are on.

C  The algorithm is basically to loop through each pin position in the current
C  head sweep, determine if it maps into the band, and then determine if the
C  bit in Outbuf representing that pin should be set or not (according to the
C  band bitmap).

      Bhp = Band_Head_Pos
      Ndd = Next_Dot_Dis
      Call Track(Ndd,Ypass*D_Print_Head_Pin_Spacing/Y_Passes,Bhp)

      First = .True.
```

```
        Exit_Blen = 0

C  For each pin...

        Do 20 Y = 0, D_Print_Head_Pins-1

        Blen = 0

C  Check for the possibility that we're in the last interlace group and that
C  our page position is below the end of the internal band.  Exit_Blen is
C  used to maintain the number of chars in Outbuf should we have to abort the
C  algorithm early here.  Otherwise Blen would be zero as set above and the
C  calling routine would think there were no dots mapped in this sweep.

        If (Bhp .Gt. B_Band_Height-1) Then
                Blen = Exit_Blen
                Goto 999
        Endif

C  Does this pin map to a band row?  If not, ignore it.

        If (Ndd .Eq. 0) Then

                If (O_Row_Major) Then
                        Bit_Numb = Bhp * B_Band_Width + Xpass
                        Bit_Bump = X_Passes
                Else
                        Bit_Numb = Bhp + Xpass * B_Band_Height
                        Bit_Bump = X_Passes * B_Band_Height
                Endif

C  Loop through each column in the current horizontal pass.

                Do 10 X = Xpass, B_Band_Width-1, X_Passes

                Blen = Blen + 1

C  If we're on the top pin we zero out this entire column.  Otherwise we
C  get the byte representing the current column, which we're in the process
C  of building.

                If (First) Then
                        Bdest(1) = 0
                Else
                        Bdest(1) = Outbuf(Blen)
                Endif

C  Compute the bit number in B_Band of the current bit we're on in the
C  band (row = Bhp, col = X).  Put the byte containing that bit into
C  Bsrc(1), and the move the bit of interest onto the current pin through
C  Idest/Bdest.
```

```
                Byte_Numb = Bit_Numb / D_Bits_Per_Byte
                Bsrc(1) = B_Band(Byte_Numb+1)

                Src_Bit = Bit_Numb - Byte_Numb * D_Bits_Per_Byte
                Bit_Shift = Y - Src_Bit
                Isrc = Ishft(Isrc,Bit_Shift)
                Isrc = Isrc .And. Bit_Clear(Y+1)
                Idest = Idest .Or. Isrc

                Outbuf(Blen) = Bdest(1)

                Bit_Numb = Bit_Numb + Bit_Bump

10              Continue

                First = .False.
                Exit_Blen = Blen

        Endif

C  Track the values of Ndd and Bhp to the next pin.

        Call Track(Ndd,D_Print_Head_Pin_Spacing,Bhp)

20      Continue

        Blen = Exit_Blen
        Goto 999

100     Continue

C  ******  Medium optimization algorithm.

C  In this version we don't have to keep checking to see if pins are mapped;
C  we know they're all mapped.  So we cut out a lot of unnecessary checking
C  and just shove the bits around.

C  Offset Bhp for this pass.

        Bhp = Band_Head_Pos + Ypass
        Blen = 0
        If (O_Row_Major) Then
                Bit_Bump = Y_Passes * B_Band_Width
        Else
                Bit_Bump = Y_Passes
        Endif

C  Loop by column.

        Do 110 X = Xpass, B_Band_Width-1, X_Passes
        -
```

```
C  Compute the bit number of this top pin, and the number of pins that are
C  in use.  We may be at the end of the band and the interlace group may
C  extend off the end of the band.

        If (O_Row_Major) Then
                Bit_Numb = Bhp * B_Band_Width + X
        Else
                Bit_Numb = X * B_Band_Height + Bhp
        Endif

        Idest = 0
        Last_Pin = Min((D_Print_Head_Pins-1),((B_Band_Height-1 - Bhp)/
     1          Y_Passes))

C  Loop by pin.  Move bits from the band to Outbuf.

        Do 120 Y = 0, Last_Pin

        Byte_Numb = Bit_Numb / D_Bits_Per_Byte
        Bsrc(1) = B_Band(Byte_Numb+1)

        Src_Bit = Bit_Numb - Byte_Numb * D_Bits_Per_Byte
        Bit_Shift = Y - Src_Bit
        Isrc = Ishft(Isrc,Bit_Shift)
        Isrc = Isrc .And. Bit_Clear(Y+1)
        Idest = Idest .Or. Isrc

C  Compute the index of the next bit.

        Bit_Numb = Bit_Numb + Bit_Bump

120     Continue

C  Store this bit map and go to the one for the next column.

        Blen = Blen + 1
        Outbuf(Blen) = Bdest(1)

110     Continue
        Goto 999

200     Continue

C  ****  Fastest optimization algorithm.

C  This algorithm derives its speed from the fact that only one vertical pass
C  is needed per interlace group.  We can take advantage of the fact that
C  bits which are vertically adjacent in the band are also adjacent in the
C  print head, so we can move the bits in groups of six rather than one at a
C  time.  This saves a lot of time.
```

```
C  Head_Mask is used to clear bits 6-31 in Ihead, and the bits for any pins
C  which fall past the end of the band because they're in the last interlace
C  group.

        Bhp = Band_Head_Pos + Ypass
        Blen = 0

        Bit_Numb = Bhp
        Bit_Bump = X_Passes * B_Band_Height
        Pins_Used = Min((D_Print_Head_Pins),(B_Band_Height-1 - Bhp + 1))
        Head_Mask = 2 ** Pins_Used - 1

C  Loop by column and extract a column of 6 bits from the band.  A group
C  of 6 bits could easily span a byte boundary, so we have to use the set
C  of two bytes which must contain the entire group of 6 bits to shift.

        Do 210 X = Xpass, B_Band_Width-1, X_Passes

        Byte_Address = Bit_Numb / D_Bits_Per_Byte
        Bit_Address = Bit_Numb - Byte_Address * D_Bits_Per_Byte

        Bhead1 = B_Band(Byte_Address+1)
        Bhead2 = B_Band(Byte_Address+2)

        Ihead = Ishft(Ihead,-Bit_Address)
        Ihead = Ihead .And. Head_Mask

        Blen = Blen + 1
        Outbuf(Blen) = Ihead

        Bit_Numb = Bit_Numb + Bit_Bump

210     Continue

999     Return
        End
```

```
      Subroutine Make_Line_Vms(Outbuf,Obmax,Blen,Xpass,Ypass,X_Passes,
      1  Y_Passes,Band_Head_Pos,Next_Dot_Dis)

C  Machine dependent routine.

C  This routine takes a set of rows in the current band and maps them to a
C  string of characters which could be sent to the printer to form one
C  head-sweep on the page.  Note that these rows in the band need not be
C  contiguous since we must interlace with the print head at certain
C  resolutions.

C  Parameters:

C  Outbuf(Obmax) - b [o] = The string of characters we generate.
C  Blen - i [o] = The number of characters we've actually put in Outbuf.
C  Xpass, Ypass - i [i] = The passes we're on in the X and Y directions
C       within the current interlace group.  The first passes are numbered 0.
C  X_Passes, Y_Passes - i [i] = The total number of passes within the current
C       interlace group.
C  Band_Head_Pos - i [i] = The internal row at or below the highest page row
C       in the current interlace group.
C  Next_Dot_Dis - i [i] = The distance from the highest page row in the
C       current interlace group to internal row Band_Head_Pos.

C  You should understand the discussion in routine Output_Band before
C  attempting to understand this routine.

C  This machine dependent version of Make_Line contains three different
C  algorithms for accomplishing its task.  The first is totally general, and
C  works for all the cases described in Output_Band.  It's also pretty slow.
C  The second algorithm is more specialized and faster.  It is used in cases
C  when the vertical printing resolution divides evenly into the print head
C  pin seperation.  The third algorithm really cooks and is used for the case
C  when the vertical print resolution equals the print head pin seperation -
C  i.e., there is a one-to-one correspondence between pins and internal rows,
C  and there is only one vertical pass in each interlace group.  However, note
C  that this algorithm can only be used when the raster is organized in column
C  major order, which puts points in contiguous columns in contiguous bits in
C  the buffer.    If the raster is in row major order than we must use
C  algorithm 2 for this case.

C  Outbuf will contain Blen characters, where Blen is the number of columns
C  in the current horizontal pass.  Each character represents a map of the
C  six pins in the print head.  Each of the six low bits in each character
C  specifies whether or not the corresponding pin (bit 0 = top pin) in the
C  print head should print or not (1 = print, 0 = don't print).  The
C  translation from these bit-maps to the actual character codes acceptable
C  as graphics instructions to the S700 is done by routine Output_Line.
C  If Blen is returned with a value of 0 it means the entire head sweep is
C  empty of printing dots.
```

```
C  Note that all the bit mapping done by this routine and its slower
C  machine-independent sibling is the major bottleneck in this damn program.

       Implicit Integer (A-Z)

       Byte Outbuf(Obmax)

C  Used for bit manipulations.

       Parameter ( Bytes_Per_Word = 2 )
       Byte Bdest(Bytes_Per_Word), Bsrc(Bytes_Per_Word)
       Integer Idest, Isrc
       Equivalence (Idest,Bdest)
       Equivalence (Isrc,Bsrc)

       Logical *1 First

       Byte Head_Image

       Include 'Band.Inc'
       Include 'Devices.Inc'
       Include 'Options.Inc'

C  See if we've got either of the special cases.

       If (.Not. O_Row_Major .And.
      1  O_Vertical_Dot_Spacing .Eq. D_Print_Head_Pin_Spacing) Goto 200

       If (Mod(D_Print_Head_Pin_Spacing,O_Vertical_Dot_Spacing) .Eq. 0)
      1  Goto 100

C  ******    General Case

C  Bhp and Ndd will now contain their respective values for each pin in the
C  head.  First we bump them up to be accurate for the top pin given which
C  pass we are on.

C  The algorithm is basically to loop through each pin position in the current
C  head sweep, determine if it maps into the band, and then determine if the
C  bit in Outbuf representing that pin should be set or not (according to the
C  band bitmap).

       Bhp = Band_Head_Pos
       Ndd = Next_Dot_Dis
       Call Track(Ndd,Ypass*D_Print_Head_Pin_Spacing/Y_Passes,Bhp)

       First = .True.
       Exit_Blen = 0

C  For each pin...
```

```
        Do 20 Y = 0, D_Print_Head_Pins-1

        Blen = 0

C  Check for the possibility that we're in the last interlace group and that
C  our page position is below the end of the internal band.  Exit_Blen is
C  used to maintain the number of chars in Outbuf should we have to abort the
C  algorithm early here.  Otherwise Blen would be zero as set above and the
C  calling routine would think there were no dots mapped in this sweep.

        If (Bhp .Gt. B_Band_Height-1) Then
                Blen = Exit_Blen
                Goto 999
        Endif

C  Does this pin map to a band row?  If not, ignore it.

        If (Ndd .Eq. 0) Then

                If (O_Row_Major) Then
                        Bit_Numb = Bhp * B_Band_Width + Xpass
                        Bit_Bump = X_Passes
                Else
                        Bit_Numb = Bhp + Xpass * B_Band_Height
                        Bit_Bump = X_Passes * B_Band_Height
                Endif

C  Loop through each column in the current horizontal pass.

                Do 10 X = Xpass, B_Band_Width-1, X_Passes

                Blen = Blen + 1

C  If we're on the top pin we zero out this entire column.  Otherwise we
C  get the byte representing the current column, which we're in the process
C  of building.

                If (First) Outbuf(Blen) = 0

C  Compute the bit number in B_Band of the current bit we're on in the
C  band (row = Bhp, col = X).  Put the byte containing that bit into
C  Bsrc(1), and the move the bit of interest onto the current pin through
C  Idest/Bdest.

                Idest = Lib$Extzv(Bit_Numb,1,B_Band)
                Idest = Ishft(Idest,Y)

                Outbuf(Blen) = Outbuf(Blen) .Or. Bdest(1)

                Bit_Numb = Bit_Numb + Bit_Bump
```

```
10              Continue

                First = .False.
                Exit_Blen = Blen

        Endif

C  Track the values of Ndd and Bhp to the next pin.

        Call Track(Ndd,D_Print_Head_Pin_Spacing,Bhp)

20      Continue

        Blen = Exit_Blen
        Goto 999

100     Continue

C  ******  Medium optimization algorithm.

C  In this version we don't have to keep checking to see if pins are mapped;
C  we know they're all mapped.  So we cut out a lot of unnecessary checking
C  and just shove the bits around.

C  Offset Bhp for this pass.

        Bhp = Band_Head_Pos + Ypass
        Blen = 0
        If (O_Row_Major) Then
                Bit_Bump = Y_Passes * B_Band_Width
        Else
                Bit_Bump = Y_Passes
        Endif

C  Loop by column.

        Do 110 X = Xpass, B_Band_Width-1, X_Passes

C  Compute the bit number of this top pin, and the number of pins that are
C  in use.  We may be at the end of the band and the interlace group may
C  extend off the end of the band.

        If (O_Row_Major) Then
                Bit_Numb = Bhp * B_Band_Width + X
        Else
                Bit_Numb = X * B_Band_Height + Bhp
        Endif

        Idest = 0
        Last_Pin = Min((D_Print_Head_Pins-1),((B_Band_Height-1 - Bhp)/
     1          Y_Passes))
```

```
        Head_Image = 0

C  Loop by pin.  Move bits from the band to Outbuf.

        Do 120 Y = 0, Last_Pin

        Idest = Lib$Extzv(Bit_Numb,1,B_Band)
        Idest = Ishft(Idest,Y)

        Head_Image = Head_Image .Or. Bdest(1)

C  Compute the index of the next bit.

        Bit_Numb = Bit_Numb + Bit_Bump

120     Continue

C  Store this bit map and go to the one for the next column.

        Blen = Blen + 1
        Outbuf(Blen) = Head_Image

110     Continue
        Goto 999

200     Continue

C  ****  Fastest optimization algorithm.

C  This algorithm derives its speed from the fact that only one vertical pass
C  is needed per interlace group.  We can take advantage of the fact that bits
C  which are vertically adjacent in the band are also adjacent in the print
C  head, so we can move the bits in groups of six rather than one at a time.
C  This saves a lot of time.  We have to use Lib$Extzv rather than the VMS
C  Fortran function Mvbits here.  The difference between them is that the
C  former extracts a bit string from the middle of a huge buffer given the bit
C  index of the start of the string.  Mvbits will only extract from an
C  integer. Mvbits can't be used here because the group of six bits could span
C  any type of storage boundary (byte, word, double-word, etc).  At first I
C  was using Mvbits in the algorithms above, but it turned out to be faster to
C  substitute Lib$Extzv there too.

        Bhp = Band_Head_Pos + Ypass
        Blen = 0

C  We compute a starting bit address and the constant by which the bit address
C  changes at each new column.  We also watch for the end of the band on the
C  last interlace group.

        Bit_Numb = Bhp
```

```
        Bit_Bump = X_Passes * B_Band_Height
        Pins_Used = Min((D_Print_Head_Pins),(B_Band_Height-1 - Bhp + 1))

C  Loop by column and extract a column of 6 bits from the band.

        Do 210 X = Xpass, B_Band_Width-1, X_Passes

        Blen = Blen + 1
        Outbuf(Blen) = Lib$Extzv(Bit_Numb,Pins_Used,B_Band)

        Bit_Numb = Bit_Numb + Bit_Bump

210     Continue

999     Return
        End
```

```
      Subroutine Make_Space_String(Heps,String,Slen)

C  Creates a string of characters which, when printed in graphics mode, will
C  generated Heps heps of horizontal white space.

C  Parameters:

C  Heps - i [i] = The amount of white space needed.
C  String - c** [o] = The string we create.  We should be given at least 20
C       chars of space.
C  Slen - i [o] = The number of characters we returned in the string.

C  The characters in Space_Chars generate different amounts of space when
C  printed in graphics mode.  The amounts are the elemens of the geometric
C  sequence: 1, 2, 4, 8, ....   Thus we just check the bits in Heps to
C  determine which characters to put in the string.  See the document
C  Sanders Techology Product Bulletin - Media 12/7 Typographic Printer
C  Graphics Option.

      Implicit Integer (A-Z)

      Character *(*) String

      Character *12 Space_Chars
      Character *1 Char

      Data Space_Chars/ 'Tuvwxyz{|}~ ' /

C  Last char should be a rubout

      Space_Chars(12:12) = Char(127)

C  The most powerful spacer can only generate 2**11 heps of space, but the
C  calling routine might request more than this, so we have to generate
C  enough multiple occurences of this character to supply what he wants.

      Twolls = Heps / 2**11
      If (Twolls .Eq. 0) Goto 15

      Do 10 Ndx = 1, Twolls
10    String(Ndx:Ndx) = Space_Chars(12:12)

15    Continue

      Slen = Twolls

C  Now check the bits in Heps and insert the corresponding character when we
C  find a bit is set.

      Do 20 Bitpos = 0, 10
```

```
            If (Btest(Heps,Bitpos)) Then
                  Slen = Slen + 1
                  String(Slen:Slen) = Space_Chars(Bitpos+1:Bitpos+1)
            Endif

20          Continue

999         Return
            End
```

```
      Subroutine Merge(Coord,Newbyte,Hi)

C  Plot10 handles transmission of coordinate data in the following way.
C  Registers in the display hardware and the Plot10 system on the host both
C  keep track of a "current position" in the 10 bit Plot10 coordinate space.
C  When Plot10 wishes to transmit a new coordinate for a move or a draw, it
C  compares the new coordinate against the stored current position and
C  determines which or both of the high and low five bits in the current
C  position change in the transition to the new position.  Given restricitions
C  about ordering of data for HiX, LoX, HiY and LoY, Plot10 then transmits
C  characters representing the new values for the fields that have changed.
C  The idea is that the new coordinate will be very similar to the old one,
C  and on the average they won't have to transmit all 20 bits of data each
C  time.  This routine merges a new transmitted data coordinate with an
C  existing binary coordinate. The new coordinate consists of 5 bits, which
C  will replace either the high or the low bits of the existing 10 bit
C  coordinate.  See 4010/4010-1 Maintenance Manual, pg 2-11 for details.

C  Params:

C  COORD - I [IO] = The existing coordinate.
C  NEWBYTE - B [I] = The new coordinate.
C  HI - L1 [I] = TRUE if the new coordinate should replace the high bits of
C                the existing coord.

      Implicit Integer (A-Z)

      Byte Newbyte
      Logical *1 Hi

      Include 'Shifts.Inc'

CCCCCCCC      Data Lo5/'1f'X/, Hi5/'3e0'X/
      Data Lo5/ 31 /, Hi5/ 992 /

C  The bits above 5 in Newbyte contain information we're not concerned with
C  here.  Clear them.

      Newdata = Newbyte
      Newdata = Bic(Newdata,.Not. Lo5)

C  Allign the data bits in Newdata with their destination in Coord, and zero
C  the bit field we are changing in Coord.

      If (Hi) Then
              Newdata = Newdata * Shift5
              Coord = Bic(Coord,Hi5)
      Else
              Coord = Bic(Coord,Lo5)
      Endif
```

```
C  Use Newdata as a mask to set bits in Coord.

        Coord = Bis(Coord,Newdata)

        Return
        End
```

```
      Integer Function Mysign(Int)
C  Returns: -1 if Int<0; 0 if Int=0; 1 if Int>0
      Implicit Integer (A-Z)
      Mysign = 0
      If (Int .Ne. 0) Mysign = Int / Iabs(Int)
      Return
      End
```

```
        Logical *1 Function Next_Char

C  Places the next character in the P1OINTERP input stream into the common
C  variable p_curr_char.  Returns TRUE if an end-of-file is encountered on
C  the input stream.

        Implicit Integer (A-Z)

        Include 'P1Ointerp.Inc'

C  Is the current buffer exhausted ?

10      If (P_Blen .Eq. P_Bptr) Goto 20

C  No; get the next char.

        P_Bptr = P_Bptr + 1
        P_Curr_Char = P_Buffer(P_Bptr)
        Next_Char = .False.
        Goto 999

C  Read a new buffer.

20      Read (P_Input_Unit,900,End=200) P_Blen, (P_Buffer(N),N=1,P_Blen)
900     Format (Q,<P_Blen>A1)

C  Keep track of where we are in the file for error messages.

        P_Reccnt = P_Reccnt + 1
        P_Bptr = 0

        Goto 10

C  EOF - return TRUE.

200     Next_Char = .True.

999     Return
        End
```

```
      Subroutine Next_Vec(X1,Y1,X2,Y2,Exit_Code)

C  Error system code is 1000

C  This routine returns the next vector in the Plot10 input stream. Thus the
C  calling routine need not worry about the format of the Plot10 input file,
C  or what kind of data this file contains (characters, lines, etc) - it
C  simply sees this routine as a source of vectors.

C  Params:

C  X1, Y1 - I [O] = The coords of the start of the vector.
C  X2, Y2 - I [O] = The coords of the end of the vector.
C  EXIT_CODE - I [O] =   1      for end of file (no vector actually returned)
C                        0      for normal vector return
C                  <= -1000     for error system messages

C  The Plot-10 input stream consists of commands and data.  We parse this
C  stream with a hard-coded parsing automaton, a drawing of which should be
C  available.  See the 4010/4010-1 Maintenance Manual, Section 1, for a
C  description of the Plot-10 file format.  The commands and data mostly
C  describe moves (dark vectors) and vector draws (light vectors).  We keep
C  track of the current position (P_Xold,P_Yold) and update it whenever a
C  move or draw is performed.

C  Transitions between states in the automaton are made according to the
C  *token type* of the character last received from the input file.  This is a
C  way of dividing the ascii character set into a set of classes where
C  identical actions will be taken for all members of a class.  See the
C  routine Char_Type for more info.  Data is also divided up into different
C  types.  See the routine Data_Type for more info.

C  The input stream can also contain character data.  On a Plot10 display
C  device these would be generated by the hardware.  We generate them in
C  software.  This routine contains an interface to the Hewlett Packard Plot21
C  character generating software.  This software has been rigged to return
C  only the "next vector" in the character currently being generated, so
C  aquiring data for one character consisting of many draws will require may
C  calls to this routine and thus to the Plot21 generator.

C  This routine contains a significant hack related to character generation.
C  The Plot10 system maintains an internal buffer of commands which it dumps
C  to its output file when full.  Imagine that the user requests that Plot10
C  write a character string on the display.  Plot10 accomplishes this by
C  switching the display to alpha mode with a command, and then sending the
C  character string, followed by a switch back to graphics mode. If, however,
C  the string is too long to fit in the remainder of the buffer, Plot10 wraps
C  the string across two buffers by sending the following sequence: <part of
C  string> <newline> <US> <GS> <move command> <rest of string> Note that this
C  requires Plot10 to reset the current position since it is lost by the
C  display device during the switch from alpha mode to vector mode. To do this
```

```
C  Plot10 must compute the current position after the last character drawn in
C  the first buffer based on its knowledge of the size of the characters drawn
C  by the display hardware.  The problem here (finally!) is that our software
C  characters are not exactly the same size as those drawn by Plot10 display
C  hardware, particularly when scaled through the CHAR_SCALE parameter.
C  CHAR_ROTATION is even worse.  In any event, there would be a serious
C  discontinuity in the text on the page at the buffer-wrap division in the
C  character string. The solution is to detect buffer wraps by watching for
C  the sequence above, and to then ignore the repositioning command sent by
C  Plot10 and simply use the repositioning automatically done by the character
C  generator between adjacent characters in a string. Of course, this is not
C  fool-proof since it is possible that the user might send two character
C  strings seperated by a movement command one right after the other, and that
C  these would fall at a buffer boundary and appear just the same as the
C  sequence above, and that we would ignore the repositioning command put
C  there by the user.  Them's the breaks.  We warn the user about this in the
C  manual.

C  The recognition of the above sequence is done with the help of the variable
C  P_Buffer_Split_Hack.  As we see more and more of the elements in the
C  sequence above, the value of P_Buffer_Split_Hack increases.  Once it hits a
C  certain value (5) we know we have received the entire sequence. If the
C  sequence is interrupted at any time we reset the variable to 0. We make the
C  problem at the end of the last paragraph less serious by a check performed
C  before we're about to ignore the move command.  If the repositioning was
C  really done by Plot10 then the Y coordinate it sends will be the same as it
C  was at the start of the character string.  If it's not the same we assume
C  the user was doing the repositioning, and we do not in fact ignore the move
C  command.  Of course, it is possible that the user would reposition to the
C  same Y coordinate, but this is rather unlikely.  There are fewer cases to
C  screw us up.

C  This whole discussion above suggests a sneaky method for letting the
C  user specify character scaling and rotation dynamically from his Plot10
C  program.  It is unlikely that the user would ever want to issue three
C  move commands in sequence with no intervening draws since the first two
C  moves serve no purpose.  However, we might make them serve a purpose,
C  namely that of a code specifying a character scale or rotation factor.
C  For example, we might form the following associations:
C  Move-to (767,1023) followed by Move-to (767,1022) means that the X and Y
C  coordinates immediately following in the next Move command specify
C  character scaling and rotation values obtained by mapping the coordinate
C  ranges to some stated ranges of scaling and rotation. It would be simple
C  for the user (or us) to write routines to do all this encoding.  This
C  would seem to be easy to impliment, and a valuable feature to have.
C  Unfortunately I don't have the time to do it.

      Implicit Integer (A-Z)

      Integer X1, X2, Y1, Y2, Exit_Code
```

```
      Include 'Pl0interp.Inc'
      Include 'Termunits.Inc'
      Include 'Symvecmem.Inc'

C  The width of characters generated by Plot10 displays.

      Parameter ( Tek_Char_Width = 14 )

C  NEXT_CHAR obtains the next character in the input stream.  It returns
C  TRUE if it encounters end-of-file.

      Logical *1 Next_Char
      Logical *1 Hi, Lo, Seen801, Vector, String_Start

      Data Hi/.True./, Lo/.False./

C  We only issue the error message at label 801 once within a given call to
C  this routine.

      Seen801 = .False.

C  The first time this routine is called we go to state 1.  When we're in
C  the middle of light or dark vectors we proceed to state 4 when this
C  routine is called.  When we're in the middle of drawing a character we
C  go to the middle of state 1.2 .

      Goto (9,40,122), P_State
      Goto 800

C  BRANCH_CODE allows us to resume execution after issueing a non-fatal
C  error message.

C  Because of character scaling and rotation, our "current position" may
C  differ from the expected position Plot10 expects us to have after doing
C  character draws.  Thus we compute what Plot10 thinks is our current
C  position and keep it in P_Xkeep and P_Ykeep, restoring our actual current
C  position to these values after we finish writing a string. We must try to
C  keep the same current position as Plot10 because of the way it transmits
C  *modifications* to *existsing* bit fields.

9     P_Xkeep = 0
      P_Ykeep = 0

C  State 1 - we are in Alpha Mode.

10    String_Start = .True.

101   Branch_Code = 1
      If (Next_Char()) Goto 80
      Call Char_Type(Ctype)
```

```
C  Second element in buffer split sequence.

        If (Ctype .Eq. P_Crus) Then
                If (P_Buffer_Split_Hack .Eq. 1) Then
                        P_Buffer_Split_Hack = 2
                Else
                        P_Buffer_Split_Hack = 0
                Endif
                Goto 10
        Endif

        If (Ctype .Eq. P_Gs) Then
                P_Xold = P_Xkeep
                P_Yold = P_Ykeep
                Goto 20
        Endif
        If (Ctype .Eq. P_Printable) Goto 12

        P_Buffer_Split_Hack = 0

        If (Ctype .Eq. P_Ctlv) Goto 10
        If (Ctype .Eq. P_Bel) Goto 10

C  Non-null codes:

        If (Ctype .Eq. P_Siso) Goto 803
        If (Ctype .Eq. P_Esc) Goto 11
        If (Ctype .Eq. P_Nonprint) Goto 802

C  State 1.1 - we have received an escape sequence from Alpha mode.

11      Branch_Code = 2
        P_Buffer_Split_Hack = 0
        If (Next_Char()) Goto 804
        Call Char_Type(Ctype)

        If (Ctype .Eq. P_Sub) Goto 808
        If (Ctype .Eq. P_Enqetbff) Goto 806
        Goto 810

12      G_Char_Gen_State = -1

C  If we have the entire buffer split sequence and the Y coord is unchanged
C  since the last string we restore the previous character generator
C  coords.  If we're at the beginning of a string we then save the current
C  position so we can update it as Plot10 thinks it's being updated.

        X_Char_Start = P_Xold
        Y_Char_Start = P_Yold

        If (P_Buffer_Split_Hack .Eq. 5 .And. P_Ykeep .Eq. P_Yold) Then
```

```
                P_Xold = P_Xsave
                P_Yold = P_Ysave
        Endif

        If (String_Start) Then
                P_Ykeep = Y_Char_Start
                P_Xkeep = X_Char_Start
                String_Start = .False.
        Endif

C  Get the next vector in the current character.  In fact we obtain the
C  displacement to the end of the vector from the current point.

122     Call Swchar_Vec(P_Curr_Char,Action,Xrel,Yrel)

        X1 = P_Xold
        Y1 = P_Yold
        P_Xold = P_Xold + Xrel
        P_Yold = P_Yold + Yrel

        If (Action .Eq. Draw_Code) Then
                X2 = P_Xold
                Y2 = P_Yold
                Exit_Code = 0
                P_State = 3
                Goto 999
        Endif

C  Sometimes the character generator returns moves not draws.  We need
C  draws to make a full light vector.

        If (Action .Eq. Move_Code) Goto 122

C  There are no more vectors in this character.  Start the buffer split
C  sequence since we're at the end of a string, save the current position
C  in case another character follows, and update what Plot10 thinks our
C  current position is.

        P_Buffer_Split_Hack = 1
        P_Xsave = P_Xold
        P_Ysave = P_Yold
        P_State = 1
        P_Xkeep = P_Xkeep + Tek_Char_Width
        Goto 101

C  State 2 - We have just entered graphics mode.  We could get the third
C  step in the buffer split sequence.

20      Branch_Code = 3
        If (P_Buffer_Split_Hack .Eq. 2) Then
                P_Buffer_Split_Hack = 3
```

```
        Else
                P_Buffer_Split_Hack = 0
        Endif
        If (Next_Char()) Goto 80
        Call Char_Type(Ctype)

C  Check for command codes.

        If (Ctype .Eq. P_Siso) Goto 803
        If (Ctype .Eq. P_Bel) Goto 20
        If (Ctype .Eq. P_Crus) Goto 10
        If (Ctype .Eq. P_Esc) Goto 30

C  Assume we have coordinate data.  See what kind of data it is (HiY, LoX,
C  etc).  VECTOR keeps track of whether or not we are in dark or light
C  (TRUE) vector drawing mode.

        Vector = .False.
        Call Data_Type(Dtype)

        Goto (50,65,60,805), Dtype

C  State 3 - We process an escape sequence from either state 2 or 4.

30      Branch_Code = 4
        P_Buffer_Split_Hack = 0
        If (Next_Char()) Goto 804
        Call Char_Type(Ctype)

        If (Ctype .Eq. P_Sub) Goto 808
        If (Ctype .Eq. P_Enqetbff) Goto 806
        Goto 810

C  State 4 - We are in Graphics Mode and will draw a light vector. We might
C  proceed to the fifth step in the buffer split sequence.

40      Branch_Code = 5
        If (P_Buffer_Split_Hack .Eq. 4) Then
                P_Buffer_Split_Hack = 5
        Else
                P_Buffer_Split_Hack = 0
        Endif
        If (Next_Char()) Goto 80
        Call Char_Type(Ctype)

C  Check for commands.

        If (Ctype .Eq. P_Siso) Goto 803
        If (Ctype .Eq. P_Gs) Goto 20
        If (Ctype .Eq. P_Bel) Goto 20
        If (Ctype .Eq. P_Crus) Goto 10
```

```
        If (Ctype .Eq. P_Esc) Goto 30

C  We have coordinate data.  Save the current position as the start of the
C  vector, and process the data coordinates.

        Vector = .True.
        X1 = P_Xold
        Y1 = P_Yold
        Call Data_Type(Dtype)

        Goto (50,65,60,805), Dtype

C  Here we update the current position (p_xold,p_yold) with incoming
C  coordinate data.  The data type tells us what part of the current position
C  to update (HiX, LoY, etc).  Merge actually updates a coord.  We are done
C  updating for the current vector when LoX is received.

C  State 5 - Update HiY.

50      Call Merge(P_Yold,P_Curr_Char,Hi)

C  Process the next data value.

        If (Next_Char()) Goto 80

        Call Data_Type(Dtype)
        Goto (807,65,60,805), Dtype

C  State 6 - Update LoY.

60      Call Merge(P_Yold,P_Curr_Char,Lo)

        If (Next_Char()) Goto 80

        Call Data_Type(Dtype)
        Goto (70,65,807,805), Dtype

C  State 7 - Update HiX.

70      Call Merge(P_Xold,P_Curr_Char,Hi)

        If (Next_Char()) Goto 80

        Call Data_Type(Dtype)
        Goto (807,65,807,805), Dtype

C  State 6.5 - Update LoX.  The receipt of low X data means the current draw
C  (be it dark or light) is to be executed.  If this is a dark draw, we have
C  established a current position for the start of a vector; we now loop back
C  to state 4 to receive that vector.  If this is a light vector we have just
C  constructed the end coordinate; we now return the current vector to the
```

```
C  user.

65      Call Merge(P_Xold,P_Curr_Char,Lo)

        P_Seen_Vect = .True.

C  Step 3 in the buffer split sequence ?

        If (P_Buffer_Split_Hack .Eq. 3) Then
                P_Buffer_Split_Hack = 4
        Else
                P_Buffer_Split_Hack = 0
        Endif

        If (.Not. Vector) Goto 40

        X2 = P_Xold
        Y2 = P_Yold

        Exit_Code = 0
        P_State = 2
        Goto 999

C  State 8 - Legal end-of-file.

80      Exit_Code = 1
        Goto 999

C  Error processing... For some fatal errors we exit immediately, for others
C  we report where in the file the error occurred.  For all warning errors we
C  return to some state above which is determined by BRANCH_CODE.

C  Initialization not performed.

800     Exit_Code = -1000
        Goto 999

801     Continue
        If (.Not. Seen801) Write (T_Out,901)
901     Format (/' P10interp-W-Input file contains raw text in alpha
        1 mode.')
        Seen801 = .True.
        Goto 10

C  Input file contains raw control chars in alpha mode.

802     Exit_Code = -1001
        Goto 990

803     Continue
        Write (T_Out,903)
```

```
903     Format (/' Pl0interp-W-Input file attempts to change alpha mode
        1 character sets.')
        Goto (10,809,20,809,40), Branch_Code

C  Unexpected end of input file.

804     Exit_Code = -1002
        Goto 999

C  Illegal coordinate encountered in input file.

805     Exit_Code = -1003
        Goto 990

806     Continue
        If (P_Seen_Vect)  Write (T_Out,906)
906     Format (/' Pl0interp-W-Input file attempts to enquire, make copy,
        1 or page.')
        Goto (809,10,809,40,809), Branch_Code

C  Input file has illegal coordinate data sequence.

807     Exit_Code = -1004
        Goto 990

808     Continue
        Write (T_Out,908)
908     Format (/' Pl0interp-W-Input file attempts to activate cursor.')
        Goto (809,10,809,10,809), Branch_Code

809     Continue
        Write (T_Out,909) Branch_Code
909     Format (/' Pl0interp-I-Internal branching error; branch code = ',
        1 I5)
        Stop

C  Input file contains illegal escape sequence.

810     Exit_Code = -1005
        Goto 990

990     Write (T_Out,911) P_Reccnt, P_Bptr
911     Format (' Error occurred in plot10 input file in record ',I3,
        1   ' at character ',I3,'.')
        Goto 999

999     Return
        End
```

```
      Subroutine Next_Vec_In_Band_Alternating(S_Band_Top,S_Band_Bot,Sx1,Sy1,
     1  Sx2,Sy2,X_Ptos,Y_Ptos,Tymin,Tymax,Exit_Code)

C  Error system code is 1100

C  Returns the next vector in the current band by alternating reading from
C  and writing to two scratch files.  We make one pass over the Plot10
C  input file and fill the first band with the vectors that belong there.
C  We then write all vectors which fall within any of the remaining bands
C  into a scratch file.  For the next band we read vectors from this
C  scratch file, fill the band, and write only those vectors which fall
C  within the remaining bands to a second scratch file.  Now we read
C  vectors from this second scratch file and write remaining ones to the
C  first scratch file.  We continue alternating between scratch files in
C  this way, writing (hopefully) fewer and fewer vectors each time.

C  Parameters:

C  S_Band_Top - i [i] = The Sanders coordinate of the top of the current band.
C  S_Band_Bot - i [i] = The Sanders coordinate of the bottom of the current
C                       band.
C  Sx1, Sx2, Sy1, Sy2 - i [o] = The end points of the vector we return.
C  X_Ptos, Y_Ptos - r [i] = Scale factors to convert Tektronix coordinates
C                           to Sanders coordinates.
C  Tymin, Tymax - i [io] = The range of Y coordinates in the input file.

      Implicit Integer (A-Z)

      Real X_Ptos, Y_Ptos

      Integer Tunits(2)

      Include 'Pl0interp.Inc'
      Include 'Devices.Inc'
      Include 'Options.Inc'
      Include 'Nvibstate.Inc'

      Parameter ( Tu1 = 11, Tu2 = 12 )

      Data Tunits/ Tu1, Tu2 /

C  We can be in one of three states: passing over the Plot10 input file,
C  reading from the first scratch file, or reading from the second scratch
C  file.

      Goto (100,200,200), N_Nvib_State+1

C  Call Pl0init and open the first scratch file the first time we get here.

100   If (Done_Pl0init) Goto 110
```

```
        Done_Plotinit = .True.

        Call Plot_Init(Exit_Code)
        If (Exit_Code .Lt. 0) Goto 999

        Open (Unit=Tunits(1), Name='Nvib.Tmp', Type='Scratch',
     1     Form='Unformatted', Err=800)

110     Continue

C  Get the next vector from the Plot10 file.  Exit conditions are: error,
C  vector received, end of file.

        Call Next_Vec(Tx1,Ty1,Tx2,Ty2,Exit_Code)
        If (Exit_Code) 999,400,120

120     Continue

C  When we reach the end of the Plot10 input file we open the second
C  scratch file.

        Rewind Tunits(1)
        N_Nvib_State = 1

        Open (Unit=Tunits(2), Name='Nvib2.Tmp', Type='Scratch',
     1  Form='Unformatted', Err=800)
        Goto 999

C  Read from the appropriate scratch file until we reach its end.

200     Read (Tunits(N_Nvib_State),End=300) Tx1,Ty1,Tx2,Ty2
        Goto 410

C  At the end of a scratch file switch states and rewind the files.
C  Tell the calling routine that's it for this band.

300     N_Nvib_State = 3 - N_Nvib_State
        Rewind Tunits(1)
        Rewind Tunits(2)
        Exit_Code = 1
        Goto 999

400     Continue

C  We've got a vector.  If we're to rotate the picture left or right, do
C  so by exchanging coordinates (transposing the picture), and then
C  reflecting the point about the center X or Y coord of the picture.

        If (O_Image_Rotation .Ne. 0) Then
                Call Exchange(Tx1,Ty1)
                Call Exchange(Tx2,Ty2)
```

```
                If (O_Image_Rotation .Eq. 1) Then
                        Ty1 = D_Tek_Ymax - Ty1
                        Ty2 = D_Tek_Ymax - Ty2
                Else
                        Tx1 = D_Tek_Xmax - Tx1
                        Tx2 = D_Tek_Xmax - Tx2
                Endif
        Endif

C  Make sure the T-1 variables contain the bottom of the vector.

        If (Ty2 .Lt. Ty1) Then
                Call Exchange(Tx1,Tx2)
                Call Exchange(Ty1,Ty2)
        Endif

C  Keep track of the highest and lowest Y coords we've seen.

        Tymax = Max(Tymax,Ty2)
        Tymin = Min(Tymin,Ty1)

C  Map the Y coords to Sanders coordinates.  Don't waste time on X coords
C  we might not use.

410     Sy1 = Ty1 * Y_Ptos + .5
        Sy2 = Ty2 * Y_Ptos + .5

C  Write the vector to the scratch file if it's not totally contained by the
C  current band.

        If ( .Not. (Sy1 .Ge. S_Band_Bot .And. Sy2 .Le. S_Band_Top)) Then
                If (N_Nvib_State .Eq. 1) Then
                        Out_Unit = Tunits(2)
                Else
                        Out_Unit = Tunits(1)
                Endif
                Write (Out_Unit) Tx1,Ty1,Tx2,Ty2
        Endif

C  Only return the vector if it's within the current band.

        If (Sy1 .Gt. S_Band_Top .Or. Sy2 .Lt. S_Band_Bot)
        1       Goto (110,200,200), N_Nvib_State+1

C  We are going to use the X coords, so we map them.

        Sx1 = Tx1 * X_Ptos + .5
        Sx2 = Tx2 * X_Ptos + .5
        Goto 999
```

```
C  Error opening scratch file.

800      Exit_Code = -1100
         Goto 999

999      Return
         End
```

```
      Subroutine Next_Vec_In_Band_Multipass(S_Band_Top,S_Band_Bot,Sx1,Sy1,
     1  Sx2,Sy2,X_Ptos,Y_Ptos,Tymin,Tymax,Exit_Code)

C  Returns the next vector in the current band by making multiple passes over
C  the Plot10 input file.  This is inefficient since we decode vectors over
C  and over again even if they have been drawn in previous bands and will
C  never be needed again.  We also waste time repeating the interpreting
C  process itself many times.  For these reasons the Alternating version of
C  this routine should be faster.  However, because of its relatively high
C  overhead it can be slower for small files or for pictures with lots of long
C  vertical vectors.

C  S_Band_Top - i [i] = The Sanders coordinate of the top of the current band.
C  S_Band_Bot - i [i] = The Sanders coordinate of the bottom of the current
C                       band.
C  Sx1, Sx2, Sy1, Sy2 - i [o] = The end points of the vector we return.
C  X_Ptos, Y_Ptos - r [i] = Scale factors to convert Tektronix coordinates
C                           to Sanders coordinates.
C  Tymin, Tymax - i [io] = The range of Y coordinates in the input file.

      Implicit Integer (A-Z)

      Real X_Ptos, Y_Ptos

      Include 'P10interp.Inc'
      Include 'Options.Inc'

C  Initialize the Plot10 interpreting system the first time this routine is
C  called.

      If (P_Done_P10init) Goto 10

      P_Done_P10init = .True.

      Call P10_Init(Exit_Code)
      If (Exit_Code .Lt. 0) Goto 999

10    Continue

C  Get the next vector in the input file.

      Call Next_Vec(Tx1,Ty1,Tx2,Ty2,Exit_Code)
      If (Exit_Code) 999,30,20

C  We get here when there are no more vectors in the input file, in which case
C  we rewind the file and reset all the P10interp system variables in
C  preperation for the next band.

20    Continue

      Rewind P_Input_Unit
```

```
        P_State = 1
        P_Reccnt = 0
        P_Blen = 0
        P_Bptr = 0
        P_Xold = 0
        P_Yold = 0
        P_Seen_Vect = .False.
        Goto 999

30      Continue

C  We've got a vector.  If we're to rotate the picture left or right, do
C  so by exchanging coordinates (transposing the picture), and then
C  reflecting the point about the center X or Y coord of the picture.

        If (O_Image_Rotation .Ne. 0) Then
                Call Exchange(Tx1,Ty1)
                Call Exchange(Tx2,Ty2)

                If (O_Image_Rotation .Eq. 1) Then
                        Ty1 = D_Tek_Ymax - Ty1
                        Ty2 = D_Tek_Ymax - Ty2
                Else
                        Tx1 = D_Tek_Xmax - Tx1
                        Tx2 = D_Tek_Xmax - Tx2
                Endif
        Endif

C  Make sure the T-1 variables contain the bottom of the vector.

        If (Ty2 .Lt. Ty1) Then
                Call Exchange(Tx1,Tx2)
                Call Exchange(Ty1,Ty2)
        Endif

C  Keep track of the highest and lowest Y coords we've seen.

        Tymax = Max(Tymax,Ty2)
        Tymin = Min(Tymin,Ty1)

C  Map the Y coords to Sanders coordinates.  Don't waste time on X coords
C  we might not use.

        Sy1 = Ty1 * Y_Ptos + .5
        Sy2 = Ty2 * Y_Ptos + .5

C  Only return the vector if it's within the current band.

        If (Sy1 .Gt. S_Band_Top .Or. Sy2 .Lt. S_Band_Bot) Goto 10

C  We are going to use the X coords, so we map them.
```

```
            Sx1 = Tx1 * X_Ptos + .5
            Sx2 = Tx2 * X_Ptos + .5
            Goto 999

999         Return
            End
```

```
      Subroutine Output_Band(White_Band,X_Passes,Y_Passes,Xtra_Space,
      1  Font_X_Spacing)

C  Error system code is 900

C  This routinecontrols the transmission of the current band to the printer.

C  Parameters:

C  White_Band - Ll [i] = TRUE if there are no bits set in the current band.
C  X_Passes, Y_Passes - i [i] = The number of passes we must make in the X and
C       Y directions to print the raster at the resolution specified by the
C       user.
C  Xtra_Space - i [i] = When the X resolution specified by the user is greater
C       than the resolution of the font we have selected, we must insert
C       Xtra_Space heps of space between each of the dots we print to make the
C       resolution we print at equal to the requested resolution.
C  Font_X_Spacing - i [i] = The X resolution of the current font.

C  Printing a band is a fairly complex problem.  Perhaps it would be wise to
C  first state exactly what is to be acheived here.  We have an array of bits.
C  We want this array to appear on the page with a user-specified distance
C  between each element of the array.  To accomplish this task we have a
C  print head which prints six vertical dots at a spacing of 4 veps, and
C  which moves horizontaly in steps of either 8 or 16 heps.  These are our two
C  graphics fonts for printing.  But the head can be positioned with a
C  vertical resolution of 1 vep and a horizontal resolution of 1 hep.

C  There are three different cases to be considered.  These are when the
C  desired resolution along the given axis is:
C  1) Less than the font's resolution along that axis.
C  2) Equal to the font's resolution along that axis.
C  3) Greater than the font's resolution along that axis.

C  There are actually two different problems here: one applies to the X axis
C  and one to the Y axis.  They are different because along the Y axis there
C  are six pins lined up at a fixed spacing, while the print head is only one
C  pin wide. Let us discuss these two problems seperately.

C  The solution is fairly simple for the X axis.  Case (2) is obviously
C  trivial. There is a one-to-one mapping between steps within the font and
C  dots within the array.  Case (3) is still simple.  We choose the highest
C  font resolution (8 or 16 heps) still less than the desired resolution, and
C  insert D-F heps of extra space between each dot printed by the font, where
C  D is the desired resolution and F is the font resolution.  Things get a bit
C  tricky for (1).  The way I thought about this was to consider a horizontal
C  grid with spacings of 1 hep. I then considered patterns of dots overlaid on
C  this grid at the desired resolution, and I divided the grid up into blocks
C  at intervals of the font resolution.  The first printing dot position in the
C  row is in the far left of the first block.  As we look to the right along
C  the row we can imagine a pattern forming on the grid of where points from
```

```
C  the band can fall on the page.  There will be a certain distance after
C  which there will again be a dot in the far left of a block - one period.
C  The number of dots that fall within the first block is the number of passes
C  we must make with the print head in the X direction - because on one pass
C  it is impossible to lay more than one dot within one block, given the
C  definition of a block.  Imagine two cases here: a desired resolution of 4
C  heps and a desired resolution of 5 heps.  In both cases we need two passes.
C  In the first case, all the dots printed in one pass are seperated by 8 heps
C  - the font resolution.  But in the second case the dots printed in one pass
C  are seperated by 10 heps.  Thus it is just as if we were printing two
C  overlaid pictures at 10 heps resolution - just like case (3) above.  So we
C  must insert 2 heps of extra space between each dot we print.


