Computer Science Department
Report No. STAN-CS-80-780

January 1980

THE COMPUTER MODERN FAMILY OF TYPEFACES

by

Donald E. Knuth

Research sponsored by

National Science Foundaticn
Off ice of Naval Research
IBM Corporation
Xerox Corporation

COMPUTER SCIENCE DEPARTMENT
Stanford University

Computer Science Department
Report No. STAN-CS-80-780

January, 1980
(first printing)

THE COMPUTER MODERN FAMILY OF TYPEFACES

by Donald E. Knuth

The preparation of this report was supported in part by the National Science Foundation, the Office of Naval Research, and the IBM Corporation. The author also wishes to thank the Xerox Corporation for providing facilities with which this report was printed, and for providing computer time while the Stanford computers were being moved to a new building.

PREFACE

This report gives machine-independent definitions of all the styles of type planned for use in future editions of *The Art of Computer* Programming. Its main purpose is to provide a detailed example of a complete family of font definitions using METAFONT, so that people who want new symbols for their own books and papers will understand how to incorporate them easily. The fonts are intended to have the same spirit as those used in earlier editions of The Art of Computer *Programming*, but each character has been redesigned and defined in the METAFONT idiom. It is hoped that some readers will be inspired to make similar defintions of other important families of fonts. The bulk of this report consists of about 400 short METAFONT programs for the various symbols needed, and as such it is pretty boring, but there are some nice illustrations.

It seems appropriate to give the name "Computer Modern" to this family of fonts, because of their associations with computers and because of the fact that the analogous Monotype fonts are called "Modern 8A" . Monotype Modern 8A served for many years as a *de facto* standard for high quality typesetting of mathematics, since these fonts had the most complete collection of characters and special symbols in all the necessary sizes. But the typesetting of technical material has low priority in the printing industry; so the Modern fonts had still not been adapted to photo-optical or photo-digital typesetting equipment by 1977, when Volume 2 of The Art *of* Computer Programming (second edition) was due to appear. Meanwhile the hot-lead Monotype equipment was rapidly becoming extinct **and** prohibitively expensive, so there was no good way to print the second edition in the style of the first.

A preliminary version of Computer Modern was designed by the author in the fall of 1977 and the spring of 1978, using a prototype version of METAFONT. This prototype system included subroutines for drawing curves with pens and erasers, but it did not have METAFONT's declarative language; all characters were drawn by means of subroutine calls written in SAIL code. During this time Robert, Filman suggested that it would be much better to hnvc a language that could be interpreted, so that simple changes to a font definition would not require recompiling a large program. During the summer of 1978, the author therefore used the experience gained while defining all the characters in proto-METAFONT to design a new language embodying the operations that had turned out to be necessary and desirable. METAFONT itself was programmed during the first part of 1979, and all of the Computer Modern character definitions were revised and rewritten in the new language during late 1979. The resulting programs appear in this report.

The design of Computer Modern is nearly complete, but some improvements will doubtless be made. In fact,, one of the goals of this report is to circulate the preliminary definitions in order to obtain critical comments before it is too late to make changes easily. The need to spread this information quickly explains the somewhat paradoxical fact that 1978 Cornputer Modern fonts have been used to typeset this report, even though it describes the 1979 Computer Modern characters. With luck, the 1978 version of Computer Modern fonts will disappear from the faces of the earth by the spring of 1980, when they will he replaced by the final form of the designs appearing here. The author hopes to produce a properly typeset book describing METAFONT and Computer Modern when a "steady state" is achieved.

Since the publication of the METAFONT manual, the language has changed in one respect (incorporated into the programs here): The height of lpens and rpcns is now specified independently by lpenht and rpenht statements. Formerly the hpenht statement was used for all three heights, but this turned out to be an unnecessary restriction.

Appendix E of the METAFONT manual was written when only two of the Computer Modern letters had been defined—namely, the "A" and the "B" used as examples in that appendix. The introduction to the present report is essentially a revised version of Appendix E, incorporating several dozen improvements and correcting some embarrassing errors.

—D.E.K., January 1980

CONTENTS

INTRODUCTION

A complete font design is a complex system, so there arc several levels at which one might understand it and use it depending on how much of the "black box" is being opened. At the outermost level, all of the details can be left alone and we simply generate a particular font. For example, there is a file called "cmr10 mf", and when METRFONT is applied to that file it will produce the "Computer Modern Roman 10 point" font. Another file "cmsss8 mf" produces "Computer Modern Slanted Sans Serif 8 point," and so on. But if we actually look inro files like cmr10. mf and cmsss8. mf, we find that they are quite short; they merely set up the values of certain parameters and input the file "roman. mf", which specifics the actual METAFONT programs for individual letters. Thercfort: it is easy to make up a customized font for a particular application, simply by setting up new values of the parameters and Inputting roman mf ourselves.

At a still deeper level, we can also look at the file roman mf, which provides 128 short programs for the individual character shapes (followed by ligature and kerning definitions). These short programs are fairly independent, and they aren't completely inscrutable; it isn't difficult to substitute a new routine or two for characters that we wish to modify, since the programs make use of some fairly flexible subroutines that appear in file cmbase . mf.

At the deepest level, we could also fiddle with the subroutine definitions in cmbase. mf— and of course that would essentially amount to the creation of a new family of fonts.

In this report we shall study the Computer Modern fonts by wcrking c ur way in from the outermost level, then going back out again. File cmr10. mf looks !ike this:

"Computer Modern Roman 10 point";
$ph = \frac{250}{36}$; $px = \frac{160}{36}$; $pe = \frac{90}{36}$; $pd = \frac{70}{36}$;
$pb = \frac{20}{36}$; p o = $\frac{4}{36}$; $ps = \frac{20}{36}$; pa = .5(ph— pd);
p w = $\frac{9}{36}$; $pwi = \frac{27}{36}$; $pwii = \frac{32}{36}$; $pwiii = \frac{38}{36}$;
$pwiv = \frac{32}{36}$; $pwv = \frac{38}{36}$; aspect == 1.0;
p u = $\frac{20}{36}$; lcs = 1.075; ucs = 1.7; SC = 0; Is == 0;
slant = 0; *sqrttwo* == sqrt 2; *fixwidth* == 0;
halfd = 0; *varg* == 0; *lowast* = *0* ; ligs = 1.
input **cmbase**; call *fontbegin*;
input **roman**;
end.

In other words, the file sets up a lot of parameters and then it does "input roman" to create the font.

We can obtain a great variety of related fonts by setting these parameters in different ways, once we know what they mean; and here's what they mean:

By convention, all of the parameters whose name begins with "*p*" are in units of printers' points. First come eight parameters covering important vertical dimensions:

ph is the h-height, the distance from the baseline to the top of an "h".

px is the x-height, the distance from the baseline to the top of an "x".

pe is the e-height, the distance from the baseline to the bar of an "e".

pd is the descender depth, the distance from the baseline to the bottom of a "p".

pb is the border height; characters extend as much as $ph +$ pb above the baseline and pd + pb below it.

po is the amount of overshoot for optical adjustments at sharp corners; e.g., "A" is this much taller than "B".

ps is the vertical distance al which serif bracketing is tangent Lo the stems.

pa is the axis height, the distance from the baseline to the point where mathematical symbols like "+" and "=" have vertical symmetry.

Then there arc seven parameters affecting the pen sizes:

pw i s the hairline width, used in the thinnest parts o f letters

pwi is the stern width, used for the vertical strokes in an "h".

pwii is the curve width, used in an "o" at its widest point.

pwiii is the dot width, the diameter of the dot on an "i".

pwiv is the upper-case stem width, used for the vertical strokes in an "H".

pwv is the upper-case curve width, u&d in an "O" at its widest point.

aspect is the ratio of a hairline pen's height to its width.

ĩ 'ext come five parameters concerning horizontal dimensions:

pu is the unit width, 1/18 of an em

lcs is the amount by which serifs of lower-case letters project from the stems, in units of *pu*.

ucs is the arnount by which serifs of upper-case letters project from the stems, in units of *pu*.

sc is the serif correction in units of *pu*; each letter specifies rrultiples of sc by which its width is to be decreased at the left and the right.

ls is the amount of letter spacing in units of *pu*; each lettei is made this many units wider than the design actually specifies.

Finally we have miscellaneous parameters that control special effects:

slant is the amount of additional increase in x per unit increase in y, usctl to slant letters either forwards or backwards.

sqrttwo is used to control the ellipticity of the bowls of letters, as explained in Chapter8 of the METAFONT manual.

halfd is nonzero if certain clraracters like "," are to descend only half as fai as lower-case letters do.

varg is nonzero if the simple "g" shape is to replace the classical "g".

lowast is nonzero if the asterisk is to be lowercd so that it is centered on the axis.

ligs is nonzero if the roman font is to have the character set that TEX expects for text fonts with ligatures. (Otherwise eighteen special symbols are substituted for the ligature-oriented characters.)

File cms10. mf ("Computer Modern Slanted LO point,") is exactly the same as file cmr10 mf, except for its title and the fact that ***slant = 0.15***. Similarly, the settings of parameters in file cmb10 mf ("Computer Modern Bold 10 point") arc nearly identical to those of cmr10 mf, except that the pens are bigger:

$$pw = \tfrac{15}{36};\quad pwi = \tfrac{10}{36};\quad pwii = \tfrac{15}{36};\quad pwiii = \tfrac{50}{36};$$
$$pwiv = \tfrac{50}{36};\quad pwv = \tfrac{50}{36};$$

furthermore serifs arc shorter (lcs = .85, ucs = 1.5).

File c m r 5 m f generates 5-point type, but it is not simply obtained by halving the parameters of cmr IO The eight vertic ill dimensions $ph, px, , pa$ are exactly half as large as before, but the pen sizes and the horizontal dimensions get smaller at different rates so as to enhance the readability of such tiny letters. The following settings arc used:

$$pw = \tfrac{7}{36},\quad pwi = \tfrac{17}{36},\quad pwii = \tfrac{19}{36};$$
$$pwiii = \tfrac{20}{36},\quad pwiv = \tfrac{19}{36},\quad pwv = \tfrac{20}{36};$$
$$pu = \tfrac{12.5}{36},\quad lcs = 0.84,\quad ucs = 1.32.$$

Two more examples should suffice to illustrate the variation of these parameters. The bold sans-serif font used in this sentence is called "Computer Modern Sans Serif 10 point Bold Extended" (cmssb). It uses the same vertical dimensions and miscellaneous

settings as cmr10, and gets its other c'aracteristics from the following parameter values:

$$pw = pwi = pwii = pwiii = \tfrac{37}{36};$$
$$pwiv = pwv = \tfrac{42}{36};\quad aspect = \tfrac{23}{37};$$
$$pu = \tfrac{22}{36};\quad lcs = ucs = 0;\quad sc = \tfrac{9}{22};\quad ls = 0.$$

To get the typewriter font "cmtt" used in this sentence, set

$$ph = \tfrac{210}{36};\quad px = \tfrac{150}{36};\quad pe = \tfrac{75}{36};\quad pd = \tfrac{80}{36};$$
$$pb = \tfrac{30}{36};\quad po = \tfrac{4}{36};\quad ps = 0;\quad pa = .5ph;$$
$$pw = pwi = pwii = pwiv = pwv = \tfrac{20}{36};$$
$$pwiii = \tfrac{30}{36};\quad \mathbf{aspect} = 1.0;$$
$$pu = \tfrac{23}{36};\ lcs = \tfrac{11}{23};\ ucs = \tfrac{25}{23};\quad sc = 0;\quad ls = 0;$$
$$slant = 0;\quad sqrttwo = \mathrm{sqrt}\ 2,\quad fixwidth = 1;$$
$$halfd = 1;\quad vnrg = 0\ lowast = 1;\quad ligs = \mathbf{0}.$$

By making stranger settings of the paramotors you can also got strangor fonts liko this. Tho font dofinitions aro not ablo to produco satisfactory rosults for all possiblo combinations of paramotor sottings, but thoy do try to handlo all "roasonablo" variations.

The programs for Computer Modern can be usedin several ways. The general procedure is to run **METRFONT** and type

```
mode = (mode number); input (font name);
```

the routines will act differently depending on the specified mode. At prcsrnt mode 0 generates proof sheets and shows the letters as they are being drawn, with a resolution of 36 pixels per point; mode 1 generates a font for the XGP or Versatec or Varian, etc., with a resolution of 3.6 pixels per point; modr 2 generates a font for the CRS with a resolution of 73.7973 pixels per point, displaying the titles of the letters as they are being drawn; mode 3 is like mode l but for a Dover; and other modes cause a filr **mode mf** to be read in, where this file specifies METAFONT's mode of oprration. In mode 0 the letters appear on a background grid as shown in the illustrations of this report, so that you can see the settings of the parameters in a convenient. way. If you wish to see the characters as they are being drawn, you can type "chardisplay; *mode* = I, input (font name)", etc.

All of the illustrations of individual characters in this report were generated wit Ir mode 0, except for the characters of the math extension font; these were done in a similar way but only half size (with 18 pixels per point), because so many of the characters of this font are quite large.

Actually mode 0 is rarely used with an entire font like cmr10, it is generally used only to test out new characters. In that case you can make up a file called "test mf" containing the characters you wish to try, and simply input the system file "proof mf", which has the following form:

mode = 0; input cmbase;
p h = $\tfrac{250}{36}$; . . . (set up for **cmr10**) . . . ; call fontbegin.
input test;

```
new pw, . . (set up for cmb10⟩ . . . ;    call fontbegin.
input test;
new pw, . . (set up for cmssb) . . . ;    call fontbegin.
input test;
new.ph, . . (set up for cmtt). . . ;    call fontbegin.
input test;
new ph, . . . (set up for cmsss8⟩ . . ;    call fontbegin.
input test;
end.
```

Thus, it runs your test file against live different settings of the parameters.

The programs for individual characters in this report fall into four main groups: First comes the roman group, which makes text fonts either with ligatures (like cmr10) or without (like cmtt). Then comes the italic group, which is somewhat similar, but it either makes text fonts with ligatures (like cmti10) or italic fonts for mathematical formulas (like cmi10). The third group is called symbol, and it makes math symbols (like cmsy10). Finally there is the mathex group, for extended math symbol fonts (like cmathx).

Let's go one level deeper and take a look at the programs for individual lctters. Such programs are expressed in terms of variables something like the parameters we have been discussing, but the variables are slightly different since the letters are to be drawn on a raster and we need to work in raster units instead of printers' points. The point-oriented variables ph, px, pe, etc., have corresponding raster-oriented variables, satisfying the approximate relation

$$\text{(raster-oriented variable)} \approx pixels\cdot\langle\text{point-oriented variable}),$$

where pixels is the number of pixels per point. This relation is only approximate, not exact, because the raster-oriented variables have been rounded to values that help to provide satisfactory discretization of the characters. As explained in Chapter 7 of the manual, good designs are written with discreteness in mind, although METAFONT tries to do the right thing automatically when it can.

There arc seven raster-oriented variables corresponding to seven of the eight pixel-oriented vertical dimensions, namely

$$h \leftrightarrow ph,\ \mathrm{m} \leftrightarrow px,\ e \leftrightarrow pe,\ d \leftrightarrow pd,\ b \leftrightarrow pb,\ o \leftrightarrow po,\ \mathrm{a} \leftrightarrow \mathrm{pa};$$

in other words, we just drop the "p", except in the case of "px" (since a variable can't be named "x"). Variable m is used lo stand for the x-height, since a line al this height is traditionally called the "mean line." The baseline of each character is row 0, so the bottom pixel of a letter like "h" has y-coordinate 0. The lop pixel of an "h" is in row h, which is always an integer. (Note that there arc actually $h+1$ occupied rows, not h, although h is called the h-height.) The top pixel of an "m" is in row m, and the bottom pixels ol the descender letters (g,j,p,q,y) appear in row $-d$. All three of these variables (h, m, d) are integers, and so is the overshoot variable o (which is used as a correction to h, m, or d in certain cases). Variable e is either an integer or an integer plus $\frac{1}{2}$, whichever is better for a pen of the hpen height, since the bar of an "e" is drawn with an hpen and its y-coordinate is e. Variable b is an integer calculated in such a way that tall characters can run up to row $h + b$ and deep characters can descend to row -d $- b$; more precisely, it is the smallest

i..teger such that h + d + $2b$ + 1 rows of the raster occupy a vertical distance that exceeds or equals the true point size $ph + pd$ + 2pb.

The pen sizes in Computer Modern programs for individual letters arc generally expressed in terms of the following variables, each of which has a positive integer value intended to approximate thk "true" infinite-resolution value (and slightly increased in order to look right on the output device, depending on the current mode):

w_0, the hairline width;
w_1, the stem width;
w_2, the curve width;
w_3, the dot diameter;
w_4, the upper-case stem width;
w_5, the upper-case curve width;
w_6, the hairline height;
w_7, the stem height;
w_8, the curve height;
w_9, the upper-cast stem height;
w_{10}, the rule thickness for math symbols.

Note that the last five of these variables have no "p-variable" equivalent; they satisfy the approximate relations

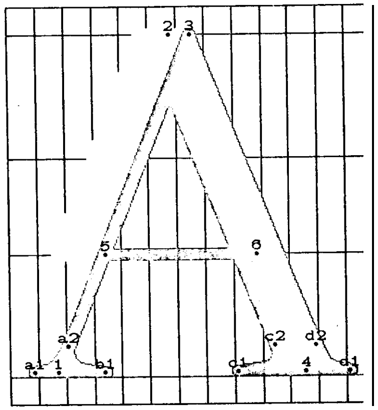
$$w_6/w_0 \approx w_7/w_1 \approx w_8/w_2 \approx w_9/w_4 \approx aspect; \quad w_{10} \approx .25[w_0, w_2].$$

The hpenht, lpenht, and rpenht arc w_6, and the vpenwd is w_0. Thus, an hpen of size w_0 is equivalent to a vpen of size w_6; we may call it the "hairline pen" for the font.

In the horizontal dimension, the Computer Modern programs make frequent use of variable u, the approximate unit width when there arc 18 units to an em. The width of a character is cxpressrd in terms of units (e.g., an "h" is $10u$ wide, unless there is a serif correction sc $\neq$ 0 or some additional letter spacing ls $\neq$ 0), and key positions can be specified as a certain number of units from the left (e.g., the sterns of an "h" are centered at $2.5u$ and 7.521). The vertical guidelines in the illustrations of this report are one unit apart.

If the character is t units wide, variable u has been calculated so that t times u is an integer r, the rightmost column of the character. (The value of u itself is usually not an integer, nor need t be an integer.) Just as a character typically occupies rows 0 through h, inclusive, in the vertical direction, we use columns 0 through r inclusive in the horizontal direction, although most characters leave while space at the left and right boundaries. The integer r is calculated so that, in the absence of corrections due to serifs or letter spacing, etc., $r+2$ is the nearest integer to the character's true width $(t\cdot pu\cdot pixels)$; lhc reason for this extra "$+2$" is that, low-resolution devices should keep a blank column (column $r+1$) between adjacent characters. However, it is best for conceptual purposes to think of r as the character's actual width, and to think of "$r - 2.5u$" as a point $2\frac{1}{2}$ units from the right edge, etc.

Variable oo is an integer approximation to one-half of o; it is used for "half of an overshoot," when curved lines approach the bottom or the top of a character. Variable dd is like d except it is only half as large, when $halfd$ is nonzero. A few other variables like this are defined for use in several different characters, either in file cmbase or at the beginning of a file containing character definitions; but the variables described above are by far the most important.



Upper-case “A” defined by the example program.

We’re ready now to look rnore closely at a program for the upper-case letter “A” (see the boxed example on page 13). The first line of that program simply gives the title that will appear on proof sheets, or possibly on the terminal when the character is being drawn. Then comes a call to the *charbegin* subroutine, with seven **parameters**: the character code, the width of the character in units, the respective amounts in units that are to be trimmed from the left and from the right, and finally the character’s height, depth, and italic **correction**. **These** last three **parameters** must be in **absolute** units of printers’ points, hence *ph* (not *h*) must be used for the height.

The next. few lines givc eight equations lo define the locations of points 1, 2, 3, and 4. First point 1 is positioned so that, using an hpen of size w_0 (the hairline pen), the pen’s left edge will be 1.5 units from the left edge of the character, and the bottom will be on the **baseline**. Similarly point 4 is placed so that the pen’s right edge will be 1.5 units from the right edge of the character and the bottom will be on the baseline, where this time the pen is an hpen of size w_5. (The upper-case curve width w_5 is used here in preference to the stem width w_1, since a diagonal stroke tends to decrease the effective pen width.) The positioning of points 2 and 3 is more interesting: the idea is that we want to draw a line from 2 to 4 with an hpen of width w_5, and another from 3 to 1 with an hpen of width w_0. First. we define y_2 and y_3 so that the top occurs at the h-height h, plus the “overshoot” o that givrs this letter a touch of class. Then we state that $x_3 - x_1 = x_4 - x_2$, so that the two diagonal strokes will have the same slopes (the same arnount of change in the x direction). Finally we stipulate that $\mathrm{rt}_5 x_2 = \mathrm{rt}_0 x_3$, so that the line from 2 to 4 will have the same top right boundary as the line from 1 to 3. These equations give METAFONT enough infornation to determine points 2 and 3 uniquely.

After drawing the right diagonal stroke, we need to erase part of the stem line at the top, where it protrudes to the left of the left stroke (which is ‘thinner). Befoie erasing anything,

```
"The letter A";
call charbegin(`A, 13, 2sc,2sc, ph, 0,0);
hpen;
lft0x1 = round 1.5u;   bot0y1 = 0;
rt5x4 = round(r — 1.5u);   bot5y4 = 0;
top0y3 = top5y2 = h + 0;
x3 — x1 = x4 — x2;   rt5x2 = rt0x3;
w5 draw 2.. 4;                                   % right diagonal strol
y5 = y6 = e;
new aa, bb;                     % auxiliary variables for intersection of lin
x5 — 1 = aa[x1, x3];   y5 = aa[y1, y3];
xi, + 1 = bb[x1, x2];   y6 = bb[y1, y2];
w0 draw 5. .6;                                                 % bar li
lpen#;   w5 draw 3 . 5;                       % erase excess at upper le
hpen;   w0 draw 3 1;                                % left diagonal strol
if ucs ≠ 0:
     call . a serif (I, 0, 3, —.5ucs);
     call ` b serif (1, 0, 3,+ucs);                           % left, ser
     call . c serif (4, 5, 2, —ucs);
     call ` d serif (4,5, 2, +.5ucs);                        % right, ser
fi.
```

A METAFONT program for upper-case “A”.

however, we may as well draw the bar line. Computer Modern fonts place this line at the e-height, the same level as the bar line in an “e”, hence $y_5 = y_6$ = e. The calculation of x_5 and x_6 is slightly trickier; x_5 lies between x_1 and x_3, and the ratio of its distance is the same as the ratio for y_5 with respect to y_1 and y_3. The equations “$x_5 = aa[x_1, x_3]$; $y_5 = aa[y_1, y_3]$” would almost surely work to define a suitable point; but the prograrn actually uses $x_5 - 1$ instead of x_5, just to be absolutely safe against weird possibilities of rounding that might cause the bar line to stick out at the left. (It doesn’t hurt to start a line one pixel to the right of a point, that, lies on another line.)

