

LINUS ROMER

---

---

*The*

# FETAMONT

*Typeface*

---

---

DESIGN AND CONSTRUCTIONS

JANUARY 10, 2014

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Comparison With Existing Logos</b>	<b>2</b>
<b>3</b>	<b>Naming Scheme For The Fetamont Faces</b>	<b>3</b>
<b>4</b>	<b>Special Techniques</b>	<b>3</b>
4.1	Arc Constructions . . . . .	4
4.2	Glyph names . . . . .	5
4.3	Combined Characters . . . . .	5
4.4	Kerning Classes with METAFONT . . . . .	7
4.5	Italic Corrections . . . . .	11
4.6	Producing Outlines . . . . .	11
<b>5</b>	<b>Construction Of The Most Important Letters</b>	<b>13</b>
<b>6</b>	<b>Font Tables</b>	<b>97</b>

## 1 Introduction

The *Logo* typeface, which is known from logos like METAFONT or METAPOST, used to be very limited in the number of glyphs. *Fetamont* is a new typeface that extends the Logo typeface in two ways:

- Fetamont consists of 256 glyphs, such that the T1-encoding<sup>1</sup> table is complete now.
- Fetamont has additional faces like “light ultracondensed” or “script”.

Fetamont is distributed over CTAN under the terms of the *L<sup>A</sup>T<sub>E</sub>X Project Public License* (LPPL).

This document describes the design and the constructions of the typeface itself. The L<sup>A</sup>T<sub>E</sub>X support for the Fetamont typeface is described in [Romer14].

## 2 Comparison With Existing Logos

The following picture shows the METAPOST and the METAFONT logos written in Fetamont (gray) and Taco Hoekwaters Logo (outlined).



---

<sup>1</sup>The T1-encoding is also known as the EC- or Cork-encoding

There are hardly any differences, only the «S» is significantly different, because D. E. Knuth has changed the shape of the «S» in 1997. Also the other faces of Taco Hoekwaters *Logo* are very similar to their corresponding Fetamont faces. Widths and kernings may rarely differ by one unit (except for the «A» in *Logo 9*, which has a strange width).

The comparison with the METATYPE1 logo from [Jackowski01] also shows virtually no differences.<sup>2</sup>

METATYPE1

The following picture compares *Fetamont Bold Condensed 40* with a traced version of the *Title Font* from `manfnt.mf`.

METAFONT

### 3 Naming Scheme For The Fetamont Faces

The file name of every face begins with the prefix `ffm`, which stands for «free typeface fetamont». The suffixes normally contain a symbol for the weight: `l` for light, `r` for regular, `b` for bold and `h` for heavy. The number at the end stands for the optical size (e. g. 10 pt). Depending on the face, the suffix is made of additional symbols:

Upright				Oblique			
r8	b8	h8		o8	bo8	ho8	
r9	b9	h9		o9	bo9	ho9	
l10	r10	b10	h10	lo10	o10	bo10	ho10
Condensed Upright				Condensed Oblique			
lc10	c10			lco10	co10		
		bc40				bco40	
Ultracondensed Upright				Ultracondensed Oblique			
lq10				lqo10			
Script Upright				Script Oblique			
lw10	w10	bw10	hw10	lwo10	wo10	bwo10	hwo10

Section 6 shows the font tables of all these faces.

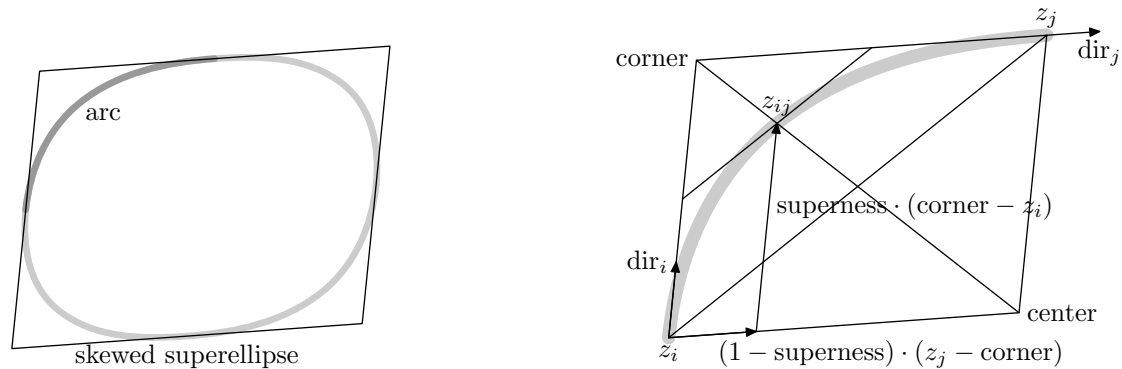
### 4 Special Techniques

Fetamont uses some special METAFONT techniques that are not well known (or have been unknown). The following subsections will document these techniques.

<sup>2</sup>I have never seen the original sources of the «Y» and the «1» but I think that my imitated «Y» and «1» are extremely close to the original.

## 4.1 Arc Constructions

Practically all curved paths in *Fetamont* are made out of so called *arcs*. An arc is kind of a quarter of a skewed superellipse. The skew is only needed, if the arcs have to look randomized like in the script style of fetamont.



In order to draw such an arc, the user defines the starting points  $z_i$ , the starting direction  $dir_i$ , the ending point  $z_j$ , the ending direction  $dir_j$  and a so called *superness*. The macro `arc( $z_i, dir_i, z_j, dir_j$ )` then defines the path as follows:

- Compute the point  $z_{ij}$ , which is at  $center + superness \cdot (corner - center)$  in vector terms. So if e. g.  $superness = 0.8$ ,  $z_{ij}$  is reached after travelling 80 % of the straight path from corner to center. One can see easily, that  $z_{ij}$  can also be computed by

$$z_{ij} = z_i + superness \cdot (corner - z_i) + (1 - superness) \cdot (z_j - corner)$$

- Now make a nice curve, that leaves  $z_i$  in the direction  $dir_i$ , passes  $z_{ij}$  in the direction  $z_j - z_i$  and ends in  $z_j$  heading for the direction  $dir_j$ .

Here is the METAFONT translation of this construction report:

```
vardef arc(expr zi,diri,zj,dirj) =
  zi{diri}...
  begingroup
    save corner,zij;
    pair corner,zij;
    corner=zi+whatever*diri=zj+whatever*dirj;
    zij=zi
      +superness*(corner-zi)
      +(1-superness)*(zj-corner);
    zij
  endgroup{zj-zi}
  ...zj{dirj}
enddef;
```

Everything inbetween `begingroup` and `endgroup` is just the computation of  $z_{ij}$ .

Note that Donald E. Knuth used a little different approach to draw randomized arcs for his «crazy shapes» of the Logo typeface.

## 4.2 Glyph names

Plain METAFONT automatically assigns well known letters like "A" with the corresponding encoding slot 65. But this does not work for letters like "Ä" (nor "Adieresis") as these letters will be placed in different encoding slots depending on the encoding. So these letters have to be declared directly by its encoding number (code). However, this will become problematic if one wants to change the encoding.

I solved this problem by a macro `enc` that uses very long conditionals to assign a unique code to each unicode name:

```
def enc(expr name)=
  if (font_coding_scheme_="T1"):
    if name="grave":
      0
    elseif name="acute":
      1
    elseif name="circumflex":
      2
    ...
    elseif name="germandbls":
      255
    else:
      errmessage("unknown name to encode");
    fi
  else:
    errmessage("tell me somewhere that the font_coding_scheme is T1");
  fi
enddef;
```

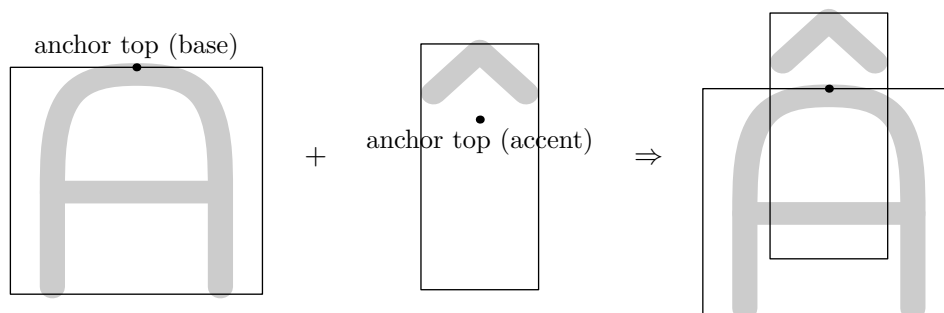
One may think that this is a very bad programming style and that a macro using arrays would be much more elegant. I agree! But then again I have found this to be the fastest solution.

With the `enc` macro one can treat «A» and «Ä» equally: `enc("A") = 65` and `enc("Adieresis") = 196`.

## 4.3 Combined Characters

To draw accented and other combined characters it is helpful to use *anchors*. The concept of anchors is quite usual in typesetting outside of the METAFONT world. However, anchors have been rarely seen in METAFONT up to now.

The idea is easy: Put an anchor at the base glyph and at the accent glyph and then overlay the two glyphs such that the anchors coincide.



Normally there are several kind of anchors needed. E. g. «Â» and «Ā» need two different anchors and so do «Ĺ» and «Ł». Fetamont needs three kind of anchors: «top», «topright» and «bot». So there are three arrays that can store the anchors:<sup>3</sup>

```
pair charanchortops_[];
pair charanchorbots_[];
pair charanchortoprights_[];
```

If one writes `charanchortops_[charcode]=(.5w,h)`; one stores a «top» anchor for the current glyph at the point  $(.5w, h)$ . Of course one needs more information, so there exist additional arrays:

```
numeric charwidths_[];
numeric charheights_[];
numeric chardepths_[];
numeric charitalcorrs_[];
picture charpictures_[];
```

The empty places in these arrays are always automatically filled in at the end of each character:

```
extra_endchar:=extra_endchar&"charpictures_[charcode]:=currentpicture;"
&"charwidths_[charcode]=charwd;"
&"charheights_[charcode]=charht;"
&"chardepths_[charcode]=chardp;"
&"charitalcorrs_[charcode]=charic;"
```

You can now combine two characters with the macro `ffmcombinedchar` which takes the following parameters:

namea	name of the base character
nameb	name of the accent character
namec	name of the new (combined) character
anchor	name of the anchor
height	new total height
depth	new total depth

---

<sup>3</sup>There is a naming convention that symbolic tokens ending in «\_» should not be used in high level programming.

The definition of the macro is now quite straight forward. The `code_offset` is needed, because the same constructions are used twice, as the lowercase letters are formed by small capitals.

```
def ffmcombinedchar(expr namea,nameb,namec,anchor,height,depth) =
  beginchar(enc(namea)+code_offset,
    charwidths_[enc(nameb)+code_offset],height,depth);
  charic:=charitalcorrs_[enc(nameb)+code_offset];
  addto currentpicture also charpictures_[enc(nameb)+code_offset];
  if anchor="top":
    addto currentpicture also charpictures_[enc(namec)] shifted
      ((charanchortops_[enc(nameb)+code_offset]
        -charanchortops_[enc(namec)]) slanted slant);
  elseif anchor="bot":
    addto currentpicture also charpictures_[enc(namec)] shifted
      ((charanchorbots_[enc(nameb)+code_offset]
        -charanchorbots_[enc(namec)]) slanted slant);
  elseif anchor="topright":
    addto currentpicture also charpictures_[enc(namec)] shifted
      ((charanchortoprightrights_[enc(nameb)+code_offset]
        -charanchortoprightrights_[enc(namec)]) slanted slant);
  else:
    errmessage "Wrong anchor name";
  fi
endchar;
enddef;
```

#### 4.4 Kerning Classes with METAFONT

Like anchor positioning, the concept of kerning classes is widely known but not frequently used in METAFONT. It is clear that “OT” needs the same kerning as “DT”. But be aware, “TO” needs a different kerning as “TD”! So there are two kind of kerning classes:

- *first kerning classes* group glyphs together that share the same shape to the right like “D” and “O”
- *second kerning classes* group glyphs together that share the same shape to the left like “C” and “O”

We define the arrays `kernclassesf_[] []` and `kernclassess_[] []` to store these informations:

```
numeric kernclassesf_[] [],
  kernclassess_[] [],
  ligmatrix_[] [] [];
```

The third array called `ligmatrix` will store all relevant kerning and ligature information. Now

```
addkernclassf("V","W");
addkernclasss("T","Tcaron","Tcedilla");
```

will group “V” and “W” to a first kerning class and “T”, “Tcaron” and “Tcedilla” to a second kerning class. The definitions of the macros `addkernclassf` and `addkernclasss` are analogous, they just deal with different arrays.

METAFONT has no straight way to determine the length of arrays or subarrays, so these lengths have to be stored somewhere. Thus, the zeroth row of the array consist of only one item: `kernclassesf_ [0] [0]` stores the number of rows (which corresponds the number of first kerningclasses). Each kerning class is stored in a row. The zeroth item of these rows is always the length of the row (which corresponds the number of glyphs in the kerning class).

```
def addkernclassf(text a) =
  kernclassesf_ [0] [0] := kernclassesf_ [0] [0] + 1; % number of kernclassesf
  begingroup
    save i;
    i := 0; % number of chars in current class
    for b = a:
      i := i + 1;
      kernclassesf_ [kernclassesf_ [0] [0]] [i] := enc(b);
    endfor
    % number of chars in current class is stored at 0th position
    kernclassesf_ [kernclassesf_ [0] [0]] [0] := i;
  endgroup
enddef;
```

The macros `addclasskern` and `addlig` will now add kerning information to kerning classes or add ligatures for single glyphs, respectively. These informations are stored in the `ligmatrix_ [] [] []`. In order to understand the definitions of the macros `addclasskern` and `addlig` it is important to know how this storage works:

For every glyph number of the encoding (from 0 to 255) the array `ligmatrix_ [] [] []` has a subarray reserved, so `ligmatrix_ [] [] []` consists of 256 rows. Each row contains the complete kerning and ligature data for the glyph whose encoding number equals the row number.

Let us say that the glyph number  $f = 102$  shall be kerned together with  $a = 97$  by the amount of  $-.5u\#$  and kerned together with  $t = 116$  by the amount of  $u\#$ . Furthermore  $f$  shall be combined with  $l = 108$  to the ligature  $fl = 29$ . So the 102th row will hold this information as follows:

$$\text{ligmatrix}_{102} = ( \underbrace{3}_{\text{length}}, \underbrace{(97, -.5u\#)}_{\text{kern with a}}, \underbrace{(116, u\#)}_{\text{kern with t}}, \underbrace{(-108, 29)}_{\text{ligature fl}} )$$



The minus flag before glyph numbers distincts ligatures from kernings.

At the beginning, the ligmatrix is empty, so each row has length 0 which is stored at the zeroth position of the rows:

```
for i=0 upto 255:
  ligmatrix_[i][0][0]:=0;
endfor
```

The call `addclasskern("f","a",-5u#)` will kern the first kerning class that contains «f» as first item and the second kerning class that contains «a» as first item by the amount of `-5u#`. The macro `addclasskern` writes the kerning information directly into the `ligmatrix_[] [] []` for all class members, the only problem is to find the indices of the kerning classes:

```
def addclasskern(expr first,second,kvalue) =
  begingroup
    save i,j,m,n;
    % get the indices i and j of the two classes:
    i:=0; % default value (cannot be true)
    j:=0; % default value (cannot be true)
    forever:
      i:=i+1;
      exitif kernclassesf_[i][1]=enc(first);
      if i>255:
        errmessage("unknown first kerning class");
      fi
    endfor
    forever:
      j:=j+1;
      exitif kernclassess_[j][1]=enc(second);
      if j>255:
        errmessage("unknown first kerning class");
      fi
    endfor
    for k=1 upto kernclassesf_[i][0]:
      m:=kernclassesf_[i][k]; % current first glyph
      for l=1 upto kernclassess_[j][0]:
        ligmatrix_[m][0][0]:=ligmatrix_[m][0][0]+1;
        n:=ligmatrix_[m][0][0]; % current last entry index (being written)
        ligmatrix_[m][n][0]:=kernclassess_[j][l];
        ligmatrix_[m][n][1]:=kvalue;
      endfor
    endfor
  endgroup
enddef;
```

The call `addlig("f","l","fl")` stores in the `ligmatrix_[] [] []` the instruction, that the combination of «f» and «l» shall be replaced by the «fl» ligature:

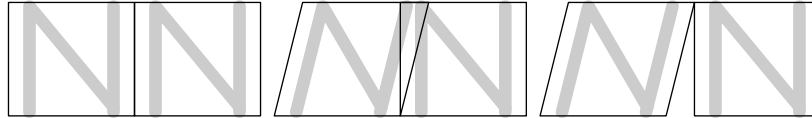
```
def addlig(expr first,second,third) =
  begingroup
    save i,n;
    i:=enc(first); % encoding number of first
    ligmatrix_[i][0][0]:=ligmatrix_[i][0][0]+1;
    n:=ligmatrix_[i][0][0]; % current last ligature entry index of i
    ligmatrix_[i][0][0]:=n;
    ligmatrix_[i][n][0]:=-enc(second); % minus is a flag for "ligature"
    ligmatrix_[i][n][1]:=enc(third);
  endgroup
enddef;
```

At the very end, the macro `writeligtable` writes all information from `ligmatrix_[] [] []` in a METAFONT friendly way:

```
def writeligtable = % write all kernings/ligatures at once
  begingroup
    save n;
    for i=0 upto 255: % current glyph i
      n:=ligmatrix_[i][0][0]; % number n of entries for glyph i
      if n<>0: % skip empty entries
        ligtable i:
          for j=1 upto n-1: %last entry needs a semicolon
            if ligmatrix_[i][j][0]<0: % the minus is a flag for "ligature"
              -ligmatrix_[i][j][0]:=ligmatrix_[i][j][1],
            else:
              ligmatrix_[i][j][0] kern ligmatrix_[i][j][1],
            fi
          endfor
          %last entry needs a semicolon:
          if ligmatrix_[i][n][0]<0: % the minus is a flag for "ligature"
            -ligmatrix_[i][n][0]:=ligmatrix_[i][n][1];
          else:
            ligmatrix_[i][n][0] kern ligmatrix_[i][n][1];
          fi
        fi
      endfor
    endgroup
  enddef;
```

## 4.5 Italic Corrections

Letter spacing is unproblematic if two upright letters are combined, like «NN». But if the first letter is italic, the letters will get too close (like «*NN*») and need additional space (like «*NN*»). This additional space is called *italic correction*.



D. E. Knuth has already defined an italic correction for the letter «T», because this is the last letter of the logos METAFONT and METAPOST. As for the *Computer Modern* typeface he found `italcorr ht#*slant+.5u#` to be a suitable italic correction. However, this is not a perfect idea because the italic correction should tend to 0 (and not `.5u#`) when the slant tends to 0. Hence, every character in Fetafont different to «T» has an italic correction proportional to the slant and the letter height. E. g. the letter «A» has an italic correction of `.8ht#*slant`.

## 4.6 Producing Outlines

The METAFONT sources have been converted to outline font formats like Type 1 or OpenType by a python script. This script calls METAPOST to produce PostScript files for each glyph. These glyphs are imported by the `fontforge` module. Khaled Hosny already used this technique in [Hosny11] to produce the outlines of *Punk Nova*. Because the glyph widths get lost by importing, also the `tfm` module from the `mftrace` project is needed (see [Nienhuys06]).