C  Printing along the Y axis is more complex.  Case (3) is no longer trivial
C  since we can't insert extra space between sequentially printed dots.  We
C  again consider a grid (now vertical) consisting of all printable dots in a
C  column, with points spaced at the desired resolution highlighted. We now ask
C  the question: how many of the different positions within a block have dots
C  in them? If they all do, then we must print at a resolution of 1 vep.  If
C  only every other one does we must print at a resolution of 2 veps, etc.
C  And obviously printing at a resolution of 2 veps with a 4 vep font with no
C  capability of inserting extra space requires two passes. So, in
C  Calc_Band_Params we see that the number of passes needed to print at a
C  given resolution along the Y axis is:

C  Font_Y_Spacing / Gcd(Font_Y_Spacing,O_Vertical_Dot_Spacing)

C  where Gcd is a Greatest Common Denominator function.  Note this applies to
C  cases (1) and (3).  Note that if the desired resolution is 3 veps we must
C  print at a resolution of 1 vep since in the course of 1 period a dot can
C  map to any position within a block.

C  Now we must consider the problem of actually figuring out which band dots
C  go in which pin positions during the different interlaces. We start with
C  the aim of being able to print with as many pins as possible on each print
C  head sweep.  It simply won't do to step along one row at a time using only
C  the top pin on the head to print. If we visualize a print head stepping
C  down the grid at intervals of (4 veps) / (# passes), we see that at each
C  head position, some pins will lay at positions on the page that correspond
C  to points in the band (the pins are mapped = the pins might print), and
C  some pins lay at positions that fall between points in the band (the pins
C  are unmapped = the pins cannot print).  This observation is the basis of
C  the algorithm used to print.

C  First imagine dividing a band into smaller units - called an interlace
C  group. Imagine we are printing at a resolution of 2 veps.  This requires
C  two passes with the printhead offset from its first position by 2 veps.
C  Since we print six dots on each pass we print a total of twelve dots in the
C  two passes, so the interlace group consists of twelve dots.  After printing
```

```
C  these two passes we must re-position the print head to print the next
C  interlace unit of twelve dots.

C  The print head will be moved in a constant, periodic pattern consisting of
C  repeating interlace groups.  It begins at the top of an interlace group,
C  performs the required number of passes, and then proceeds to the top of the
C  next interlace group, where it starts over again.  The printer starts at
C  the top of a band, and proceeds downwards in known distances along the
C  page.  The top of the band in memory (the band) corresponds to the top of
C  the band on the page (the image).  Stepping between rows in the band is
C  equivalent to moving a certain distance along the image (the desired
C  resolution).  What the algorithm has to do is to decide when a given print
C  head pin lies at the exact same distance from the top of the image as some
C  row lies from the top of the band.

C  This is done by maintaining the variables Band_Head_Pos (BHP) and
C  Next_Dot_Dis (NDD).  BHP is the row in the band at or below the top image
C  row in the current interlace group.  NDD is the distance in veps to that
C  closest row.  If NDD is zero then the top pin in the head on the first pass
C  in the interlace group falls at the same place on the image as the band
C  row pointed to by BHP.  Thus that rows maps onto that pin.

C  The routine Track takes care of incrementing BHP and NDD as we move along
C  the image.  This incrementing is done at two different times.  When we
C  move to a new interlace group the print head must shift down to a new
C  position within the image.  By moving this distance in the image it also
C  moves through some number of rows (possibly not an integer number) in the
C  band.  The new position may be between rows or right on top of a row
C  (NDD=0), and based on the distance moved and the image distance represented
C  by a step between band rows (the desired Y resolution), we can compute
C  new values for BHP and NDD.

C  Track is also used to compute values for BHP and NDD for the other pins in
C  the head.  Given base values for these variables at the top pin in the
C  head, each of the other pin positions in the head represent offsets along
C  the image and hence spacings within the band.  BHP and NDD are computed
C  for each pin within the head, and if NDD is zero for a given pin then that
C  pin actually maps to a row within the band.  Otherwise the pin will not
C  strike on that pass.  This situation is complicated by one more detail,
C  namely the offset of the head during the different passes within an
C  interlace group. These offsets simply mean that the base values of BHP and
C  NDD for the top pin in the head will be slightly different for each
C  vertical pass.

C  This discussion has been quite general, and applies to all the cases above
C  (1-3).  However, note that for case (2), and for certain instances of (1)
C  in which the font resolution is an integer multiple of the desired
C  resolution, we are really doing a lot of extra work with all this
C  computation of BHP and NDD. For in these cases none of the pins on the head
C  ever fall between band rows.  All the pins always map to a band row.
```

```
C  There is one further complication here, which is that it is possile that
C  the height of the image may not equal the size of an integer number of
C  interlace groups.  In this case some of the last pins on the head in that
C  last interlace group will map off the edge of the band. So at the end of
C  this last interlace group we will shift the head down to the position in
C  the image corresponding to the end of the band instead of the end of the
C  interlace group.  And we will also be sure that these pins do not print.

       Implicit Integer (A-Z)

       Logical *1 White_Band

       Include 'Devices.Inc'
       Include 'Band.Inc'
       Include 'Options.Inc'

C  Outbuf holds the internal image of the graphics characters that are to
C  be put on one output line.  Since the maximum number of characters that
C  can be printed on one line in one pass is:
C  (13 inch max page width) * (960 heps / inch) / (8 hep minimum resolution)
C                           = 1560
C  If Outbuf is smaller than this there will be a software limitation on the
C  maximum size and resolution the user can print at.

       Parameter ( Obmax = 1600 )
       Byte Outbuf(Obmax)

C  Be sure Outbuf is big enough to print the band.

       If (B_Band_Width/X_Passes .Gt. Obmax) Then
              Exit_Code = -900
              Goto 999
       Endif

C  Compute the number of interlace groups for this band.

       Interlace_Groups =
       1  Ceiling(Float(O_Vertical_Dot_Spacing*B_Band_Height) /
       2     Float(D_Print_Head_Size))

C  Compute the shift in head position at the end of the last interlace group.
C  See note above.

       Final_Head_Movement = D_Print_Head_Size -
       1  (Interlace_Groups * D_Print_Head_Size -
       2        O_Vertical_Dot_Spacing * B_Band_Height)

C  Compute the vertical distance between head sweeps within an interlace
C  group (even if there's no Y interlacing).

       Y_Interlace_Spacing = D_Print_Head_Pin_Spacing / Y_Passes
```

```
C  Initialize vars...  Blen is the number of chars in Outbuf.

        Band_Head_Pos = 0
        Next_Dot_Dis = 0
        Blen = 0

C  Loop through all the interlace groups.

        Do 10 Inter_Group = 1, Interlace_Groups

C  Loop through the X and Y passes within each interlace group.

        Do 20 Ypass = 0, Y_Passes-1
        Do 30 Xpass = 0, X_Passes-1

C  If we know the band is all white space we can skip a lot of computation.

        If (.Not. White_Band) Then

C  Compute the image of the current head sweep.

                Call Make_Line_Vms(Outbuf,Obmax,Blen,Xpass,Ypass,X_Passes,
     1                  Y_Passes,Band_Head_Pos,Next_Dot_Dis)

C  If the current head sweep is not white space then print it.

                If (Blen .Ne. 0)
     1                  Call Output_Line(Outbuf,Blen,Xtra_Space,Font_X_Spacing)

C  If we're not on the last X pass at this Y level, then shift the head to
C  the right to prepare for the next X pass.

                If (Xpass .Ne. X_Passes-1)
     1                  Call Pcl_Xinterlace((Xpass+1)*O_Horizontal_Dot_Spacing)

        Endif

30      Continue

C  If we're not at the last Y pass within this interlace group then shift the
C  head down to prepare for the next Y pass.

        If (Ypass .Ne. Y_Passes-1)  Call Pcl_Ydown(Y_Interlace_Spacing)

20      Continue

C  Shift the head down to prepare for the next interlace group, which may or
C  may not map to the current band.

        If (Inter_Group .Eq. Interlace_Groups) Then
```

```
            Head_Skip = Final_Head_Movement - Y_Interlace_Spacing*
     1              (Y_Passes-1)
            Else
            Head_Skip = D_Print_Head_Size -
     1                                 Y_Interlace_Spacing*(Y_Passes-1)
            Endif

            Call Pcl_Ydown(Head_Skip)

C   Track to the next base values of BHP and NDD.

            Call Track(Next_Dot_Dis,D_Print_Head_Size,Band_Head_Pos)

10          Continue

999         Return
            End
```

```
      Subroutine Output_Line(Outbuf,Blen,Xtra_Space,Font_X_Spacing)

C  This routine takes a buffer of data to be sent to the printer as one head
C  sweep, and manipulates it so it can really be sent to the printer. This
C  involves converting the data into graphic characters, packing them into an
C  output buffer of limited size, and compressing the white space in the
C  string to graphic spacing characters to eliminate a shitload of I/O.  Each
C  byte of Outbuf is now a bit map of the printing pins in one print head
C  strike.  Each bit of the six low bits of the byte represent one pin in the
C  head (low bit = top pin).  Each of these bytes will map into a printing
C  character which is sent to the printer. See the document "Sanders Technology
C  Product Bulletin: Media 12/7 Typographic Printer Graphics Option Users's
C  Instructions" (!) for more info.

C  NOTE:  Currently it impossible to print at any X resolution above 8 heps
C  (except of course 16 heps) because the graphic spacing characters do not
C  work so we cannot insert extra space between dots.

C  Parameters:

C  Outbuf(Blen) - b [i] = The buffer of characters to be printed.
C  Xtra_Space - i [i] = The number of heps of space to insert between each
C       horizontal print head position so we print at the required resolution.
C  Font_X_Spacing - i [i] = The X resolution of the font we're printing with.

      Implicit Integer (A-Z)

      Byte Outbuf(Blen)

      Include 'Devices.Inc'
      Include 'Options.Inc'

      Byte Pbuf(D_Max_Output_Line_Length), Low_L

      Character *9 Move_Command
      Character *15 Extra_Space_String, White_Space_String
      Character *3 Numstr

      Logical *1 In_Space_Block, Squeezing, Come_Back

      Parameter ( Leave_Spaces = 15, Esc = 27, Max_2char_Esl_Param = 31*64 )

      Low_L = 108

C  Convert the extra space value into a graphic spacing string.  See this
C  routine for more info.

      Call Make_Space_String(Xtra_Space,Extra_Space_String,Sslen)

C  Pbuf is the output buffer; Plen is its current length.
C  Squeezing is true when we are in the middle of a block of spaces in Outbuf
```

```
C  which is long enough to warrant conversion into a space string.
C  In_Space_Block is true when we are in a block of spaces in Outbuf.

        Plen = 0
        In_Space_Block = .False.
        Squeezing = .False.

C  Scan through Outbuf looking for long blocks of spaces which we convert to a
C  (much shofrter) graphics spacing string.

        Do 10 Bndx = 1, Blen

C  Have we got a space (no printing pins) ?

        If (Outbuf(Bndx) .Eq. 0) Goto 100

C  If we're not at the end of a block of spaces, just go and process the
C  current printing character.

        If (.Not. In_Space_Block) Goto 50

C  We are at the end of a block of spaces.  If we are compressing them into
C  graphic spacing characters, go do so.  If we're not, just output them as
C  a set of spaces.

        If (Squeezing) Goto 30

C  Insert a block of spaces into the output buffer.

        Do 20 Pndx = Plen+1, Plen+Space_Count
20      Pbuf(Pndx) = ' '
        In_Space_Block = .False.
        Plen = Plen + Space_Count
        Goto 50

30      Continue

C  Convert a block of spaces to graphic spacing characters.

C  I've coded two ways of printing white space: using horizontal movement
C  commands and using graphic spacing characters (which don't work).
C  If they ever do work, uncomment the c% and c# lines in this file and
C  get rid of the horizontal movement command logic (between lines marked
C  "c&").

ccccccccc%      White_Size = font_x_spacing * Space_Count

ccccccccc%      Call Make_Space_String(White_Size,White_Space_String,Wslen)

ccccccccc%      Do 40 Pndx = Plen+1, Plen+Wslen
ccccccccc%40    Pbuf(Pndx) = Ichar(White_Space_String(Pndx-Plen:Pndx-Plen))
```

```
ccccccccc%      Plen = Plen + Wslen

c& - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

C  Compute the number of heps to move.

        White_Size = Font_X_Spacing * Space_Count

C  Create a movement command in the current language.  If we're using ESL
C  there's a limit to how far we can tell the carriage to move with one
C  command, so we may have to issue several commands.

38      If (O_Rcl_Mode) Then
                Write (Move_Command,900) D_Delimiter, White_Size, D_Delimiter
900             Format (A1,'Fe',I5,A1)
                Move_Comm_Length = 9
        Else
                If (White_Size .Gt. Max_2char_Esl_Param) Then
                        Space_Size = Max_2char_Esl_Param
                        White_Size = White_Size - Space_Size
                        Come_Back = .True.
                Else
                        Space_Size = White_Size
                        Come_Back = .False.
                Endif

                Call Esl_Numb(Space_Size,2,Numstr)
                Write (Move_Command,902) Esc, Low_L, Numstr(1:2)
902             Format (A1,A1,A2)
                Move_Comm_Length = 4
        Endif

C  Pack the movement command into the output buffer.  Go back if we didn't
C  get the whole distance because of ESL.

        Do 40 Pndx = Plen+1, Plen+Move_Comm_Length
40      Pbuf(Pndx) = Ichar(Move_Command(Pndx-Plen:Pndx-Plen))
        Plen = Plen + Move_Comm_Length
        If (Come_Back) Goto 38

c& - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

        In_Space_Block = .False.
        Squeezing = .False.

C  Insert a printing graphic character into the output buffer. Convert from
C  bit-map to graphic character by just doing a little addition.

50      Plen = Plen + 1
        Pbuf(Plen) = Outbuf(Bndx) + 32
```

```
C  The following is supposed to modify the printing X resolution by inserting
C  graphic spacing characters.  But these don't work so we won't do it.

ccccccccc#        If (Sslen .Ne. 0) Then
ccccccccc#                Do 55 Ndx = 1, Sslen
ccccccccc#55              Pbuf(Plen+Ndx) = Ichar(Extra_Space_String(Ndx:Ndx))
ccccccccc#                Plen = Plen + Sslen
ccccccccc#        Endif

C  Flush the output buffer if it's about to overflow.  We need to keep a lot
C  of space reserved because long command sequences can crop up out of
C  nowhere.

          If (Plen .Eq. D_Max_Output_Line_Length-40)
          1               Call Flush_Pbuf(Pbuf,D_Max_Output_Line_Length,Plen)
          Goto 10

C  We're in a space block.  If it's of length >= Leave_Spaces we'll compress
C  it.

100       If (Plen .Gt. D_Max_Output_Line_Length - Leave_Spaces - 40)
          1               Call Flush_Pbuf(Pbuf,D_Max_Output_Line_Length,Plen)

          If (In_Space_Block) Goto 110

          In_Space_Block = .True.
          Space_Count = 1
          Goto 10

110       Space_Count = Space_Count + 1
          If (Space_Count .Gt. Leave_Spaces) Squeezing = .True.
          Goto 10

10        Continue

C  We're now at the end of the current head sweep.  There's a mechanical
C  problem in the printer.  When it's printing bidirectionally it has trouble
C  getting back up to speed immediately, so if there's a straight line of text
C  along the right of the picture it will come out jagged because the printer
C  is computing the head position improperly.  Thus we always tell the head to
C  move past the far right of the current line so it has time to get back up
C  to speed by the time it gets back into the area where it will strike. We
C  have to code this for ESL and RCL.  Don't bother with this if there's
C  nothing in the output buffer, which will probably occur when a line is all
C  white space.

          If (Plen .Ne. 0) Then
                  If (O_Rcl_Mode) Then
                          Pbuf(Plen+1) = Ichar(D_Delimiter)
                          Pbuf(Plen+2) = 'F'
                          Pbuf(Plen+3) = 'E'
```

```
                Pbuf(Plen+4) = '9'
                Pbuf(Plen+5) = '9'
                Pbuf(Plen+6) = Ichar(D_Delimiter)
                Plen = Plen + 6
        Else
                Pbuf(Plen+1) = Esc
                Pbuf(Plen+2) = Low_L
                Pbuf(Plen+3) = 'B'
                Pbuf(Plen+4) = '@'
                Plen = Plen + 4
        Endif
        Call Flush_Pbuf(Pbuf,D_Max_Output_Line_Length,Plen)
Endif

Return
End
```

```
      Subroutine Pl0_Init(Exit_Code)

C  Error system code is 1200

C  Initializes the Plot-10 interpreting package.

      Implicit Integer (A-Z)

      Include 'Pl0interp.Inc'

      Integer Exit_Code

C  Try to open the file.

      Open (Unit=P_Input_Unit, Type='Old',  Access='Sequential',
     1  Form='Formatted', Readonly, Err=800)

C  Success; initialize common variables.

      P_State = 1
      P_Reccnt = 0
      P_Blen = 0
      P_Bptr = 0
      P_Xold = 0
      P_Yold = 0
      P_Seen_Vect = .False.
      Goto 999

C  Error opening file.

800   Exit_Code = -1200
      Goto 999

999   Return
      End
```

```
        Subroutine Pcl

C   This routine contains a number of different entry points, each of which is
C   in charge of issuing a different Printer Command Language command. Of
C   course, there are two different printer command languages: Readable Command
C   Language and Escape Sequence Language.  Each entry here can issue its
C   command in either of these languages, and does so based on the option
C   O_Rcl_Mode, which tells which mode the user has selected. Obviously, this
C   implies that the decision about which language to issue a command in is a
C   run-time decision.  It would be possible for the program installer to
C   comment out lines relating to one language if he wanted to only support one
C   language and save a little program space.  It's probably not worth the
C   effort though considering the amount of space you'd probably save (~300
C   bytes max?).

C   The method is simply to pack a command sequence into a character string
C   through concatenations, and then send it to the printer.

        Implicit Integer (A-Z)

        Include 'Devices.Inc'
        Include 'Options.Inc'

        Character *1 Low_D, Low_A, Low_O, Low_L
        Character *20 Command, Numstr*5, Newdel*1, Font_Id*4
        Character *1 Char

        Parameter ( Esc = 27, Cr = 13 )

C   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   Pcl_Defdel

        Entry Pcl_Defdel

C   Set the delimiter for RCL to the default delimiter ("!").

        If (O_Rcl_Mode) Then
                Command = D_Delimiter // 'Cd' // D_Delimiter
                D_Delimiter = D_Default_Delimiter

                Call Ch_Toprinter(Command,4)
        Endif

        Return


C   - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   Pcl_Fion

        Entry Pcl_Fion

C   Set RCL fill on.
```

```
        If (O_Rcl_Mode) Then
                Command = D_Delimiter // 'Fion' // D_Delimiter

                Call Ch_Toprinter(Command,6)
        Endif
        Return

C  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   Pcl_Fioff

        Entry Pcl_Fioff

C  Turn RCL fill off.

        If (O_Rcl_Mode) Then
                Command = D_Delimiter // 'Fioff' // D_Delimiter

                Call Ch_Toprinter(Command,7)
        Endif
        Return


C  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   Pcl_Af

        Entry Pcl_Af(Log_Font,Font_Id)

C  Assign a logical font number to a font identification code.

C  Parameters:

C  Log_Font - i [i] = The logical font number to associate with the font.
C  Font_Id - c*4 [i] = The code for the font to associate with the number
C       (e.g., '0817').

        If (O_Rcl_Mode) Then
                Call Tostring(Log_Font,Numstr)
                Command = D_Delimiter // 'Af' // Numstr // ',' // 'Graf' //
        1                       Font_Id // D_Delimiter

                Call Ch_Toprinter(Command,18)
        Else
                Call Esl_Numb(Log_Font,1,Numstr)
                Low_D = Char(100)
                Command = Char(Esc)// Low_D // Numstr(1:1) // 'Graf' // Font_Id

                Call Ch_Toprinter(Command,11)
        Endif

        Return

C  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   Pcl_Cd
```

```
        Entry Pcl_Cd(Newdel)

C  Define a new delimiter for RCL.

C  Parameters:

C  Newdel - c*1 [i] = The delimiter to use now.

        If (O_Rcl_Mode) Then
                Command =
     1          D_Delimiter // 'Cd' // Newdel // ',' // Newdel // D_Delimiter
                Call Ch_Toprinter(Command,7)

                D_Delimiter = Newdel
        Endif

        Return


C  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   Pcl_Sf

        Entry Pcl_Sf(Font)

C  Selects a font by logical font number.

C  Parameters:

C  Font - i [i] = The logical font number to select.

        If (O_Rcl_Mode) Then
                Call Tostring(Font,Numstr)
                Command = D_Delimiter // 'Sf' // Numstr // D_Delimiter
                Call Ch_Toprinter(Command,9)
        Else
                Call Esl_Numb(Font,1,Numstr)
                Low_A = Char(97)
                Command = Char(Esc) // Low_A // Numstr(1:1)
                Call Ch_Toprinter(Command,3)
        Endif

        Return


C  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   Pcl_Xinterlace

        Entry Pcl_Xinterlace(Heps)

C  Used to reposition the print head during X interlacing, this routine issues
C  a <carriage-return> <right-movement> command to displace the head slightly
C  from the left margin to begin an interlace pass.
```

```
C  Parameters:

C  Heps - i [i] = The number of heps to displace the head from the margin.

      If (O_Rcl_Mode) Then
              Call Tostring(Heps,Numstr)
              Command = D_Delimiter // 'Cr' // D_Delimiter // D_Delimiter //
     1                    'Fe' // Numstr // D_Delimiter
              Call Ch_Toprinter(Command,13)
      Else
              Call Esl_Numb(Heps,2,Numstr)
              Low_L = Char(108)
              Command = Char(Cr) // Char(Esc) // Low_L // Numstr(1:2)
              Call Ch_Toprinter(Command,5)
      Endif

      Return

C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  Pcl_Ydown

      Entry Pcl_Ydown(Veps)

C  Issues a <carriage-return> <move-down> command, used to re-position the
C  print head vertically.

C  Parameters:

C  Veps - i [i] = The number of veps to move the head down.

      If (O_Rcl_Mode) Then
              Call Tostring(Veps,Numstr)
              Command = D_Delimiter // 'Cr' // D_Delimiter // D_Delimiter //
     1                    'Vf' // Numstr // D_Delimiter
              Call Ch_Toprinter(Command,13)
      Else
              Call Esl_Numb(Veps,2,Numstr)
              Low_O = Char(111)
              Command = Char(Cr) // Char(Esc) // Low_O // Numstr(1:2)
              Call Ch_Toprinter(Command,5)
      Endif

      Return
      End
```

```
      Subroutine Read_Bands(Font_X_Spacing,Exit_Code)

C  Error system code is 1400

C  This routine reads a set of bands from a raster file and sends them to the
C  printer.

C  Parameters:

C  Font_X_Spacing - i [i] = The X resolution of the font we're printing with.

      Implicit Integer (A-Z)

      Include 'Devices.Inc'
      Include 'Band.Inc'
      Include 'Options.Inc'

      Real Adummy
      Byte Bdummy

      Logical *1 White_Band

      Parameter ( Font_Y_Spacing = 4 )

C  Open the raster file.

      Open (Unit = D_Raster_Unit, Type = 'Old', Form='Unformatted',
     1       Access='Sequential', Readonly, Err=800)

C  Skip any comment records at the beginning.

      Read (D_Raster_Unit,Err=801) Recs_To_Skip
      If (Recs_To_Skip .Ne. 0) Then
              Do 10 N = 1, Recs_To_Skip
10            Read(D_Raster_Unit,Err=801) Bdummy
      Endif

C  Read number of bands and band width.

      Read (D_Raster_Unit,Err=801) Nbands, B_Band_Width

C  Calculate only number of passes and extra space.

      Call Calc_Band_Params(Font_X_Spacing,Font_Y_Spacing,Adummy,Adummy,
     1  Idummy,Idummy,Idymmy,X_Passes,Y_Passes,Xtra_Space,
     2  Exit_Code)
      If (Exit_Code .Lt. 0) Goto 999

C  For each band....

      Do 20 Iband = Nbands, 1, -1
```

```
C  Read height and number of bytes for band.

        Read (D_Raster_Unit,Err=801) B_Band_Height, Band_Storage

C  Make sure we can accomodate the band.

        If (B_Band_Height .Gt. B_Band_Bytes) Then
                Exit_Code = -1401
                Goto 999
        Endif

C  Read the band.

        Read (D_Raster_Unit,Err=801) (B_Band(N),N=1,Band_Storage)

C  See if it's all whitespace.

        Do 30 N = 1, Band_Storage
30      If (B_Band(N) .Ne. 0) Goto 40

        White_Band = .True.
        Goto 50

40      White_Band = .False.

50      Continue

C  Print the band if the user wants us to.

        If (O_Print)
        1  Call Output_Band(White_Band,X_Passes,Y_Passes,Xtra_Space,
        2  Font_X_Spacing,Exit_Code)
        If (Exit_Code .Gt. 0) Goto 999

20      Continue

        Close (Unit = D_Raster_Unit)
        Goto 999

800     Exit_Code = -1400
        Goto 999

801     Exit_Code = -1401
        Goto 999

999     Return
        End
```

```
      Subroutine Save_Band

C  Writes the current band to the raster data file.

      Implicit Integer (A-Z)

      Include 'Devices.Inc'
      Include 'Band.Inc'

C  Compute the number of bytes we're using to store the band.

      Band_Storage = Ceiling(Float(B_Band_Width * B_Band_Height) /
     1       Float(D_Bits_Per_Byte))

C  Write out band height, storage requirement, and the band itself.

      Write (D_Raster_Unit) B_Band_Height, Band_Storage

      Write (D_Raster_Unit) (B_Band(N),N=1,Band_Storage)

999   Return
      End
```

```
      Subroutine Script(Char,Xpos,Ypos,Exit_Code)

C  This routine returns the next relative move/draw to generate the current
C  character in a script font.

C  Parameters:

C  CHAR - b [i] = The character to generate.
C  XPOS,YPOS - R [o] = A relative move/draw coordinate.
C  EXIT_CODE - i [o] = 2 for a move
C                      3 for a draw
C                     -1 for no more graphic elements in character.

C  This routine was originally part of the Hewlett-Packard Plot21 system.
C  Its logic, but not its data areas, have been substantially re-written.

C  For documentation of the program logic, see the body of the virtually
C  identical routine TRIPLX.

      Implicit Integer (A-Z)

      Integer Ic(876),In(114),Ie(280)

      Byte Char

      Real Xm, Ym , Xpos, Ypos

      Include 'Chargen.Inc'
      Include 'Symvecmem.Inc'
      Include 'Options.Inc'

C     ********************************************************
C     *                                                      *
C     *                                                      *
C     *      THIS MODULE GENERATES SCRIPT CHARACTERS         *
C     *                                                      *
C     *                                                      *
C     ********************************************************
C
C
C
C     ********************************************************
C     *           CHARACTER TABLE                            *
C     ********************************************************
C------<< 65 A >>
      Data Ic( 1  )/ 2   /,Ic( 2  )/ 3   /,Ic( 3  )/ 4   /
      Data Ic( 4  )/ 5   /,Ic( 5  )/ 6   /,Ic( 6  )/ 7   /
      Data Ic( 7  )/ 8   /,Ic( 8  )/ 9   /,Ic( 9  )/ 10  /
      Data Ic( 10 )/ 11  /,Ic( 11 )/ 12  /,Ic( 12 )/ 2   /
      Data Ic( 13 )/ 13  /,Ic( 14 )/ 142 /,Ic( 15 )/ 181 /
      Data Ic( 16 )/ 18  /,Ic( 17 )/ 19  /
```

```
C------<< 66 B >>
        Data Ic( 18 )/ 143 /,Ic( 19 )/ 38  /,Ic( 20 )/ 39  /
        Data Ic( 21 )/ 15  /,Ic( 22 )/ 144 /,Ic( 23 )/ 13  /
        Data Ic( 24 )/ 28  /,Ic( 25 )/ 40  /,Ic( 26 )/ 1   /
        Data Ic( 27 )/ 29  /,Ic( 28 )/ 41  /,Ic( 29 )/ 18  /
        Data Ic( 30 )/ 146 /,Ic( 31 )/ 43  /,Ic( 32 )/ 44  /
C------<< 67 C >>
        Data Ic( 33 )/ 45  /,Ic( 34 )/ 46  /,Ic( 35 )/ 21  /
        Data Ic( 36 )/ 47  /,Ic( 37 )/ 147 /,Ic( 38 )/ 48  /
        Data Ic( 39 )/ 44  /,Ic( 40 )/ 43  /,Ic( 41 )/ 42  /
        Data Ic( 42 )/ 40  /,Ic( 43 )/ 148 /,Ic( 44 )/ 34  /
C------<< 68 D >>
        Data Ic( 45 )/ 51  /,Ic( 46 )/ 149 /,Ic( 47 )/ 52  /
        Data Ic( 48 )/ 151 /,Ic( 49 )/ 16  /,Ic( 50 )/ 152 /
        Data Ic( 51 )/ 54  /,Ic( 52 )/ 32  /,Ic( 53 )/ 153 /
        Data Ic( 54 )/ 42  /,Ic( 55 )/ 27  /,Ic( 56 )/ 14  /
        Data Ic( 57 )/ 50  /,Ic( 58 )/ 56  /,Ic( 59 )/ 181 /
        Data Ic( 60 )/ 39  /
C------<< 69 E >>
        Data Ic( 61 )/ 57  /,Ic( 62 )/ 46  /,Ic( 63 )/ 21  /
        Data Ic( 64 )/ 47  /,Ic( 65 )/ 53  /,Ic( 66 )/ 48  /
        Data Ic( 67 )/ 44  /,Ic( 68 )/ 58  /,Ic( 69 )/ 59  /
        Data Ic( 70 )/ 40  /,Ic( 71 )/ 25  /,Ic( 72 )/ 178 /
        Data Ic( 73 )/ 21  /,Ic( 74 )/ 18  /,Ic( 75 )/ 42  /
        Data Ic( 76 )/ 40  /,Ic( 77 )/ 13  /,Ic( 78 )/ 25  /
        Data Ic( 79 )/ 178 /,Ic( 80 )/ 21  /,Ic( 81 )/ 152 /
        Data Ic( 82 )/ 34  /
C------<< 70 F >>
        Data Ic( 83 )/ 154 /,Ic( 84 )/ 1   /,Ic( 85 )/ 7   /
        Data Ic( 86 )/ 66  /
C------<< 71 G >>
        Data Ic( 87 )/ 2   /,Ic( 88 )/ 3   /,Ic( 89 )/ 67  /
        Data Ic( 90 )/ 68  /,Ic( 91 )/ 33  /,Ic( 92 )/ 69  /
        Data Ic( 93 )/ 155 /,Ic( 94 )/ 25  /,Ic( 95 )/ 178 /
        Data Ic( 96 )/ 39  /,Ic( 97 )/ 70  /,Ic( 98 )/ 37  /
        Data Ic( 99 )/ 53  /,Ic( 100 )/ 34  /,Ic( 101 )/ 71  /
        Data Ic( 102 )/ 72  /,Ic( 103 )/ 25  /,Ic( 104 )/ 27  /
        Data Ic( 105 )/ 156 /
C------<< 72 H >>
        Data Ic( 106 )/ 157 /,Ic( 107 )/ 75  /,Ic( 108 )/ 76  /
        Data Ic( 109 )/ 3   /,Ic( 110 )/ 77  /,Ic( 111 )/ 36  /
        Data Ic( 112 )/ 34  /,Ic( 113 )/ 74  /,Ic( 114 )/ 30  /
        Data Ic( 115 )/ 29  /,Ic( 116 )/ 27  /,Ic( 117 )/ 65  /
        Data Ic( 118 )/ 64  /,Ic( 119 )/ 158 /,Ic( 120 )/ 192 /
C------<< 73 I >>
        Data Ic( 121 )/ 78  /,Ic( 122 )/ 11  /,Ic( 123 )/ 10  /
        Data Ic( 124 )/ 44  /,Ic( 125 )/ 9   /,Ic( 126 )/ 31  /
        Data Ic( 127 )/ 34  /,Ic( 128 )/ 53  /,Ic( 129 )/ 47  /
        Data Ic( 130 )/ 145 /,Ic( 131 )/ 22  /,Ic( 132 )/ 72  /
        Data Ic( 133 )/ 79  /,Ic( 134 )/ 25  /,Ic( 135 )/ 27  /
        Data Ic( 136 )/ 159 /,Ic( 137 )/ 185 /,Ic( 138 )/ 48  /
```

```
      Data Ic( 139 )/ 193 /
C-----<< 74 J >>
      Data Ic( 140 )/ 80  /,Ic( 141 )/ 10  /,Ic( 142 )/ 81  /
      Data Ic( 143 )/ 82  /,Ic( 144 )/ 83  /,Ic( 145 )/ 33  /
      Data Ic( 146 )/ 195 /,Ic( 147 )/ 50  /,Ic( 148 )/ 22  /
      Data Ic( 149 )/ 72  /,Ic( 150 )/ 84  /,Ic( 151 )/ 64  /
      Data Ic( 152 )/ 150 /,Ic( 153 )/ 194 /,Ic( 154 )/ 35  /
      Data Ic( 155 )/ 36  /,Ic( 156 )/ 77  /,Ic( 157 )/ 85  /
C-----<< 75 K >>
      Data Ic( 158 )/ 157 /,Ic( 159 )/ 86  /,Ic( 160 )/ 161 /
      Data Ic( 161 )/ 29  /,Ic( 162 )/ 28  /,Ic( 163 )/ 27  /
      Data Ic( 164 )/ 26  /,Ic( 165 )/ 27  /,Ic( 166 )/ 159 /
      Data Ic( 167 )/ 1   /,Ic( 168 )/ 47  /,Ic( 169 )/ 15  /
      Data Ic( 170 )/ 87  /,Ic( 171 )/ 15  /,Ic( 172 )/ 162 /
C-----<< 76 L >>
      Data Ic( 173 )/ 88  /,Ic( 174 )/ 47  /,Ic( 175 )/ 89  /
      Data Ic( 176 )/ 77  /,Ic( 177 )/ 36  /,Ic( 178 )/ 34  /
      Data Ic( 179 )/ 31  /,Ic( 180 )/ 44  /,Ic( 181 )/ 2   /
      Data Ic( 182 )/ 13  /,Ic( 183 )/ 25  /,Ic( 184 )/ 72  /
      Data Ic( 185 )/ 72  /,Ic( 186 )/ 151 /,Ic( 187 )/ 17  /
      Data Ic( 188 )/ 18  /,Ic( 189 )/ 47  /,Ic( 190 )/ 183 /
C-----<< 77 M >>
      Data Ic( 191 )/ 90  /,Ic( 192 )/ 163 /,Ic( 193 )/ 14  /
      Data Ic( 194 )/ 72  /,Ic( 195 )/ 189 /,Ic( 196 )/ 3   /
      Data Ic( 197 )/ 41  /,Ic( 198 )/ 179 /,Ic( 199 )/ 14  /
      Data Ic( 200 )/ 72  /,Ic( 201 )/ 189 /,Ic( 202 )/ 3   /
      Data Ic( 203 )/ 41  /,Ic( 204 )/ 179 /,Ic( 205 )/ 14  /
      Data Ic( 206 )/ 72  /,Ic( 207 )/ 63  /,Ic( 208 )/ 22  /
      Data Ic( 209 )/ 162 /
C-----<< 78 N >>
      Data Ic( 210 )/ 93  /,Ic( 211 )/ 163 /,Ic( 212 )/ 14  /
      Data Ic( 213 )/ 72  /,Ic( 214 )/ 189 /,Ic( 215 )/ 160 /
      Data Ic( 216 )/ 179 /,Ic( 217 )/ 14  /,Ic( 218 )/ 72  /
      Data Ic( 219 )/ 63  /,Ic( 220 )/ 22  /,Ic( 221 )/ 162 /
C-----<< 79 O >>
      Data Ic( 222 )/ 51  /,Ic( 223 )/ 40  /,Ic( 224 )/ 148 /
      Data Ic( 225 )/ 164 /,Ic( 226 )/ 69  /,Ic( 227 )/ 153 /
      Data Ic( 228 )/ 2   /,Ic( 229 )/ 27  /,Ic( 230 )/ 22  /
      Data Ic( 231 )/ 50  /,Ic( 232 )/ 56  /,Ic( 233 )/ 181 /
      Data Ic( 234 )/ 39  /
C-----<< 80 P >>
      Data Ic( 235 )/ 143 /,Ic( 236 )/ 19  /,Ic( 237 )/ 39  /
      Data Ic( 238 )/ 146 /,Ic( 239 )/ 43  /,Ic( 240 )/ 30  /
C-----<< 81 Q >>
      Data Ic( 241 )/ 165 /,Ic( 242 )/ 26  /,Ic( 243 )/ 95  /
      Data Ic( 244 )/ 96  /,Ic( 245 )/ 28  /,Ic( 246 )/ 40  /
      Data Ic( 247 )/ 2   /,Ic( 248 )/ 30  /,Ic( 249 )/ 48  /
      Data Ic( 250 )/ 193 /,Ic( 251 )/ 17  /,Ic( 252 )/ 152 /
C-----<< 82 R >>
      Data Ic( 253 )/ 143 /,Ic( 254 )/ 70  /,Ic( 255 )/ 39  /
      Data Ic( 256 )/ 146 /,Ic( 257 )/ 58  /,Ic( 258 )/ 145 /
```

```
      Data Ic( 259 )/ 97  /,Ic( 260 )/ 15  /,Ic( 261 )/ 39  /
      Data Ic( 262 )/ 192 /
C------<< 83 S >>
      Data Ic( 263 )/ 29  /,Ic( 264 )/ 166 /,Ic( 265 )/ 5   /
      Data Ic( 266 )/ 35  /,Ic( 267 )/ 54  /,Ic( 268 )/ 33  /
      Data Ic( 269 )/ 155 /,Ic( 270 )/ 25  /,Ic( 271 )/ 142 /
      Data Ic( 272 )/ 181 /,Ic( 273 )/ 16  /,Ic( 274 )/ 15  /
      Data Ic( 275 )/ 144 /,Ic( 276 )/ 13  /,Ic( 277 )/ 156 /
C------<< 84 T >>
      Data Ic( 278 )/ 154 /
C------<< 85 U >>
      Data Ic( 279 )/ 93  /,Ic( 280 )/ 167 /,Ic( 281 )/ 142 /
      Data Ic( 282 )/ 182 /,Ic( 283 )/ 186 /,Ic( 284 )/ 54  /
      Data Ic( 285 )/ 92  /,Ic( 286 )/ 91  /,Ic( 287 )/ 1   /
      Data Ic( 288 )/ 63  /,Ic( 289 )/ 23  /,Ic( 290 )/ 52  /
      Data Ic( 291 )/ 22  /,Ic( 292 )/ 162 /
C------<< 86 V >>
      Data Ic( 293 )/ 98  /,Ic( 294 )/ 167 /,Ic( 295 )/ 178 /
      Data Ic( 296 )/ 47  /,Ic( 297 )/ 3   /,Ic( 298 )/ 4   /
      Data Ic( 299 )/ 35  /,Ic( 300 )/ 54  /,Ic( 301 )/ 33  /
      Data Ic( 302 )/ 69  /,Ic( 303 )/ 161 /,Ic( 304 )/ 29  /
      Data Ic( 305 )/ 13  /,Ic( 306 )/ 25  /,Ic( 307 )/ 14  /
      Data Ic( 308 )/ 50  /,Ic( 309 )/ 16  /,Ic( 310 )/ 39  /
C------<< 87 W >>
      Data Ic( 311 )/ 99  /,Ic( 312 )/ 163 /,Ic( 313 )/ 22  /
      Data Ic( 314 )/ 100 /,Ic( 315 )/ 101 /,Ic( 316 )/ 102 /
      Data Ic( 317 )/ 103 /,Ic( 318 )/ 7   /,Ic( 319 )/ 5   /
      Data Ic( 320 )/ 4   /,Ic( 321 )/ 3   /
C------<< 88 X >>
      Data Ic( 322 )/ 104 /,Ic( 323 )/ 2   /,Ic( 324 )/ 30  /
      Data Ic( 325 )/ 48  /,Ic( 326 )/ 34  /,Ic( 327 )/ 160 /
      Data Ic( 328 )/ 179 /,Ic( 329 )/ 22  /,Ic( 330 )/ 105 /
      Data Ic( 331 )/ 22  /,Ic( 332 )/ 15  /,Ic( 333 )/ 39  /
      Data Ic( 334 )/ 47  /,Ic( 335 )/ 147 /,Ic( 336 )/ 48  /
      Data Ic( 337 )/ 30  /,Ic( 338 )/ 2   /,Ic( 339 )/ 1   /
      Data Ic( 340 )/ 106 /,Ic( 341 )/ 161 /,Ic( 342 )/ 2   /
      Data Ic( 343 )/ 28  /,Ic( 344 )/ 27  /,Ic( 345 )/ 26  /
      Data Ic( 346 )/ 107 /,Ic( 347 )/ 26  /,Ic( 348 )/ 27  /
      Data Ic( 349 )/ 159 /,Ic( 350 )/ 184 /
C------<< 89 Y >>
      Data Ic( 351 )/ 98  /,Ic( 352 )/ 167 /,Ic( 353 )/ 142 /
      Data Ic( 354 )/ 168 /,Ic( 355 )/ 166 /,Ic( 356 )/ 164 /
      Data Ic( 357 )/ 108 /,Ic( 358 )/ 1   /,Ic( 359 )/ 91  /
      Data Ic( 360 )/ 63  /,Ic( 361 )/ 109 /,Ic( 362 )/ 64  /
      Data Ic( 363 )/ 79  /,Ic( 364 )/ 65  /,Ic( 365 )/ 194 /
      Data Ic( 366 )/ 35  /,Ic( 367 )/ 180 /,Ic( 368 )/ 110 /
C------<< 90 Z >>
      Data Ic( 369 )/ 165 /,Ic( 370 )/ 65  /,Ic( 371 )/ 111 /
      Data Ic( 372 )/ 112 /,Ic( 373 )/ 28  /,Ic( 374 )/ 42  /
      Data Ic( 375 )/ 184 /,Ic( 376 )/ 53  /,Ic( 377 )/ 38  /
      Data Ic( 378 )/ 39  /,Ic( 379 )/ 145 /,Ic( 380 )/ 22  /
```

```
      Data Ic( 381 )/ 150 /,Ic( 382 )/ 26  /,Ic( 383 )/ 194 /
      Data Ic( 384 )/ 35  /,Ic( 385 )/ 180 /,Ic( 386 )/ 113 /
C-----<< 97 A >>
      Data Ic( 387 )/ 169 /,Ic( 388 )/ 182 /,Ic( 389 )/ 147 /
      Data Ic( 390 )/ 108 /,Ic( 391 )/ 158 /,Ic( 392 )/ 192 /
C-----<< 98 B >>
      Data Ic( 393 )/ 115 /,Ic( 394 )/ 190 /,Ic( 395 )/ 149 /
      Data Ic( 396 )/ 116 /,Ic( 397 )/ 71  /,Ic( 398 )/ 21  /
      Data Ic( 399 )/ 41  /,Ic( 400 )/ 3   /,Ic( 401 )/ 36  /
      Data Ic( 402 )/ 170 /
C-----<< 99 C >>
      Data Ic( 403 )/ 118 /,Ic( 404 )/ 74  /,Ic( 405 )/ 30  /
      Data Ic( 406 )/ 187 /,Ic( 407 )/ 39  /,Ic( 408 )/ 38  /
      Data Ic( 409 )/ 77  /,Ic( 410 )/ 35  /
C-----<<100 D >>
      Data Ic( 411 )/ 169 /,Ic( 412 )/ 182 /,Ic( 413 )/ 147 /
      Data Ic( 414 )/ 119 /,Ic( 415 )/ 1   /,Ic( 416 )/ 120 /
      Data Ic( 417 )/ 158 /,Ic( 418 )/ 192 /
C-----<<101 E >>
      Data Ic( 419 )/ 121 /,Ic( 420 )/ 3   /,Ic( 421 )/ 53  /
      Data Ic( 422 )/ 34  /,Ic( 423 )/ 161 /,Ic( 424 )/ 29  /
      Data Ic( 425 )/ 28  /,Ic( 426 )/ 25  /,Ic( 427 )/ 22  /
      Data Ic( 428 )/ 15  /,Ic( 429 )/ 39  /,Ic( 430 )/ 47  /
      Data Ic( 431 )/ 192 /
C-----<<102 F >>
      Data Ic( 432 )/ 122 /,Ic( 433 )/ 115 /,Ic( 434 )/ 164 /
      Data Ic( 435 )/ 33  /,Ic( 436 )/ 161 /,Ic( 437 )/ 28  /
      Data Ic( 438 )/ 25  /,Ic( 439 )/ 123 /,Ic( 440 )/ 84  /
      Data Ic( 441 )/ 171 /,Ic( 442 )/ 125 /,Ic( 443 )/ 168 /
      Data Ic( 444 )/ 192 /
C-----<<103 G >>
      Data Ic( 445 )/ 169 /,Ic( 446 )/ 182 /,Ic( 447 )/ 186 /
      Data Ic( 448 )/ 1   /,Ic( 449 )/ 91  /,Ic( 450 )/ 63  /
      Data Ic( 451 )/ 126 /,Ic( 452 )/ 172 /
C-----<<104 H >>
      Data Ic( 453 )/ 173 /,Ic( 454 )/ 188 /,Ic( 455 )/ 162 /
C-----<<105 I >>
      Data Ic( 456 )/ 174 /,Ic( 457 )/ 175 /,Ic( 458 )/ 162 /
C-----<<106 J >>
      Data Ic( 459 )/ 174 /,Ic( 460 )/ 129 /,Ic( 461 )/ 172 /
C-----<<107 K >>
      Data Ic( 462 )/ 173 /,Ic( 463 )/ 28  /,Ic( 464 )/ 42  /
      Data Ic( 465 )/ 39  /,Ic( 466 )/ 50  /,Ic( 467 )/ 162 /
C-----<<108 L >>
      Data Ic( 468 )/ 122 /,Ic( 469 )/ 190 /,Ic( 470 )/ 149 /
      Data Ic( 471 )/ 116 /,Ic( 472 )/ 71  /,Ic( 473 )/ 162 /
C-----<<109 M >>
      Data Ic( 474 )/ 130 /,Ic( 475 )/ 176 /,Ic( 476 )/ 196 /
      Data Ic( 477 )/ 14  /,Ic( 478 )/ 188 /,Ic( 479 )/ 162 /
C-----<<110 N >>
      Data Ic( 480 )/ 32  /,Ic( 481 )/ 176 /,Ic( 482 )/ 14  /
```