Now the **lpen#** is used Lo erase unwanted black pixels, changing them back to white. Actually this erases more than we wanted to get rid of, since it has a rectangular shape and we are erasing at an angle; but that doesn’t matter, because the left diagonal stroke blackens all the nccrssary pixels. (Note that the eraser also does away with part of the guidelines in the proof drawing of the figure)

Finally the *serif* subroutine is used lo attach fancy serifs at points 1 and 4; these serifs extend .5*ucs* units outwards and *ucs* units inwards Details of this subroutine appear below.

Once you understand this program for “A”, you will have no trouble writing programs for “V” and “v”, as well as for the Greek letter “,A”; and you will be well on your wny Lo having “M” and “W”. Similarly, the code below for “B” leads t o “D” and “P” with little further ado.

We shall now plunge into the deepest level, the subroutines in cmbase mf that Lake care of nasty details. These subroutines are presented on the following pages for reference purposes; it’s probably best not to read them until you have to.

The file cmbase.mf

```
eps = .000314159;                              % a very small random positive number
if mode = 0:   proofmode; drawdisplay; titletrace;
     pixels.= 18; blacker = 0;
else:   if mode = I:   fntmode; tfxmode; no modtrace;
          pixels = 3.6;   blacker = 1.2;               % XGP, Versatec, Varian, etc.
     else:   if mode = 2: crsmode; tfxmode; titletrace; no modtrace;
               pixels = 73.7973;   blacker = 1;                      % Alphatype
          else:   if mode = 3: fnlrnodc;   tfxmode;   no mod trace;
                    pixels = 3.6 *(1.1/1.3)*(381/200);   blacker = 1.2;   % Dover
               else:   input mode;
               fi;
          fi;
     fi;
fi.


subrout inc fontbegin :                             % Initialize before making a font:
no eqtrace;                                    % Turn off tracing within this subroutine
new typesize;                                          % the vertical size of the font
new cf;                           % conversion factor, approximately equal to pixels
new h, d, dd, m, e, o, oo, b, s, a;                 % raster-oriented vertical dimensions
new w0, w1, w2, w3, w4, w5, w6, w7, w8, w9, w10, w11;       % raster-oriented pen sizes
new deltaw, bold;                          % unrounded raster-oriented pen size values
new armic, lcic;                                     % italic corrections commonly used
new lcss;                                                    % lower case short serif
new prt;                                                   % rule thickness in points
w0 = round(pixels·pw + blacker);                                    % hairline width
w1 = round(pixels·pwi + blacker);                                      % slern width
w2 = round(pixels·pwii + blacker);                                     % curve width
w3 = round(pixels·pwiii + blacker);                                   % dot diameter
w4 = round(pixels·pwiv + blacker);                             % uppercase stem width
w5 = round(pixels·pwv + blacker);                             % uppercase curve width
w6 = round(pixels·pw·aspect + blacker);                            % hairline height
w7 = round(pixels·pwi·aspect + blacker);                               % stem height
w8 = round(pixels·pwii aspect + blacker);                             % curve height
w9 = round(pixels·pwiv·aspect + blacker);                   % uppercase stem height
prt = .25[pw, pwii];                                       % rule thickness in points
w10 = round(pixels·prt + blacker);                   % raster-oriented rule thickness
w11 = round(pixels(.2[pw, pwi])·aspect + blacker);            % hairline plus a little
deltaw = pixels·(pwii -- pwi);                              % one step of boldening
bold = .5[pwii, pwiii]·pixels + blacker;
```

```
% The following corrections are for low resolution:
if w3/w1 > 3/2(pwiii/pwi):   new w2, w3;   w2 -- w3 = w1;
fi;
i f w5/w4 > 3/2(pwv/pwiv): new w5; w  j = w4;
fi;
if w8/w7 > 3/2(pwii/pwi): new w8;   w8 = w7;
fi;
hpenht w6; vpenwd w0; lpenht w6; rpenht w6;
typesize = ph + pd + 2pb;   cf·typesize = pixels·typesize — 1;
h = round cf·ph;   d = round cf·pd;
new pdd;   pdd = (1 —.5halfd)pd;   dd = round cf·pdd;
m = round cf·px;
o = round cf·po;   00 = round .5cf·po;   s = cf·ps;
a = .5 round 2cf·pa;
6 = - round( 5(h + d — typesize·pixels));
hpen;   e = good6 cf·pe;
maxht h + b + 2;
trxy slant;
if ucs ≠ 0:   armic = ph·slant + (s c --- 1)pu;
else:   armic = ph·slant + (s c — .5)pu;
fi;
if pwii > 1.5pu:   lcic = --.25pu;
else:   lcic = .5pwii --- pu;
fi;
if pw = pwi:   lcss = lcs;
else:   lcss = .5lcs;
fi.
```

```
subroutine charbegin(var charno)                     % seven-hi' character code
          (var charu w)                              % character width in units
      (var lftcorr, var rtcorr)                      % serif-oriented corrections in units
(var charh, var chard, var chari):                   % charht, chardp, charic values in points
no eqtrace;  no calltrace;  no drawdisplay;          % no tracing in this subroutine
new uw, modu w;                                      % the correct character width in units
new r;                                               % raster-oriented character width
new u;                                               % raster-oriented design unit
new tu;                                              % unmodified raster-oriented unit
new italcorr;                                        % italic correction
new lcorr, rcorr;                                    % left and right corrections
if chari ≥ 0:   italcorr = chari;   else:   italcorr = 0;
fi;
if danger ≠ 0:                                       % rounding of character width is necessary
      lcorr = danger·round((lftcorr — ls)/danger);
      rcorr = danger. round((rtcorr – ls)/danger);
else:   lcorr = lftcorr — ls;   rcorr = rtcorr — ls;
fi;
tu = pu·pixels;   uw = charuw —(lcorr + rcorr);
if fixwidth = 0:   moduw = uw;
else:   moduw = 9; new italcorr;   italcorr = 0;
fi;
r = charuw·u = round(( moduw·tu — 2)·charuw/uw);
charcode charno;   charic italcorr;
if charh > 0:   charht charh;
else:   charht 0;
fi;
if chard > 0: chardp chard;
else:   chardp 0;
fi;
charwd moduw·pu;   chardw moduw·tu;
incx round(—lcorr·u);
if mode = 0:   call box(roundlcorr·u);
fi.
```

```
subroutine box(var offset):                  % Draw guidelines and box around a charnctcr:
no drawtrace;  no proofmode;
new topp, bott, left, right, pos;
topp = h + b;   bott = - d — b;
left = offset;   right = offset + u·uw;
x1 = x3 = x5 = x7 = x9 = x11 = x13 = x15 = x17 = left;
x2 = x4 = x6 = x8 = x10 = x12 = x14 = x16 = x18 = right;
y1 = y2 = 0;   cpen;   1 draw 1..2;                                    % baseline
y3 = y4 = e;   draw 3.. 4;                                             % e-height
y5 = y6 = m;   draw 5..6;                                              % mean line (x-height)
y7 = y8 = h ;   draw 7.. 8;                                            % h-height
y9 = y10 = topp;   draw 9.. 10;                                        % top of character
y11 = y12 = —d;   draw 11.. 12;                                        % descender line
y13 = y14 = bott;   draw 13.. 14;                                      % bottom of character
trxy 0;                                                                % temporarily turn off the slant
y15 = y16 = topp;   y17 = y18 = bott;
draw 15.. 17; draw 16.. 18;                                            % left and right edges
if italcorr > 0:   x19 = x20 = right + italcorr·pixels;
      y19 = topp; y20 = 0;  draw 19.. 20;                              % show italic correction
fi;
trxy slant;                                                            % restore slanted transformation
pos = 0; call unitlines.                                               % draw the unit guidelines

subroutine unitlines:                        %Recursive subroutine to draw guidelines:
x1 = x2 = pos;   y1 = topp;  y2 = bott; cpen;
if pos ≥ left:   1 draw 1..2;
fi;
new pos;   pos = x1 + u;
if pos ≤ right: call unitlines;
fi.
```

% The following subroutines arc used to draw common features of characters.

```
subroutine $serif$(index $i$)                              % point where serif appears
          (index k)                                        % w-variable for stem line
          (index $j$)                                 % another point on the stem line
          (var $sl$):                                                % serif length
$y_1 = y_i$;
if $y_i < y_j$:   $y_2 = \gamma_i + s$;
else:   $y_2 = y_i - s$;
fi;
 hpen;
if $sl < 0$:    $\mathrm{lft}_0 x_1 = \mathrm{lft}_k x_i + sl\cdot u - eps$;
     $\mathrm{lft}_0 x_2 = \mathrm{lft}_k(y_2 - y_i)/(y_j - y_i)[x_i, x_j]$;
else:   $\mathrm{rt}_0 x_1 = \mathrm{rt}_k x_i + sl\cdot u + eps$;
     $\mathrm{rt}_0 x_2 = \mathrm{rt}_k(y_2 - y_i)/(y_j - y_i)[x_i, x_j]$;
fi;
no proofmode;
$x_3 = \frac{1}{3}[x_1 - sl\cdot u, \frac{1}{2}[x_1, x_2]]$;
$y_3 = \frac{1}{3}[y_i, \frac{1}{2}[y_1, y_2]]$;
minvr 0; minvs 0;
$w_0$ ddraw $1\{x_i - x_1, 0\}\,..\,3\,..\,2\{x_j - x_i, y_j - y_i\}$, 1.. 1 .. $i$;
rninvr 0.5;   minvs 0.5.


subroutine darc(index $i$)                                     % starting point
          (index $j$)                                    % opposite corner point
          (var maxwidth):                  % the pen grows from $w_0$ to this Size
$x_5 = x_i$;   $x_2 = x_4 = 1/sqrttwo[x_i, x_j]$;   $x_3 = x_j$;
$y_5 = y_j$;   $y_3 = \frac{1}{2}[y_i, y_j]$;
$y_2 = 1/sqrttwo[y_3, y_i]$;   $y_4 = 1/sqrttwo[y_3, y_j]$;
hpcn;   draw $|w_0|i\{x_3 - x_i, 0\}\,.\,|\frac{2}{3}[w_0, maxwidth]|2\{x_3 - x_i, y_3 - y_i\}$ . .
     $|maxwidth\#|3\{0, y_3 - y_i\}$ . .
     $|\frac{2}{3}[w_0, maxwidth]|4\{x_5 - x_3, y_5 - y_3\}\,.\,|w_0|5\{x_5 - x_3, 0\}$.


subroutine arc(index i)                                    % horizontal endpoint
          (index $j$)                                        % vertical endpoint
          (var maxwidth):                  % the pen grows from $w_0$ to this size
$x_1 = 1/sqrttwo[x_i, x_j]$; $y_1 = 1/sqrttwo[y_j, y_i]$;
hpen;   draw $|w_0|i\{x_j - x_i, 0\}\,.\,|\frac{2}{3}[w_0, maxwidth]|1\{x_j - x_i, y_j - y_i\}$ .
     $|maxwidth|j\{0, y_j - y_i\}$.
```

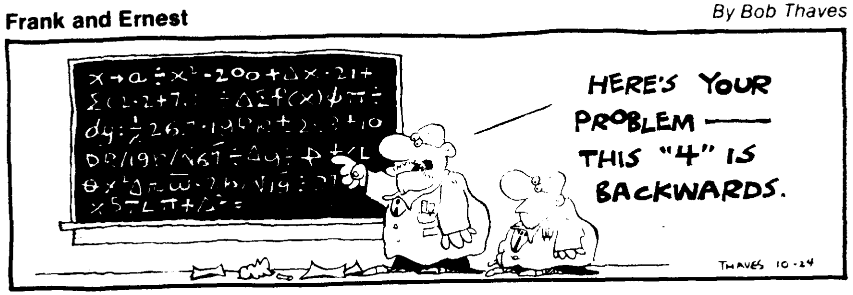
```
subroutine $arm$(index $i$)                                    %: starting point
          (indexj)                                         % horizontal endpoint
          (index k):                                            % serif endpoint
hpen;
if $ucs \neq 0$:
     if $w_0 = w_4$:   $x_1 = x_2 = x_k$;   $y_1 = y_j$;
          if $y_k < y_j$:   $y_2 = y_j - aspect\cdot ucs\cdot u$;
          else:   $y_2 = y_j + aspect\cdot ucs\cdot u$;
          fi;
          $w_0$ draw $i\,..\,1$; draw $1\,..\,2$;
     else:  minvr 0; minvs 0;
          $x_1 = x_j - (x_k - x_j)$;   $y_1 = .3[y_j, y_k]$;
          $x_2 = x_k$;   $y_2 = 2[y_j, y_k]$;
          $w_0$ ddraw $i\,..\,j\,..\,k, i\{x_j - x_i, 0\}\,..\,1\,..\,k(\,..\,2)$;
          minvr 0.5;   minvs 0.5;
     fi;
else:   $x_1 = x_k$;   $y_1 = y_j$;   $w_0$ draw $i\,..\,1$;
fi.


subroutine $scomp$(index $i$)                                   %starting point
          (index p)                           % turning point (y,, to be defined)
          (index $j$)                        % transition point (to be defined)
          (index $k$)                                             % ending point
          (var slope):                                            % ending slope
% This subroutine computes $y_p, x_j, y_j$ s o that $y_k - y_j =: slope.(x_k\,-\,-\,x_j)$
% and so that the following curve is consistent with an ellipse:
% $i\{x_p - x_i, 0\}\,.\,.\,p\{0, y_p - y_i\}\,j\{x_k - x_p, slope.(x_k -$ x,,)}.
$y_k - y_j = slope(x_k - x_j)$;
new $aa$, bb;   $aa = slope(x_p - x_i)$;   $bb = y_k - y_i - slope(x_k - x_i)$;
$x_j - x_i = -2aa\cdot bb(x_p - x_i)/(aa\cdot aa + bb\cdot bb)$;
$y_p - y_i = .5(bb\cdot bb - aa\cdot aa)/bb$.
```

```
subroutine sdraw(index i)                                  % starting point
        (index p)                % upper turning point (y_p to be defined)
        (index k)                                            % middle point
        (index q)                % lower Lurning point (y,, to be defined)
        (index j)                                            % ending point
        (var perrwd)                       % effective width of hpen used
        (var penht)                      % effective pen height at point k
        (var slope):                                  % slope at point k
new w_48, w_49;   w_48 = penht;   w_19 == pen wd;
cpen;   top_6 y_5 = top_48 y_k;   bot_6 y_6 == bot_18 y_k;   x_5 == x_6 = x_k;
if x_p < 5,:   rt_19 x_p = rt_0 x_1;   lft_19 x_p = lft_0 x_2;   rt_49 x_q = rt_0 x_9;   lft_49 x_q = lft_0 x_10;
else: lft_49 x_p = lft_0 x_1;   rt_19 x_p = rt_0 x_2;   lft_49 x_q = lft_0 x_9;   rt_19 x_q = rt_0 x_10;
fi;
y_2 = y_p;   y_9 = y_q;
call scomp(i, 1, 3, 5, slope);                    % compute y_1 and point 3
call scomp(i,2, 4, 6, slope);                     % compute y_2 and point 4
call scomp(j,9, 7, 5, slope);                     % compute y_9 and point 7
call scomp(j,10, 8, 6, slope);                   % compute y_10 and point 8
hpen;   w_0 ddraw i{x_1 — x_i, 0 } 1{0, y_1 -- y_i} 3{x_q — x_p, slope(x_q — x_p)} .
        7{x_q — x_p, slope(x_q — x_p)} . .
        9{0, y_j — y_9} . j{x_j — x_9, 0},
    i{x_2 — x_i, O} . . 2{0, y_2 — y_i} . 4{x_q — x_p, slope(x_q — x_p)} . .
        8{x_q — x_p, slope(x_q — x_p)} . .
        10{0, y_j — y_10} . . j{x_j — x_10, 0}.                  % the s-curve

subroutine zcomp(index i)                                  % starting point
        (index p)                     % turning point (x_p to be defined)
        (index j)                 % transition point (to be defined)
        (index k)                                            % ending point
        (var slope):                         % reciprocal of ending slope
% This subroutine is dual to scomp.
% It computes x_p, x_j, y, so that x_k — x_j = slope·(y_k — y_j)
% and so that the following curve is consistent with an ellipse:
% i{0, y_p — y_i} . p{x_p — x_i, 0} . j{slope.(y_k — y_p), y_k — y_p}.
x_k — x_j = slope(y_k — y,);
new aa, bb;   aa = slope(y_p — y_i);   b b = x_k — x_i — slope(y_k — y_i);
y_j — y, = —2aa·bb(y_p -- y_i)/(aa·aa + bb·bb);
x_p — x_i = .5(bb·bb -- aa·aa)/bb.
```

```
subroutine zdraw(index i)                                  % starting point
        (index p)                  % left turning point (x_p to be defined)
        (index k)                                            % middle point
        (index q)                 % right turning point (x_q Lo be defined)
        (index j)                                            % ending point
        (var penht)                       % effective height of hpen used
        (var pcnwd)                      % effective pen width at point k
        (var slope) :                    % reciprocal of slope at point k
% This subroutine is dual to sdraw.
new w_18, w_49;   w_48 = pen wd;   w_49 = penht;
cpen;   rt_0 x_5 = rt_18 x_k;   lft_0 x_6 = lft_48 x_k;   y_5 = y_6 = y_k;
if y_p > y_i:   Lop_19 y_p = top_6 y_1;   bot_19 y_p = bot_6 y_2;
    top_19 y_q = top_6 y_9;   bot_49 y_q = bot_6 y_10;
else:   bot_19 y_p = bot_6 y_1;   top_19 y_p = top_6 y_2;
    bot_19 y_q = bot_6 y_9;   top_19 y_q = top_6 y_10;
fi;
x_2 = x_p;   x_9 = x_q;
call zcomp(i, 1, 3, 5, slope);                    % compute x_1 and point 3
call zcomp(i, 2, 4, 6, slope);                    % compute x_2 and point 4
call zcomp(j,9, 7, 5, slope);                     % compute x_9 and point 7
call zcomp(j, 10, 8, 6, slope);                  % compute x_10 and point 8
hpen;   w_0 ddraw i{0, y_1 — y_i} . . 1{x_1 — x_i, 0} . 3{slope(y_q — y_p), y,, — y,,}.
        7{slope(y_q — y_p), y_q — y_p}
        9{x_j — x_9, 0} j{0, y_j -- y_9},
    i{0, y_2 — y_i} . 2{x_2 -- x_i, 0} . 4{slope(y_q -- y,,), y_q -- y,,}.
        8{slope(y_q — y_p), y_q — y_p} . .
        10{x_j — x_10, 0} . j{0, y_j — y_10}.                    % the s-curve

subroutine bar(index i, index j):
% This subroutine is similar to "vpen;  w_9 draw i . . j",
% but the vpcn slants with italic.
no proofmode;                      % the points computed aren't interesting
vpen;   top_6 y_1 = top_9 y_i;   bot_6 y_2 = bot_9 y_i;
top_6 y_3 = top_9 y_j;   bot_6 y_4 = bot_9 y_j;
x_1 = x_2 = x_i;   x_3 = x_4 = x_j;
w_6 ddraw 1.. 3, 2 . . 4.

subroutine fstroke(index i)                                  % dot position
        (index j):                                % s-coordinate of stem
hpen;   x_1 == x,;   bot_1 y_j == 0;   y_1 == .5[m, h];
rt_0 x_3 == rt_3 x_i;   y_3 = y_i ;
x_2 = .5[x_3, x_1];   top_0 y_2 = h + oo;
draw |w_1|j . . |w_1#|1{0, I} . |w_0#|2{1, 0} . . 3{0, —1};   % shoulder and stem
cpcn;   w_3 draw i;                                                  % bulb
if lcs ≠ 0:   call `a serif(j,1, 1, —lcs);
    call` b serif (j,1,1, lcs);                                     % serif
fi.
```

```
subroutine hstroke(index i)                                    % x-coordir.te of left stem
        (index j)                                              % x-coordinate of right stem
        (indexk):                                              % will be set to base of right stern
hpcn;   x_k = x_j;   bot_1 y_k = 0;
rt_0 x_1 = rt_1 x_i; y_1 = 1/4[e, m]; y ,  = 1/3[e, m];
x_2 = .5[x_i, x_j];   top_3 y_2 = m + 00;
x_3 = 1/sqrttwo[x_2, x_j];   y_3 = 1/sqrttwo[y_j, y_2];
draw |w_0|1{0, I}. |w_0#|2{1,0} . . |.6[w_0, w_1]|3{x_j - x_2, y_j -- y_2} .
     |w_1#|j{0,-1} . . k.                                      % shoulder and stem

subroutine cdraw(index i, index j)                             % given points
        (intlcx p, index q):                                   % given widths, w_p ≥ w_q
% An implementation of the forbidden "cpen;  draw | w_ |i |w_q|j" . ·
cpen;  w_p draw i;                                             % plot the biggrr dot
new aa;  (aa + eps) sqrt((x_j - x,)(x, - x_i) + (y_j - y_i)(y_j - y_i)) = w_p - w,,;
x_2 - x_1 = aa(y_i - y,);   y_2 - y_1 = aa(x_j - x_i);
x_i = .5[x_1, x_2]; y_i  =  .5[y_1, y_2];                      % perpendicular points
w_q ddraw 1 . . j, 2 . . j.                                    % fill in the rest

subroutine qcirc(index i)                                      % horizontal endpoint
     (index j)                                                 % intermediate point
     (index k)                                                 % vertical endpoint
     (var size):                               % size of cpen that draws a quarter circle
cpen;  x_j = 1/sqrttwo[x_i, x_k];   y , = 1/sqrttwo[y_k, y_i];
size draw i{x_k - x_i, 0} . . j{x_k - x_i, y_k - y_i} . . k{O, y_k - y_i}.

subroutine hcirc(index viii, index i, index ii, index iii, index iv, var size):
x_iv = x_viii;   y_ii == .5[y_iv, y_viii];
call qcirc(viii, i, ii, size);   call qcirc(iv, iii, ii, size).

subroutine circle(index i, index ii, index iii, index iv,
     index v, index vi, index vii, index viii, var size):
x_iv = x_viii = .5[x_vi, x_ii];   y_ii = y_vi = .5[y_iv, y_viii];
call qcirc(viii, i, ii, size);   call qcirc(iv, iii, ii, size);
call qcirc(iv, v, vi, size);   call q( irc( viii, vii, vi, size).

subroutine entry(var z)                                        % x-coordinate for upward stroke
     (index j):                              % z-coordinate for downward stroke (y, will be set)
% This subroutine draws a little hook at the beginning left of an italic character,
% ending with the pen traveline vertically at point j with size w_1.
hpen;  x_1 = good_0 z;  y_1 = 2/3 m;  y_j == 3/4 m;  x_2 = x_j - 1.5u;  top_0 y_2 = m + 00;
draw |w_0|1{(x_j -- 2.5u) - x_1, m} |w_0#|2{ 1,0} |w_1#|j{0, - 1 ) .
```

