

```

1  !-----
2  !CFont.f95 version 1.3 2021.09.05.1155cdt JMS
3  !-----
4  !2021.09.05.1155cdt JMS -CFont.f95 - current build environment:
5  !                               -Traveler2/Athlon64/Wi nXPPro/APF9.0/Ogl1.2.1
6  !2004.10.02.0920cst JMS -Modified to Fortran 95... Version 1.3
7  !CFont12.F90
8  !2004.02.24.0920cst JMS -CFont.F90: "Contrast Font" Demonstration/Generation.
9  !                               Version 1.2
10 !                               New name. Supercedes "Nixie vsn 1.111"
11 !                               Uses RMB / Menu for control.
12 !2004.02.18.1200cst JMS -Nixie111.F90: Standardized time calls. (Mitchell)
13 !2004.02.17.1310cst JMS -Nixie11.F90 : First post on www.setterholm.com
14 !                               -A700/Nt4.0: sp6/CVF6.6b/Ogl1.2.1/Glut3.7.6
15 !-----
16 ! " ! " remarks. The rest of the line is ignored by the compiler.-----7 9
17 ! " ; " separates statements on the same line.
18 ! " & " line continuation to the next line.
19 ! " z' " is a prefix for hexadecimal numbers.-----7 9
20 !-----
21 !-----
22 Module CFontSup
23 ! The elements of this module are tied to an OpenGL screen interface.
24 use opengl_gl
25 use opengl_glu
26 use opengl_glut
27 !use opengl
28 !use CFontSup !^CFontSup-CFontUse, KeybdUse
29
30 implicit none
31 public :: CFontUse & ! Font writing interface.
32           , MenuInit & !
33           , KeybdUse ! (key, x, y) Keyboard interface
34 integer*4, public :: KeyChar
35 integer*4, public :: iZoom
36 contains
37 !-----7 9
38 Subroutine ScrnUse !"Callback" called by OpenGL to update the screen.
39 implicit none
40 character :: cLabel*80
41 integer*4 :: i, j, n
42 !Next screen iteration - double buffer mode (drawing to the back screen).
43 call glClearColor(.0, .3, .2, 0.)
44 call glClear(ior(GL_COLOR_BUFFER_BIT, GL_DEPTH_BUFFER_BIT))
45 call glEnable(GL_DEPTH_TEST)
46 call glMatrixMode(GL_PROJECTION) ; call glLoadIdentity
47 ! The 640 x 480 pixel window becomes an isometric volume with dimensions...
48 call glortho( -1.333_gl double, 1.333_gl double & ! Left-to-right
49              , -1.000_gl double, 1.000_gl double & ! bottom-to-top
50              , 1.000_gl double, -1.000_gl double ) ! near-to-far (ref.)
51 !seems: far-to-near
52 call glMatrixMode(GL_MODELVIEW) ; call glLoadIdentity
53
54 call glPushMatrix !-----This exercises an existing GLUT bitmap font.
55 call Colors(12); cLabel= "Close other applications before use."&
56 // " Use this program at your own risk. "
57 do 10 i=1, 80
58   n=iChar(cLabel(i:i)); if (n.eq.0) goto 11
59   call glRasterPos3D(-1.32d0+(i-1)*(9*2.666)/640. , .93d0 , .0d0)
60 10 call glBitmapcharacter(GLUT_BITMAP_9_BY_15, n)
61 11 call glFlush
62 call glPopMatrix !-----End of use of an existing GLUT bitmap font.
63
64 call CFontUse !This demonstrates the Contrast Font on the screen.
65
66 do 30 j=1, 2
67   call glPushMatrix !-----This exercises an existing GLUT bitmap font.
68   call Colors(15)
69   cLabel=" ^ These are line numbers in subroutine CFontUse."
70   if(j.eq.2) cLabel= &

```

```

71     "Documentation: http://ftp.setterholm.com/Fortran/CFont *.pdf"
72     do 20 i=1,80
73         n=iChar(cLabel(i:i)); if (n.eq.0) goto 21
74         call glRasterPos3D(-1.33d0+(i-1)*(9*2.666d0)/640.d0,      &
75                          -.75d0-j*.10d0,                      ,.0d0)
76 20     call glutBitmapCharacter(GLUT_BITMAP_9_BY_15,n)
77 21     call glFlush
78     call glPopMatrix      !-----End of use of an existing GLUT bitmap font.
79 30 continue
80
81     call glutSwapBuffers !Scene drawing done. Swap back and front buffers.
82     call glutPostRedisplay
83     return              !Return to OpenGL
84 End Subroutine ScrnUse
85 !-----7 9
86 Subroutine CFontUse !Exercises the Contrast Font (nee: "Nixie" inaccurately).
87 implicit none      ! in multiple calls to subroutine AlphaCF.
88 character::Chn*40,C0*1      !,C8*8,C9*9
89 real*8      ::x(3)
90 integer*4::iColorC      !i,,n
91 real*4      ::OneZero = 1.0
92 real*4      ::OneZIncr= -.001
93 integer*4::iClock(8)
94 character::cClock(3)*12
95
96     c0=char(0)
97     if(OneZero.lt..001) OneZIncr= .001
98     if(OneZero.gt..999) OneZIncr=-.001
99     OneZero=OneZero+OneZIncr
100 !Line#1
101 write(Chn,"(' 1 CFONT.EXE VSN 1.3 2021.09.05 JMS',a1)") c0
102 x(1)=-1.3d0 ;x(2)= .8d0 ;x(3)=.0d0 ;call AlphaCF(Chn,x,15,0,-1)
103 !Line#2 (Character color changes over time)
104 iColorC=int(OneZero*15.9)
105 write(Chn,"(' 2 CONTRAST FONT GENERATION/DEMO. ',i2,a1)") iColorC,c0
106 x(1)=-1.3d0 ;x(2)= .6d0 ;x(3)= .0d0 ;call AlphaCF(Chn,x,iColorC,0,6)
107 !Line#3
108 Chn=" 3 ABCDEFGHIJKLMNOPQRSTUVWXYZ"//c0
109 x(1)=-1.3d0 ;x(2)= .4d0 ;x(3)= .0d0 ;call AlphaCF(Chn,x,0,-1,-1)
110 !Line#4
111 Chn=' 4 0123456789abcdef "' // "&()*+,-./:<=>~" // char(176)//c0
112 x(1)=-1.3d0 ;x(2)= .3d0 ;x(3)= .0d0 ;call AlphaCF(Chn,x,0,-1,-1)
113 !Line#5
114 Chn=" 5 ABCDEFGHIJKLMNOPQRSTUVWXYZ"//c0
115 x(1)=-1.3d0 ;x(2)= .2d0 ;x(3)= .0d0 ;call AlphaCF(Chn,x,-1,11,-1)
116 !Line#6
117 Chn=' 6 0123456789abcdef "' // "&()*+,-./:<=>~" // char(176)//c0
118 x(1)=-1.3d0 ;x(2)= .1d0 ;x(3)= .0d0 ;call AlphaCF(Chn,x,-1,11,-1)
119 !Line#7
120 Chn=" 7 ABCDEFGHIJKLMNOPQRSTUVWXYZ"//c0
121 x(1)=-1.3d0 ;x(2)= .0d0 ;x(3)= .0d0 ;call AlphaCF(Chn,x,0,11,-1)
122 !Line#8
123 Chn=' 8 0123456789abcdef "' // "&()*+,-./:<=>~" // char(176)//c0
124 x(1)=-1.3d0 ;x(2)=-.1d0 ;x(3)= .0d0 ;call AlphaCF(Chn,x,0,11,-1)
125
126 !Date/Time:
127 call Date_And_Time(cClock(1),cClock(2),cClock(3),iClock) !Date and time
128 !Line#9
129 write(chn,"(' 9 ',i4.4,'.',i2.2,'.',i2.2 ,a1 )") &
130 iClock(1),iClock(2),iClock(3),c0
131 x(1)=-1.3d0 ;x(2)=-.3d0 ;x(3)= .0d0 ;call AlphaCF(Chn,x,12,0,0)
132 !Line#10
133 write(chn,"(' 10 ',5x,i2.2,':',i2.2,':',i2.2,':',i2.2 ,a1 )") &
134 iClock(5),iClock(6),iClock(7),iClock(8)/10,c0
135 x(1)=-1.3d0 ;x(2)=-.4d0 ;x(3)= .0d0 ;call AlphaCF(Chn,x,12,0,-1)
136
137 !Line#11
138 Chn=' 11 ' //char(KeyChar)//c0
139 x(1)=-1.3d0 ;x(2)=-.5d0 ;x(3)= .0d0 ;call AlphaCF(Chn,x,15,0,9)
140 Chn=' TYPE Z TO ZOOM ->' //char(0) ;call AlphaCF(Chn,x,10,-1,-1)

```

```

141 !Line#12 (Glides up and down.)