```
      Data Ic( 483 )/ 188 /,Ic( 484 )/ 162 /
C------<<111 O >>
      Data Ic( 485 )/ 131 /,Ic( 486 )/ 187 /,Ic( 487 )/ 182 /
      Data Ic( 488 )/ 3   /,Ic( 489 )/ 53  /,Ic( 490 )/ 34  /
      Data Ic( 491 )/ 48  /,Ic( 492 )/ 44  /,Ic( 493 )/ 43  /
      Data Ic( 494 )/ 13  /,Ic( 495 )/ 142 /,Ic( 496 )/ 39  /
      Data Ic( 497 )/ 38  /,Ic( 498 )/ 186 /
C------<<112 P >>
      Data Ic( 499 )/ 132 /,Ic( 500 )/ 164 /,Ic( 501 )/ 23  /
      Data Ic( 502 )/ 129 /,Ic( 503 )/ 1   /,Ic( 504 )/ 119 /
      Data Ic( 505 )/ 34  /,Ic( 506 )/ 160 /,Ic( 507 )/ 179 /
      Data Ic( 508 )/ 144 /,Ic( 509 )/ 13  /,Ic( 510 )/ 28  /
      Data Ic( 511 )/ 1   /,Ic( 512 )/ 133 /,Ic( 513 )/ 39  /
      Data Ic( 514 )/ 38  /,Ic( 515 )/ 37  /,Ic( 516 )/ 3   /
      Data Ic( 517 )/ 4   /
C------<<113 Q >>
      Data Ic( 518 )/ 169 /,Ic( 519 )/ 182 /,Ic( 520 )/ 3   /
      Data Ic( 521 )/ 1   /,Ic( 522 )/ 92  /,Ic( 523 )/ 24  /
      Data Ic( 524 )/ 79  /,Ic( 525 )/ 171 /,Ic( 526 )/ 134 /
      Data Ic( 527 )/ 3   /,Ic( 528 )/ 77  /,Ic( 529 )/ 4   /
C------<<114 R >>
      Data Ic( 530 )/ 115 /,Ic( 531 )/ 164 /,Ic( 532 )/ 14  /
      Data Ic( 533 )/ 38  /,Ic( 534 )/ 21  /,Ic( 535 )/ 14  /
      Data Ic( 536 )/ 24  /,Ic( 537 )/ 46  /,Ic( 538 )/ 21  /
      Data Ic( 539 )/ 41  /,Ic( 540 )/ 192 /
C------<<115 S >>
      Data Ic( 541 )/ 135 /,Ic( 542 )/ 164 /,Ic( 543 )/ 14  /
      Data Ic( 544 )/ 56  /,Ic( 545 )/ 15  /,Ic( 546 )/ 14  /
      Data Ic( 547 )/ 28  /,Ic( 548 )/ 1   /,Ic( 549 )/ 133 /
      Data Ic( 550 )/ 39  /,Ic( 551 )/ 70  /,Ic( 552 )/ 192 /
C------<<116 T >>
      Data Ic( 553 )/ 122 /,Ic( 554 )/ 35  /,Ic( 555 )/ 54  /
      Data Ic( 556 )/ 1   /,Ic( 557 )/ 136 /,Ic( 558 )/ 129 /
      Data Ic( 559 )/ 14  /,Ic( 560 )/ 168 /,Ic( 561 )/ 192 /
      Data Ic( 562 )/ 1   /,Ic( 563 )/ 137 /,Ic( 564 )/ 138 /
C------<<117 U >>
      Data Ic( 565 )/ 177 /,Ic( 566 )/ 175 /,Ic( 567 )/ 162 /
C------<<118 V >>
      Data Ic( 568 )/ 68  /,Ic( 569 )/ 54  /,Ic( 570 )/ 158 /
      Data Ic( 571 )/ 183 /,Ic( 572 )/ 170 /
C------<<119 W >>
      Data Ic( 573 )/ 136 /,Ic( 574 )/ 27  /,Ic( 575 )/ 24  /
      Data Ic( 576 )/ 142 /,Ic( 577 )/ 47  /,Ic( 578 )/ 166 /
      Data Ic( 579 )/ 1   /,Ic( 580 )/ 108 /,Ic( 581 )/ 175 /
      Data Ic( 582 )/ 168 /,Ic( 583 )/ 166 /,Ic( 584 )/ 170 /
C------<<120 X >>
      Data Ic( 585 )/ 32  /,Ic( 586 )/ 35  /,Ic( 587 )/ 195 /
      Data Ic( 588 )/ 87  /,Ic( 589 )/ 21  /,Ic( 590 )/ 38  /
      Data Ic( 591 )/ 77  /,Ic( 592 )/ 35  /,Ic( 593 )/ 1   /
      Data Ic( 594 )/ 139 /,Ic( 595 )/ 30  /,Ic( 596 )/ 2   /
      Data Ic( 597 )/ 13  /,Ic( 598 )/ 140 /,Ic( 599 )/ 13  /
```

```
      Data Ic( 600 )/ 2   /,Ic( 601 )/ 30  /
C------<<121 Y >>
      Data Ic( 602 )/ 177 /,Ic( 603 )/ 129 /,Ic( 604 )/ 172 /
C------<<122 Z >>
      Data Ic( 605 )/ 68  /,Ic( 606 )/ 35  /,Ic( 607 )/ 160 /
      Data Ic( 608 )/ 16  /,Ic( 609 )/ 144 /,Ic( 610 )/ 27  /
      Data Ic( 611 )/ 40  /,Ic( 612 )/ 179 /,Ic( 613 )/ 22  /
      Data Ic( 614 )/ 150 /,Ic( 615 )/ 194 /,Ic( 616 )/ 180 /
      Data Ic( 617 )/ 141 /,Ic( 618 )/ 4   /
C------<<142   >>
      Data Ic( 619 )/ 14  /,Ic( 620 )/ 15  /
C------<<143   >>
      Data Ic( 621 )/ 20  /,Ic( 622 )/ 21  /,Ic( 623 )/ 22  /
      Data Ic( 624 )/ 23  /,Ic( 625 )/ 24  /,Ic( 626 )/ 25  /
      Data Ic( 627 )/ 26  /,Ic( 628 )/ 27  /,Ic( 629 )/ 28  /
      Data Ic( 630 )/ 29  /,Ic( 631 )/ 30  /,Ic( 632 )/ 31  /
      Data Ic( 633 )/ 32  /,Ic( 634 )/ 33  /,Ic( 635 )/ 34  /
      Data Ic( 636 )/ 35  /,Ic( 637 )/ 36  /,Ic( 638 )/ 3   /
      Data Ic( 639 )/ 37  /
C------<<144   >>
      Data Ic( 640 )/ 14  /,Ic( 641 )/ 25  /
C------<<145   >>
      Data Ic( 642 )/ 21  /,Ic( 643 )/ 15  /
C------<<146   >>
      Data Ic( 644 )/ 145 /,Ic( 645 )/ 22  /,Ic( 646 )/ 25  /
      Data Ic( 647 )/ 13  /,Ic( 648 )/ 28  /,Ic( 649 )/ 42  /
C------<<147   >>
      Data Ic( 650 )/ 3   /,Ic( 651 )/ 34  /
C------<<148   >>
      Data Ic( 652 )/ 27  /,Ic( 653 )/ 26  /,Ic( 654 )/ 25  /
      Data Ic( 655 )/ 23  /,Ic( 656 )/ 49  /,Ic( 657 )/ 50  /
      Data Ic( 658 )/ 21  /,Ic( 659 )/ 182 /,Ic( 660 )/ 183 /
C------<<149   >>
      Data Ic( 661 )/ 28  /,Ic( 662 )/ 25  /,Ic( 663 )/ 23  /
C------<<150   >>
      Data Ic( 664 )/ 24  /,Ic( 665 )/ 25  /
C------<<151   >>
      Data Ic( 666 )/ 150 /,Ic( 667 )/ 27  /,Ic( 668 )/ 28  /
      Data Ic( 669 )/ 2   /,Ic( 670 )/ 184 /,Ic( 671 )/ 193 /
C------<<152   >>
      Data Ic( 672 )/ 18  /,Ic( 673 )/ 38  /,Ic( 674 )/ 183 /
C------<<153   >>
      Data Ic( 675 )/ 55  /,Ic( 676 )/ 9   /,Ic( 677 )/ 30  /
      Data Ic( 678 )/ 43  /
C------<<154   >>
      Data Ic( 679 )/ 60  /,Ic( 680 )/ 2   /,Ic( 681 )/ 185 /
      Data Ic( 682 )/ 34  /,Ic( 683 )/ 37  /,Ic( 684 )/ 38  /
      Data Ic( 685 )/ 61  /,Ic( 686 )/ 38  /,Ic( 687 )/ 3   /
      Data Ic( 688 )/ 1   /,Ic( 689 )/ 62  /,Ic( 690 )/ 63  /
      Data Ic( 691 )/ 64  /,Ic( 692 )/ 65  /,Ic( 693 )/ 27  /
      Data Ic( 694 )/ 28  /,Ic( 695 )/ 2   /,Ic( 696 )/ 185 /
```

```
      Data Ic( 697 )/ 48  /,Ic( 698 )/ 193 /
C-----<<155   >>
      Data Ic( 699 )/ 31  /,Ic( 700 )/ 30  /,Ic( 701 )/ 29  /
      Data Ic( 702 )/ 13  /
C-----<<156   >>
      Data Ic( 703 )/ 40  /,Ic( 704 )/ 59  /,Ic( 705 )/ 58  /
      Data Ic( 706 )/ 11  /,Ic( 707 )/ 44  /,Ic( 708 )/ 48  /
C-----<<157   >>
      Data Ic( 709 )/ 73  /,Ic( 710 )/ 191 /,Ic( 711 )/ 39  /
      Data Ic( 712 )/ 15  /,Ic( 713 )/ 14  /,Ic( 714 )/ 23  /
      Data Ic( 715 )/ 64  /,Ic( 716 )/ 65  /,Ic( 717 )/ 27  /
      Data Ic( 718 )/ 2   /,Ic( 719 )/ 184 /,Ic( 720 )/ 1   /
C-----<<158   >>
      Data Ic( 721 )/ 72  /,Ic( 722 )/ 22  /,Ic( 723 )/ 21  /
      Data Ic( 724 )/ 41  /
C-----<<159   >>
      Data Ic( 725 )/ 28  /,Ic( 726 )/ 2   /
C-----<<160   >>
      Data Ic( 727 )/ 3   /,Ic( 728 )/ 47  /
C-----<<161   >>
      Data Ic( 729 )/ 48  /,Ic( 730 )/ 30  /
C-----<<162   >>
      Data Ic( 731 )/ 21  /,Ic( 732 )/ 41  /,Ic( 733 )/ 3   /
      Data Ic( 734 )/ 53  /,Ic( 735 )/ 35  /
C-----<<163   >>
      Data Ic( 736 )/ 191 /,Ic( 737 )/ 179 /
C-----<<164   >>
      Data Ic( 738 )/ 35  /,Ic( 739 )/ 34  /
C-----<<165   >>
      Data Ic( 740 )/ 94  /,Ic( 741 )/ 25  /,Ic( 742 )/ 13  /
      Data Ic( 743 )/ 159 /,Ic( 744 )/ 44  /,Ic( 745 )/ 48  /
      Data Ic( 746 )/ 33  /,Ic( 747 )/ 36  /,Ic( 748 )/ 37  /
      Data Ic( 749 )/ 38  /,Ic( 750 )/ 39  /,Ic( 751 )/ 15  /
      Data Ic( 752 )/ 49  /,Ic( 753 )/ 24  /
C-----<<166   >>
      Data Ic( 754 )/ 3   /,Ic( 755 )/ 36  /
C-----<<167   >>
      Data Ic( 756 )/ 163 /,Ic( 757 )/ 14  /,Ic( 758 )/ 23  /
      Data Ic( 759 )/ 24  /,Ic( 760 )/ 23  /
C-----<<168   >>
      Data Ic( 761 )/ 21  /,Ic( 762 )/ 47  /
C-----<<169   >>
      Data Ic( 763 )/ 114 /,Ic( 764 )/ 44  /,Ic( 765 )/ 43  /
      Data Ic( 766 )/ 187 /
C-----<<170   >>
      Data Ic( 767 )/ 33  /,Ic( 768 )/ 31  /,Ic( 769 )/ 117 /
      Data Ic( 770 )/ 21  /,Ic( 771 )/ 47  /,Ic( 772 )/ 3   /
C-----<<171   >>
      Data Ic( 773 )/ 124 /,Ic( 774 )/ 188 /,Ic( 775 )/ 21  /
      Data Ic( 776 )/ 3   /,Ic( 777 )/ 33  /
C-----<<172   >>
```

```
      Data Ic( 778 )/ 25  /,Ic( 779 )/ 194 /,Ic( 780 )/ 180 /
      Data Ic( 781 )/ 3   /,Ic( 782 )/ 77  /,Ic( 783 )/ 4   /
C-----<<173   >>
      Data Ic( 784 )/ 115 /,Ic( 785 )/ 190 /,Ic( 786 )/ 149 /
      Data Ic( 787 )/ 52  /,Ic( 788 )/ 127 /,Ic( 789 )/ 33  /
      Data Ic( 790 )/ 34  /,Ic( 791 )/ 35  /,Ic( 792 )/ 195 /
      Data Ic( 793 )/ 14  /
C-----<<174   >>
      Data Ic( 794 )/ 128 /,Ic( 795 )/ 46  /,Ic( 796 )/ 41  /
      Data Ic( 797 )/ 74  /,Ic( 798 )/ 29  /,Ic( 799 )/ 1   /
      Data Ic( 800 )/ 84  /,Ic( 801 )/ 54  /
C-----<<175   >>
      Data Ic( 802 )/ 64  /,Ic( 803 )/ 14  /
C-----<<176   >>
      Data Ic( 804 )/ 35  /,Ic( 805 )/ 3   /,Ic( 806 )/ 21  /
      Data Ic( 807 )/ 196 /
C-----<<177   >>
      Data Ic( 808 )/ 68  /,Ic( 809 )/ 54  /,Ic( 810 )/ 175 /
      Data Ic( 811 )/ 168 /,Ic( 812 )/ 166 /,Ic( 813 )/ 35  /
      Data Ic( 814 )/ 1   /,Ic( 815 )/ 33  /
C-----<<178   >>
      Data Ic( 816 )/ 22  /,Ic( 817 )/ 15  /
C-----<<179   >>
      Data Ic( 818 )/ 39  /,Ic( 819 )/ 15  /
C-----<<180   >>
      Data Ic( 820 )/ 4   /,Ic( 821 )/ 77  /
C-----<<181   >>
      Data Ic( 822 )/ 16  /,Ic( 823 )/ 17  /
C-----<<182   >>
      Data Ic( 824 )/ 39  /,Ic( 825 )/ 47  /
C-----<<183   >>
      Data Ic( 826 )/ 37  /,Ic( 827 )/ 36  /
C-----<<184   >>
      Data Ic( 828 )/ 30  /,Ic( 829 )/ 48  /
C-----<<185   >>
      Data Ic( 830 )/ 43  /,Ic( 831 )/ 44  /
C-----<<186   >>
      Data Ic( 832 )/ 3   /,Ic( 833 )/ 53  /
C-----<<187   >>
      Data Ic( 834 )/ 2   /,Ic( 835 )/ 28  /,Ic( 836 )/ 13  /
      Data Ic( 837 )/ 25  /,Ic( 838 )/ 142 /
C-----<<188   >>
      Data Ic( 839 )/ 24  /,Ic( 840 )/ 14  /
C-----<<189   >>
      Data Ic( 841 )/ 23  /,Ic( 842 )/ 63  /,Ic( 843 )/ 1   /
      Data Ic( 844 )/ 91  /,Ic( 845 )/ 92  /,Ic( 846 )/ 54  /
      Data Ic( 847 )/ 53  /
C-----<<190   >>
      Data Ic( 848 )/ 35  /,Ic( 849 )/ 68  /,Ic( 850 )/ 34  /
      Data Ic( 851 )/ 33  /,Ic( 852 )/ 161 /
C-----<<191   >>
```

```
        Data Ic( 853 )/ 185 /,Ic( 854 )/ 74  /,Ic( 855 )/ 34  /
        Data Ic( 856 )/ 3   /,Ic( 857 )/ 41  /
C------<<192   >>
        Data Ic( 858 )/ 186 /,Ic( 859 )/ 35  /
C------<<193   >>
        Data Ic( 860 )/ 53  /,Ic( 861 )/ 47  /,Ic( 862 )/ 39  /
C------<<194   >>
        Data Ic( 863 )/ 28  /,Ic( 864 )/ 184 /,Ic( 865 )/ 33  /
C------<<195   >>
        Data Ic( 866 )/ 160 /,Ic( 867 )/ 21  /
C------<<196   >>
        Data Ic( 868 )/ 46  /,Ic( 869 )/ 23  /,Ic( 870 )/ 24  /
        Data Ic( 871 )/ 1   /,Ic( 872 )/ 33  /,Ic( 873 )/ 34  /
        Data Ic( 874 )/ 35  /,Ic( 875 )/ 195 /,Ic( 876 )/ 0   /
C
C     ******************************************************
C     *           CHARACTER INDEX TABLE                    *
C     ******************************************************
C
        Data In( 1  )/ 0  /,In( 2  )/ 17 /,In( 3  )/ 32 /
        Data In( 4  )/ 44 /,In( 5  )/ 60 /,In( 6  )/ 82 /
        Data In( 7  )/ 86 /,In( 8  )/ 105 /,In( 9  )/ 120 /
        Data In( 10 )/ 139 /,In( 11 )/ 157 /,In( 12 )/ 172 /
        Data In( 13 )/ 190 /,In( 14 )/ 209 /,In( 15 )/ 221 /
        Data In( 16 )/ 234 /,In( 17 )/ 240 /,In( 18 )/ 252 /
        Data In( 19 )/ 262 /,In( 20 )/ 277 /,In( 21 )/ 278 /
        Data In( 22 )/ 292 /,In( 23 )/ 310 /,In( 24 )/ 321 /
        Data In( 25 )/ 350 /,In( 26 )/ 368 /,In( 27 )/ 386 /
        Data In( 28 )/ 386 /,In( 29 )/ 386 /,In( 30 )/ 386 /
        Data In( 31 )/ 386 /,In( 32 )/ 386 /,In( 33 )/ 386 /
        Data In( 34 )/ 392 /,In( 35 )/ 402 /,In( 36 )/ 410 /
        Data In( 37 )/ 418 /,In( 38 )/ 431 /,In( 39 )/ 444 /
        Data In( 40 )/ 452 /,In( 41 )/ 455 /,In( 42 )/ 458 /
        Data In( 43 )/ 461 /,In( 44 )/ 467 /,In( 45 )/ 473 /
        Data In( 46 )/ 479 /,In( 47 )/ 484 /,In( 48 )/ 498 /
        Data In( 49 )/ 517 /,In( 50 )/ 529 /,In( 51 )/ 540 /
        Data In( 52 )/ 552 /,In( 53 )/ 564 /,In( 54 )/ 567 /
        Data In( 55 )/ 572 /,In( 56 )/ 584 /,In( 57 )/ 601 /
        Data In( 58 )/ 604 /,In( 59 )/ 618 /,In( 60 )/ 620 /
        Data In( 61 )/ 639 /,In( 62 )/ 641 /,In( 63 )/ 643 /
        Data In( 64 )/ 649 /,In( 65 )/ 651 /,In( 66 )/ 660 /
        Data In( 67 )/ 663 /,In( 68 )/ 665 /,In( 69 )/ 671 /
        Data In( 70 )/ 674 /,In( 71 )/ 678 /,In( 72 )/ 698 /
        Data In( 73 )/ 702 /,In( 74 )/ 708 /,In( 75 )/ 720 /
        Data In( 76 )/ 724 /,In( 77 )/ 726 /,In( 78 )/ 728 /
        Data In( 79 )/ 730 /,In( 80 )/ 735 /,In( 81 )/ 737 /
        Data In( 82 )/ 739 /,In( 83 )/ 753 /,In( 84 )/ 755 /
        Data In( 85 )/ 760 /,In( 86 )/ 762 /,In( 87 )/ 766 /
        Data In( 88 )/ 772 /,In( 89 )/ 777 /,In( 90 )/ 783 /
        Data In( 91 )/ 793 /,In( 92 )/ 801 /,In( 93 )/ 803 /
        Data In( 94 )/ 807 /,In( 95 )/ 815 /,In( 96 )/ 817 /
```

```
      Data In( 97 )/ 819 /,In( 98 )/ 821 /,In( 99 )/ 823 /
      Data In( 100 )/ 825 /,In( 101 )/ 827 /,In( 102 )/ 829 /
      Data In( 103 )/ 831 /,In( 104 )/ 833 /,In( 105 )/ 838 /
      Data In( 106 )/ 840 /,In( 107 )/ 847 /,In( 108 )/ 852 /
      Data In( 109 )/ 857 /,In( 110 )/ 859 /,In( 111 )/ 862 /
      Data In( 112 )/ 865 /,In( 113 )/ 867 /,In( 114 )/ 875 /
C
C     ********************************************************
C     *       BASIC ELEMENTS TABLE                           *
C     ********************************************************
C
      Data Ie( 1 )/-2  /,Ie( 2  )/ 0  /,Ie( 3  )/ 2  /
      Data Ie( 4  )/ 1  /,Ie( 5  )/ 3  /,Ie( 6  )/ 3  /
      Data Ie( 7  )/ 3  /,Ie( 8  )/ 4  /,Ie( 9  )/ 4  /
      Data Ie( 10 )/ 7  /,Ie( 11 )/ 3  /,Ie( 12 )/ 6  /
      Data Ie( 13 )/ 0  /,Ie( 14 )/-21 /,Ie( 15 )/-1  /
      Data Ie( 16 )/ 3  /,Ie( 17 )/-2  /,Ie( 18 )/ 3  /
      Data Ie( 19 )/-2  /,Ie( 20 )/ 2  /,Ie( 21 )/-3  /
      Data Ie( 22 )/ 2  /,Ie( 23 )/-1  /,Ie( 24 )/-1  /
      Data Ie( 25 )/ 0  /,Ie( 26 )/-2  /,Ie( 27 )/ 1  /
      Data Ie( 28 )/-2  /,Ie( 29 )/ 2  /,Ie( 30 )/-2  /
      Data Ie( 31 )/ 3  /,Ie( 32 )/-2  /,Ie( 33 )/ 3  /
      Data Ie( 34 )/-1  /,Ie( 35 )/ 5  /,Ie( 36 )/ 0  /
      Data Ie( 37 )/ 10 /,Ie( 38 )/ 19 /,Ie( 39 )/ 1  /
      Data Ie( 40 )/-1  /,Ie( 41 )/ 0  /,Ie( 42 )/-3  /
      Data Ie( 43 )/-1  /,Ie( 44 )/-4  /,Ie( 45 )/-1  /
      Data Ie( 46 )/-3  /,Ie( 47 )/-1  /,Ie( 48 )/-2  /
      Data Ie( 49 )/-2  /,Ie( 50 )/-3  /,Ie( 51 )/-2  /
      Data Ie( 52 )/-2  /,Ie( 53 )/-2  /,Ie( 54 )/-1  /
      Data Ie( 55 )/-1  /,Ie( 56 )/ 0  /,Ie( 57 )/-1  /
      Data Ie( 58 )/ 1  /,Ie( 59 )/ 0  /,Ie( 60 )/ 3  /
      Data Ie( 61 )/ 1  /,Ie( 62 )/ 5  /,Ie( 63 )/ 1  /
      Data Ie( 64 )/ 3  /,Ie( 65 )/ 1  /,Ie( 66 )/ 2  /
      Data Ie( 67 )/ 2  /,Ie( 68 )/ 3  /,Ie( 69 )/ 2  /
      Data Ie( 70 )/ 2  /,Ie( 71 )/ 3  /,Ie( 72 )/ 1  /
      Data Ie( 73 )/ 3  /,Ie( 74 )/ 0  /,Ie( 75 )/ 2  /
      Data Ie( 76 )/-1  /,Ie( 77 )/-3  /,Ie( 78 )/-1  /
      Data Ie( 79 )/ 1  /,Ie( 80 )/ 0  /,Ie( 81 )/-3  /
      Data Ie( 82 )/ 0  /,Ie( 83 )/-2  /,Ie( 84 )/ 1  /
      Data Ie( 85 )/-1  /,Ie( 86 )/ 2  /,Ie( 87 )/ 11 /
      Data Ie( 88 )/ 15 /,Ie( 89 )/ 0  /,Ie( 90 )/-1  /
      Data Ie( 91 )/ 2  /,Ie( 92 )/ 0  /,Ie( 93 )/ 0  /
      Data Ie( 94 )/ 2  /,Ie( 95 )/ 0  /,Ie( 96 )/-4  /
      Data Ie( 97 )/ 1  /,Ie( 98 )/-3  /,Ie( 99 )/ 11 /
      Data Ie( 100 )/ 21 /,Ie( 101 )/-1  /,Ie( 102 )/-6  /
      Data Ie( 103 )/ 1  /,Ie( 104 )/ 1  /,Ie( 105 )/ 2  /
      Data Ie( 106 )/ 4  /,Ie( 107 )/ 0  /,Ie( 108 )/ 4  /
      Data Ie( 109 )/ 2  /,Ie( 110 )/-3  /,Ie( 111 )/ 13 /
      Data Ie( 112 )/ 17 /,Ie( 113 )/-3  /,Ie( 114 )/ 1  /
      Data Ie( 115 )/-4  /,Ie( 116 )/ 0  /,Ie( 117 )/ 9  /
      Data Ie( 118 )/ 15 /,Ie( 119 )/ 4  /,Ie( 120 )/-1  /
```

```
Data Ie( 121 )/-5  /,Ie( 122 )/-1  /,Ie( 123 )/-2  /
Data Ie( 124 )/-7  /,Ie( 125 )/-2  /,Ie( 126 )/-6  /
Data Ie( 127 )/-2  /,Ie( 128 )/-4  /,Ie( 129 )/ 9  /
Data Ie( 130 )/ 0  /,Ie( 131 )/ 4  /,Ie( 132 )/ 4  /
Data Ie( 133 )/ 3  /,Ie( 134 )/ 5  /,Ie( 135 )/ 1  /
Data Ie( 136 )/ 4  /,Ie( 137 )/ 4  /,Ie( 138 )/ 0  /
Data Ie( 139 )/ 0  /,Ie( 140 )/-6  /,Ie( 141 )/-1  /
Data Ie( 142 )/-5  /,Ie( 143 )/ 4  /,Ie( 144 )/ 14 /
Data Ie( 145 )/ 0  /,Ie( 146 )/ 1  /,Ie( 147 )/ 6  /
Data Ie( 148 )/ 6  /,Ie( 149 )/ 9  /,Ie( 150 )/ 3  /
Data Ie( 151 )/ 3  /,Ie( 152 )/ 2  /,Ie( 153 )/ 14 /
Data Ie( 154 )/ 5  /,Ie( 155 )/-2  /,Ie( 156 )/-5  /
Data Ie( 157 )/ 11 /,Ie( 158 )/-3  /,Ie( 159 )/-2  /
Data Ie( 160 )/ 5  /,Ie( 161 )/-1  /,Ie( 162 )/ 6  /
Data Ie( 163 )/ 0  /,Ie( 164 )/ 6  /,Ie( 165 )/-3  /
Data Ie( 166 )/-9  /,Ie( 167  )/ 4  /,Ie( 168  )/ 2  /
Data Ie( 169 )/ 23 /,Ie( 170 )/ 15 /,Ie( 171 )/ 0  /
Data Ie( 172 )/-7  /,Ie( 173 )/ 4  /,Ie( 174 )/ 9  /
Data Ie( 175 )/ 4  /,Ie( 176 )/ 1  /,Ie( 177 )/-4  /
Data Ie( 178 )/ 14 /,Ie( 179 )/ 2  /,Ie( 180 )/ 7  /
Data Ie( 181 )/ 3  /,Ie( 182 )/ 8  /,Ie( 183 )/ 1  /
Data Ie( 184 )/ 14 /,Ie( 185 )/ 12 /,Ie( 186 )/ 15 /
Data Ie( 187 )/-4  /,Ie( 188 )/-4  /,Ie( 189 )/-3  /
Data Ie( 190 )/-2  /,Ie( 191 )/ 0  /,Ie( 192 )/-5  /
Data Ie( 193 )/ 2  /,Ie( 194 )/ 14 /,Ie( 195 )/-1  /
Data Ie( 196 )/ 14 /,Ie( 197 )/-1  /,Ie( 198 )/-15 /
Data Ie( 199 )/ 10 /,Ie( 200 )/ 21 /,Ie( 201 )/-2  /
Data Ie( 202 )/-21 /,Ie( 203 )/ 3  /,Ie( 204 )/ 7  /
Data Ie( 205 )/ 5  /,Ie( 206 )/ 15 /,Ie( 207 )/-2  /
Data Ie( 208 )/-9  /,Ie( 209 )/ 7  /,Ie( 210 )/ 12 /
Data Ie( 211 )/-4  /,Ie( 212 )/-9  /,Ie( 213 )/ 2  /
Data Ie( 214 )/ 6  /,Ie( 215 )/-3  /,Ie( 216 )/-10 /
Data Ie( 217 )/ 5  /,Ie( 218 )/ 3  /,Ie( 219 )/-3  /
Data Ie( 220 )/-3  /,Ie( 221 )/-4  /,Ie( 222 )/-3  /
Data Ie( 223 )/ 6  /,Ie( 224 )/ 3  /,Ie( 225 )/ 12 /
Data Ie( 226 )/ 6  /,Ie( 227 )/ 4  /,Ie( 228 )/ 5  /
Data Ie( 229 )/-1  /,Ie( 230 )/-7  /,Ie( 231 )/ 1  /
Data Ie( 232 )/-4  /,Ie( 233 )/ 11 /,Ie( 234 )/ 7  /
Data Ie( 235 )/ 6  /,Ie( 236 )/ 18 /,Ie( 237 )/-4  /
Data Ie( 238 )/-12 /,Ie( 239 )/ 6  /,Ie( 240 )/ 2  /
Data Ie( 241 )/ 6  /,Ie( 242 )/ 5  /,Ie( 243 )/-2  /
Data Ie( 244 )/-8  /,Ie( 245 )/-3  /,Ie( 246 )/-7  /
Data Ie( 247 )/ 1  /,Ie( 248 )/ 9  /,Ie( 249 )/-4  /
Data Ie( 250 )/-11 /,Ie( 251 )/-1  /,Ie( 252 )/-8  /
Data Ie( 253 )/ 10 /,Ie( 254 )/ 14 /,Ie( 255 )/-6  /
Data Ie( 256 )/-18 /,Ie( 257 )/-4  /,Ie( 258 )/ 5  /
Data Ie( 259 )/ 9  /,Ie( 260 )/ 9  /,Ie( 261 )/ 2  /
Data Ie( 262 )/ 5  /,Ie( 263 )/-4  /,Ie( 264 )/ 1  /
Data Ie( 265 )/ 0  /,Ie( 266 )/ 7  /,Ie( 267 )/ 5  /
Data Ie( 268 )/ 5  /,Ie( 269 )/ 3  /,Ie( 270 )/ 9  /
Data Ie( 271 )/-8  /,Ie( 272 )/ 8  /,Ie( 273 )/ 7  /
```

```
      Data Ie( 274 )/ 0  /,Ie( 275 )/-3  /,Ie( 276 )/ 3  /
      Data Ie( 277 )/-4  /,Ie( 278 )/-7  /,Ie( 279 )/ 4  /
      Data Ie( 280 )/ 3  /
C
C
      Exit_Code = 0
      Icomp=142

      If (G_Char_Gen_State .Eq. 0) Then
            Ipen=3
            Iptr=0
      Else
            Ipen = 2
            Goto 16
      Endif

      Isym=Char
5     If ((Isym .Ge. 65) .And. (Isym .Le. 122)) Go To 10
      Isym=Isym-19
C
C OBTAIN INDEXES (BEGIN/END) TO THE CHARACTER TABLE
C
10    Ifirst=In(Isym-64)+1
      Ilast=In(Isym-64+1)
C
C LOOP THRU CHARACTER TABLE
C
15    I = Ifirst
16    If (I .Gt. Ilast) Goto 25
      J=Ic(I)
C
C IS THE CHARACTER TABLE ELEMENT A COMPOUND ?
C
      If (J .Ge. Icomp) Go To 30
C
C IS THE CHARACTER TABLE ELEMENT A PEN UP COMMAND ?
C
      If (J .Eq. 1) Go To 18
C
C OBTAIN THE X,Y BASIC ELEMENT PAIR FROM THE BASIC ELEMENTS TABLE
C
      Ym=Float(Ie(J*2+1-3))
      Xm=Float(Ie(J*2-3))
C
      Ypos = O_Char_Scale/21.0 * ( Ym*G_Cos_Crot + Xm*G_Sin_Crot )
      Xpos = O_Char_Aspect_Ratio*O_Char_Scale/21.0 *
     1       ( Xm*G_Cos_Crot - Ym*G_Sin_Crot )
C
      G_Char_Gen_State = Ipen
      I = I + 1
      Goto 45
```

```
18      Ipen = 3
        I = I + 1
        Goto 16
C
C ANY LOOP VARIABLES ON THE STACK ?
C
25      If (Iptr .Gt. 0) Go To 40

        G_Char_Gen_State = -1
        Go To 45
C
C DO WE NEED TO SAVE LOOP VARIABLES ?
C
30      If ( I .Eq. Ilast) Go To 35
        Index1=I+1
        Index2=Ilast
C
C PUT THE LOOP VARIABLES ON THE STACK
C
        Call Cgen_Push(Index1,Index2)
35      Isym=J
        Go To 5
C
C RETRIEVE THE LOOP VARIABLES FROM THE STACK
C
40      Call Cgen_Pop(Index1,Index2)
        Ifirst=Index1
        Ilast=Index2
        Go To 15
C
C RETURN INFO REGARDING THE FIRST MOVE ONLY
C
45      Exit_Code = G_Char_Gen_State

        Return
        End
```

```
        Subroutine Set_Pix(X,Y)

C  Sets a pixel within the raster stored in B_Band.

C  Parameters:

C  X, Y - i [i] = The column and row coordinates within the raster of the
C       point to be set.  Their index origin is 0.

        Implicit Integer (A-Z)

        Include 'Devices.Inc'
        Include 'Band.Inc'
        Include 'Options.Inc'

        Byte Bitpos(8)
        Data Bitpos/ '01'X, '02'X, '04'X, '08'X, '10'X, '20'X, '40'X, '80'X /

C  Ignore points outside the raster (band).

        If (Y .Lt. 0 .Or. Y .Gt. B_Band_Height-1 .Or.
     1  X .Lt. 0 .Or. X .Gt. B_Band_Width-1) Goto 999

C  Compute the number of the bit in the array to be set.  Depends on how
C  the raster is organized (row/column major).

        If (O_Row_Major) Then
                Bit_Numb = B_Band_Width * (B_Band_Height-1 - Y) + X
        Else
                Bit_Numb = B_Band_Height * X + (B_Band_Height-1 - Y)
        Endif

C  Compute the byte address within B_Band of the bit to be set, and the offset
C  of the bit within the byte.

        Byte_Address = Bit_Numb / D_Bits_Per_Byte
        Bit_Address = Bit_Numb - Byte_Address * D_Bits_Per_Byte

C  Set the appropriate bit.

        B_Band(Byte_Address+1) = B_Band(Byte_Address+1) .Or.
     1  Bitpos(Bit_Address+1)

999     Return
        End
```

```
      Subroutine Skip_Spaces

C  This routine skips blocks of spaces at the current position in the string
C  typed by the user.

      Implicit Integer (A-Z)

      Include 'Lexan.Inc'

      Do While (L_Inpline(L_Iptr:L_Iptr) .Eq. ' ' .And. L_Iptr .Le. L_Inplen)
            L_Iptr = L_Iptr + 1
      End Do

999   Return
      End
```

```
      Subroutine Space1(Ichar,Isp1,Isp2)

C  These character spacing tables contain two entries for each character.
C  These spacing entries describe two things: the width of the cell within
C  which each character sits (they are apparently not all the same size),
C  and the amount by which the left edge of each character is offset from
C  the left edge of the cell.  If we call the two numbers in the table for
C  each char A and B, A is apparently the first quantity described above,
C  and B-A is apparently the second.  This may be wrong.

C     ******************************************************
C     *         CHARACTER SPACING TABLE FOR STICK CHAR SET *
C     ******************************************************
        Dimension Is(184)
        Data Is( 1   )/ 4  /,Is( 2   )/ 14 /,Is( 3   )/ 1  /
        Data Is( 4   )/ 17 /,Is( 5   )/-1  /,Is( 6   )/ 20 /
        Data Is( 7   )/-1  /,Is( 8   )/ 19 /,Is( 9   )/-3  /
        Data Is( 10  )/ 21 /,Is( 11  )/-4  /,Is( 12  )/ 22 /
        Data Is( 13  )/ 5  /,Is( 14  )/ 13 /,Is( 15  )/ 2  /
        Data Is( 16  )/ 16 /,Is( 17  )/ 2  /,Is( 18  )/ 16 /
        Data Is( 19  )/ 1  /,Is( 20  )/ 17 /,Is( 21  )/-4  /
        Data Is( 22  )/ 22 /,Is( 23  )/ 4  /,Is( 24  )/ 14 /
        Data Is( 25  )/-4  /,Is( 26  )/ 22 /,Is( 27  )/ 4  /
        Data Is( 28  )/ 14 /,Is( 29  )/-2  /,Is( 30  )/ 20 /
        Data Is( 31  )/-1  /,Is( 32  )/ 19 /,Is( 33  )/-1  /
        Data Is( 34  )/ 19 /,Is( 35  )/-1  /,Is( 36  )/ 19 /
        Data Is( 37  )/-1  /,Is( 38  )/ 19 /,Is( 39  )/-1  /
        Data Is( 40  )/ 19 /,Is( 41  )/-1  /,Is( 42  )/ 19 /
        Data Is( 43  )/-1  /,Is( 44  )/ 19 /,Is( 45  )/-1  /
        Data Is( 46  )/ 19 /,Is( 47  )/-1  /,Is( 48  )/ 19 /
        Data Is( 49  )/-1  /,Is( 50  )/ 19 /,Is( 51  )/ 4  /
        Data Is( 52  )/ 14 /,Is( 53  )/ 4  /,Is( 54  )/ 14 /
        Data Is( 55  )/-3  /,Is( 56  )/ 21 /,Is( 57  )/-4  /
        Data Is( 58  )/ 22 /,Is( 59  )/-3  /,Is( 60  )/ 21 /
        Data Is( 61  )/ 0  /,Is( 62  )/ 18 /,Is( 63  )/-4  /
        Data Is( 64  )/ 23 /,Is( 65  )/ 0  /,Is( 66  )/ 18 /
        Data Is( 67  )/-2  /,Is( 68  )/ 19 /,Is( 69  )/-1  /
        Data Is( 70  )/ 20 /,Is( 71  )/-2  /,Is( 72  )/ 19 /
        Data Is( 73  )/-1  /,Is( 74  )/ 18 /,Is( 75  )/-1  /
        Data Is( 76  )/ 17 /,Is( 77  )/-1  /,Is( 78  )/ 20 /
        Data Is( 79  )/-2  /,Is( 80  )/ 20 /,Is( 81  )/ 5  /
        Data Is( 82  )/ 13 /,Is( 83  )/ 1  /,Is( 84  )/ 17 /
        Data Is( 85  )/-2  /,Is( 86  )/ 19 /,Is( 87  )/-1  /
        Data Is( 88  )/ 16 /,Is( 89  )/-3  /,Is( 90  )/ 21 /
        Data Is( 91  )/-2  /,Is( 92  )/ 20 /,Is( 93  )/-2  /
        Data Is( 94  )/ 20 /,Is( 95  )/-2  /,Is( 96  )/ 19 /
        Data Is( 97  )/-2  /,Is( 98  )/ 20 /,Is( 99  )/-2  /
        Data Is( 100 )/ 19 /,Is( 101 )/-1  /,Is( 102 )/ 19 /
        Data Is( 103 )/ 1  /,Is( 104 )/ 17 /,Is( 105 )/-2  /
        Data Is( 106 )/ 20 /,Is( 107 )/ 0  /,Is( 108 )/ 18 /
        Data Is( 109 )/-3  /,Is( 110 )/ 21 /,Is( 111 )/-1  /
```

```
Data Is( 112 )/ 19 /,Is( 113 )/ 0  /,Is( 114 )/ 18 /
Data Is( 115 )/-1  /,Is( 116 )/ 19 /,Is( 117 )/ 0  /
Data Is( 118 )/ 0  /,Is( 119 )/ 0  /,Is( 120 )/ 0  /
Data Is( 121 )/ 0  /,Is( 122 )/ 0  /,Is( 123 )/ 0  /
Data Is( 124 )/ 0  /,Is( 125 )/ 0  /,Is( 126 )/ 0  /
Data Is( 127 )/ 0  /,Is( 128 )/ 0  /,Is( 129 )/ 0  /
Data Is( 130 )/ 19 /,Is( 131 )/-1  /,Is( 132 )/ 18 /
Data Is( 133 )/ 0  /,Is( 134 )/ 18 /,Is( 135 )/ 0  /
Data Is( 136 )/ 19 /,Is( 137 )/ 0  /,Is( 138 )/ 18 /
Data Is( 139 )/ 4  /,Is( 140 )/ 16 /,Is( 141 )/ 0  /
Data Is( 142 )/ 19 /,Is( 143 )/ 0  /,Is( 144 )/ 19 /
Data Is( 145 )/ 5  /,Is( 146 )/ 13 /,Is( 147 )/ 4  /
Data Is( 148 )/ 14 /,Is( 149 )/ 0  /,Is( 150 )/ 17 /
Data Is( 151 )/ 5  /,Is( 152 )/ 13 /,Is( 153 )/-6  /
Data Is( 154 )/ 24 /,Is( 155 )/ 0  /,Is( 156 )/ 19 /
Data Is( 157 )/ 0  /,Is( 158 )/ 19 /,Is( 159 )/-1  /
Data Is( 160 )/ 18 /,Is( 161 )/ 0  /,Is( 162 )/ 19 /
Data Is( 163 )/ 2  /,Is( 164 )/ 15 /,Is( 165 )/ 1  /
Data Is( 166 )/ 18 /,Is( 167 )/ 4  /,Is( 168 )/ 16 /
Data Is( 169 )/ 0  /,Is( 170 )/ 19 /,Is( 171 )/ 1  /
Data Is( 172 )/ 17 /,Is( 173 )/-2  /,Is( 174 )/ 20 /
Data Is( 175 )/ 1  /,Is( 176 )/ 18 /,Is( 177 )/ 1  /
Data Is( 178 )/ 17 /,Is( 179 )/ 1  /,Is( 180 )/ 18 /
Data Is( 181 )/ 0  /,Is( 182 )/ 0  /,Is( 183 )/ 5  /
Data Is( 184 )/ 13 /
Isptr=Ichar-32
Isp1=Is(Isptr*2-1)
Isp2=Is(Isptr*2)
Return
End
```

```
      Subroutine Space2(Ichar,Ispl,Isp2)

C  See Spacel.For for some details.

C     ********************************************************
C     *        CHARACTER SPACING TABLE FOR SCRIPT CHAR SET *
C     ********************************************************
      Dimension Is(116)
      Data Is( 1)/-2  /,Is( 2)/ 18 /,Is( 3)/-3  /
      Data Is( 4)/ 20 /,Is( 5)/-1  /,Is( 6)/ 19 /
      Data Is( 7)/-2  /,Is( 8)/ 21 /,Is( 9)/-1  /
      Data Is( 10)/ 19 /,Is( 11)/-1  /,Is( 12)/ 19 /
      Data Is( 13)/-2  /,Is( 14)/ 21 /,Is( 15)/-3  /
      Data Is( 16)/ 21 /,Is( 17)/ 0  /,Is( 18)/ 17 /
      Data Is( 19)/ 1  /,Is( 20)/ 16 /,Is( 21)/-3  /
      Data Is( 22)/ 21 /,Is( 23)/ 0  /,Is( 24)/ 19 /
      Data Is( 25)/-9  /,Is( 26)/ 24 /,Is( 27)/-4  /
      Data Is( 28)/ 20 /,Is( 29)/-1  /,Is( 30)/ 20 /
      Data Is( 31)/-3  /,Is( 32)/ 22 /,Is( 33)/-1  /
      Data Is( 34)/ 21 /,Is( 35)/-3  /,Is( 36)/ 22 /
      Data Is( 37)/-1  /,Is( 38)/ 19 /,Is( 39)/-1  /
      Data Is( 40)/ 18 /,Is( 41)/-4  /,Is( 42)/ 20 /
      Data Is( 43)/-3  /,Is( 44)/ 20 /,Is( 45)/-6  /
      Data Is( 46)/ 22 /,Is( 47)/-3  /,Is( 48)/ 21 /
      Data Is( 49)/-3  /,Is( 50)/ 20 /,Is( 51)/-1  /
      Data Is( 52)/ 20 /,Is( 53)/ 0  /,Is( 54)/ 0  /
      Data Is( 55)/ 0  /,Is( 56)/ 0  /,Is( 57)/ 0  /
      Data Is( 58)/ 0  /,Is( 59)/ 0  /,Is( 60)/ 0  /
      Data Is( 61)/ 0  /,Is( 62)/ 0  /,Is( 63)/ 0  /
      Data Is( 64)/ 0  /,Is( 65)/ 3  /,Is( 66)/ 19 /
      Data Is( 67)/ 4  /,Is( 68)/ 18 /,Is( 69)/ 4  /
      Data Is( 70)/ 15 /,Is( 71)/ 3  /,Is( 72)/ 19 /
      Data Is( 73)/ 5  /,Is( 74)/ 15 /,Is( 75)/ 6  /
      Data Is( 76)/ 14 /,Is( 77)/ 3  /,Is( 78)/ 18 /
      Data Is( 79)/ 4  /,Is( 80)/ 19 /,Is( 81)/ 7  /
      Data Is( 82)/ 14 /,Is( 83)/ 7  /,Is( 84)/ 14 /
      Data Is( 85)/ 4  /,Is( 86)/ 18 /,Is( 87)/ 6  /
      Data Is( 88)/ 14 /,Is( 89)/-4  /,Is( 90)/ 21 /
      Data Is( 91)/ 1  /,Is( 92)/ 19 /,Is( 93)/ 3  /
      Data Is( 94)/ 17 /,Is( 95)/ 2  /,Is( 96)/ 17 /
      Data Is( 97)/ 3  /,Is( 98)/ 18 /,Is( 99)/ 4  /
      Data Is( 100 )/ 17 /,Is( 101 )/ 5  /,Is( 102 )/ 16 /
      Data Is( 103 )/ 6  /,Is( 104 )/ 15 /,Is( 105 )/ 3  /
      Data Is( 106 )/ 18 /,Is( 107 )/ 3  /,Is( 108 )/ 18 /
      Data Is( 109 )/ 0  /,Is( 110 )/ 21 /,Is( 111 )/ 1  /
      Data Is( 112 )/ 17 /,Is( 113 )/ 3  /,Is( 114 )/ 18 /
      Data Is( 115 )/ 3  /,Is( 116 )/ 17 /
      Isptr=Ichar-64
      Ispl=Is(Isptr*2-1)
      Isp2=Is(Isptr*2)
      Return
```

End

```
      Subroutine Space3(Ichar,Ispl,Isp2)

C  See Spacel.For for some details.