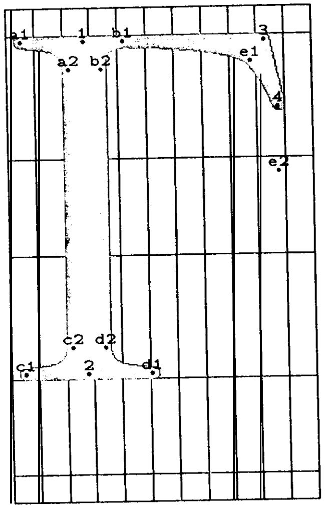
```
subroutine skewentry(var z)                                    % x-coordinate for upward stroke
     (index j):                              % z-coordinate for downward stroke (y_j will be set)
% This subroutine is analogous to entry, but the pen starts out vertical
% and ends at the skewed slope {-u, --m} to compensate for optical illusion.
hpen;  x_1 = good_0 z;  y_1 = 2/3 m;  y_j = 3/4 m;
x_2 = x_j - 1.25u;  top_0 y_2 = m + oo;
draw |w_0|1{0, 1} . . |w_0#|2{1, 0} . |w_1#|j{-u, - m } .

subroutine exit(index i)                     % x-coordinate for downward stroke (y_i will be set)
     (var z):                                                  % x-coordinate for upward stroke
% This subroutine draws a little hook at the ending right of an italic character,
% beginning with the pen traveling vertically at point i with size w_1.
hpcn;  x_2 = good_0 z;  y_2 = 1/3 m;  y_i = 1/4 m;  x_1 = x_i + 1.5u;  bot_0 y_1 = -oo;
draw |w_1#|i{0, -1} . |w_0#|1{ 1, 0} 2{x_2 - (x_i + 2.5u), m } .

subroutine skewexit(index i)                 % z-coordinate for downward stroke (y_i will be set)
     (var z) :                                                 % x-coordinate for upward stroke
% This subroutine is analogous to exit, but the pen begins with the skewed
% slope {-u, -m} to compensate for optical illusion, and ends vcrticnlly.
hpen;  x_2 = good_0 z;  y_2 = 1/3 m ;  y_i = 1/4 m;  x_1 = x_i + 1.2521;  bot_0 y_1 = - 0 0 ;
draw |w_1#|i{-u, -m} . |w_0#|1{1, O} . . 2{0, 1}.

subroutine italhstroke(index i)                                % starting point
        (index j):                             % s-coordinate of right stern (y, will be set)
hpen;  x_1 = .6[x_i, x,];  x_2 == x_j - .4u;  top_0 y_1 = m  + oo;  y_2 == .75[e, y_1];
y , = .3[e, m ] ;
draw |w_0|i{0, 1} . |w_0#|1{1, 0} . |.75[w_0, w_1]|2 . . |w_1#|j{0, - 1 ) .

subroutine pistroke:                             % makes the bar of pi, tau, variant omega
vpen;  x_1 = good_0(0);  y_1 = m - m/3. 14159;
x_2 = 2u;  top_7 y_2 = m  ; y_3 = y_2; x_3 = r - - 1.5u;
draw |w_6#|1{x_2 - x_1, 3.14159(y_2 - y_1)} . |w_7#|2{1, 0} . . 3;                   % bar
cpen;  w_7 draw 3.                                             % make the end point round

subroutine endv(index i):                          % draws final bulb starting at this point
cpen;  x_1 == x, -- u;  x_2 == x_i --- 6u;  top_1 y_1 -- m + 0 0 ;  y_2 = y_1;
hpen;  draw |w_0#|i{0, 1} . |w_1#|1( . . 2);                                      % stroke
cpen;  w_1 draw 1                                                                  % bulb

subroutine max(var a, var 6):                                  % sets acc = max(a, b)
new acc;
if a > b:  acc = a;
else:  acc = b;
fi.
```

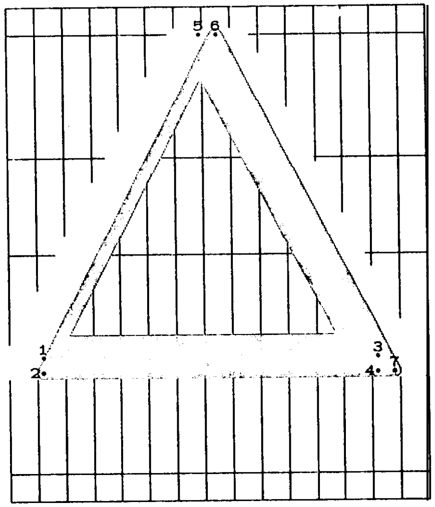


ROMAN CHARACTEr DESIGNS

The file roman mf

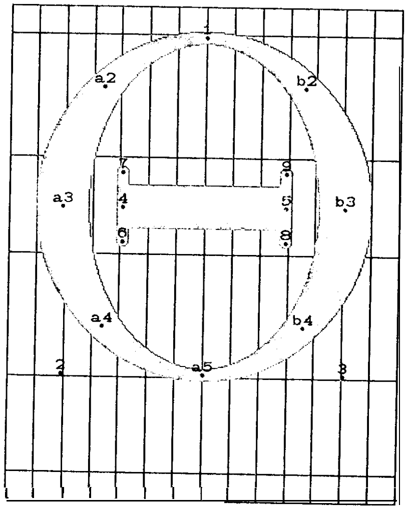
```
% The Computer  Modrn Roman family of fonts (by D. E. Knuth, 1979)
danger == mi = 0;
input romitu;                                           % upper case (majuscules)
input roman 1;                                          % lower case(minuscules)
input romand;                         % numerals, ampersand, and question mark
input romans;                                  %miscellaneousletter combinations
input romi  tp;               % punctuation symbols common lo allroman and italic
input romita;             % accents and other symbols common to roman and italic text
if ligs ≠ 0:  input roml ig;                     % letter ligatures (codes ' I73 -' 177)
    input romitl;                                      % ligatures common with italic
    input  romext;                      % nonstandard characters (codes '043, '044)
else:  input romits;                                      % substitutes for ligatures
fi;
if fixwidth = 0:   new k, kk, kkk;                          % three degrees of kerning
    k == —.5pu;   kk = —1.5pu;   kkk == —2.5pu;
    lig   'k:   `v: `w: `x: `y:
        `o kern k, . e kern k, `c kern k;
    lig  'F: `V:
        `A kern kkk, `o kern kk, `e kern kk,
        'a kern kk, `u kern kk, ` r kern kk,
     'K: 'X:
        ` 0 kern k, `C kern k, `G kern k, ` Q kern k;
    lig 'T:   `y kern kk,
    'Y:   `o kern kk, ` e kern kk,
        ` a kern kk, `u kern kk, .  r kern kk,
     'P:   ` W:   'A  kern kk;
    iig '0:   `A kern k, . W kern k, `Y kern k, `  V  kern k, `X kern k;
    if lcs ≠ 0: lig 'h:   'm:   'n:
        `t kern k, `u kern k, `b kern k,;
    fi;
    lig `o:
        . v kern k, `x kern k, .  y  kern k,
    `t:
        `w kern k:..,
    if   ucs ≠  0:  lig `R: ;
    fi;
    lig 'A:   `t  kernk, `L:
        ` T kern  kk, `O kern k, `U kern k, `C kern k,
        ` W  kern  kk, `Y  kern  kk, `G kern k, . V kern  kkk,
        . Q kern k;
    texinfo slant, 6pu, 3pu, 2pu, px, 18pu, 2pu;
else: texinfo slant, 9pu, 0, 0, px, 9pu, 9pu;
fi.
```

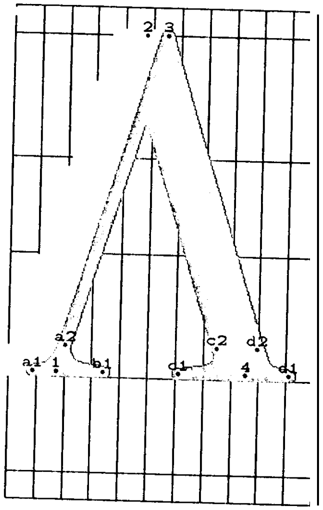




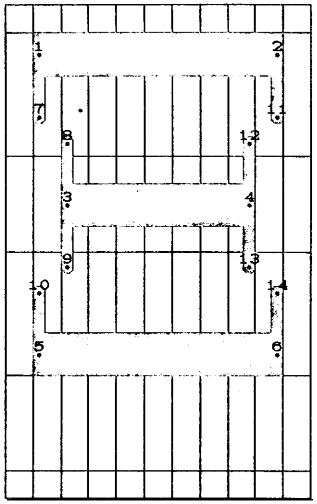
The file rom i tu mf

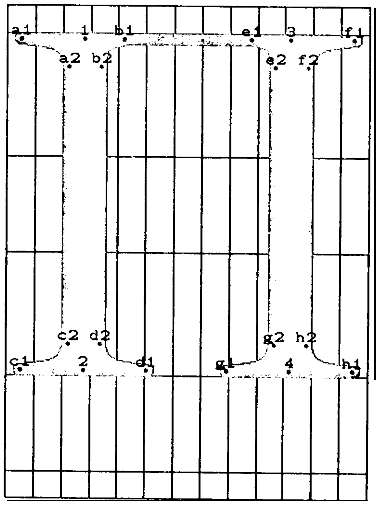
```
% These upper-case Rornan and Creek alphabets were prepared by D. E. Knuth in November,
% 1979, inspired by thc Monotypc alphabets used in Thc Art of Computer Programming.
% For text spacing, set mi = 0; for math spacing, set mi = 1.
% Character codes '000-'012 and '101-'132 arc generated.

new mc, lbowl, rbowl, rstcm, rv, hic;                    % quantities used to compute spacing
mc = mi/pu;                                              % converts to rciativc units when mi = 1
lbowl = .3ph·slant + .5pu;                               % used at left of upper-case bowl
rbowl = .7ph·slant − .5pu;                               % used at right of upper-cast howl
rstem = ph·slant + (ucs + 2sc − 1.5)pu;                  % used at right of tall stcm
rv = ph·slant + (.5ucs + 2sc − 1)pu;                     % used at right of tall diagonal
hic = 1 − .5mi;                  % used when half the italic corrcclion goes into rtcorr

"Upper case Greck Gamma";
call charbegin('000, 11, 2sc, sc − mc(armic − 2.5pu), ph, 0, mi[armic, 2.5pu]);
hpen;  lft_1 x_1 = round 2u;  x_2 = x_1;  top_1 y_1 = h;  bot_1 y_2 = 0;
w_1 draw 1.. 2;                                                          % stern
if ucs ≠ 0:
      call 'a serif(1, 4, 2, −ucs);
      call . b serif(1, 4, 2, .5ucs);                                    % upper stem serif
      call `c serif(2, 4, 1, −ucs);
      call `d serif(2, 4, 1, ucs);                                       % lower stem serif
fi;
new ss;  ss = 1.4aspect·ucs·u + eps;
if ss + w_6 > .25h:  new ss;  ss = .25h − w_6 + eps;
fi;
rt_0 x_3 = round(r − 1.5u);  x_4 = x_3 + .5u;  y_3 = y_1;  y_4 = y_3 − ss;
call `e arm(1, 3, 4).                                                    % upper arm and serif

"Upper case Greck Dclta";
call charbegin( '001, 15, 0, 0, ph, 0, 0);
hpen;  lft_0 x_2 = round u;  rt_4 x_4 = round(r − u);  bot_0 y_2 = 0;  bot_4 y_4 = 0;
top_4 y_5 = h + o;  y_6 = y_5;
x_6 − x_2 = x_4 − x_5;  rt_1 x_5 = rt_0 x_6;  x_1 = x_2;  x_3 = x_4;
vpen;  bot_8 y_1 = 0;  y_3 = y_1;
w_9 draw 1 .. 3;                                                         % bar line
rpen#;  w_9 draw 5.. 4;                                   % erase excess at lower right
hpen;  w_1 draw 5. . 4;                                                  % right diagonal
lpen#;  w_1 draw 6. . 2;  w_9 draw 6.. 2;                               % erase excess at left
hpcn; w_0 draw 6.. 2;                                                    % left diagonal
Y7 = y_4;  rt_0 x_7 = rt_4 x_4;
w_0 draw 2.. 7.. 7.. 6.                                     % sharpen lower right corner
```



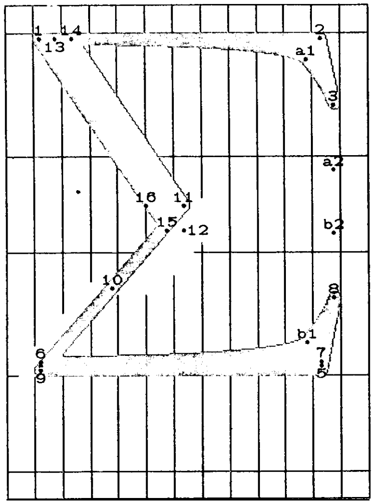


```
'Upper case Creek Theta";
call charbegin( '002, 14,mc·lbowl,—mc·rbowl,ph, pdd, mi[rbowl,0]);
hpen;
if fixwidth ≠ 0: new save;   save == sqrttwo;
     new sqrttwo;   sqrttwo = sqrt save;                          % super-superellipse
     lft_5 x_2 == round 1.5u;
else:   lft_5 x_2 = round u;
fi;
x_1 = r — x_1;                                                % axis of left-right symmetry
top_0 y_1 == h + 00;
bot_0 y_2 = -00;   y_3 = y_2;   x_3 = r — x_2;
call . a darc( 1, 2, w_5);                                          % left part of bowl
call ` b darc(1, 3, w_5);                                           % right part of bowl
lft_0 x_4 == round(rt_5 x_2 + u);   x_5 = r — x_4;   y_4 == y_5 == .5[y_1, y_2];
vpcn;
if ucs = 0:   w_9 draw 4..5;                                                 % bar tine
else: call bar(4,5);                                                         % bar line
     x_6 = x_7 = x_4;   x_8 == x_9 == x_5;   y_7 = y_9 == .6h;   y_6 = y_8 = .4h;
     w_0 draw 6..7;                                                        % left serif
     draw 8.. 9;                                                          % right serif
fi;
if fixwidth ≠ 0: new sqrttwo;   sqrttwo = save;
fi.

"Upper case Creek Lambda";
call charbegin( '003, 11,2sc,2sc, ph, 0,0);
hpen; lft_0 x_1 == round 1.5u; bot_0 y_1 == 0;
rt_5 x_4 == round(r —· 1.5u);   bot_5 y_4 == 0;
top_0 y_3 == top_5 y_2 == h + o;
x_3 — x_1 == x_4 —· x_2;   rt_5 x_2 == rt_0 x_3;
w_5 draw 2..4;                                                 % right diagonal stroke
lpen#;   wj draw 3.. 1;                                   % erase excess at upper left
hpen;   w_0 draw 3.. 1;                                         % left diagonal stroke
if ucs ≠ 0:
     call 'a serif(1,0,3,—.5ucs);
     call ` b serif(1, 0, 3, +ucs);                                       % left serifs
     call ` c serif (3, 5, 2, —·ucs);
     call ` d serif(4,5, 2 , + .5ucs);                                   % right serifs
fi.
```



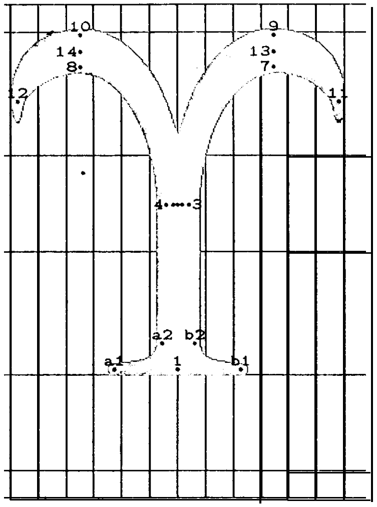


```
"Upper case Greek Xi";
call charbegin('004,11, 0, —.5mc(ph·slant — .5pu), ph, 0, hic(ph·slant — .5pu));
vpen;   top8y1 = h;   y1 = y2;   y3 = y1 = good8 .5h;   bot8y5 = 0;   y5 = y6;
lft8x1 = round u;   rt8x2 = round(r — u);   x5 = x1;   x6 = x2;
lft8x3 = round 2u;   rt8x4 = round(r — 2u);
if ucs = 0:   w9 draw 1 . . 2;                                      % upper bar
     draw 3.. 4;                                                    % middle bar
     draw 5.. 6;                                                    % lower bar
else: call bar(1, 2);                                               % upper bar
     call bar(3, 4);                                                % middle bar
     call bar(5, 6);                                                % tower bar
     new ss;   ss = ucs·aspect·u + eps;
     if ss > .2h:   new ss;   ss = .2h + eps;
     fi;
     x ;  = x10 = x1;   x8 = x9 = x3;   x11 = x1,   = x2;   x12 = x13 = x 4 ;
     bot6y7 = bot8y1 — ss;   top6y8 = top8y3 + ss;
     bot6y9 = bot8y3 — ss;   top6y10 = top8y5 + ss;
     y11 = y7; y12 = y8; y13 = y9;   y1 1 = y10;
     w6 draw 7.. 1; draw 11 . . 2;                                  % upper serifs
     draw 8. . 9; draw 12.. 13;                                     % middle serifs
     draw 10 . . 5; draw 14 . . 6;                                  % tower serifs
fi.

"Upper case Greek Pi";
call charbegin('005, 13, 2sc, 2sc — .5mc·rstem, ph, 0, hic·rstem);
hpen;   lft 1x1 = round 2u;   x2 = x1;
rt4x3 = round(r — 2u);   x4 = x3;
top 1y1 = h;   y1 = y3;
bot 1y2 = 0;   y2 = y4;
w4 draw 1 . . 2;                                                    % left stem
w4 draw 3.. 4;                                                      % right, stem
if ucs ≠ 0:   call 'a serif(1, 4, 2, —ucs);
          call `b serif(1, 4, 2, .5ucs);                           % upper left serif
     call ` c serif(2, 4, 1, — ucs);
          call `d serif(2, 4, 1, ucs);                             % tower left serif
     call ` e serif (3, 4, 4, —.5ucs);
          call ` f serif(3, 4, 4, ucs);                            % upper right serif
     call `g serif(4, 4, 3, —ucs);
          call ` h serif(4, 4, 3, ucs);                            % tower right serif
fi;
w0 draw 1 . . 3.                                                    % bar
```

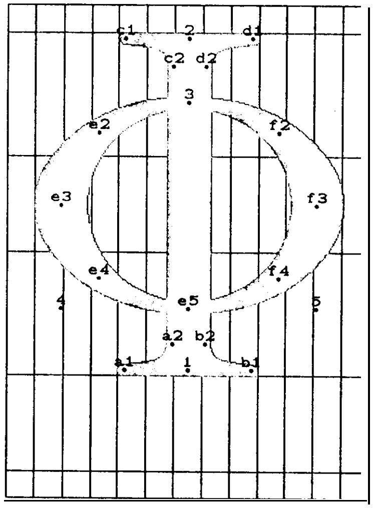


4

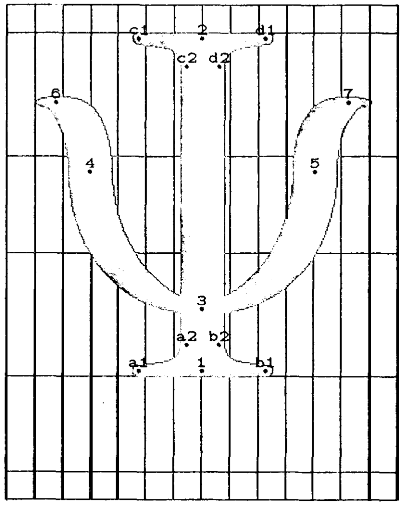
```
"Upper case Greek Sigma";
call charbegin( '006, 13, 0, sc — .5mc·armic, ph, 0, hic·armic);
new w99;   w99 = round .25[w6, w9];
hpen;   lft0x1 = round u;   rt0x2 = round(r — 1.5u);   x3 = x2 + .5u;
new ss;   ss = 1.4aspect·ucs·u + eps;
if ss + w6 > .25h:   new ss;   ss = .25h — w6 + eps;
fi;
top0y1 = h;   y2 = y1;   y3 = y2 — ss;
vpen;   bot99y4 = 0;   y5 = y4;   top6y6 = top99y4;   y7 = y6;   y8 = y7 + ss;
x4 = x6 = x1;   x j = x7 = x2;   x8 = x3;
call `a arm(1, 2, 3);                                              % upper arm and serif
call `b arm(6, 7, 8);                                              % lower arm and serif
w99 draw 4. . 5;
if ucs = 0:   draw 5.. 8;                                          % thicken lower arm
else:   if w0 ≠ w1:   draw |w99|5 . |w6|8;                         % thicken lower arm
     else: draw 5. .8;                                             % thicken lower arm
     fi;
fi;
hpen;   x9 = x1;   x10 = .5[x9, x11];   x16 = round 5u;   lft1x12 = lft0x15;
lft1x99 = x16;   rt1x99 = rt0x11;   lft1x13 = lft0x1;   rt1x13 = rt0x11;
bot0y9 = 0;   y10 = .5[y9, y10;   y11 = y16 = .5h;   y12 = y15;   y13 = y11 = y1;
new aa, bb;   lft0x1j = aa[lft0x1, x16];   y15 = aa[y1, y16];
lft0x15 = bb[x9, x11];   y15 = bb[y9, y11];
w4 draw 13. . 12;                                                  % upper diagonal
w0 draw 1. . 15;                                         % sharpen upper left corner
rpen#;   w4 draw 10. . 11;                                     % erase excess at right
lpen#;   w99 draw 9.. 10;                                       % erase excess at left
hpen;   w0 draw 9. . 11;                                               % lower diagonal
draw 14.. 11.                                              % sharpen middle corner
```