142           Chn="12 RIGHT MOUSE BUTTON - MENU (FAR) "//c0
143   if(OneZIncr.lt.0.) Chn="12 RIGHT MOUSE BUTTON - MENU (NEAR) "//c0
144   x(1)=-1.3d0; x(2)=(OneZero-.5263d0)*1.9d0; x(3)=.0d0+OneZIncr*20.d0
145           call AlphaCF(Chn, x, -1, 0, -1)
146 !Line#13 (Glides up and down. Stops before the bottom of the screen.)
147           Chn="13 RIGHT MOUSE BUTTON - MENU (FAR) "//c0
148   if(OneZIncr.lt.0.) Chn="13 RIGHT MOUSE BUTTON - MENU (NEAR) "//c0
149   if(x(2).lt.-.7d0) x(2)=-.7d0; x(3)=x(3)-.01; call AlphaCF(Chn, x, 15, -1, -1)
150 !Zoomed View (This registers as intended only in a 640 x 480 pixel window.)
151           call glRasterPos3D(.19d0, -.6d0, -.1d0)
152           call glPixelZoom(4., 4.)
153   if(iZoom.gt.0) call glCopyPixels( 6, 119, 66, 23, GL_COLOR)
154           call colors(0)
155           call glRectD(.18d0, -.61d0, 1.305d0, -.205d0)
156   return
157 End Subroutine CFontUse
158 !-----7 9
159 Subroutine MenuInit
160 integer (kind=glcint) :: menuid
161   menuid = glutCreateMenu(MenuUse)
162   call glutAddMenuEntry("Zoom (toggle on:off)", 1, glcint)
163   call glutAddMenuEntry("CFontIn.Txt -> CFontOut.Txt", 2, glcint)
164   call glutAddMenuEntry(" ***Quit*** ", 3, glcint)
165   call glutAttachMenu(glut_right_button) !This identifies the menu activator.
166 End Subroutine MenuInit
167 !-----7 9
168 Subroutine MenuUse(Value)
169   implicit none
170   integer(kind=glcint), intent(inout) :: Value
171
172   select case(Value) !These are menu-based controls.
173     case( 1_glcint); iZoom=1-iZoom !Displays zoom view of character.
174     case( 2_glcint); call CFontGen ! Reads: "CFontIn.Txt",
175     ! produces: "CFontOut.Txt", and stops.
176     case( 3_glcint); stop 'End of run (CFont:MenuUse)!'
177   end select !Value
178   return
179 end subroutine MenuUse
180 !-----7 9
181 Subroutine KeybdUse(key, x, y) !"Callback" called by OpenGL after a key pressed.
182   implicit none
183   character(kind=glubyte), intent(inout):: key
184   integer(kind=glcint), intent(inout):: x
185   integer(kind=glcint), intent(inout):: y
186   KeyChar=iChar(key) !Processes a single character.
187   select case(Key) !These are keyboard-based controls.
188     case('q'); stop 'End of run (CFont:KeybdUse: key="q" )'
189     case('Z', 'z'); iZoom=1-iZoom !Toggles zoom view
190   end select !Key
191   return !Return to OpenGL
192 End Subroutine KeyBdUse
193 !-----7 9
194 Subroutine Colors(nCol) !These are approximately the 16 DOS colors.
195   implicit none
196   integer*4 :: IndCol(3, 0:15), i, n, nCol
197   real*4 :: Red, Green, Blue
198
199 ! 0 gxBLACK 1 gxBLUE 2 gxGREEN 3 gxCYAN
200 ! 4 gxRED 5 gxMAGENTA 6 gxBROWN 7 gxGRAY
201 ! 8 gxDARKGRAY 9 gxLIGHTBLUE 10 gxLIGHTGREEN 11 gxLIGHTCYAN
202 ! 12 gxLIGHTRED 13 gxLIGHTMAGENTA 14 gxYELLOW 15 gxWHITE
203 data ((IndCol(i, n), i=1, 3), n=0, 15) /
204 0, 0, 0, 0, 0, 170, 0, 0, 170, 170, 170, 170 &
205 , 170, 0, 0, 170, 0, 170, 170, 85, 0, 170, 170, 170 &
206 , 85, 85, 85, 85, 85, 255, 85, 255, 85, 85, 255, 255 &
207 , 255, 85, 85, 255, 85, 255, 255, 255, 255, 85, 255, 255, 255 /
208
209   if((nCol.lt.0).or.(nCol.gt.15)) stop 'Colors: color# outside the range [0, 15]'
210   Red =IndCol(1, nCol)/255.

```

```

211   Green=IndCol (2, nCol) /255.
212   Blue =IndCol (3, nCol) /255.
213   call glColor3f(Red, Green, Blue)
214   return
215 End Subroutine Colors
216 !-----7 9
217 Subroutine AlphaCF(Label L, x, i ColorC, i ColorV, i ColorB)
218 !2004. 02. 17. 1310cst JMS -A700/Nt4. 0: sp6/CVF6. 6b/Ogl1. 2. 1/Glut3. 7. 6
219 ! Jeff Setterholm
220 !CFont-tube font. Inspired by Genus Graphics font: "LED30"
221
222 !Subroutine "AlphaCF": Copyright 2004 by Jeffrey M Setterholm
223 !This subroutine may be incorporated or adapted into your software
224 ! without permission and without license fees.
225
226 ! In this implementation:
227 ! The "&" (38d) symbol produces full-on. (perhaps: 255d)
228
229 ! Example calls:
230 ! character::Chn*40
231 ! real*8 ::x(3)
232 ! Chn="ABCDEFGH IJKLMNOPQRSTUVWXYZ"
233 ! x(1)=-1. d0 ;x(2)=. 1d0 ;x(3)=. 9d0 ;call AlphaCF(Chn, x, 15, 8, 0)
234 ! Chn=' 0123456789abcdef ' // '&'()*+,-./:=-~" // char(176) // char(0)
235 ! x(1)=-1. d0 ;x(2)=. 0d0 ;x(3)=. 9d0 ;call AlphaCF(Chn, x, 15, -1, -1)
236
237 implicit none
238 character::LabelL*40 !Arguments
239 real*8 ::x(3)
240 integer*4::i ColorC, i ColorV, i ColorB
241
242 integer*4::i, Init, i Pass, j, n !Internals
243 integer*4:: i Ascii (0: 255)
244 character::cAscii C(0: 255)*1
245 ! character::cAscii V(0: 255)*1
246
247 integer*4::nCharTot=60 !These arrays are dimensioned for 60 characters total.
248 integer*2 ::i FLineC(21, 60) !Characters
249 integer(kind=gl ubyte) :: FontC( 42 , 60)
250 integer(kind=gl ubyte), equivalence( i FLineC , FontC )
251 integer(kind=gl ubyte) , target:: FontCT( 42 , 60)
252 integer*2 ::i FLineV(21, 60) !Voids
253 integer(kind=gl ubyte) :: FontV( 42 , 60)
254 integer(kind=gl ubyte), equivalence( i FLineV, FontV )
255 integer(kind=gl ubyte) , target:: FontVT( 42 , 60)
256 integer(kind=gl ubyte) , pointer:: FontP(:) !image pointer
257
258 character:: LabL40*40, LabL1(40)*1 !The input label is 40 characters max.,
259 equivalence(LabL40 , LabL1 ) !including the null (zero) termination.
260 real (kind=GLfloat)::Xgl , Ygl , Zgl
261
262 ! " " , " " , "&" , " " , " (" , " )" , "*" , "+" , " " , "-" (&=full on)
263 ! , " " , "/" , "0" , "1" , "2" , "3" , "4" , "5" , "6" , "7" These are the characters below.