C     ********************************************************
C     *      CHARACTER SPACING TABLE FOR TRIPLEX CHAR SET    *
C     ********************************************************
      Dimension Is(180)
      Data Is( 1)/ 4  /,Is( 2)/ 15 /,Is( 3)/ 0  /
      Data Is( 4)/ 18 /,Is( 5)/ 0  /,Is( 6)/ 0  /
      Data Is( 7)/-1  /,Is( 8)/ 19 /,Is( 9)/ 0  /
      Data Is( 10)/ 0  /,Is( 11)/-4  /,Is( 12)/ 22 /
      Data Is( 13)/ 5  /,Is( 14)/ 14 /,Is( 15)/ 2  /
      Data Is( 16)/ 16 /,Is( 17)/ 2  /,Is( 18)/ 16 /
      Data Is( 19)/ 1  /,Is( 20)/ 17 /,Is( 21)/-3  /
      Data Is( 22)/ 22 /,Is( 23)/ 4  /,Is( 24)/ 15 /
      Data Is( 25)/-3  /,Is( 26)/ 22 /,Is( 27)/ 4  /
      Data Is( 28)/ 15 /,Is( 29)/-2  /,Is( 30)/ 21 /
      Data Is( 31)/-1  /,Is( 32)/ 19 /,Is( 33)/-1  /
      Data Is( 34)/ 19 /,Is( 35)/-1  /,Is( 36)/ 19 /
      Data Is( 37)/-1  /,Is( 38)/ 19 /,Is( 39)/-1  /
      Data Is( 40)/ 19 /,Is( 41)/-1  /,Is( 42)/ 19 /
      Data Is( 43)/-1  /,Is( 44)/ 19 /,Is( 45)/-1  /
      Data Is( 46)/ 19 /,Is( 47)/-1  /,Is( 48)/ 19 /
      Data Is( 49)/-1  /,Is( 50)/ 19 /,Is( 51)/ 4  /
      Data Is( 52)/ 15 /,Is( 53)/ 4  /,Is( 54)/ 15 /
      Data Is( 55)/ 0  /,Is( 56)/ 0  /,Is( 57)/-3  /
      Data Is( 58)/ 22 /,Is( 59)/ 0  /,Is( 60)/ 0  /
      Data Is( 61)/ 0  /,Is( 62)/ 19 /,Is( 63)/ 0  /
      Data Is( 64)/ 0  /,Is( 65)/-1  /,Is( 66)/ 19 /
      Data Is( 67)/-2  /,Is( 68)/ 20 /,Is( 69)/-2  /
      Data Is( 70)/ 19 /,Is( 71)/-2  /,Is( 72)/ 20 /
      Data Is( 73)/-2  /,Is( 74)/ 19 /,Is( 75)/-2  /
      Data Is( 76)/ 18 /,Is( 77)/-2  /,Is( 78)/ 21 /
      Data Is( 79)/-3  /,Is( 80)/ 21 /,Is( 81)/ 3  /
      Data Is( 82)/ 15 /,Is( 83)/ 1  /,Is( 84)/ 17 /
      Data Is( 85)/-3  /,Is( 86)/ 19 /,Is( 87)/ 0  /
      Data Is( 88)/ 18 /,Is( 89)/-4  /,Is( 90)/ 22 /
      Data Is( 91)/-3  /,Is( 92)/ 21 /,Is( 93)/-2  /
      Data Is( 94)/ 20 /,Is( 95)/-2  /,Is( 96)/ 20 /
      Data Is( 97)/-2  /,Is( 98)/ 20 /,Is( 99)/-2  /
      Data Is( 100 )/ 20 /,Is( 101 )/-1  /,Is( 102 )/ 19 /
      Data Is( 103 )/-1  /,Is( 104 )/ 19 /,Is( 105 )/-3  /
      Data Is( 106 )/ 21 /,Is( 107 )/-1  /,Is( 108 )/ 19 /
      Data Is( 109 )/-3  /,Is( 110 )/ 21 /,Is( 111 )/-1  /
      Data Is( 112 )/ 19 /,Is( 113 )/-2  /,Is( 114 )/ 20 /
      Data Is( 115 )/-1  /,Is( 116 )/ 19 /,Is( 117 )/ 4  /
      Data Is( 118 )/ 15 /,Is( 119 )/ 4  /,Is( 120 )/ 15 /
      Data Is( 121 )/ 0  /,Is( 122 )/ 0  /,Is( 123 )/ 0  /
      Data Is( 124 )/ 0  /,Is( 125 )/ 0  /,Is( 126 )/ 0  /
      Data Is( 127 )/ 0  /,Is( 128 )/ 0  /,Is( 129 )/ 0  /
```

```
Data Is( 130 )/ 20 /,Is( 131 )/-2  /,Is( 132 )/ 19 /
Data Is( 133 )/-1  /,Is( 134 )/ 18 /,Is( 135 )/-1  /
Data Is( 136 )/ 20 /,Is( 137 )/-1  /,Is( 138 )/ 18 /
Data Is( 139 )/ 2  /,Is( 140 )/ 16 /,Is( 141 )/ 0  /
Data Is( 142 )/ 19 /,Is( 143 )/-2  /,Is( 144 )/ 21 /
Data Is( 145 )/ 3  /,Is( 146 )/ 15 /,Is( 147 )/ 2  /
Data Is( 148 )/ 15 /,Is( 149 )/-2  /,Is( 150 )/ 20 /
Data Is( 151 )/ 3  /,Is( 152 )/ 15 /,Is( 153 )/-8  /
Data Is( 154 )/ 26 /,Is( 155 )/-2  /,Is( 156 )/ 21 /
Data Is( 157 )/-1  /,Is( 158 )/ 19 /,Is( 159 )/-2  /
Data Is( 160 )/ 19 /,Is( 161 )/-1  /,Is( 162 )/ 19 /
Data Is( 163 )/ 0  /,Is( 164 )/ 17 /,Is( 165 )/ 1  /
Data Is( 166 )/ 18 /,Is( 167 )/ 2  /,Is( 168 )/ 17 /
Data Is( 169 )/-2  /,Is( 170 )/ 21 /,Is( 171 )/ 0  /
Data Is( 172 )/ 18 /,Is( 173 )/-3  /,Is( 174 )/ 21 /
Data Is( 175 )/-1  /,Is( 176 )/ 19 /,Is( 177 )/-1  /
Data Is( 178 )/ 18 /,Is( 179 )/ 0  /,Is( 180 )/ 18 /
Isptr=Ichar-32
Isp1=Is(Isptr*2-1)
Isp2=Is(Isptr*2)
Return
End
```

```
      Subroutine Space4(Ichar,Ispl,Isp2)

C  See Spacel.For for some details.

C     ********************************************************
C     *        CHARACTER SPACING TABLE FOR GOTHIC CHAR SET *
C     ********************************************************
      Dimension Is(180)
      Data Is( 1 )/ 3  /,Is( 2 )/ 15 /,Is( 3 )/ 0  /
      Data Is( 4 )/ 18 /,Is( 5 )/ 0  /,Is( 6 )/ 0  /
      Data Is( 7 )/-1  /,Is( 8 )/ 19 /,Is( 9 )/ 0  /
      Data Is( 10)/ 0  /,Is( 11)/-4  /,Is( 12)/ 22 /
      Data Is( 13)/ 5  /,Is( 14)/ 14 /,Is( 15)/ 2  /
      Data Is( 16)/ 16 /,Is( 17)/ 2  /,Is( 18)/ 16 /
      Data Is( 19)/ 1  /,Is( 20)/ 17 /,Is( 21)/-3  /
      Data Is( 22)/ 22 /,Is( 23)/ 3  /,Is( 24)/ 15 /
      Data Is( 25)/-3  /,Is( 26)/ 22 /,Is( 27)/ 3  /
      Data Is( 28)/ 15 /,Is( 29)/-2  /,Is( 30)/ 21 /
      Data Is( 31)/-1  /,Is( 32)/ 19 /,Is( 33)/-1  /
      Data Is( 34)/ 19 /,Is( 35)/-1  /,Is( 36)/ 19 /
      Data Is( 37)/-1  /,Is( 38)/ 19 /,Is( 39)/-1  /
      Data Is( 40)/ 19 /,Is( 41)/-1  /,Is( 42)/ 19 /
      Data Is( 43)/-1  /,Is( 44)/ 19 /,Is( 45)/-1  /
      Data Is( 46)/ 19 /,Is( 47)/-1  /,Is( 48)/ 19 /
      Data Is( 49)/-1  /,Is( 50)/ 19 /,Is( 51)/ 3  /
      Data Is( 52)/ 15 /,Is( 53)/ 3  /,Is( 54)/ 15 /
      Data Is( 55)/ 0  /,Is( 56)/ 0  /,Is( 57)/-3  /
      Data Is( 58)/ 22 /,Is( 59)/ 0  /,Is( 60)/ 0  /
      Data Is( 61)/ 0  /,Is( 62)/ 18 /,Is( 63)/ 0  /
      Data Is( 64)/ 0  /,Is( 65)/-2  /,Is( 66)/ 20 /
      Data Is( 67)/-3  /,Is( 68)/ 21 /,Is( 69)/-4  /
      Data Is( 70)/ 20 /,Is( 71)/-2  /,Is( 72)/ 21 /
      Data Is( 73)/-2  /,Is( 74)/ 20 /,Is( 75)/-3  /
      Data Is( 76)/ 20 /,Is( 77)/-4  /,Is( 78)/ 21 /
      Data Is( 79)/-3  /,Is( 80)/ 21 /,Is( 81)/ 0  /
      Data Is( 82)/ 19 /,Is( 83)/-1  /,Is( 84)/ 19 /
      Data Is( 85)/-3  /,Is( 86)/ 21 /,Is( 87)/-2  /
      Data Is( 88)/ 20 /,Is( 89)/-5  /,Is( 90)/ 23 /
      Data Is( 91)/-4  /,Is( 92)/ 21 /,Is( 93)/-4  /
      Data Is( 94)/ 22 /,Is( 95)/-1  /,Is( 96)/ 21 /
      Data Is( 97)/-4  /,Is( 98)/ 22 /,Is( 99)/-3  /
      Data Is( 100)/ 21 /,Is( 101)/-2  /,Is( 102)/ 21 /
      Data Is( 103)/-4  /,Is( 104)/ 20 /,Is( 105)/-3  /
      Data Is( 106)/ 21 /,Is( 107)/-2  /,Is( 108)/ 21 /
      Data Is( 109)/-4  /,Is( 110)/ 23 /,Is( 111)/-2  /
      Data Is( 112)/ 20 /,Is( 113)/-2  /,Is( 114)/ 21 /
      Data Is( 115)/-1  /,Is( 116)/ 19 /,Is( 117)/ 3  /
      Data Is( 118)/ 15 /,Is( 119)/ 3  /,Is( 120)/ 15 /
      Data Is( 121)/ 0  /,Is( 122)/ 0  /,Is( 123)/ 0  /
      Data Is( 124)/ 0  /,Is( 125)/ 0  /,Is( 126)/ 0  /
      Data Is( 127)/ 0  /,Is( 128)/ 0  /,Is( 129)/ 1  /
```

```
Data Is( 130)/ 18 /,Is( 131)/ 0  /,Is( 132)/ 18 /
Data Is( 133)/ 1  /,Is( 134)/ 15 /,Is( 135)/ 0  /
Data Is( 136)/ 17 /,Is( 137)/ 1  /,Is( 138)/ 15 /
Data Is( 139)/ 1  /,Is( 140)/ 14 /,Is( 141)/ 0  /
Data Is( 142)/ 18 /,Is( 143)/ 0  /,Is( 144)/ 18 /
Data Is( 145)/ 4  /,Is( 146)/ 14 /,Is( 147)/ 4  /
Data Is( 148)/ 14 /,Is( 149)/ 0  /,Is( 150)/ 17 /
Data Is( 151)/ 4  /,Is( 152)/ 14 /,Is( 153)/-4  /
Data Is( 154)/ 22 /,Is( 155)/ 0  /,Is( 156)/ 18 /
Data Is( 157)/ 0  /,Is( 158)/ 18 /,Is( 159)/ 0  /
Data Is( 160)/ 18 /,Is( 161)/ 0  /,Is( 162)/ 18 /
Data Is( 163)/ 1  /,Is( 164)/ 15 /,Is( 165)/ 1  /
Data Is( 166)/ 17 /,Is( 167)/ 4  /,Is( 168)/ 14 /
Data Is( 169)/ 0  /,Is( 170)/ 18 /,Is( 171)/ 0  /
Data Is( 172)/ 18 /,Is( 173)/-4  /,Is( 174)/ 22 /
Data Is( 175)/-1  /,Is( 176)/ 18 /,Is( 177)/ 0  /
Data Is( 178)/ 18 /,Is( 179)/ 0  /,Is( 180)/ 18 /
Isptr=Ichar-32
Isp1=Is(Isptr*2-1)
Isp2=Is(Isptr*2)
Return
End
```

```
      Subroutine Stick(Char,Xpos,Ypos,Exit_Code)
C
C  This routine returns the next relative move/draw to generate the current
C  character in a stick font.

C  Parameters:

C  CHAR - b [i] = The character to generate.
C  XPOS,YPOS - R [o] = A relative move/draw coordinate.
C  EXIT_CODE - i [o] = 2 for a move
C                      3 for a draw
C                     -1 for no more graphic elements in character.

C  This routine was originally part of the Hewlett-Packard Plot21 system.
C  Its logic, but not its data areas, have been substantially re-written.

C  For documentation of the program logic, see the body of the virtually
C  identical routine TRIPLX.

      Implicit Integer (A-Z)

      Integer Ic(735),In(119), Ie(398)

      Byte Char

      Real Xm, Ym , Xpos, Ypos

      Include 'Chargen.Inc'
      Include 'Symvecmem.Inc'
      Include 'Options.Inc'

C     ********************************************************
C     *                                                      *
C     *                                                      *
C     *      THIS MODULE GENERATES STICK CHARACTERS          *
C     *                                                      *
C     *                                                      *
C     ********************************************************
C
C
C
C     ********************************************************
C     *             CHARACTER TABLE                          *
C     ********************************************************
C------<< 33 ! >>
      Data Ic( 1)/ 2   /,Ic( 2  )/ 201 /,Ic( 3  )/ 4   /
      Data Ic( 4  )/ 225 /
C------<< 34 " >>
      Data Ic( 5  )/ 9   /,Ic( 6  )/ 10  /,Ic( 7  )/ 1   /
      Data Ic( 8  )/ 11  /,Ic( 9  )/ 10  /
C------<< 35 # >>
```

```
      Data Ic( 10 )/ 12  /,Ic( 11 )/ 13  /,Ic( 12 )/ 1   /
      Data Ic( 13 )/ 14  /,Ic( 14 )/ 13  /,Ic( 15 )/ 1   /
      Data Ic( 16 )/ 15  /,Ic( 17 )/ 16  /,Ic( 18 )/ 1   /
      Data Ic( 19 )/ 17  /,Ic( 20 )/ 16  /
C-----<< 36 $ >>
      Data Ic( 21 )/ 18  /,Ic( 22 )/ 19  /,Ic( 23 )/ 1   /
      Data Ic( 24 )/ 20  /,Ic( 25 )/ 19  /,Ic( 26 )/ 1   /
      Data Ic( 27 )/ 21  /,Ic( 28 )/ 202 /
C-----<< 37 % >>
      Data Ic( 29 )/ 9   /,Ic( 30 )/ 203 /,Ic( 31 )/ 204 /
      Data Ic( 32 )/ 43  /,Ic( 33 )/ 1   /,Ic( 34 )/ 44  /
      Data Ic( 35 )/ 203 /
C-----<< 38 & >>
      Data Ic( 36 )/ 45  /,Ic( 37 )/ 46  /,Ic( 38 )/ 6   /
      Data Ic( 39 )/ 47  /,Ic( 40 )/ 5   /,Ic( 41 )/ 33  /
      Data Ic( 42 )/ 48  /,Ic( 43 )/ 49  /,Ic( 44 )/ 26  /
      Data Ic( 45 )/ 34  /,Ic( 46 )/ 24  /,Ic( 47 )/ 50  /
      Data Ic( 48 )/ 6   /,Ic( 49 )/ 51  /,Ic( 50 )/ 36  /
      Data Ic( 51 )/ 37  /,Ic( 52 )/ 7   /,Ic( 53 )/ 52  /
      Data Ic( 54 )/ 7   /,Ic( 55 )/ 37  /,Ic( 56 )/ 36  /
      Data Ic( 57 )/ 51  /,Ic( 58 )/ 50  /,Ic( 59 )/ 34  /
      Data Ic( 60 )/ 33  /,Ic( 61 )/ 27  /,Ic( 62 )/ 53  /
      Data Ic( 63 )/ 54  /,Ic( 64 )/ 55  /,Ic( 65 )/ 32  /
      Data Ic( 66 )/ 29  /,Ic( 67 )/ 39  /,Ic( 68 )/ 7   /
      Data Ic( 69 )/ 46  /
C-----<< 39 ' >>
      Data Ic( 70 )/ 2   /,Ic( 71 )/ 10  /
C-----<< 40 ( >>
      Data Ic( 72 )/ 56  /,Ic( 73 )/ 26  /,Ic( 74 )/ 49  /
      Data Ic( 75 )/ 57  /,Ic( 76 )/ 58  /,Ic( 77 )/ 59  /
      Data Ic( 78 )/ 60  /,Ic( 79 )/ 61  /,Ic( 80 )/ 54  /
      Data Ic( 81 )/ 32  /
C-----<< 4) >>
      Data Ic( 82 )/ 62  /,Ic( 83 )/ 32  /,Ic( 84 )/ 54  /
      Data Ic( 85 )/ 61  /,Ic( 86 )/ 60  /,Ic( 87 )/ 59  /
      Data Ic( 88 )/ 58  /,Ic( 89 )/ 57  /,Ic( 90 )/ 49  /
      Data Ic( 91 )/ 26  /
C-----<< 42 * >>
      Data Ic( 92 )/ 63  /,Ic( 93 )/ 64  /,Ic( 94 )/ 1   /
      Data Ic( 95 )/ 65  /,Ic( 96 )/ 66  /,Ic( 97 )/ 1   /
      Data Ic( 98 )/ 67  /,Ic( 99 )/ 68  /
C-----<< 43 + >>
      Data Ic( 100 )/ 69  /,Ic( 101 )/ 70  /,Ic( 102 )/ 1   /
      Data Ic( 103 )/ 71  /,Ic( 104 )/ 72  /
C-----<< 44 , >>
      Data Ic( 105 )/ 73  /,Ic( 106 )/ 225 /,Ic( 107 )/ 205 /
C-----<< 45 - >>
      Data Ic( 108 )/ 74  /,Ic( 109 )/ 72  /
C-----<< 46 . >>
      Data Ic( 110 )/ 73  /,Ic( 111 )/ 225 /
C-----<< 47/ >>
```

```
      Data Ic( 112 )/ 75  /,Ic( 113 )/ 76  /
C------<< 48 0 >>
      Data Ic( 114 )/ 77  /,Ic( 115 )/ 25  /,Ic( 116 )/ 49  /
      Data Ic( 117 )/ 58  /,Ic( 118 )/ 31  /,Ic( 119 )/ 60  /
      Data Ic( 120 )/ 54  /,Ic( 121 )/ 40  /,Ic( 122 )/ 39  /
      Data Ic( 123 )/ 42  /,Ic( 124 )/ 78  /,Ic( 125 )/ 79  /
      Data Ic( 126 )/ 80  /,Ic( 127 )/ 81  /,Ic( 128 )/ 82  /
      Data Ic( 129 )/ 23  /,Ic( 130 )/ 35  /
C------<< 49 1 >>
      Data Ic( 131 )/ 83  /,Ic( 132 )/ 38  /,Ic( 133 )/ 84  /
      Data Ic( 134 )/ 85  /
C------<< 50 2 >>
      Data Ic( 135 )/ 206 /,Ic( 136 )/ 27  /,Ic( 137 )/ 33  /
      Data Ic( 138 )/ 49  /,Ic( 139 )/ 88  /,Ic( 140 )/ 16  /
C------<< 51 3 >>
      Data Ic( 141 )/ 89  /,Ic( 142 )/ 90  /,Ic( 143 )/ 91  /
      Data Ic( 144 )/ 41  /,Ic( 145 )/ 29  /,Ic( 146 )/ 8   /
      Data Ic( 147 )/ 207 /
C------<< 52 4 >>
      Data Ic( 148 )/ 94  /,Ic( 149 )/ 95  /,Ic( 150 )/ 96  /
      Data Ic( 151 )/ 85  /
C------<< 53 5 >>
      Data Ic( 152 )/ 97  /,Ic( 153 )/ 98  /,Ic( 154 )/ 99  /
      Data Ic( 155 )/ 7   /,Ic( 156 )/ 42  /,Ic( 157 )/ 41  /
      Data Ic( 158 )/ 208 /,Ic( 159 )/ 207 /
C------<< 54 6 >>
      Data Ic( 160 )/ 100 /,Ic( 161 )/ 51  /,Ic( 162 )/ 23  /
      Data Ic( 163 )/ 35  /,Ic( 164 )/ 25  /,Ic( 165 )/ 49  /
      Data Ic( 166 )/ 58  /,Ic( 167 )/ 101 /,Ic( 168 )/ 102 /
      Data Ic( 169 )/ 32  /,Ic( 170 )/ 40  /,Ic( 171 )/ 103 /
      Data Ic( 172 )/ 42  /,Ic( 173 )/ 209 /,Ic( 174 )/ 46  /
      Data Ic( 175 )/ 106 /,Ic( 176 )/ 22  /,Ic( 177 )/ 23  /
      Data Ic( 178 )/ 47  /,Ic( 179 )/ 25  /,Ic( 180 )/ 26  /
      Data Ic( 181 )/ 92  /
C------<< 55 7 >>
      Data Ic( 182 )/ 107 /,Ic( 183 )/ 108 /,Ic( 184 )/ 109 /
C------<< 56 8 >>
      Data Ic( 185 )/ 110 /,Ic( 186 )/ 25  /,Ic( 187 )/ 33  /
      Data Ic( 188 )/ 27  /,Ic( 189 )/ 28  /,Ic( 190 )/ 29  /
      Data Ic( 191 )/ 111 /,Ic( 192 )/ 208 /,Ic( 193 )/ 28  /
      Data Ic( 194 )/ 210 /,Ic( 195 )/ 24  /,Ic( 196 )/ 23  /
      Data Ic( 197 )/ 6   /,Ic( 198 )/ 51  /,Ic( 199 )/ 80  /
      Data Ic( 200 )/ 37  /,Ic( 201 )/ 104 /,Ic( 202 )/ 42  /
      Data Ic( 203 )/ 112 /,Ic( 204 )/ 38  /,Ic( 205 )/ 37  /
      Data Ic( 206 )/ 36  /,Ic( 207 )/ 51  /,Ic( 208 )/ 23  /
      Data Ic( 209 )/ 24  /
C------<< 57 9 >>
      Data Ic( 210 )/ 113 /,Ic( 211 )/ 92  /,Ic( 212 )/ 26  /
      Data Ic( 213 )/ 25  /,Ic( 214 )/ 47  /,Ic( 215 )/ 23  /
      Data Ic( 216 )/ 22  /,Ic( 217 )/ 106 /,Ic( 218 )/ 46  /
      Data Ic( 219 )/ 105 /,Ic( 220 )/ 104 /,Ic( 221 )/ 42  /
```

```
      Data Ic( 222 )/ 103 /,Ic( 223 )/ 208 /,Ic( 224 )/ 102 /
      Data Ic( 225 )/ 101 /,Ic( 226 )/ 58  /,Ic( 227 )/ 49  /
      Data Ic( 228 )/ 25  /,Ic( 229 )/ 35  /,Ic( 230 )/ 23  /
      Data Ic( 231 )/ 51  /
C-----<< 58 : >>
      Data Ic( 232 )/ 211 /
C-----<< 59 ; >>
      Data Ic( 233 )/ 211 /,Ic( 234 )/ 205 /
C-----<< 60 < >>
      Data Ic( 235 )/ 115 /,Ic( 236 )/ 116 /,Ic( 237 )/ 117 /
C-----<< 61 = >>
      Data Ic( 238 )/ 118 /,Ic( 239 )/ 72  /,Ic( 240 )/ 1   /
      Data Ic( 241 )/ 119 /,Ic( 242 )/ 72  /
C-----<< 62 > >>
      Data Ic( 243 )/ 120 /,Ic( 244 )/ 117 /,Ic( 245 )/ 116 /
C-----<< 63 ? >>
      Data Ic( 246 )/ 206 /,Ic( 247 )/ 205 /,Ic( 248 )/ 121 /
      Data Ic( 249 )/ 31  /,Ic( 250 )/ 1   /,Ic( 251 )/ 4   /
      Data Ic( 252 )/ 225 /
C-----<< 64 @ >>
      Data Ic( 253 )/ 122 /,Ic( 254 )/ 51  /,Ic( 255 )/ 212 /
      Data Ic( 256 )/ 34  /,Ic( 257 )/ 5   /,Ic( 258 )/ 92  /
      Data Ic( 259 )/ 31  /,Ic( 260 )/ 28  /,Ic( 261 )/ 29  /
      Data Ic( 262 )/ 41  /,Ic( 263 )/ 38  /,Ic( 264 )/ 37  /
      Data Ic( 265 )/ 1   /,Ic( 266 )/ 123 /,Ic( 267 )/ 26  /
      Data Ic( 268 )/ 92  /,Ic( 269 )/ 31  /,Ic( 270 )/ 28  /
      Data Ic( 271 )/ 8   /,Ic( 272 )/ 1   /,Ic( 273 )/ 124 /
      Data Ic( 274 )/ 125 /,Ic( 275 )/ 27  /,Ic( 276 )/ 29  /
      Data Ic( 277 )/ 39  /,Ic( 278 )/ 209 /,Ic( 279 )/ 36  /
      Data Ic( 280 )/ 106 /,Ic( 281 )/ 213 /,Ic( 282 )/ 23  /
      Data Ic( 283 )/ 93  /,Ic( 284 )/ 25  /,Ic( 285 )/ 34  /
      Data Ic( 286 )/ 26  /,Ic( 287 )/ 33  /,Ic( 288 )/ 92  /
      Data Ic( 289 )/ 31  /,Ic( 290 )/ 53  /,Ic( 291 )/ 28  /
      Data Ic( 292 )/ 32  /,Ic( 293 )/ 204 /,Ic( 294 )/ 7   /
      Data Ic( 295 )/ 1   /,Ic( 296 )/ 126 /,Ic( 297 )/ 125 /
      Data Ic( 298 )/ 27  /,Ic( 299 )/ 8   /
C-----<< 65 A >>
      Data Ic( 300 )/ 103 /,Ic( 301 )/ 77  /,Ic( 302 )/ 127 /
      Data Ic( 303 )/ 1   /,Ic( 304 )/ 128 /,Ic( 305 )/ 129 /
C-----<< 66 B >>
      Data Ic( 306 )/ 130 /,Ic( 307 )/ 214 /,Ic( 308 )/ 210 /
      Data Ic( 309 )/ 132 /,Ic( 310 )/ 215 /
C-----<< 67 C >>
      Data Ic( 311 )/ 224 /
C-----<< 68 D >>
      Data Ic( 312 )/ 39  /,Ic( 313 )/ 133 /,Ic( 314 )/ 135 /
      Data Ic( 315 )/ 208 /,Ic( 316 )/ 28  /,Ic( 317 )/ 53  /
      Data Ic( 318 )/ 101 /,Ic( 319 )/ 92  /,Ic( 320 )/ 33  /
      Data Ic( 321 )/ 26  /,Ic( 322 )/ 25  /,Ic( 323 )/ 136 /
C-----<< 69 E >>
      Data Ic( 324 )/ 137 /,Ic( 325 )/ 138 /,Ic( 326 )/ 85  /
```

```
      Data Ic( 327 )/ 139 /,Ic( 328 )/ 1   /,Ic( 329 )/ 140 /
      Data Ic( 330 )/ 141 /
C-----<< 70 F >>
      Data Ic( 331 )/ 41  /,Ic( 332 )/ 133 /,Ic( 333 )/ 139 /
      Data Ic( 334 )/ 1   /,Ic( 335 )/ 142 /,Ic( 336 )/ 141 /
C-----<< 71 G >>
      Data Ic( 337 )/ 224 /,Ic( 338 )/ 80  /,Ic( 339 )/ 143 /
C-----<< 72 H >>
      Data Ic( 340 )/ 144 /,Ic( 341 )/ 221 /,Ic( 342 )/ 97  /
      Data Ic( 343 )/ 221 /,Ic( 344 )/ 145 /,Ic( 345 )/ 16  /
C-----<< 73 I >>
      Data Ic( 346 )/ 2   /,Ic( 347 )/ 85  /
C-----<< 74 J >>
      Data Ic( 348 )/ 146 /,Ic( 349 )/ 216 /,Ic( 350 )/ 35  /
      Data Ic( 351 )/ 50  /,Ic( 352 )/ 6   /,Ic( 353 )/ 106 /
      Data Ic( 354 )/ 36  /
C-----<< 75 K >>
      Data Ic( 355 )/ 144 /,Ic( 356 )/ 221 /,Ic( 357 )/ 97  /
      Data Ic( 358 )/ 148 /,Ic( 359 )/ 1   /,Ic( 360 )/ 149 /
      Data Ic( 361 )/ 150 /
C-----<< 76 L >>
      Data Ic( 362 )/ 151 /,Ic( 363 )/ 85  /,Ic( 364 )/ 152 /
C-----<< 77 M >>
      Data Ic( 365 )/ 103 /,Ic( 366 )/ 133 /,Ic( 367 )/ 127 /
      Data Ic( 368 )/ 77  /,Ic( 369 )/ 85  /
C-----<< 78 N >>
      Data Ic( 370 )/ 39  /,Ic( 371 )/ 133 /,Ic( 372 )/ 153 /
      Data Ic( 373 )/ 133 /
C-----<< 79 O >>
      Data Ic( 374 )/ 217 /
C-----<< 80 P >>
      Data Ic( 375 )/ 39  /,Ic( 376 )/ 133 /,Ic( 377 )/ 214 /
      Data Ic( 378 )/ 210 /,Ic( 379 )/ 132 /
C-----<< 81 Q >>
      Data Ic( 380 )/ 217 /,Ic( 381 )/ 1   /,Ic( 382 )/ 155 /
      Data Ic( 383 )/ 156 /
C-----<< 82 R >>
      Data Ic( 384 )/ 39  /,Ic( 385 )/ 215 /,Ic( 386 )/ 132 /
      Data Ic( 387 )/ 1   /,Ic( 388 )/ 135 /,Ic( 389 )/ 157 /
C-----<< 83 S >>
      Data Ic( 390 )/ 158 /,Ic( 391 )/ 202 /
C-----<< 84 T >>
      Data Ic( 392 )/ 2   /,Ic( 393 )/ 221 /,Ic( 394 )/ 159 /
      Data Ic( 395 )/ 16  /
C-----<< 85 U >>
      Data Ic( 396 )/ 144 /,Ic( 397 )/ 160 /,Ic( 398 )/ 53  /
      Data Ic( 399 )/ 32  /,Ic( 400 )/ 40  /,Ic( 401 )/ 39  /
      Data Ic( 402 )/ 42  /,Ic( 403 )/ 209 /,Ic( 404 )/ 161 /
C-----<< 86 V >>
      Data Ic( 405 )/ 162 /,Ic( 406 )/ 127 /,Ic( 407 )/ 77  /
C-----<< 87 W >>
```

```
      Data Ic( 408 )/ 163 /,Ic( 409 )/ 164 /,Ic( 410 )/ 9   /
      Data Ic( 411 )/ 164 /,Ic( 412 )/ 9   /
C------<< 88 X >>
      Data Ic( 413 )/ 144 /,Ic( 414 )/ 153 /,Ic( 415 )/ 1   /
      Data Ic( 416 )/ 133 /,Ic( 417 )/ 165 /
C------<< 89 Y >>
      Data Ic( 418 )/ 162 /,Ic( 419 )/ 166 /,Ic( 420 )/ 167 /
      Data Ic( 421 )/ 1   /,Ic( 422 )/ 77  /,Ic( 423 )/ 168 /
C------<< 90 Z >>
      Data Ic( 424 )/ 144 /,Ic( 425 )/ 16  /,Ic( 426 )/ 165 /
      Data Ic( 427 )/ 16  /
C------<< 97 A >>
      Data Ic( 428 )/ 113 /,Ic( 429 )/ 201 /,Ic( 430 )/ 169 /
      Data Ic( 431 )/ 222 /
C------<< 98 B >>
      Data Ic( 432 )/ 151 /,Ic( 433 )/ 221 /,Ic( 434 )/ 169 /
      Data Ic( 435 )/ 218 /
C------<< 99 C >>
      Data Ic( 436 )/ 170 /,Ic( 437 )/ 222 /
C------<<100 D >>
      Data Ic( 438 )/ 171 /,Ic( 439 )/ 221 /,Ic( 440 )/ 169 /
      Data Ic( 441 )/ 222 /
C------<<101 E >>
      Data Ic( 442 )/ 172 /,Ic( 443 )/ 152 /,Ic( 444 )/ 36  /
      Data Ic( 445 )/ 51  /,Ic( 446 )/ 6   /,Ic( 447 )/ 212 /
      Data Ic( 448 )/ 34  /,Ic( 449 )/ 223 /,Ic( 450 )/ 104 /
C------<<102 F >>
      Data Ic( 451 )/ 97  /,Ic( 452 )/ 35  /,Ic( 453 )/ 34  /
      Data Ic( 454 )/ 92  /,Ic( 455 )/ 173 /,Ic( 456 )/ 1   /
      Data Ic( 457 )/ 174 /,Ic( 458 )/ 135 /
C------<<103 G >>
      Data Ic( 459 )/ 113 /,Ic( 460 )/ 216 /,Ic( 461 )/ 93  /
      Data Ic( 462 )/ 50  /,Ic( 463 )/ 1   /,Ic( 464 )/ 175 /
      Data Ic( 465 )/ 222 /
C------<<104 H >>
      Data Ic( 466 )/ 89  /,Ic( 467 )/ 221 /,Ic( 468 )/ 219 /
C------<<105 I >>
      Data Ic( 469 )/ 108 /,Ic( 470 )/ 225 /,Ic( 471 )/ 1   /
      Data Ic( 472 )/ 178 /,Ic( 473 )/ 3   /
C------<<106 J >>
      Data Ic( 474 )/ 179 /,Ic( 475 )/ 225 /,Ic( 476 )/ 1   /
      Data Ic( 477 )/ 178 /,Ic( 478 )/ 173 /,Ic( 479 )/ 92  /
      Data Ic( 480 )/ 34  /,Ic( 481 )/ 35  /
C------<<107 K >>
      Data Ic( 482 )/ 89  /,Ic( 483 )/ 221 /,Ic( 484 )/ 96  /
      Data Ic( 485 )/ 88  /,Ic( 486 )/ 1   /,Ic( 487 )/ 180 /
      Data Ic( 488 )/ 181 /
C------<<108 L >>
      Data Ic( 489 )/ 2   /,Ic( 490 )/ 85  /
C------<<109 M >>
      Data Ic( 491 )/ 182 /,Ic( 492 )/ 201 /,Ic( 493 )/ 219 /
```

```
      Data Ic( 494 )/ 1   /,Ic( 495 )/ 219 /
C------<<110 N >>
      Data Ic( 496 )/ 183 /,Ic( 497 )/ 201 /,Ic( 498 )/ 219 /
C------<<111 O >>
      Data Ic( 499 )/ 184 /,Ic( 500 )/ 34  /,Ic( 501 )/ 223 /
      Data Ic( 502 )/ 209 /,Ic( 503 )/ 36  /,Ic( 504 )/ 106 /
      Data Ic( 505 )/ 22  /,Ic( 506 )/ 212 /
C------<<112 P >>
      Data Ic( 507 )/ 185 /,Ic( 508 )/ 221 /,Ic( 509 )/ 186 /
      Data Ic( 510 )/ 218 /
C------<<113 Q >>
      Data Ic( 511 )/ 113 /,Ic( 512 )/ 221 /,Ic( 513 )/ 186 /
      Data Ic( 514 )/ 222 /
C------<<114 R >>
      Data Ic( 515 )/ 187 /,Ic( 516 )/ 201 /,Ic( 517 )/ 188 /
      Data Ic( 518 )/ 105 /,Ic( 519 )/ 104 /,Ic( 520 )/ 38  /
      Data Ic( 521 )/ 41  /
C------<<115 S >>
      Data Ic( 522 )/ 170 /,Ic( 523 )/ 51  /,Ic( 524 )/ 23  /
      Data Ic( 525 )/ 93  /,Ic( 526 )/ 25  /,Ic( 527 )/ 33  /
      Data Ic( 528 )/ 28  /,Ic( 529 )/ 29  /,Ic( 530 )/ 189 /
      Data Ic( 531 )/ 29  /,Ic( 532 )/ 28  /,Ic( 533 )/ 190 /
      Data Ic( 534 )/ 33  /,Ic( 535 )/ 25  /,Ic( 536 )/ 93  /
      Data Ic( 537 )/ 23  /,Ic( 538 )/ 51  /
C------<<116 T >>
      Data Ic( 539 )/ 2   /,Ic( 540 )/ 173 /,Ic( 541 )/ 53  /
      Data Ic( 542 )/ 29  /,Ic( 543 )/ 39  /,Ic( 544 )/ 1   /
      Data Ic( 545 )/ 191 /,Ic( 546 )/ 135 /
C------<<117 U >>
      Data Ic( 547 )/ 183 /,Ic( 548 )/ 177 /,Ic( 549 )/ 53  /
      Data Ic( 550 )/ 29  /,Ic( 551 )/ 41  /,Ic( 552 )/ 38  /
      Data Ic( 553 )/ 84  /,Ic( 554 )/ 1   /,Ic( 555 )/ 176 /
      Data Ic( 556 )/ 3   /
C------<<118 V >>
      Data Ic( 557 )/ 185 /,Ic( 558 )/ 192 /,Ic( 559 )/ 187 /
C------<<119 W >>
      Data Ic( 560 )/ 193 /,Ic( 561 )/ 194 /,Ic( 562 )/ 183 /
      Data Ic( 563 )/ 194 /,Ic( 564 )/ 183 /
C------<<120 X >>
      Data Ic( 565 )/ 183 /,Ic( 566 )/ 195 /,Ic( 567 )/ 1   /
      Data Ic( 568 )/ 196 /,Ic( 569 )/ 197 /
C------<<121 Y >>
      Data Ic( 570 )/ 185 /,Ic( 571 )/ 192 /,Ic( 572 )/ 1   /
      Data Ic( 573 )/ 187 /,Ic( 574 )/ 198 /,Ic( 575 )/ 57  /
      Data Ic( 576 )/ 26  /,Ic( 577 )/ 34  /,Ic( 578 )/ 47  /
C------<<122 Z >>
      Data Ic( 579 )/ 183 /,Ic( 580 )/ 90  /,Ic( 581 )/ 197 /
      Data Ic( 582 )/ 90  /
C------<<124 | >>
      Data Ic( 583 )/ 199 /,Ic( 584 )/ 200 /
C------<<201   >>
```

```
      Data Ic( 585 )/ 3   /,Ic( 586 )/ 1   /
C-----<<202   >>
      Data Ic( 587 )/ 22  /,Ic( 588 )/ 23  /,Ic( 589 )/ 24  /
      Data Ic( 590 )/ 25  /,Ic( 591 )/ 26  /,Ic( 592 )/ 27  /
      Data Ic( 593 )/ 28  /,Ic( 594 )/ 8   /,Ic( 595 )/ 29  /
      Data Ic( 596 )/ 30  /,Ic( 597 )/ 29  /,Ic( 598 )/ 8   /
      Data Ic( 599 )/ 28  /,Ic( 600 )/ 31  /,Ic( 601 )/ 26  /
      Data Ic( 602 )/ 25  /,Ic( 603 )/ 24  /,Ic( 604 )/ 23  /
      Data Ic( 605 )/ 22  /
C-----<<203   >>
      Data Ic( 606 )/ 32  /,Ic( 607 )/ 27  /,Ic( 608 )/ 33  /
      Data Ic( 609 )/ 34  /,Ic( 610 )/ 35  /,Ic( 611 )/ 22  /
      Data Ic( 612 )/ 36  /,Ic( 613 )/ 37  /,Ic( 614 )/ 38  /
      Data Ic( 615 )/ 39  /
C-----<<204   >>
      Data Ic( 616 )/ 29  /,Ic( 617 )/ 40  /,Ic( 618 )/ 41  /
      Data Ic( 619 )/ 42  /,Ic( 620 )/ 38  /
C-----<<205   >>
      Data Ic( 621 )/ 27  /,Ic( 622 )/ 33  /,Ic( 623 )/ 5   /
C-----<<206   >>
      Data Ic( 624 )/ 86  /,Ic( 625 )/ 46  /,Ic( 626 )/ 37  /
      Data Ic( 627 )/ 7   /,Ic( 628 )/ 38  /,Ic( 629 )/ 87  /
      Data Ic( 630 )/ 29  /,Ic( 631 )/ 8   /,Ic( 632 )/ 28  /
C-----<<207   >>
      Data Ic( 633 )/ 53  /,Ic( 634 )/ 27  /,Ic( 635 )/ 92  /
      Data Ic( 636 )/ 26  /,Ic( 637 )/ 25  /,Ic( 638 )/ 93  /
      Data Ic( 639 )/ 23  /,Ic( 640 )/ 6   /,Ic( 641 )/ 51  /
C-----<<208   >>
      Data Ic( 642 )/ 40  /,Ic( 643 )/ 32  /
C-----<<209   >>
      Data Ic( 644 )/ 104 /,Ic( 645 )/ 105 /
C-----<<210   >>
      Data Ic( 646 )/ 31  /,Ic( 647 )/ 33  /,Ic( 648 )/ 5   /
      Data Ic( 649 )/ 25  /
C-----<<211   >>
      Data Ic( 650 )/ 114 /,Ic( 651 )/ 225 /,Ic( 652 )/ 1   /
      Data Ic( 653 )/ 64  /,Ic( 654 )/ 225 /
C-----<<212   >>
      Data Ic( 655 )/ 50  /,Ic( 656 )/ 93  /
C-----<<213   >>
      Data Ic( 657 )/ 51  /,Ic( 658 )/ 22  /,Ic( 659 )/ 50  /
C-----<<214   >>
      Data Ic( 660 )/ 131 /,Ic( 661 )/ 40  /,Ic( 662 )/ 8   /
      Data Ic( 663 )/ 28  /
C-----<<215   >>
      Data Ic( 664 )/ 133 /,Ic( 665 )/ 214 /,Ic( 666 )/ 205 /
      Data Ic( 667 )/ 25  /
C-----<<216   >>
      Data Ic( 668 )/ 147 /,Ic( 669 )/ 92  /,Ic( 670 )/ 5   /
      Data Ic( 671 )/ 34  /
C-----<<217   >>
```

```
      Data Ic( 672 )/ 110 /,Ic( 673 )/ 220 /,Ic( 674 )/ 105 /
      Data Ic( 675 )/ 154 /,Ic( 676 )/ 106 /,Ic( 677 )/ 213 /
      Data Ic( 678 )/ 24  /
C-----<<218   >>
      Data Ic( 679 )/ 104 /,Ic( 680 )/ 38  /,Ic( 681 )/ 41  /
      Data Ic( 682 )/ 29  /,Ic( 683 )/ 32  /,Ic( 684 )/ 53  /
      Data Ic( 685 )/ 27  /,Ic( 686 )/ 92  /,Ic( 687 )/ 26  /
      Data Ic( 688 )/ 34  /,Ic( 689 )/ 93  /,Ic( 690 )/ 50  /
      Data Ic( 691 )/ 22  /
C-----<<219   >>
      Data Ic( 692 )/ 176 /,Ic( 693 )/ 84  /,Ic( 694 )/ 38  /
      Data Ic( 695 )/ 41  /,Ic( 696 )/ 29  /,Ic( 697 )/ 53  /
      Data Ic( 698 )/ 177 /
C-----<<220   >>
      Data Ic( 699 )/ 34  /,Ic( 700 )/ 26  /,Ic( 701 )/ 33  /
      Data Ic( 702 )/ 92  /,Ic( 703 )/ 101 /,Ic( 704 )/ 53  /
      Data Ic( 705 )/ 28  /,Ic( 706 )/ 32  /,Ic( 707 )/ 29  /
      Data Ic( 708 )/ 87  /,Ic( 709 )/ 38  /,Ic( 710 )/ 104 /
      Data Ic( 711 )/ 37  /
C-----<<221   >>
      Data Ic( 712 )/ 85  /,Ic( 713 )/ 1   /
C-----<<222   >>
      Data Ic( 714 )/ 22  /,Ic( 715 )/ 212 /,Ic( 716 )/ 34  /
      Data Ic( 717 )/ 223 /,Ic( 718 )/ 104 /
C-----<<223   >>
      Data Ic( 719 )/ 26  /,Ic( 720 )/ 92  /,Ic( 721 )/ 27  /
      Data Ic( 722 )/ 53  /,Ic( 723 )/ 32  /,Ic( 724 )/ 29  /
      Data Ic( 725 )/ 41  /,Ic( 726 )/ 38  /
C-----<<224   >>
      Data Ic( 727 )/ 134 /,Ic( 728 )/ 213 /,Ic( 729 )/ 24  /
      Data Ic( 730 )/ 220 /
C-----<<225   >>
      Data Ic( 731 )/ 5   /,Ic( 732 )/ 6   /,Ic( 733 )/ 7   /
      Data Ic( 734 )/ 8   /,Ic( 735 )/ 1   /
C
C     ******************************************************
C     *          CHARACTER INDEX TABLE                     *
C     ******************************************************
C
      Data In( 1  )/ 0   /,In( 2  )/ 0   /,In( 3  )/ 4   /
      Data In( 4  )/ 9   /,In( 5  )/ 20  /,In( 6  )/ 28  /
      Data In( 7  )/ 35  /,In( 8  )/ 69  /,In( 9  )/ 71  /
      Data In( 10 )/ 81  /,In( 11 )/ 91  /,In( 12 )/ 99  /
      Data In( 13 )/ 104 /,In( 14 )/ 107 /,In( 15 )/ 109 /
      Data In( 16 )/ 111 /,In( 17 )/ 113 /,In( 18 )/ 130 /
      Data In( 19 )/ 134 /,In( 20 )/ 140 /,In( 21 )/ 147 /
      Data In( 22 )/ 151 /,In( 23 )/ 159 /,In( 24 )/ 181 /
      Data In( 25 )/ 184 /,In( 26 )/ 209 /,In( 27 )/ 231 /
      Data In( 28 )/ 232 /,In( 29 )/ 234 /,In( 30 )/ 237 /
      Data In( 31 )/ 242 /,In( 32 )/ 245 /,In( 33 )/ 252 /
      Data In( 34 )/ 299 /,In( 35 )/ 305 /,In( 36 )/ 310 /
```

```
      Data In( 37 )/ 311 /,In( 38 )/ 323 /,In( 39 )/ 330 /
      Data In( 40 )/ 336 /,In( 41 )/ 339 /,In( 42 )/ 345 /
      Data In( 43 )/ 347 /,In( 44 )/ 354 /,In( 45 )/ 361 /
      Data In( 46 )/ 364 /,In( 47 )/ 369 /,In( 48 )/ 373 /
      Data In( 49 )/ 374 /,In( 50 )/ 379 /,In( 51 )/ 383 /
      Data In( 52 )/ 389 /,In( 53 )/ 391 /,In( 54 )/ 395 /
      Data In( 55 )/ 404 /,In( 56 )/ 407 /,In( 57 )/ 412 /
      Data In( 58 )/ 417 /,In( 59 )/ 423 /,In( 60 )/ 427 /
      Data In( 61 )/ 427 /,In( 62 )/ 427 /,In( 63 )/ 427 /
      Data In( 64 )/ 427 /,In( 65 )/ 427 /,In( 66 )/ 427 /
      Data In( 67 )/ 431 /,In( 68 )/ 435 /,In( 69 )/ 437 /
      Data In( 70 )/ 441 /,In( 71 )/ 450 /,In( 72 )/ 458 /
      Data In( 73 )/ 465 /,In( 74 )/ 468 /,In( 75 )/ 473 /
      Data In( 76 )/ 481 /,In( 77 )/ 488 /,In( 78 )/ 490 /
      Data In( 79 )/ 495 /,In( 80 )/ 498 /,In( 81 )/ 506 /
      Data In( 82 )/ 510 /,In( 83 )/ 514 /,In( 84 )/ 521 /
      Data In( 85 )/ 538 /,In( 86 )/ 546 /,In( 87 )/ 556 /
      Data In( 88 )/ 559 /,In( 89 )/ 564 /,In( 90 )/ 569 /
      Data In( 91 )/ 578 /,In( 92 )/ 582 /,In( 93 )/ 582 /
      Data In( 94 )/ 584 /,In( 95 )/ 586 /,In( 96 )/ 605 /
      Data In( 97 )/ 615 /,In( 98 )/ 620 /,In( 99 )/ 623 /
      Data In( 100 )/ 632 /,In( 101 )/ 641 /,In( 102 )/ 643 /
      Data In( 103 )/ 645 /,In( 104 )/ 649 /,In( 105 )/ 654 /
      Data In( 106 )/ 656 /,In( 107 )/ 659 /,In( 108 )/ 663 /
      Data In( 109 )/ 667 /,In( 110 )/ 671 /,In( 111 )/ 678 /
      Data In( 112 )/ 691 /,In( 113 )/ 698 /,In( 114 )/ 711 /
      Data In( 115 )/ 713 /,In( 116 )/ 718 /,In( 117 )/ 726 /
      Data In( 118 )/ 730 /,In( 119 )/ 735 /
C
C     ********************************************************
C     *        BASIC ELEMENTS TABLE                          *
C     ********************************************************
C
      Data Ie( 1  )/ 9  /,Ie( 2  )/ 21 /,Ie( 3  )/ 0  /
      Data Ie( 4  )/-14 /,Ie( 5  )/ 1  /,Ie( 6  )/-6  /
      Data Ie( 7  )/-1  /,Ie( 8  )/-1  /,Ie( 9  )/-1  /
      Data Ie( 10 )/ 1  /,Ie( 11 )/ 1  /,Ie( 12 )/ 1  /
      Data Ie( 13 )/ 1  /,Ie( 14 )/-1  /,Ie( 15 )/ 5  /
      Data Ie( 16 )/ 21 /,Ie( 17 )/ 0  /,Ie( 18 )/-7  /
      Data Ie( 19 )/ 8  /,Ie( 20 )/ 7  /,Ie( 21 )/ 10 /
      Data Ie( 22 )/ 25 /,Ie( 23 )/-7  /,Ie( 24 )/-32 /
      Data Ie( 25 )/ 13 /,Ie( 26 )/ 32 /,Ie( 27 )/-6  /
      Data Ie( 28 )/ 19 /,Ie( 29 )/ 14 /,Ie( 30 )/ 0  /
      Data Ie( 31 )/-15 /,Ie( 32 )/-6  /,Ie( 33 )/ 7  /
      Data Ie( 34 )/ 25 /,Ie( 35 )/ 0  /,Ie( 36 )/-29 /
      Data Ie( 37 )/ 4  /,Ie( 38 )/ 29 /,Ie( 39 )/ 5  /
      Data Ie( 40 )/ 22 /,Ie( 41 )/-2  /,Ie( 42 )/ 2  /
      Data Ie( 43 )/-3  /,Ie( 44 )/ 1  /,Ie( 45 )/-4  /
      Data Ie( 46 )/ 0  /,Ie( 47 )/-3  /,Ie( 48 )/-1  /
      Data Ie( 49 )/-2  /,Ie( 50 )/-2  /,Ie( 51 )/ 0  /
      Data Ie( 52 )/-2  /,Ie( 53 )/ 1  /,Ie( 54 )/-2  /
```