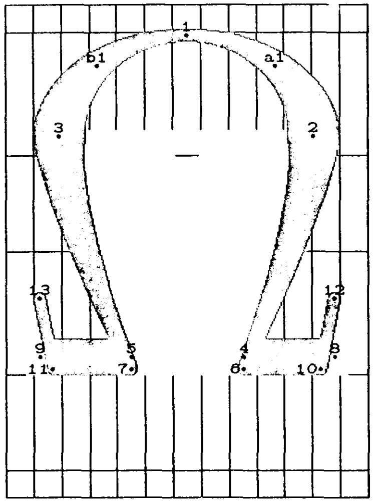


6
2
5

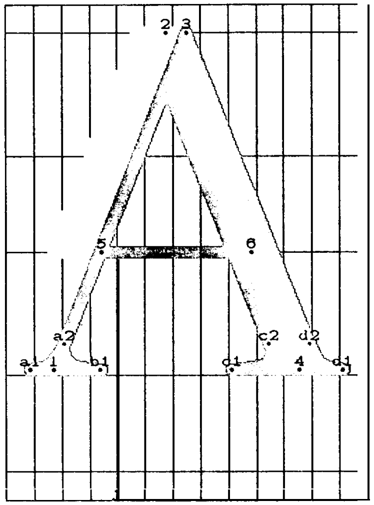


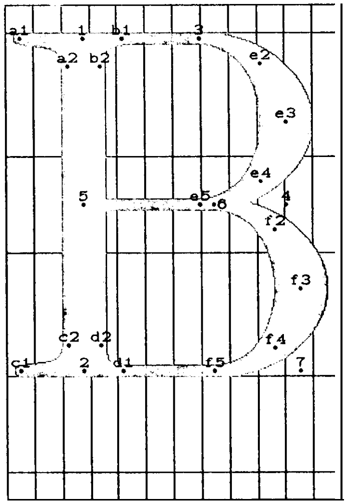
```
'Uppercase Greek Upsilon";
call charbegin( '007, 14, mc(.8ph·slant + .5pu), —mc(.8ph·slant — 3pu),
                          ph, 0, mi[.8ph·slant —.5pu,2.5pu]);
hpen;   x_1 = good_1 .5r;   bot_4 y_1 = 0;   x_2 = x_1;   y_2 = .5h;
w_1 draw I . . 2;                                                        % stem
if ucs ≠ 0:   call 'a serif(1, 4, 2, —ucs);
     call ˜ b serif(1, 4, 2, ucs);                                       % serif
fi;
new w_99;   w_99 = round .5w_4;
rt_99 x_3 = rt_4 x_2;   lft_99 x_4 = lft_4 x_2;
lft_99 x_5 = x_2 — .25u;   x 7 = x_9 = x_13 = 10.5u;
vpen;   rt_8 x_11 = round(r — u);   y_11 = .8h;
top_8 y_13 = top_6 y_9 = h + oo;   bot_8 y_13 = bot_0 y_7;
y_3 = y_1 = y_5 = y_6 = y_2;   y_7 = y_8;   y_9 = y_10;   y_11 = y_12;   y_13 = y_14;
x_3 + x_4 = x_5 + x_6 = x_7 + x_8 = x_9 + x_10 = x_11 + x_12 = x_13 + x_14;   % Icfl-right symmetry
w_9 draw 13{1,0} . 11{0, —1};                                            % right stroke
draw 14{—1, 0 } 12{0, —1};                                               % left stroke
hpen;   w_99 ddraw 5{0, 1} . 9{1, 0}, 3{0,1} . . 7{1,0};
ddraw 6{0, 1}. . 10{—1, 0}, 4{0,1} . . 8{---1, 0).                        %middle strokes

"Upper case Greek Phi";
call charbegin( '010, 13, mc(.5ph·slant + .5pu), —mc(.5ph·slant — .5pu),
                          ph, 0, mi[.5ph·slant -- .5pu,0]);
hpcn;   x_1 = good,, .5r;   bot_4 y_1 = 0;   x_2 = x_1;   top_4 y_2 = h;
w_1 draw 1 . . 2;                                                        % stem
if ucs ≠ 0:   call 'a serif(1, 4, 2, -ucs);
     call ˜ b serif(1, 4, 2, ucs);                                       % upper serif
     call ˜ c serif(2, 4, 1, —ucs);
     call ˜ d serif(2, 4, 1, ucs);                                       % lower serif
fi;
x_3 = x_1;   y_3 = .8h;   y_4 = y_5 = .2h;
lft_5 x_4 = round u;   x_3 — x_4 = x_5 — x_3;                             % left-right symmetry
call ˜ e darc(3, 4, w_5);   call . f darc(3, 5, wj).                      % bowl
```



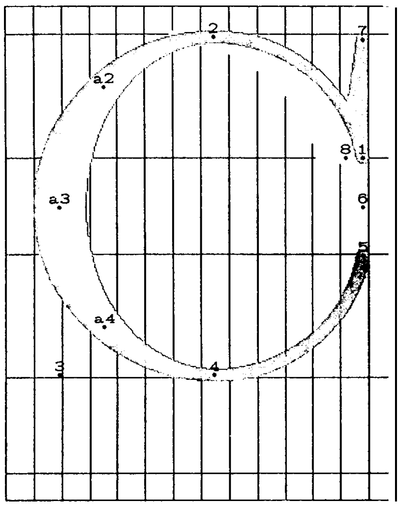


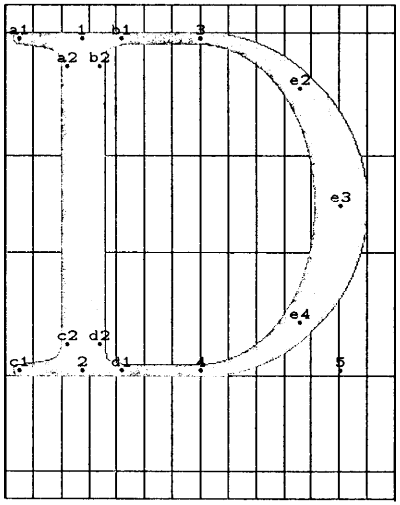
```
"Upper case Greek Psi";
call charbegin('011, 14, mc(.8ph·slant + .5pu), —.5mc(.8ph·slant —- .5pu),
                    ph, 0, hic(.8ph·slant — .5pu));
hpen;  x1 = good1.5r;  bot1y1 == 0;  x2 = x1;  top1y2 = h;
w4 draw 1..2;                                                          % stem
if ucs ≠ 0:  call 'a serif(1, 4, 2, —ucs);
    call ` b serif (1, 4, 2, ucs);                                     % upper serif
    call . c serif(2, 4, l, —ucs);
    call ` d serif (2, 4, 1, ucs);                                     % lower serif
fi;
x3 = x1;  y3 = .2h;  lft1x6 = u; x4 = good1 3u; y4 = .6h; y6 == .8h;
y4 == y5;  y6 == y7;  x3 - - x4 == x5 — x3;  x3 - - x6 == x7 — x3;    % left-right symmetry
w4 draw 6{ 1, O}. . 4{0, —1} .. 3{1, 0};                                %left stroke
draw 7{—1, 0} .. 5{0, —1} .. 3{—1, 0}.                                 % right stroke

"Upper case Greek Omega";
call charbegin('012, 13, mc(2/3 pe slant + .5pu), —.5mc(.75ph·slant — .5pu),
                    ph, 0, hic(.75ph·slant — .5pu));
hpen;  top0y1 == h + oo;  y2 == .7h;  rt5x2 == rt0x8 == round(r -- u);
vpen;  bot7y8 == bot6y6 == 0;  y1 == y8;  x1 == x6 == good0 8.5u;
x1 + x1 == x2 + x3 == x 1 + x j == x6 + x7 == x8 + x9 == r;           % Icft-right symmetry
y2 = y3;  y4 == y5;  y6 == y7;  y8 == y9;
w7 draw 4.. 8;                                                         % right bar
draw 5.. 9;                                                            % left bar
lpen#;  x6 -- x7 draw 2{0, — 1}  6{x6 — x2, 5/7(y6 — y2)};            %erase middle
rpen#;  x6 — x7 draw 3{0, —1} . 7{x7 — x3, 5/7(y7 — y3)};             % ditto
call 'a arc(1, 2, w5);  call ` b arc(1, 3 , w5);                       % bowl
hpen;  draw |w5|2{0, —1} : . |w0|6{x6 — x2, 5/7(y6 — y2)};            % right diagonal
draw |w5|3{0, --I}. . |w0|7{x7 — x3, 5/7(y7 — y3)};                   % left diagonal
if ucs ≠ 0 : x12 == x8; bot0y10 == 0 ; top0y12 == 2/3e;
    x10 + x11 = x12 + x13 == r;  y10 == y11;  y12 == y13;
    if w7 = w9:  x10 == x12;
    else:  x10 + .5u == x12;
        rpen#;  .5u + w0 draw 10.. 12;                                 % erase excess at right
        lpen#;  .5u + w0 draw 11 .. 13;                                % erase excess at left
    fi;
    hpen; w0 draw 10 .. 12;  draw 11 .. 13;                            % serifs
fi.
```



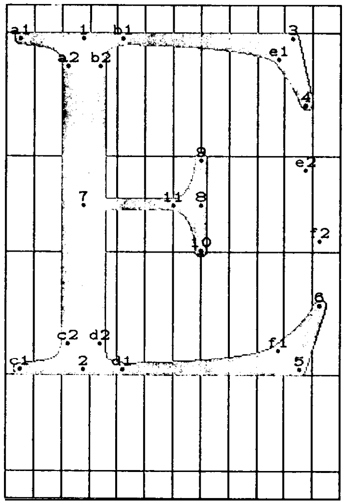


```
"The letter A";
call charbegin(` A, 13, 2sc,2sc, ph, 0,0);
hpen;
lft0 x1 = round 1.5u;   bot0 y1 = 0;
rt5 x4 = round(r — 1.5~);   bot5 y4 = 0;
top0 y3 = top5 y2 = h + o;
x3 — x1 = x4 — x2;   rt5 x2 = rt0 x3;
w5 draw 2.. 4;                                          % right diagonal stroke
y5 = y6 = e;
new aa,. bb;                          % auxiliary variables for intersection of lines
xj — 1 = aa[x1,x3];   y5 = aa[y1, y3];
x6 + 1 = bb[x1, x2];   y6 = bb[y4, y2];
w0 draw 5.. 6;                                                        % bar line
lpen#;   w5 draw 3.. 5;                             % erase excess at upper left
hpen;   w0 draw 3. . 1;                                   % left diagonal stroke
if ucs ≠ 0:
     call 'a serif(1, 0, 3, —.5ucs);
     call `b serif(1, 0, 3, +ucs);                                 % left serifs
     call `c serif(4, 5, 2, --ucs);
     call `d serif(4, 5, 2, +.5ucs);                              % right serifs
fi.

"The letter B";
call charbegin(`B, 12, 2sc,--.5mc(.75ph·slant — 5pu), ph, 0, hic(.75ph·slant -- .5pu));
hpen;   lft 1x1 = lft 1x2 = round 2u;   top 1y1 = h;
bot 1y2 = 0;
w4 draw 1.. 2;                                                            % stem
if ucs ≠ 0:
     call ` a serif (1, 4, 2, —ucs);
     call . b serif (1, 4, 2, .5ucs);                              % upper serif
     call ` c serif(2, 4, 1, —ucs);
     call `d serif(2, 4, 1, .5ucs);                                % lower serif
fi;
x3 = ½[2u, r];   y3 = y1;
rt5 x4 = round(r — u);   y4 = good0 ½h;
w0 draw 1..3;                                                 % upper bar line
call ` e darc(3, 4, w5);                                         % upper bowl
x5 = x1;   x6 = x3 + ½u;   y4 = y5 = y6;
rt5 x7 = round(r — ½u);   bot5 y7 = 0;
w0 draw 5 6;                                                 % middle bar line
call `f darc(6, 7, w5);                                          % lower bowl
x8 = x6;   y8 = y7; w0 draw 2. .8.                          % lower bar line
```

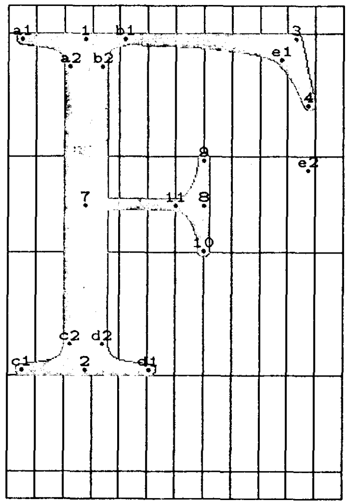




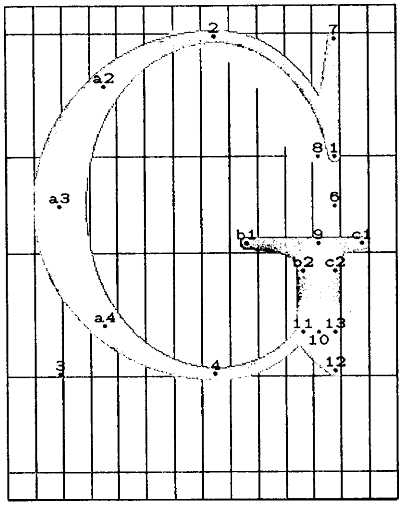
```
'The letter C";
if ucs ≠ 0:   call charbegin(`C, 14, mc·lbowl, —.5mc(ph·slant — .5pu),
                        ph, 0, hic(ph·slant — .5pu));
else:   call charbegin(`C, 11, mc·lbowl, —.5mc(ph·slant — .5pu),
                        ph, 0, hic(ph·slant — .5pu));
fi;
hpen;
rt_0x_1 = round(r — u);   x_5 = x_1;   lft_5x_3 = round u;   x_2 = x_4 = 7.5u;
top_0y_2 = h + oo;   bot_0y_3 = —oo;   y_3 = y_4;   y_6 = .5[y_2, y_4];
if ucs = 0: x_6 = 13u; new aa; x_1 = aa[x_2, x_6];
    y_1 = (sqrt(1 — aa·aa))[y_6, y_2];
else: if m < .6h: y_1 = good_6 2/3h; else:   y_1 = good_6m;
    fi;
    x_6 = x_1;   x_7 = x_1;   top_0y_7 = h;   lft_0x_8 = lft_1x_1;   y_8 = y_1;
    w_0 ddraw 1..7, 8..7;                                   % upper serif
    lpen#;   w_4 draw (6. )1.2{—1, O};                      % erase spurious part
fi;
hpen;   y_5 = h — y_1;
w_0 draw (6. .)1..2{—1, O};                                 % upper right stroke
call ` a darc(2, 3, w_5);                                   % main stroke
w_0 draw 4{1, 0}..5(..6).                                   % lower right stroke

"The letter D";
call charbegin(`D, 14, 2sc, —mc·rbowl, ph, 0, mi[rbowl, 0]);
hpen;   lft_4x_1 = round 2u;   x_2 = x_1;   top_1y_1 = h;   bot_1y_2 = 0;
w_4 draw I..2;                                              % stem
if ucs ≠ 0:
    call ` a serif (1, 4, 2, — ucs);
    call ` b serif(1, 4, 2, .5ucs);                         % upper serif
    call ` c serif(2, 4, 1, —ucs);
    call ` d serif(2, 4, 1, .5ucs);                         % lower serif
fi;
x_3 = x_4 = 7u;   rt_5x_5 = round(r. .u);
y_3 = y_1;   y_4 = y_5 = y_2;
w_0 draw 1.3;                                               % upper bar line
call ` e darc(3, 5, w_5);                                   % bowl
w_0 draw 4..2.                                              % lower bar line
```



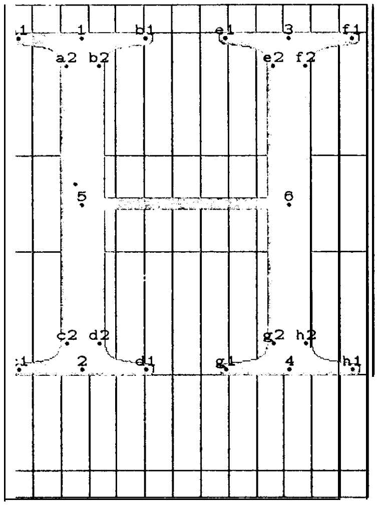
```
“The letter E”;
call $charbegin$(` E, 12, 2$sc$, sc — .5$mc \cdot armic$, ph, 0, $hic \cdot armic$);
hpen; $\mathrm{lft}_4 x_1$ = round 2$u$; $x_2 = x_1$; $\mathrm{top}_4 y_1$ = h ; $\mathrm{bot}_4 y_2$ = 0 ;
$w_4$ draw 1.. 2;                                                      % stem
if $ucs \neq 0$:
     call ‘a $serif$ (1, 4, 2, —$ucs$);
     call` b $serif$(1, 4, 2, .5$ucs$);                           %upper slem serif
     call . c serif (2, 4, 1, —$ucs$);
     call `d serif(2, 4, 1, .5$ucs$);                             % lower stem serif
fi;
new $ss$;   $ss$ = 1.4aspect.ucs.u + eps;
if $ss + w_6 > .25h$:   new $ss$;   $ss$ = .25$h$ - - $w_6$ + cps;
fi;
$\mathrm{rt}_0 x_3$ = round($r$ — 1.521);   $x_4 = x_3 + .5u$;   $y_3 == y_1$;   $y_4 == y_3 - ss$;
$\mathrm{rt}_0 x_5$ = round($r$ — 1.25~);   $x_6 = x_4 + .5u$;   $y_5 == y_2$;   $y_6 == y_5 + ss$;
call `e $arm$(1, 3, 4);                                          % upper arm and serif
call `f arm(2, 5, 6);                                            %lower arm and serif
$x_7 = x_1$;   $y_7 = y_8 = .5[y_1, y_2]$;   $x_8 = \mathrm{good}_0\, 7u$;
$w_0$ draw 7. . 8;                                                      %middle arm
if $ucs \neq 0$:   $x_9 = x_{10} = x_8$;   $y_9 = y_8 + .7ss$;   $y_{10} = y_8 - .7ss$;
     if $w_0 = w_4$:   $w_0$ draw 9.. 10;                            % middle arm serif
     else:   $x_{11} = x_8 - u$;   $y_{11} = y_8$;
          minvr 0;   minvs 0;
          $w_0$ ddraw 11{1, 0} . . 10{0, —1}, 8.. 10;
          ddraw 11{ 1, 0} . . 9{0, 1}, 8 . . 9;                       %middle arm serif
          minvr .5; minvs .5;
     fi;
fi.
```

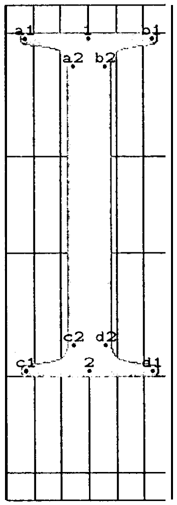


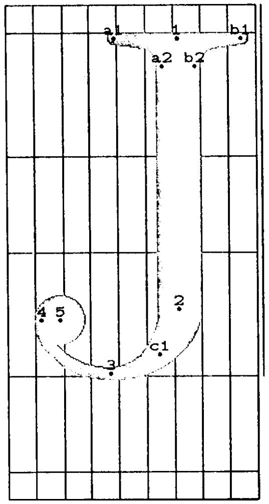
```
"The letter F";
call charbegin(`F,12, 2sc, sc — mc(armic — 2.5pu), ph, 0, mi[armic, 2.5pu]);
hpen;   lft4x1 = round 2u;   x2 = x1;   top4y1 = h;   bot 1y2 = 0;
w1 draw 1 .. 2;                                                             % stem
if ucs ≠ 0:
     call `a serif (1, 4, 2, — ucs);
     call `b serif(1, 4, 2, .5ucs);                                 % upper stem serif
     call `c serif(2, 4, 1, —ucs);
     call `d serif (2, 4, 1, ucs);                                  % lower stem serif
fi;
new ss;   ss = 1.4aspect·ucs·u + eps;
if ss + w6 > .25h:   new ss;   ss = .25h — w6 + eps;
fi;
rt0x3 = round(r -- 1.5u);   x4 = x3 + .5u;   y3 = y1;   y4 = y3 — ss;
call `earm(1, 3, 4);                                             % upper arm and serif
x7 = x1;   y7 = y8 = .5[y1, y2];   x8 = good0 7u;
w0 draw 7.. 8;                                                         % lower arm
if ucs ≠ 0:   x9 = x10 = x8;   y9 = y8 + .7ss;   y10 = y8 — 7ss;
     if w0 = w4:   w0 draw 9.. 10;                                 % lower arm serif
     else:   x11 = x8 — u;   y11 = y8;
          minvr 0;   minvs 0;
          w0 ddraw 1 1{1, 0} 10{0, —1}, 8.. 10;
          ddraw 11{1, 0} .. 9{0, 1}, 8 .. 9;                      % lower arm serif
          minvr .5;   minvs .5;
     fi;
fi.
```