The following script contains the most important parts of the conversion.

```
#!/usr/bin/python

import os
import sys
import fontforge
import tfm # this is tfm.py from mftrace
import glob
import subprocess
import tempfile
import shutil

def usage():
    print "Example usage: %s mysource" % sys.argv[0]

if __name__ == "__main__":
    if len(sys.argv) < 2:
        usage()
```

```

        sys.exit()

    print "Creating_font_file..."
    style = sys.argv[1]
    designsize = 10
    fontname = sys.argv[1]
    font      = fontforge.font()

    print "Setting_general_font_information..."
    fontforge.loadEncodingFile("t1.enc")
    font.encoding="T1Encoding"

    print "Running_METAPOST_for_tfm_and_glyphs_definition..."
    mffile = os.path.abspath("%s" % fontname)
    tempdir = tempfile.mkdtemp()
    magnification = 1003.75/designsize
    subprocess.call(
        ['mpost',
         '&mfplain',
         '\mode=localfont;',
         'mag:=%s;' % magnification,
         'outputtemplate:="%c.eps";',
         'input_%s;' % mffile,
         'bye'],
        stdout=subprocess.PIPE, stderr=subprocess.PIPE,
        cwd=tempdir,
    )

    print "Importing_glyphs..."
    glyph_files = glob.glob(os.path.join(tempdir, "*.eps"))
    for file in glyph_files:
        code = int(os.path.splitext(os.path.basename(file))[0])
        glyph = font.createMappedChar(code)
        glyph.importOutlines(file, ("toobigwarn", "correctdir"))

    print "Adding_metrics..."
    metric = tfm.read_tfm_file ("%s/%s.tfm" % (tempdir, fontname) )
    for glyph in font.glyphs():
        metric_width = metric.get_char(glyph.encoding).width
        glyph.width = int (round (metric_width / designsize * 1000))
    font.mergeFeature("%s/%s.tfm" % (tempdir, fontname))
    shutil.copyfile("%s/%s.tfm" % (tempdir, fontname), "%s.tfm" % fontname)

    shutil.rmtree(tempdir)

```

```

print "Add␣space␣for␣non-TeX..."
normal_space = font[32].width ##take width from visible space
font.encoding = "unicode"
font.createChar(32)
font[32].width = normal_space ##space
font.encoding = "T1Encoding"
font.encoding = "compactd"

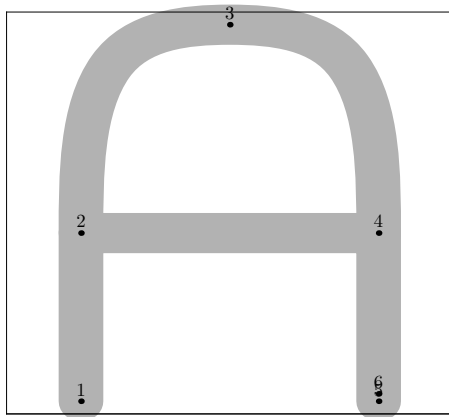
print "Finetuning..."
font.selection.all()
font.addExtrema()
font.removeOverlap()
font.simplify()
font.round()
font.simplify()
font.autoHint()

print "Saving␣sfd-file␣'%s'..." % fontname
font.save("%s.sfd" % fontname)
print "Generating␣otf-file␣'%s'..." % fontname
font.generate("%s.otf" % fontname)

```

## 5 Construction Of The Most Important Letters

### The Letter A



```

fmchar("A", 15, ht#, 0);
italcorr .8ht# * slant;
x1 = leftstemloc + noise;
x2 = leftstemloc + noise;

```

```

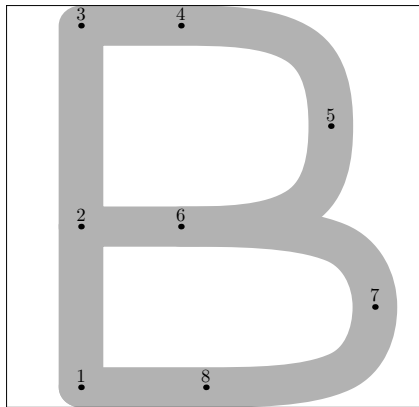
 $x_3 = .5w + noise;$ 
 $w - x_4 = leftstemloc + noise;$ 
 $w - x_5 = leftstemloc + noise;$ 
 $bot\ y_1 = noise - o;$ 
 $y_2 = barheight + noise;$ 
 $top\ y_3 = h + o;$ 
 $y_4 = barheight + noise;$ 
 $bot\ y_5 = noise - o;$ 
 $bot\ y_6 = 0;$ 
 $z_6 = whatever[z_4, z_5];$ 
draw  $z_1 - z_2 - z_4 - z_5;$ 
draw  $half(z_2, z_2 - z_1, z_3, randrt, z_4, z_5 - z_4);$ 
 $charanchor tops\_ [charcode] = (.5w, h);$ 
 $charanchor bots\_ [charcode] = z_6;$ 
labels(1, 2, 3, 4, 5, 6);
endchar;

```

%no noise because of Aring

%no noise

## The Letter *B*



```

ffmchar("B", 14,  $ht\#$ , 0);
italcorr  $ht\# * slant;$ 
 $x_1 = leftstemloc + noise;$ 
 $x_3 = leftstemloc + noise;$ 
 $x_7 = .5[w - x_1, lft\ w] + noise;$ 
 $x_5 = .85[x_1, x_7] + noise;$ 
 $x_8 = .5[x_1, x_5] + noise;$ 
 $x_4 = .4[x_1, x_5] + noise;$ 
 $x_6 = .4[x_1, x_5] + noise;$ 
 $bot\ y_1 = noise;$ 
 $y_2 = barheight + noise;$ 
 $top\ y_3 = h + noise;$ 

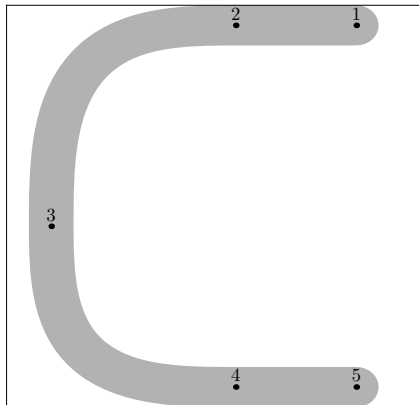
```

```

y4 = y3 + noise;
y6 = y2 + noise;
bot y8 = noise;
y5 = .5[y4, y6] + noise;
y7 = .5[y6, y8] + noise;
z2 = whatever[z1, z3];
z9 = whatever[z2, z6];
draw z1 - z3 - z4
& half(z4, z4 - z3, z5, -randup, z6, z2 - z6)
& z6 - z2;
draw half(z6, z6 - z2, z7, -randup, z8, z1 - z8)
& z8 - z1;
labels(1, 2, 3, 4, 5, 6, 7, 8);
endchar;

```

## The Letter C



```

fmchar("C", 14, ht#, 0);
italcorr ht# * slant;
x1 = w - leftstemloc + ho + noise;
x2 = .55w + noise;
x3 = good.x(1.5u + s + noise);
x4 = .55w + noise;
x5 = w - leftstemloc + ho + noise;
top y1 = h + noise;
top y2 = h + noise;
y3 = barheight + noise;
bot y4 = 0;
z5 = z4 + whatever * randrt;
draw z1 - z2
& half(z2, z2 - z1, z3, -randup, z4, z5 - z4)

```

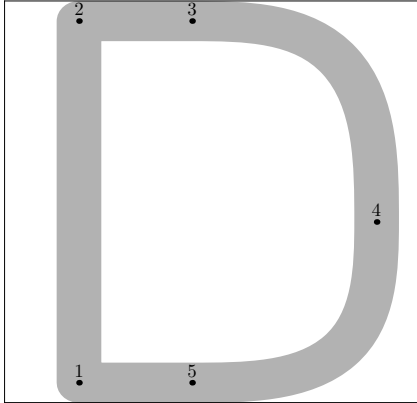
%no noise!

```

& z4 - z5;
charanchortops__[charcode] = (.5w, h);
charanchorbots__[charcode] = z4;
labels(1, 2, 3, 4, 5);
endchar;

```

## The Letter *D*



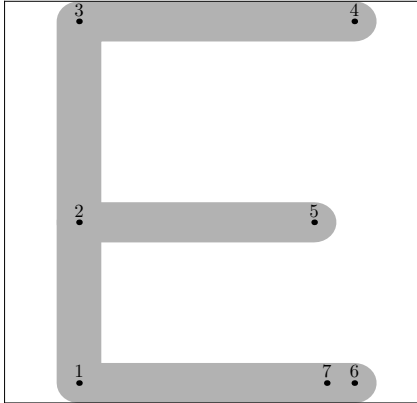
```

ffmchar("D", 14, ht#, 0);
italcorr .9ht# * slant;
x1 = leftstemloc + noise;
x2 = leftstemloc + noise;
x3 = .45w + noise;
x5 = .45w + noise;
w - x4 = good.x(1.5u + s + noise);
bot y1 = noise;
bot y5 = noise;
top y2 = h + noise;
top y3 = h + noise;
y4 = barheight + noise;
draw z1 - z2 - z3
& half(z3, z3 - z2, z4, -randup, z5, z1 - z5)
& z5 - cycle;
charanchortops__[charcode] = (.5w, h);
labels(1, 2, 3, 4, 5);
endchar;

```



## The Letter *E*

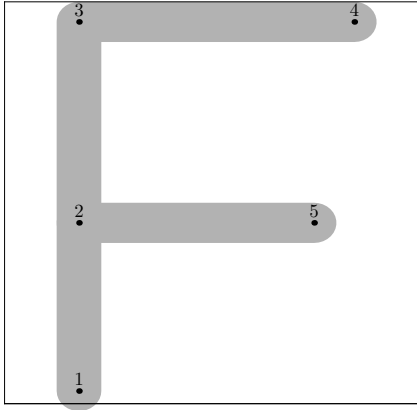


```

ffmchar("E", 14, ht#, 0);
italcorr .9ht# * slant;
x1 = leftstemloc + noise;
x3 = leftstemloc + noise;
x4 = w - leftstemloc + ho + noise;
x5 = w - leftstemloc + ho - xgap + noise;
x6 = w - leftstemloc + ho + noise;
x7 = .9[x1, x6];
bot y1 = noise;
y2 = barheight + noise;
top y3 = h + noise;
top y4 = h + noise;
y5 = barheight + noise;
bot y7 = 0;
z2 = whatever[z1, z3];
z6 = whatever[z1, z7];
draw z6 - z1 - z3 - z4;
draw z2 - z5;
charanchortops_[charcode] = (.5[leftstemloc, w - leftstemloc + o], h);
charanchorbots_[charcode] = z7;
labels(1, 2, 3, 4, 5, 6, 7);
endchar;

```

## The Letter *F*

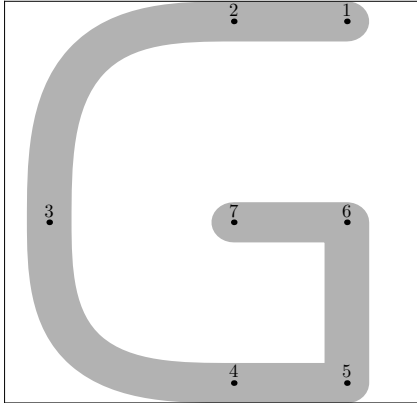


```

ffmchar("F", 14, ht#, 0);
italcorr .9ht# * slant;
x1 = leftstemloc + noise;
x3 = leftstemloc + noise;
x4 = w - leftstemloc + ho + noise;
x5 = w - leftstemloc + ho - xgap + noise;
bot y1 = noise - o;
y2 = barheight + noise;
top y3 = h + noise;
top y4 = h + noise;
y5 = barheight + noise;
bot y6 = noise;
z2 = whatever[z1, z3];
draw z1 - z3 - z4;
draw z2 - z5;
labels(1, 2, 3, 4, 5);
endchar;

```

## The Letter G

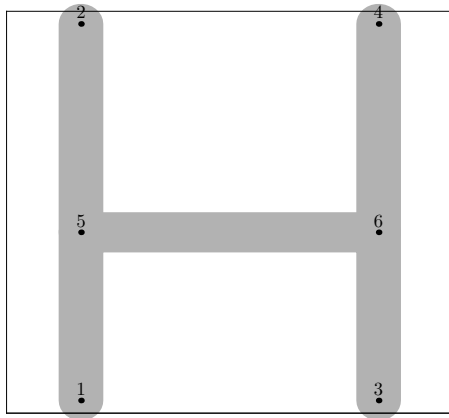


```

ffmchar("G", 14, ht#, 0);
italcorr ht# * slant;
x3 = good.x(1.5u + s + noise);
x1 = w - leftstemloc + noise;
x5 = w - leftstemloc + noise;
x6 = w - leftstemloc + noise;
x2 = .55w + noise;
x4 = .55w + noise;
x7 = .55w + noise;
y3 = barheight + noise;
y6 = barheight + noise;
y7 = barheight + noise;
top y2 = h + noise;
top y1 = h + noise;
bot y4 = noise;
bot y5 = noise;
draw z1 - z2
& half(z2, z2 - z1, z3, -randup, z4, z5 - z4)
& z4 - z5 - z6 - z7;
charanchortops_[charcode] = (.5w, h);
labels(1, 2, 3, 4, 5, 6, 7);
endchar;

```

## The Letter *H*

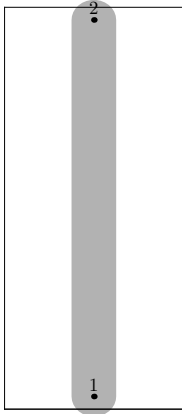


```

ffmchar("H", 15, ht#, 0);
italcorr .8ht# * slant;
x1 = leftstemloc + noise;
x2 = leftstemloc + noise;
w - x3 = leftstemloc + noise;
w - x4 = leftstemloc + noise;
bot y1 = noise - o;
top y2 = h + o + noise;
bot y3 = noise - o;
top y4 = h + o + noise;
y5 = barheight + noise;
y6 = barheight + noise;
z5 = whatever[z1, z2];
z6 = whatever[z3, z4];
draw z1 - z5 - z6 - z3;
draw z5 - z2;
draw z6 - z4;
labels(1, 2, 3, 4, 5, 6);
endchar;

```

## The Letter *I*

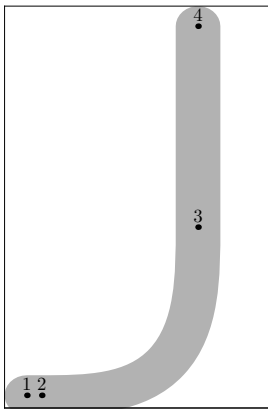


```

ffmchar("I", 6,  $ht\#$ , 0);
italcorr  $.8ht\# * slant$ ;
 $x_1 = .5w + noise$ ;
 $x_2 = .5w + noise$ ;
 $bot\ y_1 = noise - o$ ;
 $top\ y_2 = h + o + noise$ ;
draw  $z_1 - z_2$ ;
 $charanchor\ tops\_ [charcode] = (.5w + noise, h)$ ;
labels(1, 2);
endchar;

```

## The Letter *J*



```

ffmchar("J", 9,  $ht\#$ , 0);
italcorr  $.8ht\# * slant$ ;
 $lft\ x_1 = noise - eps$ ;
 $x_2 = x_1 + .5u$ ;

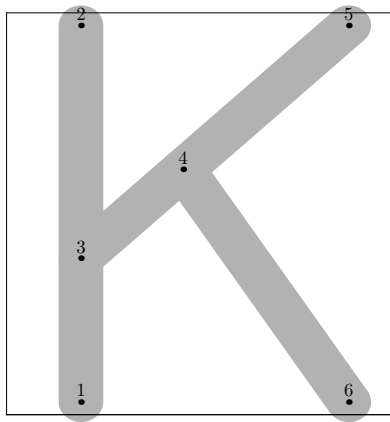
```

```

 $w - x_3 = \text{leftstemloc} + \text{noise};$ 
 $w - x_4 = \text{leftstemloc} + \text{noise};$ 
 $\text{bot } y_1 = \text{noise} - o;$ 
 $y_3 = \text{barheight} + \text{noise};$ 
 $\text{top } y_4 = h + \text{noise};$ 
 $z_2 = z_1 + \text{whatever} * \text{randrt};$ 
draw  $z_1 - z_2$ 
 $\& \text{arc}(z_2, \text{randrt}, z_3, z_4 - z_3)$ 
 $\& z_3 - z_4;$ 
labels(1, 2, 3, 4);
endchar;

```

### The Letter *K*



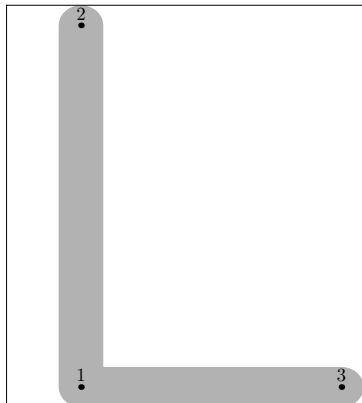
```

ffmchar("K", 13,  $ht\#$ , 0);
italcorr  $ht\# * \text{slant};$ 
 $x_1 = \text{leftstemloc} + \text{noise};$ 
 $x_2 = \text{leftstemloc} + \text{noise};$ 
 $w - x_5 = \text{good}.x(1.5u + s + \text{noise});$ 
 $w - x_6 = \text{good}.x(1.5u + s + \text{noise});$ 
 $\text{bot } y_1 = \text{noise} - o;$ 
 $\text{bot } y_6 = \text{noise} - o;$ 
 $\text{top } y_2 = h + o + \text{noise};$ 
 $\text{top } y_5 = h + o + \text{noise};$ 
 $y_3 = .618[y_2, y_1] + \text{noise};$ 
 $z_3 = \text{whatever}[z_1, z_2];$ 
 $z_4 = \text{whatever}[z_3, z_5] = \text{whatever}[z_2, z_6];$ 
draw  $z_1 - z_2;$ 
draw  $z_3 - z_5;$ 
draw  $z_4 - z_6;$ 
labels(1, 2, 3, 4, 5, 6);

```

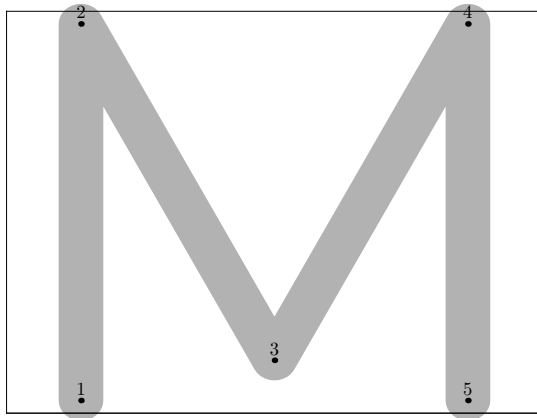
```
endchar;
```

## The Letter *L*



```
ffmchar("L", 12, ht#, 0);
x1 = leftstemloc + noise;
x2 = leftstemloc + noise;
rt x3 = w - eps + noise;
bot y1 = noise;
bot y3 = noise;
top y2 = h + noise;
draw z3 - z1 - z2;
charanchortops_[charcode] = (leftstemloc, h);
charanchortoprights_[charcode] = (.618w, h);
labels(1, 2, 3);
endchar;
```

## The Letter *M*

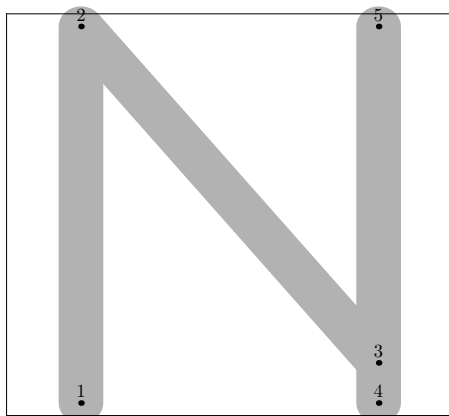


```

ffmchar("M", 18, ht#, 0);
italcorr ht# * slant;
 $x_1 = \text{leftstemloc} + \text{noise};$ 
 $x_2 = \text{leftstemloc} + \text{noise};$ 
 $x_3 = .5w + \text{noise};$ 
 $x_4 = w - \text{leftstemloc} + \text{noise};$ 
 $x_5 = w - \text{leftstemloc} + \text{noise};$ 
 $\text{bot } y_1 = \text{noise} - o;$ 
 $\text{top } y_2 = h + o + \text{noise};$ 
 $\text{bot } y_3 = \text{ygap} - o + \text{noise};$ 
 $\text{top } y_4 = h + o + \text{noise};$ 
 $\text{bot } y_5 = \text{noise} - o;$ 
draw  $z_1 - z_2 - z_3 - z_4 - z_5;$ 
labels(1, 2, 3, 4, 5);
endchar;

```

## The Letter *N*



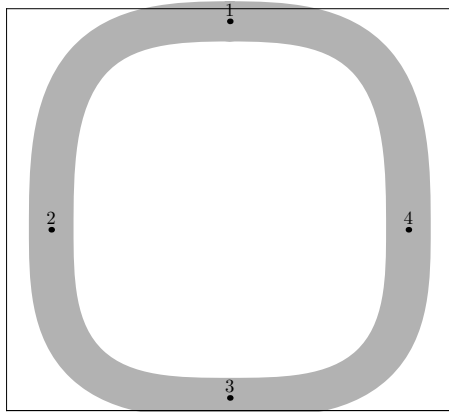


```

fmchar("N", 15, ht#, 0);
italcorr .8ht# * slant;
 $x_1 = \text{leftstemloc} + \text{noise};$ 
 $x_2 = \text{leftstemloc} + \text{noise};$ 
 $x_4 = w - \text{leftstemloc} + \text{noise};$ 
 $x_5 = w - \text{leftstemloc} + \text{noise};$ 
 $\text{bot } y_1 = \text{noise} - o;$ 
 $\text{top } y_2 = h + o + \text{noise};$ 
 $y_3 = y_4 + \text{ygap} + \text{noise};$ 
 $\text{bot } y_4 = \text{noise} - o;$ 
 $\text{top } y_5 = h + o + \text{noise};$ 
 $z_3 = \text{whatever}[z_4, z_5];$ 
draw  $z_1 - z_2 - z_3;$ 
draw  $z_4 - z_5;$ 
 $\text{charanchortops\_}[\text{charcode}] = (.5w, h);$ 
labels(1, 2, 3, 4, 5);
endchar;

```

## The Letter O



```

fmchar("O", 15, ht#, 0);
italcorr .8ht# * slant;
 $x_1 = .5w + \text{noise};$ 
 $x_2 = \text{good}.x(1.5u + s + \text{noise});$ 
 $x_3 = .5w + \text{noise};$ 
 $w - x_4 = \text{good}.x(1.5u + s + \text{noise});$ 
 $\text{top } y_1 = h + o + \text{noise};$ 
 $y_2 = \text{barheight} + \text{noise};$ 
 $\text{bot } y_3 = \text{noise} - o;$ 
 $y_4 = \text{barheight} + \text{noise};$ 
draw full( $z_1, -\text{randrt}, z_2, -\text{randup}, z_3, \text{randrt}, z_4, \text{randup}$ );

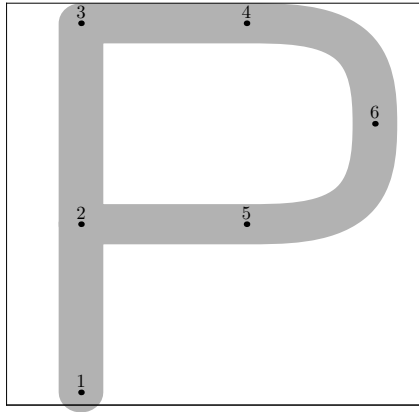
```

```

charanchortops_[charcode] = (.5w, h);
labels(1, 2, 3, 4);
endchar;

```

## The Letter *P*

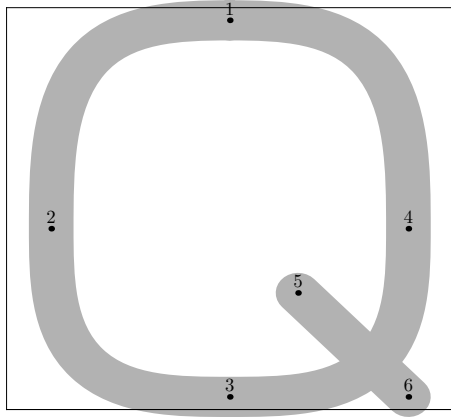


```

fmchar("P", 14, ht#, 0);
italcorr .8ht# * slant;
x1 = leftstemloc + noise;
x3 = leftstemloc + noise;
x4 = .618[x1, w - x1] + noise;
x5 = .618[x1, w - x1] + noise;
x6 = .5[w - x1, lft w] + noise;
y2 = barheight + noise;
y5 = barheight + noise;
bot y1 = noise - o;
top y3 = h + noise;
top y4 = h + noise;
y6 = .5[y4, y5] + noise;
z2 = whatever[z1, z3];
draw z1 - z3 - z4
& half(z4, z4 - z3, z6, -randup, z5, z2 - z5)
& z5 - z2;
labels(1, 2, 3, 4, 5, 6);
endchar;