```
Data Ie( 55 )/ 2  /,Ie( 56 )/-1  /,Ie( 57 )/ 6  /
Data Ie( 58 )/-2  /,Ie( 59 )/ 0  /,Ie( 60 )/-3  /
Data Ie( 61 )/ 2  /,Ie( 62 )/-2  /,Ie( 63 )/-1  /
Data Ie( 64 )/-2  /,Ie( 65 )/-2  /,Ie( 66 )/-1  /
Data Ie( 67 )/-2  /,Ie( 68 )/ 0  /,Ie( 69 )/ 0  /
Data Ie( 70 )/ 2  /,Ie( 71 )/ 1  /,Ie( 72 )/ 2  /
Data Ie( 73 )/ 2  /,Ie( 74 )/ 1  /,Ie( 75 )/ 2  /
Data Ie( 76 )/ 0  /,Ie( 77 )/ 3  /,Ie( 78 )/-1  /
Data Ie( 79 )/ 3  /,Ie( 80 )/ 0  /,Ie( 81 )/ 3  /
Data Ie( 82 )/ 1  /,Ie( 83 )/-18 /,Ie( 84 )/-21 /
Data Ie( 85 )/ 16 /,Ie( 86 )/ 7  /,Ie( 87 )/ 19 /
Data Ie( 88 )/ 12 /,Ie( 89 )/ 0  /,Ie( 90 )/ 1  /
Data Ie( 91 )/-1  /,Ie( 92 )/ 0  /,Ie( 93 )/-2  /
Data Ie( 94 )/-5  /,Ie( 95 )/-2  /,Ie( 96 )/-3  /
Data Ie( 97 )/-2  /,Ie( 98 )/ 1  /,Ie( 99 )/-1  /
Data Ie( 100 )/ 2  /,Ie( 101 )/ 7  /,Ie( 102 )/ 4  /
Data Ie( 103 )/ 1  /,Ie( 104 )/-3  /,Ie( 105 )/ 2  /
Data Ie( 106 )/-3  /,Ie( 107 )/ 5  /,Ie( 108 )/-7  /
Data Ie( 109 )/ 13 /,Ie( 110 )/ 25 /,Ie( 111 )/-2  /
Data Ie( 112 )/-4  /,Ie( 113 )/-1  /,Ie( 114 )/-5  /
Data Ie( 115 )/ 0  /,Ie( 116 )/-4  /,Ie( 117 )/ 1  /
Data Ie( 118 )/-5  /,Ie( 119 )/ 2  /,Ie( 120 )/-4  /
Data Ie( 121 )/ 5  /,Ie( 122 )/ 25 /,Ie( 123 )/ 9  /
Data Ie( 124 )/ 15 /,Ie( 125 )/ 0  /,Ie( 126 )/-12 /
Data Ie( 127 )/-5  /,Ie( 128 )/ 9  /,Ie( 129 )/ 10 /
Data Ie( 130 )/-6  /,Ie( 131 )/ 0  /,Ie( 132 )/ 6  /
Data Ie( 133 )/-10 /,Ie( 134 )/-6  /,Ie( 135 )/ 9  /
Data Ie( 136 )/ 18 /,Ie( 137 )/ 0  /,Ie( 138 )/-18 /
Data Ie( 139 )/-9  /,Ie( 140 )/ 9  /,Ie( 141 )/ 18 /
Data Ie( 142 )/ 0  /,Ie( 143 )/ 10 /,Ie( 144 )/ 1  /
Data Ie( 145 )/ 0  /,Ie( 146 )/ 9  /,Ie( 147 )/ 18 /
Data Ie( 148 )/ 25 /,Ie( 149 )/-18 /,Ie( 150 )/-32 /
Data Ie( 151 )/ 8  /,Ie( 152 )/ 21 /,Ie( 153 )/ 2  /
Data Ie( 154 )/ 3  /,Ie( 155 )/ 1  /,Ie( 156 )/ 5  /
Data Ie( 157 )/ 0  /,Ie( 158 )/ 3  /,Ie( 159 )/-1  /
Data Ie( 160 )/ 5  /,Ie( 161 )/-2  /,Ie( 162 )/ 3  /
Data Ie( 163 )/ 5  /,Ie( 164 )/ 17 /,Ie( 165 )/ 3  /
Data Ie( 166 )/ 3  /,Ie( 167 )/ 0  /,Ie( 168 )/-21 /
Data Ie( 169 )/ 3  /,Ie( 170 )/ 16 /,Ie( 171 )/ 4  /
Data Ie( 172 )/ 0  /,Ie( 173 )/-10 /,Ie( 174 )/-10 /
Data Ie( 175 )/ 4  /,Ie( 176 )/ 21 /,Ie( 177 )/ 11 /
Data Ie( 178 )/ 0  /,Ie( 179 )/-6  /,Ie( 180 )/-8  /
Data Ie( 181 )/-1  /,Ie( 182 )/-3  /,Ie( 183 )/-3  /
Data Ie( 184 )/ 0  /,Ie( 185 )/ 17 /,Ie( 186 )/ 7  /
Data Ie( 187 )/-15 /,Ie( 188 )/ 0  /,Ie( 189 )/ 10 /
Data Ie( 190 )/ 14 /,Ie( 191 )/ 14 /,Ie( 192 )/ 21 /
Data Ie( 193 )/-10 /,Ie( 194 )/ 0  /,Ie( 195 )/-1  /
Data Ie( 196 )/-9  /,Ie( 197 )/ 15 /,Ie( 198 )/ 18 /
Data Ie( 199 )/ 0  /,Ie( 200 )/-5  /,Ie( 201 )/ 1  /
Data Ie( 202 )/-4  /,Ie( 203 )/ 1  /,Ie( 204 )/ 0  /
Data Ie( 205 )/ 2  /,Ie( 206 )/ 2  /,Ie( 207 )/ 1  /
```

```
Data Ie( 208 )/ 3  /,Ie( 209 )/-1  /,Ie( 210 )/ 3  /
Data Ie( 211 )/ 6  /,Ie( 212 )/ 0  /,Ie( 213 )/ 10 /
Data Ie( 214 )/ 21 /,Ie( 215 )/-14 /,Ie( 216 )/ 0  /
Data Ie( 217 )/ 7  /,Ie( 218 )/ 21 /,Ie( 219 )/ 4  /
Data Ie( 220 )/-1  /,Ie( 221 )/ 4  /,Ie( 222 )/ 1  /
Data Ie( 223 )/ 15 /,Ie( 224 )/ 14 /,Ie( 225 )/ 10 /
Data Ie( 226 )/ 13 /,Ie( 227 )/ 17 /,Ie( 228 )/ 18 /
Data Ie( 229 )/-16 /,Ie( 230 )/-9  /,Ie( 231 )/ 16 /
Data Ie( 232 )/-9  /,Ie( 233 )/ 0  /,Ie( 234 )/ 12 /
Data Ie( 235 )/-18 /,Ie( 236 )/-6  /,Ie( 237 )/ 1  /
Data Ie( 238 )/ 18 /,Ie( 239 )/-4  /,Ie( 240 )/-2  /
Data Ie( 241 )/ 14 /,Ie( 242 )/ 13 /,Ie( 243 )/-5  /
Data Ie( 244 )/ 8  /,Ie( 245 )/ 7  /,Ie( 246 )/ 11 /
Data Ie( 247 )/-1  /,Ie( 248 )/-8  /,Ie( 249 )/-2  /
Data Ie( 250 )/ 13 /,Ie( 251 )/ 8  /,Ie( 252 )/-21 /
Data Ie( 253 )/-13 /,Ie( 254 )/ 7  /,Ie( 255 )/ 10 /
Data Ie( 256 )/ 0  /,Ie( 257 )/ 2  /,Ie( 258 )/ 11 /
Data Ie( 259 )/ 9  /,Ie( 260 )/ 0  /,Ie( 261 )/-9  /
Data Ie( 262 )/ 0  /,Ie( 263 )/ 0  /,Ie( 264 )/ 21 /
Data Ie( 265 )/ 17 /,Ie( 266 )/ 16 /,Ie( 267 )/ 7  /
Data Ie( 268 )/ 0  /,Ie( 269 )/-7  /,Ie( 270 )/ 0  /
Data Ie( 271 )/ 16 /,Ie( 272 )/ 21 /,Ie( 273 )/-13 /
Data Ie( 274 )/ 0  /,Ie( 275 )/ 13 /,Ie( 276 )/ 0  /
Data Ie( 277 )/-13 /,Ie( 278 )/ 11 /,Ie( 279 )/ 8  /
Data Ie( 280 )/ 0  /,Ie( 281 )/-13 /,Ie( 282 )/-10 /
Data Ie( 283 )/-5  /,Ie( 284 )/ 0  /,Ie( 285 )/ 2  /
Data Ie( 286 )/ 21 /,Ie( 287 )/-14 /,Ie( 288 )/ 11 /
Data Ie( 289 )/ 13 /,Ie( 290 )/ 21 /,Ie( 291 )/ 0  /
Data Ie( 292 )/-16 /,Ie( 293 )/-14 /,Ie( 294 )/-14 /
Data Ie( 295 )/ 5  /,Ie( 296 )/ 5  /,Ie( 297 )/ 9  /
Data Ie( 298 )/-12 /,Ie( 299 )/ 3  /,Ie( 300 )/ 21 /
Data Ie( 301 )/ 12 /,Ie( 302 )/ 0  /,Ie( 303 )/ 14 /
Data Ie( 304 )/-21 /,Ie( 305 )/ 0  /,Ie( 306 )/ 5  /
Data Ie( 307 )/ 3  /,Ie( 308 )/-17 /,Ie( 309 )/ 6  /
Data Ie( 310 )/-6  /,Ie( 311 )/ 7  /,Ie( 312 )/-11 /
Data Ie( 313 )/ 16 /,Ie( 314 )/ 18 /,Ie( 315 )/-7  /
Data Ie( 316 )/ 21 /,Ie( 317 )/ 0  /,Ie( 318 )/-15 /
Data Ie( 319 )/ 0  /,Ie( 320 )/ 15 /,Ie( 321 )/ 1  /
Data Ie( 322 )/ 21 /,Ie( 323 )/-1  /,Ie( 324 )/ 21 /
Data Ie( 325 )/ 5  /,Ie( 326 )/-21 /,Ie( 327 )/-14 /
Data Ie( 328 )/-21 /,Ie( 329 )/ 8  /,Ie( 330 )/-10 /
Data Ie( 331 )/ 0  /,Ie( 332 )/-11 /,Ie( 333 )/-8  /
Data Ie( 334 )/-10 /,Ie( 335 )/ 0  /,Ie( 336 )/ 11 /
Data Ie( 337 )/ 15 /,Ie( 338 )/ 11 /,Ie( 339 )/ 15 /
Data Ie( 340 )/ 21 /,Ie( 341 )/ 3  /,Ie( 342 )/ 8  /
Data Ie( 343 )/ 0  /,Ie( 344 )/-17 /,Ie( 345 )/-3  /
Data Ie( 346 )/ 14 /,Ie( 347 )/ 9  /,Ie( 348 )/ 17 /
Data Ie( 349 )/ 0  /,Ie( 350 )/ 10 /,Ie( 351 )/ 0  /
Data Ie( 352 )/-10 /,Ie( 353 )/-1  /,Ie( 354 )/-7  /
Data Ie( 355 )/ 11 /,Ie( 356 )/ 21 /,Ie( 357 )/ 4  /
Data Ie( 358 )/ 4  /,Ie( 359 )/ 7  /,Ie( 360 )/-8  /
```

```
      Data Ie( 361 )/-2  /,Ie( 362 )/ 14 /,Ie( 363 )/ 4  /
      Data Ie( 364 )/ 14 /,Ie( 365 )/ 8  /,Ie( 366 )/ 14 /
      Data Ie( 367 )/ 3  /,Ie( 368 )/ 14 /,Ie( 369 )/ 0  /
      Data Ie( 370 )/ 18 /,Ie( 371 )/ 6  /,Ie( 372 )/ 14 /
      Data Ie( 373 )/ 0  /,Ie( 374 )/ 8  /,Ie( 375 )/ 5  /
      Data Ie( 376 )/-1  /,Ie( 377 )/ 0  /,Ie( 378 )/-1  /
      Data Ie( 379 )/-8  /,Ie( 380 )/ 14 /,Ie( 381 )/ 6  /
      Data Ie( 382 )/-14 /,Ie( 383 )/ 1  /,Ie( 384 )/ 14 /
      Data Ie( 385 )/ 4  /,Ie( 386 )/-14 /,Ie( 387 )/ 11 /
      Data Ie( 388 )/-14 /,Ie( 389 )/ 0  /,Ie( 390 )/ 14 /
      Data Ie( 391 )/-11 /,Ie( 392 )/-14 /,Ie( 393 )/-6  /
      Data Ie( 394 )/-14 /,Ie( 395 )/ 9  /,Ie( 396 )/ 25 /
      Data Ie( 397 )/ 0  /,Ie( 398 )/-32 /
C
C
      Exit_Code = 0
      Icomp=201

      If (G_Char_Gen_State .Eq. 0) Then
            Ipen=3
            Iptr=0
      Else
            Ipen = 2
            Goto 16
      Endif

      Isym=Char
5     If ((Isym .Ge. 32) .And. (Isym .Le. 124)) Go To 10
      If ((Isym .Ge. 201) .And. (Isym .Le. 225)) Isym = Isym - 76
C
C OBTAIN INDEXES (BEGIN/END) TO THE CHARACTER TABLE
C
10    Ifirst=In(Isym-31)+1
      Ilast=In(Isym-31+1)
C
C LOOP THRU CHARACTER TABLE
C
15    I = Ifirst
16    If (I .Gt. Ilast) Goto 25
      J=Ic(I)
C
C IS THE CHARACTER TABLE ELEMENT A COMPOUND ?
C
      If (J .Ge. Icomp) Go To 30
C
C IS THE CHARACTER TABLE ELEMENT A PEN UP COMMAND ?
C
      If (J .Eq. 1) Go To 18
C
C OBTAIN THE X,Y BASIC ELEMENT PAIR FROM THE BASIC ELEMENTS TABLE
C
```

```
        Ym=Float(Ie(J*2+1-3))
        Xm=Float(Ie(J*2-3))
C
        Ypos = O_Char_Scale/21.0 * ( Ym*G_Cos_Crot + Xm*G_Sin_Crot )
        Xpos = O_Char_Aspect_Ratio*O_Char_Scale/21.0 *
     1       ( Xm*G_Cos_Crot - Ym*G_Sin_Crot )
C
        G_Char_Gen_State = Ipen
        I = I + 1
        Goto 45

18      Ipen = 3
        I = I + 1
        Goto 16
C
C ANY LOOP VARIABLES ON THE STACK ?
C
25      If (Iptr .Gt. 0) Go To 40

        G_Char_Gen_State = -1
        Go To 45
C
C DO WE NEED TO SAVE LOOP VARIABLES ?
C
30      If ( I .Eq. Ilast) Go To 35
        Index1=I+1
        Index2=Ilast
C
C PUT THE LOOP VARIABLES ON THE STACK
C
        Call Cgen_Push(Index1,Index2)
35      Isym=J
        Go To 5
C
C RETRIEVE THE LOOP VARIABLES FROM THE STACK
C
40      Call Cgen_Pop(Index1,Index2)
        Ifirst=Index1
        Ilast=Index2
        Go To 15
C
C RETURN INFO REGARDING THE FIRST MOVE ONLY
C
45      Exit_Code = G_Char_Gen_State

        Return
        End
```

```
      Subroutine Swchar_Vec(Char,Action,Xrel,Yrel)

C  This routine generates character sets in software and returns the next
C  draw or move within the character currently being generated.  That is, to
C  generate one character this routine must be called many times, where each
C  call returns the next graphic element within the character.  The routine
C  keeps track of its "place" within the character internally.

C  Parameters:

C  CHAR - b [i] = The character to generate.
C  ACTION - i [o] = The  way the calling routine should treat the coords
C                       we return:   2 = Coords describe a relative move
C                                    3 = Relative draw
C                                   -1 = Relative move, char is done.  This
C                                    positions us at the start of the next char.
C  XREL, YREL - i [o] = The coords we return for the next pen position.

C  There are several parameters affecting the character generator.  The user
C  is able to scale the characters with the parameter O_Char_Scale, and to
C  rotate them with the parameter O_Char_Rotation.  He is also able to change
C  the current ratio of X::Y size for the characters with the parameter
C  O_Char_Aspect_Ratio.  This scaling and rotation is done below right before
C  we return coords.

C  The coordinate system in which the characters are drawn is a bit confusing.
C  The original Plot21 software scaled all the coordinates it generates by
C  1/21.  This apparently brings all the coordinates into the range -1. < x <
C  1.  All the coordinates are relative moves/draws.  To make the characters
C  the same size as those generated by the Tektronix 4010 hardware I scale the
C  coordinates by the factor PLOT21_TO_PLOT10.  So when O_Char_Scale is 1 the
C  characters should be the same size as those generated by Plot10. This
C  routine returns integer coordinates.  To avoid the great degradation of
C  character quality which occurs when one simply converts the real coords to
C  integers due to the accumulation of rounding errors, we keep track of the
C  *desired* real position on the screen, and constantly compute integer moves
C  to bring the actual beam position as close as possible to the desired real
C  position.  This does improve character quality considerably.

C      This character generator was ripped out of the Hewlett-Packard
C  Plot-21 software.  The following comments describe the lower level
C  routines which provide the actual character generation data:

C     ***************************************************
C     *                                                 *
C     *  PHILOSOPHY OF THE CHARACTER GENERATOR :        *
C     *                                                 *
C     *                                                 *
C     *                                                 *
C     *  THE CHARACTER GENERATOR CONSISTS OF FOUR       *
C     *  CHARACTER SETS : ENHANCED STICK, SCRIPT,       *
```

```
C     *  TRIPLEX ROMAN AND GOTHIC.  EACH SET CONSISTS   *
C     *  OF FOUR TABLES :                               *
C     *                                                 *
C     *                                                 *
C     *  INDEX TABLE - COMPOSED OF BEGIN/END POINTERS   *
C     *                TO THE CHARACTER TABLE.          *
C     *                                                 *
C     *                                                 *
C     *  CHARACTER TABLE - COMPOSED OF POINTERS TO X,Y  *
C     *                    DATA IN THE BASIC ELEMENTS   *
C     *                    TABLE.                       *
C     *                                                 *
C     *  BASIC ELEMENTS TABLE - COMPOSED OF X,Y DATA    *
C     *                         USED IN DRAWING THE     *
C     *                         CHARACTERS.             *
C     *                                                 *
C     *  SPACE TABLE - COMPOSED OF SPACING DATA FOR     *
C     *                EACH CHARACTER.                  *
C     *                                                 *
C     *                                                 *
C     *  THE CHARACTER GENERATOR IS DESIGNED SUCH THAT  *
C     *  A SPECIFIC CHARACTER IS PRODUCED BY COMBINING  *
C     *  A NUMBER OF SUB-ELEMENTS.  THIS CONCEPT SAVES  *
C     *  TABLE SPACE AND PROVIDES FOR MORE EFFICIENT    *
C     *  CHARACTER GENERATION.                          *
C     *                                                 *
C     *  SINCE THE CHARACTER GENERATION PROCESS IS      *
C     *  ONE THAT INVOLVES RECURSION, IT IS NECESSARY   *
C     *  TO SIMULATE A STACK, SO THAT NECESSARY LOOP    *
C     *  VARIABLES MAY BE SAVED AND RETRIEVED DURING    *
C     *  THE PRODUCTION OF EACH CHARACTER.              *
C     *                                                 *
C     *  THE ASCII DECIMAL EQUIVALENT REPRESENTATION OF *
C     *  THE DESIRED CHARACTER IS USED TO LOCATE AN     *
C     *  ENTRY INTO THE INDEX TABLE.                    *
C     *                                                 *
C     *  THE INDEX TABLE PROVIDES A SET OF BEGIN/END    *
C     *  PONTERS TO THE CHARACTER TABLE.                *
C     *                                                 *
C     *  THE CHARACTER GENERATOR THEN LOOPS THRU THESE  *
C     *  VALUES FROM BEGIN TO END.  THIS RANGE OF VALUES*
C     *  POINTS TO SUB-ELEMENTS IN THE ELEMENTS TABLE   *
C     *  FOR ALL OF THE X,Y PAIRS REQUIRED TO CREATE    *
C     *  THE SPECIFIED CHARACTER.                       *
C     *                                                 *
C     *  A VALUE OF 1 IN THE CHARACTER TABLE SIGNALS A  *
C     *  PEN UP COMMAND.                                *
C     *                                                 *
C     *  INFORMATION CONCERNING THE SPACING BETWEEN     *
C     *  CHARACTERS IS OBTAINED FROM BOTH THE CHARACTER *
C     *  GENERATOR AND SPACE TABLE.                     *
```

```
C     *                                                    *
C     ******************************************************

        Implicit Integer (A-Z)

        Byte Char

        Include 'Symvecmem.Inc'
        Include 'Options.Inc'

        Logical *1 Valid

        Real Xdisp, Ydisp, Xdelta, Xm, Ym
        Real Asign

C  Are we starting the current character ?

        If (G_Char_Gen_State .Ne. -1) Goto 100

C  Yes.  Tell the actual generating routines that they are starting.
C  Next set VALID if the current char is either a space or a character
C  which is not in the current font.  Such a char is treated like a space.
C  If it's not a space get spacing information from the character generating
C  system.  Such information tells not only how wide the cell occupied by
C  the character is, but how far to the right of the cell's left hand side
C  the character actually begins.

        G_Char_Gen_State = 0
        Ichar = Char

        Call Char_Check(Ichar,50+O_Char_Set,Valid)

C  If we've got a char not in the set, create spacing information for a
C  character of width 18, which seems to be the standard.  Clearly it doesn't
C  matter what the offset of a space into the cell is.

        If (.Not. Valid) Then
                G_Xmin = 1. * O_Char_Aspect_Ratio * O_Char_Scale / 21.
                G_Xmax = 19. * O_Char_Scale / 21.
                G_Xprev = 0.
                G_Yprev = 0.
                Goto 110
        Endif

C  Generate spacing info for a character in a set.  See one of the routines
C  below for more info.

        If (O_Char_Set .Eq. 1) Call Space1(Ichar,Ispl,Isp2)
        If (O_Char_Set .Eq. 2) Call Space2(Ichar,Ispl,Isp2)
        If (O_Char_Set .Eq. 3) Call Space3(Ichar,Ispl,Isp2)
        If (O_Char_Set .Eq. 4) Call Space4(Ichar,Ispl,Isp2)
```

```
C  Calculate the character's horizontal offset into the cell.

        G_Xmin = O_Char_Aspect_Ratio * O_Char_Scale * Float(Isp1) / 21.
        G_Xmax = O_Char_Aspect_Ratio * O_Char_Scale * Float(Isp2) / 21.
        Xdisp = -G_Xmin * G_Cos_Crot
        Ydisp = -G_Xmin * G_Sin_Crot

C  g_xloc,g_yloc maintain the current real position.  g_xprev,g_yprev contain
C  the accumulation of all the moves/draws, in other words, the current beam
C  displacement from its position before we started drawing this char.
C  g_xloc,g_yloc start at 0,0.  Now we return the move to offset the character
C  into its cell.

        Action = Move_Code
        G_Xloc = Xdisp * Plot21_To_Plot10
        G_Yloc = Ydisp * Plot21_To_Plot10
        Xrel = G_Xloc + .5*Asign(G_Xloc)
        Yrel = G_Yloc + .5*Asign(G_Xloc)
        G_Xprev = Xrel
        G_Yprev = Yrel
        Goto 999

100     Continue

C  Compute the next move/draw for the current character in the selected
C  character set.

        If (O_Char_Set .Eq. 1) Call Stick(Char,Xm,Ym,Exit_Code)
        If (O_Char_Set .Eq. 2) Call Script(Char,Xm,Ym,Exit_Code)
        If (O_Char_Set .Eq. 3) Call Triplx(Char,Xm,Ym,Exit_Code)
        If (O_Char_Set .Eq. 4) Call Gothic(Char,Xm,Ym,Exit_Code)
C
C  Does the character generator contain no more info for the current char ?

        If (Exit_Code .Eq. -1) Goto 110

C  Set the action code according to what the low level routines told us.

        If (Exit_Code .Eq. 2) Then
                Action = Draw_Code
        Else
                Action = Move_Code
        Endif

C  Update the current position by the coords returned by the generator, and
C  then compute the integer move needed to get us there from the current beam
C  position.

        G_Xloc = G_Xloc + Xm * Plot21_To_Plot10
        G_Yloc = G_Yloc + Ym * Plot21_To_Plot10
```

```
            Xrel = G_Xloc - G_Xprev + .5*Asign(G_Xloc-G_Xprev)
            Yrel = G_Yloc - G_Yprev + .5*Asign(G_Yloc-G_Yprev)
            G_Xprev = G_Xprev + Xrel
            G_Yprev = G_Yprev + Yrel
            Goto 999

C   There are no more graphic elements in the current char.  What we do now is
C   compute a move to put the beam at the bottom right hand corner of the cell
C   for this character - or in place for the next character.  XDELTA will
C   contain the width of the character cell.  Thus the final position of the
C   beam should be at (XDELTA,0) in our zero-origin, un-rotated, unscaled
C   coord system.

110         Xdelta = G_Xmax - G_Xmin
            G_Xloc = Xdelta*G_Cos_Crot * Plot2l_To_Plot10
            G_Yloc = Xdelta*G_Sin_Crot * Plot2l_To_Plot10
            Xrel = G_Xloc - G_Xprev + .5*Asign(G_Xloc-G_Xprev)
            Yrel = G_Yloc - G_Yprev + .5*Asign(G_Yloc-G_Yprev)
            Action = Done_Code

999         Return
            End
```

```
      Subroutine Toprinter_Vms(Bstring,Slen)

C  Machine dependent stream I/O routine.

C  Sends a string of byte data to the printer with no carriage control (no CR
C  or LF is generated by VMS).  So only the characters in the string are
C  written to the printer.

C  Parameters:

C  Bstring(Slen) - b [i] = The string to print.
C  Slen - i [i] = The length of the string.

      Implicit Integer (A-Z)

      Byte Bstring(Slen)

      Include 'Vmsio.Inc'

C  We're doing a no-wait Queue I/O request associated with system event flag
C  1.  Note many parameters must be passed by value rather than by reference
C  (the normal fortran method).  The fact that we're doing a no-wait Qio may
C  serve to make this form of i/o more efficient than Fortran i/o since the
C  program can go on to do more computation and not wait for the printer or
C  for the i/o to finish once it's issued the i/o request.

C  We don't issue this error message through the standard error system because
C  there would be too many routines to change that call this one.

      If ( V_Success .Ne. Sys$Qio(%Val(V_Event_Flag1),%Val(V_Channel),
     1  %Val(V_Io$Writevblk),,,,Bstring,%Val(Slen),,
     2  %Val(V_No_Carriage_Control),,) )
     3  Stop 'Toprinter_Vms-F-Error Returned From Sys$Qio.'

      Return
      End
```

```
      Subroutine Tostring(Number,String)

C  Converts a binary integer number to a character representation of the
C  number.  We use Fortran internal I/O with I5 format.

C  Parameters:

C  Number - i [i] = The number of interest.
C  String - c** [o] = The string we form.

      Implicit Integer (A-Z)

      Character *(*) String

C  Use fortran internal io.

      Write (String,900) Number
900   Format (I5)

999   Return
      End
```

```
        Character *1 Function Toupper(Ch)

C  Converts a character to upper case.  We assume we're in the ascii character
C  set.

C  Parameters:

C  Ch - c1 [i] = The character to convert.

        Implicit Integer (A-Z)

        Character *1 Ch

        Ich = Ichar(Ch)

        If (Ich .Ge. 97 .And. Ich .Le. 122) Then
                Toupper = Char(Ich - 32)
        Else
                Toupper = Ch
        Endif

999     Return
        End
```

```
      Subroutine Track(Next_Dot_Dis,Add,Band_Head_Pos)

C  This routine maintains the values of the variables Next_Dot_Dis and
C  Band_Head_Pos as we move through the image.  These variables relate our
C  position in the image to a position in the band.  More specifically,
C  Band_Head_Pos points to that row in the band which maps to a position
C  within the image which is closest to but not above the position of the
C  current pin within the image.  Next_Dot_Dis is the number of veps between
C  the current pin and the mapped position of that row.  Thus it's zero if the
C  pin and mapped row are at the same position, and it's some positive number
C  when the row maps a bit below the pin.  This routine is called whenever the
C  print head is moved or Make_Line proceeds to a new pin in the head.

C  Parameters:

C  Next_Dot_Dis, Band_Head_Pos - i [io] = Explained above.
C  Add - i [i] = The number of veps by which the print head moved, or the
C       distance between pins on the head.

      Implicit Integer (A-Z)

      Include 'Options.Inc'

C  This may relate to a situation which will not exist in this modern version
C  of the program, but we'll leave it in anyway.

      If (Next_Dot_Dis .Lt. 0) Then
            Next_Dot_Dis = Next_Dot_Dis + Add
            Goto 999
      Endif

C  If we've moved a distance smaller than the distance to the mapped row
C  position, we just subtract from the distance remaining to the row.
C  Otherwise we've moved to either on top of a row, or between two rows
C  further down.  We have to determine which, and how many rows we've passed
C  in the process.

      If (Add .Lt. Next_Dot_Dis) Then
            Next_Dot_Dis = Next_Dot_Dis - Add
      Else

C  Distance we move past the Band_Head_Pos row.

            Dis = Add - Next_Dot_Dis

C  We know (since we're in this clause) that we're moving past the BHP row,
C  so make a note of it.

            If (Next_Dot_Dis .Gt. 0) Band_Head_Pos = Band_Head_Pos + 1

C  We may be moving past more than one row.
```

```
                Band_Head_Pos = Band_Head_Pos + Dis / O_Vertical_Dot_Spacing

C   Compute the distance to the next row if we're not on top of a row.

                Remainder = Dis - (Dis / O_Vertical_Dot_Spacing *
        1                         O_Vertical_Dot_Spacing)
                If (Remainder .Ne. 0) Then
                        Next_Dot_Dis = O_Vertical_Dot_Spacing - Remainder
                Else
                        Next_Dot_Dis = 0
                Endif
        Endif

999     Return
        End
```

```
      Subroutine Trip (Ipair,Nx,Ny)
      Dimension Ie(816)
C
C     ********************************************************
C     *          BASIC ELEMENTS TABLE FOR TRIPLEX CHAR SET *
C     ********************************************************
C
      Data Ie( 1 )/ 9  /,Ie( 2 )/ 10 /,Ie( 3 )/-1  /
      Data Ie( 4 )/ 8  /,Ie( 5 )/ 0  /,Ie( 6 )/ 2  /
      Data Ie( 7 )/ 1  /,Ie( 8 )/ 1  /,Ie( 9 )/ 0  /
      Data Ie( 10 )/-14 /,Ie( 11 )/ 1  /,Ie( 12 )/ 0  /
      Data Ie( 13 )/ 0  /,Ie( 14 )/ 14 /,Ie( 15 )/-1  /
      Data Ie( 16 )/ 0  /,Ie( 17 )/ 1  /,Ie( 18 )/-1  /
      Data Ie( 19 )/ 0  /,Ie( 20 )/-2  /,Ie( 21 )/-1  /
      Data Ie( 22 )/-8  /,Ie( 23 )/-1  /,Ie( 24 )/-7  /
      Data Ie( 25 )/-1  /,Ie( 26 )/-1  /,Ie( 27 )/ 0  /
      Data Ie( 28 )/-1  /,Ie( 29 )/ 0  /,Ie( 30 )/ 1  /
      Data Ie( 31 )/-1  /,Ie( 32 )/ 1  /,Ie( 33 )/ 4  /
      Data Ie( 34 )/ 14 /,Ie( 35 )/ 2  /,Ie( 36 )/ 6  /
      Data Ie( 37 )/ 0  /,Ie( 38 )/-6  /,Ie( 39 )/ 1  /
      Data Ie( 40 )/ 6  /,Ie( 41 )/ 8  /,Ie( 42 )/-6  /
      Data Ie( 43 )/ 7  /,Ie( 44 )/ 25 /,Ie( 45 )/ 0  /
      Data Ie( 46 )/-29 /,Ie( 47 )/ 4  /,Ie( 48 )/ 29 /
      Data Ie( 49 )/ 4  /,Ie( 50 )/ 20 /,Ie( 51 )/ 2  /
      Data Ie( 52 )/ 0  /,Ie( 53 )/-1  /,Ie( 54 )/ 2  /
      Data Ie( 55 )/-3  /,Ie( 56 )/ 1  /,Ie( 57 )/-4  /
      Data Ie( 58 )/ 0  /,Ie( 59 )/-3  /,Ie( 60 )/-1  /
      Data Ie( 61 )/-2  /,Ie( 62 )/-2  /,Ie( 63 )/ 0  /
      Data Ie( 64 )/-3  /,Ie( 65 )/ 1  /,Ie( 66 )/-2  /
      Data Ie( 67 )/ 3  /,Ie( 68 )/-2  /,Ie( 69 )/ 6  /
      Data Ie( 70 )/-2  /,Ie( 71 )/ 2  /,Ie( 72 )/-1  /
      Data Ie( 73 )/-1  /,Ie( 74 )/-2  /,Ie( 75 )/-11 /
      Data Ie( 76 )/ 14 /,Ie( 77 )/-11 /,Ie( 78 )/ 13 /
      Data Ie( 79 )/ 18 /,Ie( 80 )/ 12 /,Ie( 81 )/-2  /
      Data Ie( 82 )/-5  /,Ie( 83 )/-2  /,Ie( 84 )/-3  /
      Data Ie( 85 )/-2  /,Ie( 86 )/-1  /,Ie( 87 )/-2  /
      Data Ie( 88 )/ 1  /,Ie( 89 )/ 0  /,Ie( 90 )/ 3  /
      Data Ie( 91 )/ 1  /,Ie( 92 )/ 2  /,Ie( 93 )/ 6  /
      Data Ie( 94 )/ 4  /,Ie( 95 )/ 2  /,Ie( 96 )/ 2  /
      Data Ie( 97 )/ 1  /,Ie( 98 )/-3  /,Ie( 99 )/ 2  /
      Data Ie( 100 )/-3  /,Ie( 101 )/ 4  /,Ie( 102 )/-5  /
      Data Ie( 103 )/ 3  /,Ie( 104 )/-3  /,Ie( 105 )/-17 /
      Data Ie( 106 )/-2  /,Ie( 107 )/ 6  /,Ie( 108 )/ 5  /
      Data Ie( 109 )/ 1  /,Ie( 110 )/ 4  /,Ie( 111 )/-1  /
      Data Ie( 112 )/ 4  /,Ie( 113 )/-1  /,Ie( 114 )/-4  /
      Data Ie( 115 )/-11 /,Ie( 116 )/-1  /,Ie( 117 )/ 4  /
      Data Ie( 118 )/ 4  /,Ie( 119 )/-3  /,Ie( 120 )/ 6  /
      Data Ie( 121 )/ 1  /,Ie( 122 )/-4  /,Ie( 123 )/ 3  /
      Data Ie( 124 )/-4  /,Ie( 125 )/ 9  /,Ie( 126 )/ 14 /
      Data Ie( 127 )/ 12 /,Ie( 128 )/ 25 /,Ie( 129 )/-2  /
      Data Ie( 130 )/-4  /,Ie( 131 )/-1  /,Ie( 132 )/-5  /
```

```
Data Ie( 133 )/ 0  /,Ie( 134 )/-4  /,Ie( 135 )/ 1  /
Data Ie( 136 )/-5  /,Ie( 137 )/ 2  /,Ie( 138 )/-4  /
Data Ie( 139 )/ 2  /,Ie( 140 )/-2  /,Ie( 141 )/-4  /
Data Ie( 142 )/ 26 /,Ie( 143 )/-1  /,Ie( 144 )/-3  /
Data Ie( 145 )/ 2  /,Ie( 146 )/ 24 /,Ie( 147 )/-1  /
Data Ie( 148 )/-6  /,Ie( 149 )/ 1  /,Ie( 150 )/-6  /
Data Ie( 151 )/ 6  /,Ie( 152 )/ 25 /,Ie( 153 )/ 4  /
Data Ie( 154 )/ 26 /,Ie( 155 )/-2  /,Ie( 156 )/ 24 /
Data Ie( 157 )/ 9  /,Ie( 158 )/ 9  /,Ie( 159 )/ 0  /
Data Ie( 160 )/ 12 /,Ie( 161 )/ 2  /,Ie( 162 )/-10 /
Data Ie( 163 )/ 2  /,Ie( 164 )/ 10 /,Ie( 165 )/ 5  /
Data Ie( 166 )/-9  /,Ie( 167 )/-10 /,Ie( 168 )/ 6  /
Data Ie( 169 )/-10 /,Ie( 170 )/ 4  /,Ie( 171 )/ 10 /
Data Ie( 172 )/ 6  /,Ie( 173 )/-8  /,Ie( 174 )/-6  /
Data Ie( 175 )/ 10 /,Ie( 176 )/ 4  /,Ie( 177 )/ 10 /
Data Ie( 178 )/ 1  /,Ie( 179 )/ 0  /,Ie( 180 )/ 17 /
Data Ie( 181 )/ 0  /,Ie( 182 )/-17 /,Ie( 183 )/ 8  /
Data Ie( 184 )/ 8  /,Ie( 185 )/-17 /,Ie( 186 )/ 0  /
Data Ie( 187 )/ 17 /,Ie( 188 )/ 0  /,Ie( 189 )/ 8  /
Data Ie( 190 )/-4  /,Ie( 191 )/ 2  /,Ie( 192 )/ 1  /
Data Ie( 193 )/-1  /,Ie( 194 )/ 5  /,Ie( 195 )/ 18 /
Data Ie( 196 )/ 9  /,Ie( 197 )/ 9  /,Ie( 198 )/ 3  /
Data Ie( 199 )/ 1  /,Ie( 200 )/-7  /,Ie( 201 )/ 18 /
Data Ie( 202 )/ 32 /,Ie( 203 )/-18 /,Ie( 204 )/-32 /
Data Ie( 205 )/ 8  /,Ie( 206 )/ 21 /,Ie( 207 )/ 3  /
Data Ie( 208 )/-1  /,Ie( 209 )/-2  /,Ie( 210 )/ 0  /
Data Ie( 211 )/-2  /,Ie( 212 )/ 3  /,Ie( 213 )/ 1  /
Data Ie( 214 )/ 5  /,Ie( 215 )/ 2  /,Ie( 216 )/ 3  /
Data Ie( 217 )/ 3  /,Ie( 218 )/ 1  /,Ie( 219 )/ 0  /
Data Ie( 220 )/-5  /,Ie( 221 )/-3  /,Ie( 222 )/ 19 /
Data Ie( 223 )/ 8  /,Ie( 224 )/ 0  /,Ie( 225 )/ 0  /
Data Ie( 226 )/ 5  /,Ie( 227 )/-3  /,Ie( 228 )/-19 /
Data Ie( 229 )/ 8  /,Ie( 230 )/ 19 /,Ie( 231 )/ 0  /
Data Ie( 232 )/-19 /,Ie( 233 )/ 1  /,Ie( 234 )/ 19 /
Data Ie( 235 )/ 0  /,Ie( 236 )/-18 /,Ie( 237 )/ 0  /
Data Ie( 238 )/ 21 /,Ie( 239 )/-3  /,Ie( 240 )/-3  /
Data Ie( 241 )/-1  /,Ie( 242 )/-17 /,Ie( 243 )/ 10 /
Data Ie( 244 )/ 0  /,Ie( 245 )/-6  /,Ie( 246 )/ 1  /
Data Ie( 247 )/ 3  /,Ie( 248 )/ 2  /,Ie( 249 )/ 3  /
Data Ie( 250 )/ 17 /,Ie( 251 )/ 4  /,Ie( 252 )/ 0  /
Data Ie( 253 )/-3  /,Ie( 254 )/-2  /,Ie( 255 )/-5  /
Data Ie( 256 )/-2  /,Ie( 257 )/ 12 /,Ie( 258 )/ 19 /
Data Ie( 259 )/-3  /,Ie( 260 )/ 8  /,Ie( 261 )/-4  /
Data Ie( 262 )/-2  /,Ie( 263 )/-5  /,Ie( 264 )/-7  /
Data Ie( 265 )/ 5  /,Ie( 266 )/-1  /,Ie( 267 )/-5  /
Data Ie( 268 )/ 2  /,Ie( 269 )/ 5  /,Ie( 270 )/-3  /
Data Ie( 271 )/ 2  /,Ie( 272 )/ 8  /,Ie( 273 )/-3  /
Data Ie( 274 )/ 12 /,Ie( 275 )/ 3  /,Ie( 276 )/ 0  /
Data Ie( 277 )/ 11 /,Ie( 278 )/ 6  /,Ie( 279 )/-11 /
Data Ie( 280 )/ 3  /,Ie( 281 )/ 10 /,Ie( 282 )/ 18 /
Data Ie( 283 )/-11 /,Ie( 284 )/-15 /,Ie( 285 )/ 16 /
```

```
Data Ie( 286 )/ 0  /,Ie( 287 )/-10 /,Ie( 288 )/-6  /
Data Ie( 289 )/-5  /,Ie( 290 )/ 1  /,Ie( 291 )/ 3  /
Data Ie( 292 )/ 3  /,Ie( 293 )/ 1  /,Ie( 294 )/ 3  /
Data Ie( 295 )/-1  /,Ie( 296 )/ 3  /,Ie( 297 )/-2  /
Data Ie( 298 )/ 2  /,Ie( 299 )/-3  /,Ie( 300 )/ 0  /
Data Ie( 301 )/-4  /,Ie( 302 )/-1  /,Ie( 303 )/ 10 /
Data Ie( 304 )/-8  /,Ie( 305 )/-4  /,Ie( 306 )/ 11 /
Data Ie( 307 )/-7  /,Ie( 308 )/ 5  /,Ie( 309 )/ 1  /
Data Ie( 310 )/ 15 /,Ie( 311 )/ 13 /,Ie( 312 )/ 18 /
Data Ie( 313 )/ 1  /,Ie( 314 )/ 9  /,Ie( 315 )/ 9  /
Data Ie( 316 )/ 1  /,Ie( 317 )/-5  /,Ie( 318 )/ 11 /
Data Ie( 319 )/ 0  /,Ie( 320 )/-7  /,Ie( 321 )/ 2  /
Data Ie( 322 )/ 21 /,Ie( 323 )/ 13 /,Ie( 324 )/ 0  /
Data Ie( 325 )/-5  /,Ie( 326 )/-5  /,Ie( 327 )/ 0  /
Data Ie( 328 )/ 4  /,Ie( 329 )/ 4  /,Ie( 330 )/ 5  /
Data Ie( 331 )/-5  /,Ie( 332 )/ 3  /,Ie( 333 )/-2  /
Data Ie( 334 )/-9  /,Ie( 335 )/-4  /,Ie( 336 )/ 19 /
Data Ie( 337 )/ 7  /,Ie( 338 )/ 21 /,Ie( 339 )/-6  /
Data Ie( 340 )/-1  /,Ie( 341 )/-9  /,Ie( 342 )/-10 /
Data Ie( 343 )/-7  /,Ie( 344 )/ 2  /,Ie( 345 )/ 10 /
Data Ie( 346 )/ 8  /,Ie( 347 )/ 0  /,Ie( 348 )/ 9  /
Data Ie( 349 )/-5  /,Ie( 350 )/ 5  /,Ie( 351 )/ 0  /
Data Ie( 352 )/-10 /,Ie( 353 )/ 18 /,Ie( 354 )/ 13 /
Data Ie( 355 )/ 0  /,Ie( 356 )/-8  /,Ie( 357 )/ 4  /
Data Ie( 358 )/ 16 /,Ie( 359 )/ 4  /,Ie( 360 )/ 9  /
Data Ie( 361 )/-7  /,Ie( 362 )/ 21 /,Ie( 363 )/-7  /
Data Ie( 364 )/-20 /,Ie( 365 )/ 7  /,Ie( 366 )/ 18 /
Data Ie( 367 )/ 6  /,Ie( 368 )/-18 /,Ie( 369 )/-5  /
Data Ie( 370 )/ 18 /,Ie( 371 )/-11 /,Ie( 372 )/ 6  /
Data Ie( 373 )/ 9  /,Ie( 374 )/ 0  /,Ie( 375 )/-13 /
Data Ie( 376 )/-6  /,Ie( 377 )/ 6  /,Ie( 378 )/ 0  /
Data Ie( 379 )/ 5  /,Ie( 380 )/ 0  /,Ie( 381 )/ 7  /
Data Ie( 382 )/ 0  /,Ie( 383 )/-16 /,Ie( 384 )/ 1  /
Data Ie( 385 )/ 3  /,Ie( 386 )/ 21 /,Ie( 387 )/ 0  /
Data Ie( 388 )/-21 /,Ie( 389 )/ 1  /,Ie( 390 )/ 20 /
Data Ie( 391 )/-5  /,Ie( 392 )/ 21 /,Ie( 393 )/ 12 /
Data Ie( 394 )/ 0  /,Ie( 395 )/-10 /,Ie( 396 )/-2  /
Data Ie( 397 )/-12 /,Ie( 398 )/ 0  /,Ie( 399 )/ 15 /
Data Ie( 400 )/ 9  /,Ie( 401 )/-3  /,Ie( 402 )/ 9  /
Data Ie( 403 )/-11 /,Ie( 404 )/ 21 /,Ie( 405 )/-2  /
Data Ie( 406 )/-19 /,Ie( 407 )/ 15 /,Ie( 408 )/ 18 /
Data Ie( 409 )/-12 /,Ie( 410 )/ 13 /,Ie( 411 )/ 4  /
Data Ie( 412 )/ 18 /,Ie( 413 )/-10 /,Ie( 414 )/ 0  /
Data Ie( 415 )/ 14 /,Ie( 416 )/ 18 /,Ie( 417 )/-4  /
Data Ie( 418 )/ 18 /,Ie( 419 )/-9  /,Ie( 420 )/ 21 /
Data Ie( 421 )/-1  /,Ie( 422 )/ 6  /,Ie( 423 )/-10 /
Data Ie( 424 )/-10 /,Ie( 425 )/ 0  /,Ie( 426 )/ 8  /
Data Ie( 427 )/-11 /,Ie( 428 )/-7  /,Ie( 429 )/ 0  /
Data Ie( 430 )/ 6  /,Ie( 431 )/-14 /,Ie( 432 )/ 21 /
Data Ie( 433 )/ 6  /,Ie( 434 )/ 1  /,Ie( 435 )/ 4  /
Data Ie( 436 )/-1  /,Ie( 437 )/-8  /,Ie( 438 )/-9  /
```