264 ! , "8" , "9" , " " , "=" , "A" , "B" , "C" , "D" , "E" , "F"
265 ! , "G" , "H" , "I" , "J" , "K" , "L" , "M" , "N" , "O" , "P"
266 ! , "Q" , "R" , "S" , "T" , "U" , "V" , "W" , "X" , "Y" , "Z"
267 ! , "a" , "b" , "c" , "d" , "e" , "f" , "~" , "o"
268
269 !This is the CFontOut.Txt output:
270 Data(cAscii C(j), (i FLineC(i, j), i=1, 21), j= 1, 30) / & !Characters.
271 " " , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' &
272 , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' &
273 , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' &
274 , " " , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' &
275 , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 2002' , z' 7007' , z' 7007' &
276 , z' 7007' , z' 7007' , z' 7007' , z' 7007' , z' 7007' , z' 2002' , z' 0000' &
277 , "&" , z' FFFF' , z' FFFF' , z' FFFF' , z' FFFF' , z' FFFF' , z' FFFF' , z' FFFF' & !(full-on)
278 , z' FFFF' , z' FFFF' , z' FFFF' , z' FFFF' , z' FFFF' , z' FFFF' , z' FFFF' &
279 , z' FFFF' , z' FFFF' , z' FFFF' , z' FFFF' , z' FFFF' , z' FFFF' , z' FFFF' &
280 , " " , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' , z' 0000' &

```

281 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 8000',z' C001',z' C001' &
 282 ,z' C001',z' C001',z' C001',z' C001',z' C001',z' 8000',z' 0000' &
 283 ,(" ,z' 0000',z' 3800',z' 7000',z' A000',z' C001',z' C001',z' C001' &
 284 ,z' C001',z' C001',z' 8001',z' 0000',z' 8001',z' C001',z' C001' &
 285 ,z' C001',z' C001',z' C001',z' A000',z' 7000',z' 3800',z' 0000' &
 286 ,") " ,z' 0000',z' 000E',z' 0007',z' 8002',z' C001',z' C001',z' C001' &
 287 ,z' C001',z' C001',z' C000',z' 0000',z' C000',z' C001',z' C001' &
 288 ,z' C001',z' C001',z' C001',z' 8002',z' 0007',z' 000E',z' 0000' &
 289 ,"*" ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 3006',z' 7007' &
 290 ,z' 6003',z' 4001',z' 380E',z' 7C1F',z' 380E',z' 4001',z' 6003' &
 291 ,z' 7007',z' 3006',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
 292 ,"+ " ,z' 0000',z' 0000',z' 0000',z' 0000',z' 8000',z' C001',z' C001' &
 293 ,z' C001',z' C001',z' BC1E',z' 7E3F',z' BC1E',z' C001',z' C001' &
 294 ,z' C001',z' C001',z' 8000',z' 0000',z' 0000',z' 0000',z' 0000' &
 295 ,", " ,z' 0000',z' 8003',z' C001',z' E000',z' 7000',z' 0000',z' 0000' &
 296 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
 297 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
 298 ,"- " ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
 299 ,z' 0000',z' 0000',z' F007',z' F80F',z' F007',z' 0000',z' 0000' &
 300 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
 301 ,". " ,z' 0000',z' C001',z' E003',z' C001',z' 0000',z' 0000',z' 0000' &
 302 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
 303 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
 304 ,"/ " ,z' 0000',z' 0000',z' 0000',z' 0020',z' 0030',z' 0038',z' 001C' &
 305 ,z' 000E',z' 0007',z' 0003',z' 4001',z' 6000',z' 7000',z' 3800' &
 306 ,z' 1C00',z' 0E00',z' 0600',z' 0200',z' 0000',z' 0000',z' 0000' &
 307 ,"0" ,z' 0000',z' F007',z' F80F',z' F417',z' 0E38',z' 0E38',z' 0E38' &
 308 ,z' 0E38',z' 0E38',z' 0630',z' 0000',z' 0630',z' 0E38',z' 0E38' &
 309 ,z' 0E38',z' 0E38',z' 0E38',z' F417',z' F80F',z' F007',z' 0000' &
 310 ,"1" ,z' 0000',z' 780F',z' 7007',z' A002',z' C001',z' C001',z' C001' &
 311 ,z' C001',z' C001',z' 8000',z' 0000',z' 8000',z' C001',z' C001' &
 312 ,z' C001',z' C001',z' C001',z' C002',z' 4007',z' 800F',z' 0000' &
 313 ,"2" ,z' 0000',z' FE1F',z' FC2F',z' F837',z' 0038',z' 0038',z' 0038' &
 314 ,z' 0038',z' 0038',z' F017',z' F80F',z' F407',z' 0E00',z' 0E00' &
 315 ,z' 0E00',z' 0E00',z' 0E00',z' F407',z' F80F',z' F007',z' 0000' &
 316 ,"3" ,z' 0000',z' F03F',z' F81F',z' F40F',z' 0E00',z' 0E00',z' 0E00' &
 317 ,z' 0E00',z' 0E00',z' F407',z' F80F',z' F407',z' 0E00',z' 0E00' &
 318 ,z' 0E00',z' 0E00',z' 0E00',z' F40F',z' F81F',z' F03F',z' 0000' &
 319 ,"4" ,z' 0000',z' 0200',z' 0600',z' 0E00',z' 0E00',z' 0E00',z' 0E00' &
 320 ,z' 0E00',z' 0E00',z' F61F',z' FA2F',z' F637',z' 0E38',z' 0E38' &
 321 ,z' 0E38',z' 0E38',z' 0E38',z' 0E38',z' 0630',z' 0220',z' 0000' &
 322 ,"5" ,z' 0000',z' F007',z' F80F',z' F407',z' 0E00',z' 0E00',z' 0E00' &
 323 ,z' 0E00',z' 0E00',z' F407',z' F80F',z' F037',z' 0038',z' 0038' &
 324 ,z' 0038',z' 0038',z' 0038',z' F037',z' F82F',z' FC1F',z' 0000' &
 325 ,"6" ,z' 0000',z' F007',z' F80F',z' F417',z' 0E38',z' 0E38',z' 0E38' &
 326 ,z' 0E38',z' 0E38',z' F437',z' F80F',z' F037',z' 0038',z' 0038' &
 327 ,z' 0038',z' 0038',z' 0038',z' F017',z' F80F',z' F807',z' 0000' &
 328 ,"7" ,z' 0000',z' 0200',z' 0600',z' 0E00',z' 0E00',z' 0E00',z' 0E00' &
 329 ,z' 0E00',z' 0E00',z' 0600',z' 0000',z' 0600',z' 0E00',z' 0E00' &
 330 ,z' 0E00',z' 0E00',z' 0E00',z' F60F',z' FA1F',z' FC3F',z' 0000' &
 331 ,"8" ,z' 0000',z' F007',z' F80F',z' F417',z' 0E38',z' 0E38',z' 0E38' &
 332 ,z' 0E38',z' 0E38',z' F417',z' F80F',z' F417',z' 0E38',z' 0E38' &
 333 ,z' 0E38',z' 0E38',z' 0E38',z' F417',z' F80F',z' F007',z' 0000' &
 334 ,"9" ,z' 0000',z' F03F',z' F81F',z' F40F',z' 0E00',z' 0E00',z' 0E00' &
 335 ,z' 0E00',z' 0E00',z' F607',z' F80F',z' F617',z' 0E38',z' 0E38' &
 336 ,z' 0E38',z' 0E38',z' 0E38',z' F417',z' F80F',z' F007',z' 0000' &
 337 ,": " ,z' 0000',z' 0000',z' 0000',z' 0000',z' 8000',z' C001',z' C001' &
 338 ,z' C001',z' 8000',z' 0000',z' 0000',z' 0000',z' 8000',z' C001' &
 339 ,z' C001',z' C001',z' 8000',z' 0000',z' 0000',z' 0000',z' 0000' &
 340 ,"<" ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' E001',z' C003' &
 341 ,z' 8007',z' 000F',z' 001E',z' 0000',z' 001E',z' 000F',z' 8007' &
 342 ,z' C003',z' E001',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
 343 ,"=" ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' F007' &
 344 ,z' F80F',z' F007',z' 0000',z' 0000',z' 0000',z' F007',z' F80F' &
 345 ,z' F007',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
 346 ,">" ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' C003',z' E001' &
 347 ,z' F000',z' 7800',z' 3C00',z' 0000',z' 3C00',z' 7800',z' F000' &
 348 ,z' E001',z' C003',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
 349 ,"A" ,z' 0000',z' 0220',z' 0630',z' 0E38',z' 0E38',z' 0E38',z' 0E38' &
 350 ,z' 0E38',z' 0E38',z' F637',z' F80F',z' F637',z' 0E38',z' 0E38' &

```