```

## The Letter Q

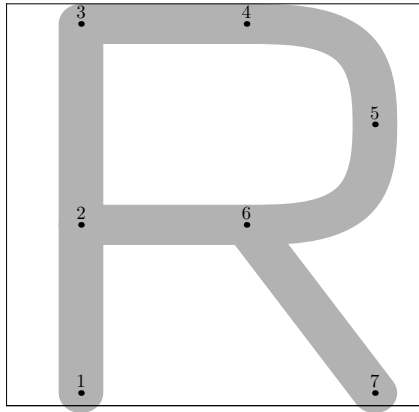


```

ffmchar("Q", 15, ht#, 0);
italcorr .8ht# * slant;
x1 = .5w + noise;
x2 = good.x(1.5u + s + noise);
x3 = .5w + noise;
w - x4 = good.x(1.5u + s + noise);
x5 = min(.618[x4, x3], x4 - 2px) + noise;
w - x6 = good.x(1.5u + s + noise);
top y1 = h + o + noise;
y2 = barheight + noise;
bot y3 = noise - o;
y4 = barheight + noise;
y5 = .618[y3, y4] + noise;
bot y6 = noise - o;
%z5=z6+whatever*dir(100+angle(direction 1 of (arc(z3,randrt,z4,randup))));
draw full(z1, -randrt, z2, -randup, z3, randrt, z4, randup);
draw z5 - z6;
labels(1, 2, 3, 4, 5, 6);
endchar;

```

## The Letter *R*

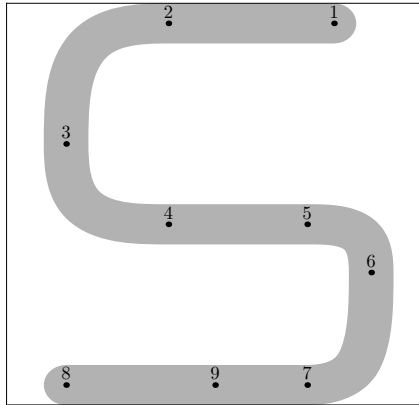


```

ffmchar("R", 14, ht#, 0);
italcorr .8ht# * slant;
x1 = leftstemloc + noise;
x3 = leftstemloc + noise;
x4 = .618[x1, w - x1] + noise;
x6 = .618[x1, w - x1] + noise;
x5 = .5[w - x1, lft w] + noise;
x7 = .5[w - x1, lft w] + noise;
y2 = barheight + noise;
y6 = barheight + noise;
bot y1 = noise - o;
bot y7 = noise - o;
top y3 = h + noise;
top y4 = h + noise;
y5 = .5[y4, y6] + noise;
z2 = whatever[z1, z3];
draw z1 - z3 - z4
& half(z4, z4 - z3, z5, -randup, z6, z2 - z6)
& z6 - z2;
draw z6 - z7;
charanchortops_[charcode] = (.5w, h);
labels(1, 2, 3, 4, 5, 6, 7);
endchar;

```

## The Letter S

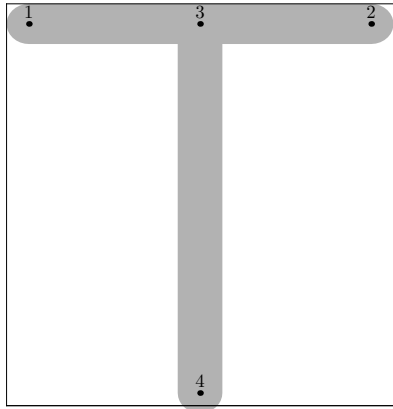


```

ffmchar("S", 14, ht#, 0);
italcorr .8ht# * slant;
 $x_3 = \text{good}.x(2u + s + \text{noise});$ 
 $x_8 = \text{good}.x(2u + s + \text{noise});$ 
 $w - x_1 = \text{good}.x(3u + s + \text{noise});$ 
 $x_2 = .382[x_3, x_1] + \text{noise};$ 
 $x_4 = .382[x_3, x_1] + \text{noise};$ 
 $x_5 = .9[x_3, x_1] + \text{noise};$ 
 $x_7 = .9[x_3, x_1] + \text{noise};$ 
 $w - x_6 = \text{good}.x(1.75u + .5s + \text{noise});$ 
 $x_9 = .618[x_8, x_7];$ 
 $\text{top } y_1 = h + \text{noise};$ 
 $\text{top } y_2 = h + \text{noise};$ 
 $y_3 = .6[y_2, y_4] + \text{noise};$ 
 $y_4 = \text{barheight} + \text{noise};$ 
 $y_5 = \text{barheight} + \text{noise};$ 
 $y_6 = .3[y_5, y_7] + \text{noise};$ 
 $\text{bot } y_9 = 0;$ 
 $z_7 = z_9 + \text{whatever} * \text{randrt};$ 
 $z_8 = \text{whatever}[z_7, z_9];$ 
draw  $z_1 - z_2$ 
& half( $z_2, z_2 - z_1, z_3, -\text{randup}, z_4, z_5 - z_4$ )
&  $z_4 - z_5$ 
& half( $z_5, z_5 - z_4, z_6, -\text{randup}, z_7, z_8 - z_7$ )
&  $z_7 - z_8;$ 
 $\text{charanchortops\_}[charcode] = (.5w, h);$ 
 $\text{charanchorbots\_}[charcode] = z_9;$ 
labels(1, 2, 3, 4, 5, 6, 7, 8, 9);
endchar;

```

## The Letter *T*

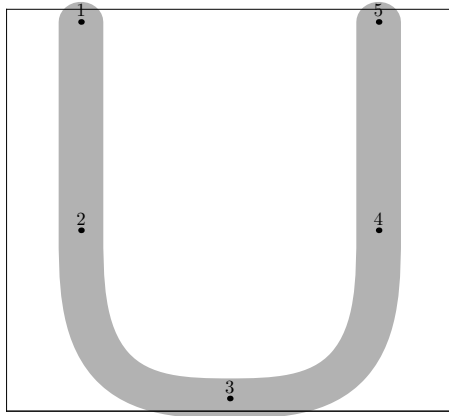


```

ffmchar("T", 13, ht#, 0);
italcorr ht# * slant + .5u#;
if .5w ≠ good.x .5w: change_width; fi
lft x1 = noise − eps;
rt x2 = w + noise;
x3 = .5w + noise;
x4 = .5w + noise;
top y1 = h + noise;
top y2 = h + noise;
bot y4 = noise − o;
z3 = whatever[z1, z2];
draw z1 − z2;
draw z3 − z4;
charanchortops_[charcode] = (.5w, h);
charanchorbots_[charcode] = (x4, 0);
labels(1, 2, 3, 4);
endchar;

```

## The Letter *U*

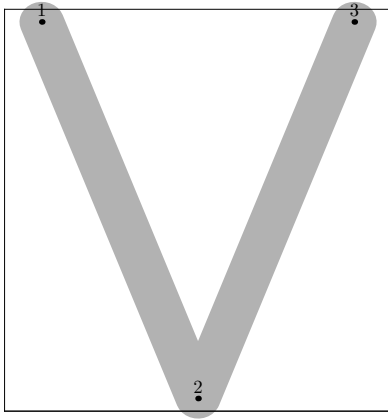


```

ffmchar("U", 15, ht#, 0);
italcorr ht# * slant;
 $x_1 = \text{leftstemloc} + \text{noise};$ 
 $x_2 = \text{leftstemloc} + \text{noise};$ 
 $w - x_4 = \text{leftstemloc} + \text{noise};$ 
 $w - x_5 = \text{leftstemloc} + \text{noise};$ 
 $x_3 = .5[x_1, x_4] + \text{noise};$ 
 $\text{top } y_1 = h + o + \text{noise};$ 
 $y_2 = \text{barheight} + \text{noise};$ 
 $\text{bot } y_3 = \text{noise} - o;$ 
 $y_4 = \text{barheight} + \text{noise};$ 
 $\text{top } y_5 = h + o + \text{noise};$ 
draw  $z_1 - z_2$ 
& half( $z_2, z_2 - z_1, z_3, \text{randrt}, z_4, z_5 - z_4$ )
&  $z_4 - z_5;$ 
 $\text{charanchortops\_}[\text{charcode}] = (x_3, h);$ 
labels(1, 2, 3, 4, 5);
endchar;

```

## The Letter V

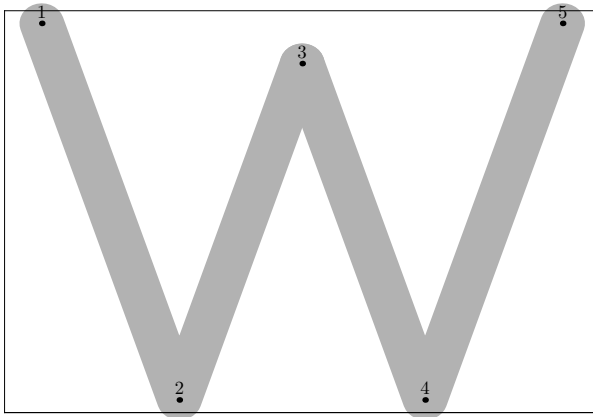


```

ffmchar("V", 13, ht#, 0);
italcorr ht# * slant;
 $x_1 = \text{good}.x(1.5u + s + \text{noise}) - ho;$ 
 $w - x_3 = \text{good}.x(1.5u + s + \text{noise}) - ho;$ 
 $x_2 = .5[x_1, x_3] + \text{noise};$ 
 $top\ y_1 = h + o + \text{noise};$ 
 $bot\ y_2 = \text{noise} - o;$ 
 $top\ y_3 = h + o + \text{noise};$ 
draw  $z_1 - z_2 - z_3;$ 
labels(1, 2, 3);
endchar;

```

## The Letter W



```

ffmchar("W", 20, ht#, 0);
italcorr ht# * slant;
 $x_1 = \text{good}.x(1.5u + s + \text{noise}) - ho;$ 

```

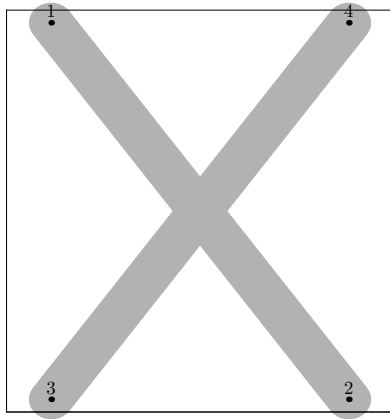


```

 $w - x_5 = \text{good}.x(1.5u + s + \text{noise}) - ho;$ 
 $x_3 = .5[x_1, x_5] + \text{noise};$ 
 $\text{top } y_1 = h + o + \text{noise};$ 
 $\text{bot } y_2 = \text{noise} - o;$ 
 $y_3 = y_1 - ygap + \text{noise};$ 
 $\text{bot } y_4 = \text{noise} - o;$ 
 $\text{top } y_5 = h + o + \text{noise};$ 
 $z_4 = z_5 + \text{whatever} * (x_5 - x_1, 4 * (y_1 - y_2) - 2ygap);$ 
 $z_2 = z_1 + \text{whatever} * (x_1 - x_5, 4 * (y_1 - y_2) - 2ygap);$ 
draw  $z_1 - z_2 - z_3;$ 
draw  $z_3 - z_4 - z_5;$ 
labels(1, 2, 3, 4, 5);
endchar;

```

### The Letter X

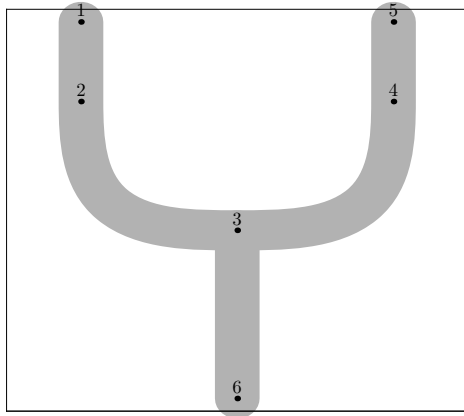


```

ffmchar("X", 13,  $ht\#$ , 0);
italcorr  $ht\# * \text{slant}$ ;
 $x_1 = \text{good}.x(1.5u + s + \text{noise});$ 
 $x_3 = \text{good}.x(1.5u + s + \text{noise});$ 
 $w - x_2 = \text{good}.x(1.5u + s + \text{noise});$ 
 $w - x_4 = \text{good}.x(1.5u + s + \text{noise});$ 
 $\text{top } y_1 = h + o + \text{noise};$ 
 $\text{top } y_4 = h + o + \text{noise};$ 
 $\text{bot } y_3 = \text{noise} - o;$ 
 $\text{bot } y_2 = \text{noise} - o;$ 
draw  $z_1 - z_2;$ 
draw  $z_3 - z_4;$ 
labels(1, 2, 3, 4);
endchar;

```

## The Letter Y

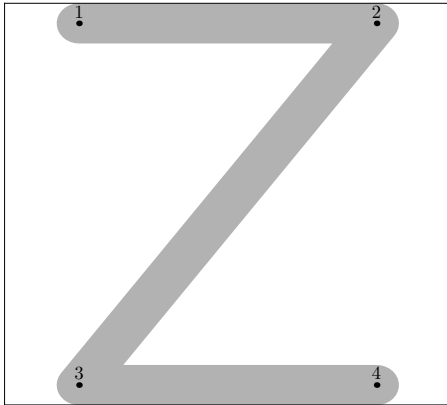


```

ffmchar("Y", 15.5, ht#, 0);
italcorr ht# * slant;
x1 = leftstemloc + noise;
x2 = leftstemloc + .5noise;
w - x4 = leftstemloc + .5noise;
w - x5 = leftstemloc + noise;
x3 = .5[x1, x5] + noise;
x6 = .5[x1, x5] + noise;
top y1 = h + o + noise;
top y5 = h + o + noise;
bot y6 = noise - o;
y3 = barheight + noise;
y2 = .618[y3, y5] + noise;
y4 = .618[y3, y5] + noise;
draw z1 - z2
& half(z2, z2 - z1, z3, randrt, z4, z5 - z4)
& z4 - z5;
draw z6 - z3;
charanchortops_[charcode] = (.5w, h);
labels(1, 2, 3, 4, 5, 6);
endchar;

```

## The Letter Z

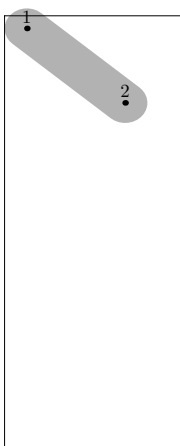


```

ffmchar("Z", 15, ht#, 0);
italcorr ht# * slant;
 $x_1 = \text{leftstemloc} + \text{noise};$ 
 $w - x_2 = \text{leftstemloc} + \text{noise};$ 
 $x_3 = \text{leftstemloc} + \text{noise};$ 
 $w - x_4 = \text{leftstemloc} + \text{noise};$ 
 $\text{top } y_1 = h + \text{noise};$ 
 $\text{top } y_2 = h + \text{noise};$ 
 $\text{bot } y_3 = \text{noise};$ 
 $\text{bot } y_4 = \text{noise};$ 
draw  $z_1 - z_2 - z_3 - z_4;$ 
 $\text{charanchortops\_}[charcode] = (.5w, h);$ 
labels(1, 2, 3, 4);
endchar;

```

## The Letter *grave*

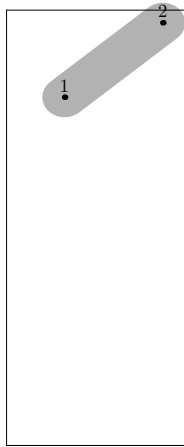


```

fmchar("grave", 6,  $x_{ht\#} + acc_{ht\#}$ , 0);
 $lft\ x_1 = noise$ ;
 $rt\ x_2 = .8w + noise$ ;
 $top\ y_1 = h + o + noise$ ;
 $bot\ y_2 = .2[x_{ht}, h] + noise$ ;
draw  $z_1 - z_2$ ;
 $charanchor_{tops\_}[charcode] = (.5w, x_{ht})$ ;
labels(1, 2);
endchar;

```

### The Letter *acute*

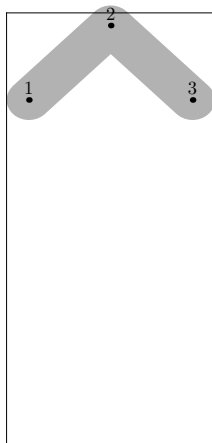


```

fmchar("acute", 6,  $x_{ht\#} + acc_{ht\#}$ , 0);
 $lft\ x_1 = .2w + noise$ ;
 $rt\ x_2 = w + noise$ ;
 $bot\ y_1 = .2[x_{ht}, h] + noise$ ;
 $top\ y_2 = h + o + noise$ ;
draw  $z_1 - z_2$ ;
 $charanchor_{tops\_}[charcode] = (.5w, x_{ht})$ ;
labels(1, 2);
endchar;

```

## The Letter *circumflex*

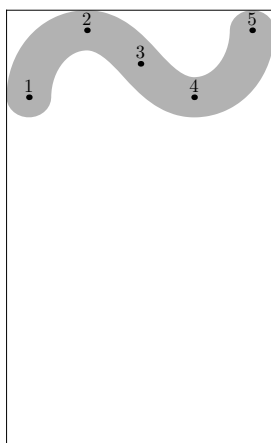


```

ffmchar("circumflex",7,x_ht# + acc_ht#,0);
lft x1 = noise;
x2 = .5w + noise;
rt x3 = w + noise;
bot y1 = .2[x_ht,h] + noise;
bot y3 = .2[x_ht,h] + noise;
top y2 = h + o + noise;
draw z1 - z2 - z3;
charanchortops_[charcode] = (.5w,x_ht);
labels(1,2,3);
endchar;

```

## The Letter *tilde*



```

ffmchar("tilde",9,x_ht# + acc_ht#,0);
lft x1 = eps + noise;

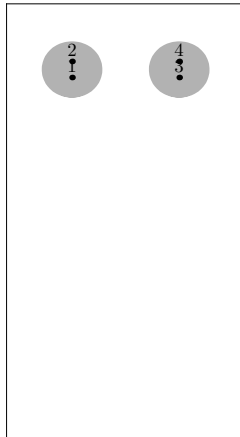
```

```

 $x_2 = .3w + noise;$ 
 $x_4 = .7w + noise;$ 
 $x_3 = .5w + noise;$ 
 $rt\ x_5 = w - eps + noise;$ 
 $bot\ y_1 = .2[x\_ht, h] + noise;$ 
 $bot\ y_4 = .2[x\_ht, h] + noise;$ 
 $top\ y_2 = h + noise;$ 
 $top\ y_5 = h + noise;$ 
 $y_3 = .6[x\_ht, h] + noise;$ 
if angledirection1of( $z_2\{right\} \dots z_3 \dots z_4\{right\}$ ) < -90:
draw  $z_1\{randup\} \dots z_2\{randrt\} \dots z_3\{-randup\} \dots z_4\{randrt\} \dots z_5\{randup\};$ 
else:
draw  $z_1\{randup\} \dots z_2\{randrt\} \dots z_3 \dots z_4\{randrt\} \dots z_5\{randup\};$ 
fi
charanchortops_ $[charcode] = (.5w, x\_ht);$ 
labels(1, 2, 3, 4, 5);
endchar;

```

### The Letter *dieresis*



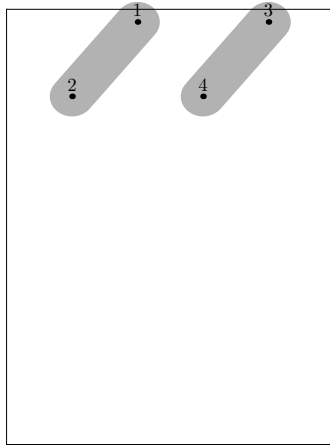
```

ffmchar("dieresis", 8,  $x\_ht\# + acc\_ht\#, 0$ );
 $x_1 = x_2 = .5w - \max(1.8u, (1 + dotincr) * .7px) + noise;$ 
 $x_3 = x_4 = .5w + \max(1.8u, (1 + dotincr) * .7px) + noise;$ 
 $bot\ y_1 = .3[x\_ht, h] + noise;$ 
 $y_2 = y_1 + dotincr * py;$ 
 $bot\ y_3 = .3[x\_ht, h] + noise;$ 
 $y_4 = y_3 + dotincr * py;$ 
draw dotcircle( $z_1, z_2$ );
draw dotcircle( $z_3, z_4$ );
charanchortops_ $[charcode] = (.5w, x\_ht);$ 

```

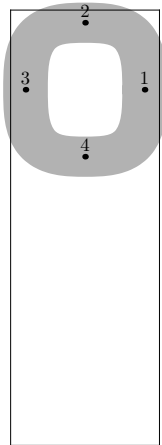
```
labels(1, 2, 3, 4);
endchar;
```

### The Letter *hungarumlaut*



```
ffmchar("hungarumlaut", 11, x_ht# + acc_ht#, 0);
x_2 = .2w + noise;
x_3 = .8w + noise;
x_1 = .4w + noise;
x_4 = .6w + noise;
top y_1 = h + o + noise;
top y_3 = h + o + noise;
bot y_2 = .2[x_ht, h] + noise;
bot y_4 = .2[x_ht, h] + noise;
draw z_1 - z_2;
draw z_3 - z_4;
charanchortops_[charcode] = (.4w, x_ht);
labels(1, 2, 3, 4);
endchar;
```

## The Letter *ring*

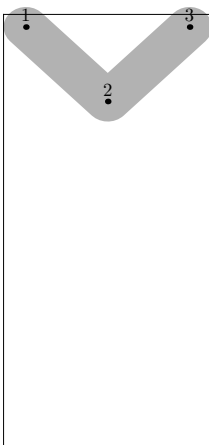


```

ffmchar("ring",5,x_ht# + acc_ht#,0);
lft x3 = -o + noise;
rt x1 = w + o + noise;
x2 = .5w + noise;
x4 = .5w + noise;
top y4 = x_ht + o;                                     %no noise because of Aring
top y2 = h + o + noise;
y1 = .5[y2,y4] + noise;
y3 = .5[y2,y4] + noise;
draw full(z1, randup, z2, -randrt, z3, -randup, z4, randrt);
charanchortops_[charcode] = (.5w, x_ht);
labels(1,2,3,4);
endchar;