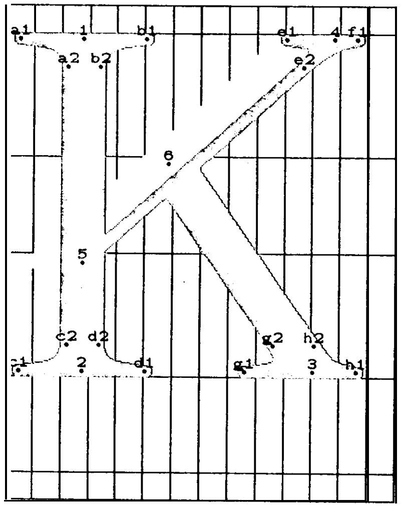
```
' The letter G";
if ucs == 0:   if m < .6h:   call charbegin(`G, 14, mc·lbowl,
                                sc — mc( 2/3 ph·slant + (sc -- 1.5)pu),
                                ph, 0, (1 — mi)( 2/3 ph·slant + (sc — 1.5)pu));
     else:   call charbegin(`G, 14, mc·lbowl,
                                1 — mc(px·slant + (sc — I .5)pu),
                                ph, 0, (1 — mi)(px·slant + (sc — 1.5)pu));
     fi;
else:   call charbegin(`G, 14, mc·lbowl,
                             sc — mc(ph·slant + (sc — 1.5)pu),
                             ph, 0, (1 -- mi)(ph·slant + (sc — 1.5)pu));
fi;
hpen;
rt0x1 = rt 1x9 == round(r — 2u);   lft5x3 = round u;   x2 == x 1 == 7.5u;
top0y2 = h + 00;   bot0y3 = --00,   y3 = y4;   y6 == .5[y2, y4];
if m < .6h:   y1 == good,, 2/3 h;   else:   y1 == good6 m;
fi;
x6 == x1;   y9 == good6 .1[e, m];   x9 = x10;
if ucs ≠ 0:   x7 = x1;   top0y7 == h;   lft0x8 == lft 1x1;   y8 == y1;
     w0 ddraw 1 . . 7, 8 . . 7;                                          %  upper serif
     lpen#;   w1 draw (6. . )1 . . 2{---1, 0};                   % erase spurious part
fi;
hpen;   w0 draw (6 . )1 . 2{—1, 0};                             % upper right stroke
call `a darc(2, 3, w5);                                                  % main stroke
if w0 = w4:   w0 draw 4{1, 0} . . 9{0, 1};                     % lower right stroke
     bot0y10 == 0; draw 9.. 10;                                      % stem and spur
else:   y11 = y10 == y13 == 1/3 y9;
     lft0x11 = lft.1x9;
     rt0x12 = rt0x13 == rt 1x9;   bot0y12 = 0;
     w0 draw 4{1, 0} 11 . . 9{0, 1};                                % lower right stroke
     w4 draw 9.. 10;                                                           % stem
     w0 ddraw 13. 12, 11{0, --1} . . 12{2(x12 --- x11), y12 — y11};           % spur
fi;
if ucs < 2:   call . b scrif(9, 4, 10, -2);
else:   call ` b serif(9, 4, JO, —ucs);                              % lower serif
fi;
if ucs > 0:   call `c serif(9, 4, 10, -|-1);
fi.
```

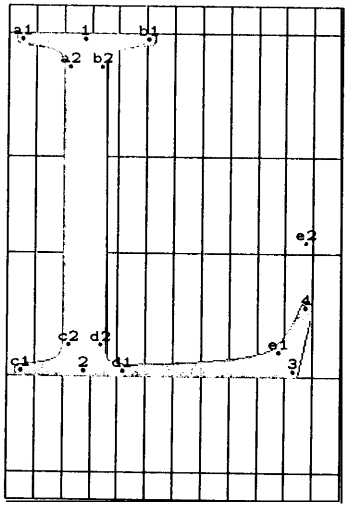




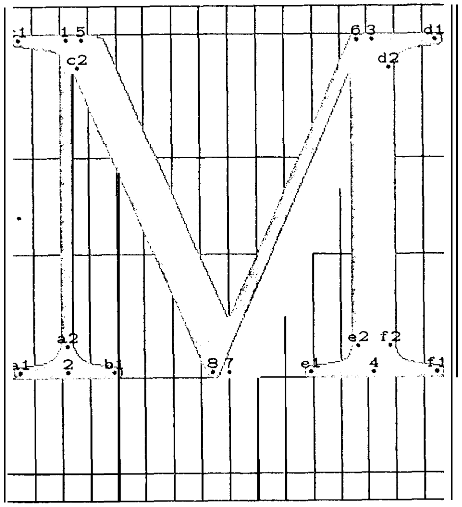


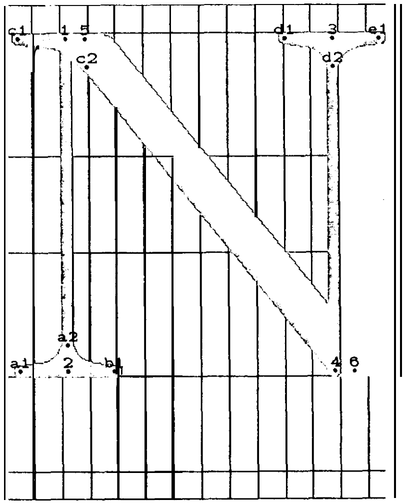
```
“The letter II”;
call charbegin(`H,13, 2sc, 2sc — .5mc·rstem, ph, 0, hic·rstem);
hpen;   lft 1x1 = round 2u;   x2 = x1;
rt 4x3 = round(r — 2u);   x4 = x3;
top 1y1 = h;   y1 = y3;
bot 1y2 = 0;   y2 = y4;
w4 draw 1 . . 2;                                              % left stem
w4 draw 3 . . 4;                                              % right stem
if ucs ≠ 0:   call `a serif(1, 4, 2, —ucs);
            call ` b serif(1, 4, 2, ucs);                     % upper left serif
      call ` c serif (2, 4, 1, —ucs);
            c a l l `d serif(2, 4, 1,ucs);                    %lower left serif
      call `e serif(3, 4, 4, — ucs);
            call ` f serif(3, 4, 4, ucs);                     % upper right serif
      call `g srrif(4, 4, 3, —ucs);
            call ` h serif(4, 4, 3,ucs);                      % lower right serif
fi;
x5 = x1;   x6 = x3;   y5 = y6 = .5h;
w0 draw 5 . . 6.                                              % bar

“The letter I”;
call charbegin(` I, 6, 0, —.5mc(ph·slant — .5pu), ph, 0, hic(ph·slant — .5pu));
hpen;   x1 = x2 = good, .5r;   top 1y1 = h ;   bot 1y2 = 0;
w4 draw 1.. 2;                                                % stem
new ss;
if ucs ≠ 0: ss = ucs; else:  ss = 2 — .5w4/u;
fi;
call ‘a serif(1, 4, 2, —ss);   call ` b serif(1, 4, 2, ss);   % upper serif
call `c serif(2, 4, 1, —ss);   call `d serif(2, 4, 1, ss).    %lower serif

“The letter J”;
call charbegin(`J, 9, 0, 2sc -- .5mc·rstem, ph, 0, hic·rstem);
hpen; rt 1x1 = round(r — 2u);   top 1y1  = h  ; y2 = .2h; x2 = x1;
w4 draw 1.. 2;                                                % stem
if ucs ≠ 0:   call . a serif( I, 4, 2,—ucs);
     call` b serif (1, 4, 2, ucs);                            % serif
fi;
lft0x4 = lft 3x5 = round u;   y4 = y5 = 1/6h;
x3 = .5[x4, x2];   bot0y3 = —oo;
call ` c arc(3, 2, w1);                                       % tail
w0 d r a w 3{—1,0} . . 4{0, 1};   cpen;   w3 draw 5.          % bulb
```



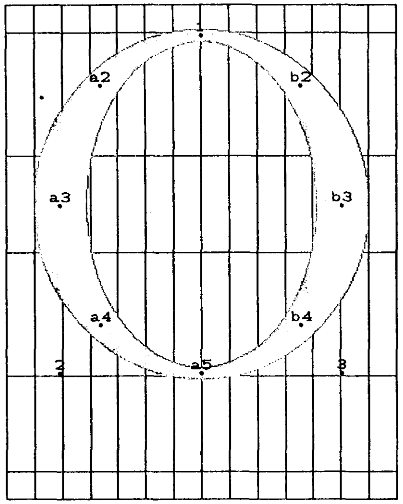


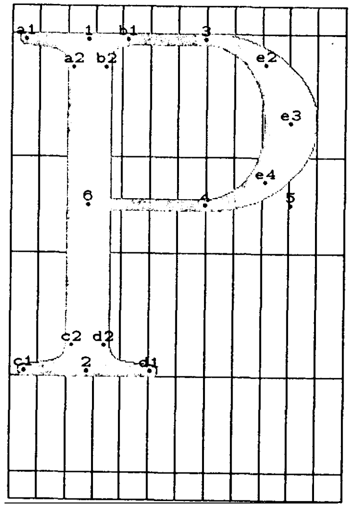
```
"The letter K";
call charbegin(`K, 14, 2sc, 2sc — .5mc(ph·slant + (.5ucs + 2sc — 1.5)pu),
                    ph, 0, hic(ph·slant + (.5ucs + 2sc -- 1.5)pu));
hpen;   lft_4 x_1 = round 2u;   x_1 = x_2 = x_5;
rt_5 x_3 = rt_0 x_4 = round(r — 2u);
top_4 y_1 == top_0 y_4 = h ;   bot_4 y_2 == bot_5 y_3 == 0 ;   y_5 = 1/3 h;
new aa, bb;                                          % auxiliaries for intersection of lines
rt_5 x_6 == aa[rt_5 x_1, rt_5 x_3];   y_6 == aa[y_1, y_3];
rt_5 x_6 = bb[x_5, x_4];   y_6 = bb[y_5, y_4];
w_5 draw 6.. 3;                                                         %lower diagonal
lpen#;   w_5 draw 4.. 5;                                                 % erase excess
hpen;   w_4 draw 1 . . 2;                                                       % stem
w_0 draw 4.. 5;                                                        % upper diagonal
if ucs ≠ 0:   call `a serif(1, 4, 2, —ucs);
     call` b serif(1, 4, 2, ucs);                                   % upper stem serif
     call ` c serif (2, 4, 1, —ucs);
     call `d serif(2, 4, 1, ucs);                                   % lower stern serif
     call `e serif(4, 0, 5, —ucs);
     call` f serif(4, 0, 5, .5ucs);                             % upper diagonal serif
     call `g serif(3, 5, 1, —ucs);
     call `h serif(3, 5, 1, .5ucs);                             % lower diagonal serif
fi.

"The letter L";
call charbegin(`L, 12, 2sc, sc, ph, 0, 0);
hpen;   lft_1 x_1 == round 2u;   x_2 == x_1;   top_4 y_1 == h ;   bot_4 y_2 = 0;
w_4 draw 1. . 2;                                                                % stem
if ucs ≠ 0:   call 'a serif(1, 4, 2, -- ucs);
     call` b serif(1, 4, 2, ucs);                                   % upper stein serif
     call ` c scrif(2, 4, 1, —ucs);
     call `d serif(2, 4, 1, .5ucs);                                  % lower stem serif
fi;
rt_0 x_3 = round(r -- 1.5u);   x_4 == x_3 + .5u;   y_3 = y_2;   y_4 == y_3 + 1.4aspect·ucs·u + eps;
call `e arm(2, 3, 4).                                              % arm and arm serif
```



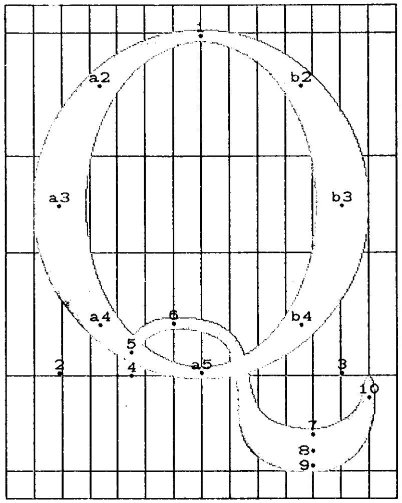


```
'The letter M";
call charbegin(`M,16,2sc,2sc—.5mc·rstem,ph,0,hic·rstem);
hpen;lft0x1== round 221;   x1 = x2;   top0y1 = h;   bot0y2 = 0;
w0 draw 1..2;                                                    % left stem
rt1x3 = round(r — 2u);   x3 = x4;   top1y3 = h ;   bot1y4 = 0 ;
lft1x5 = lft0x1;   lft0x6 = lft1x3;   y1 = y5; y6 = y3;
x7 — x5 = x6 — x8;   lft1x7 = lft0x8;   bot1y7 = 0;   y8 = y7;
w1 draw 5.. 7;                                                   % left diagonal
rpen#;   w1 draw 8..6;                             % erase excess at lower right
hpen;   w0 draw 8. 6;                                        % right diagonal
w1 draw 3..4;                                                    % right stem
if ucs ≠ 0:   call ' a serif(2,0,1,—ucs);
      call . b scrif(2, 0, 1,ucs);                          %lower left serif
      call . c scrif (5, 4, 7,— ucs);                       % upper left serif
      call `d serif(3, 4, 4, ucs);                          % upper right serif
      call ` e scrif(4, 4, 3,—ucs);
      call ` f serif(4,4, 3, ucs);                          %lower right serif
fi.

"The letter N";
call charbegin(`N, 14, 2sc, 2sc — .5mc·rstem, ph, 0, hicrstem);
hpcn;   lft0x1 == round 2u;   x1 == x2;   top0y1 = h;   bot0y2 = 0;
w0 draw 1.2;                                                     % left stem
rt0x3 == round(r — 2u);   x3 == x4;   top0y3 == h;   bot0y4 = 0 ;
lft 5x5 == lft0x1;   lft5x6 = lft0x4; y1 = y5; y6 = y4;
w5 draw 5. 6;                                                    % diagonal
rpen#;   wj draw 4.. 3;                            % erase excess at lower right
hpen;   w0 draw 4. 3;                                        % right diagonal
if ucs ≠ 0:   call 'a serif(2,0,1,—ucs);
      call` b serif (2,0,1, ucs);                          % lower left serif
      call ` c serif( 5, 5, 6,— ucs);                      % upper left serif
      call ` d serif(3, 0,4, -- ucs);
      call `e serif (3, 0, 4,ucs);                         % upper right serif
fi.
```

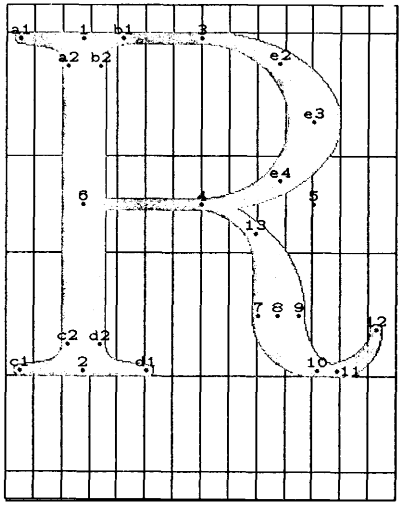




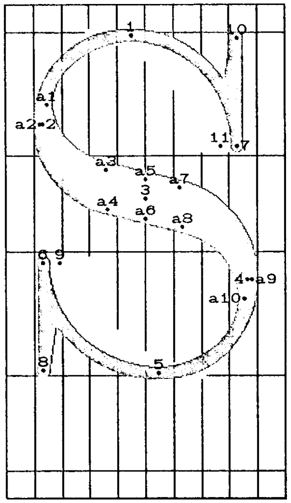
```
"The letter O";
call charbegin(`O, 14, mc·lbowl,--mc·rbowl,ph, 0, mi[rbowl,0]);
hpen;
if fixwidth ≠ 0: new save;   save = sqrttwo;
      new sqrttwo;  sqrttwo = sqrt save;                        % super-superellipse
      lft5x2 = round 1.5u;
else:  lft5x2 = round u;
fi;
x1 = r — x1;                                                % axis of left-right symmetry
top0y1 = h  + oo;
bot0y2 = - 0 0 ;   y3 = y2;   x3 = r — x2;
call . a darc(1, 2, w5);                                            % left part of bowl
call ` b darc(1, 3, w5);                                            % right part of bowl
if fixwidth ≠ 0: new sqrttwo;   sqrttwo = save;
fi.

"The letter P";
call charbegin(`P, 12, 2sc,--mc(.75ph·slant—3pu), ph, 0, mi[.75ph·slant -- .5pu,2.5pu]);
hpen; lft4x1 = round 2u;   x2 = x1;   top4y1 = h ;   bot4y2 = 0;
w4 draw 1. . 2;                                                              % stem
if ucs ≠ 0:
      call 'a serif(1, 4, 2,—ucs);
      call ` b serif(1, 4, 2, .5ucs);                                        % upper serif
      call ` c serif (2, 4, 1, —ucs);
      call `d serif (2, 4, 1, ucs);                                          % lower serif
fi;
x3 = x4 =  7 u ;   rt5x5 = round(r — u);   x6 = x1;
y3 = y1;   y6 = y4 = y5 = good6 .5h;
w0 draw 1 . . 3;                                                        % upper bar line
call `e darc(3, 5, w5);                                                           % bowl
w0 draw 4.. 6.                                                          % lower bar line
```



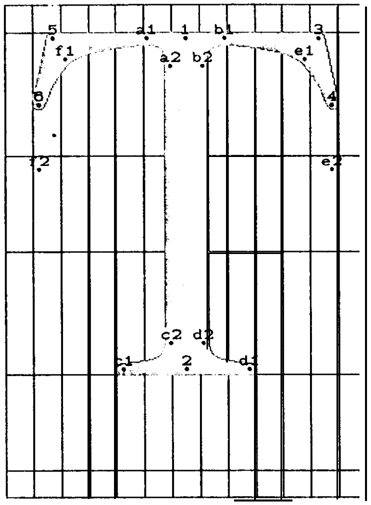
```
"The letter Q";
call charbegin(`Q, 14, mc·lbowl,—mc·rbowl, ph, pdd,mi[rbowl,0]);
hpen;
if fixwidth ≠ 0:   new save;   save = sqrttwo;
     new sqrttwo;   sqrttwo = sqrt save;                  % super-superellipse
     lft5x2 = round 1.5u;
else:   lft5x2 = round u;
fi;
x1 = r — x1;                                  % axis of left-right symmetry
top0y1 = h + oo;
bot0y2 = -00;   Y3 = y2;   x3 = r — x2;
call `a darc(1, 2, w5);                              % left part of bowl
call `b darc(1,3,w5);                               % right part of bowl
x1 = x5 = 4.5u;   y1 = 0;
new aa;   x5 = aa[x1, x2];   y5 = (sqrt(1 — aa·aa))[.5y1 + .5y2, y2];
x6 = 6u;   y6 = .1h + 1.5w6;
w0 draw (4.. )5 ..6{1, O};                            % left part of tail
vpen;   x7 = x8 = x9 = 11u;   bot8y8 = bot6y9 = —dd;   top6y7 = top8y8;
x10 = round 13u;   top8y10 = 0;
w6 ddraw 6{1, 0} 7{1, 0}, 6{1, 0} . 9{1,0};        % middle part of tail
w9 draw 8{1,0}10{0,1};                                % right part of tail
if fixwidth ≠ 0:   new sqrttwo;   sqrttwo = save;
fi.
```

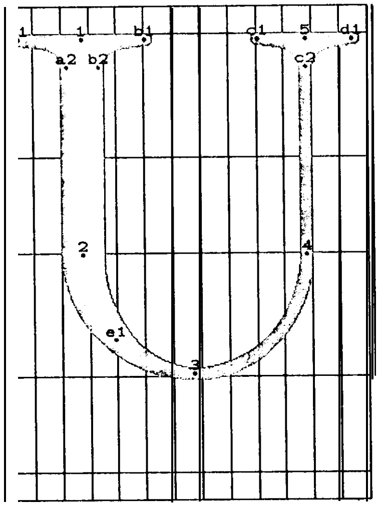


```
'The letter R";
hpen;
if ucs = 0:   call $charbegin$(`R, 12.5, 2$sc$, —$mc$(.75$ph$·$slant$ — .75$pu$),
                       ph, 0, $mi$[.75$ph$·$slant$ — .75$pu$, 0]);
      $rt_5 x_5$ = round($r$ — .75$u$);
else:   call $charbegin$(`R, 14, 2$sc$, 0, ph, 0, .75$ph$·$slant$ — 1.5$pu$);
      $rt_5 x_5$ = round($r$ — 2$u$);
fi;
$lft_4 x_1$ = round 2$u$;   $x_2$ = $x_1$;   $top_4 y_1$ = h;   $bot_4 y_2$ = 0;
$w_4$ draw 1. . 2;                                                    % stem
if $ucs$ ≠ 0:
      call 'a $serif$(1,4,2,—$ucs$);
      call`b $serif$(1, 4, 2, .5$ucs$);                               %upper serif
      call ` c serif(2, 4, 1, —ucs);
      call`d $serif$(2, 4, 1 , $ucs$);                                 % lower serif
fi;
$x_3$ = $x_4$ = 7 $u$ ;   $x_6$ = $x_1$;
$y_3$ = $y_1$;   $y_6$ = $y_4$ = $y_5$ = = good$_6$ .5$h$;
$w_0$ draw 1 . . 3;                                                  % upper bar line
call ` e $darc$(3, 5, $w_5$);                                                 % bowl
$w_0$ draw 4.. 6;                                                    % lower bar line
if ucs = 0:   rt $_1 x_7$ = round($r$ — $u$);   $bot_4 y_7$ = 0;
      $w_4$ draw 4 . . 7;                                              % diagonal stroke
else: $x_8$ = good$_5$($x_5$ — 1.25~); $lft_5 x_8$ = $lft_0 x_7$;   $rt_5 x_8$ = $rt_0 x_9$;
$y_7$ = $y_8$ = $y_9$ = $\frac{1}{3}$[$y_2$, $y_4$];
$y_{13}$ = $\frac{5}{6}$[$y_2$, $y_4$];   $x_{13}$ = 1/$sqrttwo$[$x_4$, $x_8$];
draw |$w_0$#|4{1, 0} . |.25[$w_0$, $w_5$]|13{$x_8$ — x 1, $\frac{2}{3}$($y_8$ — $y_4$)} . .
      |$w_5$#|8{0, —1};                                                % upper tail
$y_{10}$ = $y_{11}$ = $y_2$;   $y_{12}$ = .25[$y_2$, $y_4$];
$x_{10}$ = .5[$x_7$, $x_{12}$];   $x_{11}$ = .5[$x_9$, $x_{12}$];   $rt_0 x_{12}$ = round($r$ — .5$u$);
$w_0$ ddraw 7{0, —1} . , 10{1, 0} . . 12{0, 1},
      9{0, --1} . . 11{1, 0} . . 12{0, 1};                           % lower tail and hook
fi.
```