351 ,z' 0E38',z' 0E38',z' 0E38',z' F417',z' F80F',z' F007',z' 0000' &
352 , "B",z' 0000',z' F01F',z' F82F',z' F437',z' 0E38',z' 0E38',z' 0E38' &
353 ,z' 0E38',z' 0E38',z' F437',z' F80F',z' F437',z' 0E38',z' 0E38' &
354 ,z' 0E38',z' 0E38',z' 0E38',z' F437',z' F82F',z' F01F',z' 0000' &
355 , "C",z' 0000',z' F807',z' FCOF',z' FE17',z' 0038',z' 0038',z' 0038' &
356 ,z' 0038',z' 0038',z' 0030',z' 0000',z' 0030',z' 0038',z' 0038' &
357 ,z' 0038',z' 0038',z' 0038',z' FE17',z' FCOF',z' F807',z' 0000' &
358 , "D",z' 0000',z' F01F',z' F82F',z' F437',z' 0E38',z' 0E38',z' 0E38' &
359 ,z' 0E38',z' 0E38',z' 0630',z' 0000',z' 0630',z' 0E38',z' 0E38' &
360 ,z' 0E38',z' 0E38',z' 0E38',z' F437',z' F82F',z' F01F',z' 0000' /
361 Data(cAsciiC(j), (iFLineC(i,j), i=1, 21), j=31, 60) / & !Characters.
362 , "E",z' 0000',z' FE1F',z' FC2F',z' F837',z' 0038',z' 0038',z' 0038' &
363 ,z' 0038',z' 0038',z' F037',z' F80F',z' F037',z' 0038',z' 0038' &
364 ,z' 0038',z' 0038',z' 0038',z' F837',z' FC2F',z' FE1F',z' 0000' &
365 , "F",z' 0000',z' 0020',z' 0030',z' 0038',z' 0038',z' 0038',z' 0038' &
366 ,z' 0038',z' 0038',z' F037',z' F80F',z' F037',z' 0038',z' 0038' &
367 ,z' 0038',z' 0038',z' 0038',z' F837',z' FC2F',z' FE1F',z' 0000' &
368 , "G",z' 0000',z' F007',z' F80F',z' F417',z' 0E38',z' 0E38',z' 0E38' &
369 ,z' 0E38',z' 0E38',z' 3630',z' 7A00',z' 3C30',z' 0038',z' 0038' &
370 ,z' 0038',z' 0038',z' 0038',z' 0038',z' FC17',z' F80F',z' F007',z' 0000' &
371 , "H",z' 0000',z' 0220',z' 0630',z' 0E38',z' 0E38',z' 0E38',z' 0E38' &
372 ,z' 0E38',z' 0E38',z' F637',z' F80F',z' F637',z' 0E38',z' 0E38' &
373 ,z' 0E38',z' 0E38',z' 0E38',z' 0E38',z' 0630',z' 0220',z' 0000' &
374 , "I",z' 0000',z' 780F',z' 7007',z' A002',z' C001',z' C001',z' C001' &
375 ,z' C001',z' C001',z' 8000',z' 0000',z' 8000',z' C001',z' C001' &
376 ,z' C001',z' C001',z' C001',z' A002',z' 7007',z' 780F',z' 0000' &
377 , "J",z' 0000',z' F007',z' F80F',z' F417',z' 0E38',z' 0E38',z' 0E30' &
378 ,z' 0E20',z' 0E00',z' 0600',z' 0000',z' 0600',z' 0E00',z' 0E00' &
379 ,z' 0E00',z' 0E00',z' 0E00',z' 0E00',z' 0600',z' 0200',z' 0000' &
380 , "K",z' 0000',z' 0220',z' 0630',z' 0E38',z' 1E38',z' 3C38',z' 7838' &
381 ,z' F038',z' E031',z' C023',z' 001C',z' 001E',z' 002F',z' 8037' &
382 ,z' C03B',z' E039',z' F038',z' 7838',z' 3C30',z' 1E20',z' 0000' &
383 , "L",z' 0000',z' FE1F',z' FC2F',z' F837',z' 0038',z' 0038',z' 0038' &
384 ,z' 0038',z' 0038',z' 0030',z' 0000',z' 0030',z' 0038',z' 0038' &
385 ,z' 0038',z' 0038',z' 0038',z' 0030',z' 0030',z' 0020',z' 0000' &
386 , "M",z' 0000',z' 0220',z' 0630',z' 8E38',z' CE39',z' CE39',z' CE39' &
387 ,z' CE39',z' CE39',z' 8630',z' 0000',z' 8630',z' CE39',z' CE39' &
388 ,z' CE39',z' CE39',z' CE39',z' D635',z' BA2E',z' 7C1F',z' 0000' &
389 , "N",z' 0000',z' 1E20',z' 3C30',z' 7A38',z' F638',z' 6E38',z' 8E38' &
390 ,z' CE39',z' CE39',z' 8630',z' 0000',z' 8630',z' CE39',z' CE39' &
391 ,z' 8E38',z' 0E3B',z' 8E37',z' 0E2F',z' 061E',z' 023C',z' 0000' &
392 , "O",z' 0000',z' FC1F',z' FA2F',z' F637',z' 0E38',z' 0E38',z' 0E38' &
393 ,z' 0E38',z' 0E38',z' 0630',z' 0000',z' 0630',z' 0E38',z' 0E38' &
394 ,z' 0E38',z' 0E38',z' 0E38',z' F637',z' FA2F',z' FC1F',z' 0000' &
395 , "P",z' 0000',z' 0020',z' 0030',z' 0038',z' 0038',z' 0038',z' 0038' &
396 ,z' 0038',z' 0038',z' F037',z' F80F',z' F437',z' 0E38',z' 0E38' &
397 ,z' 0E38',z' 0E38',z' 0E38',z' F437',z' F82F',z' F01F',z' 0000' &
398 , "Q",z' 0000',z' E61F',z' EE2F',z' DC37',z' 3838',z' 7638',z' 6E38' &
399 ,z' 0E38',z' 0E38',z' 0630',z' 0000',z' 0630',z' 0E38',z' 0E38' &
400 ,z' 0E38',z' 0E38',z' 0E38',z' F637',z' FA2F',z' FC1F',z' 0000' &
401 , "R",z' 0000',z' 0620',z' 0E30',z' 1E38',z' 3C38',z' 7838',z' F038' &
402 ,z' E039',z' 0038',z' F037',z' F80F',z' F437',z' 0E38',z' 0E38' &
403 ,z' 0E38',z' 0E38',z' 0E38',z' F437',z' F82F',z' F01F',z' 0000' &
404 , "S",z' 0000',z' F007',z' F80F',z' F407',z' 0E00',z' 0E00',z' 0E00' &
405 ,z' 0E00',z' 0600',z' F407',z' F80F',z' F037',z' 0038',z' 0038' &
406 ,z' 0038',z' 0038',z' 0038',z' F017',z' F80F',z' F007',z' 0000' &
407 , "T",z' 0000',z' 0001',z' 8001',z' C001',z' C001',z' C001',z' C001' &
408 ,z' C001',z' C001',z' 8000',z' 0000',z' 8000',z' C001',z' C001' &
409 ,z' C001',z' C001',z' C001',z' B80E',z' 7C1F',z' 7E3F',z' 0000' &
410 , "U",z' 0000',z' FC1F',z' FA2F',z' F637',z' 0E38',z' 0E38',z' 0E38' &
411 ,z' 0E38',z' 0E38',z' 0630',z' 0000',z' 0630',z' 0E38',z' 0E38' &
412 ,z' 0E38',z' 0E38',z' 0E38',z' 0E38',z' 0630',z' 0220',z' 0000' &
413 , "V",z' 0000',z' 6003',z' 7007',z' 7007',z' 380E',z' 380E',z' 1C1C' &
414 ,z' 1C1C',z' 0E38',z' 0630',z' 0000',z' 0630',z' 0E38',z' 0E38' &
415 ,z' 0E38',z' 0E38',z' 0E38',z' 0E38',z' 0630',z' 0220',z' 0000' &
416 , "W",z' 0000',z' 3C1F',z' BA2E',z' D635',z' CE39',z' CE39',z' CE39' &
417 ,z' CE39',z' CE39',z' 8630',z' 0000',z' 0630',z' 0E38',z' 0E38' &
418 ,z' 0E38',z' 0E38',z' 0E38',z' 0E38',z' 0630',z' 0220',z' 0000' &
419 , "X",z' 0000',z' 0E38',z' 0E38',z' 1C1C',z' 1C1C',z' 380E',z' 380E' &
420 ,z' 7007',z' 7007',z' 6003',z' 0000',z' 6003',z' 7007',z' 7007' &

```



```

421 ,z' 380E',z' 380E',z' 1C1C',z' 1C1C',z' 0E38',z' 0E38',z' 0000' &
422 ,"Y",z' 0000',z' 0001',z' 8001',z' C001',z' C001',z' C001',z' C001' &
423 ,z' C001',z' C001',z' C001',z' 8000',z' 6003',z' 7007',z' 7007' &
424 ,z' 380E',z' 380E',z' 1C1C',z' 1C1C',z' 0E38',z' 0E38',z' 0000' &
425 ,"Z",z' 0000',z' FE1F',z' FC2F',z' F837',z' 0038',z' 003C',z' 001E' &
426 ,z' 000F',z' 0007',z' 4003',z' 6001',z' 7000',z' 7800',z' 3C00' &
427 ,z' 1E00',z' 0E00',z' 0E00',z' F60F',z' FA1F',z' FC3F',z' 0000' &
428 ,"a",z' 0000',z' C20F',z' E617',z' EE3B',z' 0E38',z' 0E38',z' 0E38' &
429 ,z' 0E38',z' 0E38',z' F637',z' F80F',z' F007',z' 0000',z' 0000' &
430 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
431 ,"b",z' 0000',z' F01F',z' F82F',z' F437',z' 0E38',z' 0E38',z' 0E38' &
432 ,z' 0E38',z' 0E38',z' F437',z' F80F',z' F037',z' 0038',z' 0038' &
433 ,z' 0038',z' 0038',z' 0038',z' 0030',z' 0020',z' 0000',z' 0000' &
434 ,"c",z' 0000',z' F807',z' FCF',z' FE17',z' 0038',z' 0038',z' 0038' &
435 ,z' 0038',z' 0038',z' FE37',z' FCF',z' F807',z' 0000',z' 0000' &
436 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
437 ,"d",z' 0000',z' F01F',z' F82F',z' F437',z' 0E38',z' 0E38',z' 0E38' &
438 ,z' 0E38',z' 0E38',z' F637',z' F80F',z' F607',z' 0E00',z' 0E00' &
439 ,z' 0E00',z' 0E00',z' 0E00',z' 0400',z' 0000',z' 0000',z' 0000' &
440 ,"e",z' 0000',z' F807',z' FCF',z' F817',z' 0038',z' 0038',z' F037' &
441 ,z' F80F',z' F437',z' 0E38',z' 0E38',z' F417',z' F80F',z' F007' &
442 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
443 ,"f",z' 0000',z' 0020',z' 0030',z' 0038',z' 0038',z' 0038',z' 0038' &
444 ,z' 0038',z' C037',z' E00F',z' C037',z' 0038',z' 0038',z' 0038' &
445 ,z' 0038',z' F017',z' F80F',z' F007',z' 0000',z' 0000',z' 0000' &
446 ,"~",z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
447 ,z' 1000',z' 5820',z' DC30',z' CE39',z' 861D',z' 020D',z' 0004' &
448 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
449 ,"o",z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' & ! (=char(176))
450 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 6003',z' 7007',z' 380E' &