```

## The Letter *caron*



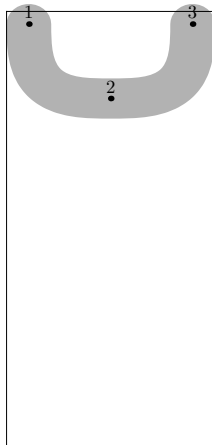


```

ffmchar("caron", 7,  $x_{ht\#} + acc_{ht\#}$ , 0);
 $lft\ x_1 = noise$ ;
 $x_2 = .5w + noise$ ;
 $rt\ x_3 = w + noise$ ;
 $top\ y_1 = h + o + noise$ ;
 $top\ y_3 = h + o + noise$ ;
 $bot\ y_2 = .2[x_{ht}, h] + noise$ ;
draw  $z_1 - z_2 - z_3$ ;
 $charanchortops\_ [charcode] = (.5w, x_{ht})$ ;
labels(1, 2, 3);
endchar;

```

### The Letter *breve*

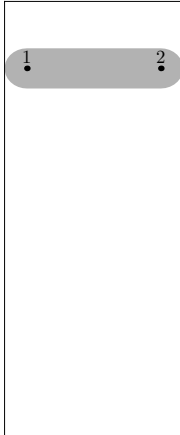


```

ffmchar("breve", 7,  $x_{ht\#} + acc_{ht\#}$ , 0);
 $lft\ x_1 = noise$ ;
 $x_2 = .5w + noise$ ;
 $rt\ x_3 = w + noise$ ;
 $top\ y_1 = h + o + noise$ ;
 $top\ y_3 = h + o + noise$ ;
 $bot\ y_2 = .2[x_{ht}, h] + noise$ ;
draw  $half(z_1, -randup, z_2, randrt, z_3, randup)$ ;
 $charanchortops\_ [charcode] = (.5w, x_{ht})$ ;
labels(1, 2, 3);
endchar;

```

## The Letter *macron*



```

ffmchar("macron",6, $x_{ht\#} + acc_{ht\#}$ ,0);
 $lft\ x_1 = noise$ ;
 $rt\ x_2 = w + noise$ ;
 $y_1 = .5[x_{ht},h] + noise$ ;
 $y_2 = .5[x_{ht},h] + noise$ ;
draw  $z_1 - z_2$ ;
 $charanchor_tops_{[charcode]} = (.5w, x_{ht})$ ;
labels(1,2);
endchar;

```

## The Letter *dotaccent*



```

ffmchar("dotaccent",4, $x_{ht\#} + acc_{ht\#}$ ,0);
 $x_1 = x_2 = .5w + noise$ ;
 $bot\ y_1 = .5[x_{ht},h] + noise$ ;
 $y_2 = y_1 + dotincr * py$ ;

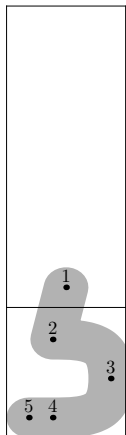
```

```

draw dotcircle(z1, z2);
charanchortops_[charcode] = (.5w, xht);
labels(1, 2);
endchar;

```

### The Letter *cedilla*



```

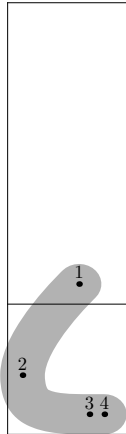
ffmchar("cedilla", 4, xht#, accdepth#);
x1 = .5w;
lft x2 = .2w + .5noise;
rt x3 = w + o + .5noise;
x4 = x2;
lft x5 = 0;
bot y1 = 0;
y2 = .4[y1, y4];
y3 = .7[y1, y4];
bot y4 = noise - d;
z5 = z4 + whatever * randrt;
draw z5 - z4
& half(z4, z4 - z5, z3, randup, z2, -randrt)
& z2 - z1;
charanchorbots_[charcode] = z1;
labels(1, 2, 3, 4, 5);
endchar;

```

%no noise!

%no noise!

## The Letter *ogonek*

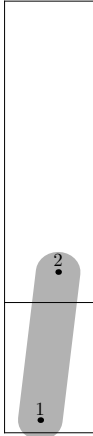


```

ffmchar("ogonek", 4,  $x_{ht\#}$ ,  $acc\_depth\#$ );
 $x_1 = .6w$ ;                                     %no noise!
 $lft\ x_2 = -o + .5noise$ ;
 $rt\ x_4 = w + .5noise$ ;
 $x_3 = x_4 - .5u$ ;
 $bot\ y_1 = 0$ ;                                     %no noise!
 $y_2 = .7[y_1, y_4]$ ;
 $bot\ y_3 = noise - d$ ;
 $z_4 = z_3 + whatever * randrt$ ;
pair $randir$ ;
 $randir := -randup$ ;
draw  $z_1 .. tensioninfinity\ mand\ 1 .. z_2\{randir\}$ 
&  $arc(z_2, randir, z_3, randrt)$ 
&  $z_3 - z_4$ ;
 $charanchorbots\_ [charcode] = z_1$ ;
labels(1, 2, 3, 4, 5);
endchar;

```

### The Letter *quotesinglbase*

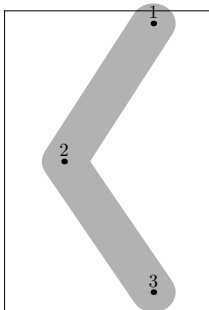


```

ffmchar("quotesinglbase",3,x_ht#,comma_depth#);
 $x_1 = .4w + noise;$ 
 $x_2 = .6w + noise;$ 
 $bot\ y_1 = noise - d - o;$ 
 $y_2 = -d + ht - .5[barheight, x\_ht] + noise;$ 
draw  $z_1 - z_2;$ 
labels(1,2);
endchar;

```

### The Letter *guilsinglleft*



```

ffmchar("guilsinglleft",7,x_ht#,0);
italcorr  $x\_ht# * slant;$ 
 $x_1 = good.x(w - 2u - s + noise);$ 
 $x_2 = good.x(2u + s + noise);$ 
 $x_3 = good.x(w - 2u - s + noise);$ 
 $top\ y_1 = h + o + noise;$ 
 $bot\ y_3 = noise;$ 
 $y_2 = .5h + noise;$ 

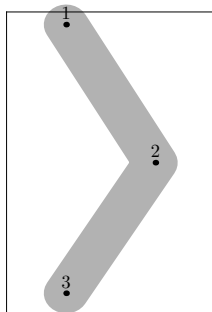
```

```

draw  $z_1 - z_2 - z_3$ ;
labels(1, 2, 3);
endchar;

```

### The Letter *guilsinglright*

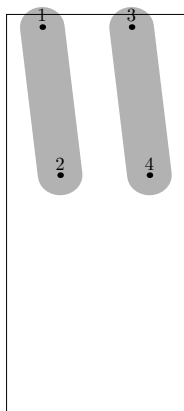


```

ffmchar("guilsinglright", 7,  $x_{ht\#}$ , 0);
italcorr .7 $x_{ht\#}$  * slant;
 $x_1 = \text{good.x}(2u + s + \text{noise})$ ;
 $x_2 = \text{good.x}(w - 2u - s + \text{noise})$ ;
 $x_3 = \text{good.x}(2u + s + \text{noise})$ ;
 $\text{top } y_1 = h + o + \text{noise}$ ;
 $\text{bot } y_3 = \text{noise}$ ;
 $y_2 = .5h + \text{noise}$ ;
draw  $z_1 - z_2 - z_3$ ;
labels(1, 2, 3);
endchar;

```

### The Letter *quotedblleft*

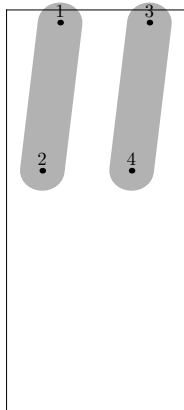


```

fmchar("quotedblleft",6,ht#,0);
italcorr ht# * slant;
 $x_1 = .2w + noise;$ 
 $x_4 = .8w + noise;$ 
 $x_2 = .3w + noise;$ 
 $x_3 = .7w + noise;$ 
 $top\ y_1 = h + o + noise;$ 
 $top\ y_3 = h + o + noise;$ 
 $y_2 = .5[barheight, x\_ht] + noise;$ 
 $y_4 = .5[barheight, x\_ht] + noise;$ 
draw  $z_1 - z_2;$ 
draw  $z_3 - z_4;$ 
labels(1,2,3,4);
endchar;

```

### The Letter *quotedblright*



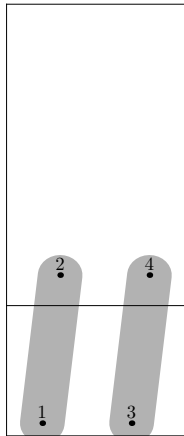
```

fmchar("quotedblright",6,ht#,0);
italcorr ht# * slant;
 $x_2 = .2w + noise;$ 
 $x_3 = .8w + noise;$ 
 $x_1 = .3w + noise;$ 
 $x_4 = .7w + noise;$ 
 $top\ y_1 = h + o + noise;$ 
 $top\ y_3 = h + o + noise;$ 
 $y_2 = .5[barheight, x\_ht] + noise;$ 
 $y_4 = .5[barheight, x\_ht] + noise;$ 
draw  $z_1 - z_2;$ 
draw  $z_3 - z_4;$ 
labels(1,2,3,4);

```

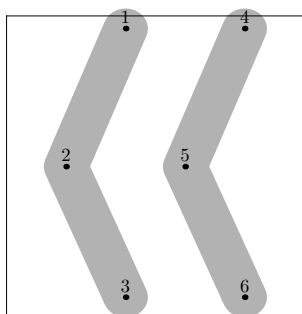
```
endchar;
```

### The Letter *quotedblbase*



```
ffmchar("quotedblbase",6,x_ht#,comma_depth#);
x_2 = .3w + noise;
x_3 = .7w + noise;
x_1 = .2w + noise;
x_4 = .8w + noise;
bot y_1 = noise - d - o;
bot y_3 = noise - d - o;
y_2 = -d + ht - .5[barheight,x_ht] + noise;
y_4 = -d + ht - .5[barheight,x_ht] + noise;
draw z_1 - z_2;
draw z_3 - z_4;
labels(1,2,3,4);
endchar;
```

### The Letter *guillemotleft*



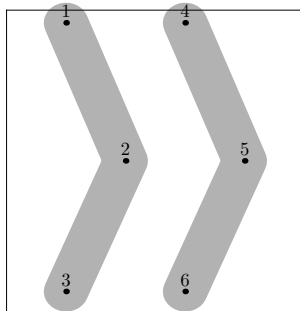


```

ffmchar("guillemotleft",10, $x_{ht\#}$ ,0);
italcorr  $x_{ht\#} * slant$ ;
 $x_1 = .4w + noise$ ;
 $x_2 = good.x(2u + s + noise)$ ;
 $x_3 = .4w + noise$ ;
 $x_4 = good.x(w - 2u - s + noise)$ ;
 $x_5 = .6w + noise$ ;
 $x_6 = good.x(w - 2u - s + noise)$ ;
 $top\ y_1 = h + o + noise$ ;
 $bot\ y_3 = noise$ ;
 $y_2 = .5h + noise$ ;
 $top\ y_4 = h + o + noise$ ;
 $bot\ y_6 = noise$ ;
 $y_5 = .5h + noise$ ;
draw  $z_1 - z_2 - z_3$ ;
draw  $z_4 - z_5 - z_6$ ;
labels(1,2,3,4,5,6);
endchar;

```

### The Letter *guillemotright*



```

ffmchar("guillemotright",10, $x_{ht\#}$ ,0);
italcorr  $.7x_{ht\#} * slant$ ;
 $x_1 = good.x(2u + s + noise)$ ;
 $x_2 = .4w + noise$ ;
 $x_3 = good.x(2u + s + noise)$ ;
 $x_4 = .6w + noise$ ;
 $x_5 = good.x(w - 2u - s + noise)$ ;
 $x_6 = .6w + noise$ ;
 $top\ y_1 = h + o + noise$ ;
 $bot\ y_3 = noise$ ;
 $y_2 = .5h + noise$ ;
 $top\ y_4 = h + o + noise$ ;

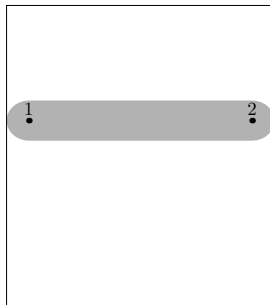
```

```

bot y6 = noise;
y5 = .5h + noise;
draw z1 - z2 - z3;
draw z4 - z5 - z6;
labels(1, 2, 3, 4, 5, 6);
endchar;

```

### The Letter *endash*

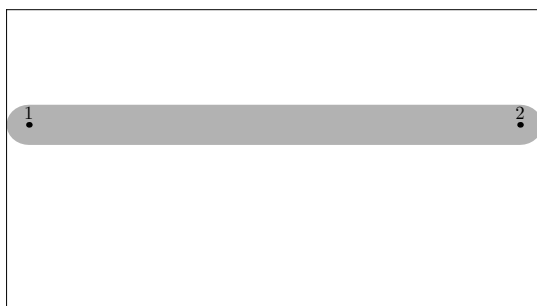


```

ffmchar("endash", 9, x_ht#, 0);
italcorr .618x_ht# * slant;
lft x1 = noise;
rt x2 = w + noise;
y1 = .618h + noise;
y2 = .618h + noise;
draw z1 - z2;
labels(1, 2);
endchar;

```

### The Letter *emdash*



```

ffmchar("emdash", 18, x_ht#, 0);
italcorr .618x_ht# * slant;
lft x1 = noise;

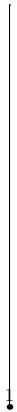
```

```

rt  $x_2 = w + noise$ ;
 $y_1 = .618h + noise$ ;
 $y_2 = .618h + noise$ ;
draw  $z_1 - z_2$ ;
labels(1,2);
endchar;

```

### The Letter *cwm*



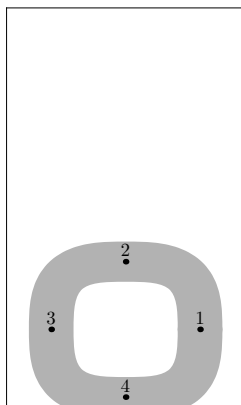
```

ffmchar("cwm", 0, ht#, 0);
 $z_1 = (0, 0)$ ;
undraw  $z_1$ ;
labels(1);
endchar;

```

%just for fontforge

### The Letter *perthousandzero*



```

fmchar("perthousandzero", 8, ht#, 0);
x3 = good.x(1.5u + s + noise);
w - x1 = good.x(1.5u + s + noise);
x2 = .5w + noise;
x4 = .5w + noise;
y1 = .2h + noise;
top y2 = .4h + o + noise;
y3 = .2h + noise;
bot y4 = noise - o;
draw full(z1, randup, z2, -randrt, z3, -randup, z4, randrt);
labels(1, 2, 3, 4);
endchar;

```

### The Letter *visiblespace*

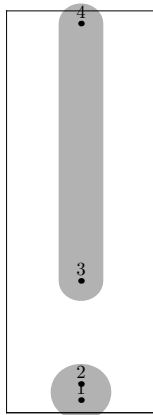


```

fmchar("visiblespace", 6, ht#, comma_depth);
lft x1 = good.x(.5u + noise);
lft x2 = good.x(.5u + noise);
rt x3 = good.x(w - .5u + noise);
rt x4 = good.x(w - .5u + noise);
top y1 = o + noise;
top y4 = o + noise;
bot y2 = noise - d;
bot y3 = noise - d;
draw z1 - z2 - z3 - z4;
labels(1, 2, 3, 4);
endchar;

```

### The Letter *exclam*

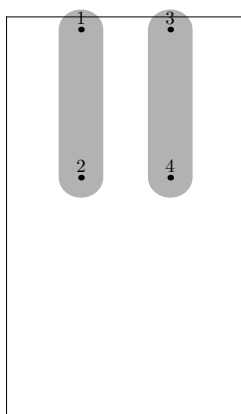


```

ffmchar("exclam", 5, ht#, 0);
italcorr .8ht# * slant;
 $x_1 = x_2 = .5w + noise;$ 
 $x_3 = .5w + noise;$ 
 $x_4 = .5w + noise;$ 
 $bot\ y_1 = noise - o;$ 
 $y_2 = y_1 + dotincr * py;$ 
 $top\ y_4 = h + o + noise;$ 
 $bot\ y_3 = \max(.618barheight, top\ y_2 + eps) + noise;$ 
draw dotcircle( $z_1, z_2$ );
draw  $z_3 - z_4$ ;
labels(1, 2, 3, 4);
endchar;

```

### The Letter *quotedbl*

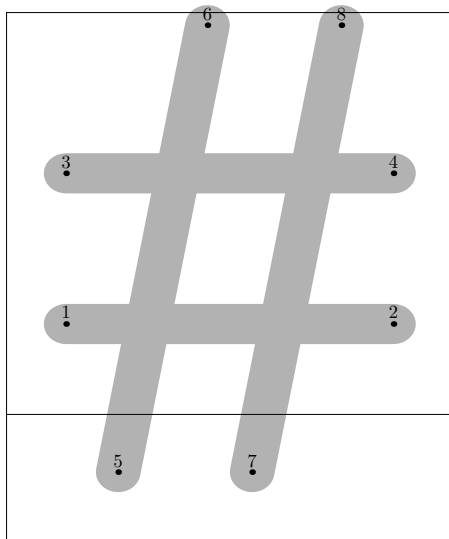


```

ffmchar("quotedbl", 8,  $ht^\#$ , 0);
italcorr  $ht^\# * slant$ ;
 $x_1 = leftstemloc + noise$ ;
 $x_2 = leftstemloc + noise$ ;
 $x_3 = w - leftstemloc + noise$ ;
 $x_4 = w - leftstemloc + noise$ ;
 $top y_1 = h + o + noise$ ;
 $top y_3 = h + o + noise$ ;
 $y_2 = .5[barheight, x\_ht] + noise$ ;
 $y_4 = .5[barheight, x\_ht] + noise$ ;
draw  $z_1 - z_2$ ;
draw  $z_3 - z_4$ ;
labels(1, 2, 3, 4);
endchar;

```

### The Letter *numbersign*



```

ffmchar("numbersign", 15,  $ht^\#$ ,  $comma\_depth^\#$ );
italcorr  $.8ht^\# * slant$ ;
 $x_1 = good.x(2u + s + noise)$ ;
 $x_2 = good.x(w - 2u - s + noise)$ ;
 $x_3 = good.x(2u + s + noise)$ ;
 $x_4 = good.x(w - 2u - s + noise)$ ;
 $x_5 = .25w + noise$ ;
 $x_6 = .45w + noise$ ;
 $x_7 = .55w + noise$ ;
 $x_8 = .75w + noise$ ;

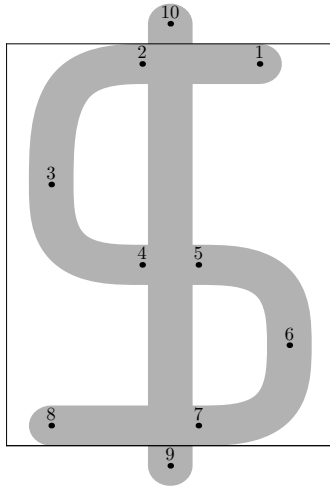
```

```

 $y_1 = .3x_{ht} + noise;$ 
 $y_2 = .3x_{ht} + noise;$ 
 $y_3 = .8x_{ht} + noise;$ 
 $y_4 = .8x_{ht} + noise;$ 
 $bot\ y_5 = 1.1x_{ht} - h - o + noise;$ 
 $top\ y_6 = h + o + noise;$ 
 $bot\ y_7 = 1.1x_{ht} - h - o + noise;$ 
 $top\ y_8 = h + o + noise;$ 
draw  $z_1 - z_2;$ 
draw  $z_3 - z_4;$ 
draw  $z_5 - z_6;$ 
draw  $z_7 - z_8;$ 
labels(1, 2, 3, 4, 5, 6, 7, 8);
endchar;

```

### The Letter *dollar*



```

ffmchar("dollar", 11,  $ht\#$ , 0);
italcorr  $.7ht\# * slant;$ 
 $x_3 = good.x(1.5u + s + noise);$ 
 $x_8 = good.x(1.5u + s + noise);$ 
 $w - x_1 = leftstemloc + noise;$ 
 $w - x_6 = good.x(1.5u + s + noise);$ 
 $x_2 = .382[x_3, x_6] + noise;$ 
 $x_4 = .382[x_3, x_6] + noise;$ 
 $x_5 = .618[x_3, x_6] + noise;$ 
 $x_7 = .618[x_3, x_6] + noise;$ 
 $x_9 = .5w + noise;$ 
 $x_{10} = .5w + noise;$ 

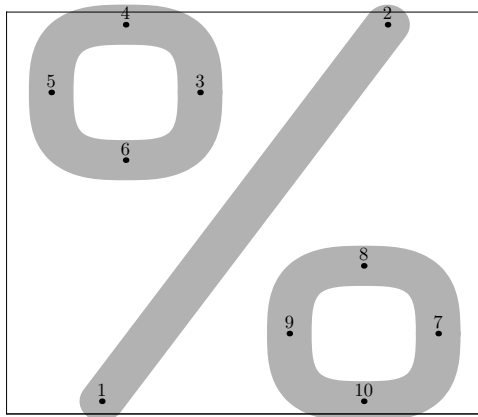
```

```

top y1 = h + noise;
top y2 = h + noise;
y3 = .6[y2, y4] + noise;
y4 = barheight + noise;
y5 = barheight + noise;
y6 = .5[y5, y7] + noise;
bot y7 = noise;
bot y8 = noise;
top y9 = noise;
bot y10 = h + noise;
draw z1 - z2
& half(z2, z2 - z1, z3, -randup, z4, z5 - z4)
& z4 - z5
& half(z5, z5 - z4, z6, -randup, z7, z8 - z7)
& z7 - z8;
draw z9 - z10;
labels(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
endchar;

```

### The Letter *percent*



```

ffmchar("percent", 16, ht#, 0);
italcorr .8ht# * slant;
x5 = good.x(1.5u + s + noise);
w - x7 = good.x(1.5u + s + noise);
x1 = .2w + noise;
x2 = .8w + noise;
x3 = 6.5u + s + noise;
x4 = 4u + s + noise;
x6 = 4u + s + noise;