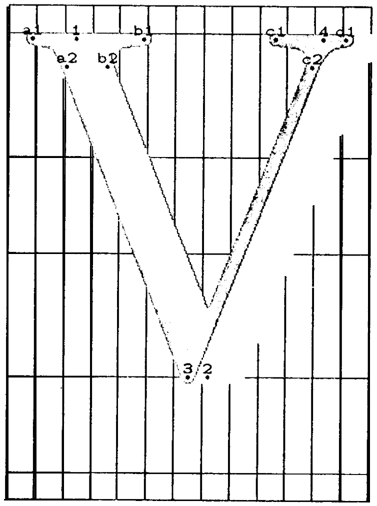


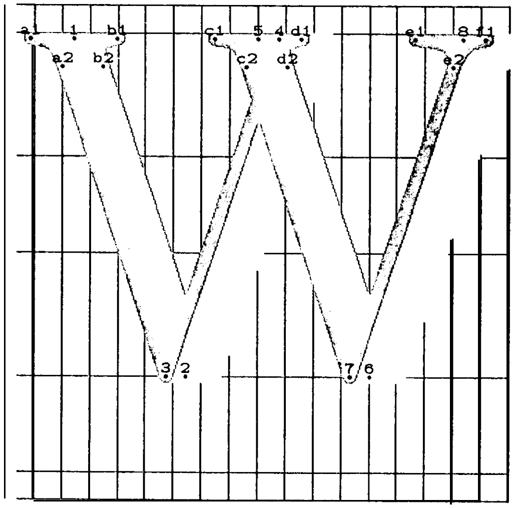
```
“The letter S”;
call $charbegin(`S, 10, 0, -.5mc(ph \cdot slant - .5pu)$, ph, 0, $hic(ph \cdot slant - .5pu))$;
new $w_{99}$;   $w_{99} =$ round $.2[w_0, w_4]$;
hpen;   $\mathrm{top}_0 y_1 = h + oo$;   $\mathrm{bot}_0 y_5 = -oo$;
$x_3 = .5r$;   $y_3 = .52h$;   $\mathrm{lft}_{99} x_2 =$ round $u$;   $\mathrm{rt}_{99} x_4 = \mathrm{round}(r - u)$;
if ucs = 0:   $x_1 = x_5 = x_3$;   $x_6 = x_2$;   x7 = $x_4$;
    $y_6 = \frac{1}{3}[\mathrm{top}_0 0, y_3]$;  $y_7 = \frac{1}{4}[\mathrm{bot}_0 h, y_3]$;
    $w_0$ draw $6\{x_5 - x_6, 3(y_5 - y_6)\} . 5\{1, O\}$;                 % lower left stroke
    draw $7\{x_1 - x_7, 2(y_1 - y_7)\} . 1\{-1, 0\}$;                     % upper right stroke
else:   if $w_0 = w_4$:   $x_1 = x_5 = x_3$;   $x_6 = x_2$;   x7 = x4;
        $y_6 = \frac{1}{3}[\mathrm{top}_0 0, y_3]$;   $y_7 = \frac{1}{4}[\mathrm{bot}_0 h, y_3]$;
        $w_0$ draw $6\{x_5 -- x_6, 3(y_5 - y_6)\} . 5\{1, 0\}$;            % lower left stroke
        draw $7\{x_1 - x_7, 2(y_1 -- y_7)\} . 1\{-1, 0\}$;                % upper right stroke
        $x_8 = x_9 = x_6$;   $y_9 = y_6 + ucs \cdot aspect \cdot u + eps$;   $\mathrm{bot}_0 y_8 = 0$;
        $x_{10} = x_{11} = x_7$;   $y_{11} = y_7 - ucs \cdot aspect \cdot u - eps$;   $\mathrm{top}_0 y_{10} = h$;
        $w_0$ draw 8. 9;                                                  % lower serif
        draw 10 . . 11;                                                   % upper serif
    else:   $x_1 = x_3 -- .5u$;   $x_5 = x_3 + .5u$;   $x_6 = x_2$;   $\mathrm{rt}_0 x_7 = \mathrm{round}(r - I .5u)$;
        $y_6 = \mathrm{good}_6 \frac{1}{3}h - 1$;   $y_7 =$ good, $\frac{2}{3}h + 1$;
        $\mathrm{bot}_0 y_8 = 0$;   $y_9 = y_6$;   $x_8 = x_6$;   $\mathrm{rt}_1 x_6 = \mathrm{rt}_0 x_9$;
        $\mathrm{top}_0 y_{10} = h$ ; $y_{11} = y_7$; $x_{10} = x_7$; $\mathrm{lft}_1 x_7 = \mathrm{lft}_0 x_{11}$;
        $w_0$ ddraw 6. . 8, 9 . . 8;                                       % lower serif
        ddraw 7.. 10, 11 . . 10;                                          % upper serif
        rpen#;   $w_4$ draw $6\{0, -1\} . . 5\{1, O\}$;                    % erase excess
        lpen#;   $w_1$ draw $7\{0, 1\} . 1\{-1, O\}$;                      % ditto
        hpen;   $w_0$ draw $6\{0, -1\} . . 5\{1, O\}$;                     % lower left stroke
        draw $7\{0, 1\} . . 1\{-1, O\}$;                                  % upper right stroke
    fi;
fi;
call ‘a $sdraw(1, 2, 3, 4, 5, w_{99}, \mathrm{round}(pixels \cdot pwv \cdot aspect +$ blacker$)$,
                $-h/(50u))$.                                              % middle stroke
```



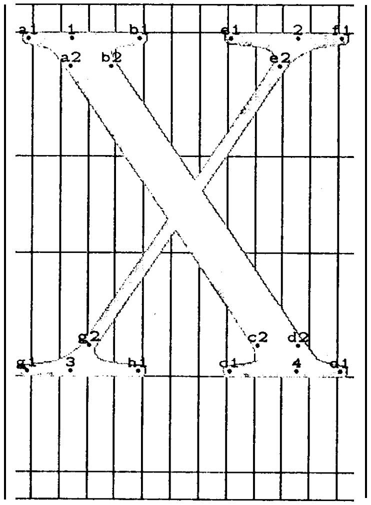


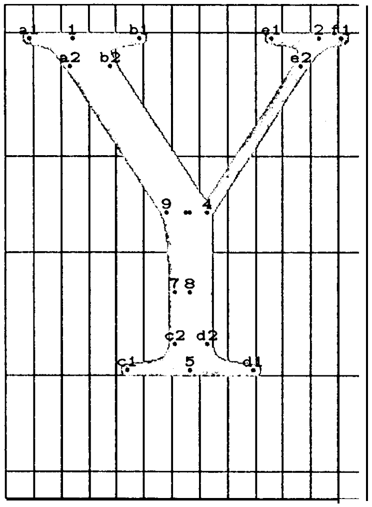
```
"The letter T";
call charbegin(`T, 13, sc + .75mc·ph·slant, sc — mc(armic — 2.5pu),
                  ph, 0, mi[armic, 2.5pu]);
hpen;   x1 = good4 6.5u;   x2 = x1;
top4 y1 = h;   bot4 y2 = 0;
w4 draw 1 . . 2;                                                    % stem
if ucs ≠ 0:   call 'a serif(1, 4, 2, —.5ucs);
      call `b serif(1, 4, 2, .5ucs);                       % upper bracketing
      call `c serif(2, 4, 1, —ucs);
      call `d serif(2, 4, 1, ucs);                               % lower serif
fi;
new ss;   ss = 1.4aspect·ucs·u + eps;
if ss + w6 > .25h:   new ss;   ss = .25h — w6 + cps;
fi;
rt0 x3 = round(r — 1.5u);   x4 = x3 + .5u;   y3 = y1;   y4 = y3 — ss;
lft0 x5 = round 1.5u;   x6 = x5 — .5u;   y5 = y1;   y6 = y5 - - ss;
call `e arm( 1, 3, 4);                             %upper right arm and serif
call` f arm(1, 5, 6).                               % upper left arm and serif

"The letter U";
call charbegin(`U, 13, 2sc(1 — mi) + mc(.3ph·slant + pu), 2sc --- .5mc·rstem,
                  ph, 0, hic·rstem);
hpen;
if fixwidth ≠ 0:   lft 1 x1 = round(1.5u);   rt0 x5 = round(r -- 1.521);
else:   lft 1 x1 = round 2u;   rt0 x5 = round(r — 2u);
fi;
x2 = x1;   x4 = x5;   x3 = .5[x2, x4];
top 1 y1 = h;   y5 = y1;   y2 = y4 = .36h;   bot0 y3 = —oo;
if ucs ≠ 0:   call `a serif(1, 4, 2, —ucs);
      call `b serif (1, 4, 2, ucs);                               % left serif
      call `c serif (5, 0, 4, — ucs);
      call `d serif(5, 0, 4, ucs);                               % right serif
fi;
w4 draw 1 . . 2;                                         % upper left stroke
call ` e arc(3, 2, w4);                                  % lower left stroke
w0 draw 3{1, 0} . . 4{0, 1};                            % lower right stroke
draw 4 . . 5.                                           % upper right stroke
```





```
‘The letter V”;
call $charbegin$(`V, 13, $2sc$ + $mc(ph{\cdot}slant + .5pu), 2sc - mc(rv - 2.5pu)$,
                    $ph$, 0, $mi[rv, 2\ 5pu]$);
hpen;  $\mathrm{lft}_5x_1$ = round $1.5u$;  $\mathrm{rt}_0x_4 =$ round$(r - 1.5u)$;
$x_2 - x_1 = x_4 - x_3$;  $\mathrm{lft}_0x_3 = \mathrm{lft}_5x_2$;
$\mathrm{top}_5y_1$ = $h$ ; $y_4$ = $y_1$; $\mathrm{bot}_5y_2 = -o$; $y_3$ = $y_2$;
$w_5$ draw 1 . . 2;                                        % left diagonal stroke
rpen#;  Wj draw 3.. 4;                                % erase excess at lower right
hpen;  $w_0$ draw 3. . 4;                                 % right diagonal stroke
if $ucs \neq 0$:   call ‘a $serif$(1, 5, 2, $-.5ucs$);
    call` b $serif$(1, 5, 2, $ucs$);                                % left serif
    call `c serif (4, 0, 3, $--ucs$);
    call `d $serif$(4, 0, 3, $.5ucs$);                               % right serif
fi.

“The letter W”;
call $charbegin$(`W, 18, $2sc + mc(ph{\cdot}slant + .5pu), 2sc - mc(rv-2.5pu)$,
                    $ph$, 0, $mi[rv,2.5pu]$);
hpen;  $\mathrm{lft}_5x_1 =$ round $1.5u$;
$\mathrm{rt}_0x_9 = \mathrm{rt}_5x_{10}$;  $x_{11} - x_{10} = x_9 - x_1$;  $\mathrm{rt}_0x_{11} = r - 1.5u$;
% $x_9, x_{10}$, and $x_{11}$ are approximations to $x_4, x_5$, and $x_6$.
$x_5$ -- $x_1$ = $x_6 - x_2 = x_7 - x_3$ = $x_8 - x_4$ = round$(x_{10} - x_1)$;
% The idea is to draw Lwo V’s displaced by an integer amount.
$\mathrm{top}_5y_1 = h$;  $\mathrm{bot}_5y_2 = --o$;  $y_3 =: y_6 = y_7 == y_2$;  $y_4 = y_5 = y_8 = y_1$;
$\mathrm{rt}_0x_4 = \mathrm{rt}_5x_5$;  $\mathrm{lft}_0x_3 = \mathrm{lft}_5x_2$;  $x_2 - x_1$ = $x_4 - x_3$;
$w_5$ draw 1.. 2;                                              % first diagonal
rpen#;  $w_5$ draw 3. . 4;                                      % erase excess
hpen;  $w_0$ draw 3.. 4;                                      % second diagonal
$w_5$ draw 5 . . 6;                                            % third diagonal
rpen#;  $w_5$ draw 7 8;                                       % erase excess
hpen;  $w_0$ draw 7.. 8;                                      % fourth diagonal
if $ucs \neq 0$:   call ‘a $serif$(1,5, 2,$-.5ucs$);
    if $w_0 = w_5$:   call `b $serif$(1, 5, 2, $ucs$);
    else:   call` b $serif$(1, 5, 2, $.5ucs$);
    fi;                                                          % left serif
    call `c $serif$(5, 5, 6, $-.5ucs$);
    call `d $serif$(5, 5, 6, $.5ucs$);                             % middle serif
    call `e $serif$ (9, 0, 7, $--ucs$);
    call - f $serif$(8, 0, 7,$.5ucs$);                              % right serif
fi.
```

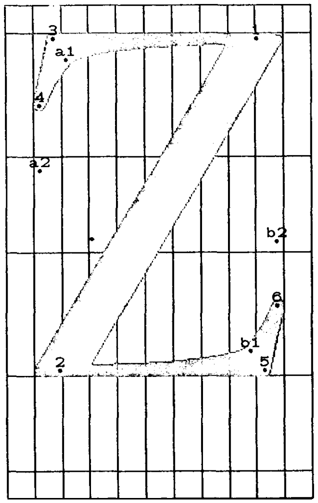


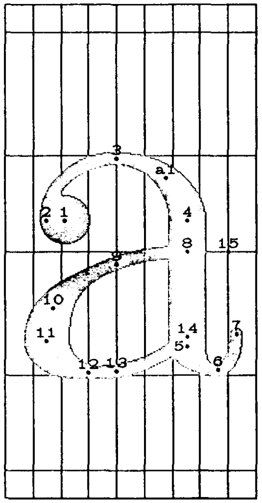


3

6

```
"The letter X";
call charbegin(`X,13, 2sc, 2sc — .5mc·rv, ph, 0, hic·rv);
h p c n ; lft_5 x_1 = r o u n d 1.5u; rt_5 x_4 == round(r — 1.5u); x_3 = x_1; x_4 == x_2;
top_5 y_1 == h; bot_5 y_1 == 0; y_2 == y_1; y_3 = y_4;
w_5 draw 1 . . 4;                                          %upper left to lower right diagonal
w_0 draw 3 . . 2;                                          %lower left to upper right diagonal
if ucs ≠ 0:   call 'a serif(1, 5, 4, —.5ucs);
      call ` b serif(1, 5, 4, ucs);                                           % upper left serif
      call . c serif(4, 5, 1, —ucs); .
      call `d serif(4, 5, 1, .5ucs);                                          %lower right serif
      new ss;   ss == .5(w_5 — w_0)/u;                       % correction to ucs makes w_0 like w_5
      call `e serif(2, 0, 3, — ucs — ss);
      call . f serif (2, 0, 3, .5ucs + ss);                                   % upper right serif
      call`g serif(3,0, 2, - .5ucs --- ss);
      call . h serif(3, 0, 2, ucs + ss);                                      % lower left serif
fi.

"The letter Y";
call charbegin(`Y, 13, 2sc + mc(ph·slant + .5pu), 2sc — mc(rv — 2.5pu),
                        ph, 0, mi[rv, 2.5pu]);
hpen; lft_5 x_1 == r o u n d 1.5u; rt_0 x_2 = round(r — 1.5u);
top_5 y_1 == top_0 y_2 == h;
rt_5 x_13 == rt_0 x_14;   x_13 - - x_1 == x_2 — x_14;            % approximations before rounding
x_3 == good_5 x_13;   rt_5 x_3 --- rt_1 x_6 == rt_0 x_4;   lft_0 x_9 == lft_5 x_3;
y_3 = y_4 = y_6 == y_9 == .48h;   y_7 == y_8 = .5[y_5, y_6];   bol_1 y_5 == 0;
x_5 == x_6 = x_8;   lft_0 x_7 == lft_1 x_8;
w_5 draw 1.. 3;                                                             % left diagonal
minvr 0;   minvs 0;
w_0 ddraw 9{x_3 — x_1, y_3 -- y_1} . . 7{0, —1}, 6. . 8;                      % transition
minvr .5;   minvs .5;
w_4 draw 6.. 5;                                                             % stem
w_0 draw 4 . . 2;                                                           % right diagonal
if ucs ≠ 0:   call '  a serif(1, 5, 3, —.5ucs);
      call ` b serif(1, 5, 3, ucs);                                           % upper left serif
      call . c serif (5, 4, 6, —ucs);
      call ` d serif (5, 4, 6, ucs);                                          % lower serif
      call `e serif (2, 0, 4, -- ucs);
      call . f serif(2, 0, 4, .5ucs);                                         % upper right serif
fi.
```

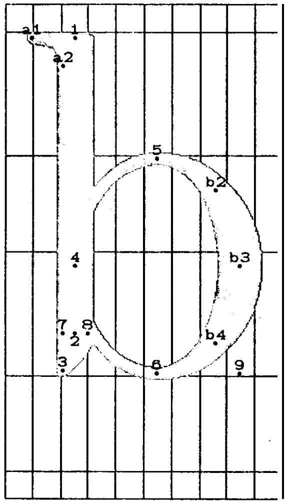


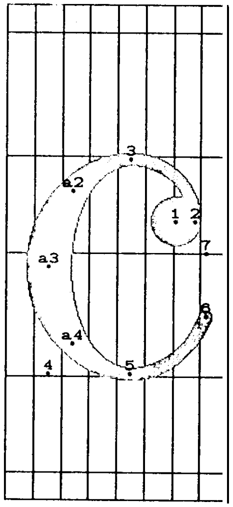


```
"The letter Z";
call charbegin(`Z, 11, 0, —.5mc(ph·slant — .5pu), ph, 0, hic(ph·slant — .5pu));
hpen;   lft_5 x_2 = round u;   rt_5 x_1 = round(r — u);
top_5 y_1 = h;   bot_5 y_2 = 0;
new ss;   ss = 1.4aspect·ucs·u + eps;
if ss + w_6 > .25h:   new ss;   ss = .25h — w_6 + eps;
fi;
lft_0 x_3 = round 1.5u;   x_4 = x_3 — .5u;   y_3 = y_1;   y_4 = y_3 — ss;
rt_0 x_5 = round(r — 1.5~);   x_6 = x_5 + .5u;   y_5 = y_2;   y_6 = y_5 + ss;
call `a arm(1, 3, 4);                                              % upper arm and serif
w_5 draw 1..2;                                                           % diagonal
call `b arm(2, 5, 6).                                              % lower arm and serif
```

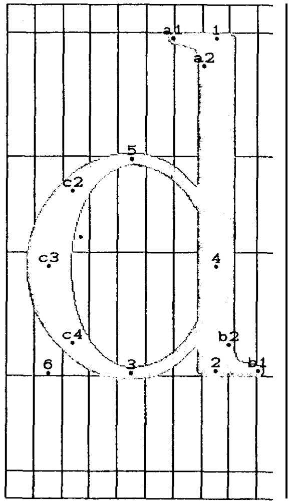
The file roman1 mf

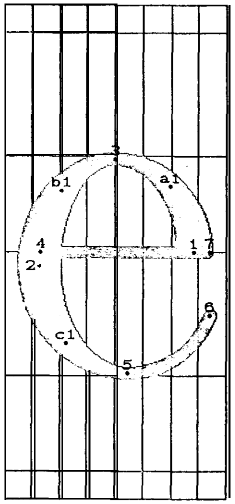
```
% This lower-case roman alphabet was prepared by D E. Knuth in November, 1979,
% inspired by the Monotype alphabet used in The Art of Computer Programming.
% Its spacing is intended for text only.
% Character codes ´141–´172 are generated.

"The letter a";
call charbegin(`a, 9, 0, sc, px, 0, 1/3[pe, px]·slant + .5pwi + (sc — 2)pu);
cpen; lft_3 x_1 = round 1.25u;
if top_3(top_3 top_6 e + 2) > .9[e, m]:   top_3 y_1 = .9[e, m];
else:   y_1 = top_3 top_6 e + 2;
fi;
w_3 draw 1;                                                                  % bulb
hpcn;   lft_0 x_2 = lft_3 x_1;   y_2 = y_1;   x_3 = 4u;   top_0 y_3 = m + oo;
x_4 = xj = good_1(r — 2.511);   y_4 = 1/3[e, m];
w_0 draw 2{0,1}.3{1,0};   call `a arc(3, 4, w_1);                      % shoulder
if lcs = 0:   bot_1 y_5 = 0;   w_1 draw 4.. 5;                             % stem
else:   y_5 = .2[top_1 0, e];   rt_0 x_6 = .5[rt_1 x_5, rt_0 x_7];
      bot_0 y_6 = 0;   rt_0 x_7 = round(r — .5u);   y_7 = e/3;
      w_1 draw 4..5;                                                       % stem
      draw |w_1#|5{0, —1}.|w_0#|6{1, 0}..7{0, 1};                         % hook
fi;
x_8 = x_4;   y_8 = e;   x_9 = 4u;   y_9 = .9[w_6, y_8];   x_10 = x_1 I + .25u;   y_10 = .5[w_6, y_8];
x_11 = good_2 1.5u;   y_11 = .2[w_6, y_8];
x_12 = 3u;   bot_0 y_12 = -- oo;   x_13 = 4u;   y_13 = .015[y_12, y_8];
x_1, = x_8;   y_14 = .3[y_12, y_8];   x_15 = r — u;   y_1 j = e;
draw |w_0|8{—1, 0}.|w_0#|9..|.8[w_0, w_2]|10..|w_2#|11{0, —1}..
      |.7[w_0, w_2]|12{1, 0}..|w_0#|13..14(..15).                          % bowl
```