451 ,z' 180C',z' 0000',z' 180C',z' 380E',z' 7007',z' 6003',z' 0000' &
452 /
453 Data(cAsciiC(j), (iFLineV(i,j), i=1, 21), j= 1, 30) / & !Voids.
454 " ",z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
455 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
456 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
457 ,"'",z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
458 ,z' 0000',z' 0000',z' 0000',z' 7007',z' D80D',z' 8808',z' 8808' &
459 ,z' 8808',z' 8808',z' 8808',z' 8808',z' 8808',z' D80D',z' 7007' &
460 ,"&",z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' & ! (full-on)
461 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
462 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
463 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
464 ,z' 0000',z' 0000',z' 0000',z' C001',z' 6003',z' 2002',z' 2002' &
465 ,z' 2002',z' 2002',z' 2002',z' 2002',z' 2000',z' 2002',z' 6003',z' C001' &
466 ,"(",z' 7C00',z' C400',z' 8C01',z' 5803',z' 2002',z' 2002',z' 2002' &
467 ,z' 2002',z' 2002',z' 6002',z' C003',z' 6002',z' 2002',z' 2002' &
468 ,z' 2002',z' 2002',z' 2002',z' 5803',z' 8C01',z' C400',z' 7C00' &
469 ,")",z' 001F',z' 8011',z' C018',z' 600D',z' 2006',z' 2002',z' 2002' &
470 ,z' 2002',z' 2002',z' 2003',z' E001',z' 2003',z' 2002',z' 2002' &
471 ,z' 2002',z' 2002',z' 2006',z' 600D',z' C018',z' 8011',z' 001F' &
472 ,"*",z' 0000',z' 0000',z' 0000',z' 0000',z' 780F',z' C809',z' 8808' &
473 ,z' 980C',z' BC1E',z' C631',z' 8220',z' C631',z' BC1E',z' 980C' &
474 ,z' 8808',z' C809',z' 780F',z' 0000',z' 0000',z' 0000',z' 0000' &
475 ,"+",z' 0000',z' 0000',z' 0000',z' C001',z' 6003',z' 2002',z' 2002' &
476 ,z' 2002',z' 3E3E',z' 4361',z' 8140',z' 4361',z' 3E3E',z' 2002' &
477 ,z' 2002',z' 2002',z' 6003',z' C001',z' 0000',z' 0000',z' 0000' &
478 ,",",z' C007',z' 6004',z' 3006',z' 1803',z' 8801',z' F800',z' 0000' &
479 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
480 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
481 ,"-",z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
482 ,z' 0000',z' F80F',z' 0C18',z' 0410',z' 0C18',z' F80F',z' 0000' &
483 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
484 ,".",z' E003',z' 3006',z' 1004',z' 3006',z' E003',z' 0000',z' 0000' &
485 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
486 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
487 ,"/",z' 0000',z' 0000',z' 0070',z' 0058',z' 004C',z' 0046',z' 0063' &
488 ,z' 8031',z' 8018',z' E00C',z' B006',z' 9803',z' 8C00',z' C600' &
489 ,z' 6300',z' 3100',z' 1900',z' 0D00',z' 0700',z' 0000',z' 0000' &
490 ,"0",z' F80F',z' 0C18',z' 0630',z' 0B68',z' F147',z' 1144',z' 1144' &

```

491 ,z' 1144',z' 1144',z' 194C',z' 0F78',z' 194C',z' 1144',z' 1144' &
492 ,z' 1144',z' 1144',z' F147',z' 0B68',z' 0630',z' 0C18',z' F80F' &
493 ,"1",z' FC1F',z' 8410',z' 8C18',z' 580D',z' 3006',z' 2002',z' 2002' &
494 ,z' 2002',z' 2002',z' 6003',z' C001',z' 6003',z' 2002',z' 2002' &
495 ,z' 2002',z' 2002',z' 2006',z' 200D',z' A018',z' 6010',z' C01F' &
496 ,"2",z' FF3F',z' 0160',z' 0350',z' 0648',z' FC47',z' 0044',z' 0044' &
497 ,z' 0044',z' F847',z' 0C68',z' 0630',z' 0B18',z' F10F',z' 1100' &
498 ,z' 1100',z' 1100',z' F10F',z' 0B18',z' 0610',z' 0C18',z' F80F' &
499 ,"3",z' F87F',z' 0C40',z' 0660',z' 0B30',z' F11F',z' 1100',z' 1100' &
500 ,z' 1100',z' F10F',z' 0B18',z' 0610',z' 0B18',z' F10F',z' 1100' &
501 ,z' 1100',z' 1100',z' F11F',z' 0B30',z' 0660',z' 0C40',z' F87F' &
502 ,"4",z' 0700',z' 0D00',z' 1900',z' 1100',z' 1100',z' 1100',z' 1100' &
503 ,z' 1100',z' F13F',z' 0960',z' 0550',z' 0948',z' F147',z' 1144' &
504 ,z' 1144',z' 1144',z' 1144',z' 1144',z' 1144',z' 194C',z' 0D58',z' 0770' &
505 ,"5",z' F80F',z' 0C18',z' 0610',z' 0B18',z' F10F',z' 1100',z' 1100' &
506 ,z' 1100',z' F10F',z' 0B18',z' 0670',z' 0C48',z' F847',z' 0044' &
507 ,z' 0044',z' 0044',z' F847',z' 0C48',z' 0650',z' 0260',z' FE3F' &
508 ,"6",z' F80F',z' 0C18',z' 0630',z' 0B68',z' F147',z' 1144',z' 1144' &
509 ,z' 1144',z' F147',z' 0B48',z' 0670',z' 0C48',z' F847',z' 0044' &
510 ,z' 0044',z' 0044',z' F847',z' 0C68',z' 0430',z' 0418',z' FCOF' &
511 ,"7",z' 0700',z' 0D00',z' 1900',z' 1100',z' 1100',z' 1100',z' 1100' &
512 ,z' 1100',z' 1100',z' 1900',z' 0F00',z' 1900',z' 1100',z' 1100' &
513 ,z' 1100',z' 1100',z' F11F',z' 0930',z' 0560',z' 0340',z' FE7F' &
514 ,"8",z' F80F',z' 0C18',z' 0630',z' 0B68',z' F147',z' 1140',z' 1140' &
515 ,z' 1140',z' F147',z' 0B68',z' 0630',z' 0B68',z' F147',z' 1144' &
516 ,z' 1144',z' 1144',z' F147',z' 0B68',z' 0630',z' 0C18',z' F80F' &
517 ,"9",z' F87F',z' 0C40',z' 0660',z' 0B30',z' F11F',z' 1100',z' 1100' &
518 ,z' 1100',z' F10F',z' 0918',z' 0730',z' 0968',z' F147',z' 1144' &
519 ,z' 1144',z' 1144',z' F147',z' 0B68',z' 0630',z' 0C18',z' F80F' &
520 ,":",z' 0000',z' 0000',z' 0000',z' C001',z' 6003',z' 2002',z' 2002' &
521 ,z' 2002',z' 6003',z' C001',z' 0000',z' C001',z' 6003',z' 2002' &
522 ,z' 2002',z' 2002',z' 6003',z' C001',z' 0000',z' 0000',z' 0000' &
523 ,"<",z' 0000',z' 0000',z' 0000',z' 0000',z' F803',z' 1806',z' 300C' &
524 ,z' 6018',z' C030',z' 8021',z' 003F',z' 8021',z' C030',z' 6018' &
525 ,z' 300C',z' 1806',z' FC03',z' 0000',z' 0000',z' 0000',z' 0000' &
526 ,"=",z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' F80F',z' 0C18' &