```

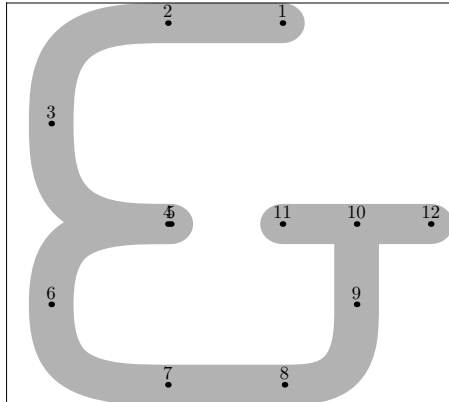


```

 $w - x_8 = 4u + s + noise;$ 
 $w - x_9 = 6.5u + s + noise;$ 
 $w - x_{10} = 4u + s + noise;$ 
 $bot\ y_1 = noise - o;$ 
 $top\ y_2 = h + o + noise;$ 
 $y_3 = .8h + noise;$ 
 $top\ y_4 = h + o + noise;$ 
 $y_5 = .8h + noise;$ 
 $bot\ y_6 = .6h - o + noise;$ 
 $y_7 = .2h + noise;$ 
 $top\ y_8 = .4h + o + noise;$ 
 $y_9 = .2h + noise;$ 
 $bot\ y_{10} = noise - o;$ 
draw  $z_1 - z_2;$ 
draw full( $z_3, randup, z_4, -randrt, z_5, -randup, z_6, randrt$ );
draw full( $z_7, randup, z_8, -randrt, z_9, -randup, z_{10}, randrt$ );
labels(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
endchar;

```

### The Letter *ampersand*



```

ffmchar("ampersand", 15,  $ht^\#$ , 0);
italcorr  $ht^\# * slant$ ;
 $x_3 = good.x(1.5u + s + noise);$ 
 $x_6 = good.x(1.5u + s + noise);$ 
 $rt\ x_{12} = w - eps;$ 
 $x_1 = .618w + noise;$ 
 $x_{11} = .618w + noise;$ 
 $x_{10} = .5[x_{11}, x_{12}] + noise;$ 
 $x_9 = .5[x_{11}, x_{12}] + noise;$ 
 $x_2 = .618[x_9, x_6] + noise;$ 

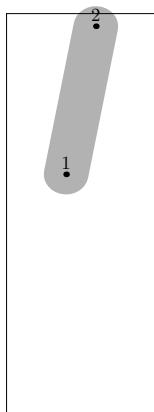
```

```

x4 = .618[x9, x6] + noise;
x7 = .618[x9, x6] + noise;
x8 = .618[x7, x9] + noise;
x5 = x4 + .1u;
top y1 = h + .5noise;
top y2 = h + .5noise;
bot y7 = .5noise;
bot y8 = .5noise;
y4 = barheight + noise;
y11 = barheight + noise;
y12 = barheight + noise;
y3 = .5[y2, y4] + noise;
y6 = .5[y4, y7] + noise;
y9 = .5[y4, y7] + noise;
z5 = z4 + whatever * randrt;
z10 = whatever[z11, z12];
draw z1 - z2
& half(z2, z2 - z1, z3, -randup, z4, z5 - z4)
& z4 - z5;
draw half(z4, z4 - z5, z6, -randup, z7, z8 - z7)
& z7 - z8
& arc(z8, z8 - z7, z9, z10 - z9)
& z9 - z10;
draw z11 - z12;
labels(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12);
endchar;

```

### The Letter *quoteright*



```

ffmchar("quoteright", 5, ht#, 0);
italcorr ht# * slant;

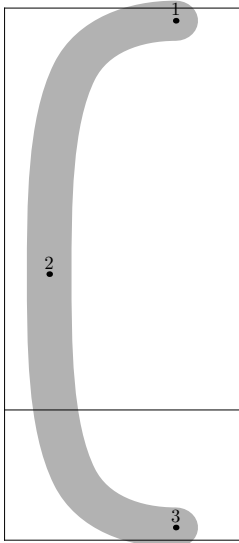
```

```

 $x_1 = .4w + noise;$ 
 $x_2 = .6w + noise;$ 
 $top\ y_2 = h + o + noise;$ 
 $y_1 = .5[barheight, x\_ht] + noise;$ 
draw  $z_1 - z_2;$ 
 $charanchor\ toprights\_ [charcode] = (.5w, h);$ 
labels(1, 2);
endchar;

```

### The Letter *parenleft*

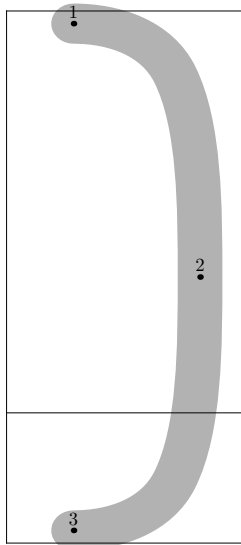


```

ffmchar("parenleft", 8,  $ht\#$ ,  $comma\_depth\#$ );
italcorr  $.8ht\# * slant;$ 
 $x_2 = good.x(1.5u + s + noise);$ 
 $w - x_1 = leftstemloc - ho + noise;$ 
 $w - x_3 = leftstemloc - ho + noise;$ 
 $top\ y_1 = h + o + noise;$ 
 $bot\ y_3 = noise - o - d;$ 
 $y_2 = .5[-d, h] + noise;$ 
draw  $half(z_1, -randrt, z_2, -randup, z_3, randrt);$ 
labels(1, 2, 3);
endchar;

```

## The Letter *parenright*

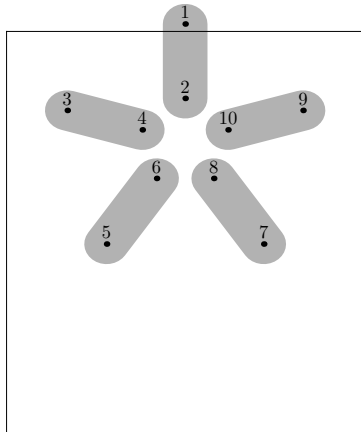


```

ffmchar("parenright",8,ht#,comma__depth#);
italcorr.8ht#*slant;
 $w - x_2 = good.x(1.5u + s + noise);$ 
 $x_1 = leftstemloc - ho + noise;$ 
 $x_3 = leftstemloc - ho + noise;$ 
 $top y_1 = h + o + noise;$ 
 $bot y_3 = noise - o - d;$ 
 $y_2 = .5[-d, h] + noise;$ 
draw half( $z_1$ ,randrt, $z_2$ , -randup, $z_3$ , -randrt);
labels(1,2,3);
endchar;

```

## The Letter *asterisk*

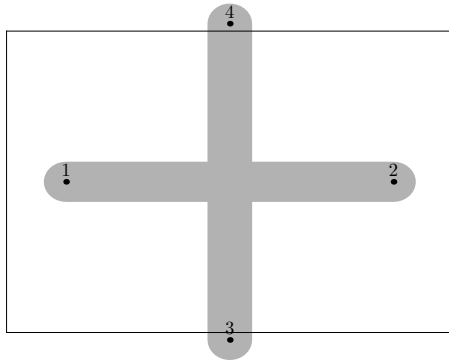


```

fmchar("asterisk",12,ht#,0);
numericouterradius,innerradius;
outerradius = .5w - 2u - s;
innerradius = px;
pathoutercircle,innercircle;
outercircle = (.5w,h + o + noise) .. (.5w,h - 2 * outerradius) .. cycle;
innercircle = (.5w,h + o + noise - outerradius + innerradius)
.. (.5w,h + o + noise - outerradius - innerradius) .. cycle;
z1 = point0of outercircle + (noise,noise);
z2 = point0of innercircle;
z3 = point.4of outercircle + (noise,noise);
z4 = point.4of innercircle;
z5 = point.8of outercircle + (noise,noise);
z6 = point.8of innercircle;
z7 = point1.2of outercircle + (noise,noise);
z8 = point1.2of innercircle;
z9 = point1.6of outercircle + (noise,noise);
z10 = point1.6of innercircle;
draw z1 - z2;
draw z3 - z4;
draw z5 - z6;
draw z7 - z8;
draw z9 - z10;
labels(1,2,3,4,5,6,7,8,9,10);
endchar;

```

## The Letter *plus*

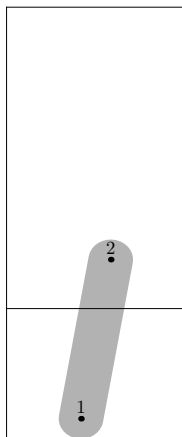


```

ffmchar("plus", 15,  $x_{ht\#}$ , 0);
italcorr .5 $x_{ht\#}$  * slant;
 $x_1 = good.x(2u + s + noise)$ ;
 $x_2 = good.x(w - 2u - s + noise)$ ;
 $x_3 = .5w + noise$ ;
 $x_4 = .5w + noise$ ;
 $y_1 = .5h + noise$ ;
 $y_2 = .5h + noise$ ;
 $y_3 = noise - o$ ;
 $y_4 = h + o + noise$ ;
draw  $z_1 - z_2$ ;
draw  $z_3 - z_4$ ;
labels(1, 2, 3, 4);
endchar;

```

## The Letter *comma*



```

ffmchar("comma", 6,  $x_{ht\#}$ ,  $comma\_depth\#$ );

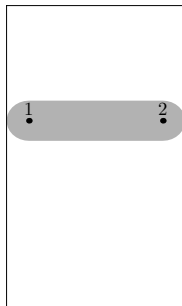
```

```

 $x_1 = \text{leftstemloc};$ 
 $x_2 = w - x_1;$ 
 $\text{top } y_2 = .382\text{barheight};$ 
 $\text{bot } y_1 = -d;$ 
draw  $z_1 - z_2;$ 
labels(1,2);
endchar;

```

### The Letter *hyphen*



```

ffmchar("hyphen", 6,  $x_{ht\#}$ , 0);
italcorr  $.618x_{ht\#} * \text{slant};$ 
 $\text{lft } x_1 = \text{noise};$ 
 $\text{rt } x_2 = w + \text{noise};$ 
 $y_1 = .618h + \text{noise};$ 
 $y_2 = .618h + \text{noise};$ 
draw  $z_1 - z_2;$ 
labels(1,2);
endchar;

```

### The Letter *period*



```

ffmchar("period", 5,  $x_{ht\#}$ , 0);
 $x_1 = x_2 = .5w + \text{noise};$ 

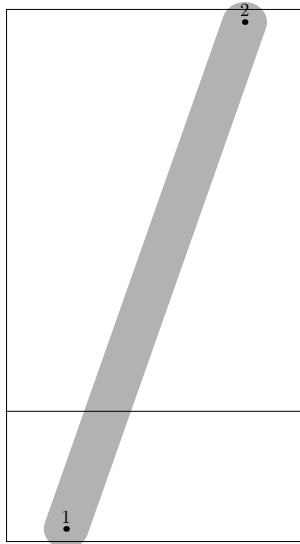
```

```

    bot  $y_1 = noise - o$ ;
     $y_2 = y_1 + dotincr * py$ ;
    draw dotcircle( $z_1, z_2$ );
    labels(1,2);
endchar;

```

### The Letter *slash*



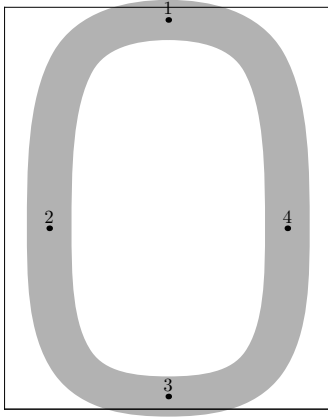
```

fmchar("slash", 10,  $ht^\#$ ,  $comma\_depth^\#$ );
italcorr  $ht^\# * slant$ ;
 $x_1 = good.x(2u + s + noise)$ ;
 $x_2 = good.x(w - 2u - s + noise)$ ;
    bot  $y_1 = noise - d - o$ ;
    top  $y_2 = h + o + noise$ ;
    draw  $z_1 - z_2$ ;
    labels(1,2);
endchar;

```



### The Letter *zero*

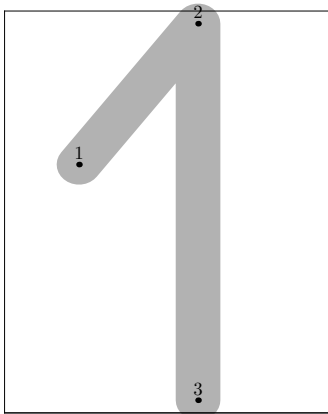


```

ffmchar("zero", 11, ht#, 0);
italcorr .6ht# * slant;
x1 = .5w + noise;
x2 = good.x(1.5u + s + noise);
x3 = .5w + noise;
w - x4 = good.x(1.5u + s + noise);
top y1 = h + o + noise;
y2 = barheight + noise;
bot y3 = noise - o;
y4 = barheight + noise;
draw full(z1, -randrt, z2, -randup, z3, randrt, z4, randup);
labels(1, 2, 3, 4);
endchar;

```

### The Letter *one*



```

ffmchar("one", 11, ht#, 0);

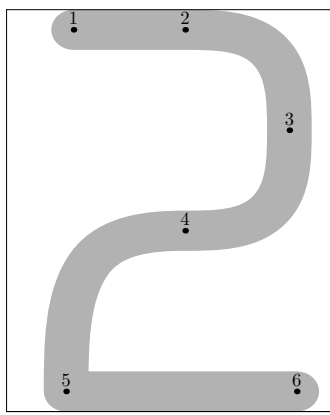
```

```

italcorr .7ht# * slant;
 $x_1 = \text{leftstemloc} + \text{noise}$ ;
 $w - x_2 = \text{good}.x(4.5u + s + \text{noise})$ ;
 $w - x_3 = \text{good}.x(4.5u + s + \text{noise})$ ;
 $y_1 = .618h + \text{noise}$ ;
 $\text{top } y_2 = h + o + \text{noise}$ ;
 $\text{bot } y_3 = \text{noise} - o$ ;
draw  $z_1 - z_2 - z_3$ ;
labels(1, 2, 3);
endchar;

```

## The Letter *two*



```

ffmchar("two", 11, ht#, 0);
italcorr .8ht# * slant;
 $x_1 = \text{leftstemloc} - ho + \text{noise}$ ;
 $x_2 = .5[x_1, x_6] + \text{noise}$ ;
 $w - x_3 = \text{good}.x(1.5u + s + \text{noise})$ ;
 $x_4 = .5[x_1, x_6] + \text{noise}$ ;
 $x_5 = \text{good}.x(2u + s + \text{noise})$ ;
 $x_6 = .5[w - x_1, \text{lft } w] + ho + \text{noise}$ ;
 $\text{top } y_2 = h + \text{noise}$ ;
 $y_3 = .5[y_4, y_2] + \text{noise}$ ;
 $y_4 = \text{barheight} + \text{noise}$ ;
 $\text{bot } y_5 = \text{noise}$ ;
 $\text{bot } y_6 = \text{noise}$ ;
 $z_1 = z_2 + \text{whatever} * \text{randrt}$ ;
pairrandir;
 $\text{randir} := -\text{randrt}$ ;
draw  $z_1 - z_2$ 
&  $\text{half}(z_2, z_2 - z_1, z_3, -\text{randup}, z_4, \text{randir})$ 

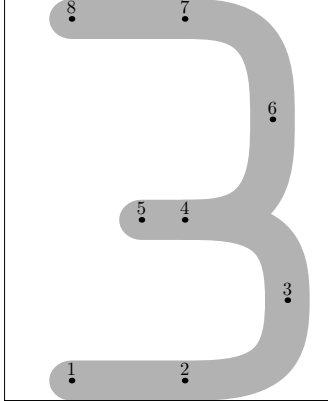
```

```

& arc(z4, randir, z5, -randup)
& z5 - z6;
labels(1, 2, 3, 4, 5, 6);
endchar;

```

### The Letter *three*



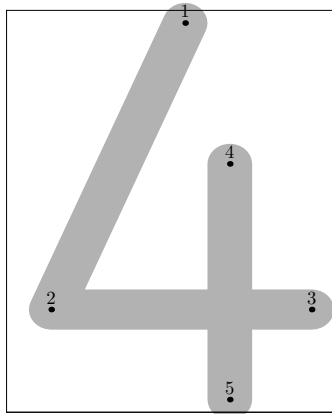
```

fmchar("three", 11, ht#, 0);
italcorr .8ht# * slant;
x1 = leftstemloc - ho + noise;
x8 = leftstemloc - ho + noise;
w - x3 = good.x(1.5u + s + noise);
w - x6 = good.x(2u + s + noise);
x5 = .618[x1, x2] + noise;
x2 = .55w + noise;
x4 = .55w + noise;
x7 = .55w + noise;
bot y1 = noise;
bot y2 = noise;
top y7 = h + noise;
top y8 = h + noise;
y4 = barheight + .5noise;
y5 = barheight + .5noise;
y3 = .5[y2, y4] + noise;
y6 = .5[y4, y7] + noise;
draw z1 - z2
& half(z2, z2 - z1, z3, randup, z4, z5 - z4)
& z4 - z5;
draw half(z4, z4 - z5, z6, randup, z7, z8 - z7)
& z7 - z8;
labels(1, 2, 3, 4, 5, 6, 7, 8);

```

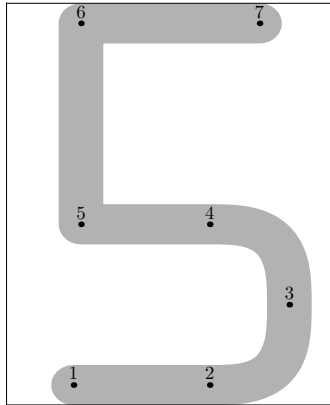
```
endchar;
```

### The Letter *four*



```
ffmchar("four", 11, ht#, 0);
italcorr .7ht# * slant;
x2 = good.x(1.5u + s + noise);
w - rt x3 = eps + noise;
w - x4 = good.x(3.5u + s + noise);
w - x5 = good.x(3.5u + s + noise);
rt x1 = lft x4 + noise;
y4 = .618h + noise;
top y1 = h + o + noise;
bot y5 = noise - o;
y2 = .618[y4, y5] + noise;
y3 = .618[y4, y5] + noise;
draw z1 - z2 - z3;
draw z4 - z5;
labels(1, 2, 3, 4, 5);
endchar;
```

## The Letter *five*

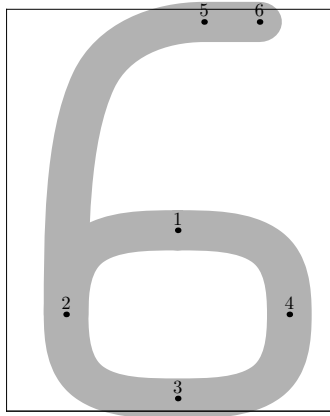


```

ffmchar("five", 11, ht#, 0);
italcorr .8ht# * slant;
 $x_5 = \text{leftstemloc} + \text{noise};$ 
 $x_6 = \text{leftstemloc} + \text{noise};$ 
 $x_7 = w - x_5 + \text{noise};$ 
 $x_1 = x_5 - ho + \text{noise};$ 
 $w - x_3 = \text{good}.x(1.5u + s + \text{noise});$ 
 $x_2 = .618[x_5, x_3] + \text{noise};$ 
 $x_4 = .618[x_5, x_3] + \text{noise};$ 
 $\text{bot } y_1 = \text{noise};$ 
 $\text{bot } y_2 = \text{noise};$ 
 $\text{top } y_6 = h + \text{noise};$ 
 $\text{top } y_7 = h + \text{noise};$ 
 $y_4 = \text{barheight} + .5\text{noise};$ 
 $y_5 = \text{barheight} + .5\text{noise};$ 
 $y_3 = .5[y_2, y_4] + \text{noise};$ 
draw  $z_1 - z_2$ 
& half( $z_2, z_2 - z_1, z_3, \text{randup}, z_4, z_5 - z_4$ )
&  $z_4 - z_5 - z_6 - z_7;$ 
labels(1, 2, 3, 4, 5, 6, 7);
endchar;

```

## The Letter *six*

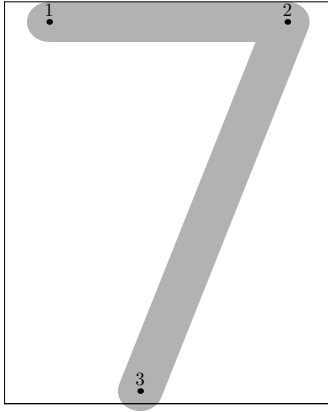


```

ffmchar("six", 11, ht#, 0);
italcorr .8ht# * slant;
x1 = .5[x2, x4] + noise;
x2 = good.x(2u + s + noise);
x3 = .5[x2, x4] + noise;
w - x4 = good.x(1.5u + s + noise);
x5 = .618[x2, x4] + noise;
w - x6 = leftstemloc + noise;
y1 = barheight + noise;
y2 = .5[y1, y3] + noise;
bot y3 = noise - o;
y4 = .5[y1, y3] + noise;
top y5 = h + o + noise;
z6 = z5 + whatever * randrt;
pairrandir;
randir := randup;
draw full(z1, -randrt, z2, -randir, z3, randrt, z4, randup);
draw arc(z2, randir, z5, z6 - z5)
& z5 - z6;
labels(1, 2, 3, 4, 5, 6);
endchar;

```

### The Letter *seven*

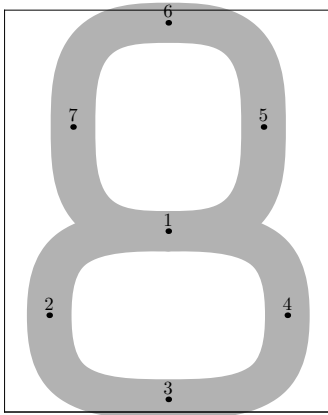


```

ffmchar("seven", 11, ht#, 0);
italcorr ht# * slant;
 $x_1 = \text{good}.x(1.5u + s + \text{noise});$ 
 $w - x_2 = \text{good}.x(1.5u + s + \text{noise});$ 
 $x_3 = .618[x_2, x_1] + \text{noise};$ 
 $\text{top } y_1 = h + \text{noise};$ 
 $\text{top } y_2 = h + \text{noise};$ 
 $\text{bot } y_3 = \text{noise} - o;$ 
draw  $z_1 - z_2 - z_3;$ 
labels(1, 2, 3);
endchar;