```
Data Ie( 439 )/ 5  /,Ie( 440 )/ 1  /,Ie( 441 )/-12 /
Data Ie( 442 )/ 10 /,Ie( 443 )/ 7  /,Ie( 444 )/ 7  /
Data Ie( 445 )/-2  /,Ie( 446 )/ 7  /,Ie( 447 )/-7  /
Data Ie( 448 )/ 0  /,Ie( 449 )/ 10 /,Ie( 450 )/ 21 /
Data Ie( 451 )/-17 /,Ie( 452 )/ 21 /,Ie( 453 )/-15 /
Data Ie( 454 )/-10 /,Ie( 455 )/-15 /,Ie( 456 )/-11 /
Data Ie( 457 )/-19 /,Ie( 458 )/ 21 /,Ie( 459 )/ 8  /
Data Ie( 460 )/ 1  /,Ie( 461 )/-14 /,Ie( 462 )/-19 /
Data Ie( 463 )/-8  /,Ie( 464 )/-21 /,Ie( 465 )/ 3  /
Data Ie( 466 )/ 20 /,Ie( 467 )/ 0  /,Ie( 468 )/-16 /
Data Ie( 469 )/ 2  /,Ie( 470 )/ 20 /,Ie( 471 )/ 3  /
Data Ie( 472 )/ 16 /,Ie( 473 )/ 11 /,Ie( 474 )/ 11 /
Data Ie( 475 )/-7  /,Ie( 476 )/-10 /,Ie( 477 )/ 7  /
Data Ie( 478 )/-11 /,Ie( 479 )/-6  /,Ie( 480 )/ 11 /
Data Ie( 481 )/-7  /,Ie( 482 )/ 13 /,Ie( 483 )/ 8  /
Data Ie( 484 )/-13 /,Ie( 485 )/-19 /,Ie( 486 )/-21 /
Data Ie( 487 )/-18 /,Ie( 488 )/ 21 /,Ie( 489 )/ 13 /
Data Ie( 490 )/ 1  /,Ie( 491 )/-13 /,Ie( 492 )/-19 /
Data Ie( 493 )/ 5  /,Ie( 494 )/ 21 /,Ie( 495 )/ 15 /
Data Ie( 496 )/ 0  /,Ie( 497 )/-13 /,Ie( 498 )/ 21 /
Data Ie( 499 )/ 0  /,Ie( 500 )/ 20 /,Ie( 501 )/ 7  /
Data Ie( 502 )/-21 /,Ie( 503 )/-6  /,Ie( 504 )/ 18 /
Data Ie( 505 )/-5  /,Ie( 506 )/ 0  /,Ie( 507 )/ 18 /
Data Ie( 508 )/-1  /,Ie( 509 )/-2  /,Ie( 510 )/ 21 /
Data Ie( 511 )/-22 /,Ie( 512 )/-21 /,Ie( 513 )/-21 /
Data Ie( 514 )/ 21 /,Ie( 515 )/ 17 /,Ie( 516 )/ 1  /
Data Ie( 517 )/-16 /,Ie( 518 )/-19 /,Ie( 519 )/ 12 /
Data Ie( 520 )/ 1  /,Ie( 521 )/ 0  /,Ie( 522 )/ 0  /
Data Ie( 523 )/ 14 /,Ie( 524 )/-21 /,Ie( 525 )/-11 /
Data Ie( 526 )/ 0  /,Ie( 527 )/ 12 /,Ie( 528 )/-18 /
Data Ie( 529 )/-12 /,Ie( 530 )/ 18 /,Ie( 531 )/ 14 /
Data Ie( 532 )/ 0  /,Ie( 533 )/-20 /,Ie( 534 )/-21 /
Data Ie( 535 )/ 16 /,Ie( 536 )/ 1  /,Ie( 537 )/-4  /
Data Ie( 538 )/-18 /,Ie( 539 )/ 10 /,Ie( 540 )/ 9  /
Data Ie( 541 )/-12 /,Ie( 542 )/-10 /,Ie( 543 )/-5  /
Data Ie( 544 )/-18 /,Ie( 545 )/-4  /,Ie( 546 )/ 7  /
Data Ie( 547 )/ 2  /,Ie( 548 )/-5  /,Ie( 549 )/ 2  /
Data Ie( 550 )/-6  /,Ie( 551 )/-6  /,Ie( 552 )/ 9  /
Data Ie( 553 )/ 3  /,Ie( 554 )/-6  /,Ie( 555 )/-18 /
Data Ie( 556 )/-3  /,Ie( 557 )/-8  /,Ie( 558 )/ 0  /
Data Ie( 559 )/-10 /,Ie( 560 )/ 21 /,Ie( 561 )/ 5  /
Data Ie( 562 )/ 2  /,Ie( 563 )/-5  /,Ie( 564 )/-1  /
Data Ie( 565 )/ 11 /,Ie( 566 )/ 1  /,Ie( 567 )/-9  /
Data Ie( 568 )/-17 /,Ie( 569 )/ 0  /,Ie( 570 )/-15 /
Data Ie( 571 )/ 0  /,Ie( 572 )/ 18 /,Ie( 573 )/-19 /
Data Ie( 574 )/ 0  /,Ie( 575 )/ 14 /,Ie( 576 )/ 1  /
Data Ie( 577 )/ 7  /,Ie( 578 )/ 20 /,Ie( 579 )/-14 /
Data Ie( 580 )/ 0  /,Ie( 581 )/ 1  /,Ie( 582 )/ 21 /
Data Ie( 583 )/ 4  /,Ie( 584 )/-21 /,Ie( 585 )/-3  /
Data Ie( 586 )/ 16 /,Ie( 587 )/ 3  /,Ie( 588 )/-16 /
Data Ie( 589 )/ 4  /,Ie( 590 )/-16 /,Ie( 591 )/ 3  /
```

```
Data Ie( 592 )/ 15 /,Ie( 593 )/-21 /,Ie( 594 )/ 0  /
Data Ie( 595 )/ 12 /,Ie( 596 )/-21 /,Ie( 597 )/-3  /
Data Ie( 598 )/ 21 /,Ie( 599 )/-12 /,Ie( 600 )/-19 /
Data Ie( 601 )/-1  /,Ie( 602 )/ 21 /,Ie( 603 )/-18 /
Data Ie( 604 )/-21 /,Ie( 605 )/-6  /,Ie( 606 )/ 21 /
Data Ie( 607 )/ 0  /,Ie( 608 )/-9  /,Ie( 609 )/-6  /
Data Ie( 610 )/ 20 /,Ie( 611 )/ 0  /,Ie( 612 )/ 10 /
Data Ie( 613 )/ 6  /,Ie( 614 )/ 10 /,Ie( 615 )/-15 /
Data Ie( 616 )/ 0  /,Ie( 617 )/-14 /,Ie( 618 )/-21 /
Data Ie( 619 )/ 15 /,Ie( 620 )/ 1  /,Ie( 621 )/-8  /
Data Ie( 622 )/-19 /,Ie( 623 )/ 12 /,Ie( 624 )/ 21 /
Data Ie( 625 )/ 11 /,Ie( 626 )/ 0  /,Ie( 627 )/-12 /
Data Ie( 628 )/-21 /,Ie( 629 )/ 9  /,Ie( 630 )/-20 /
Data Ie( 631 )/ 11 /,Ie( 632 )/ 21 /,Ie( 633 )/ 8  /
Data Ie( 634 )/ 14 /,Ie( 635 )/ 5  /,Ie( 636 )/ 11 /
Data Ie( 637 )/-4  /,Ie( 638 )/ 12 /,Ie( 639 )/-4  /
Data Ie( 640 )/ 13 /,Ie( 641 )/-5  /,Ie( 642 )/ 9  /
Data Ie( 643 )/-8  /,Ie( 644 )/ 3  /,Ie( 645 )/-2  /
Data Ie( 646 )/ 19 /,Ie( 647 )/ 0  /,Ie( 648 )/-20 /
Data Ie( 649 )/ 9  /,Ie( 650 )/ 8  /,Ie( 651 )/ 14 /
Data Ie( 652 )/ 10 /,Ie( 653 )/-11 /,Ie( 654 )/ 8  /
Data Ie( 655 )/ 4  /,Ie( 656 )/ 11 /,Ie( 657 )/ 13 /
Data Ie( 658 )/ 21 /,Ie( 659 )/-4  /,Ie( 660 )/ 20 /
Data Ie( 661 )/-2  /,Ie( 662 )/ 11 /,Ie( 663 )/-9  /
Data Ie( 664 )/ 8  /,Ie( 665 )/ 2  /,Ie( 666 )/-17 /
Data Ie( 667 )/ 4  /,Ie( 668 )/ 8  /,Ie( 669 )/-9  /
Data Ie( 670 )/-1  /,Ie( 671 )/ 9  /,Ie( 672 )/ 5  /
Data Ie( 673 )/ 14 /,Ie( 674 )/ 19 /,Ie( 675 )/ 2  /
Data Ie( 676 )/ 19 /,Ie( 677 )/-5  /,Ie( 678 )/ 14 /
Data Ie( 679 )/-9  /,Ie( 680 )/-14 /,Ie( 681 )/ 15 /
Data Ie( 682 )/ 13 /,Ie( 683 )/-4  /,Ie( 684 )/ 8  /
Data Ie( 685 )/-2  /,Ie( 686 )/ 8  /,Ie( 687 )/ 4  /
Data Ie( 688 )/-2  /,Ie( 689 )/-5  /,Ie( 690 )/-8  /
Data Ie( 691 )/-11 /,Ie( 692 )/ 4  /,Ie( 693 )/-12 /
Data Ie( 694 )/ 4  /,Ie( 695 )/-6  /,Ie( 696 )/ 0  /
Data Ie( 697 )/-2  /,Ie( 698 )/ 12 /,Ie( 699 )/-3  /
Data Ie( 700 )/ 13 /,Ie( 701 )/ 7  /,Ie( 702 )/ 1  /
Data Ie( 703 )/-2  /,Ie( 704 )/-6  /,Ie( 705 )/ 1  /
Data Ie( 706 )/ 13 /,Ie( 707 )/ 0  /,Ie( 708 )/-12 /
Data Ie( 709 )/-7  /,Ie( 710 )/ 14 /,Ie( 711 )/ 0  /
Data Ie( 712 )/-11 /,Ie( 713 )/ 9  /,Ie( 714 )/ 21 /
Data Ie( 715 )/ 3  /,Ie( 716 )/ 19 /,Ie( 717 )/-3  /
Data Ie( 718 )/-6  /,Ie( 719 )/ 7  /,Ie( 720 )/-8  /
Data Ie( 721 )/-7  /,Ie( 722 )/ 7  /,Ie( 723 )/ 6  /
Data Ie( 724 )/-7  /,Ie( 725 )/-3  /,Ie( 726 )/ 14 /
Data Ie( 727 )/-18 /,Ie( 728 )/-14 /,Ie( 729 )/ 14 /
Data Ie( 730 )/-5  /,Ie( 731 )/-11 /,Ie( 732 )/-12 /
Data Ie( 733 )/ 7  /,Ie( 734 )/ 2  /,Ie( 735 )/-25 /
Data Ie( 736 )/ 0  /,Ie( 737 )/-29 /,Ie( 738 )/ 14 /
Data Ie( 739 )/ 3  /,Ie( 740 )/ 14 /,Ie( 741 )/-18 /
Data Ie( 742 )/ 14 /,Ie( 743 )/-4  /,Ie( 744 )/-11 /
```

```
      Data Ie( 745 )/-10 /,Ie( 746 )/-7  /,Ie( 747 )/ 13 /
      Data Ie( 748 )/ 13 /,Ie( 749 )/-2  /,Ie( 750 )/ 18 /
      Data Ie( 751 )/ 2  /,Ie( 752 )/-7  /,Ie( 753 )/ 5  /
      Data Ie( 754 )/ 14 /,Ie( 755 )/ 7  /,Ie( 756 )/ 12 /
      Data Ie( 757 )/ 14 /,Ie( 758 )/ 12 /,Ie( 759 )/-10 /
      Data Ie( 760 )/ 11 /,Ie( 761 )/ 16 /,Ie( 762 )/ 3  /
      Data Ie( 763 )/-5  /,Ie( 764 )/ 12 /,Ie( 765 )/-10 /
      Data Ie( 766 )/ 9  /,Ie( 767 )/ 7  /,Ie( 768 )/ 14 /
      Data Ie( 769 )/-15 /,Ie( 770 )/ 14 /,Ie( 771 )/ 13 /
      Data Ie( 772 )/-10 /,Ie( 773 )/ 6  /,Ie( 774 )/-14 /
      Data Ie( 775 )/-9  /,Ie( 776 )/ 0  /,Ie( 777 )/ 5  /
      Data Ie( 778 )/-12 /,Ie( 779 )/-9  /,Ie( 780 )/ 12 /
      Data Ie( 781 )/ 1  /,Ie( 782 )/ 14 /,Ie( 783 )/ 4  /
      Data Ie( 784 )/-14 /,Ie( 785 )/ 3  /,Ie( 786 )/ 11 /
      Data Ie( 787 )/ 3  /,Ie( 788 )/ 10 /,Ie( 789 )/-13 /
      Data Ie( 790 )/ 0  /,Ie( 791 )/ 3  /,Ie( 792 )/-11 /
      Data Ie( 793 )/-16 /,Ie( 794 )/ 11 /,Ie( 795 )/ 10 /
      Data Ie( 796 )/-14 /,Ie( 797 )/-9  /,Ie( 798 )/ 14 /
      Data Ie( 799 )/-10 /,Ie( 800 )/-12 /,Ie( 801 )/-1  /
      Data Ie( 802 )/ 14 /,Ie( 803 )/-16 /,Ie( 804 )/-14 /
      Data Ie( 805 )/-9  /,Ie( 806 )/-7  /,Ie( 807 )/ 3  /
      Data Ie( 808 )/ 6  /,Ie( 809 )/ 13 /,Ie( 810 )/ 14 /
      Data Ie( 811 )/-10 /,Ie( 812 )/-14 /,Ie( 813 )/ 10 /
      Data Ie( 814 )/ 14 /,Ie( 815 )/ 7  /,Ie( 816  )/-13 /
C
      Nx=Ie(Ipair*2-3)
      Ny=Ie(Ipair*2-2)
      Return
      End
```

```
      Subroutine Triplx(Char,Xpos,Ypos,Exit_Code)

C  This routine returns the next relative move/draw to generate the current
C  character in a roman triplex font.

C  Parameters:

C  CHAR - b [i] = The character to generate.
C  XPOS,YPOS - R [o] = A relative move/draw coordinate.
C  EXIT_CODE - i [o] = 2 for a move
C                      3 for a draw
C                     -1 for no more graphic elements in character.

C  This routine was originally part of the Hewlett-Packard Plot21 system.
C  Its logic, but not its data areas, have been substantially re-written.

      Implicit Integer (A-Z)

      Integer Ic(2113),In(208)

      Byte Char

      Real Xm, Ym , Xpos, Ypos

C  CHARGEN.INC contains a stack data structure which must be preserved across
C  calls to this routine.

      Include 'Chargen.Inc'
      Include 'Symvecmem.Inc'
      Include 'Options.Inc'

C     ********************************************************
C     *                                                      *
C     *                                                      *
C     *  THIS MODULE GENERATES ROMAN TRIPLEX CHARACTERS      *
C     *                                                      *
C     *                                                      *
C     ********************************************************
C
C
C
C     ********************************************************
C     *                 CHARACTER TABLE                      *
C     ********************************************************
C------<< 33 ! >>
      Data Ic( 1  )/ 2   /,Ic( 2  )/ 3   /
      Data Ic( 3  )/ 4   /,Ic( 4  )/ 5   /
      Data Ic( 5  )/ 6   /,Ic( 6  )/ 7   /
      Data Ic( 7  )/ 8   /,Ic( 8  )/ 410 /
      Data Ic( 9  )/ 7   /,Ic( 10 )/ 10  /
      Data Ic( 11 )/ 11  /,Ic( 12 )/ 12  /
```

```
      Data Ic( 13 )/ 1   /,Ic( 14 )/ 13  /
      Data Ic( 15 )/ 413 /
C-----<< 34 " >>
      Data Ic( 16 )/ 18  /,Ic( 17 )/ 414 /
      Data Ic( 18 )/ 1   /,Ic( 19 )/ 22  /
      Data Ic( 20 )/ 414 /
C-----<< 36 $ >>
      Data Ic( 21 )/ 23  /,Ic( 22 )/ 24  /
      Data Ic( 23 )/ 1   /,Ic( 24 )/ 25  /
      Data Ic( 25 )/ 24  /,Ic( 26 )/ 1   /
      Data Ic( 27 )/ 26  /,Ic( 28 )/ 415 /
      Data Ic( 29 )/ 416 /,Ic( 30 )/ 29  /
      Data Ic( 31 )/ 30  /,Ic( 32 )/ 418 /
      Data Ic( 33 )/ 30  /,Ic( 34 )/ 29  /
      Data Ic( 35 )/ 17  /,Ic( 36 )/ 28  /
      Data Ic( 37 )/ 419 /
C-----<< 38 & >>
      Data Ic( 38 )/ 41  /,Ic( 39 )/ 415 /
      Data Ic( 40 )/ 17  /,Ic( 41 )/ 9   /
      Data Ic( 42 )/ 14  /,Ic( 43 )/ 38  /
      Data Ic( 44 )/ 42  /,Ic( 45 )/ 43  /
      Data Ic( 46 )/ 32  /,Ic( 47 )/ 44  /
      Data Ic( 48 )/ 30  /,Ic( 49 )/ 45  /
      Data Ic( 50 )/ 420 /,Ic( 51 )/ 48  /
      Data Ic( 52 )/ 49  /,Ic( 53 )/ 47  /
      Data Ic( 54 )/ 4   /,Ic( 55 )/ 421 /
      Data Ic( 56 )/ 422 /,Ic( 57 )/ 33  /
      Data Ic( 58 )/ 50  /,Ic( 59 )/ 51  /
      Data Ic( 60 )/ 492 /,Ic( 61 )/ 27  /
      Data Ic( 62 )/ 47  /,Ic( 63 )/ 16  /
      Data Ic( 64 )/ 1   /,Ic( 65 )/ 54  /
      Data Ic( 66 )/ 420 /,Ic( 67 )/ 5   /
      Data Ic( 68 )/ 1   /,Ic( 69 )/ 55  /
      Data Ic( 70 )/ 56  /,Ic( 71 )/ 1   /
      Data Ic( 72 )/ 11  /,Ic( 73 )/ 57  /
      Data Ic( 74 )/ 1   /,Ic( 75 )/ 30  /
      Data Ic( 76 )/ 423 /,Ic( 77 )/ 50  /
      Data Ic( 78 )/ 51  /,Ic( 79 )/ 492 /
      Data Ic( 80 )/ 1   /,Ic( 81 )/ 59  /
      Data Ic( 82 )/ 45  /,Ic( 83 )/ 420 /
      Data Ic( 84 )/ 60  /,Ic( 85 )/ 1   /
      Data Ic( 86 )/ 61  /,Ic( 87 )/ 62  /
      Data Ic( 88 )/ 63  /,Ic( 89 )/ 492 /
      Data Ic( 90 )/ 521 /
C-----<< 39 ' >>
      Data Ic( 91 )/ 64  /,Ic( 92 )/ 414 /
C-----<< 40 ( >>
      Data Ic( 93 )/ 65  /,Ic( 94 )/ 32  /
      Data Ic( 95 )/ 43  /,Ic( 96 )/ 66  /
      Data Ic( 97 )/ 67  /,Ic( 98 )/ 68  /
      Data Ic( 99 )/ 69  /,Ic( 100 )/ 70  /
```

```
      Data Ic( 101 )/ 51  /,Ic( 102 )/ 424 /
      Data Ic( 103 )/ 72  /,Ic( 104 )/ 73  /
      Data Ic( 105 )/ 58  /,Ic( 106 )/ 20  /
      Data Ic( 107 )/ 62  /,Ic( 108 )/ 50  /
      Data Ic( 109 )/ 1   /,Ic( 110 )/ 74  /
      Data Ic( 111 )/ 38  /,Ic( 112 )/ 73  /
      Data Ic( 113 )/ 75  /,Ic( 114 )/ 20  /
      Data Ic( 115 )/ 76  /,Ic( 116 )/ 50  /
      Data Ic( 117 )/ 34  /
C-----<< 41) >>
      Data Ic( 118 )/ 77  /,Ic( 119 )/ 71  /
      Data Ic( 120 )/ 51  /,Ic( 121 )/ 70  /
      Data Ic( 122 )/ 69  /,Ic( 123 )/ 68  /
      Data Ic( 124 )/ 67  /,Ic( 125 )/ 66  /
      Data Ic( 126 )/ 43  /,Ic( 127 )/ 32  /
      Data Ic( 128 )/ 1   /,Ic( 129 )/ 78  /
      Data Ic( 130 )/ 50  /,Ic( 131 )/ 62  /
      Data Ic( 132 )/ 20  /,Ic( 133 )/ 58  /
      Data Ic( 134 )/ 425 /,Ic( 135 )/ 79  /
      Data Ic( 136 )/ 426 /,Ic( 137 )/ 76  /
      Data Ic( 138 )/ 20  /,Ic( 139 )/ 75  /
      Data Ic( 140 )/ 73  /,Ic( 141 )/ 38  /
C-----<< 42 * >>
      Data Ic( 142 )/ 80  /,Ic( 143 )/ 81  /
      Data Ic( 144 )/ 14  /,Ic( 145 )/ 82  /
      Data Ic( 146 )/ 14  /,Ic( 147 )/ 17  /
      Data Ic( 148 )/ 83  /,Ic( 149 )/ 17  /
      Data Ic( 150 )/ 1   /,Ic( 151 )/ 84  /
      Data Ic( 152 )/ 85  /,Ic( 153 )/ 7   /
      Data Ic( 154 )/ 22  /,Ic( 155 )/ 519 /
      Data Ic( 156 )/ 86  /,Ic( 157 )/ 16  /
      Data Ic( 158 )/ 1   /,Ic( 159 )/ 20  /
      Data Ic( 160 )/ 87  /,Ic( 161 )/ 9   /
      Data Ic( 162 )/ 88  /,Ic( 163 )/ 9   /
      Data Ic( 164 )/ 16  /,Ic( 165 )/ 89  /
      Data Ic( 166 )/ 16  /
C-----<< 43 + >>
      Data Ic( 167 )/ 90  /,Ic( 168 )/ 9   /
      Data Ic( 169 )/ 91  /,Ic( 170 )/ 7   /
      Data Ic( 171 )/ 92  /,Ic( 172 )/ 1   /
      Data Ic( 173 )/ 93  /,Ic( 174 )/ 427 /
C-----<< 44 , >>
      Data Ic( 175 )/ 96  /,Ic( 176 )/ 429 /
C-----<< 45 - >>
      Data Ic( 177 )/ 99  /,Ic( 178 )/ 427 /
C-----<< 46 . >>
      Data Ic( 179 )/ 100 /,Ic( 180 )/ 413 /
C-----<< 47/ >>
      Data Ic( 181 )/ 101 /,Ic( 182 )/ 9   /
      Data Ic( 183 )/ 102 /,Ic( 184 )/ 7   /
      Data Ic( 185 )/ 103 /
```

```
C------<< 48 0 >>
        Data Ic( 186 )/ 104 /,Ic( 187 )/ 27  /
        Data Ic( 188 )/ 105 /,Ic( 189 )/ 51  /
        Data Ic( 190 )/ 69  /,Ic( 191 )/ 33  /
        Data Ic( 192 )/ 67  /,Ic( 193 )/ 43  /
        Data Ic( 194 )/ 430 /,Ic( 195 )/ 29  /
        Data Ic( 196 )/ 107 /,Ic( 197 )/ 98  /
        Data Ic( 198 )/ 46  /,Ic( 199 )/ 108 /
        Data Ic( 200 )/ 109 /,Ic( 201 )/ 110 /
        Data Ic( 202 )/ 422 /,Ic( 203 )/ 67  /
        Data Ic( 204 )/ 111 /,Ic( 205 )/ 69  /
        Data Ic( 206 )/ 499 /,Ic( 207 )/ 1   /
        Data Ic( 208 )/ 112 /,Ic( 209 )/ 38  /
        Data Ic( 210 )/ 58  /,Ic( 211 )/ 111 /
        Data Ic( 212 )/ 62  /,Ic( 213 )/ 431 /
        Data Ic( 214 )/ 113 /,Ic( 215 )/ 47  /
        Data Ic( 216 )/ 56  /,Ic( 217 )/ 114 /
        Data Ic( 218 )/ 57  /,Ic( 219 )/ 432 /
        Data Ic( 220 )/ 115 /,Ic( 221 )/ 428 /
        Data Ic( 222 )/ 108 /,Ic( 223 )/ 114 /
        Data Ic( 224 )/ 98  /,Ic( 225 )/ 421 /
C------<< 49 1 >>
        Data Ic( 226 )/ 116 /,Ic( 227 )/ 433 /
        Data Ic( 228 )/ 118 /,Ic( 229 )/ 503 /
        Data Ic( 230 )/ 10  /,Ic( 231 )/ 120 /
        Data Ic( 232 )/ 121 /,Ic( 233 )/ 434 /
        Data Ic( 234 )/ 122 /,Ic( 235 )/ 123 /
        Data Ic( 236 )/ 1   /,Ic( 237 )/ 124 /
        Data Ic( 238 )/ 435 /
C------<< 50 2 >>
        Data Ic( 239 )/ 436 /,Ic( 240 )/ 438 /
        Data Ic( 241 )/ 128 /,Ic( 242 )/ 129 /
        Data Ic( 243 )/ 44  /,Ic( 244 )/ 32  /
        Data Ic( 245 )/ 73  /,Ic( 246 )/ 33  /
        Data Ic( 247 )/ 1   /,Ic( 248 )/ 130 /
        Data Ic( 249 )/ 439 /,Ic( 250 )/ 507 /
        Data Ic( 251 )/ 50  /,Ic( 252 )/ 11  /
        Data Ic( 253 )/ 38  /,Ic( 254 )/ 32  /
        Data Ic( 255 )/ 132 /,Ic( 256 )/ 1   /
        Data Ic( 257 )/ 133 /,Ic( 258 )/ 5   /
        Data Ic( 259 )/ 27  /,Ic( 260 )/ 134 /
        Data Ic( 261 )/ 127 /,Ic( 262 )/ 97  /
        Data Ic( 263 )/ 1   /,Ic( 264 )/ 14  /
        Data Ic( 265 )/ 14  /,Ic( 266 )/ 30  /
        Data Ic( 267 )/ 135 /,Ic( 268 )/ 136 /
        Data Ic( 269 )/ 127 /,Ic( 270 )/ 440 /
        Data Ic( 271 )/ 4   /
C------<< 51 3 >>
        Data Ic( 272 )/ 436 /,Ic( 273 )/ 441 /
        Data Ic( 274 )/ 31  /,Ic( 275 )/ 1   /
        Data Ic( 276 )/ 137 /,Ic( 277 )/ 441 /
```

```
      Data Ic( 278 )/ 1   /,Ic( 279 )/ 507 /
      Data Ic( 280 )/ 441 /,Ic( 281 )/ 44  /
      Data Ic( 282 )/ 37  /,Ic( 283 )/ 516 /
      Data Ic( 284 )/ 33  /,Ic( 285 )/ 73  /
      Data Ic( 286 )/ 434 /,Ic( 287 )/ 138 /
      Data Ic( 288 )/ 139 /,Ic( 289 )/ 37  /
      Data Ic( 290 )/ 71  /,Ic( 291 )/ 441 /
      Data Ic( 292 )/ 500 /,Ic( 293 )/ 30  /
      Data Ic( 294 )/ 29  /,Ic( 295 )/ 17  /
      Data Ic( 296 )/ 28  /,Ic( 297 )/ 522 /
      Data Ic( 298 )/ 140 /,Ic( 299 )/ 441 /
      Data Ic( 300 )/ 1   /,Ic( 301 )/ 141 /
      Data Ic( 302 )/ 15  /,Ic( 303 )/ 519 /
      Data Ic( 304 )/ 9   /
C-----<< 52 4 >>
      Data Ic( 305 )/ 142 /,Ic( 306 )/ 503 /
      Data Ic( 307 )/ 118 /,Ic( 308 )/ 503 /
      Data Ic( 309 )/ 10  /,Ic( 310 )/ 120 /
      Data Ic( 311 )/ 143 /,Ic( 312 )/ 144 /
      Data Ic( 313 )/ 1   /,Ic( 314 )/ 145 /
      Data Ic( 315 )/ 443 /
C-----<< 53 5 >>
      Data Ic( 316 )/ 147 /,Ic( 317 )/ 523 /
      Data Ic( 318 )/ 15  /,Ic( 319 )/ 502 /
      Data Ic( 320 )/ 105 /,Ic( 321 )/ 139 /
      Data Ic( 322 )/ 444 /,Ic( 323 )/ 148 /
      Data Ic( 324 )/ 4   /,Ic( 325 )/ 445 /
      Data Ic( 326 )/ 29  /,Ic( 327 )/ 151 /
      Data Ic( 328 )/ 417 /,Ic( 329 )/ 83  /
      Data Ic( 330 )/ 123 /,Ic( 331 )/ 44  /
      Data Ic( 332 )/ 152 /,Ic( 333 )/ 30  /
      Data Ic( 334 )/ 1   /,Ic( 335 )/ 153 /
      Data Ic( 336 )/ 447 /,Ic( 337 )/ 155 /
      Data Ic( 338 )/ 501 /,Ic( 339 )/ 156 /
      Data Ic( 340 )/ 113 /
C-----<< 54 6 >>
      Data Ic( 341 )/ 157 /,Ic( 342 )/ 501 /
      Data Ic( 343 )/ 5   /,Ic( 344 )/ 9   /
      Data Ic( 345 )/ 411 /,Ic( 346 )/ 16  /
      Data Ic( 347 )/ 421 /,Ic( 348 )/ 151 /
      Data Ic( 349 )/ 417 /,Ic( 350 )/ 38  /
      Data Ic( 351 )/ 58  /,Ic( 352 )/ 20  /
      Data Ic( 353 )/ 50  /,Ic( 354 )/ 448 /
      Data Ic( 355 )/ 27  /,Ic( 356 )/ 444 /
      Data Ic( 357 )/ 148 /,Ic( 358 )/ 16  /
      Data Ic( 359 )/ 445 /,Ic( 360 )/ 29  /
      Data Ic( 361 )/ 106 /,Ic( 362 )/ 508 /
      Data Ic( 363 )/ 504 /,Ic( 364 )/ 158 /
      Data Ic( 365 )/ 38  /,Ic( 366 )/ 58  /
      Data Ic( 367 )/ 20  /,Ic( 368 )/ 50  /
      Data Ic( 369 )/ 10  /,Ic( 370 )/ 1   /
```

```
      Data Ic( 371 )/ 159 /,Ic( 372 )/ 47  /
      Data Ic( 373 )/ 46  /,Ic( 374 )/ 432 /
      Data Ic( 375 )/ 160 /,Ic( 376 )/ 508 /
      Data Ic( 377 )/ 38  /,Ic( 378 )/ 58  /
      Data Ic( 379 )/ 161 /,Ic( 380 )/ 50  /
      Data Ic( 381 )/ 34  /,Ic( 382 )/ 449 /
      Data Ic( 383 )/ 27  /,Ic( 384 )/ 97  /
      Data Ic( 385 )/ 5   /,Ic( 386 )/ 148 /
      Data Ic( 387 )/ 46  /,Ic( 388 )/ 149 /
      Data Ic( 389 )/ 17  /,Ic( 390 )/ 45  /
C------<< 55 7 >>
      Data Ic( 391 )/ 162 /,Ic( 392 )/ 20  /
      Data Ic( 393 )/ 1   /,Ic( 394 )/ 163 /
      Data Ic( 395 )/ 164 /,Ic( 396 )/ 43  /
      Data Ic( 397 )/ 73  /,Ic( 398 )/ 68  /
      Data Ic( 399 )/ 27  /,Ic( 400 )/ 165 /
      Data Ic( 401 )/ 56  /,Ic( 402 )/ 47  /
      Data Ic( 403 )/ 166 /,Ic( 404 )/ 148 /
      Data Ic( 405 )/ 46  /,Ic( 406 )/ 38  /
      Data Ic( 407 )/ 14  /,Ic( 408 )/ 106 /
      Data Ic( 409 )/ 167 /,Ic( 410 )/ 106 /
      Data Ic( 411 )/ 32  /,Ic( 412 )/ 38  /
      Data Ic( 413 )/ 450 /,Ic( 414 )/ 27  /
      Data Ic( 415 )/ 134 /,Ic( 416 )/ 1   /
      Data Ic( 417 )/ 168 /,Ic( 418 )/ 38  /
      Data Ic( 419 )/ 73  /,Ic( 420 )/ 68  /
      Data Ic( 421 )/ 1   /,Ic( 422 )/ 169 /
      Data Ic( 423 )/ 5   /,Ic( 424 )/ 27  /
      Data Ic( 425 )/ 37  /
C------<< 56 8 >>
      Data Ic( 426 )/ 170 /,Ic( 427 )/ 506 /
      Data Ic( 428 )/ 441 /,Ic( 429 )/ 31  /
      Data Ic( 430 )/ 30  /,Ic( 431 )/ 29  /
      Data Ic( 432 )/ 420 /,Ic( 433 )/ 110 /
      Data Ic( 434 )/ 451 /,Ic( 435 )/ 33  /
      Data Ic( 436 )/ 502 /,Ic( 437 )/ 31  /
      Data Ic( 438 )/ 451 /,Ic( 439 )/ 68  /
      Data Ic( 440 )/ 502 /,Ic( 441 )/ 105 /
      Data Ic( 442 )/ 127 /,Ic( 443 )/ 110 /
      Data Ic( 444 )/ 440 /,Ic( 445 )/ 165 /
      Data Ic( 446 )/ 416 /,Ic( 447 )/ 29  /
      Data Ic( 448 )/ 440 /,Ic( 449 )/ 46  /
      Data Ic( 450 )/ 416 /,Ic( 451 )/ 1   /
      Data Ic( 452 )/ 171 /,Ic( 453 )/ 38  /
      Data Ic( 454 )/ 33  /,Ic( 455 )/ 431 /
      Data Ic( 456 )/ 113 /,Ic( 457 )/ 47  /
      Data Ic( 458 )/ 46  /,Ic( 459 )/ 432 /
      Data Ic( 460 )/ 172 /,Ic( 461 )/ 453 /
      Data Ic( 462 )/ 173 /,Ic( 463 )/ 44  /
      Data Ic( 464 )/ 524 /,Ic( 465 )/ 449 /
      Data Ic( 466 )/ 127 /,Ic( 467 )/ 97  /
```

```
      Data Ic( 468 )/ 148 /,Ic( 469 )/ 165 /
      Data Ic( 470 )/ 149 /,Ic( 471 )/ 45  /
C-----<< 57 9 >>
      Data Ic( 472 )/ 60  /,Ic( 473 )/ 501 /
      Data Ic( 474 )/ 174 /,Ic( 475 )/ 38  /
      Data Ic( 476 )/ 14  /,Ic( 477 )/ 44  /
      Data Ic( 478 )/ 106 /,Ic( 479 )/ 29  /
      Data Ic( 480 )/ 150 /,Ic( 481 )/ 149 /
      Data Ic( 482 )/ 16  /,Ic( 483 )/ 148 /
      Data Ic( 484 )/ 49  /,Ic( 485 )/ 110 /
      Data Ic( 486 )/ 27  /,Ic( 487 )/ 105 /
      Data Ic( 488 )/ 71  /,Ic( 489 )/ 50  /
      Data Ic( 490 )/ 20  /,Ic( 491 )/ 58  /
      Data Ic( 492 )/ 38  /,Ic( 493 )/ 32  /
      Data Ic( 494 )/ 31  /,Ic( 495 )/ 151 /
      Data Ic( 496 )/ 45  /,Ic( 497 )/ 28  /
      Data Ic( 498 )/ 522 /,Ic( 499 )/ 175 /
      Data Ic( 500 )/ 420 /,Ic( 501 )/ 1   /
      Data Ic( 502 )/ 159 /,Ic( 503 )/ 10  /
      Data Ic( 504 )/ 50  /,Ic( 505 )/ 20  /
      Data Ic( 506 )/ 58  /,Ic( 507 )/ 504 /
      Data Ic( 508 )/ 176 /,Ic( 509 )/ 45  /
      Data Ic( 510 )/ 17  /,Ic( 511 )/ 149 /
      Data Ic( 512 )/ 46  /,Ic( 513 )/ 148 /
      Data Ic( 514 )/ 450 /,Ic( 515 )/ 1   /
      Data Ic( 516 )/ 27  /,Ic( 517 )/ 37  /
      Data Ic( 518 )/ 426 /,Ic( 519 )/ 161 /
      Data Ic( 520 )/ 58  /,Ic( 521 )/ 38  /
      Data Ic( 522 )/ 14  /,Ic( 523 )/ 44  /
C-----<< 58 : >>
      Data Ic( 524 )/ 64  /,Ic( 525 )/ 412 /
      Data Ic( 526 )/ 410 /,Ic( 527 )/ 177 /
      Data Ic( 528 )/ 413 /
C-----<< 59 ; >>
      Data Ic( 529 )/ 64  /,Ic( 530 )/ 412 /
      Data Ic( 531 )/ 410 /,Ic( 532 )/ 122 /
      Data Ic( 533 )/ 429 /
C-----<< 61 = >>
      Data Ic( 534 )/ 178 /,Ic( 535 )/ 427 /
      Data Ic( 536 )/ 1   /,Ic( 537 )/ 179 /
      Data Ic( 538 )/ 427 /
C-----<< 63 ? >>
      Data Ic( 539 )/ 180 /,Ic( 540 )/ 454 /
      Data Ic( 541 )/ 4   /,Ic( 542 )/ 47  /
      Data Ic( 543 )/ 450 /,Ic( 544 )/ 506 /
      Data Ic( 545 )/ 438 /,Ic( 546 )/ 14  /
      Data Ic( 547 )/ 132 /,Ic( 548 )/ 1   /
      Data Ic( 549 )/ 181 /,Ic( 550 )/ 10  /
      Data Ic( 551 )/ 68  /,Ic( 552 )/ 455 /
      Data Ic( 553 )/ 507 /,Ic( 554 )/ 446 /
      Data Ic( 555 )/ 455 /,Ic( 556 )/ 31  /
```

```
      Data Ic( 557 )/ 33  /,Ic( 558 )/ 7   /
      Data Ic( 559 )/ 46  /,Ic( 560 )/ 410 /
      Data Ic( 561 )/ 161 /,Ic( 562 )/ 413 /
C-----<< 65 A >>
      Data Ic( 563 )/ 144 /,Ic( 564 )/ 182 /
      Data Ic( 565 )/ 183 /,Ic( 566 )/ 455 /
      Data Ic( 567 )/ 184 /,Ic( 568 )/ 456 /
      Data Ic( 569 )/ 186 /,Ic( 570 )/ 456 /
      Data Ic( 571 )/ 187 /,Ic( 572 )/ 188 /
      Data Ic( 573 )/ 1   /,Ic( 574 )/ 189 /
      Data Ic( 575 )/ 457 /,Ic( 576 )/ 191 /
      Data Ic( 577 )/ 509 /,Ic( 578 )/ 193 /
      Data Ic( 579 )/ 449 /,Ic( 580 )/ 90  /
      Data Ic( 581 )/ 434 /,Ic( 582 )/ 49  /
      Data Ic( 583 )/ 504 /,Ic( 584 )/ 49  /
      Data Ic( 585 )/ 71  /
C-----<< 66 B >>
      Data Ic( 586 )/ 460 /,Ic( 587 )/ 438 /
      Data Ic( 588 )/ 500 /,Ic( 589 )/ 428 /
      Data Ic( 590 )/ 165 /,Ic( 591 )/ 421 /
      Data Ic( 592 )/ 1   /,Ic( 593 )/ 35  /
      Data Ic( 594 )/ 439 /,Ic( 595 )/ 199 /
      Data Ic( 596 )/ 192 /,Ic( 597 )/ 105 /
      Data Ic( 598 )/ 437 /,Ic( 599 )/ 498 /
      Data Ic( 600 )/ 500 /,Ic( 601 )/ 200 /
      Data Ic( 602 )/ 1   /,Ic( 603 )/ 201 /
      Data Ic( 604 )/ 461 /,Ic( 605 )/ 203 /
      Data Ic( 606 )/ 464 /
C-----<< 67 C >>
      Data Ic( 607 )/ 465 /,Ic( 608 )/ 49  /
      Data Ic( 609 )/ 47  /,Ic( 610 )/ 1   /
      Data Ic( 611 )/ 206 /,Ic( 612 )/ 511 /
      Data Ic( 613 )/ 1   /,Ic( 614 )/ 207 /
      Data Ic( 615 )/ 512 /,Ic( 616 )/ 37  /
C-----<< 68 D >>
      Data Ic( 617 )/ 459 /,Ic( 618 )/ 123 /
      Data Ic( 619 )/ 105 /,Ic( 620 )/ 71  /
      Data Ic( 621 )/ 426 /,Ic( 622 )/ 111 /
      Data Ic( 623 )/ 73  /,Ic( 624 )/ 38  /
      Data Ic( 625 )/ 32  /,Ic( 626 )/ 31  /
      Data Ic( 627 )/ 208 /,Ic( 628 )/ 1   /
      Data Ic( 629 )/ 209 /,Ic( 630 )/ 426 /
      Data Ic( 631 )/ 111 /,Ic( 632 )/ 73  /
      Data Ic( 633 )/ 38  /,Ic( 634 )/ 1   /
      Data Ic( 635 )/ 210 /,Ic( 636 )/ 37  /
      Data Ic( 637 )/ 51  /,Ic( 638 )/ 62  /
      Data Ic( 639 )/ 111 /,Ic( 640 )/ 58  /
      Data Ic( 641 )/ 43  /,Ic( 642 )/ 434 /
      Data Ic( 643 )/ 211 /,Ic( 644 )/ 464 /
C-----<< 69 E >>
      Data Ic( 645 )/ 466 /,Ic( 646 )/ 144 /
```

```
      Data Ic( 647 )/ 467 /,Ic( 648 )/ 217 /
      Data Ic( 649 )/ 468 /,Ic( 650 )/ 1   /
      Data Ic( 651 )/ 127 /,Ic( 652 )/ 469 /
C-----<< 70 F >>
      Data Ic( 653 )/ 466 /,Ic( 654 )/ 470 /
      Data Ic( 655 )/ 468 /
C-----<< 71 G >>
      Data Ic( 656 )/ 465 /,Ic( 657 )/ 27  /
      Data Ic( 658 )/ 10  /,Ic( 659 )/ 214 /
      Data Ic( 660 )/ 1   /,Ic( 661 )/ 222 /
      Data Ic( 662 )/ 511 /,Ic( 663 )/ 1   /
      Data Ic( 664 )/ 207 /,Ic( 665 )/ 512 /
      Data Ic( 666 )/ 449 /,Ic( 667 )/ 223 /
      Data Ic( 668 )/ 111 /,Ic( 669 )/ 1   /
      Data Ic( 670 )/ 212 /,Ic( 671 )/ 20  /
      Data Ic( 672 )/ 455 /,Ic( 673 )/ 224 /
      Data Ic( 674 )/ 471 /
C-----<< 72 H >>
      Data Ic( 675 )/ 472 /,Ic( 676 )/ 226 /
      Data Ic( 677 )/ 473 /,Ic( 678 )/ 518 /
      Data Ic( 679 )/ 227 /,Ic( 680 )/ 442 /
      Data Ic( 681 )/ 127 /,Ic( 682 )/ 442 /
      Data Ic( 683 )/ 228 /,Ic( 684 )/ 123 /
      Data Ic( 685 )/ 1   /,Ic( 686 )/ 229 /
      Data Ic( 687 )/ 442 /,Ic( 688 )/ 127 /
      Data Ic( 689 )/ 442 /,Ic( 690 )/ 230 /
      Data Ic( 691 )/ 463 /,Ic( 692 )/ 231 /
      Data Ic( 693 )/ 463 /,Ic( 694 )/ 232 /
      Data Ic( 695 )/ 474 /,Ic( 696 )/ 231 /
      Data Ic( 697 )/ 435 /
C-----<< 73 I >>
      Data Ic( 698 )/ 104 /,Ic( 699 )/ 475 /
      Data Ic( 700 )/ 470 /,Ic( 701 )/ 464 /
C-----<< 74 J >>
      Data Ic( 702 )/ 226 /,Ic( 703 )/ 92  /
      Data Ic( 704 )/ 73  /,Ic( 705 )/ 455 /
      Data Ic( 706 )/ 234 /,Ic( 707 )/ 235 /
      Data Ic( 708 )/ 425 /,Ic( 709 )/ 236 /
      Data Ic( 710 )/ 92  /,Ic( 711 )/ 73  /
      Data Ic( 712 )/ 430 /,Ic( 713 )/ 45  /
      Data Ic( 714 )/ 28  /,Ic( 715 )/ 4   /
      Data Ic( 716 )/ 5   /,Ic( 717 )/ 493 /
      Data Ic( 718 )/ 410 /,Ic( 719 )/ 4   /
      Data Ic( 720 )/ 501 /,Ic( 721 )/ 237 /
      Data Ic( 722 )/ 471 /
C-----<< 75 K >>
      Data Ic( 723 )/ 472 /,Ic( 724 )/ 175 /
      Data Ic( 725 )/ 238 /,Ic( 726 )/ 17  /
      Data Ic( 727 )/ 1   /,Ic( 728 )/ 239 /
      Data Ic( 729 )/ 240 /,Ic( 730 )/ 1   /
      Data Ic( 731 )/ 241 /,Ic( 732 )/ 240 /
```