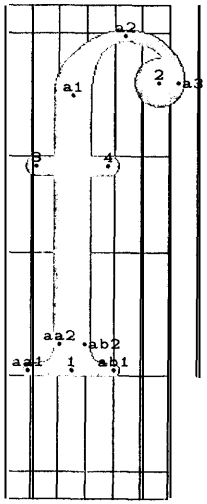




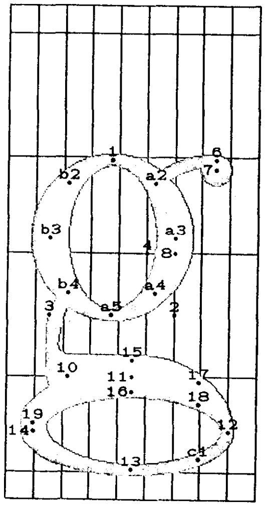
```
"The letter b";
call charbegin(`b, 10, sc, 0, ph, 0, .5px·slant + lcic);
hpen;   x1 == x2 = x4 == good, 2.5u;   top1y1 = h;
lft0x7 == lft1x1;   x3 = x7;   x5 == x6 == .5(r + u);   rt0x8 == rt1x2;
bot0y3 == 0;   y4 = .5[y5, y6];   top0y5 == m + oo;   bot0y6 == —oo;
new aa;   rt1x2 == aa[x6, x4];
y2 = y7 = y8 = (sqrt(1 -- aa·aa))[y4, y6];                  % where bowl meets stem
if lcs ≠ 0:   call `a serif(1, 1, 2, --lcs);                       % upper serif
fi;
w1 draw 1.. 2;                                                           % stem
w0 ddraw 7.. 3,8{0, -1). .3{x3 — x8, .5(y3 — y8)};                        % spur
w0 draw 6{— I, 0} . 4{0,1} . 5{1,0};                           % left part of bowl
if w2 > 1.5u:   rt2x9 == round(r — .75u);
      else:   x9 == good2(r — l.5u);
fi;
y9 = y6;   call `b darc(5, 9, w2).                        % right part of bowl

"The letter c";
call charbegin(`c, 8, 0, 0, px, 0, px·slant -- pu);
cpen;   rt3x1 = round(r — u);
if top3(top3top6e + 2) > .9[e, m]:   top3y1 == .9[e, m];
else:   y1 == top 3top6e + 2;
fi;
w3 draw 1;                                                               % bulb
hpcn;   rt0x2 = rt3x1;   y2 = y1;   x3 == Xj == .5(r + u);   top0y3 == m +- oo;
w0 draw 2{0,1}.. 3{—1, O};                                           % shoulder
if w2 > 1.5u:   lft2x4 == round(.75u);
else: x4 == good2 1.5u;
fi;
y4 == y5;   bot0y5 == —oo;
call `a darc(3, 4, w2);                                                % stroke
if w0 = w1:   x6 == x2;   x7 -- x5 == x5 -- x4;   yi == .5[y3, y5];
      new aa;   x6 == aa[x5, x7];   y6 = (sqrt(1 — aa·aa))[y7, y5];
else:      lft0x6 = rt0x2; y6 = .5e — 1;  x7 == x6; y7 == e;
fi;
w0 draw 5{ 1, 0} . 6(.7).                                               % point
```

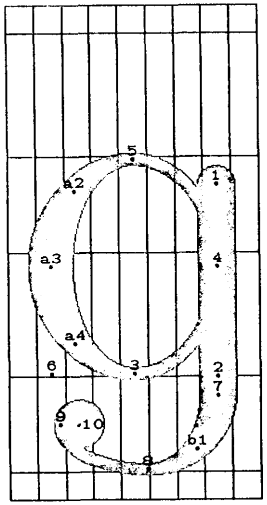




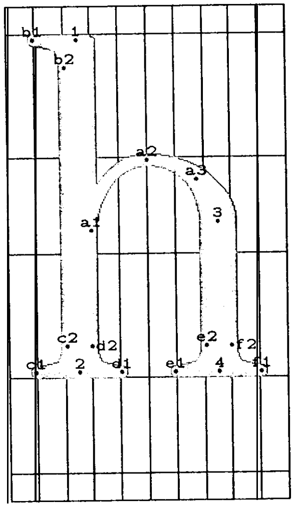


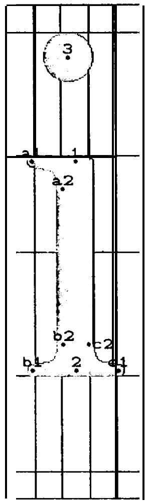
```
"The letter d";
call charbegin(`d,10, 0, sc, ph, 0, ph·slant + .5pwi + (sc — 2)pu);
hpen;   x_1 = x_2 = good_1(r — 2.5~);   top_1 y_1 = h;   bot_1 y_2 = 0;
w_1 draw 1..2;                                                  % stem
if lcs ≠ 0: call 'a serif(1, 1, 2, —lcs);                       % upper serif
     call `b serif(2, 1, 1, lcs);                               % lower serif
fi;
bot_0 y_3 = -00;   top_0 y_5 = m + 00;   y_4 = .5[y_3, y_5];
x_3 = x_5 = .5(r — u);   x_4 = x_2;
w_0 draw 3{1,0}..4{0,1}..5{—1,0};                               % right part of bowl
if w_2 > 1.5u:   lft_2 x_6 = round .75u;
else:   x_6 = good_2 1.5u;
fi;
y_6 = - y_3;   call ` c darc(5, 6, w_2).                        % left part of bowl

"The letter e";
call charbegin(`e,8,0,0, px, 0, pe·slant + lcic + .25pu);
hpen;   x_3 = .5r;   top_0 y_3 = m + 00;   y_1 = e;   y_2 = .5[y_5, y_3];
if w_2 > 1.5~:   lft_2 x_2 = round .5u;   rt_1 x_1 = round(r — .5u);
else:   x_2 = good_2 1.25u;   x_1 = good_1(r — 1.25~);
fi;
bot_0 y_5 = —oo;   xa.-- x_3 + .5u;
call `a arc(3, 1, w_1);                                         % bowl
call `b arc(3, 2, w_2);   call `c arc(5, 2, w_2);               % stroke
new aa;   y_1 = y_4 = aa[y_2, y_3];   x_4 — 1 = (sqrt(1 — aa·aa))[x_3, x_2];
w_0 draw 4..1;                                                  % bar
if w_0 = w_1:   x_6 = x_1;   x_7 — x_5 = x_1 — x_3;   y_7 = y_2;
     new an;   x_6 = aa[x_5, x_7];   y_6 = (sqrt(1 — aa·aa))[y_7, y_5];
else:   rt_0 x_6 = rt_2 x_1;   x_7 = x_6;   y_6 = .5e — 1;   y_7 = e;
fi;
w_0 draw 5{1,0}..6(..7).                                        % point

"The letter f";
call charbegin(`f,6,0,0,ph, 0, ph·slant + pu);
hpen;   x_1 = good_1 2.5u;
if fixwidth = 0:   rt_3 x_2 = round(r + .5u);
else:   rt_3 x_2 = round(r — u);
fi;
cpcn;   top_3 y_2 = .8[m, h];
call 'a fstroke(2, 1);                            % bulb, shoulder, stem, and serif
cpen;   top_10 y_3 = m;   y_4 = y_3;   lft_10 x_3 = lft_1 x_1 — u — eps;   rt_10 x_4 = rt_1 x_1 + u + eps;
w_10 draw 3..4.                                                 % bar
```

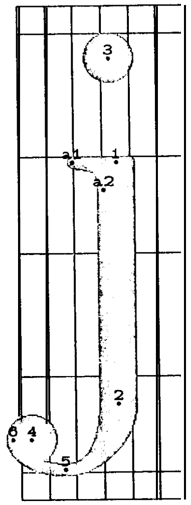




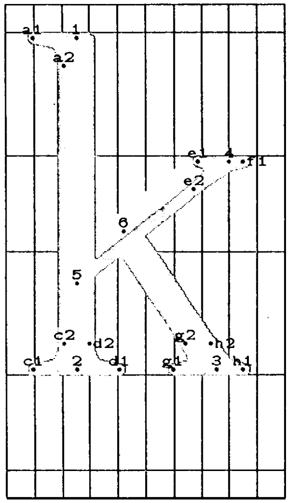
```
"The letter g";
if varg = 0:                              % the following program is for a 'classic' g shape
    call charbegin(`g, 9, 0, 0, px, pd, px·slant + .5pwiii — pu);
    hpen;   x3 = good1 1.5u;   x2 = good, 2/3r;   x1 — x3 = x2 — x1;
    top0y1 = m + oo;   y2 = y3 = round .5e;
    call 'a darc( 1, 2, w1);   call ` b darc(1, 3, w1);                          % bowl
    x4 = x2 — u ;   y4 = e;
    x5 = 1/sqrttwo[x1, x2];   y5 = 1/sqrttwo[.5y1 + .5y2, y1];
    x6 = x 7 =  r — 1.5u;
    new w99;   w99 = round .5[w6, w3];
    cpen;   top6y6 = top99y7 = m + 0 0;
    w99 draw 7;   w6 draw (4 . )5 . 6{1, O } ;                                    % ear
    vpen;   x8 == x2;   y8 = e;
    x9 = 1/sqrttwo[x1, x3];   y9 = 1/sqrttwo[.5y1 + .5y3, y3];
    x10 = x9;   y10 = y11 = good, 0;   x11 == x1 3 == .5r;   x12 == good0(r — u);
    y12 = .5[bot8y11, top6y13];   bot6y13 = —d — oo;
    draw (8.. )|w6|9 . . |w8#|10{1, 0}  . 11{1, 0};                     % left part of link
    top8y11 = top6y15;   x15 == x16 = x11;   bot8y11 = bot6y16;
    x17 == 1/sqrttwo[x15, x12];   y17 = 1/sqrttwo[y12, y15];
    x18 = 1/sqrttwo[x16, x12];   y18 = 1/sqrttwo[y12, y16];
    hpen;   w0  ddraw  15{1, 0}  17{x12 - - x1 5, y12 — y15} 12{0, —1},
        16{1, 0} . 18{x12 — x16, y12 — y16} . . 12{0, —1};                % right part of link
    call `c arc(13, 12, w0);                                            % right part of tail
    new w99;   w99 == round .5[w0, w1];
    x14 = x19 = good99u;   y14 = .5[y13, y16];   y19 == .5[y13, y11];
    w99 ddraw 13{--1, 0} . . 14{0, 1} . . 16{1, 0},
        13{—1, 0} . . 19{0, 1} . 11{1, 0};                               % left part of tail
else:                                     % the following program is for a 'simple' g shape
    call charbegin(`g, 9, 0, 0, px, pd, .9px·slant + .5pwi -- pu);
    hpen;   x1 == x2 = good1(r — 1.5u);   x4 == x1;   x3 == x j == .5r;
    bot0y3 == —oo;   top0y5 == m + oo;   y4 = .5[y3, y5];
    w0 draw 3{ 1, 0} . . 4{0, 1} . . 5{—1, 0};                         % right part of bowl
    if w2 > 1.5~:   lft2x6 = round .75u;
    else:   x6 = good2 1.521;
    fi;
    y6 == y3;   call 'a darc(5, 6, w2);                                 % left part of bowl
    cpen;   top1y1 == .9[e, m ] ; y2 == 0 ;   w1 d r a w l . 2 ;                      % stem
    hpen;   x7 == x2; bot1y7 == — .25d;   w1 draw 2.. 7;                              % link
    x10 == good3 2.5u;   x8 == .5[x10, x7];   lft0x9 == lft3x10;
    y9 = y10;   bot0y8 == --d - 0 0 ;
    cpen;   bot3y10 =  --. 75d;   w3 draw 10;                                         % bulb
    hpen;   w0 draw 9{0, —1} . . 8{ I, O};   call `b arc(8, 7, w1);                 % tail
fi.
```

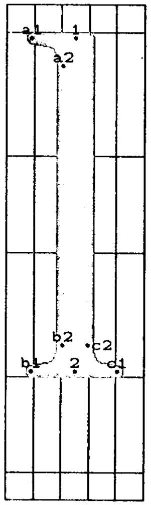


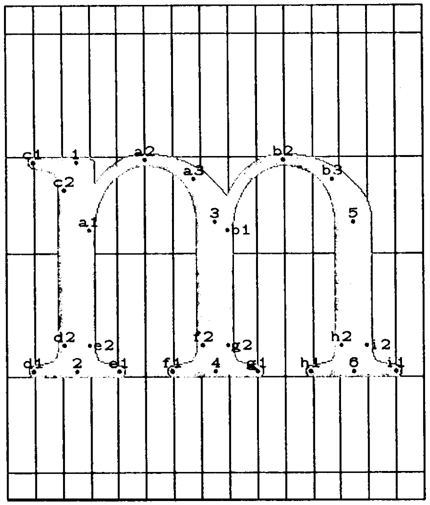




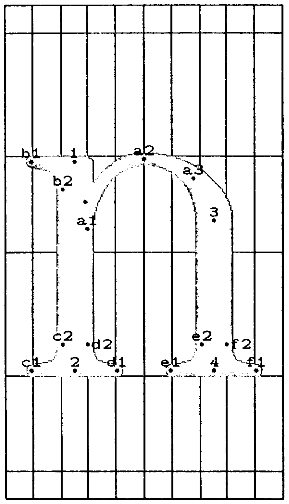
```
"The letter h";
call charbegin(` h, LO, sc, sc, ph, 0, $\frac{1}{3}$[pe, px]·slant + .5pwi + (sc — 2)pu);
hpen;   $x_1$ = $x_2$ = good, 2.5u;   $x_3$ = $\text{good}_1$(r — 2.5u);
$\text{top}_1 y_1$ = h ;   $\text{bot}_1 y_2$ = 0;
$w_1$ draw 1 .. 2;                                          % left stem
call ` a hstrokc(2, 3, 4);                                  % shoulder and right stem
if Jcs ≠ 0:   call ` b serif(1, l, 2, —lcs);                % upper serif
      call ` c serif(2, 1, l, —lcs);
      call ` d serif(2, 1, 1, lcs);                         % lower left serif
      call ` e serif(4, 1, 3, —lcs);
      call ` f serif (4, 1, 3, lcs);                        % lower right serif
fi.

"The letter i";
call charbegin(` i, 5, sc, sc, ph, 0, ph slant + .5pwi + (sc -- 2)pu);
hpen;   $x_1$ = $x_2$ = good, .5r;   top, $y_1$ = m;   $\text{bot}_1 y_2$ = 0;
$w_1$ draw 1 .. 2;                                          % stem
cpen;   $\text{top}_3 y_3$ = h ;   $\text{rt}_3 x_3$ = $\text{rt}_1 x_1$;   $w_3$ draw 3;   % dot
if Jcs ≠ 0:   call 'a serif(1, l, 2, —lcs);                 % upper serif
      call . b serif(2, 1, 1, --Jcs);
      call ` c serif(2, 1, 1, Jcs);                         % lower serif
fi.

"The letter j";
call charbegin(`j, 6, s c, 0, ph, pd, ph·slant + .5pwi — 2pu);
hpen;   $x_1$ = $x_2$ = $\text{good}_1$(r — 2.5u);
if fixwidth = 0:   $\text{lft}_3 x_4$ = round(—.5u);
else:   $\text{lft}_3 x_1$ = round u;
fi;
cpen;   $\text{top}_3 y_3$ = h ;   $\text{rt}_3 x_3$ = $\text{rt}_1 x_1$;   $w_3$ draw 3;   % dot
$\text{bot}_3 y_4$ = —.9d; $w_3$ draw 4 ;                    % bulb
hpen; $\text{top}_1 y_1$ =   m  ; $\text{bot}_1 y_2$ = —$\frac{1}{3}$d;
$\text{bot}_0 y_5$ = —d — oo;   y, = $y_6$;   $\text{lft}_0 x_6$ = $\text{lft}_3 x_4$;   $x_5$ = .5[$x_2$, $x_6$];
draw |$w_1$|1 .. |$w_1$#|2{0, -1). |$w_0$#|5{—1, 0} .. 6{0, 1} ;   % stem and tail
if Jcs ≠ 0:   call 'a serif(1, 1, 2, -1cs);                 % serif
fi.
```

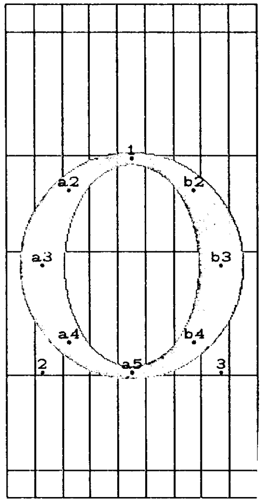


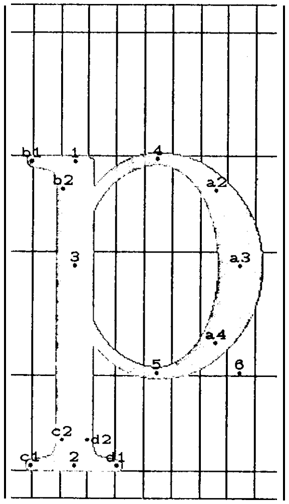




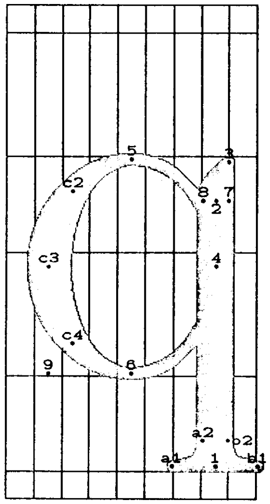
```
"The letter k";
call charbegin(`k, 10, sc, sc, ph, 0, px·slant + .5pwi + (lcss + sc — 2)pu);
hpen;   x1 = x2 = x5 == good1 2.5u;
x3 == good1(r   .2.5u);   rt1x3 = rt0x4;
top1y1 = h ; top0y4 == m  ; bot1y2 == bot1y3 =  0 ;    y5 = .75e;
new aa, bb;                                      % auxiliaries for intersection of lines
rt1x6 == aa[rt1x1, rt1x3];   y6 == aa[bot0m, y3];
rt1x6 == bb[x5, x4];   y6 == bb[y5, y4];
w1 draw 6.. 3;                                                       %lower diagonal
lpen#; w1 draw 4. 5;                                                 %erase excess
hpcn;   w1 draw 1 . . 2;                                                   % stem
w0 draw 4. . 5;                                                   % upper diagonal
if lcs ≠ 0:   call `a serif(1, 1, 2, —lcs);                       %upper stem serif
     call . c serif(2,1,1, —lcs);
     call `d serif(2, 1, 1, lcs);                                 %lower stem serif
     call . e serif(4, 0, 5, —lcs);
     call `f serif (4, 0, 5, lcss);                            %upper diagoriat serif
     call `g serif (3, 1, 6, —lcs);
     call `h serif(3, 1, 6, lcss);                              %lower diagonal serif
fi.

"The letter l";
call charbegin(` 1, 5, sc, sc, ph, 0, ph·slant + .5pwi + (sc – 2)pu);
hpen;   x1 = x2 = good, .5r;   top1y1 = h;   bot1y2 == 0;
w1 draw 1 . . 2;                                                           % stem
if lcs ≠ 0:   call `a serif(1, 1, 2, —lcs);                            % upper serif
     call . b serif(2, 1, 1, —lcs);
     call ` c serif (2, 1, 1, lcs);                                    % lower serif
fi.

"The letter m";
callcharbegin(`m, 15, sc, sc, px, 0, 1/3[pe, px]·slant +.5pwi + (sc — 2)pu);
hpen;   x1 == x2 == good1 2.5u;   x3 == good, .5r;   x5 — x3 = x3 — x1;
top1y1 = m;   bot1y2 = 0;
w1 draw 1 . . 2;                                                      %left stem
call 'a hstrokc(2, 3, 4);                             %left shoulder and middle stern
call ` b hstroke(4, 5, 6);                            %right shoulder and right stem
if lcs ≠ 0:   call `c serif(1, l, 2, —lcs);                            %upper serif
     call `d serif(2, 1, 1, —lcs);
     call ` e serif (2, l, 1, lcs);                                %lower left serif
     call ` f serif(4, 1, 3, —lcs);
     call `g serif(4,1,3,lcs);                                   %lower middle serif
     call ` h serif(6, 1, 5, —lcs);
     call ` i serif(6, 1, 5, lcs);                                % tower right serif
fi.
```

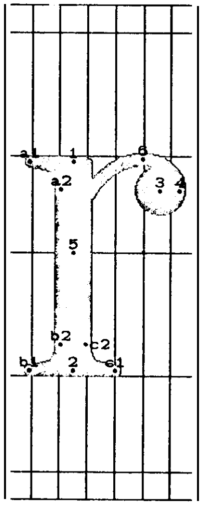




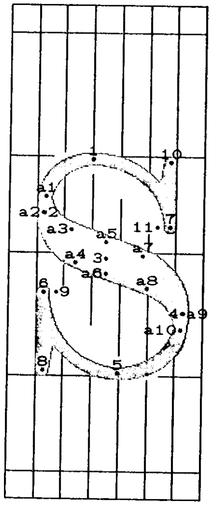


```
"The letter n";
call charbegin(`n, LO, sc, sc, px, 0, 1/3[pe, px]·slant + .5pwi + (sc — 2)pu);
hpen;   x_1 = x_2 == good, 2.5~;   x_3 == good_1(r — 2.5~);
top_1 y_1 == m;   bot_1 y_2 = 0;
w_1 draw 1..2;                                              %left stem
call `a hstroke(2, 3, 4);                      % shoulder and right stem
if lcs ≠ 0:   call . b serif(1, 1, 2, —lcs);               % upper serif
     call ` c serif (2, 1, 1, —lcs);
     call `d serif(2, 1, 1, lcs);                      % tower left serif
     call `e serif(4, I, 3, —lcs);
     call `f serif (4, 1, 3, lcs);                    % lower right serif
fi.

"The letter o";
callcharbegin(`o, 9, 0, 0, px, 0, .5px·slant);
hpen;   x_1 = r — x_1;                      % axis of lefl-right symmetry
lft_2 x_2 = round fixwidth[.5u, 1.54;
x_1 — x_2 = x_3 — x_1;   top_0 y_1 == m + 00;   bot_0 y_2 == -00;   y_2 = y_3;
call `a darc(1, 2, w_2);                            % left part of bowl
call ` b darc(1, 3, w_2).                           % right part of bowl

"The letter p";
callcharbegin(` p, 10, sc, 0, px, pd,.5px·slant + lcic);
hpen;   x_1 = x_2 == x_3 == good, 2.5u;   x_4 == x_5 = .5(r + u);
if w_2 > 1.521:   rt_2 x_6 = round(r — .75u);
else:   x_6 = good_2(r — l.5u);
fi;
top_1 y_1 = m;   bot_1 y_2 = —d;   top_0 y_1 == m + 00;   bot_0 y_5 = -00;
y_3 = .5[y_4, y_5];   y_6 = y_5;
w_1 draw 1.. 2;                                                 % stem
w_0 draw 5{—1, O}..3{0, 1}.4{1, 0};                  % left part of bowl
call ` a darc(4, 6, w_2);                           % right part of bowl
if lcs ≠ 0:   call `b serif(1, l, 2, —lcs);               % upper serif
     call ` c serif (2, 1, 1, —lcs);
     call ` d serif(2, 1, 1, lcs);                          % tower serif
fi.
```