527 ,z' 0410',z' 0C18',z' F80F',z' 0000',z' F80F',z' 0C18',z' 0410' &
528 ,z' 0C18',z' F80F',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
529 ,">",z' 0000',z' 0000',z' 0000',z' 0000',z' E01F',z' 300C',z' 1806' &
530 ,z' 0C03',z' 8601',z' C200',z' 7E00',z' C200',z' 8601',z' 0C03' &
531 ,z' 1806',z' 300C',z' E01F',z' 0000',z' 0000',z' 0000',z' 0000' &
532 ,"A",z' 0770',z' 0D58',z' 194C',z' 1144',z' 1144',z' 1144',z' 1144' &
533 ,z' F147',z' F147',z' 0948',z' 0770',z' 0948',z' F147',z' 1144' &
534 ,z' 1144',z' 1144',z' F147',z' 0B68',z' 0630',z' 0C18',z' F80F' &
535 ,"B",z' F83F',z' 0C60',z' 0650',z' 0B48',z' F147',z' 1144',z' 1144' &
536 ,z' 1144',z' F147',z' 0B48',z' 0670',z' 0B48',z' F147',z' 1144' &
537 ,z' 1144',z' 1144',z' F147',z' 0B48',z' 0650',z' 0C60',z' F83F' &
538 ,"C",z' FCOF',z' 0618',z' 0330',z' 0168',z' FF47',z' 0044',z' 0044' &
539 ,z' 0044',z' 0044',z' 004C',z' 0078',z' 004C',z' 0044',z' 0044' &
540 ,z' 0044',z' 0044',z' FF47',z' 0168',z' 0330',z' 0618',z' FCOF' &
541 ,"D",z' F83F',z' 0C60',z' 0650',z' 0B48',z' F147',z' 1144',z' 1144' &
542 ,z' 1144',z' 1144',z' 194C',z' 0F78',z' 194C',z' 1144',z' 1144' &
543 ,z' 1144',z' 1144',z' F147',z' 0B48',z' 0650',z' 0C60',z' F83F' /
544 Data(cAsciIC(j), (iFLineV(i,j), i=1, 21), j=31, 60) / & !Voids.
545 ,"E",z' FF3F',z' 0160',z' 0350',z' 0648',z' FC47',z' 0044',z' 0044' &
546 ,z' 0044',z' F847',z' 0C48',z' 0470',z' 0C48',z' F847',z' 0044' &
547 ,z' 0044',z' 0044',z' FC47',z' 0648',z' 0350',z' 0160',z' FF3F' &
548 ,"F",z' 0070',z' 0058',z' 004C',z' 0044',z' 0044',z' 0044',z' 0044' &
549 ,z' 0044',z' F847',z' 0C48',z' 0470',z' 0C48',z' F847',z' 0044' &
550 ,z' 0044',z' 0044',z' FC47',z' 0648',z' 0350',z' 0160',z' FF3F' &
551 ,"G",z' F80F',z' 0C18',z' 0630',z' 0B68',z' F147',z' 1144',z' 1144' &
552 ,z' 1144',z' 7144',z' C94C',z' 8578',z' C34C',z' 7E44',z' 0044' &
553 ,z' 0044',z' 0044',z' FE47',z' 0268',z' 0630',z' 0C18',z' F80F' &
554 ,"H",z' 0770',z' 0D58',z' 194C',z' 1144',z' 1144',z' 1144',z' 1144' &
555 ,z' 1144',z' F147',z' 0948',z' 0770',z' 0948',z' F147',z' 1144' &
556 ,z' 1144',z' 1144',z' 1144',z' 1144',z' 194C',z' 0D58',z' 0770' &
557 ,"I",z' FC1F',z' 8410',z' 8C18',z' 580D',z' 3006',z' 2002',z' 2002' &
558 ,z' 2002',z' 2002',z' 6003',z' C001',z' 6003',z' 2002',z' 2002' &
559 ,z' 2002',z' 2002',z' 3006',z' 580D',z' 8C18',z' 8410',z' FC1F' &
560 ,"J",z' F80F',z' 0C18',z' 0630',z' 0B68',z' F147',z' 1144',z' 114C' &

561 ,z' 1158',z' 1170',z' 1900',z' 0F00',z' 1900',z' 1100',z' 1100' &
562 ,z' 1100',z' 1100',z' 1100',z' 1100',z' 1900',z' 0D00',z' 0700' &
563 , "K",z' 0770',z' 0D58',z' 194C',z' 3144',z' 6144',z' C344',z' 8645' &
564 ,z' 0C47',z' 184E',z' 305C',z' E063',z' 8061',z' C050',z' 6048' &
565 ,z' 3044',z' 1846',z' 0C47',z' 8645',z' C34C',z' 6158',z' 3F70' &
566 , "L",z' FF3F',z' 0160',z' 0350',z' 0648',z' FC47',z' 0044',z' 0044' &
567 ,z' 0044',z' 0044',z' 004C',z' 0078',z' 004C',z' 0044',z' 0044' &
568 ,z' 0044',z' 0044',z' 0044',z' 004C',z' 0048',z' 0058',z' 0070' &
569 , "M",z' 0770',z' 0D58',z' D949',z' 7147',z' 3146',z' 3146',z' 3146' &
570 ,z' 3146',z' 3146',z' 3146',z' 794F',z' CF79',z' 794F',z' 3146',z' 3146' &
571 ,z' 3146',z' 3146',z' 3146',z' 294A',z' 4551',z' 8360',z' FE3F' &
572 , "N",z' 3F70',z' 6158',z' C34C',z' 8545',z' 0945',z' 9145',z' 7147' &
573 ,z' 3146',z' 3146',z' 794F',z' CF79',z' 794F',z' 3146',z' 3146' &
574 ,z' 7147',z' D144',z' 5148',z' D150',z' 9961',z' 0D43',z' 077E' &
575 , "O",z' FE3F',z' 0360',z' 0550',z' 0948',z' F147',z' 1144',z' 1144' &
576 ,z' 1144',z' 1144',z' 194C',z' 0F78',z' 194C',z' 1144',z' 1144' &
577 ,z' 1144',z' 1144',z' F147',z' 0948',z' 0550',z' 0360',z' FE3F' &
578 , "P",z' 0070',z' 0058',z' 004C',z' 0044',z' 0044',z' 0044',z' 0044' &
579 ,z' 0044',z' F847',z' 0C48',z' 0670',z' 0B48',z' F147',z' 0144' &
580 ,z' 0144',z' 0144',z' F147',z' 0B48',z' 0650',z' 0C60',z' F83F' &
581 , "Q",z' FF3F',z' 1960',z' 1150',z' 2348',z' C747',z' 8944',z' 9144' &
582 ,z' F144',z' 1144',z' 194C',z' 0F78',z' 194C',z' 1144',z' 1144' &
583 ,z' 1144',z' 1144',z' F147',z' 0948',z' 0550',z' 0360',z' FE3F' &
584 , "R",z' 0F70',z' 1958',z' 314C',z' 6144',z' C344',z' 8645',z' 0C47' &
585 ,z' 1846',z' F847',z' 0C48',z' 0670',z' 0B48',z' F147',z' 1144' &
586 ,z' 1144',z' 1144',z' F147',z' 0B48',z' 0650',z' 0C60',z' F83F' &
587 , "S",z' F80F',z' 0C18',z' 0610',z' 0B18',z' F10F',z' 1100',z' 1100' &
588 ,z' 1100',z' F90F',z' 0A18',z' 0670',z' 0C48',z' F847',z' 0044' &
589 ,z' 0044',z' 0044',z' F847',z' 0C68',z' 0430',z' 0C18',z' F80F' &
590 , "T",z' 8003',z' C002',z' 6002',z' 2002',z' 2002',z' 2002',z' 2002' &
591 ,z' 2002',z' 2002',z' 6003',z' C001',z' 6003',z' 2002',z' 2002' &
592 ,z' 2002',z' 2002',z' 3C1E',z' 4631',z' 8360',z' 8140',z' FF7F' &
593 , "U",z' FE3F',z' 0360',z' 0550',z' 0948',z' F147',z' 1144',z' 1144' &
594 ,z' 1144',z' 1144',z' 194C',z' 0F78',z' 194C',z' 1144',z' 1144' &
595 ,z' 1144',z' 1144',z' 1144',z' 1144',z' 194C',z' 0D58',z' 0770' &
596 , "V",z' F007',z' 980C',z' 8808',z' 8C18',z' C411',z' 4631',z' 6223' &
597 ,z' 2362',z' 3146',z' 194C',z' 0F78',z' 194C',z' 1144',z' 1144' &
598 ,z' 1144',z' 1144',z' 1144',z' 1144',z' 194C',z' 0D58',z' 0770' &
599 , "W",z' FE3F',z' C360',z' 4551',z' 294A',z' 3146',z' 3146',z' 3146' &
600 ,z' 3146',z' 3146',z' 794F',z' CF79',z' 194C',z' 1144',z' 1144' &
601 ,z' 1144',z' 1144',z' 1144',z' 1144',z' 194C',z' 0D58',z' 0770' &
602 , "X",z' 1F7C',z' 1144',z' 3146',z' 2362',z' 6222',z' 4631',z' C411' &
603 ,z' 8C18',z' 8808',z' 980C',z' F007',z' 980C',z' 8808',z' 8C18' &