```

### The Letter *eight*



```

ffmchar("eight", 11, ht#, 0);
italcorr .7ht# * slant;
 $x_1 = .5w + \text{noise};$ 

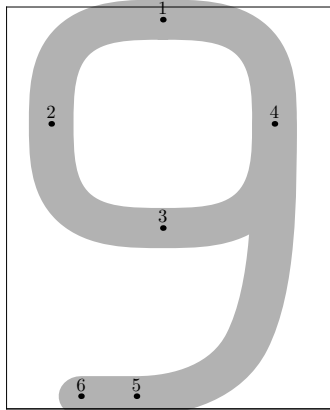
```

```

 $x_3 = .5w + noise;$ 
 $x_6 = .5w + noise;$ 
 $x_2 = good.x(1.5u + s + noise);$ 
 $w - x_4 = good.x(1.5u + s + noise);$ 
 $x_7 = .2[x_2, x_1] + noise;$ 
 $w - x_5 = .2[x_2, x_1] + noise;$ 
 $y_1 = barheight + noise;$ 
 $bot\ y_3 = noise - o;$ 
 $top\ y_6 = h + o + noise;$ 
 $y_2 = .5[y_1, y_3] + noise;$ 
 $y_4 = .5[y_1, y_3] + noise;$ 
 $y_7 = .5[y_1, y_6] + noise;$ 
 $z_5 = z_7 + whatever * (z_4 - z_2);$ 
pairrandir;
randir := randrt;
draw full( $z_1, -randir, z_2, -randup, z_3, randrt, z_4, randup$ );
draw full( $z_1, randir, z_5, randup, z_6, -randrt, z_7, -randup$ );
labels(1, 2, 3, 4, 5, 6, 7);
endchar;

```

### The Letter *nine*



```

ffmchar("nine", 11,  $ht^\#$ , 0);
italcorr  $.7ht^\# * slant$ ;
 $x_1 = .5[x_2, x_4] + noise;$ 
 $x_2 = good.x(1.5u + s + noise);$ 
 $x_3 = .5[x_2, x_4] + noise;$ 
 $w - x_4 = good.x(2u + s + noise);$ 
 $x_5 = .618[x_4, x_2] + noise;$ 
 $x_6 = leftstemloc + noise;$ 
 $top\ y_1 = h + o + noise;$ 

```

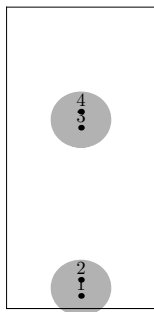


```

y2 = .5[y1, y3] + noise;
y4 = .5[y1, y3] + noise;
bot y5 = noise - o;
y3 = barheight + noise;
z6 = z5 + whatever * randrt;
pairrandir;
randir := randup;
draw full(z1, -randrt, z2, -randup, z3, randrt, z4, randir);
draw arc(z4, -randir, z5, z6 - z5)
& z5 - z6;
labels(1, 2, 3, 4, 5, 6);
endchar;

```

### The Letter *colon*

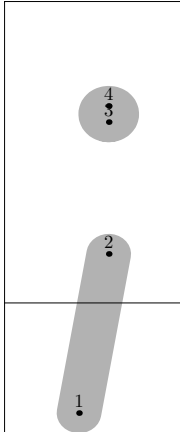


```

ffmchar("colon", 5, x_ht#, 0);
italcorr .8barheight# * slant;
x1 = x2 = .5w + noise;
x3 = x4 = .5w + noise;
bot y1 = noise - o;
y2 = y1 + dotincr * py;
y3 = y4 - dotincr * py = barheight + noise;
draw dotcircle(z1, z2);
draw dotcircle(z3, z4);
labels(1, 2, 3, 4);
endchar;

```

## The Letter *semicolon*

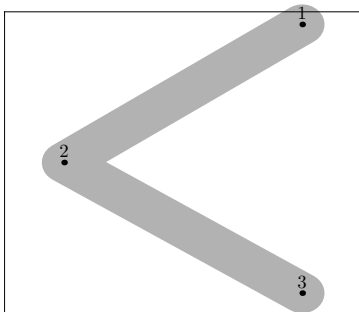


```

fmchar("semicolon", 6,  $x_{ht\#}$ ,  $comma\_depth\#$ );
italcorr  $barheight\# * slant$ ;
 $x_1 = leftstemloc + noise$ ;
 $w - x_2 = leftstemloc + noise$ ;
 $w - x_3 = w - x_4 = leftstemloc + noise$ ;
 $y_3 = y_4 - dotincr * py = barheight$ ;
 $top\ y_2 = .382y_3$ ;
 $bot\ y_1 = -d$ ;
draw  $z_1 - z_2$ ;
draw  $dotcircle(z_3, z_4)$ ;
labels(1, 2, 3, 4);
endchar;

```

## The Letter *less*



```

fmchar("less", 12,  $x_{ht\#}$ , 0);
italcorr  $x_{ht\#} * slant$ ;
 $x_1 = good.x(w - 2u - s + noise)$ ;
 $x_2 = good.x(2u + s + noise)$ ;

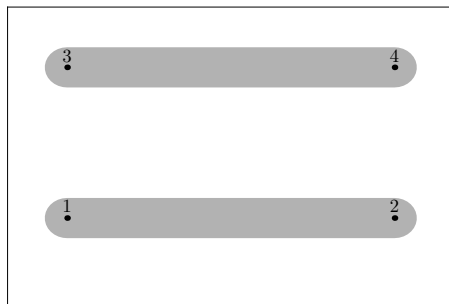
```

```

 $x_3 = \text{good}.x(w - 2u - s + \text{noise});$ 
 $\text{top } y_1 = h + o + \text{noise};$ 
 $\text{bot } y_3 = \text{noise};$ 
 $y_2 = .5h + \text{noise};$ 
draw  $z_1 - z_2 - z_3;$ 
labels(1, 2, 3);
endchar;

```

### The Letter *equal*

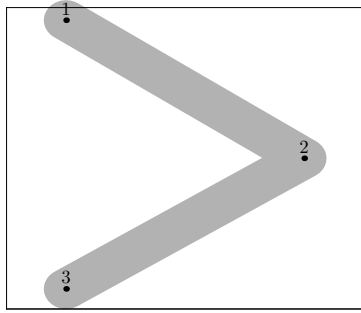


```

ffmchar("equal", 15,  $x_{ht\#}$ , 0);
italcorr  $.8x_{ht\#} * \text{slant};$ 
 $x_1 = \text{good}.x(2u + s + \text{noise});$ 
 $x_2 = \text{good}.x(w - 2u - s + \text{noise});$ 
 $x_3 = \text{good}.x(2u + s + \text{noise});$ 
 $x_4 = \text{good}.x(w - 2u - s + \text{noise});$ 
 $y_1 = .3h + \text{noise};$ 
 $y_2 = .3h + \text{noise};$ 
 $y_3 = .8h + \text{noise};$ 
 $y_4 = .8h + \text{noise};$ 
draw  $z_1 - z_2;$ 
draw  $z_3 - z_4;$ 
labels(1, 2, 3, 4);
endchar;

```

### The Letter *greater*

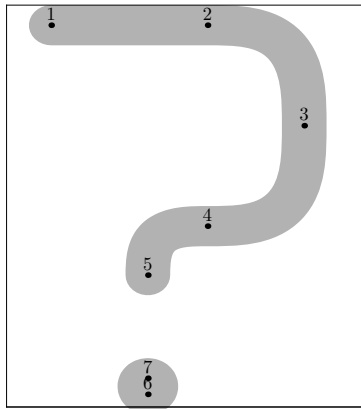


```

fmchar("greater", 12,  $x_{ht\#}$ , 0);
italcorr .5 $x_{ht\#}$  * slant;
 $x_1 = \text{good}.x(2u + s + \text{noise})$ ;
 $x_2 = \text{good}.x(w - 2u - s + \text{noise})$ ;
 $x_3 = \text{good}.x(2u + s + \text{noise})$ ;
 $\text{top } y_1 = h + o + \text{noise}$ ;
 $\text{bot } y_3 = \text{noise}$ ;
 $y_2 = .5h + \text{noise}$ ;
draw  $z_1 - z_2 - z_3$ ;
labels(1, 2, 3);
endchar;

```

### The Letter *question*



```

fmchar("question", 12,  $ht\#$ , 0);
italcorr .8 $ht\#$  * slant;
 $x_1 = \text{good}.x(1.5u + s + \text{noise})$ ;
 $w - x_3 = \text{good}.x(2u + s + \text{noise})$ ;
 $x_5 = .618[x_3, x_1] + \text{noise}$ ;

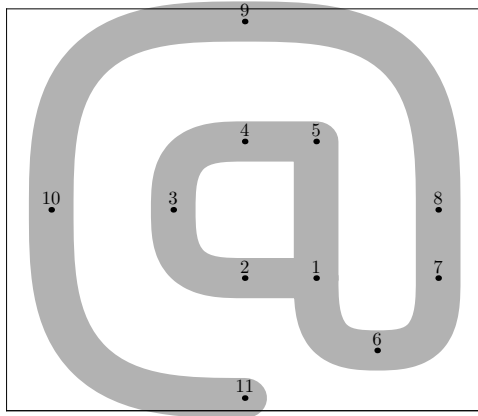
```

```

 $x_2 = .618[x_1, x_3] + noise;$ 
 $x_4 = .618[x_1, x_3] + noise;$ 
 $x_6 = x_7 = .618[x_3, x_1] + noise;$ 
 $top\ y_1 = h + noise;$ 
 $top\ y_2 = h + noise;$ 
 $y_4 = barheight + noise;$ 
 $y_3 = .5[y_2, y_4] + noise;$ 
 $bot\ y_6 = noise - o;$ 
 $y_7 = y_6 + dotincr * py;$ 
 $bot\ y_5 = \max(.618y_4, top\ y_7 + eps) + noise;$ 
pairrandir;
randir := -randrt;
draw  $z_1 - z_2$ 
& half( $z_2, z_2 - z_1, z_3, -randup, z_4, randir$ )
& arc( $z_4, randir, z_5, -randup$ );
draw dotcircle( $z_6, z_7$ );
labels(1, 2, 3, 4, 5, 6, 7);
endchar;

```

### The Letter *at*



```

ffmchar("at", 16, ht#, 0);
italcorr .8ht# * slant;
 $x_{10} = good.x(1.5u + s + noise);$ 
 $w - x_8 = good.x(1.5u + s + noise);$ 
 $w - x_7 = 1.5u + s + noise;$ 
 $x_2 = .5w + noise;$ 
 $x_4 = .5w + .5noise;$ 
 $x_9 = .5w + noise;$ 
 $x_{11} = .5w + noise;$ 
 $x_1 = .65w + noise;$ 

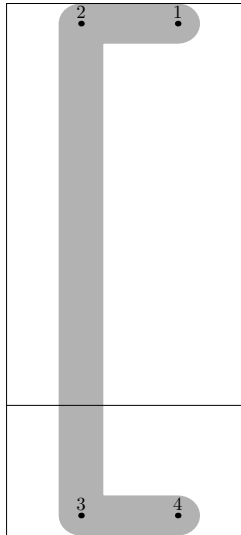
```

```

 $x_5 = .65w + .5noise;$ 
 $x_3 = .35w + noise;$ 
 $x_6 = .5[x_1, x_7];$ 
 $top\ y_9 = h + o + noise;$ 
 $bot\ y_{11} = noise - o;$ 
 $y_8 = .5h + noise;$ 
 $y_{10} = .5h + noise;$ 
 $y_3 = .5h + noise;$ 
 $y_1 = .33h + noise;$ 
 $y_2 = .33h + noise;$ 
 $y_7 = .33h + noise;$ 
 $y_4 = .67h + .5noise;$ 
 $y_5 = .67h + .5noise;$ 
 $y_6 = .15h + noise;$ 
pairrandir;
randir = -randup;
draw  $z_1 - z_2$ 
& half( $z_2, z_2 - z_1, z_3, randup, z_4, z_5 - z_4$ )
&  $z_4 - z_5 - z_1$ 
& half( $z_1, z_1 - z_5, z_6, randrt, z_7, z_8 - z_7$ )
&  $z_7 - z_8$ 
& half( $z_8, z_8 - z_7, z_9, -randrt, z_{10}, randir$ )
& arc( $z_{10}, randir, z_{11}, randrt$ );
labels(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11);
endchar;

```

### The Letter *bracketleft*

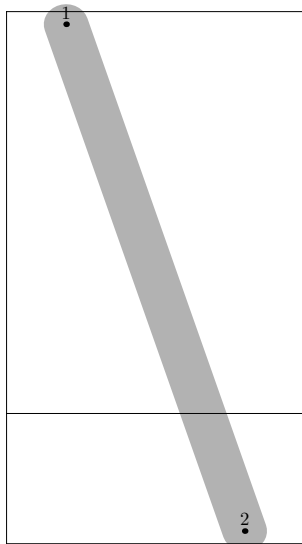


```

ffmchar("bracketleft", 8,  $ht^\#$ ,  $comma\_depth^\#$ );
italcorr  $ht^\# * slant$ ;
 $x_2 = leftstemloc + noise$ ;
 $x_3 = leftstemloc + noise$ ;
 $w - x_1 = leftstemloc - ho + noise$ ;
 $w - x_4 = leftstemloc - ho + noise$ ;
 $top\ y_1 = h + noise$ ;
 $top\ y_2 = h + noise$ ;
 $bot\ y_3 = noise - d$ ;
 $bot\ y_4 = noise - d$ ;
draw  $z_1 - z_2 - z_3 - z_4$ ;
labels(1, 2, 3, 4);
endchar;

```

### The Letter *backslash*

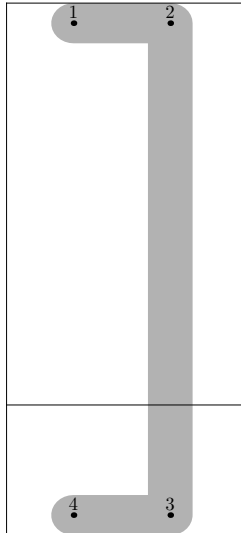


```

ffmchar("backslash", 10,  $ht^\#$ ,  $comma\_depth^\#$ );
 $x_1 = good.x(2u + s + noise)$ ;
 $x_2 = good.x(w - 2u - s + noise)$ ;
 $bot\ y_2 = noise - d - o$ ;
 $top\ y_1 = h + o + noise$ ;
draw  $z_1 - z_2$ ;
labels(1, 2);
endchar;

```

## The Letter *bracketright*



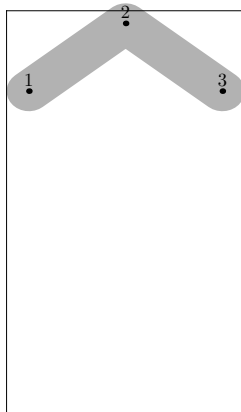
```

ffmchar("bracketright", 8, ht#, comma_depth#);
italcorr ht# * slant;
 $w - x_2 = \text{leftstemloc} + \text{noise};$ 
 $w - x_3 = \text{leftstemloc} + \text{noise};$ 
 $x_1 = \text{leftstemloc} - \text{ho} + \text{noise};$ 
 $x_4 = \text{leftstemloc} - \text{ho} + \text{noise};$ 
 $\text{top } y_1 = h + \text{noise};$ 
 $\text{top } y_2 = h + \text{noise};$ 
 $\text{bot } y_3 = \text{noise} - d;$ 
 $\text{bot } y_4 = \text{noise} - d;$ 
draw  $z_1 - z_2 - z_3 - z_4;$ 
labels(1, 2, 3, 4);
endchar;

```



### The Letter *asciicircum*

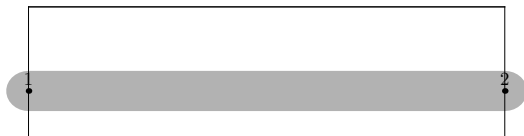


```

fmchar("asciicircum", 8, ht#, 0);
lft  $x_1 = eps + noise$ ;
 $x_2 = .5w + noise$ ;
rt  $x_3 = w - eps + noise$ ;
bot  $y_1 = x\_ht + noise$ ;
bot  $y_3 = x\_ht + noise$ ;
top  $y_2 = h + o + noise$ ;
draw  $z_1 - z_2 - z_3$ ;
labels(1, 2, 3);
endchar;

```

### The Letter *underscore*

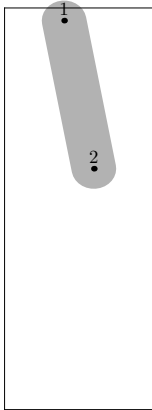


```

fmchar("underscore", 16, 0, comma\_depth);
 $x_1 = 0$ ;
 $x_2 = w$ ;
bot  $y_1 = -.8d$ ;
bot  $y_2 = -.8d$ ;
draw  $z_1 - z_2$ ;
labels(1, 2);
endchar;

```

### The Letter *quoteleft*

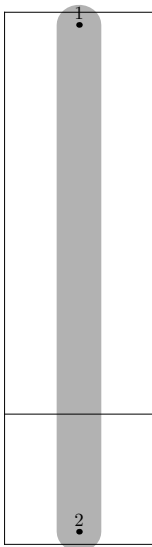


```

ffmchar("quoteleft", 5,  $ht^\#$ , 0);
italcorr  $ht^\# * slant$ ;
 $x_1 = .4w + noise$ ;
 $x_2 = .6w + noise$ ;
 $top\ y_1 = h + o + noise$ ;
 $y_2 = .5[barheight, x\_ht] + noise$ ;
draw  $z_1 - z_2$ ;
labels(1, 2);
endchar;

```

### The Letter *bar*



```

ffmchar("bar", 5,  $ht^\#$ ,  $comma\_depth^\#$ );
 $x_1 = .5w + noise$ ;

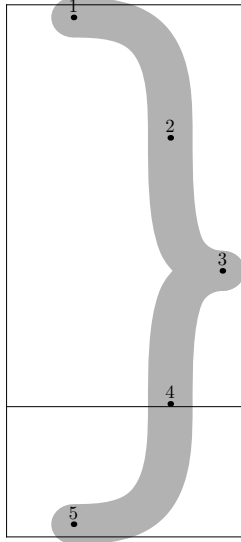
```

```

 $x_2 = .5w + noise;$ 
 $top\ y_1 = h + o + noise;$ 
 $bot\ y_2 = noise - d - o;$ 
draw  $z_1 - z_2;$ 
labels(1, 2);
endchar;

```

### The Letter *braceright*



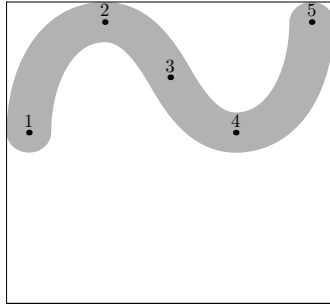
```

ffmchar("braceright", 8,  $ht\#$ ,  $comma\_depth\#$ );
italcorr  $.8ht\# * slant$ ;
 $x_1 = leftstemloc - ho + noise;$ 
 $x_5 = leftstemloc - ho + noise;$ 
 $w - x_2 = leftstemloc + noise;$ 
 $w - x_4 = leftstemloc + noise;$ 
 $rt\ x_3 = w - eps + noise;$ 
 $top\ y_1 = h + o + noise;$ 
 $bot\ y_5 = noise - o - d;$ 
 $y_3 = .5[-d, h] + noise;$ 
 $y_2 = .75[-d, h] + noise;$ 
 $y_4 = .25[-d, h] + noise;$ 
pairrandira;
 $randira = randrt$ ;
draw  $half(z_1, randrt, z_2, -randup, z_3, randira);$ 
draw  $half(z_3, -randira, z_4, -randup, z_5, -randrt);$ 
labels(1, 2, 3, 4, 5);

```

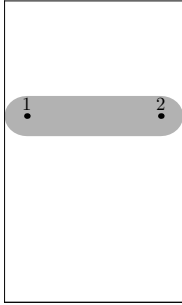
```
endchar;
```

### The Letter *asciitilde*



```
fmchar("asciitilde", 11,  $x_{ht\#}$ , 0);
 $lft\ x_1 = eps + noise$ ;
 $x_2 = .3w + noise$ ;
 $x_4 = .7w + noise$ ;
 $x_3 = .5w + noise$ ;
 $rt\ x_5 = w - eps + noise$ ;
 $bot\ y_1 = .5h + noise$ ;
 $bot\ y_4 = .5h + noise$ ;
 $top\ y_2 = h + noise$ ;
 $top\ y_5 = h + noise$ ;
 $y_3 = .75h + noise$ ;
if angledirectionlof( $z_2\{right\} \dots z_3 \dots z_4\{right\}$ ) < -90:
draw  $z_1\{randup\} \dots z_2\{randrt\}$ 
 $\dots z_3\{-randup\} \dots z_4\{randrt\} \dots z_5\{randup\}$ ;
else:
draw  $z_1\{randup\} \dots z_2\{randrt\}$ 
 $\dots z_3 \dots z_4\{randrt\} \dots z_5\{randup\}$ ;
fi
labels(1, 2, 3, 4, 5);
endchar;
```

## The Letter *dash*

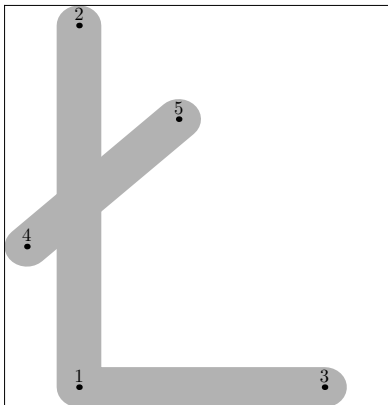


```

fmchar("dash", 6,  $x_{ht\#}$ , 0);
italcorr .618 $x_{ht\#}$  * slant;
 $lft\ x_1 = noise$ ;
 $rt\ x_2 = w + noise$ ;
 $y_1 = .618h + noise$ ;
 $y_2 = .618h + noise$ ;
draw  $z_1 - z_2$ ;
labels(1, 2);
endchar;