```
      Data Ic( 733 )/ 1   /,Ic( 734 )/ 242 /
      Data Ic( 735 )/ 243 /,Ic( 736 )/ 1   /
      Data Ic( 737 )/ 227 /,Ic( 738 )/ 442 /
      Data Ic( 739 )/ 191 /,Ic( 740 )/ 457 /
      Data Ic( 741 )/ 244 /,Ic( 742 )/ 442 /
      Data Ic( 743 )/ 127 /,Ic( 744 )/ 509 /
      Data Ic( 745 )/ 245 /,Ic( 746 )/ 463 /
      Data Ic( 747 )/ 246 /,Ic( 748 )/ 434 /
      Data Ic( 749 )/ 247 /,Ic( 750 )/ 474 /
      Data Ic( 751 )/ 190 /,Ic( 752 )/ 49  /
      Data Ic( 753 )/ 35  /
C------<< 76 L >>
      Data Ic( 754 )/ 248 /,Ic( 755 )/ 475 /
      Data Ic( 756 )/ 249 /,Ic( 757 )/ 467 /
      Data Ic( 758 )/ 250 /,Ic( 759 )/ 464 /
      Data Ic( 760 )/ 1   /,Ic( 761 )/ 139 /
      Data Ic( 762 )/ 469 /
C------<< 77 M >>
      Data Ic( 763 )/ 9   /,Ic( 764 )/ 97  /
      Data Ic( 765 )/ 251 /,Ic( 766 )/ 252 /
      Data Ic( 767 )/ 170 /,Ic( 768 )/ 510 /
      Data Ic( 769 )/ 250 /,Ic( 770 )/ 456 /
      Data Ic( 771 )/ 7   /,Ic( 772 )/ 253 /
      Data Ic( 773 )/ 254 /,Ic( 774 )/ 1   /
      Data Ic( 775 )/ 255 /,Ic( 776 )/ 433 /
      Data Ic( 777 )/ 518 /,Ic( 778 )/ 256 /
      Data Ic( 779 )/ 191 /,Ic( 780 )/ 1   /
      Data Ic( 781 )/ 257 /,Ic( 782 )/ 457 /
      Data Ic( 783 )/ 113 /,Ic( 784 )/ 442 /
      Data Ic( 785 )/ 258 /,Ic( 786 )/ 449 /
      Data Ic( 787 )/ 259 /,Ic( 788 )/ 520 /
      Data Ic( 789 )/ 260 /,Ic( 790 )/ 449 /
      Data Ic( 791 )/ 261 /,Ic( 792 )/ 435 /
C------<< 78 N >>
      Data Ic( 793 )/ 262 /,Ic( 794 )/ 97  /
      Data Ic( 795 )/ 251 /,Ic( 796 )/ 263 /
      Data Ic( 797 )/ 251 /,Ic( 798 )/ 476 /
      Data Ic( 799 )/ 264 /,Ic( 800 )/ 265 /
      Data Ic( 801 )/ 477 /,Ic( 802 )/ 266 /
      Data Ic( 803 )/ 254 /,Ic( 804 )/ 1   /
      Data Ic( 805 )/ 267 /,Ic( 806 )/ 457 /
      Data Ic( 807 )/ 268 /,Ic( 808 )/ 457 /
      Data Ic( 809 )/ 197 /,Ic( 810 )/ 449 /
      Data Ic( 811 )/ 269 /,Ic( 812 )/ 434 /
      Data Ic( 813 )/ 232 /,Ic( 814 )/ 37  /
C------<< 79 O >>
      Data Ic( 815 )/ 478 /,Ic( 816 )/ 45  /
C------<< 80 P >>
      Data Ic( 817 )/ 460 /,Ic( 818 )/ 437 /
      Data Ic( 819 )/ 498 /,Ic( 820 )/ 500 /
      Data Ic( 821 )/ 225 /,Ic( 822 )/ 1   /
```

```
      Data Ic( 823 )/ 271 /,Ic( 824 )/ 461 /
      Data Ic( 825 )/ 272 /,Ic( 826 )/ 470 /
      Data Ic( 827 )/ 464 /
C-----<< 81 Q >>
      Data Ic( 828 )/ 478 /,Ic( 829 )/ 476 /
      Data Ic( 830 )/ 273 /,Ic( 831 )/ 47  /
      Data Ic( 832 )/ 97  /,Ic( 833 )/ 7   /
      Data Ic( 834 )/ 517 /,Ic( 835 )/ 76  /
      Data Ic( 836 )/ 479 /,Ic( 837 )/ 274 /
      Data Ic( 838 )/ 275 /,Ic( 839 )/ 10  /
      Data Ic( 840 )/ 521 /
C-----<< 82 R >>
      Data Ic( 841 )/ 460 /,Ic( 842 )/ 438 /
      Data Ic( 843 )/ 500 /,Ic( 844 )/ 225 /
      Data Ic( 845 )/ 1   /,Ic( 846 )/ 174 /
      Data Ic( 847 )/ 439 /,Ic( 848 )/ 507 /
      Data Ic( 849 )/ 446 /,Ic( 850 )/ 434 /
      Data Ic( 851 )/ 151 /,Ic( 852 )/ 517 /
      Data Ic( 853 )/ 276 /,Ic( 854 )/ 479 /
      Data Ic( 855 )/ 277 /,Ic( 856 )/ 10  /
      Data Ic( 857 )/ 278 /,Ic( 858 )/ 10  /
      Data Ic( 859 )/ 7   /,Ic( 860 )/ 5   /
      Data Ic( 861 )/ 1   /,Ic( 862 )/ 279 /
      Data Ic( 863 )/ 470 /,Ic( 864 )/ 464 /
C-----<< 83 S >>
      Data Ic( 865 )/ 205 /,Ic( 866 )/ 148 /
      Data Ic( 867 )/ 20  /,Ic( 868 )/ 445 /
      Data Ic( 869 )/ 29  /,Ic( 870 )/ 151 /
      Data Ic( 871 )/ 418 /,Ic( 872 )/ 151 /
      Data Ic( 873 )/ 29  /,Ic( 874 )/ 150 /
      Data Ic( 875 )/ 149 /,Ic( 876 )/ 20  /
      Data Ic( 877 )/ 148 /
C-----<< 84 T >>
      Data Ic( 878 )/ 162 /,Ic( 879 )/ 75  /
      Data Ic( 880 )/ 216 /,Ic( 881 )/ 494 /
      Data Ic( 882 )/ 280 /,Ic( 883 )/ 473 /
      Data Ic( 884 )/ 518 /,Ic( 885 )/ 254 /
      Data Ic( 886 )/ 442 /,Ic( 887 )/ 281 /
      Data Ic( 888 )/ 480 /,Ic( 889 )/ 284 /
      Data Ic( 890 )/ 515 /,Ic( 891 )/ 285 /
      Data Ic( 892 )/ 435 /
C-----<< 85 U >>
      Data Ic( 893 )/ 162 /,Ic( 894 )/ 286 /
      Data Ic( 895 )/ 50  /,Ic( 896 )/ 448 /
      Data Ic( 897 )/ 27  /,Ic( 898 )/ 444 /
      Data Ic( 899 )/ 148 /,Ic( 900 )/ 8   /
      Data Ic( 901 )/ 45  /,Ic( 902 )/ 1   /
      Data Ic( 903 )/ 59  /,Ic( 904 )/ 286 /
      Data Ic( 905 )/ 431 /,Ic( 906 )/ 287 /
      Data Ic( 907 )/ 235 /,Ic( 908 )/ 50  /
      Data Ic( 909 )/ 10  /,Ic( 910 )/ 449 /
```

```
      Data Ic( 911 )/ 211 /,Ic( 912 )/ 442 /
      Data Ic( 913 )/ 190 /,Ic( 914 )/ 457 /
      Data Ic( 915 )/ 288 /,Ic( 916 )/ 463 /
      Data Ic( 917 )/ 289 /,Ic( 918 )/ 44  /
C-----<< 86 V >>
      Data Ic( 919 )/ 162 /,Ic( 920 )/ 252 /
      Data Ic( 921 )/ 46  /,Ic( 922 )/ 253 /
      Data Ic( 923 )/ 477 /,Ic( 924 )/ 456 /
      Data Ic( 925 )/ 73  /,Ic( 926 )/ 290 /
      Data Ic( 927 )/ 476 /,Ic( 928 )/ 291 /
      Data Ic( 929 )/ 509 /,Ic( 930 )/ 191 /
      Data Ic( 931 )/ 457 /,Ic( 932 )/ 94  /
      Data Ic( 933 )/ 424 /,Ic( 934 )/ 49  /
      Data Ic( 935 )/ 520 /,Ic( 936 )/ 246 /
      Data Ic( 937 )/ 14  /
C-----<< 87 W >>
      Data Ic( 938 )/ 292 /,Ic( 939 )/ 293 /
      Data Ic( 940 )/ 114 /,Ic( 941 )/ 294 /
      Data Ic( 942 )/ 477 /,Ic( 943 )/ 295 /
      Data Ic( 944 )/ 1   /,Ic( 945 )/ 67  /
      Data Ic( 946 )/ 108 /,Ic( 947 )/ 237 /
      Data Ic( 948 )/ 293 /,Ic( 949 )/ 114 /
      Data Ic( 950 )/ 294 /,Ic( 951 )/ 1   /
      Data Ic( 952 )/ 296 /,Ic( 953 )/ 294 /
      Data Ic( 954 )/ 106 /,Ic( 955 )/ 1   /
      Data Ic( 956 )/ 293 /,Ic( 957 )/ 108 /
      Data Ic( 958 )/ 297 /,Ic( 959 )/ 476 /
      Data Ic( 960 )/ 94  /,Ic( 961 )/ 481 /
      Data Ic( 962 )/ 45  /,Ic( 963 )/ 424 /
      Data Ic( 964 )/ 49  /,Ic( 965 )/ 520 /
      Data Ic( 966 )/ 269 /,Ic( 967 )/ 44  /
C-----<< 88 X >>
      Data Ic( 968 )/ 162 /,Ic( 969 )/ 299 /
      Data Ic( 970 )/ 1   /,Ic( 971 )/ 203 /
      Data Ic( 972 )/ 299 /,Ic( 973 )/ 1   /
      Data Ic( 974 )/ 203 /,Ic( 975 )/ 299 /
      Data Ic( 976 )/ 1   /,Ic( 977 )/ 300 /
      Data Ic( 978 )/ 37  /,Ic( 979 )/ 301 /
      Data Ic( 980 )/ 434 /,Ic( 981 )/ 302 /
      Data Ic( 982 )/ 509 /,Ic( 983 )/ 191 /
      Data Ic( 984 )/ 457 /,Ic( 985 )/ 303 /
      Data Ic( 986 )/ 457 /,Ic( 987 )/ 191 /
      Data Ic( 988 )/ 509 /,Ic( 989 )/ 227 /
      Data Ic( 990 )/ 35  /,Ic( 991 )/ 47  /
      Data Ic( 992 )/ 477 /,Ic( 993 )/ 434 /
      Data Ic( 994 )/ 246 /,Ic( 995 )/ 434 /
      Data Ic( 996 )/ 301 /,Ic( 997 )/ 449 /
      Data Ic( 998 )/ 159 /,Ic( 999 )/ 434 /
      Data Ic( 1000 )/ 7   /,Ic( 1001 )/ 47  /
      Data Ic( 1002 )/ 35  /
C-----<< 89 Y >>
```

```
      Data Ic( 1003 )/ 292 /,Ic( 1004 )/ 240 /
      Data Ic( 1005 )/ 177 /,Ic( 1006 )/ 1   /
      Data Ic( 1007 )/ 304 /,Ic( 1008 )/ 240 /
      Data Ic( 1009 )/ 305 /,Ic( 1010 )/ 1   /
      Data Ic( 1011 )/ 306 /,Ic( 1012 )/ 240 /
      Data Ic( 1013 )/ 177 /,Ic( 1014 )/ 1   /
      Data Ic( 1015 )/ 307 /,Ic( 1016 )/ 308 /
      Data Ic( 1017 )/ 476 /,Ic( 1018 )/ 309 /
      Data Ic( 1019 )/ 509 /,Ic( 1020 )/ 192 /
      Data Ic( 1021 )/ 457 /,Ic( 1022 )/ 310 /
      Data Ic( 1023 )/ 442 /,Ic( 1024 )/ 250 /
      Data Ic( 1025 )/ 449 /,Ic( 1026 )/ 110 /
      Data Ic( 1027 )/ 434 /,Ic( 1028 )/ 311 /
      Data Ic( 1029 )/ 434 /,Ic( 1030 )/ 312 /
      Data Ic( 1031 )/ 435 /
C-----<< 90 Z >>
      Data Ic( 1032 )/ 127 /,Ic( 1033 )/ 313 /
      Data Ic( 1034 )/ 291 /,Ic( 1035 )/ 20  /
      Data Ic( 1036 )/ 21  /,Ic( 1037 )/ 1   /
      Data Ic( 1038 )/ 314 /,Ic( 1039 )/ 315 /
      Data Ic( 1040 )/ 267 /,Ic( 1041 )/ 467 /
      Data Ic( 1042 )/ 120 /,Ic( 1043 )/ 315 /
      Data Ic( 1044 )/ 1   /,Ic( 1045 )/ 292 /
      Data Ic( 1046 )/ 480 /,Ic( 1047 )/ 316 /
      Data Ic( 1048 )/ 469 /
C-----<< 91 [ >>
      Data Ic( 1049 )/ 317 /,Ic( 1050 )/ 422 /
      Data Ic( 1051 )/ 33  /,Ic( 1052 )/ 10  /
      Data Ic( 1053 )/ 523 /,Ic( 1054 )/ 56  /
      Data Ic( 1055 )/ 1   /,Ic( 1056 )/ 68  /
      Data Ic( 1057 )/ 501 /,Ic( 1058 )/ 28  /
      Data Ic( 1059 )/ 10  /
C-----<< 92 \ >>
      Data Ic( 1060 )/ 318 /,Ic( 1061 )/ 429 /
C-----<< 97 A >>
      Data Ic( 1062 )/ 319 /,Ic( 1063 )/ 454 /
      Data Ic( 1064 )/ 4   /,Ic( 1065 )/ 5   /
      Data Ic( 1066 )/ 97  /,Ic( 1067 )/ 127 /
      Data Ic( 1068 )/ 37  /,Ic( 1069 )/ 437 /
      Data Ic( 1070 )/ 161 /,Ic( 1071 )/ 502 /
      Data Ic( 1072 )/ 1   /,Ic( 1073 )/ 320 /
      Data Ic( 1074 )/ 34  /,Ic( 1075 )/ 161 /
      Data Ic( 1076 )/ 431 /,Ic( 1077 )/ 321 /
      Data Ic( 1078 )/ 437 /,Ic( 1079 )/ 179 /
      Data Ic( 1080 )/ 34  /,Ic( 1081 )/ 105 /
      Data Ic( 1082 )/ 7   /,Ic( 1083 )/ 1   /
      Data Ic( 1084 )/ 322 /,Ic( 1085 )/ 14  /
      Data Ic( 1086 )/ 283 /,Ic( 1087 )/ 31  /
      Data Ic( 1088 )/ 38  /,Ic( 1089 )/ 15  /
      Data Ic( 1090 )/ 34  /,Ic( 1091 )/ 105 /
      Data Ic( 1092 )/ 139 /,Ic( 1093 )/ 428 /
```

```
      Data Ic( 1094 )/ 1   /,Ic( 1095 )/ 323 /
      Data Ic( 1096 )/ 38  /,Ic( 1097 )/ 15  /
      Data Ic( 1098 )/ 431 /,Ic( 1099 )/ 223 /
      Data Ic( 1100 )/ 152 /,Ic( 1101 )/ 422 /
      Data Ic( 1102 )/ 15  /,Ic( 1103 )/ 502 /
C-----<< 98 B >>
      Data Ic( 1104 )/ 194 /,Ic( 1105 )/ 195 /
      Data Ic( 1106 )/ 5   /,Ic( 1107 )/ 27  /
      Data Ic( 1108 )/ 1   /,Ic( 1109 )/ 324 /
      Data Ic( 1110 )/ 503 /,Ic( 1111 )/ 169 /
      Data Ic( 1112 )/ 191 /,Ic( 1113 )/ 325 /
      Data Ic( 1114 )/ 1   /,Ic( 1115 )/ 307 /
      Data Ic( 1116 )/ 482 /,Ic( 1117 )/ 211 /
      Data Ic( 1118 )/ 449 /,Ic( 1119 )/ 17  /
      Data Ic( 1120 )/ 34  /
C-----<< 99 C >>
      Data Ic( 1121 )/ 327 /,Ic( 1122 )/ 415 /
      Data Ic( 1123 )/ 150 /,Ic( 1124 )/ 45  /
      Data Ic( 1125 )/ 151 /,Ic( 1126 )/ 417 /
      Data Ic( 1127 )/ 483 /,Ic( 1128 )/ 444 /
      Data Ic( 1129 )/ 1   /,Ic( 1130 )/ 328 /
      Data Ic( 1131 )/ 525 /,Ic( 1132 )/ 37  /
C-----<<100 D >>
      Data Ic( 1133 )/ 330 /,Ic( 1134 )/ 195 /
      Data Ic( 1135 )/ 191 /,Ic( 1136 )/ 1   /
      Data Ic( 1137 )/ 331 /,Ic( 1138 )/ 433 /
      Data Ic( 1139 )/ 484 /,Ic( 1140 )/ 332 /
      Data Ic( 1141 )/ 485 /,Ic( 1142 )/ 194 /
      Data Ic( 1143 )/ 462 /,Ic( 1144 )/ 334 /
      Data Ic( 1145 )/ 505 /
C-----<<101 E >>
      Data Ic( 1146 )/ 335 /,Ic( 1147 )/ 314 /
      Data Ic( 1148 )/ 4   /,Ic( 1149 )/ 416 /
      Data Ic( 1150 )/ 29  /,Ic( 1151 )/ 106 /
      Data Ic( 1152 )/ 417 /,Ic( 1153 )/ 483 /
      Data Ic( 1154 )/ 444 /,Ic( 1155 )/ 1   /
      Data Ic( 1156 )/ 212 /,Ic( 1157 )/ 16  /
      Data Ic( 1158 )/ 432 /,Ic( 1159 )/ 336 /
      Data Ic( 1160 )/ 452 /,Ic( 1161 )/ 337 /
      Data Ic( 1162 )/ 46  /,Ic( 1163 )/ 421 /
      Data Ic( 1164 )/ 1   /,Ic( 1165 )/ 106 /
      Data Ic( 1166 )/ 508 /,Ic( 1167 )/ 524 /
      Data Ic( 1168 )/ 10  /,Ic( 1169 )/ 37  /
C-----<<102 F >>
      Data Ic( 1170 )/ 338 /,Ic( 1171 )/ 415 /
      Data Ic( 1172 )/ 17  /,Ic( 1173 )/ 151 /
      Data Ic( 1174 )/ 508 /,Ic( 1175 )/ 73  /
      Data Ic( 1176 )/ 235 /,Ic( 1177 )/ 1   /
      Data Ic( 1178 )/ 339 /,Ic( 1179 )/ 73  /
      Data Ic( 1180 )/ 286 /,Ic( 1181 )/ 1   /
      Data Ic( 1182 )/ 234 /,Ic( 1183 )/ 451 /
```

```
      Data Ic( 1184 )/ 503 /,Ic( 1185 )/ 340 /
      Data Ic( 1186 )/ 188 /,Ic( 1187 )/ 1   /
      Data Ic( 1188 )/ 341 /,Ic( 1189 )/ 443 /
C-----<<103 G >>
      Data Ic( 1190 )/ 342 /,Ic( 1191 )/ 10  /
      Data Ic( 1192 )/ 5   /,Ic( 1193 )/ 17  /
      Data Ic( 1194 )/ 9   /,Ic( 1195 )/ 508 /
      Data Ic( 1196 )/ 439 /,Ic( 1197 )/ 343 /
      Data Ic( 1198 )/ 27  /,Ic( 1199 )/ 37  /
      Data Ic( 1200 )/ 438 /,Ic( 1201 )/ 14  /
      Data Ic( 1202 )/ 44  /,Ic( 1203 )/ 106 /
      Data Ic( 1204 )/ 45  /,Ic( 1205 )/ 17  /
      Data Ic( 1206 )/ 28  /,Ic( 1207 )/ 4   /
      Data Ic( 1208 )/ 47  /,Ic( 1209 )/ 450 /
      Data Ic( 1210 )/ 451 /,Ic( 1211 )/ 68  /
      Data Ic( 1212 )/ 502 /,Ic( 1213 )/ 1   /
      Data Ic( 1214 )/ 344 /,Ic( 1215 )/ 38  /
      Data Ic( 1216 )/ 11  /,Ic( 1217 )/ 431 /
      Data Ic( 1218 )/ 345 /,Ic( 1219 )/ 440 /
      Data Ic( 1220 )/ 165 /,Ic( 1221 )/ 416 /
      Data Ic( 1222 )/ 1   /,Ic( 1223 )/ 346 /
      Data Ic( 1224 )/ 451 /,Ic( 1225 )/ 15  /
      Data Ic( 1226 )/ 502 /,Ic( 1227 )/ 105 /
      Data Ic( 1228 )/ 506 /,Ic( 1229 )/ 10  /
      Data Ic( 1230 )/ 1   /,Ic( 1231 )/ 347 /
      Data Ic( 1232 )/ 105 /,Ic( 1233 )/ 506 /
      Data Ic( 1234 )/ 1   /,Ic( 1235 )/ 348 /
      Data Ic( 1236 )/ 10  /,Ic( 1237 )/ 105 /
      Data Ic( 1238 )/ 191 /,Ic( 1239 )/ 105 /
      Data Ic( 1240 )/ 34  /,Ic( 1241 )/ 15  /
      Data Ic( 1242 )/ 38  /,Ic( 1243 )/ 31  /
      Data Ic( 1244 )/ 349 /,Ic( 1245 )/ 29  /
      Data Ic( 1246 )/ 28  /,Ic( 1247 )/ 16  /
      Data Ic( 1248 )/ 47  /,Ic( 1249 )/ 110 /
      Data Ic( 1250 )/ 422 /,Ic( 1251 )/ 15  /
      Data Ic( 1252 )/ 499 /
C-----<<104 H >>
      Data Ic( 1253 )/ 458 /,Ic( 1254 )/ 484 /
      Data Ic( 1255 )/ 526 /,Ic( 1256 )/ 245 /
      Data Ic( 1257 )/ 462 /,Ic( 1258 )/ 119 /
      Data Ic( 1259 )/ 486 /,Ic( 1260 )/ 435 /
C-----<<105 I >>
      Data Ic( 1261 )/ 104 /,Ic( 1262 )/ 487 /
      Data Ic( 1263 )/ 488 /,Ic( 1264 )/ 254 /
      Data Ic( 1265 )/ 442 /,Ic( 1266 )/ 356 /
      Data Ic( 1267 )/ 514 /,Ic( 1268 )/ 435 /
C-----<<106 J >>
      Data Ic( 1269 )/ 358 /,Ic( 1270 )/ 487 /
      Data Ic( 1271 )/ 92  /,Ic( 1272 )/ 73  /
      Data Ic( 1273 )/ 455 /,Ic( 1274 )/ 234 /
      Data Ic( 1275 )/ 286 /,Ic( 1276 )/ 425 /
```

```
      Data Ic( 1277 )/ 112 /,Ic( 1278 )/ 191 /
      Data Ic( 1279 )/ 235 /,Ic( 1280 )/ 73  /
      Data Ic( 1281 )/ 14  /,Ic( 1282 )/ 44  /
      Data Ic( 1283 )/ 151 /,Ic( 1284 )/ 17  /
      Data Ic( 1285 )/ 419 /,Ic( 1286 )/ 1   /
      Data Ic( 1287 )/ 359 /,Ic( 1288 )/ 449 /
      Data Ic( 1289 )/ 17  /,Ic( 1290 )/ 34  /
C------<<107 K >>
      Data Ic( 1291 )/ 458 /,Ic( 1292 )/ 484 /
      Data Ic( 1293 )/ 165 /,Ic( 1294 )/ 80  /
      Data Ic( 1295 )/ 476 /,Ic( 1296 )/ 360 /
      Data Ic( 1297 )/ 361 /,Ic( 1298 )/ 1   /
      Data Ic( 1299 )/ 362 /,Ic( 1300 )/ 363 /
      Data Ic( 1301 )/ 1   /,Ic( 1302 )/ 362 /
      Data Ic( 1303 )/ 363 /,Ic( 1304 )/ 1   /
      Data Ic( 1305 )/ 364 /,Ic( 1306 )/ 509 /
      Data Ic( 1307 )/ 365 /,Ic( 1308 )/ 513 /
      Data Ic( 1309 )/ 509 /,Ic( 1310 )/ 227 /
      Data Ic( 1311 )/ 462 /,Ic( 1312 )/ 366 /
      Data Ic( 1313 )/ 31  /,Ic( 1314 )/ 1   /
      Data Ic( 1315 )/ 367 /,Ic( 1316 )/ 474 /
      Data Ic( 1317 )/ 368 /,Ic( 1318 )/ 32  /
      Data Ic( 1319 )/ 1   /,Ic( 1320 )/ 47  /
      Data Ic( 1321 )/ 345 /
C------<<108 L >>
      Data Ic( 1322 )/ 104 /,Ic( 1323 )/ 473 /
      Data Ic( 1324 )/ 484 /,Ic( 1325 )/ 254 /
      Data Ic( 1326 )/ 470 /,Ic( 1327 )/ 462 /
      Data Ic( 1328 )/ 119 /,Ic( 1329 )/ 435 /
C------<<109 M >>
      Data Ic( 1330 )/ 364 /,Ic( 1331 )/ 488 /
      Data Ic( 1332 )/ 307 /,Ic( 1333 )/ 495 /
      Data Ic( 1334 )/ 83  /,Ic( 1335 )/ 495 /
      Data Ic( 1336 )/ 369 /,Ic( 1337 )/ 513 /
      Data Ic( 1338 )/ 513 /,Ic( 1339 )/ 442 /
      Data Ic( 1340 )/ 370 /,Ic( 1341 )/ 514 /
      Data Ic( 1342 )/ 486 /,Ic( 1343 )/ 486 /
      Data Ic( 1344 )/ 435 /
C------<<110 N >>
      Data Ic( 1345 )/ 371 /,Ic( 1346 )/ 488 /
      Data Ic( 1347 )/ 526 /,Ic( 1348 )/ 372 /
      Data Ic( 1349 )/ 514 /,Ic( 1350 )/ 486 /
      Data Ic( 1351 )/ 435 /
C------<<111 O >>
      Data Ic( 1352 )/ 318 /,Ic( 1353 )/ 496 /
      Data Ic( 1354 )/ 430 /,Ic( 1355 )/ 29  /
      Data Ic( 1356 )/ 150 /,Ic( 1357 )/ 149 /
      Data Ic( 1358 )/ 4   /,Ic( 1359 )/ 148 /
      Data Ic( 1360 )/ 49  /,Ic( 1361 )/ 110 /
      Data Ic( 1362 )/ 508 /,Ic( 1363 )/ 524 /
      Data Ic( 1364 )/ 10  /,Ic( 1365 )/ 449 /
```

```
      Data Ic( 1366 )/ 154 /,Ic( 1367 )/ 453 /
      Data Ic( 1368 )/ 373 /,Ic( 1369 )/ 97  /
      Data Ic( 1370 )/ 5   /,Ic( 1371 )/ 148 /
      Data Ic( 1372 )/ 165 /,Ic( 1373 )/ 149 /
      Data Ic( 1374 )/ 17  /,Ic( 1375 )/ 45  /
C-----<<112 P >>
      Data Ic( 1376 )/ 371 /,Ic( 1377 )/ 473 /
      Data Ic( 1378 )/ 484 /,Ic( 1379 )/ 287 /
      Data Ic( 1380 )/ 482 /,Ic( 1381 )/ 374 /
      Data Ic( 1382 )/ 470 /,Ic( 1383 )/ 462 /
      Data Ic( 1384 )/ 119 /,Ic( 1385 )/ 435 /
C-----<<113 Q >>
      Data Ic( 1386 )/ 375 /,Ic( 1387 )/ 325 /
      Data Ic( 1388 )/ 1   /,Ic( 1389 )/ 118 /
      Data Ic( 1390 )/ 503 /,Ic( 1391 )/ 324 /
      Data Ic( 1392 )/ 27  /,Ic( 1393 )/ 5   /
      Data Ic( 1394 )/ 510 /,Ic( 1395 )/ 376 /
      Data Ic( 1396 )/ 485 /,Ic( 1397 )/ 377 /
      Data Ic( 1398 )/ 443 /
C-----<<114 R >>
      Data Ic( 1399 )/ 378 /,Ic( 1400 )/ 488 /
      Data Ic( 1401 )/ 379 /,Ic( 1402 )/ 415 /
      Data Ic( 1403 )/ 17  /,Ic( 1404 )/ 106 /
      Data Ic( 1405 )/ 44  /,Ic( 1406 )/ 32  /
      Data Ic( 1407 )/ 425 /,Ic( 1408 )/ 346 /
      Data Ic( 1409 )/ 442 /,Ic( 1410 )/ 356 /
      Data Ic( 1411 )/ 514 /,Ic( 1412 )/ 435 /
C-----<<115 S >>
      Data Ic( 1413 )/ 380 /,Ic( 1414 )/ 47  /
      Data Ic( 1415 )/ 68  /,Ic( 1416 )/ 416 /
      Data Ic( 1417 )/ 45  /,Ic( 1418 )/ 30  /
      Data Ic( 1419 )/ 508 /,Ic( 1420 )/ 11  /
      Data Ic( 1421 )/ 499 /,Ic( 1422 )/ 134 /
      Data Ic( 1423 )/ 37  /,Ic( 1424 )/ 50  /
      Data Ic( 1425 )/ 1   /,Ic( 1426 )/ 381 /
      Data Ic( 1427 )/ 425 /,Ic( 1428 )/ 10  /
      Data Ic( 1429 )/ 37  /,Ic( 1430 )/ 134 /
      Data Ic( 1431 )/ 449 /,Ic( 1432 )/ 10  /
      Data Ic( 1433 )/ 423 /,Ic( 1434 )/ 381 /
      Data Ic( 1435 )/ 499 /,Ic( 1436 )/ 134 /
      Data Ic( 1437 )/ 37  /,Ic( 1438 )/ 34  /
      Data Ic( 1439 )/ 33  /,Ic( 1440 )/ 14  /
      Data Ic( 1441 )/ 44  /,Ic( 1442 )/ 30  /
      Data Ic( 1443 )/ 45  /,Ic( 1444 )/ 17  /
      Data Ic( 1445 )/ 28  /,Ic( 1446 )/ 68  /
      Data Ic( 1447 )/ 47  /
C-----<<116 T >>
      Data Ic( 1448 )/ 382 /,Ic( 1449 )/ 38  /
      Data Ic( 1450 )/ 44  /,Ic( 1451 )/ 106 /
      Data Ic( 1452 )/ 45  /,Ic( 1453 )/ 17  /
      Data Ic( 1454 )/ 149 /,Ic( 1455 )/ 8   /
```

```
      Data Ic( 1456 )/ 49  /,Ic( 1457 )/ 92  /
      Data Ic( 1458 )/ 50  /,Ic( 1459 )/ 10  /
      Data Ic( 1460 )/ 1   /,Ic( 1461 )/ 112 /
      Data Ic( 1462 )/ 286 /,Ic( 1463 )/ 431 /
      Data Ic( 1464 )/ 383 /,Ic( 1465 )/ 188 /
C-----<<117 U >>
      Data Ic( 1466 )/ 371 /,Ic( 1467 )/ 305 /
      Data Ic( 1468 )/ 50  /,Ic( 1469 )/ 10  /
      Data Ic( 1470 )/ 37  /,Ic( 1471 )/ 139 /
      Data Ic( 1472 )/ 97  /,Ic( 1473 )/ 440 /
      Data Ic( 1474 )/ 1   /,Ic( 1475 )/ 384 /
      Data Ic( 1476 )/ 305 /,Ic( 1477 )/ 431 /
      Data Ic( 1478 )/ 383 /,Ic( 1479 )/ 191 /
      Data Ic( 1480 )/ 177 /,Ic( 1481 )/ 50  /
      Data Ic( 1482 )/ 10  /,Ic( 1483 )/ 1   /
      Data Ic( 1484 )/ 385 /,Ic( 1485 )/ 6   /
      Data Ic( 1486 )/ 191 /,Ic( 1487 )/ 1   /
      Data Ic( 1488 )/ 321 /,Ic( 1489 )/ 497 /
      Data Ic( 1490 )/ 386 /,Ic( 1491 )/ 462 /
      Data Ic( 1492 )/ 387 /,Ic( 1493 )/ 505 /
C-----<<118 V >>
      Data Ic( 1494 )/ 371 /,Ic( 1495 )/ 388 /
      Data Ic( 1496 )/ 47  /,Ic( 1497 )/ 319 /
      Data Ic( 1498 )/ 476 /,Ic( 1499 )/ 389 /
      Data Ic( 1500 )/ 489 /,Ic( 1501 )/ 320 /
      Data Ic( 1502 )/ 489 /,Ic( 1503 )/ 391 /
      Data Ic( 1504 )/ 490 /
C-----<<119 W >>
      Data Ic( 1505 )/ 392 /,Ic( 1506 )/ 393 /
      Data Ic( 1507 )/ 148 /,Ic( 1508 )/ 394 /
      Data Ic( 1509 )/ 393 /,Ic( 1510 )/ 148 /
      Data Ic( 1511 )/ 395 /,Ic( 1512 )/ 476 /
      Data Ic( 1513 )/ 396 /,Ic( 1514 )/ 491 /
      Data Ic( 1515 )/ 332 /,Ic( 1516 )/ 491 /
      Data Ic( 1517 )/ 329 /,Ic( 1518 )/ 491 /
      Data Ic( 1519 )/ 154 /,Ic( 1520 )/ 27  /
      Data Ic( 1521 )/ 491 /,Ic( 1522 )/ 398 /
      Data Ic( 1523 )/ 481 /,Ic( 1524 )/ 110 /
      Data Ic( 1525 )/ 434 /,Ic( 1526 )/ 269 /
      Data Ic( 1527 )/ 44  /
C-----<<120 X >>
      Data Ic( 1528 )/ 371 /,Ic( 1529 )/ 399 /
      Data Ic( 1530 )/ 1   /,Ic( 1531 )/ 400 /
      Data Ic( 1532 )/ 399 /,Ic( 1533 )/ 1   /
      Data Ic( 1534 )/ 400 /,Ic( 1535 )/ 399 /
      Data Ic( 1536 )/ 1   /,Ic( 1537 )/ 364 /
      Data Ic( 1538 )/ 37  /,Ic( 1539 )/ 401 /
      Data Ic( 1540 )/ 434 /,Ic( 1541 )/ 402 /
      Data Ic( 1542 )/ 509 /,Ic( 1543 )/ 139 /
      Data Ic( 1544 )/ 457 /,Ic( 1545 )/ 403 /
      Data Ic( 1546 )/ 457 /,Ic( 1547 )/ 139 /
```

```
      Data Ic( 1548 )/ 509 /,Ic( 1549 )/ 386 /
      Data Ic( 1550 )/ 449 /,Ic( 1551 )/ 110 /
      Data Ic( 1552 )/ 434 /,Ic( 1553 )/ 284 /
      Data Ic( 1554 )/ 434 /,Ic( 1555 )/ 401 /
      Data Ic( 1556 )/ 449 /,Ic( 1557 )/ 352 /
      Data Ic( 1558 )/ 434 /,Ic( 1559 )/ 110 /
      Data Ic( 1560 )/ 37  /
C-----<<121 Y >>
      Data Ic( 1561 )/ 371 /,Ic( 1562 )/ 388 /
      Data Ic( 1563 )/ 1   /,Ic( 1564 )/ 340 /
      Data Ic( 1565 )/ 489 /,Ic( 1566 )/ 320 /
      Data Ic( 1567 )/ 489 /,Ic( 1568 )/ 404 /
      Data Ic( 1569 )/ 15  /,Ic( 1570 )/ 7   /
      Data Ic( 1571 )/ 4   /,Ic( 1572 )/ 106 /
      Data Ic( 1573 )/ 11  /,Ic( 1574 )/ 10  /
      Data Ic( 1575 )/ 27  /,Ic( 1576 )/ 97  /
      Data Ic( 1577 )/ 49  /,Ic( 1578 )/ 405 /
      Data Ic( 1579 )/ 319 /,Ic( 1580 )/ 476 /
      Data Ic( 1581 )/ 200 /,Ic( 1582 )/ 490 /
C-----<<122 Z >>
      Data Ic( 1583 )/ 406 /,Ic( 1584 )/ 407 /
      Data Ic( 1585 )/ 198 /,Ic( 1586 )/ 165 /
      Data Ic( 1587 )/ 423 /,Ic( 1588 )/ 8   /
      Data Ic( 1589 )/ 407 /,Ic( 1590 )/ 477 /
      Data Ic( 1591 )/ 408 /,Ic( 1592 )/ 200 /
      Data Ic( 1593 )/ 68  /,Ic( 1594 )/ 56  /
      Data Ic( 1595 )/ 1   /,Ic( 1596 )/ 7   /
      Data Ic( 1597 )/ 480 /,Ic( 1598 )/ 409 /
      Data Ic( 1599 )/ 469 /
C-----<<410   >>
      Data Ic( 1600 )/ 9   /,Ic( 1601 )/ 1   /
C-----<<411   >>
      Data Ic( 1602 )/ 14  /,Ic( 1603 )/ 15  /
      Data Ic( 1604 )/ 10  /,Ic( 1605 )/ 521 /
C-----<<412   >>
      Data Ic( 1606 )/ 411 /,Ic( 1607 )/ 16  /
      Data Ic( 1608 )/ 17  /,Ic( 1609 )/ 410 /
      Data Ic( 1610 )/ 15  /,Ic( 1611 )/ 15  /
      Data Ic( 1612 )/ 519 /
C-----<<413   >>
      Data Ic( 1613 )/ 412 /,Ic( 1614 )/ 9   /
C-----<<414   >>
      Data Ic( 1615 )/ 19  /,Ic( 1616 )/ 17  /
      Data Ic( 1617 )/ 14  /,Ic( 1618 )/ 20  /
      Data Ic( 1619 )/ 21  /
C-----<<415   >>
      Data Ic( 1620 )/ 16  /,Ic( 1621 )/ 9   /
      Data Ic( 1622 )/ 11  /,Ic( 1623 )/ 27  /
      Data Ic( 1624 )/ 4   /
C-----<<416   >>
      Data Ic( 1625 )/ 28  /,Ic( 1626 )/ 17  /
```

```
C------<<417   >>
        Data Ic( 1627 )/ 31  /,Ic( 1628 )/ 32  /
C------<<418   >>
        Data Ic( 1629 )/ 417 /,Ic( 1630 )/ 33  /
        Data Ic( 1631 )/ 34  /,Ic( 1632 )/ 35  /
        Data Ic( 1633 )/ 36  /,Ic( 1634 )/ 517 /
        Data Ic( 1635 )/ 498 /,Ic( 1636 )/ 1   /
        Data Ic( 1637 )/ 39  /,Ic( 1638 )/ 499 /
        Data Ic( 1639 )/ 36  /,Ic( 1640 )/ 517 /
        Data Ic( 1641 )/ 1   /,Ic( 1642 )/ 40  /
        Data Ic( 1643 )/ 38  /,Ic( 1644 )/ 11  /
        Data Ic( 1645 )/ 499 /,Ic( 1646 )/ 36  /
        Data Ic( 1647 )/ 35  /,Ic( 1648 )/ 34  /
        Data Ic( 1649 )/ 498 /,Ic( 1650 )/ 500 /
C------<<419   >>
        Data Ic( 1651 )/ 4   /,Ic( 1652 )/ 27  /
        Data Ic( 1653 )/ 11  /,Ic( 1654 )/ 9   /
        Data Ic( 1655 )/ 16  /
C------<<420   >>
        Data Ic( 1656 )/ 28  /,Ic( 1657 )/ 46  /
        Data Ic( 1658 )/ 47  /
C------<<421   >>
        Data Ic( 1659 )/ 28  /,Ic( 1660 )/ 45  /
C------<<422   >>
        Data Ic( 1661 )/ 44  /,Ic( 1662 )/ 38  /
C------<<423   >>
        Data Ic( 1663 )/ 58  /,Ic( 1664 )/ 1   /
C------<<424   >>
        Data Ic( 1665 )/ 71  /,Ic( 1666 )/ 1   /
C------<<425   >>
        Data Ic( 1667 )/ 73  /,Ic( 1668 )/ 1   /
C------<<426   >>
        Data Ic( 1669 )/ 34  /,Ic( 1670 )/ 50  /
C------<<427   >>
        Data Ic( 1671 )/ 16  /,Ic( 1672 )/ 94  /
        Data Ic( 1673 )/ 15  /,Ic( 1674 )/ 95  /
C------<<428   >>
        Data Ic( 1675 )/ 97  /,Ic( 1676 )/ 47  /
C------<<429   >>
        Data Ic( 1677 )/ 428 /,Ic( 1678 )/ 46  /
        Data Ic( 1679 )/ 17  /,Ic( 1680 )/ 9   /
        Data Ic( 1681 )/ 411 /,Ic( 1682 )/ 423 /
        Data Ic( 1683 )/ 98  /,Ic( 1684 )/ 501 /
        Data Ic( 1685 )/ 502 /
C------<<430   >>
        Data Ic( 1686 )/ 31  /,Ic( 1687 )/ 106 /
C------<<431   >>
        Data Ic( 1688 )/ 34  /,Ic( 1689 )/ 1   /
C------<<432   >>
        Data Ic( 1690 )/ 28  /,Ic( 1691 )/ 1   /
C------<<433   >>
```

```
      Data Ic( 1692 )/ 117 /,Ic( 1693 )/ 1   /
C-----<<434   >>
      Data Ic( 1694 )/ 44  /,Ic( 1695 )/ 1   /
C-----<<435   >>
      Data Ic( 1696 )/ 434 /,Ic( 1697 )/ 49  /
      Data Ic( 1698 )/ 504 /,Ic( 1699 )/ 125 /
      Data Ic( 1700 )/ 505 /
C-----<<436   >>
      Data Ic( 1701 )/ 126 /,Ic( 1702 )/ 501 /
      Data Ic( 1703 )/ 16  /,Ic( 1704 )/ 493 /
      Data Ic( 1705 )/ 9   /,Ic( 1706 )/ 17  /
      Data Ic( 1707 )/ 16  /,Ic( 1708 )/ 47  /
      Data Ic( 1709 )/ 5   /,Ic( 1710 )/ 110 /
      Data Ic( 1711 )/ 506 /
C-----<<437   >>
      Data Ic( 1712 )/ 10  /,Ic( 1713 )/ 34  /
C-----<<438   >>
      Data Ic( 1714 )/ 437 /,Ic( 1715 )/ 11  /
      Data Ic( 1716 )/ 38  /
C-----<<439   >>
      Data Ic( 1717 )/ 34  /,Ic( 1718 )/ 11  /
      Data Ic( 1719 )/ 504 /
C-----<<440   >>
      Data Ic( 1720 )/ 5   /,Ic( 1721 )/ 47  /
C-----<<441   >>
      Data Ic( 1722 )/ 34  /,Ic( 1723 )/ 33  /
      Data Ic( 1724 )/ 38  /
C-----<<442   >>
      Data Ic( 1725 )/ 113 /,Ic( 1726 )/ 1   /
C-----<<443   >>
      Data Ic( 1727 )/ 442 /,Ic( 1728 )/ 146 /
      Data Ic( 1729 )/ 435 /
C-----<<444   >>
      Data Ic( 1730 )/ 110 /,Ic( 1731 )/ 49  /
C-----<<445   >>
      Data Ic( 1732 )/ 149 /,Ic( 1733 )/ 150 /
C-----<<446   >>
      Data Ic( 1734 )/ 34  /,Ic( 1735 )/ 68  /
      Data Ic( 1736 )/ 38  /
C-----<<447   >>
      Data Ic( 1737 )/ 446 /,Ic( 1738 )/ 1   /
      Data Ic( 1739 )/ 154 /,Ic( 1740 )/ 37  /
      Data Ic( 1741 )/ 516 /,Ic( 1742 )/ 68  /
      Data Ic( 1743 )/ 73  /,Ic( 1744 )/ 14  /
      Data Ic( 1745 )/ 434 /
C-----<<448   >>
      Data Ic( 1746 )/ 71  /,Ic( 1747 )/ 105 /
C-----<<449   >>
      Data Ic( 1748 )/ 37  /,Ic( 1749 )/ 1   /
C-----<<450   >>
      Data Ic( 1750 )/ 5   /,Ic( 1751 )/ 97  /
```

```
C------<<451   >>
       Data Ic( 1752 )/ 14  /,Ic( 1753 )/ 38  /
C------<<452   >>
       Data Ic( 1754 )/ 38  /,Ic( 1755 )/ 68  /
       Data Ic( 1756 )/ 431 /
C------<<453   >>
       Data Ic( 1757 )/ 452 /,Ic( 1758 )/ 123 /
       Data Ic( 1759 )/ 47  /,Ic( 1760 )/ 165 /
       Data Ic( 1761 )/ 432 /
C------<<454   >>
       Data Ic( 1762 )/ 16  /,Ic( 1763 )/ 7   /
       Data Ic( 1764 )/ 11  /,Ic( 1765 )/ 106 /
C------<<455   >>
       Data Ic( 1766 )/ 14  /,Ic( 1767 )/ 1   /
C------<<456   >>
       Data Ic( 1768 )/ 185 /,Ic( 1769 )/ 1   /
C------<<457   >>
       Data Ic( 1770 )/ 190 /,Ic( 1771 )/ 1   /
C------<<458   >>
       Data Ic( 1772 )/ 194 /,Ic( 1773 )/ 195 /
       Data Ic( 1774 )/ 1   /,Ic( 1775 )/ 196 /
       Data Ic( 1776 )/ 433 /
C------<<459   >>
       Data Ic( 1777 )/ 458 /,Ic( 1778 )/ 518 /
       Data Ic( 1779 )/ 197 /
C------<<460   >>
       Data Ic( 1780 )/ 459 /,Ic( 1781 )/ 198 /
       Data Ic( 1782 )/ 105 /
C------<<461   >>
       Data Ic( 1783 )/ 441 /,Ic( 1784 )/ 1   /
       Data Ic( 1785 )/ 202 /,Ic( 1786 )/ 517 /
       Data Ic( 1787 )/ 111 /,Ic( 1788 )/ 38  /
       Data Ic( 1789 )/ 434 /
C------<<462   >>
       Data Ic( 1790 )/ 449 /,Ic( 1791 )/ 17  /
       Data Ic( 1792 )/ 431 /
C------<<463   >>
       Data Ic( 1793 )/ 462 /,Ic( 1794 )/ 125 /
       Data Ic( 1795 )/ 520 /
C------<<464   >>
       Data Ic( 1796 )/ 463 /,Ic( 1797 )/ 204 /
       Data Ic( 1798 )/ 435 /
C------<<465   >>
       Data Ic( 1799 )/ 205 /,Ic( 1800 )/ 148 /
       Data Ic( 1801 )/ 20  /,Ic( 1802 )/ 445 /
       Data Ic( 1803 )/ 45  /,Ic( 1804 )/ 151 /
       Data Ic( 1805 )/ 417 /,Ic( 1806 )/ 511 /
       Data Ic( 1807 )/ 448 /,Ic( 1808 )/ 139 /
       Data Ic( 1809 )/ 97  /
C------<<466   >>
       Data Ic( 1810 )/ 459 /,Ic( 1811 )/ 494 /
```