604 ,z' C411',z' 4631',z' 6223',z' 2362',z' 3146',z' 1144',z' 1F7C' &
605 , "Y",z' 8003',z' C002',z' 6002',z' 2002',z' 2002',z' 2002',z' 2002' &
606 ,z' 2002',z' 2002',z' 2002',z' 7007',z' 980C',z' 8808',z' 8C18' &
607 ,z' C411',z' 4631',z' 6223',z' 2362',z' 3146',z' 1144',z' 1F7C' &
608 , "Z",z' FF3F',z' 0160',z' 0350',z' 0648',z' FC47',z' 0043',z' 8061' &
609 ,z' C030',z' E018',z' B00C',z' 9806',z' 8C03',z' 8600',z' C300' &
610 ,z' 6100',z' 3100',z' F11F',z' 0930',z' 0560',z' 0340',z' FE7F' &
611 , "a",z' E71F',z' 3D30',z' 1968',z' 1144',z' F147',z' 1144',z' 1144' &
612 ,z' 1144',z' F147',z' 0948',z' 0770',z' 0C18',z' F80F',z' 0000' &
613 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
614 , "b",z' F83F',z' 0C60',z' 0650',z' 0B48',z' F147',z' 1144',z' 1144' &
615 ,z' 1144',z' F147',z' 0B48',z' 0670',z' 0C48',z' F847',z' 0044' &
616 ,z' 0044',z' 0044',z' 0044',z' 004C',z' 0058',z' 0070',z' 0000' &
617 , "c",z' FCOF',z' 0618',z' 0330',z' 0168',z' FF47',z' 0044',z' 0044' &
618 ,z' 0044',z' FF47',z' 0148',z' 0370',z' 0618',z' FCOF',z' 0000' &
619 ,z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
620 , "d",z' F83F',z' 0C60',z' 0650',z' 0B48',z' F147',z' 1144',z' 1144' &
621 ,z' 1144',z' F147',z' 0948',z' 0770',z' 0918',z' F10F',z' 1100' &
622 ,z' 1100',z' 1100',z' 1100',z' 0B00',z' 0E00',z' 0400',z' 0000' &
623 , "e",z' FFOF',z' 0718',z' 0330',z' 0668',z' FC47',z' F847',z' 0C48' &
624 ,z' 0670',z' 0B48',z' F147',z' F147',z' 0B68',z' 0630',z' 0C18' &
625 ,z' F80F',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
626 , "f",z' 0070',z' 0058',z' 004C',z' 0044',z' 0044',z' 0044',z' 0044' &
627 ,z' E047',z' 3048',z' 1070',z' 3048',z' E047',z' 0044',z' 0044' &
628 ,z' F847',z' 0C68',z' 0430',z' 0C18',z' F80F',z' 0000',z' 0000' &
629 , "~",z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 3800' &
630 ,z' EC70',z' A659',z' 234F',z' 3146',z' 7962',z' CD32',z' 871B' &

```

631 ,z' 000E',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' &
632 , "o",z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000',z' 0000' & !(=char(176)
633 ,z' 0000',z' 0000',z' 0000',z' F007',z' 980C',z' 8C18',z' C411' &
634 ,z' 6413',z' 3C1E',z' 6413',z' C411',z' 8C18',z' 980C',z' F007' &
635 /
636
637 if(Init.gt.0) goto 10 !Initialization
638 iAscii=0 !Zeros the iAscii array.
639 do 5 n=1,nCharTot
640 i=iChar(cAsciiC(n))
641 5 iAscii(i)=n
642 FontCT=FontC !Put the data in "targetable" (pointable) form
643 FontVT=FontV ; init=1 ! for eventual use in glBitmap().
644
645 10 continue !Display the characters
646 call glMatrixMode(GL_MODELVIEW) ; call glPushMatrix; call glLoadIdentity
647 call glPixelStorei(GL_UNPACK_ALIGNMENT,1)
648
649 LabL40=char(0)
650 LabL40=LabelL
651 LabL1(40)=char(0)
652 do 30 iPass=1,3 !Draw the background,then the character,then the voids.
653 select case(iPass) !Color selection
654 case(1); if(iColorB.lt.0) goto 30 ; call Colors(iColorB) !Background
655 case(2); if(iColorC.lt.0) goto 30 ; call Colors(iColorC) !Characters
656 case(3); if(iColorV.lt.0) goto 30 ; call Colors(iColorV) !Void
657 end select !iPass
658 do 20 i=1,40 !Process the input character string.
659 n=iChar(LabL1(i)) ; if(n.eq.0) goto 21 !Null terminates the string.
660 Xgl=x(1)+(i-1)*(16*2.666)/640.; Ygl=x(2) ; Zgl=x(3) !Compute draw point.
661 select case(iPass) !Select the pixel pattern to draw.
662 case(1); n=3; Zgl=Zgl+.001 !Background. n=3 is "&".
663 FontP=>FontCT(1:42,n) ! Drawn behind.
664 case(2); n=iAscii(n) ; if(n.eq.0) goto 20 !Character
665 FontP=>FontCT(1:42,n)
666 case(3); n=iAscii(n) ; if(n.eq.0) goto 20 !Void
667 FontP=>FontVT(1:42,n)
668 end select !iPass
669 call glRasterPos3f(Xgl,Ygl,Zgl) !Move to the 3-D draw point.
670 call glBitmap(16,21,0.,0.,0.,0.,FontP) !Draw the pixel pattern
671 20 continue
672 21 continue
673 30 continue
674 call glFlush !Command to OpenGL to "complete the process".
675 call glMatrixMode(GL_MODELVIEW) ; call glPopMatrix
676 return
677 end Subroutine AlphaCF
678 !-----7 9
679 End Module CFontSup
680 !-----7 9
681
682 Program CFont
683 use OpenGL
684 use CFontSup !^CFontSup-CFontUse,KeybdUse
685
686 implicit none
687 integer*4::wini d
688
689 write(6, "(1x,' Contrast Font Demonstration / Generation' " // &
690 "/1x,' CFont.Exe version 1.2 - Jeff Setterholm' " // &
691 "/1x/1x,' Minimize this screen unless you're converting CFontIn.Txt.')")
692
693 call glutInit !Initialize OpenGL
694 call glutInitDisplayMode(ior(GLUT_DOUBLE,ior(GLUT_RGBA, GLUT_DEPTH)))
695 call glutInitWindowSize(640,480)
696 call glutInitWindowPosition(0_glint,0_glint)
697 call glClear(GL_COLOR_BUFFER_BIT)
698 call glColor3f(1.,1.,0.)
699 wini d = glutCreateWindow("CFont.Exe Version 1.3")
700 call glEnable(GL_DEPTH_TEST)

```

```

701
702 call glutDisplayFunc(ScrniUse)      !Identify the callback routines
703 call glutIdleFunc(ScrniUse)
704 call glutKeyboardFunc(KeybdUse)
705 call MenuInit
706
707 call glutSwapBuffers
708 call glutPostRedisplay
709 call glutMainLoop                    !Transfer control to OpenGL
710 End Program CFont
711 !-----7 9
712
713 !-----7 9
714 !Pressing the "^" key: activates the following subroutine,
715 ! after which the program halts.
716 Subroutine CFontGen
717 !2004.02.24.0935cst JMS -A700/Nt4.0:sp6/CVF6.6b/Ogl1.2.1/Glut3.7.6
718
719 !Subroutine "CFontGen": Copyright 2004 by Jeffrey M. Setterholm
720 !This subroutine may be incorporated or adapted into your software
721 ! without permission and without license fees.
722
723 !Reads: "CFontIn.Txt" which has "C"s as characters & "-"s as voids/outlines.