```

## The Letter *Lslash*



```

fmchar("Lslash", 13,  $ht\#$ , 0);
 $x_1 = leftstemloc + noise$ ;
 $x_2 = leftstemloc + noise$ ;
 $w - x_3 = leftstemloc - ho + noise$ ;
 $lft\ x_4 = eps + noise$ ;
 $x_5 = .45w + noise$ ;
 $bot\ y_1 = noise$ ;
 $bot\ y_3 = noise$ ;

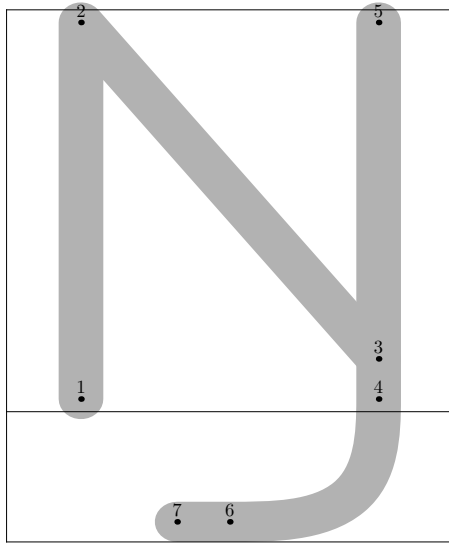
```

```

top y2 = h + noise;
y4 = .4h + noise;
z5 = z4 + whatever * dir(40);
draw z3 - z1 - z2;
draw z4 - z5;
charanchortops__[charcode] = (leftstemloc, h);
charanchortoprighs__[charcode] = (.5w, h);
labels(1, 2, 3, 4, 5);
endchar;

```

## The Letter *Eng*



```

ffmchar("Eng", 15, ht#, acc_depth#);
italcorr .8ht# * slant;
x1 = leftstemloc + noise;
x2 = leftstemloc + noise;
x4 = w - leftstemloc + noise;
x5 = w - leftstemloc + noise;
x6 = .5w + noise;
x7 = .382w + noise;
bot y1 = noise - o;
top y2 = h + o + noise;
y3 = y4 + ygap + noise;
bot y4 = noise - o;
top y5 = h + o + noise;
bot y6 = noise - d;
bot y7 = noise - d;

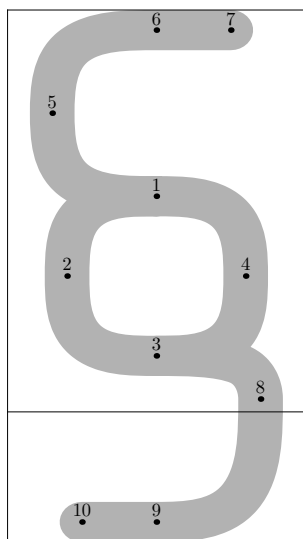
```

```

 $z_3 = \text{whatever}[z_4, z_5];$ 
draw  $z_1 - z_2 - z_3;$ 
draw  $z_7 - z_6$ 
&  $\text{arc}(z_6, z_6 - z_7, z_4, z_5 - z_4)$ 
&  $z_4 - z_5;$ 
 $\text{charanchortops\_}[charcode] = (.5w, h);$ 
labels(1, 2, 3, 4, 5, 6, 7);
endchar;

```

### The Letter *section*



```

ffmchar("section", 10,  $ht\#$ ,  $\text{comma\_depth}\#$ );
italcorr  $.5ht\# * \text{slant}$ ;
 $x_1 = .5w + \text{noise}$ ;
 $x_2 = \text{good}.x(2u + s + \text{noise})$ ;
 $x_3 = .5w + \text{noise}$ ;
 $w - x_4 = \text{good}.x(2u + s + \text{noise})$ ;
 $x_5 = \text{good}.x(1.5u + s + \text{noise})$ ;
 $x_6 = .5w + \text{noise}$ ;
 $w - x_7 = \text{leftstemloc} + \text{noise}$ ;
 $w - x_8 = \text{good}.x(1.5u + s + \text{noise})$ ;
 $x_9 = .5w + \text{noise}$ ;
 $x_{10} = \text{leftstemloc} + \text{noise}$ ;
 $y_1 = .65[-d, h] + \text{noise}$ ;
 $y_2 = .5[-d, h] + \text{noise}$ ;
 $y_3 = .35[-d, h] + \text{noise}$ ;
 $y_4 = .5[-d, h] + \text{noise}$ ;

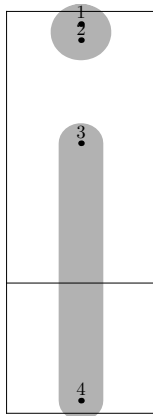
```

```

y5 = .5[y1, y6] + noise;
top y6 = h + noise;
top y7 = h + noise;
y8 = .5[y4, y9] + noise;
bot y9 = noise - d;
bot y10 = noise - d;
pairrandira, randirb;
randira = -randrt;
randirb = randrt;
draw full(z1, randira, z2, -randup, z3, randirb, z4, randup);
draw half(z1, randira, z5, randup, z6, z7 - z6) & z6 - z7;
draw half(z3, randirb, z8, -randup, z9, z10 - z9) & z9 - z10;
charanchortops_[charcode] = (.5w, h);
labels(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
endchar;

```

### The Letter *exclamdown*



```

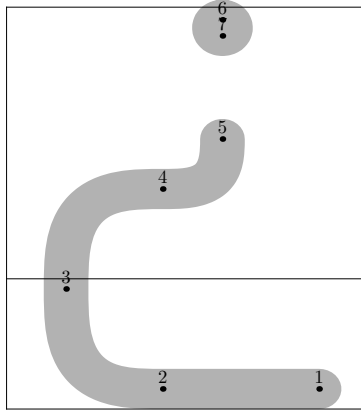
ffmchar("exclamdown", 5, ht# - comma_depth#, comma_depth#);
italcorr .8(ht# - comma_depth#) * slant;
x1 = x2 = .5w + noise;
x3 = .5w + noise;
x4 = .5w + noise;
top y1 = h + o + noise;
y2 = y1 - dotincr * py;
bot y4 = noise - d - o;
top y3 = min(h - .618barheight, bot y2 - eps) + noise;
draw dotcircle(z1, z2);
draw z3 - z4;
labels(1, 2, 3, 4);

```



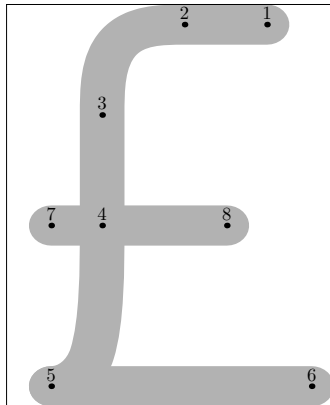
```
endchar;
```

### The Letter *questiondown*



```
fmchar("questiondown", 12, ht# - comma_depth#, comma_depth#);
w - x1 = good.x(1.5u + s + noise);
x3 = good.x(2u + s + noise);
x5 = .618[x3, x1] + noise;
x2 = .618[x1, x3] + noise;
x4 = .618[x1, x3] + noise;
x6 = x7 = .618[x3, x1] + noise;
bot y1 = noise - d;
bot y2 = noise - d;
top y6 = h + o + noise;
y7 = y6 - dotincr * py;
top y5 = min(h - .618barheight, bot y7 - eps) + noise;
y4 = .8[y2, y5] + noise;
y3 = .5[y2, y4] + noise;
pairrandir;
randir := -randrt;
draw arc(z5, -randup, z4, randir)
& half(z4, randir, z3, -randup, z2, z1 - z2)
& z2 - z1;
draw dotcircle(z6, z7);
labels(1, 2, 3, 4, 5, 6, 7);
endchar;
```

## The Letter *sterling*

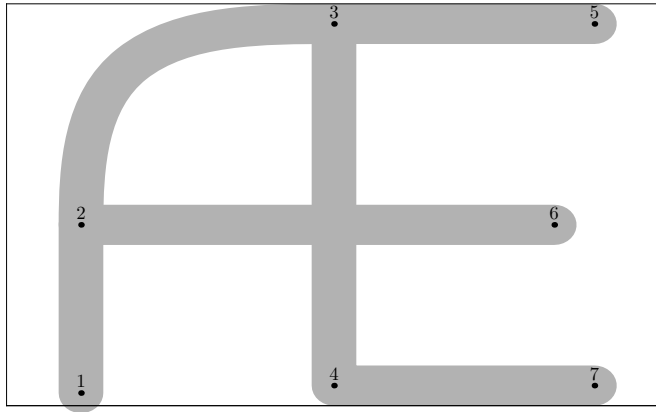


```

fmchar("sterling", 11, ht#, 0);
italcorr .8ht# * slant;
x5 = good.x(1.5u + s + noise);
x7 = good.x(1.5u + s + noise);
rt x6 = w - eps + noise;
x1 = w - leftstemloc + o + noise;
x8 = w - leftstemloc + o - xgap + noise;
x2 = .618[x5, x1] + noise;
x3 = .618[x2, x5] + noise;
x4 = .618[x2, x5] + noise;
top y1 = h + noise;
bot y5 = noise;
bot y6 = noise;
y7 = barheight + noise;
y8 = barheight + noise;
y4 = barheight + noise;
y3 = .5[barheight, h] + noise;
z2 = z1 + whatever * randir;
draw z1 - z2
& arc(z2, z2 - z1, z3, z4 - z3)
& z3 - z4
& arc(z4, z4 - z3, z5, z5 - z6);
draw z5 - z6;
draw z7 - z8;
labels(1, 2, 3, 4, 5, 6, 7, 8);
endchar;

```

## The Letter *AE*

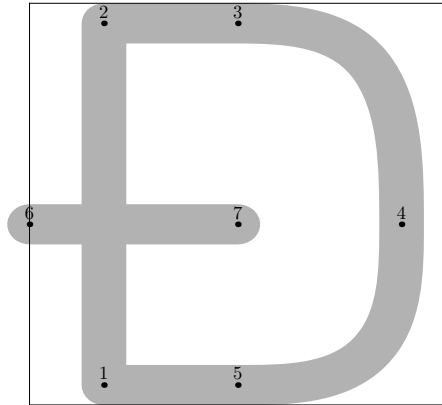


```

ffmchar("AE", 22, ht#, 0);
italcorr .9ht# * slant;
x1 = leftstemloc + noise;
x2 = leftstemloc + noise;
x3 = .5w + noise;
x4 = .5w + noise;
x5 = w - leftstemloc + o + noise;
x6 = w - leftstemloc + o - xgap + noise;
x7 = w - leftstemloc + o + noise;
bot y1 = noise - o;
y2 = barheight + noise;
top y3 = h + noise;
bot y4 = noise;
top y5 = h + noise;
y6 = barheight + noise;
bot y7 = noise;
draw z1 - z2 - z6;
draw arc(z2, z2 - z1, z3, z5 - z3);
draw z5 - z3 - z4 - z7;
labels(1, 2, 3, 4, 5, 6, 7);
endchar;

```

## The Letter *Eth*

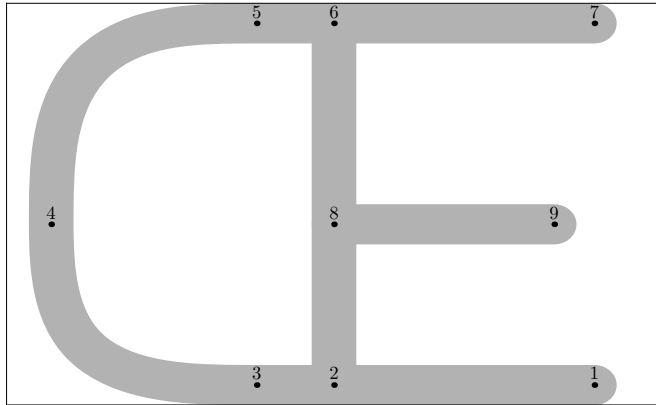


```

ffmchar("Eth", 14, ht#, 0);
italcorr .9ht# * slant;
x1 = leftstemloc + noise;
x2 = leftstemloc + noise;
x3 = .5w + noise;
x5 = .5w + noise;
w - x4 = good.x(1.5u + s + noise);
x6 = eps + noise;
x7 = .5w + noise;
bot y1 = noise;
bot y5 = noise;
top y2 = h + noise;
top y3 = h + noise;
y4 = barheight + noise;
y6 = barheight + noise;
y7 = barheight + noise;
draw z1 - z2 - z3
& half(z3, z3 - z2, z4, -randup, z5, z1 - z5)
& z5 - cycle;
draw z6 - z7;
charanchortops_[charcode] = (.5w, h);
labels(1, 2, 3, 4, 5, 6, 7);
endchar;

```

## The Letter *OE*

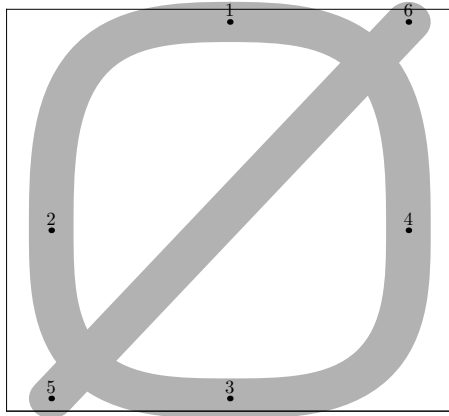


```

ffmchar("OE", 22, ht#, 0);
italcorr .9ht# * slant;
x2 = .5w + noise;
x4 = good.x(1.5u + s + noise);
x6 = .5w + noise;
x7 = w - leftstemloc + o + noise;
x9 = w - leftstemloc + o - xgap + noise;
x1 = w - leftstemloc + o + noise;
x3 = .382w + noise;
x5 = .382w + noise;
y4 = barheight + noise;
y8 = barheight + noise;
top y7 = h + noise;
y9 = barheight + noise;
bot y1 = noise;
bot y3 = noise;
top y5 = h + noise;
z2 = whatever[z1, z3];
z6 = whatever[z5, z7];
z8 = whatever[z2, z6];
draw z1 - z3
& half(z3, z3 - z1, z4, randup, z5, z7 - z5)
& z5 - z7;
draw z2 - z6;
draw z8 - z9;
labels(1, 2, 3, 4, 5, 6, 7, 8, 9);
endchar;

```

## The Letter *Oslash*

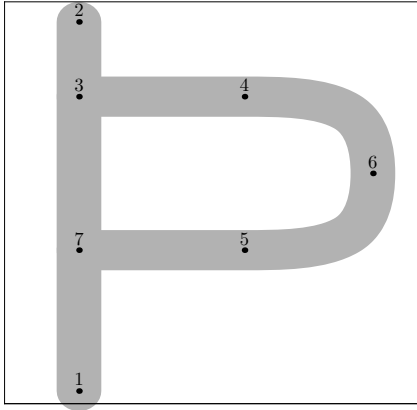


```

ffmchar("Oslash", 15, ht#, 0);
italcorr ht# * slant;
x1 = .5w + noise;
x2 = good.x(1.5u + s + noise);
x3 = .5w + noise;
w - x4 = good.x(1.5u + s + noise);
x5 = good.x(1.5u + s + noise);
w - x6 = good.x(1.5u + s + noise);
top y1 = h + o + noise;
y2 = barheight + noise;
bot y3 = noise - o;
y4 = barheight + noise;
bot y5 = noise - o;
top y6 = h + o + noise;
draw full(z1, -randrt, z2, -randup, z3, randrt, z4, randup);
draw z5 - z6;
labels(1, 2, 3, 4, 5, 6);
endchar;

```

## The Letter *Thorn*

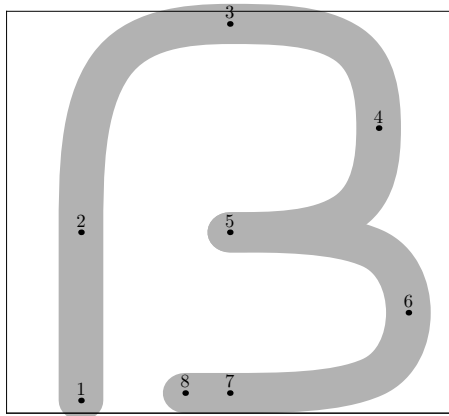


```

ffmchar("Thorn", 14, ht#, 0);
italcorr .7ht# * slant;
x1 = leftstemloc + noise;
x2 = leftstemloc + noise;
x4 = .618[x1, w - x1] + noise;
x5 = .618[x1, w - x1] + noise;
x6 = .5[w - x1, lft w] + noise;
bot y1 = noise - o;
top y2 = h + noise;
y3 = .764h + noise;
y4 = .764h + noise;
y6 = .5[y4, y5] + noise;
y5 = .382h + noise;
y7 = .382h + noise;
z3 = whatever[z1, z2];
z7 = whatever[z1, z2];
draw z1 - z2;
draw z3 - z4
& half(z4, z4 - z3, z6, -randup, z5, z7 - z5)
& z5 - z7;
labels(1, 2, 3, 4, 5, 6, 7);
endchar;

```

## The Letter *Germandbls*



```

fmchar("Germandbls", 15, ht#, 0);
italcorr .8ht# * slant;
x1 = leftstemloc + noise;
x2 = leftstemloc + noise;
x3 = .5w + noise;
x5 = .5w + noise;
x7 = .5w + noise;
w - x4 = leftstemloc + noise;
w - x6 = good.x(1.5u + s + noise);
x8 = .4w + noise;
bot y1 = noise - o;
bot y8 = noise;
y2 = barheight + noise;
top y3 = h + o + noise;
y5 = barheight + noise;
y4 = .5[y5, y3] + noise;
y6 = .5[y7, y5] + noise;
z7 = z8 + whatever * randrt;
pairrandira, randirb;
randira := randrt;
randirb := randrt;
draw z1 - z2
& arc(z2, randup, z3, randira)
& half(z3, randira, z4, -randup, z5, -randirb);
draw half(z5, randirb, z6, -randup, z7, z8 - z7)
& z7 - z8;
labels(1, 2, 3, 4, 5, 6, 7, 8);
endchar;

```



## 6 Font Tables

### Fetamont Light 10

ffml10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	´	^	~	¨	“	°	ˇ
'01*	˘	ˉ	˙	˚	˛	ı	‹	›
'02*	“	”	„	«	»	–	—	...
'03*	o	l	J	FF	FI	FL	FFI	FFL
'04*	u	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	A	B	C	·	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ā	Ĳ	Ć	Č	Ď	Ě	Ě	Ĝ
'17*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ŕ
'18*	Ř	Ś	Ŝ	Ş	Ÿ	Ț	Ú	Û
'19*	Ÿ	Ž	Ž	Ž	ı	ı	Đ	Ş
'20*	Ā	Ĳ	Ć	Č	Ď	Ě	Ě	Ĝ
'21*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ŕ
'22*	Ř	Ś	Ŝ	Ş	Ÿ	Ț	Ú	Û
'23*	Ÿ	Ž	Ž	Ž	ı	ı	Đ	Ş
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

## Fetamont Regular 10

ffmr10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	´	^	~	¨	“	°	ˇ
'01*	˘	-	·	˙	˚	ı	‹	›
'02*	“	”	„	«	»	-	—	.
'03*	◦	ı	ı	FF	FI	FL	FFI	FFL
'04*	˘	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	A	B	C	·	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ

## Fetamont Regular 9

ffmr9	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	ˆ	˜	¨	˝	◦	˘
'01*	˘	-	˙	˙	˙	˙	˙	˙
'02*	"	"	"	«	»	-	—	.
'03*	◦	ı	ı	FF	FI	FL	FFI	FFL
'04*	˘	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	A	B	C	.	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'17*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'18*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'19*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'20*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'21*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'22*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'23*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'24*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'25*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'26*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'27*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'28*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'29*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'30*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'31*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā

## Fetamont Regular 8

ffmr8	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	ˆ	˜	¨	“	°	˘
'01*	˘	-	·	˙	˙	˙	˙	˙
'02*	"	"	"	«	»	-	—	.
'03*	o	l	j	ff	fl	fl	ffi	ffl
'04*	u	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	A	B	C	.	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ā	Ē	Ć	Č	Ď	Ě	Ę	Ğ
'17*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ř
'18*	Ř	Ś	Š	Ş	Ť	Ţ	Ú	Û
'19*	Ÿ	Ž	Ž	Ž	ı	ı	ı	ı
'20*	Ā	Ē	Ć	Č	Ď	Ě	Ę	Ğ
'21*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ř
'22*	Ř	Ś	Š	Ş	Ť	Ţ	Ú	Û
'23*	Ÿ	Ž	Ž	Ž	ı	ı	ı	ı
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

# Fetamont Bold 10

ffmb10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	´	^	~	¨	˘	°	˘
'01*	˘	-	·	¸	˙	ı	‹	›
'02*	"	"	"	«	»	-	—	.
'03*	°	ı	ı	FF	FI	FL	FFI	FFL
'04*	„	!	"	#	\$	%	€	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	A	B	C	·	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ

## Fetamont Bold 9

ffmb9	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	˚	˛	˜	˝	◦	˘
'01*	˘	-	˙	˚	˛	˜	◦	˘
'02*	"	"	"	«	»	-	—	.
'03*	◦	ı	ı	FF	FI	FL	FFI	FFL
'04*	„	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	À	B	C	˙	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ř	Ř	Č	Č	Ď	Ě	Ě	Ğ
'17*	Í	Í	Í	Í	Í	Í	Í	Í
'18*	Ř	Š	Š	Š	Ť	Ť	Ů	Ů
'19*	Ÿ	Ž	Ž	Ž	ı	ı	ı	ı
'20*	Ř	Ř	Č	Č	Ď	Ě	Ě	Ğ
'21*	Í	Í	Í	Í	Í	Í	Í	Í
'22*	Ř	Š	Š	Š	Ť	Ť	Ů	Ů
'23*	Ÿ	Ž	Ž	Ž	ı	ı	ı	ı
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

## Fetamont Bold 8

ffmb8	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	ˆ	˜	¨	˝	◦	˘
'01*	˘	-	·	˙	˚	˛	◁	▷
'02*	"	"	"	«	»	-	—	.
'03*	◦	ı	ı	FF	FI	FL	FFI	FFL
'04*	„	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	·	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	ª	º	˙	˙	˙	˙	˙
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ

## Fetamont Heavy 10

ffmh10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	˚	˛	˜	˝	◦	˘
'01*	˘	-	˙	˚	˛	˜	◦	˘
'02*	"	"	"	«	»	-	—	.
'03*	◦	◦	◦	FF	FI	FL	FFI	FFL
'04*	˘	!	"	#	\$	%	€	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	˘	—
'12*	˘	A	B	C	˙	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	˘	-
'16*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ



## Fetamont Heavy 9

ffmh9	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	˚	˛	˜	˝	◦	˘
'01*	˘	-	˙	˚	˛	˜	◦	˘
'02*	"	"	"	«	»	-	—	.
'03*	◦	◦	◦	FF	FI	FL	FFI	FFL
'04*	◦	!	"	#	\$	%	€	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	˘	—
'12*	˘	A	B	C	˙	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	˘	-
'16*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ

## Fetamont Heavy 8

ffmh8	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˘	˘	˘	˘	˘	˘	˘
'01*	˘	˘	˘	˘	˘	˘	˘	˘
'02*	"	"	"	«	»	—	—	.
'03*	o	i	j	ff	fi	fl	ffi	ffl
'04*	u	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	˘	—
'12*	˘	A	B	C	˘	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	˘	-
'16*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ

## Fetamont Light Oblique 10

ffmlo10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	˘	ˆ	˜	¨	“	°	˘
'01*	˘	-	.	˙	˚	ı	‹	›
'02*	"	"	"	«	»	-	—	...
'03*	o	ı	J	FF	FI	FL	FFI	FFL
'04*	u	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	'	A	B	C	.	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ā	Ā	Ć	Č	Ď	Ě	Ę	Ğ
'17*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ř
'18*	Ř	Ś	Š	Ş	Ť	Ţ	Ů	Ű
'19*	Ÿ	Ž	Ž	Ž	ı	ı	Đ	Ş
'20*	Ā	Ā	Ć	Č	Ď	Ě	Ę	Ğ
'21*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ř
'22*	Ř	Ś	Š	Ş	Ť	Ţ	Ů	Ű
'23*	Ÿ	Ž	Ž	Ž	ı	ı	Đ	Ş
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

## Fetamont Oblique 10

fmo10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	´	^	~	¨	˘	°	˘
'01*	˘	-	·	˙	˚	˛	◁	▷
'02*	"	"	"	«	»	-	—	.
'03*	o	l	j	ff	fl	fl	ffi	ffl
'04*	u	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	'	A	B	C	.	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ā	Ā	Ć	Č	Ď	Ě	Ę	Ğ
'17*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ř
'18*	Ř	Ś	Š	Ş	Ť	Ţ	Ů	Ű
'19*	Ÿ	Ž	Ž	Ž	ı	ı	Đ	Ş
'20*	Ħ	Ħ	Ć	Č	Ď	Ě	Ę	Ğ
'21*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ř
'22*	Ř	Ś	Š	Ş	Ť	Ţ	Ů	Ű
'23*	Ÿ	Ž	Ž	Ž	ı	ı	Đ	Ş
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

## Fetamont Oblique 9

ffmo9	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	˘	ˆ	˜	¨	˝	◦	ˇ
'01*	˘	-	˙	˚	¸	/	◁	▷
'02*	"	"	"	«	»	-	—	.
'03*	◦	ı	ı	FF	FI	FL	FFI	FFL
'04*	˘	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	'	A	B	C	.	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ

## Fetamont Oblique 8

ffmo8	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	˘	ˆ	˜	¨	˝	◦	˘
'01*	˘	-	˙	˚	˛	ˆ	˜	˝
'02*	"	"	"	«	»	-	—	.
'03*	◦	ı	ı	FF	FI	FL	FFI	FFL
'04*	˘	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	A	B	C	.	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ

## Fetamont Bold Oblique 10

ffmbo10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	´	^	~	¨	˜	°	˘
'01*	˘	-	·	¸	˙	ı	‹	›
'02*	"	"	"	«	»	-	—	.
'03*	°	ı	ı	FF	FI	FL	FFI	FFL
'04*	„	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	'	A	B	C	·	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ā	Ā	Ć	Č	Ď	Ě	Ę	Ğ
'17*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ř
'18*	Ř	Ś	Š	Ş	Ť	Ţ	Ů	Ű
'19*	Ÿ	Ž	Ž	Ž	ı	ı	Đ	Š
'20*	Ā	Ā	Ć	Č	Ď	Ě	Ę	Ğ
'21*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ř
'22*	Ř	Ś	Š	Ş	Ť	Ţ	Ů	Ű
'23*	Ÿ	Ž	Ž	Ž	ı	ı	Đ	Š
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

## Fetamont Bold Oblique 9

ffmbo9	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	´	^	~	¨	˜	°	˘
'01*	˘	-	·	¸	˙	˚	◊	›
'02*	"	"	"	«	»	-	—	.
'03*	°	ı	ı	FF	FI	FL	FFI	FFL
'04*	„	!	"	#	\$	%	€	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	À	B	C	·	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ā	Ā	Ć	Č	Ď	Ě	Ę	Ğ
'17*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ř
'18*	Ř	Ś	Š	Ş	Ť	Ţ	Ů	Ű
'19*	Ÿ	Ž	Ž	Ž	ı	ı	Đ	Š
'20*	Ā	Ā	Ć	Č	Ď	Ě	Ę	Ğ
'21*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ř
'22*	Ř	Ś	Š	Ş	Ť	Ţ	Ů	Ű
'23*	Ÿ	Ž	Ž	Ž	ı	ı	Đ	Š
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß



## Fetamont Bold Oblique 8

ffmbo8	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	˚	˛	˜	˝	◦	˘
'01*	˘	-	·	˙	˚	˛	˜	˝
'02*	"	"	"	«	»	-	—	.
'03*	◦	I	J	FF	FI	FL	FFI	FFL
'04*	˘	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	A	B	C	·	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ā	Ā	Č	Č	Ď	Ě	Ě	Ĝ
'17*	Ĺ	Ĺ	Ł	Ń	Ń	Ŋ	Ŏ	Ř
'18*	Ř	Ś	Ś	Ş	Ť	Ť	Ů	Ů
'19*	Ÿ	Ž	Ž	Ž	ı	ı	ø	§
'20*	Ā	Ā	Č	Č	Ď	Ě	Ě	Ĝ
'21*	Ĺ	Ĺ	Ł	Ń	Ń	Ŋ	Ŏ	Ř
'22*	Ř	Ś	Ś	Ş	Ť	Ť	Ů	Ů
'23*	Ÿ	Ž	Ž	Ž	ı	ı	ø	§
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

## Fetamont Heavy Oblique 10

ffmho10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	˘	ˆ	˜	¨	˘	◊	˘
'01*	˘	-	˙	˚	˛	˜	◊	˘
'02*	"	"	"	«	»	-	—	.
'03*	◊	I	J	FF	FI	FL	FFI	FFL
'04*	˙	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	ˆ	—
'12*	˘	A	B	C	˙	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	X	Y	Z	{		}	˘	-
'16*	Ā	Ā	Ā	Č	Ď	Ě	Ě	Ĝ
'17*	Ĺ	Ľ	Ł	Ń	Ñ	Ŋ	Ŏ	Ř
'18*	Ř	Ś	Š	Ş	Ť	Ţ	Ů	Ű
'19*	Ÿ	Ž	Ž	Ž	ı	ı	ø	ſ
'20*	Ā	Ā	Ā	Č	Ď	Ě	Ě	Ĝ
'21*	Ĺ	Ľ	Ł	Ń	Ñ	Ŋ	Ŏ	Ř
'22*	Ř	Ś	Š	Ş	Ť	Ţ	Ů	Ű
'23*	Ÿ	Ž	Ž	Ž	ı	ı	ø	ſ
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

## Fetamont Heavy Oblique 9

ffmho9	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	˚	˛	˜	˝	◦	˘
'01*	˘	-	˙	˚	˛	˜	◦	˘
'02*	"	"	"	«	»	-	—	.
'03*	•	ı	ı	FF	FI	FL	FFI	FFL
'04*	˘	!	"	#	\$	%	€	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	˘	—
'12*	ı	A	B	C	˙	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	˘	-
'16*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'17*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'18*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'19*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'20*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'21*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'22*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'23*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'24*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'25*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'26*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'27*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'28*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'29*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'30*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'31*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā

## Fetamont Heavy Oblique 8

ffmho8	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˘	˘	˘	˘	˘	˘	˘
'01*	˘	˘	˘	˘	˘	˘	˘	˘
'02*	"	"	"	"	"	—	—	.
'03*	o	l	j	ff	fl	fl	ffi	ffl
'04*	u	!	"	#	\$	%	&	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	˘	—
'12*	`	A	B	C	.	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	˘	-
'16*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ

## Fetamont Light Condensed 10

ffmlc10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	´	^	~	¨	“	°	ˇ
'01*	˘	-	·	˙	˚	ı	‹	›
'02*	“	”	„	«	»	–	—	·
'03*	o	l	J	FF	FI	FL	FFI	FFL
'04*	u	!	"	#	\$	%	&	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	À	B	C	·	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ĥ	Ħ	Ć	Č	Ď	Ě	Ě	Ĝ
'17*	Í	Ĺ	Ł	Ń	Ň	Ŋ	Ŏ	Ŕ
'18*	Ř	Ś	Ŝ	Ş	Ţ	Ţ	Ů	Ű
'19*	Ÿ	Ž	Ž	Ž	Ų	ı	Đ	Ş
'20*	Ĥ	Ħ	Ć	Č	Ď	Ě	Ě	Ĝ
'21*	Í	Ĺ	Ł	Ń	Ň	Ŋ	Ŏ	Ŕ
'22*	Ř	Ś	Ŝ	Ş	Ţ	Ţ	Ů	Ű
'23*	Ÿ	Ž	Ž	Ž	Ų	ı	Đ	Ş
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

## Fetamont Condensed 10

ffmc10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	'	^	~	¨	˝	°	˘
'01*	˘	-	·	˙	˚	˛	˜	˚
'02*	"	"	"	«	»	-	—	.
'03*	°	l	J	FF	FI	FL	FFI	FFL
'04*	„	!	"	#	\$	%	€	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	'	A	B	C	.	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ā	Ā	Ć	Č	Ď	Ě	Ę	Ğ
'17*	Ĭ	Ĭ	Ł	Ń	Ň	Ŋ	Ŏ	Ŕ
'18*	Ř	Ś	Š	Ş	Ţ	Ț	Ů	Ű
'19*	Ÿ	Ž	Ž	Ž	ı	ı	ø	ş
'20*	Ā	Ā	Ć	Č	Ď	Ě	Ę	Ğ
'21*	Ĭ	Ĭ	Ł	Ń	Ň	Ŋ	Ŏ	Ŕ
'22*	Ř	Ś	Š	Ş	Ţ	Ț	Ů	Ű
'23*	Ÿ	Ž	Ž	Ž	ı	ı	ø	ş
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

# Fetamont Bold Condensed 40

ffmbc40	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	ˆ	˜	¨	˝	◦	˘
'01*	˘	-	˙	˙	˙	˙	˙	˙
'02*	„	„	„	«	»	-	—	.
'03*	◦	I	J	FF	FI	FL	FFI	FFL
'04*	◦	!	„	#	\$	%	&	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	'	A	B	C	'	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ā	Ā	Č	Č	Ď	Ě	Ě	Ĝ
'17*	Ĺ	Ĺ	Ł	Ń	Ń	Ŋ	Ŏ	Ř
'18*	Ř	Ś	Ś	Ś	Ť	Ť	Ů	Ů
'19*	Ÿ	Ž	Ž	Ž	Ů	İ	Đ	Š
'20*	Ā	Ā	Č	Č	Ď	Ě	Ě	Ĝ
'21*	Ĺ	Ĺ	Ł	Ń	Ń	Ŋ	Ŏ	Ř
'22*	Ř	Ś	Ś	Ś	Ť	Ť	Ů	Ů
'23*	Ÿ	Ž	Ž	Ž	Ů	İ	Đ	Š
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

## Fetamont Light Condensed Oblique 10

ffmlco10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	´	^	~	¨	“	°	√
'01*	˘	-	.	˙	˚	ı	‹	›
'02*	“	”	„	«	»	-	—	...
'03*	o	l	J	FF	FI	FL	FFI	FFL
'04*	u	!	"	#	\$	%	E	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	]	]	^	_
'12*	'	A	B	C	.	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ĥ	Ħ	Ć	Č	Ď	Ě	Ę	Ğ
'17*	Ĺ	Ł	Ł	Ń	Ň	Ÿ	Ŕ	Ŗ
'18*	Ř	Ś	Ŝ	Ş	Ţ	Ţ	Ũ	Ů
'19*	Ÿ	Ž	Ž	Ž	U	ı	ø	§
'20*	Ĥ	Ħ	Ć	Č	Ď	Ě	Ę	Ğ
'21*	Ĺ	Ł	Ł	Ń	Ň	Ÿ	Ŕ	Ŗ
'22*	Ř	Ś	Ŝ	Ş	Ţ	Ţ	Ũ	Ů
'23*	Ÿ	Ž	Ž	Ž	U	ı	ø	§
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß



## Fetamont Condensed Oblique 10

ffmco10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	´	^	~	¨	˝	°	˘
'01*	˘	-	·	˙	˚	˛	◁	▷
'02*	"	"	"	«	»	-	—	.
'03*	°	l	J	FF	FI	FL	FFI	FFL
'04*	„	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	]	]	^	—
'12*	'	A	B	C	.	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ǻ	Ǻ	Ć	Č	Ď	Ě	Ɛ	Ǧ
'17*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ő	Ř
'18*	Ř	Ś	Š	Ş	Ť	Ț	Ů	Ű
'19*	Ÿ	Ž	Ž	Ž	U	İ	Ɖ	Ƒ
'20*	Ǻ	Ǻ	Ć	Č	Ď	Ě	Ɛ	Ǧ
'21*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ő	Ř
'22*	Ř	Ś	Š	Ş	Ť	Ț	Ů	Ű
'23*	Ÿ	Ž	Ž	Ž	U	İ	Ɖ	Ƒ
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ɖ	Ñ	Ò	Ó	Ô	Õ	Ö	Ɔ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ɖ	Ñ	Ò	Ó	Ô	Õ	Ö	Ɔ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

## Fetamont Bold Condensed Oblique 40

ffmbco40	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	´	^	~	¨	˜	•	ˇ
'01*	˘	-	·	¸	˙	ı	‹	›
'02*	"	"	"	«	»	-	—	.
'03*	°	ı	ı	FF	FI	FL	FFI	FFL
'04*	˙	!	"	#	\$	%	€	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	]	]	^	_
'12*	'	A	B	C	·	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ā	Ā	Č	Č	Ď	Ě	Ę	Ğ
'17*	Ĺ	Ĺ	Ł	Ń	Ń	Ŋ	Ŏ	Ř
'18*	Ř	Ś	Š	Ş	Ť	Ť	Ů	Ű
'19*	Ÿ	Ž	Ž	Ž	ı	ı	ø	ƒ
'20*	Ĥ	Ĥ	Ć	Ć	Ď	Ě	Ę	Ğ
'21*	Ĺ	Ĺ	Ł	Ń	Ń	Ŋ	Ŏ	Ř
'22*	Ř	Ś	Š	Ş	Ť	Ť	Ů	Ű
'23*	Ÿ	Ž	Ž	Ž	ı	ı	ø	ƒ
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

## Fetamont Light Ultracondensed 10

ffmlq10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	'	^	"	-	"	°	˘
'01*	ˆ	-	˙	˚	˛	˜	˜	˜
'02*	ˆ	ˆ	ˆ	ˆ	ˆ	-	-	˙
'03*	ˆ	ˆ	ˆ	ˆˆ	ˆˆ	ˆˆ	ˆˆˆ	ˆˆˆ
'04*	˙	!	'	#	\$	%	ˆ	'
'05*	(	)	"	†	˙	-	˙	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	A	B	C	˙	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	(	)	}	"	-
'16*	Ä	Q	Ć	Č	Ö	Ž	Ç	Ğ
'17*	Í	Ł	Ł	Ń	Ń	Ń	Ō	Ŕ
'18*	Ř	Ś	Ś	Ś	Ť	Ť	Ů	Ů
'19*	Ȳ	Ž	Ž	Ž	U	i	ø	ſ
'20*	Ä	Q	Ć	Č	Ö	Ž	Ç	Ğ
'21*	Í	Ł	Ł	Ń	Ń	Ń	Ō	Ŕ
'22*	Ř	Ś	Ś	Ś	Ť	Ť	Ů	Ů
'23*	Ȳ	Ž	Ž	Ž	U	i	ˆ	ˆ
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ø	Ń	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ø	Ń	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

## Fetamont Light Ultracondensed Oblique 10

ffmlqo10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'01*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'02*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'03*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'04*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'05*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'06*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'07*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'08*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'09*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'10*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'11*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'12*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'13*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'14*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'15*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'16*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'17*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'18*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'19*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'20*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'21*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'22*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'23*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'24*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'25*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'26*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'27*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'28*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'29*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'30*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
'31*	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ

## Fetamont Light Script 10

ffmlw10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	ˆ	˜	¨	˝	◊	˘
'01*	˘	-	·	˙	˙	˙	˙	˙
'02*	"	"	"	«	»	—	—	·
'03*	◊	◊	J	FF	FI	FL	FFI	FFL
'04*	˘	!	"	#	\$	%	&	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	A	B	C	·	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ

## Fetamont Script 10

ffmw10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	ˆ	˜	¨	˝	◦	˘
'01*	˘	-	·	˙	˙	˙	˙	˙
'02*	"	"	"	«	»	-	—	.
'03*	◦	l	J	FF	FI	FL	FFI	FFL
'04*	˘	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	—
'12*	'	A	B	C	.	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ǻ	Ǻ	Č	Č	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ

## Fetamont Bold Script 10

ffmbw10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	˚	˛	˜	˝	◦	˘
'01*	˘	-	˙	˚	˛	˜	◦	˘
'02*	"	"	"	«	»	-	—	.
'03*	•	ı	ı	FF	FI	FL	FFI	FFL
'04*	˘	!	"	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	˘	—
'12*	˘	A	B	C	˙	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	˘	-
'16*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ

## Fetamont Heavy Script 10

ffmhw10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˘	˘	˘	˘	˘	˘	˘
'01*	˘	˘	˘	˘	˘	˘	˘	˘
'02*	"	"	"	«	»	—	—	.
'03*	•	ı	ı	FF	FI	FL	FFI	FFL
'04*	˘	!	"	#	\$	%	£	'
'05*	(	)	÷	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	'	—
'12*	˘	A	B	C	˘	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	X	Y	Z	{		}	~	-
'16*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ



## Fetamont Light Script Oblique 10

ffmlwo10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	ˆ	˜	˚	“	°	˘
'01*	˘	˙	ˆ	˜	˚	“	°	˘
'02*	“	”	„	«	»	–	—	·
'03*	o	l	J	FF	FI	FL	FFI	FFL
'04*	u	!	“	#	\$	%	£	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	`	A	B	C	·	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'17*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'18*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'19*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'20*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'21*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'22*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'23*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'24*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'25*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'26*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'27*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'28*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'29*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'30*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'31*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā

## Fetamont Script Oblique 10

ffmwo10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	`	´	^	~	¨	ˆ	°	√
'01*	˘	-	·	¸	˙	ı	‹	›
'02*	"	"	"	«	»	-	—	.
'03*	°	ı	ı	FF	FI	FL	FFI	FFL
'04*	˘	!	"	#	\$	%	&	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	ı	A	B	C	·	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	X	Y	Z	{		}	~	-
'16*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'17*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'18*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'19*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'20*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'21*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'22*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'23*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'24*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'25*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'26*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'27*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'28*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'29*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'30*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ
'31*	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ	Ǻ

## Fetamont Bold Script Oblique 10

ffmbwo10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˙	˚	˛	˜	˝	˚	˛
'01*	˘	-	˙	˚	˛	˝	˚	˛
'02*	"	"	"	«	»	-	—	.
'03*	o	l	j	ff	fi	fl	ffi	ffl
'04*	u	!	"	#	\$	%	&	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	^	_
'12*	'	A	B	C	'	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	~	-
'16*	Ā	Ā	Ć	Č	Ď	Ě	Ę	Ğ
'17*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ř
'18*	Ř	Ś	Š	Ş	Ť	Ţ	Ů	Ű
'19*	Ÿ	Ž	Ž	Ž	ı	ı	Đ	Ş
'20*	Ā	Ā	Ć	Č	Ď	Ě	Ę	Ğ
'21*	Ĺ	Ł	Ł	Ń	Ň	Ŋ	Ŏ	Ř
'22*	Ř	Ś	Š	Ş	Ť	Ţ	Ů	Ű
'23*	Ÿ	Ž	Ž	Ž	ı	ı	Đ	Ş
'24*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'25*	È	É	Ê	Ë	Ì	Í	Î	Ï
'26*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'27*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
'28*	À	Á	Â	Ã	Ä	Å	Æ	Ç
'29*	È	É	Ê	Ë	Ì	Í	Î	Ï
'30*	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Œ
'31*	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß

## Fetamont Heavy Script Oblique 10

ffmhwo10	'0	'1	'2	'3	'4	'5	'6	'7
'00*	˘	˘	˘	˘	˘	˘	˘	˘
'01*	˘	˘	˘	˘	˘	˘	˘	˘
'02*	"	"	"	«	»	-	—	.
'03*	•	ı	ı	FF	FI	FL	FFI	FFL
'04*	˘	!	"	#	\$	%	€	'
'05*	(	)	*	+	,	-	.	/
'06*	0	1	2	3	4	5	6	7
'07*	8	9	:	;	<	=	>	?
'08*	@	A	B	C	D	E	F	G
'09*	H	I	J	K	L	M	N	O
'10*	P	Q	R	S	T	U	V	W
'11*	X	Y	Z	[	\	]	˘	—
'12*	˘	A	B	C	˘	E	F	G
'13*	H	I	J	K	L	M	N	O
'14*	P	Q	R	S	T	U	V	W
'15*	x	y	z	{		}	˘	-
'16*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'17*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'18*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'19*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'20*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'21*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'22*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'23*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'24*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'25*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'26*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'27*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'28*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'29*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'30*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
'31*	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā

## References

[Hosny11] Khaled Hosny. <https://github.com/khaledhosny/punk-otf/blob/master/tools/build.py>, 2011

[Jackowski01] Bogusław Jackowski, Janusz M. Nowacki, and Piotr Strzelczyk. *META-*

*TYPE1: A METAPOST-based engine for generating Type 1 fonts.* [ntg.nl/eurotex/JackowskiMT.pdf](http://ntg.nl/eurotex/JackowskiMT.pdf), 2001

[Nienhuys06] Han-Wen Nienhuys. <https://github.com/hanwen/mftrace/blob/master/tfm.py>, 2006

[Romer14] Linus Romer. *The Fetamont Package*. 2014