```
      Data Ic( 1812 )/ 213 /,Ic( 1813 )/ 457 /
      Data Ic( 1814 )/ 68  /,Ic( 1815 )/ 214 /
      Data Ic( 1816 )/ 58  /,Ic( 1817 )/ 62  /
      Data Ic( 1818 )/ 1   /,Ic( 1819 )/ 215 /
C-----<<467   >>
      Data Ic( 1820 )/ 216 /,Ic( 1821 )/ 75  /
      Data Ic( 1822 )/ 1   /
C-----<<468   >>
      Data Ic( 1823 )/ 463 /,Ic( 1824 )/ 218 /
      Data Ic( 1825 )/ 515 /,Ic( 1826 )/ 164 /
      Data Ic( 1827 )/ 32  /,Ic( 1828 )/ 424 /
      Data Ic( 1829 )/ 46  /,Ic( 1830 )/ 152 /
      Data Ic( 1831 )/ 219 /,Ic( 1832 )/ 1   /
      Data Ic( 1833 )/ 220 /,Ic( 1834 )/ 435 /
C-----<<469   >>
      Data Ic( 1835 )/ 221 /,Ic( 1836 )/ 1   /
      Data Ic( 1837 )/ 31  /,Ic( 1838 )/ 125 /
      Data Ic( 1839 )/ 1   /,Ic( 1840 )/ 32  /
      Data Ic( 1841 )/ 109 /
C-----<<470   >>
      Data Ic( 1842 )/ 442 /,Ic( 1843 )/ 182 /
C-----<<471   >>
      Data Ic( 1844 )/ 442 /,Ic( 1845 )/ 225 /
      Data Ic( 1846 )/ 462 /,Ic( 1847 )/ 125 /
      Data Ic( 1848 )/ 504 /,Ic( 1849 )/ 49  /
      Data Ic( 1850 )/ 44  /
C-----<<472   >>
      Data Ic( 1851 )/ 162 /,Ic( 1852 )/ 195 /
      Data Ic( 1853 )/ 1   /,Ic( 1854 )/ 196 /
      Data Ic( 1855 )/ 433 /,Ic( 1856 )/ 518 /
C-----<<473   >>
      Data Ic( 1857 )/ 510 /,Ic( 1858 )/ 196 /
      Data Ic( 1859 )/ 433 /
C-----<<474   >>
      Data Ic( 1860 )/ 435 /,Ic( 1861 )/ 1   /
C-----<<475   >>
      Data Ic( 1862 )/ 473 /,Ic( 1863 )/ 518 /
      Data Ic( 1864 )/ 197 /,Ic( 1865 )/ 442 /
      Data Ic( 1866 )/ 233 /
C-----<<476   >>
      Data Ic( 1867 )/ 45  /,Ic( 1868 )/ 1   /
C-----<<477   >>
      Data Ic( 1869 )/ 1   /,Ic( 1870 )/ 7   /
C-----<<478   >>
      Data Ic( 1871 )/ 104 /,Ic( 1872 )/ 27  /
      Data Ic( 1873 )/ 105 /,Ic( 1874 )/ 71  /
      Data Ic( 1875 )/ 34  /,Ic( 1876 )/ 62  /
      Data Ic( 1877 )/ 33  /,Ic( 1878 )/ 58  /
      Data Ic( 1879 )/ 38  /,Ic( 1880 )/ 32  /
      Data Ic( 1881 )/ 430 /,Ic( 1882 )/ 29  /
      Data Ic( 1883 )/ 150 /,Ic( 1884 )/ 28  /
```

```
      Data Ic( 1885 )/ 57  /,Ic( 1886 )/ 46  /
      Data Ic( 1887 )/ 56  /,Ic( 1888 )/ 47  /
      Data Ic( 1889 )/ 49  /,Ic( 1890 )/ 110 /
      Data Ic( 1891 )/ 512 /,Ic( 1892 )/ 449 /
      Data Ic( 1893 )/ 210 /,Ic( 1894 )/ 511 /
      Data Ic( 1895 )/ 1   /,Ic( 1896 )/ 123 /
      Data Ic( 1897 )/ 47  /,Ic( 1898 )/ 148 /
      Data Ic( 1899 )/ 114 /,Ic( 1900 )/ 149 /
      Data Ic( 1901 )/ 432 /,Ic( 1902 )/ 270 /
      Data Ic( 1903 )/ 97  /,Ic( 1904 )/ 109 /
      Data Ic( 1905 )/ 56  /,Ic( 1906 )/ 114 /
      Data Ic( 1907 )/ 57  /,Ic( 1908 )/ 107 /
C-----<<479   >>
      Data Ic( 1909 )/ 34  /,Ic( 1910 )/ 27  /
      Data Ic( 1911 )/ 47  /,Ic( 1912 )/ 4   /
      Data Ic( 1913 )/ 1   /,Ic( 1914 )/ 30  /
      Data Ic( 1915 )/ 502 /,Ic( 1916 )/ 7   /
      Data Ic( 1917 )/ 1   /
C-----<<480   >>
      Data Ic( 1918 )/ 43  /,Ic( 1919 )/ 1   /
      Data Ic( 1920 )/ 147 /,Ic( 1921 )/ 128 /
      Data Ic( 1922 )/ 1   /,Ic( 1923 )/ 282 /
      Data Ic( 1924 )/ 283 /,Ic( 1925 )/ 1   /
C-----<<481   >>
      Data Ic( 1926 )/ 442 /,Ic( 1927 )/ 113 /
      Data Ic( 1928 )/ 457 /,Ic( 1929 )/ 298 /
      Data Ic( 1930 )/ 105 /,Ic( 1931 )/ 1   /
C-----<<482   >>
      Data Ic( 1932 )/ 47  /,Ic( 1933 )/ 97  /
      Data Ic( 1934 )/ 496 /,Ic( 1935 )/ 430 /
      Data Ic( 1936 )/ 45  /,Ic( 1937 )/ 432 /
      Data Ic( 1938 )/ 326 /,Ic( 1939 )/ 447 /
C-----<<483   >>
      Data Ic( 1940 )/ 73  /,Ic( 1941 )/ 11  /
      Data Ic( 1942 )/ 50  /,Ic( 1943 )/ 448 /
      Data Ic( 1944 )/ 27  /
C-----<<484   >>
      Data Ic( 1945 )/ 331 /,Ic( 1946 )/ 191 /
      Data Ic( 1947 )/ 510 /
C-----<<485   >>
      Data Ic( 1948 )/ 421 /,Ic( 1949 )/ 106 /
      Data Ic( 1950 )/ 417 /,Ic( 1951 )/ 483 /
      Data Ic( 1952 )/ 428 /,Ic( 1953 )/ 1   /
      Data Ic( 1954 )/ 333 /,Ic( 1955 )/ 525 /
      Data Ic( 1956 )/ 449 /
C-----<<486   >>
      Data Ic( 1957 )/ 474 /,Ic( 1958 )/ 352 /
C-----<<487   >>
      Data Ic( 1959 )/ 11  /,Ic( 1960 )/ 27  /
      Data Ic( 1961 )/ 4   /,Ic( 1962 )/ 106 /
      Data Ic( 1963 )/ 477 /,Ic( 1964 )/ 11  /
```

```
      Data Ic( 1965 )/ 1   /,Ic( 1966 )/ 17  /
      Data Ic( 1967 )/ 27  /,Ic( 1968 )/ 1   /
      Data Ic( 1969 )/ 353 /
C-----<<488   >>
      Data Ic( 1970 )/ 6   /,Ic( 1971 )/ 1   /
      Data Ic( 1972 )/ 354 /,Ic( 1973 )/ 497 /
C-----<<489   >>
      Data Ic( 1974 )/ 390 /,Ic( 1975 )/ 1   /
C-----<<490   >>
      Data Ic( 1976 )/ 509 /,Ic( 1977 )/ 139 /
      Data Ic( 1978 )/ 457 /,Ic( 1979 )/ 309 /
      Data Ic( 1980 )/ 35  /,Ic( 1981 )/ 1   /
      Data Ic( 1982 )/ 49  /,Ic( 1983 )/ 434 /
      Data Ic( 1984 )/ 284 /,Ic( 1985 )/ 14  /
C-----<<491   >>
      Data Ic( 1986 )/ 397 /,Ic( 1987 )/ 1   /
C-----<<492   >>
      Data Ic( 1988 )/ 52  /,Ic( 1989 )/ 53  /
      Data Ic( 1990 )/ 37  /
C-----<<493   >>
      Data Ic( 1991 )/ 7   /,Ic( 1992 )/ 10  /
      Data Ic( 1993 )/ 15  /,Ic( 1994 )/ 14  /
C-----<<494   >>
      Data Ic( 1995 )/ 144 /,Ic( 1996 )/ 20  /
      Data Ic( 1997 )/ 212 /,Ic( 1998 )/ 1   /
C-----<<495   >>
      Data Ic( 1999 )/ 47  /,Ic( 2000 )/ 450 /
      Data Ic( 2001 )/ 139 /,Ic( 2002 )/ 37  /
      Data Ic( 2003 )/ 516 /,Ic( 2004 )/ 305 /
      Data Ic( 2005 )/ 1   /,Ic( 2006 )/ 350 /
      Data Ic( 2007 )/ 50  /,Ic( 2008 )/ 179 /
      Data Ic( 2009 )/ 1   /,Ic( 2010 )/ 351 /
      Data Ic( 2011 )/ 516 /,Ic( 2012 )/ 177 /
      Data Ic( 2013 )/ 1   /
C-----<<496   >>
      Data Ic( 2014 )/ 27  /,Ic( 2015 )/ 105 /
      Data Ic( 2016 )/ 71  /,Ic( 2017 )/ 50  /
      Data Ic( 2018 )/ 11  /,Ic( 2019 )/ 73  /
      Data Ic( 2020 )/ 32  /
C-----<<497   >>
      Data Ic( 2021 )/ 355 /,Ic( 2022 )/ 1   /
      Data Ic( 2023 )/ 321 /,Ic( 2024 )/ 191 /
      Data Ic( 2025 )/ 6   /,Ic( 2026 )/ 1   /
C-----<<498   >>
      Data Ic( 2027 )/ 33  /,Ic( 2028 )/ 38  /
C-----<<499   >>
      Data Ic( 2029 )/ 34  /,Ic( 2030 )/ 37  /
C-----<<500   >>
      Data Ic( 2031 )/ 14  /,Ic( 2032 )/ 31  /
C-----<<501   >>
      Data Ic( 2033 )/ 15  /,Ic( 2034 )/ 519 /
```

```
        Data Ic( 2035 )/ 410 /
C------<<502   >>
        Data Ic( 2036 )/ 34  /,Ic( 2037 )/ 10  /
C------<<503   >>
        Data Ic( 2038 )/ 119 /,Ic( 2039 )/ 1   /
C------<<504   >>
        Data Ic( 2040 )/ 38  /,Ic( 2041 )/ 1   /
C------<<505   >>
        Data Ic( 2042 )/ 431 /,Ic( 2043 )/ 17  /
        Data Ic( 2044 )/ 37  /
C------<<506   >>
        Data Ic( 2045 )/ 127 /,Ic( 2046 )/ 105 /
C------<<507   >>
        Data Ic( 2047 )/ 131 /,Ic( 2048 )/ 37  /
C------<<508   >>
        Data Ic( 2049 )/ 44  /,Ic( 2050 )/ 14  /
C------<<509   >>
        Data Ic( 2051 )/ 192 /,Ic( 2052 )/ 1   /
C------<<510   >>
        Data Ic( 2053 )/ 195 /,Ic( 2054 )/ 1   /
C------<<511   >>
        Data Ic( 2055 )/ 38  /,Ic( 2056 )/ 73  /
        Data Ic( 2057 )/ 111 /,Ic( 2058 )/ 50  /
        Data Ic( 2059 )/ 34  /
C------<<512   >>
        Data Ic( 2060 )/ 44  /,Ic( 2061 )/ 43  /
        Data Ic( 2062 )/ 58  /,Ic( 2063 )/ 111 /
        Data Ic( 2064 )/ 62  /,Ic( 2065 )/ 51  /
C------<<513   >>
        Data Ic( 2066 )/ 442 /,Ic( 2067 )/ 139 /
C------<<514   >>
        Data Ic( 2068 )/ 462 /,Ic( 2069 )/ 357 /
C------<<515   >>
        Data Ic( 2070 )/ 134 /,Ic( 2071 )/ 1   /
        Data Ic( 2072 )/ 29  /,Ic( 2073 )/ 35  /
        Data Ic( 2074 )/ 1   /,Ic( 2075 )/ 150 /
        Data Ic( 2076 )/ 51  /,Ic( 2077 )/ 1   /
C------<<516   >>
        Data Ic( 2078 )/ 10  /,Ic( 2079 )/ 50  /
C------<<517   >>
        Data Ic( 2080 )/ 37  /,Ic( 2081 )/ 34  /
C------<<518   >>
        Data Ic( 2082 )/ 196 /,Ic( 2083 )/ 510 /
C------<<519   >>
        Data Ic( 2084 )/ 7   /,Ic( 2085 )/ 16  /
C------<<520   >>
        Data Ic( 2086 )/ 504 /,Ic( 2087 )/ 49  /
        Data Ic( 2088 )/ 434 /
C------<<521   >>
        Data Ic( 2089 )/ 7   /,Ic( 2090 )/ 5   /
C------<<522   >>
```

```
      Data Ic( 2091 )/ 16  /,Ic( 2092 )/ 5   /
      Data Ic( 2093 )/ 493 /,Ic( 2094 )/ 410 /
C-----<<523   >>
      Data Ic( 2095 )/ 521 /,Ic( 2096 )/ 16  /
      Data Ic( 2097 )/ 17  /,Ic( 2098 )/ 9   /
      Data Ic( 2099 )/ 14  /
C-----<<524   >>
      Data Ic( 2100 )/ 73  /,Ic( 2101 )/ 68  /
      Data Ic( 2102 )/ 50  /
C-----<<525   >>
      Data Ic( 2103 )/ 452 /,Ic( 2104 )/ 329 /
      Data Ic( 2105 )/ 508 /,Ic( 2106 )/ 524 /
      Data Ic( 2107 )/ 10  /
C-----<<526   >>
      Data Ic( 2108 )/ 307 /,Ic( 2109 )/ 495 /
      Data Ic( 2110 )/ 291 /,Ic( 2111 )/ 513 /
      Data Ic( 2112 )/ 442 /,Ic( 2113 )/ 0   /
C
C     ********************************************************
C     *              CHARACTER INDEX TABLE                   *
C     ********************************************************
C
      Data In( 1 )/ 0   /,In( 2 )/ 15  /
      Data In( 3 )/ 20  /,In( 4 )/ 20  /
      Data In( 5 )/ 37  /,In( 6 )/ 37  /
      Data In( 7 )/ 90  /,In( 8 )/ 92  /
      Data In( 9 )/ 117 /,In( 10)/ 141 /
      Data In( 11 )/ 166 /,In( 12 )/ 174 /
      Data In( 13 )/ 176 /,In( 14 )/ 178 /
      Data In( 15 )/ 180 /,In( 16 )/ 185 /
      Data In( 17 )/ 225 /,In( 18 )/ 238 /
      Data In( 19 )/ 271 /,In( 20 )/ 304 /
      Data In( 21 )/ 315 /,In( 22 )/ 340 /
      Data In( 23 )/ 390 /,In( 24 )/ 425 /
      Data In( 25 )/ 471 /,In( 26 )/ 523 /
      Data In( 27 )/ 528 /,In( 28 )/ 533 /
      Data In( 29 )/ 533 /,In( 30 )/ 538 /
      Data In( 31 )/ 538 /,In( 32 )/ 562 /
      Data In( 33 )/ 562 /,In( 34 )/ 585 /
      Data In( 35 )/ 606 /,In( 36 )/ 616 /
      Data In( 37 )/ 644 /,In( 38 )/ 652 /
      Data In( 39 )/ 655 /,In( 40 )/ 674 /
      Data In( 41 )/ 697 /,In( 42 )/ 701 /
      Data In( 43 )/ 722 /,In( 44 )/ 753 /
      Data In( 45 )/ 762 /,In( 46 )/ 792 /
      Data In( 47 )/ 814 /,In( 48 )/ 816 /
      Data In( 49 )/ 827 /,In( 50 )/ 840 /
      Data In( 51 )/ 864 /,In( 52 )/ 877 /
      Data In( 53 )/ 892 /,In( 54 )/ 918 /
      Data In( 55 )/ 937 /,In( 56 )/ 967 /
      Data In( 57 )/ 1002 /,In( 58 )/ 1031 /
```

```
Data In( 59 )/ 1048 /,In( 60 )/ 1059 /
Data In( 61 )/ 1061 /,In( 62 )/ 1061 /
Data In( 63 )/ 1061 /,In( 64 )/ 1061 /
Data In( 65 )/ 1061 /,In( 66 )/ 1103 /
Data In( 67 )/ 1120 /,In( 68 )/ 1132 /
Data In( 69 )/ 1145 /,In( 70 )/ 1169 /
Data In( 71 )/ 1189 /,In( 72 )/ 1252 /
Data In( 73 )/ 1260 /,In( 74 )/ 1268 /
Data In( 75 )/ 1290 /,In( 76 )/ 1321 /
Data In( 77 )/ 1329 /,In( 78 )/ 1344 /
Data In( 79 )/ 1351 /,In( 80 )/ 1375 /
Data In( 81 )/ 1385 /,In( 82 )/ 1398 /
Data In( 83 )/ 1412 /,In( 84 )/ 1447 /
Data In( 85 )/ 1465 /,In( 86 )/ 1493 /
Data In( 87 )/ 1504 /,In( 88 )/ 1527 /
Data In( 89 )/ 1560 /,In( 90 )/ 1582 /
Data In( 91 )/ 1599 /,In( 92 )/ 1601 /
Data In( 93 )/ 1605 /,In( 94 )/ 1612 /
Data In( 95 )/ 1614 /,In( 96 )/ 1619 /
Data In( 97 )/ 1624 /,In( 98 )/ 1626 /
Data In( 99 )/ 1628 /,In( 100 )/ 1650 /
Data In( 101 )/ 1655 /,In( 102 )/ 1658 /
Data In( 103 )/ 1660 /,In( 104 )/ 1662 /
Data In( 105 )/ 1664 /,In( 106 )/ 1666 /
Data In( 107 )/ 1668 /,In( 108 )/ 1670 /
Data In( 109 )/ 1674 /,In( 110 )/ 1676 /
Data In( 111 )/ 1685 /,In( 112 )/ 1687 /
Data In( 113 )/ 1689 /,In( 114 )/ 1691 /
Data In( 115 )/ 1693 /,In( 116 )/ 1695 /
Data In( 117 )/ 1700 /,In( 118 )/ 1711 /
Data In( 119 )/ 1713 /,In( 120 )/ 1716 /
Data In( 121 )/ 1719 /,In( 122 )/ 1721 /
Data In( 123 )/ 1724 /,In( 124 )/ 1726 /
Data In( 125 )/ 1729 /,In( 126 )/ 1731 /
Data In( 127 )/ 1733 /,In( 128 )/ 1736 /
Data In( 129 )/ 1745 /,In( 130 )/ 1747 /
Data In( 131 )/ 1749 /,In( 132 )/ 1751 /
Data In( 133 )/ 1753 /,In( 134 )/ 1756 /
Data In( 135 )/ 1761 /,In( 136 )/ 1765 /
Data In( 137 )/ 1767 /,In( 138 )/ 1769 /
Data In( 139 )/ 1771 /,In( 140 )/ 1776 /
Data In( 141 )/ 1779 /,In( 142 )/ 1782 /
Data In( 143 )/ 1789 /,In( 144 )/ 1792 /
Data In( 145 )/ 1795 /,In( 146 )/ 1798 /
Data In( 147 )/ 1809 /,In( 148 )/ 1819 /
Data In( 149 )/ 1822 /,In( 150 )/ 1834 /
Data In( 151 )/ 1841 /,In( 152 )/ 1843 /
Data In( 153 )/ 1850 /,In( 154 )/ 1856 /
Data In( 155 )/ 1859 /,In( 156 )/ 1861 /
Data In( 157 )/ 1866 /,In( 158 )/ 1868 /
Data In( 159 )/ 1870 /,In( 160 )/ 1908 /
```

```
      Data In( 161 )/ 1917 /,In( 162 )/ 1925 /
      Data In( 163 )/ 1931 /,In( 164 )/ 1939 /
      Data In( 165 )/ 1944 /,In( 166 )/ 1947 /
      Data In( 167 )/ 1956 /,In( 168 )/ 1958 /
      Data In( 169 )/ 1969 /,In( 170 )/ 1973 /
      Data In( 171 )/ 1975 /,In( 172 )/ 1985 /
      Data In( 173 )/ 1987 /,In( 174 )/ 1990 /
      Data In( 175 )/ 1994 /,In( 176 )/ 1998 /
      Data In( 177 )/ 2013 /,In( 178 )/ 2020 /
      Data In( 179 )/ 2026 /,In( 180 )/ 2028 /
      Data In( 181 )/ 2030 /,In( 182 )/ 2032 /
      Data In( 183 )/ 2035 /,In( 184 )/ 2037 /
      Data In( 185 )/ 2039 /,In( 186 )/ 2041 /
      Data In( 187 )/ 2044 /,In( 188 )/ 2046 /
      Data In( 189 )/ 2048 /,In( 190 )/ 2050 /
      Data In( 191 )/ 2052 /,In( 192 )/ 2054 /
      Data In( 193 )/ 2059 /,In( 194 )/ 2065 /
      Data In( 195 )/ 2067 /,In( 196 )/ 2069 /
      Data In( 197 )/ 2077 /,In( 198 )/ 2079 /
      Data In( 199 )/ 2081 /,In( 200 )/ 2083 /
      Data In( 201 )/ 2085 /,In( 202 )/ 2088 /
      Data In( 203 )/ 2090 /,In( 204 )/ 2094 /
      Data In( 205 )/ 2099 /,In( 206 )/ 2102 /
      Data In( 207 )/ 2107 /,In( 208 )/ 2112 /
C
C

C  ICOMP is the code for a composite character, though I'm not sure what that
C  is.

      Exit_Code = 0
      Icomp=410

C  When we begin drawing a char the pen goes down by default.  When we enter
C  this routine in the middle of a char it comes up.

      If (G_Char_Gen_State .Eq. 0) Then
            Ipen=3
            Iptr=0
      Else
            Ipen = 2
            Goto 16
      Endif

      Isym=Char
5     If ((Isym .Ge. 32) .And. (Isym .Le. 122)) Go To 10
      Isym=Isym-287   ! Has Something To Do With Composites.
C
C OBTAIN INDEXES (BEGIN/END) TO THE CHARACTER TABLE.  This part of the table
C  contains pointers to the character elements which compose the char.
C
```

```
10      Ifirst=In(Isym-32)+1
        Ilast=In(Isym-32+1)
C
C LOOP THRU CHARACTER TABLE
C
15      I = Ifirst
16      If (I .Gt. Ilast) Goto 25
        J=Ic(I)
C
C IS THE CHARACTER TABLE ELEMENT A COMPOUND ?
C
        If (J .Ge. Icomp) Go To 30
C
C IS THE CHARACTER TABLE ELEMENT A PEN UP COMMAND ?
C
        If (J .Eq. 1) Go To 18
C
C OBTAIN THE X,Y BASIC ELEMENT (i.e., graphic element) PAIR FROM THE BASIC
C  ELEMENTS TABLE.
C
        Call Trip(J,Nx,Ny)
        Ym=Float(Ny)
        Xm=Float(Nx)

C  Scale, stretch and rotate the move/draw.

        Ypos = O_Char_Scale/21.0 * ( Ym*G_Cos_Crot + Xm*G_Sin_Crot )
        Xpos = O_Char_Aspect_Ratio*O_Char_Scale/21.0 *
     1       ( Xm*G_Cos_Crot - Ym*G_Sin_Crot )

C  Return the current pen position and increment the table pointer, which is
C  saved in the common area.

        G_Char_Gen_State = Ipen
        I = I + 1
        Goto 45

C  When we get here it means a pen up command was encountered in the table in
C  place of a coordinate pair.  So we note the command and go back to fetch
C  a coordinate pair.

18      Ipen = 3
        I = I + 1
        Goto 16
C
C ANY LOOP VARIABLES ON THE STACK ?  The stack is apparently formed when
C  a character composite is found.  A graphic element can actually be a pointer
C  to a bunch of graphic elements, so we need to stack the previous table
C  bounds ?
C
25      If (Iptr .Gt. 0) Go To 40
```

```
C  If the table's empty we're done.

        G_Char_Gen_State = -1
        Go To 45
C
C DO WE NEED TO SAVE LOOP VARIABLES (on the stack) ?  Onnly do so if we're not
C  at the end of the table.
C
30      If ( I .Eq. Ilast) Go To 35
        Index1=I+1
        Index2=Ilast
C
C PUT THE LOOP VARIABLES ON THE STACK
C
        Call Cgen_Push(Index1,Index2)
35      Isym=J
        Go To 5
C
C RETRIEVE THE LOOP VARIABLES FROM THE STACK
C
40      Call Cgen_Pop(Index1,Index2)
        Ifirst=Index1
        Ilast=Index2
        Go To 15

45      Exit_Code = G_Char_Gen_State

        Return
        End
```

```
      Subroutine Vector_Gen_Asym(X1,Y1,X2,Y2)

C  This routine generates a vector in a raster with the asymetrical method.
C  In this method the number of points generated along the length of the
C  vector is proportional to the longer projection of the vector along the
C  coordinate axes.  Thus given two vectors of equal lengths, one at a 45
C  degree angle to the coordinate system and one at a 90 degree angle, the
C  latter will have more points generated because its projection along the
C  Y (and X) axis is longer than the formers.  For this reason this routine
C  is probably inferior to vector_gen_sym.  However, reason does not rule
C  all (surprise).  I like the vectors generated by this routine better.

C  Given the longer projection the method of rasterization is to step along
C  all the points in the longer projection of the vector and generate the
C  corresponding points on the other axis.  These pairs of points make up
C  the rasterization.

C  Parameters:

C  X1, Y1, X2, Y2 - i [i] = The end-points of the vector to generate.

      Implicit Integer (A-Z)

      Real Dinc, Depv
      Logical *1 Xdep

C  See which projection is longer, making sure we don't have a vector that is
C  a point.

      If (X1 .Eq. X2 .And. Y1 .Eq. Y2) Goto 200

      If (Iabs(X1-X2) .Gt. Iabs(Y1-Y2)) Goto 10

C  Y projection is longer.  Assign step variable bounds.

      Xdep = .True.
      Ind1 = Y1
      Ind2 = Y2
      Dep1 = X1
      Dep2 = X2
      Goto 100

C  X projection is longer.

10    Xdep = .False.
      Ind1 = X1
      Ind2 = X2
      Dep1 = Y1
      Dep2 = Y2

100   Continue
```

```
C  The axis we step along is the Independent variable, the axis along which
C  we generate points is the Dependent variable.  We now compute the step
C  factors along the axes.

        Dinc = Float(Dep2-Depl) / Float(Iabs(Ind2-Indl))
        Iinc = Mysign(Ind2-Indl)
        Depv = Depl

C  Step along the independent axis.

        Do 110 Indv = Indl, Ind2, Iinc

C  Set the current point in the raster.

        If (Xdep) Then
                Call Set_Pix(Iroundx(Depv),Indv)
        Else
                Call Set_Pix(Indv,Iroundx(Depv))
        Endif

C  Go to the next point along the dependent axis.

        Depv = Depv + Dinc

110     Continue
        Goto 999

C  If the vector is a point, set it.

200     Call Set_Pix(Xl,Yl)
        Goto 999

999     Return
        End
```

```
      Subroutine Vector_Gen_Sym(X1,Y1,X2,Y2)

C  This routine generates the rasterization of a vector using the Symmetrical
C  method.  The number of points generated by this method is proportional to
C  the length of the vector, giving all vectors generated a uniform density.
C  This feature should make this algorithm superior to the one used by
C  Vector_Gen_Asym.  But I happen to like the other vectors better.

C  Parameters:

C  X1, Y1, X2, Y2 - i [i] = The end points of the vector to generate.

C  The actual method is to start at one of the end points and move in steps
C  along each coordinate whose lengths are equal to the projection of the
C  vector along that coordinate axis divided by the length of the vector.

      Implicit Integer (A-Z)

      Real Sqrt
      Real Veclen, Xstep, Ystep, X, Y

C  Treat a vector that is a point specially.

      If (X1 .Eq. X2 .And. Y1 .Eq. Y2) Goto 20

C  Compute the vector length and the axis steps.

      Veclen = Sqrt(Float(X1-X2)**2 + Float(Y1-Y2)**2)
      Xstep = (X2 - X1) / Veclen
      Ystep = (Y2 - Y1) / Veclen

      X = X1
      Y = Y1

C  Generate Veclen points.  The line density could be changed by generating
C  some multiple of Veclen points.

      Do 10 N = 1, Ifix(Veclen)

C  Set the current point.

      Call Set_Pix(Iroundx(X),Iroundx(Y))

C  Next point.

      X = X + Xstep
      Y = Y + Ystep

10    Continue

C  Set the last point or process the special case where the vector is a point.
```

```
20      Call Set_Pix(X2,Y2)

        Return
        End
```

```
        Subroutine Wr_Mess(Message)

C  This routine writes a message composed of words to the terminal. The
C  message can be of any length.  We split it into lines of length less than
C  D_Terminal_Line_Length at word boundaries.  This makes it easy for the
C  programmer to code long messages.

C  Parameters:

C  Message - C* [i] = The message to write.

        Implicit Integer (A-Z)

        Character *(*) Message

        Logical *1 In_Word, Found_Start

        Include 'Termunits.Inc'
        Include 'Devices.Inc'

C  The character variable may be longer than the actual message.  Find the
C  actual message length, assuming the message is padded on the right with
C  spaces.

        Do 10 Ndx = Len(Message), 1, -1
10      If (Message(Ndx:Ndx) .Ne. ' ') Goto 20

        Write (T_Out,900)
900     Format (/X,'Wr_Mess-I-Message Variable Is Empty.'/)
        Stop

C  Now break the message into chunks which are shorter than the line length
C  at word boundaries.

20      Mlen = Ndx
        Mstart = 1

C  Write a blank line.

        Write (T_Out,902)
902     Format ()

C  Start at the beginning of the current chunk.  Look for the first non-space.
C  This is the start of the current line.  Then look for word breaks until we
C  have a line longer than the line length.  Then go back to the last word
C  break and use that string.

C  Initially we have not found the start of the line and we are not within a
C  word boundary.

25      Found_Start = .False.
```

```
        In_Word = .False.

        Do 30 Mptr = Mstart, Mlen

        If (Mptr-Mstart+1 .Gt. D_Terminal_Line_Length) Goto 40

C  Are we inside or outside a word ?

        If (Message(Mptr:Mptr) .Eq. ' ') Then
                If (In_Word) Then
                        Word_End = Mptr - 1
                        In_Word = .False.
                Endif
        Else
                In_Word = .True.
                If (.Not. Found_Start) Then
                        Line_Start = Mptr
                        Found_Start = .True.
                Endif
        Endif

30      Continue

        Word_End = Mlen

C  Display the current line.

40      Write (T_Out,901) Message(Line_Start:Word_End)
901     Format (X,A<Word_End-Line_Start+1>)

C  Loop back for the next line if we're not at the end of the buffer.

        Mstart = Word_End + 1
        If (Mstart .Lt. Mlen) Goto 25

C  Write another blank line.

        Write (T_Out,902)

999     Return
        End
```

```
        Subroutine Zero_Band

C  Zeros out the raster used to hold the band.

        Implicit Integer (A-Z)

        Include 'Band.Inc'
        Include 'Devices.Inc'

C  Figure out how many bytes the band uses.

        Bytes_Used = Ceiling(Float(B_Band_Height * B_Band_Width) /
        1                          Float(D_Bits_Per_Byte ))

C  Zero them.

        Do 10 Ndx = 1, Bytes_Used
10      B_Band(Ndx) = 0

        Return
        End
```

```
C  Name Prefix =  B

C  This include file defines the data structure used to hold a band from
C  the raster.  A band is some subset of contiguous rows from the raster.
C  Each point within the raster is stored as one within a band since the
C  printer either lays a dot or doesn't.

C  Variables:

C  B_Band - b = A buffer used to store the band in either row-major or
C               column major form.
C  B_Band_Bytes - b = The amount of space the system generator wishes to
C                     allocate for the band.  If no restrictions are to be
C                     imposed on the picture the user can display, the
C                     minimum value for this parameter is:
C       (13 inches) * (960 heps / inch) * (6 rows/printhead) / (8 bits / byte)
C                                 = 9360
C                     If the buffer is smaller the user won't be able to
C                     print huge pictures at highest resolution.
C  B_Band_Width, B_Band_Height - i = The size of rectangular array of bits
C                                    the band represents.

       Parameter ( B_Band_Bytes = 21170 )

       Byte B_Band(B_Band_Bytes)

       Integer B_Band_Width, B_Band_Height

       Common /Bandco/ B_Band_Width, B_Band_Height, B_Band
```

```
C  This file defines a set of bit masks which select different numbers of
C  the low bits in a word.

      Parameter ( Low1 = 1, Low2 = 3, Low3 = 7, Low4 = 15, Low5 = 31 )
      Parameter ( Low6 = 63, Low7 = 127, Low8 = 255, Low9 = 511 )
      Parameter ( Low10 = 1023, Low11 = 2047, Low12 = 4096 )
      Parameter ( Low13 = 2**13-1 , Low14 = 2**14-1, Low15 = 2**15-1 )
      Parameter ( Low16 = 2**16-1 )

      Parameter ( Low17 = 2**17-1, Low18 = 2**18-1, Low19 = 2**19-1 )
      Parameter ( Low20 = 2**20-1, Low21 = 2**21-1, Low22 = 2**22-1 )
      Parameter ( Low23 = 2**23-1, Low24 = 2**24-1, Low25 = 2**25-1 )
      Parameter ( Low26 = 2**26-1, Low27 = 2**27-1, Low28 = 2**28-1 )
      Parameter ( Low29 = 2**29-1, Low30 = 2**30-1 )
```

```
C  This file defines a set of masks which one can use to select the different
C  bits within a word.

      Parameter ( Bit0 = 1, Bit1 = 2, Bit2 = 4, Bit3 = 8, Bit4 = 16 )
      Parameter ( Bit5 = 32, Bit6 = 64, Bit7 = 128, Bit8 = 256 )
      Parameter ( Bit9 = 512, Bit10 = 1024, Bit11 = 2048, Bit12 = 4096 )
      Parameter ( Bit13 = 2**13, Bit14 = 2**14, Bit15 = 2**15 )
      Parameter ( Bit16 = 2**16 )

      Parameter ( Bit17 = 2**17, Bit18 = 2**18, Bit19 = 2**19 )
      Parameter ( Bit20 = 2**20, Bit21 = 2**21, Bit22 = 2**22 )
      Parameter ( Bit23 = 2**23, Bit24 = 2**24, Bit25 = 2**25 )
      Parameter ( Bit26 = 2**26, Bit27 = 2**27, Bit28 = 2**28 )
      Parameter ( Bit29 = 2**29, Bit30 = 2**30 )
```

```
C  This file defines a stack data structure used by the character generator.
      Common /Chargen/ Iptr,Istack(20), Ifirst, Ilast, I
```

```
C  Name Prefix = D

C  This include file defines a number of physical  parameters relating to the
C  devices used by Plaster: the host computer and the printer itself.

      Parameter ( D_Heps_Per_Inch = 960, D_Veps_Per_Inch = 288 )

C  Print head pin spacing is in veps.

      Parameter ( D_Print_Head_Pins = 6 , D_Print_Head_Pin_Spacing = 4 )
      Parameter
     1  ( D_Print_Head_Size = D_Print_Head_Pins * D_Print_Head_Pin_Spacing)

C  Line length is in characters per line.

      Parameter ( D_Terminal_Line_Length = 80 )

      Parameter ( D_Printer_Unit = 10, D_Raster_Unit = 13 )

C  The maximum output line length is probably a function of the Fortran
C  compiler in use, and may even be selectable in Fortran.  Use the highest
C  number possible.

      Parameter ( D_Max_Output_Line_Length = 400 )

      Parameter ( D_Bits_Per_Byte = 8 )

C  Turn on this parameter if this version of Plaster has Stream I/O
C  capability.

      Parameter ( D_Stream_Io = .True. )

C  The printer's default delimiter for RCL.

      Character *(*) D_Default_Delimiter
      Parameter ( D_Default_Delimiter = '!' )

C  In this common area we keep the Plot10 coordinate system maxima for the X
C  and Y axes, and the current delimiter symbol for RCL.

      Integer D_Plot10_Xmax, D_Plot10_Ymax

      Character *1 D_Delimiter

      Common /Device/ D_Plot10_Xmax, D_Plot10_Ymax, D_Delimiter
```

```
C  Name Prefix = L

C  This include file contains type definitions for the lexican analyzer,
C  and the data structure to hold a command line typed by the user and
C  associated pointers.

C  Token types.

      Parameter ( L_Tt_Keywd = 1, L_Tt_Equal = 2, L_Tt_String = 3 )
      Parameter ( L_Tt_Bool = 4, L_Tt_Int = 5, L_Tt_Real = 6 )
      Parameter ( L_Tt_Comma = 7, L_Tt_Eol = 8 )

C  Character types.

      Parameter ( L_Ct_Letter = 1, L_Ct_Undscr = 2, L_Ct_Syms = 3 )
      Parameter ( L_Ct_Dot = 4, L_Ct_Digit = 5, L_Ct_Comma = 6 )
      Parameter ( L_Ct_Space = 7, L_Ct_Equal = 8, L_Ct_Minus = 9 )
      Parameter ( L_Ct_Illeg = 10, L_Ct_Eol = 11 )

C  Expected token types.

      Parameter ( L_Ex_Keywd = 1, L_Ex_Equal = 2, L_Ex_Option = 3 )
      Parameter ( L_Ex_Comma = 4 )

C  The input line and the current token.

      Character *80 L_Inpline, L_Token *20

C  Pointer to our current place in the input line, the current token length,
C  and the input line length.

      Integer L_Iptr, L_Tlen, L_Inplen

      Common /Lexanc/ L_Tlen, L_Inplen, L_Inpline, L_Token, L_Iptr
```

```
C  Name Prefix = N

C  This common area contains a variable which routine
C  Next_Vec_In_Band_Alternating must maintain between calls.

      Integer N_Nvib_State
      Common /Nvibcm/ N_Nvib_State
```

```
C  Name Prefix = O

C  This include file defines a common area containing variables which keep
C  track of the values of the parameters (or options) to Plaster.  In most
C  cases the names chosen for the variables are the same as the parameter
C  names.  In all cases the names are very similar, so individual descriptions
C  should not be necessary.

      Real O_Width, O_Page_Aspect_Ratio, O_Char_Scale
      Real O_Char_Aspect_Ratio, O_Char_Rotation

      Integer O_Horizontal_Dot_Spacing, O_Vertical_Dot_Spacing
      Integer O_Char_Set, O_Image_Rotation, O_Line_Thickness
      Integer O_Pass_Algorithm

      Logical *1 O_Inch_Mode, O_Print, O_Save_Raster, O_Read_Raster
      Logical *1 O_Row_Major, O_Rcl_Mode, O_Use_Default_Io_System

      Common /Optcom/ O_Width, O_Page_Aspect_Ratio, O_Char_Scale,
     1       O_Char_Aspect_Ratio, O_Char_Rotation, O_Horizontal_Dot_Spacing,
     2       O_Vertical_Dot_Spacing, O_Char_Set, O_Image_Rotation,
     3       O_Line_Thickness, O_Pass_Algorithm, O_Inch_Mode, O_Print,
     4       O_Save_Raster, O_Read_Raster, O_Row_Major, O_Rcl_Mode,
     5       O_Use_Default_Io_System
```

```
C  Name prefix is   P

C  This file contains the P10INTERP common area and parameters associated
C  with the Plot10 interpreting subsystem.

C  Variables:

C  P_BUFFER(MAX_Blen) = A buffer to hold the current line from the input file.
C  P_BLEN = The length of the buffer.
C  P_BPTR = The last character accessed from the buffer.
C  P_XOLD,P_YOLD = The current graphics position on the screen.
C  P_RECCNT = The number of records read from the input stream at a given
C           moment.
C  P_STATE = 1 when the package is initialized, 2 and 3 when it is running.
C               2 when we are to read from the input file, 3 when we are
C               in the middle of drawing a character.
C  P_BUFFER_SPLIT_HACK = The parameter used to detect the string buffer split
C                                sequence.  See discussion in NEXTVEC.FOR .
C  P_XSAVE, P_YSAVE = Used to save the current position as given by the
C                     character generator at the end of a character draw.
C  P_XKEEP, P_YKEEP = Used to maintain what Plot10 thinks is the current
C                     position during a series of character draws.
C  P_CURR_CHAR = The current character in the input stream.
C  P_SEEN_VECT = TRUE when we have encountered a (dark or light) vector
C               in the current file.  Thus, this will be false during the
C               preliminary sequence of nulls and the first clear which Plot-10
C               generates.
C  P_DONE_P10INIT = FALSE when we don't need to call P10INIT.

      Parameter ( P_Max_Blen = 200, P_Input_Unit = 9 )

      Integer P_Blen, P_Bptr, P_Xold, P_Yold, P_Reccnt, P_State
      Integer P_Xkeep, P_Ykeep
      Integer P_Buffer_Split_Hack, P_Xsave, P_Ysave

      Logical *1 P_Seen_Vect, P_Done_P10init

      Byte P_Curr_Char, P_Buffer(P_Max_Blen)

      Common /P10int/ P_Blen, P_Bptr, P_Xold, P_Yold, P_Reccnt,
     1                P_State, P_Buffer_Split_Hack, P_Xsave, P_Ysave,
     2                P_Xkeep, P_Ykeep, P_Curr_Char, P_Seen_Vect,
     3                P_Done_P10init, P_Buffer

C  These parameters are the character types - they divide the ascii character
C  set into a set of token classes.

      Parameter ( P_Gs=1, P_Crus=2, P_Bel=3, P_Siso=4, P_Esc=5, P_Sub=6 )
      Parameter ( P_Enqetbff=7, P_Nonprint=8, P_Printable=9, P_Ctlv=10 )
```

```
C  This file defines a set of constants which one can multiply integers
C  by to perform arithmetic shifts.

      Parameter ( Shift1 = 2, Shift2 = 2**2, Shift3 = 2**3, Shift4 = 2**4 )
      Parameter ( Shift5 = 2**5, Shift6 = 2**6, Shift7 = 2**7 )
      Parameter ( Shift8 = 2**8, Shift9 = 2**9, Shift10 = 2**10 )
      Parameter ( Shift11 = 2**11, Shift12 = 2**12, Shift13 = 2**13 )
      Parameter ( Shift14 = 2**14, Shift15 = 2**15, Shift16 = 2**16 )
```

```
C  Name Prefix = G

C  This common area and parameters are a memory for the character generating
C  system.

C  G_Plot21_To_Plot10 is a scale factor used to convert characters from one
C  graphic system to another.

      Parameter ( Move_Code = 2, Draw_Code = 3, Done_Code = -1 )
      Parameter ( Plot21_To_Plot10 = 15.5 )

C  The state of the charater generator.

      Integer G_Char_Gen_State

C  The sine and cosine of the character rotation angle, and spacing information
C  for the current character.  The Loc variables maintain the current real
C  coordinate, the Prev variables maintain the integer beam position.  We
C  want the latter to be as close as possible to the former.

      Real G_Sin_Crot, G_Cos_Crot, G_Xmin, G_Xmax, G_Xloc, G_Yloc
      Real G_Xprev, G_Yprev

      Common /Symmem/ G_Char_Gen_State, G_Sin_Crot, G_Cos_Crot,
     1       G_Xmin, G_Xmax, G_Xloc, G_Yloc, G_Xprev, G_Yprev
```

```
C  This file defines the logical units for terminal i/o.
      Parameter ( T_In = 5,  T_Out = 6 )
```

```
C  Name Prefix = V
C  This include file defines some parameters used to communicate with the VMS
C  system routines used in the VMS stream I/O system, and maintains the
C  channel number used in common.

      Parameter ( V_Success = 1, V_Io$Writevblk = '30'X )
      Parameter ( V_No_Carriage_Control = 0 )
      Parameter ( V_Event_Flag1 = 1, V_Event_Flag2 = 2 )

      Integer V_Channel

      Common /Siocom/ V_Channel
```

```
C  Name prefix is   P

C  This file contains the P10INTERP common area and parameters associated
C  with the Plot10 interpreting subsystem.

C  Variables:

C  P_BUFFER(MAX_Blen) = A buffer to hold the current line from the input file.
C  P_BLEN = The length of the buffer.
C  P_BPTR = The last character accessed from the buffer.
C  P_XOLD,P_YOLD = The current graphics position on the screen.
C  P_RECCNT = The number of records read from the input stream at a given
C           moment.
C  P_STATE = 1 when the package is initialized, 2 and 3 when it is running.
C               2 when we are to read from the input file, 3 when we are
C               in the middle of drawing a character.
C  P_BUFFER_SPLIT_HACK = The parameter used to detect the string buffer split
C                        sequence.  See discussion in NEXTVEC.FOR .
C  P_XSAVE, P_YSAVE = Used to save the current position as given by the
C                     character generator at the end of a character draw.
C  P_XKEEP, P_YKEEP = Used to maintain what Plot10 thinks is the current
C                     position during a series of character draws.
C  P_CURR_CHAR = The current character in the input stream.
C  P_SEEN_VECT = TRUE when we have encountered a (dark or light) vector
C               in the current file.  Thus, this will be false during the
C               preliminary sequence of nulls and the first clear which Plot-10
C               generates.
C  P_DONE_P10INIT = FALSE when we don't need to call P10INIT.

      Parameter ( P_Max_Blen = 200, P_Input_Unit = 9 )

      Integer P_Blen, P_Bptr, P_Xold, P_Yold, P_Reccnt, P_State
      Integer P_Xkeep, P_Ykeep
      Integer P_Buffer_Split_Hack, P_Xsave, P_Ysave

      Logical *1 P_Seen_Vect, P_Done_P10init

      Byte P_Curr_Char, P_Buffer(P_Max_Blen)

      Common /P10int/ P_Blen, P_Bptr, P_Xold, P_Yold, P_Reccnt,
     1                P_State, P_Buffer_Split_Hack, P_Xsave, P_Ysave,
     2                P_Xkeep, P_Ykeep, P_Curr_Char, P_Seen_Vect,
     3                P_Done_P10init, P_Buffer

C  These parameters are the character types - they divide the ascii character
C  set into a set of token classes.

      Parameter ( P_Gs=1, P_Crus=2, P_Bel=3, P_Siso=4, P_Esc=5, P_Sub=6 )
      Parameter ( P_Enqetbff=7, P_Nonprint=8, P_Printable=9, P_Ctlv=10 )
```

```
C  This file defines a set of constants which one can multiply integers
C  by to perform arithmetic shifts.

      Parameter ( Shift1 = 2, Shift2 = 2**2, Shift3 = 2**3, Shift4 = 2**4 )
      Parameter ( Shift5 = 2**5, Shift6 = 2**6, Shift7 = 2**7 )
      Parameter ( Shift8 = 2**8; Shift9 = 2**9, Shift10 = 2**10 )
      Parameter ( Shift11 = 2**11, Shift12 = 2**12, Shift13 = 2**13 )
      Parameter ( Shift14 = 2**14, Shift15 = 2**15, Shift16 = 2**16 )
```

```
C  Name Prefix = G

C  This common area and parameters are a memory for the character generating
C  system.

C  G_Plot21_To_Plot10 is a scale factor used to convert characters from one
C  graphic system to another.

      Parameter ( Move_Code = 2, Draw_Code = 3, Done_Code = -1 )
      Parameter ( Plot21_To_Plot10 = 15.5 )

C  The state of the charater generator.

      Integer G_Char_Gen_State

C  The sine and cosine of the character rotation angle, and spacing information
C  for the current character.  The Loc variables maintain the current real
C  coordinate, the Prev variables maintain the integer beam position.  We
C  want the latter to be as close as possible to the former.

      Real G_Sin_Crot, G_Cos_Crot, G_Xmin, G_Xmax, G_Xloc, G_Yloc
      Real G_Xprev, G_Yprev

      Common /Symmem/ G_Char_Gen_State, G_Sin_Crot, G_Cos_Crot,
     1       G_Xmin, G_Xmax, G_Xloc, G_Yloc, G_Xprev, G_Yprev
```

```
C  This file defines the logical units for terminal i/o.
      Parameter ( T_In = 5,  T_Out = 6 )
```

```
C  Name Prefix = V
C  This include file defines some parameters used to communicate with the VMS
C  system routines used in the VMS stream I/O system, and maintains the
C  channel number used in common.

      Parameter ( V_Success = 1, V_Io$Writevblk = '30'X )
      Parameter ( V_No_Carriage_Control = 0 )
      Parameter ( V_Event_Flag1 = 1, V_Event_Flag2 = 2 )

      Integer V_Channel

      Common /Siocom/ V_Channel
```