724 ! A 15 x 21 pixel font. (Other sizes not tested.) e.g. the letter "A":
725 ! (symbolic) -> (hexadecimal (z''))
726 !65 "A" !-----15x21 C's -'s
727 ! . . . - - - - - . . . z' 0000' z' 0770'
728 ! . . - C C C C C C C - - . z' 0220' z' 0D58'
729 ! . - C C C C C C C C - - . z' 0630' z' 194C'
730 ! - - C - C C C C C C - C - - z' 0E38' z' 1144'
731 ! - C C C - - - - - C C C - z' 0E38' z' 1144'
732 ! - C C C - . . . . - C C C - z' 0E38' z' 1144'
733 ! - C C C - . . . . - C C C - z' 0E38' z' 1144'
734 ! - C C C - . . . . - C C C - z' 0E38' z' F147'
735 ! - C C C - - - - - C C C - z' 0E38' z' F147'
736 ! - C C - C C C C C C C - C C - z' F637' z' 0948'
737 ! - - C C C C C C C C - - - z' F80F' z' 0770'
738 ! - C C - C C C C C C C - C C - z' F637' z' 0948'
739 ! - C C C - - - - - C C C - z' 0E38' z' F147'
740 ! - C C C - - - - - C C C - z' 0E38' z' 1144'
741 ! - C C C - . . . . - C C C - z' 0E38' z' 1144'
742 ! - C C C - . . . . - C C C - z' 0E38' z' 1144'
743 ! - C C C - . . . . - C C C - z' 0E38' z' F147'
744 ! - C C C - . . . . - C C C - z' F417' z' 0B68'
745 ! - C C - - - - - C C - z' F80F' z' 0630'
746 ! - C - . . . . - C - z' F007' z' 0C18'
747 ! - - . . . . - - - z' 0000' z' F80F'
748 !^a 16th blank column on the left side is implicitly in the hex representation.
749
750 !Produces: "CFontOut.Txt" which can be copied into "AlphaCF" as the font.
751 ! A 16 x 21 pixel font (with the extra column on the left side).
752 !e.g. the letter "A" as copied into subroutine "AlphaCF" as:
753 ! , "A", z' 0000', z' 0220', z' 0630', z' 0E38', z' 0E38', z' 0E38', z' 0E38' & character
754 ! , z' 0E38', z' 0E38', z' F637', z' F80F', z' F637', z' 0E38', z' 0E38' &
755 ! , z' 0E38', z' 0E38', z' 0E38', z' F417', z' F80F', z' F007', z' 0000' &
756 ! , "A", z' 0770', z' 0D58', z' 194C', z' 1144', z' 1144', z' 1144', z' 1144' & void/border
757 ! , z' F147', z' F147', z' 0948', z' 0770', z' 0948', z' F147', z' 1144' &
758 ! , z' 1144', z' 1144', z' F147', z' 0B68', z' 0630', z' 0C18', z' F80F' &
759
760 ! (symbolic) -> (hexadecimal (z''))
761 !47 "/" !-----15x21 C's -'s
762 ! . . . . . z' 0000' z' 0000'
763 ! . . . . . z' 0000' z' 0000'
764 ! . . . . . z' 0000' z' 0070'
765 ! . . . . . - - C z' 0020' z' 0058'
766 ! . . . . . - C C z' 0030' z' 004C'
767 ! . . . . . - C C C z' 0038' z' 0046'
768 ! . . . . . - C C C z' 001C' z' 0063'
769 ! . . . . . - C C C z' 000E' z' 8031'
770 ! . . . . . - C C C z' 0007' z' 8018'

```

```

771 ! . . . . . - - - C C - - . . . . . z' 0003' z' E00C'
772 ! . . . . . - - C - C - - . . . . . z' 4001' z' B006'
773 ! . . . . . - - C C - - . . . . . z' 6000' z' 9803'
774 ! . . . . . - - C C C - . . . . . z' 7000' z' 8C00'
775 ! . . . . . - - C C C - . . . . . z' 3800' z' C600'
776 ! . . . . . - - C C C - . . . . . z' 1C00' z' 6300'
777 ! . . . . . - - C C C - . . . . . z' 0E00' z' 3100'
778 ! . . . . . - - C C - - . . . . . z' 0600' z' 1900'
779 ! . . . . . - - C - - . . . . . z' 0200' z' 0D00'
780 ! . . . . . - - - - . . . . . z' 0000' z' 0700'
781 ! . . . . . - - - - . . . . . z' 0000' z' 0000'
782 ! . . . . . - - - - . . . . . z' 0000' z' 0000'
783 !How the hexadecimal codes should look was not obvious to me.
784 !I experimented with the hexadecimal generator until it worked.
785
786 !In this implementation:
787 ! The "&" (38d) symbol produces full-on. (255d)
788
789 implicit none
790 integer*4::nHor, mVer, nCharTot, nPower
791 integer*4::iAsc, iComma, iPass, m, n, nC, nCSeq !, nA
792 integer*4::iAscii (0:255)
793 integer*4::i4Line(21,128,2) !128 characters max.
794 character:: cString(31)*1, cString31*31
795 equivalence( cString, cString31 )
796
797 nCSeq=0
798 i4Line=0
799 open(unit=14, file='CFontIn.Txt', action='read') !Import the character set.
800 read(14, *) nHor, mVer, nCharTot !Horizontal and vertical pixels + #char's
801 if((nHor.ne.15).or.(mVer.ne.21)) stop 'Expects 15 x 21 pixel font input.'
802 if(nCharTot.gt.128) stop 'Character count must be <= 128.'
803 do 10 nC=1, nCharTot
804 read(14, *) iAsc
805 if(iAsc.eq.0) goto 11 !"0" at beginning of line ends reading.
806 nCSeq=nC !Otherwise, nCharTot characters are read.
807 iAscii(iAsc)=nCSeq !nCSeq: number of Character in sequence.
808 write(6, "(1x, 'nC=', i3, 2x, 'iAsc=', i3, 2x, 3a1)") & !Prints to the DOS screen
809 nCSeq, iAsc, char(34), char(iAsc), char(34)
810 do 5 m=1, mVer
811 read(14, "(a31)") cString31
812 do 4 n=1, nHor !Hexadecimal generator.
813 nPower= 7 - n !Reverse the bits and swap the bytes.
814 if(n.gt.7) nPower=23 - n
815 if(cString(n*2+1).eq.'-') & !Voids
816 i4Line(m, nCSeq, 1)=i4Line(m, nCSeq, 1)+2**(nPower)
817 if(cString(n*2+1).eq.'C') & !Characters
818 i4Line(m, nCSeq, 2)=i4Line(m, nCSeq, 2)+2**(nPower)
819 4 continue
820 5 if(iAsc.eq.38) i4Line(m, nCSeq, 2)=z'FFFF' !Blanks a 16th column on the
821 10 continue !char(38)="&" !left side of the character.
822 11 close(14)
823 write(6, "(1x, i3, ' characters read. Writing CFontOut.Txt...')") nCSeq
824
825 open(unit=15, file='CFontOut.Txt', action='write') !Export the representation.
826 write(15, "( !This is the CFontOut.Txt output:)")
827 do 20 iPass=2, 1, -1 !Write the characters, then the voids/borders.
828 ! An extremely format-specific write follows (which works in CVF6.6b):
829 if(iPass.eq.2) write(15 &
830 , "('Data(cAsciiC(j), (iFLineC(i,j), i=1, 21), j=1, ', i3, ') / & !Characters. ')") &
831 nCharTot
832 if(iPass.eq.1) write(15 &
833 , "('Data(cAsciiC(j), (iFLineV(i,j), i=1, 21), j=1, ', i3, ') / & !Voids. ')") &
834 nCharTot
835 iComma=32
836 ! Write the characters in ASCII# order, skipping "underfined"s:
837 do 15 iAsc=0, 255 ; nCSeq=iAscii(iAsc) ; if(nCSeq.eq.0) goto 15
838 nC=34; if(iAsc.eq.34) nC=39 !puts apostrophes around the quote.
839 write(15, "(4a1, 7(' , z' , a1, z4.4, a1), ' &')") &
840 char(iComma), char(nC), char(iAsc), char(nC) &

```

```
841      , (char(39), i4Line(m, nCSeq, iPass), char(39), m=21, 15, - 1)
842      write(15, "(4x , 7(' , z' , a1, z4. 4, a1), ' &' )" &
843      (char(39), i4Line(m, nCSeq, iPass), char(39), m=14, 8, - 1)
844      write(15, "(4x , 7(' , z' , a1, z4. 4, a1), ' &' )" &
845      (char(39), i4Line(m, nCSeq, iPass), char(39), m= 7, 1, - 1)
846      iComma=44
847 15  continue
848      write(15, "(' /') /1x")
849 20  continue
850      close(15)
851      stop 'CFontOut.Txt written. Copy into CFont.F90: subroutine AlphaCF.' !Done
852      return
853 End Subroutine CFontGen
854 !-----7 9
```