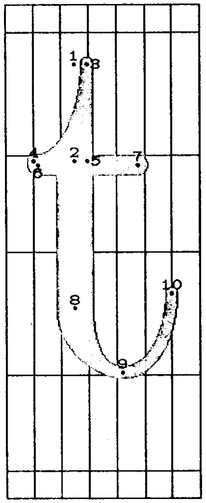


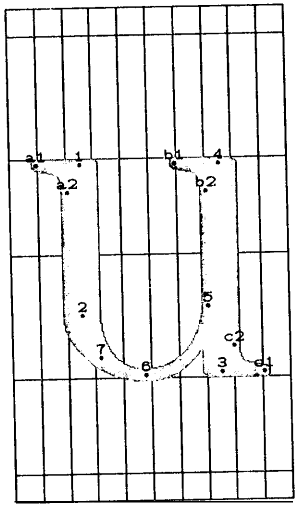


```
'The letter q";
call charbegin(`q,10,0,1, px, pd, px·slant + .5pwi — pu);
hpen;   x_1 == good_1(r — 2.5u);   x_5 = .5(r — u);   x_2 = x_1 = x_4;   bot_1y_1 = - d ;
rt_0x_7 = rt_1x_1;   x_3 = x_7;   x_5 = x_6;   lft_0x_8 = lft_1x_2;
top_0y_3 = m;   y_4 = .5[y_5, y_6];   top_0y_5 == m + oo;   bot_0y_6 = —oo;
new aa;   lft_1x_2 = aa[x_6, x_4];
y_2 = y_7 = y_8 = (sqrt(1 — aa·aa))[y_4, y_5];                    %where bowl meets stem
if lcs ≠ 0:   call 'a serif(1, 1, 2, —lcs);
      call `b serif(1, 1, 2, lcs);                                           % lower serif
fi;
w_1 draw 1 . . 2;                                                                    % stem
w_0 ddraw 7 . . 3, 8{0, 1} . 3{x_3 — x_8, .5(y_3 — y_8)};                           % spur
w_0 draw 6{1, 0} . . 4{0, 1}. 5{—1, 0};                                 % right part of bowl
if w_2 > 1.5u:   lft_2x_9 = round .75u;
else:   x_9 == good_2 1.5u;
fi;
y_9 = y_6; call `c darc(5, 9, w_2).                                      % left part of bowl

"The letter r";
call charbegin(` r, 7, sc, 0, px, 0, px·slant);
hpcn;   x_1 = x_2 = good, 2.5u;   top_1y_1 == m;   bot_1y_2 = 0;
w_1 draw 1 . . 2;                                                                    % stem
cpen;   rt_3x_3 = rt_0x_4 = round(r — .5u);   top_3y_3 = .9[e, m];   y_1 = y_3;
w_3 draw 3;                                                                          % bulb
hpen;   x_5 = x_1;   y_5 = e;   x_6 == 5u;   top_6y_6 == m + oo;
w_0 draw 5{0, 1} . 6{1, 0} . . 4{0, -1};                                        % shoulder
if lcs ≠ 0:   call 'a serif(1, 1, 2, —lcs);                                 % upper serif
      call `b serif(2, 1, 1, —lcs);
      call `c serif(2, 1, 1, lcs);                                           % lower serif
fi.
```

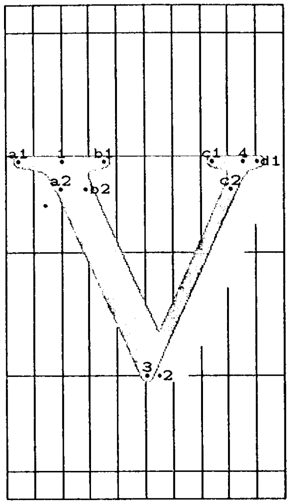


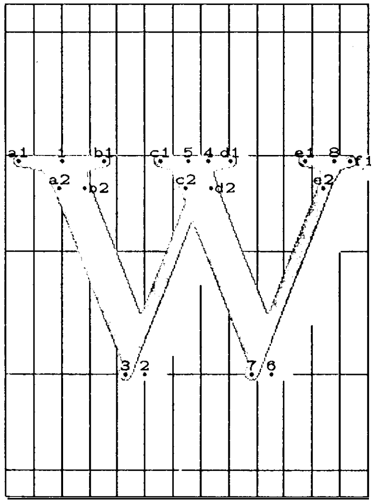
```
"The letter s";
call $charbegin($`s, 7, 0, 0, px, 0, $px \cdot slant - .5pu)$;
hpen;   $top_0 y_1 = m + oo$;   $bot_0 y_5 = -oo$;
$x_3 = .5r$;   $(y_3 - w_6 - .5w_8)/(m - 2w_6 - w_8) = e/m$;          % white space ratio
$lft_{11} x_2$ = round $u$;
if $lcs = 0$:   $x_1 = x_5 = x_3$;   $lft_0 x_6$ = round $.75u$;   x 7 $= x_4$;   rt $_{11} x_1$ = round$(r - u)$;
    $y_6 = \frac{1}{3}[top_0\, 0, y_3]$;   $y_7 = \frac{1}{4}[bot_0 m, y_3]$;
    $w_0$ draw $6\{x_5 - x_6, 3(y_5 - y_6)\} .5\{1, O\}$;          % lower left stroke
    draw $7\{x_1 - x_7, 2(y_1 - y_7)\} .1\{-1, O\}$;          % upper right stroke
else:   if $w_0 = w_1$:   $x_1 = x_5 = x_3$;   $x_6 = x_2$;   $x_7 = x_4$;   $rt_{11} x_4$ = round$(r - u)$;
        $y_6 = \frac{1}{3}[top_0\, 0, y_3]$;   $y_7 = \frac{1}{4}[bot_0 m, y_3]$;
        $w_0$ draw $6\{x_5 -- x_6, 3(y_5 -$ z/(j)}. $.5\{1, 0\}$;          % lower left stroke
        draw $7\{x_1 -- x_7, 2(y_1 - y_7)\} .1\{-1, 0\}$;          % upper right stroke
        $x_8 = x_9 = x_6$;   $y_9 = y_6 + lcs \cdot aspect \cdot u + eps$;   $bot_0 y_8 = 0$;
        $x_{10} = x_{11}$ = x 7 ;   $y_{11} = y_7 - lcs \cdot aspect \cdot u - eps$;   $top_0 y_{10} = m$;
        $w_0$ draw 8. . 9;          % lower serif
        draw 10.. 11;          % upper serif
    else:   $x_1$ = x 3 $- .5u$;   $x_5 = x_3 + .5u$;   $x_6 = x_2$;   $rt_0 x_7$ = round$(r -- u)$;   rt $_{11} x_4$ =
round$(r - .5u)$;
        $y_6 = good_6(\frac{5}{7} y_3) - 1$;   $y_7 = good_6(\frac{2}{7}[y_3, m]) + 1$;
        $bot_0 y_8 = 0$;   $y_9 = y_6$;   $x_8 = x_6$;   $rt_1 x_6 = rt_0 x_9$;
        $top_0 y_{10}$ = m;   $y_{11} = y_7$;   $x_{10} = x_7$;   lft $_1 x_7 = lft_0 x_{11}$;
        $w_0$ ddraw 6. . 8, 9. . 8;          % lower serif
        ddrnw 7.. 10, 11 . . 10;          % upper serif
        rpen#;   $w_1$ draw 6{0, - 1 ) .5{1, 0};          % erase excess
        lpen#;   $w_1$ draw $7\{0, 1\} .1\{-1, 0\}$;          % ditto
        hpen;   $w_0$ draw $6\{0, -1\} .. 5\{1, O\}$;          % lower left stroke
        draw $7\{0, 1\} .1\{-1, O\}$;          % upper right stroke
    fi;
fi;
new $aa$;
if m $- e >$ e:   $aa = m - e$ ;
else:   $aa$ = e;
fi;
call 'a $sdraw(1, 2, 3, 4, 5, w_{11}, w_8, -aa/(12u))$.          % middle stroke
```



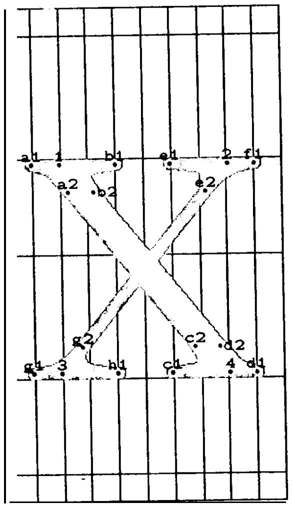


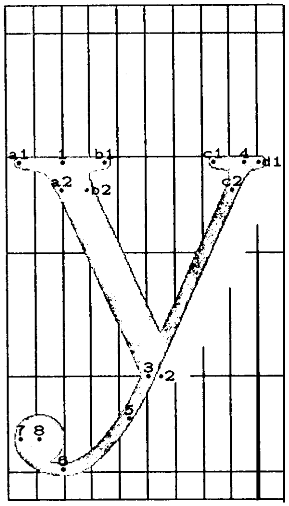
```
“The letter t”;
if $px - pe < .75(ph - px)$:    call charbegin(` t, 7, 0, 0, $2px - pe$, 0, $px \cdot slant + .5pwi - 2pu$);
      $y_1 = 2m - e$;
else:   call charbegin(`t, 7, 0, 0, .75[px, ph], 0, $px \cdot slant + .5pwi - 2pu$);
      $y_1 = .75[m, h]$;
fi;
hpen;   $x_1 = x_2 =$ good, 2.5u;   top$_0 y_2 = m$;
if $w_0 = w_1$:   w, draw 1 . . 2;                                                  % upper terminal
else:   rt$_1 x_1$ = rt$_0 x_3$;   lft$_0 x_4$ = lft$_1 x_1 - u -$ eps;   $y_4 = y_5 = y_2$;   $y_3 = y_1$;   xj $= x_3$;
      $w_0$ ddraw 4{1, 0} . . 3{0, 1}, 4. . 5;                                   % left and upper terminals
fi;
lft$_{10} x_6$ = lft$_1 x_1 - u -$ eps;   rt$_{10} x_7$ = rt$_1 x_1 + 2u +$ cps;
cpen;   top$_{10} y_6$ = m ; $y_7 = y_6$; $w_{10}$ draw 6. . 7;                          % bar
hpen;   $x_8 = x_1$;   bot$_0 y_8 = .5e$;   $w_1$ draw 2. . 8;                            % stem
if $w_0 > 1.5u$:   bot$_0 y_9 = 0$;   $x_9 = r - 2.5u$;   rt$_0 x_{10} = r - .5u$;   $x_{11} = r + u$;   $y_{11} = e$;
      new aa;   $x_{10}$ = aa[$x_9$, x11] ;   $y_{10}$ = (sqrt(1 — aa·aa))[$y_{11}$, $y_9$];
      draw |$w_1$#|8{0, —1} |$w_0$#|9{ 1, 0 } 10( 1 1 ) ;                             % hook
else : bot$_0 y_9$ = —oo; $y_{10} = \frac{2}{3}e$; $x_9 = .5[x_8, x_{10}]$; $x_{10}$ = good$_0$(r — u);
      draw |$w_1$#|8{0, --1)‘. |$w_0$#|9{1, 0} . 10{0, 1};                           % hook
fi.

“The letter u”;
call charbegin(` u, 10, sc, sc, px, 0, $px \cdot slant + .5pwi + (sc - 2)pu$);
hpen;   $x_1 = x_2 =$ good, 2.5u;   $x_3 = x_4$ = good$_1$(r — 2.5~);   lft$_0 x_5$ = lft$_1 x_3$;
top$_1 y_1$ = m;   m — $y_2 = \frac{1}{3}[e, m]$;   m — $y_5 = \frac{1}{4}[e, m]$;   % prepare for upside-down hstrokc
$x_6 = .5[x_2, x_5]$;   bot$_0 y_6$ = —oo;
x 7 = 1/sqrttwo[$x_6$, $x_2$];   y 7 = 1/sqrttwo[$y_2$, $y_6$];
draw |$w_0$|5{0, —1} . |$w_0$#|6{—1, 0} . |.6[$w_0$, $w_1$]|7{$x_2$ -- $x_6$, $y_2$ — $y_6$} . .
      |$w_1$#|2{0, 1} . 1 ;                                                        % stroke
y-1 = $y_1$;   bot$_1 y_3 = 0$;   $w_1$ draw 3. . 4;                                     % stem
if lcs $\neq$ 0:   call `a serif(1, 1, 2, —lcs);                                   % upper left serif
      call `b serif(4, 1, 3, —lcs);                                                % upper right serif
      call `c serif(3, 1, 4, lcs);                                                 % lower serif
fi.
```

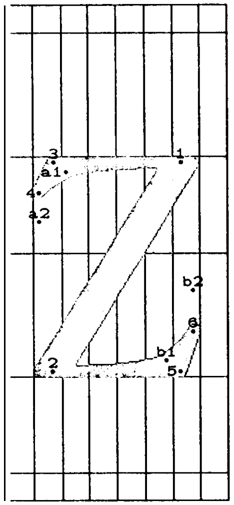




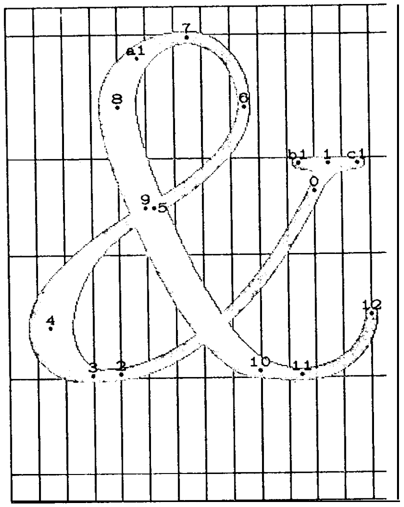
```
' The letter v";
call charbegin(`v, 10, sc, sc, px, 0, px·slant + .5pw + (lcs +· sc -- 1.5)pu);
hpen;   x1 = good, 2u;   x1 — good0(r + (Jcs -- Jcss — 2)u);
x2 — x1 = x4 — x3;   lft0x3 == lft1x2;
top 1y1 == m;   y4 = y1;   bot1 y2 == -0 ;   y3 == y2;
w1 draw 1.. 2;                                          % left diagonal stroke
rpen#; w1 draw 3.. 4;                          % erase excess at lower right
hpen;  w0 draw 3. 4;                                  % right, diagonal stroke
if lcs ≠ 0:   call `a serif(1,1, 2, —lcs);
     call` b serif (l, 1, 2, lcs);                                  % left serif
     call . c serif(4, 0, 3, —lcs);
     call . d serif(4, 0, 3, lcss);                                % right serif
fi.

"The letter w";
call charbegin(`w, 13, sc,  sc, px, 0, px slant + .5pw + (lcs + sc -- 1.5)pu);
hpen; x1 == good, 2u;   rt0x9 == rt1x10;   x11 — x10 = x9 — x1;   x11 == r + (lcs — Jcss -- 2)u;
% x9, x10, and x11 are approximations to x4, x5, and x6.
x j -- x1 == x6 — x2 == x7 — x3 == x8 — x1 == round(x10 --- x1);
% The idea is to draw two v's displacedby an integeramount.
top1y1 == m;   bot1y2 == o;   y3 == y6 == y7 == y2;   y4 == y5 == y8 == y1;
rt0x4 == rt5x5;   lft0x3 == lft5x2;   x2 — x1 == x4 — x3;
w1 draw 1. .2;                                              % first diagonal
rpen#; w1 draw 3.. 4;                                        % erase excess
hpen;  w0 draw 3 4;                                        % second diagonal
w1 draw 5.. 6;                                              % third diagonal
rpen#;  w1 draw 7. .8;                                       % erase excess
hpen;  w0 draw 7.. 8;                                      % fourth diagonal
if Jcs ≠ 0:   call `a serif(1, I, 2, —lcs);
     call . b serif(1,1, 2, lcs);                                  % left serif
     call ` c serif(5,1, 6, —lcss);
     call `d serif(5, I, 6, Jcs);                                % middle serif
     call . e serif(8, 0, 7, --lcs);
     call ` f serif (8, 0, 7, lcss);                               % right serif
fi.
```

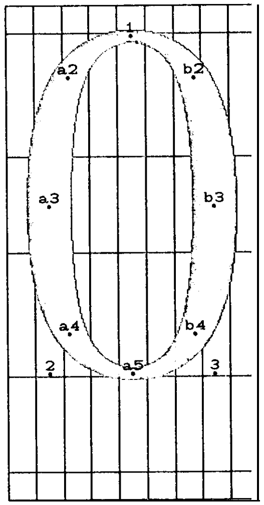




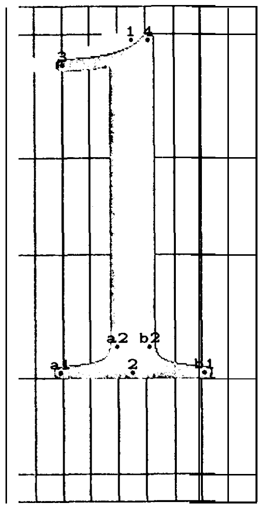


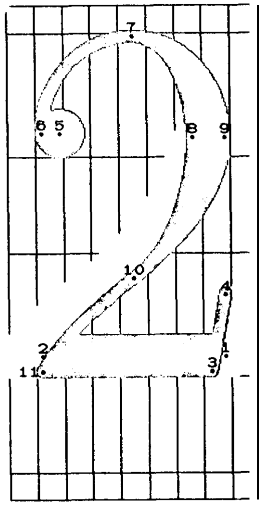
```
"The letter x";
callcharbegin(`x, 10, sc, sc, px, 0, px·slant+.5pwi + (sc + Jcs —2)pu);
hpen;   x1 == x3 = good1(2.5 — Jcs + lcss)u;   x2 == x4 = r — x1;
top, y1 = top0y2 = m;   bot0y3 = bot 1y4 = 0;
w1 draw 1 .. 4;                                          % upper left to lower right diagonal
w0 draw 3.. 2;                                           % lower left to upper right diagonal
if Jcs ≠ 0:   call 'a serif(1, 1, 4, —lcss);
      call `b serif(1, 1, 4, 2[lcss, lcs]);                          % upper left serif
      call `c serif(4, 1, 1, —2[lcss, lcs]);
      call `d serif(4, 1, 1, Jcss);                                 % lower right serif
      new ss;   ss = .5(w1 — w0)/u;                    % correction to lcs makes w0 like wl
      call `e serif(2, 0, 3, -- 2[lcss, lcs] — ss);
      call . f serif (2, 0, 3, lcss + ss);                           % upper right serif
      call `g serif(3, 0, 2, —lcss — ss);
      call` h serif (3, 0, 2, 2[lcss, lcs] + ss);                    % lower left serif
fi.

"The letter y";
call charbegin(`y, 10, sc, sc, px, pd, px·slant + .5pw + (Jcs + sc —1.5)pu);
hpen;   x1 == good, 2u;   x4 == good0(r + (Jcs -- Jcss --- 2)u);
x2 — x1 == x4 -- x3;   lft0x3 = lft 1x2;
top1y1 = m;   y4 = y1;   bot1y2 == - 0 ;   y3 = y2;
w1 draw 1. . 2;                                                % lrft diagonal stroke
rpen#;   w1 draw 3.. 4;                                  % crasc excess at lower right
cpen;   lft3x8 == lft0x7 == round .25u;
bot3y8 == —.9d; y7 == y8; w3 draw 8 ;                                        % bulb
hpen;   x6 == 2u;   bot0y6 == —d — oo;   bot0y5 == —.5d;
new aa;   x5 == aa[x3, x1];   y5 = aa[y3, y1];
w0 draw 4.. 3{x3 — x4, y3 — y1} . . 6{—1, 0} . . 7{0, 1};     % right diagonal and tail
if Jcs ≠ 0:   call `a serif( I, 1, 2, --lcs);
      call . b serif(1,1, 2, Jcs);                                        % left serif
      call ` c serif (4, 0, 3, — Jcs);
      call ` d serif (4, 0, 3, lcss);                                    % right serif
fi.

"The letter z";
call charbegin(`z, 8, 0, 0, px, 0, px·slant -- .5pu);
hpen;   lft1x2 == round u;   rt1x1 = round(r — u);
top1y1 == m;   bot1y2 == 0;
new ss;   ss == 1.4aspect·lcs·u + eps;
if ss + w6 > .25m:   new ss;   ss == .25m — w6 + eps;
fi;
lft0x3 = round 1.5u;   x4 = x3 — .5u;   y3 = y1;   y4 == y3 — ss/1.4;
rt0x5 == round(r — 1.5u);   x6 = x5 + .5u;   y5 = y2;   y6 == y5 + ss;
call 'a arm(1, 3, 4);                                      % upper arm and serif
w1 draw 1. . 2;                                               % diagonal stroke
call `b arm(2, 5, 6).                                      % lower arm and serif
```



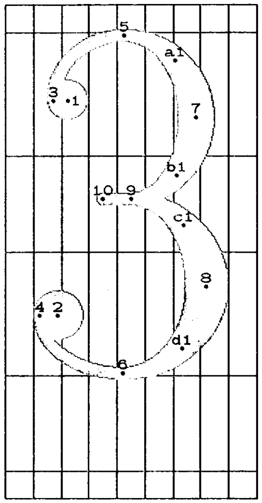


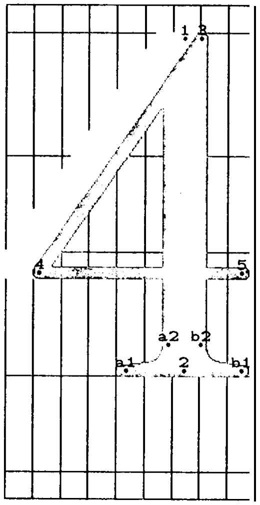
```
% This file contains the ten digits, as well as '&' and '?',
% in the so-called roman style.
% Codes '046, '060-'071, and '077 are used.

"Ampersand";
call charbegin( '046, 14, 0, 0, ph, 0, .5pe·slant + .5pw — .5pu);
hpen;   x1 = r — 2.5u;   top0y1 == m ;
x2 = 4u;   y2  =  .015[y3, e]; x3  =  3u; bot0y3  =  —oo;
x4 == good, 1.5u;   y4 == 1/7 [y3, y7];   x5 == 5.25u;   y5  =  5[y3, y7];
x6  =  good0 8.5u;   y6 == .8[y3, y7];   x7 == 6.5,;   top0y7 = h  + oo;
x8 == good1 4u;   y8 == y6;   x9 == x8 + u;   y9 == y5;   x10  = r — 5u;   y10  =  .015[y3, y7];
x11  =  x10 + 1.5u;   y1 I = y3;   x12  =  good0(r — u);   y12 == .5[y11, e];
if s = 0:   y0  =  .1[y1, y2];
else:   y0  =  y1 — s;
fi;
(x1 — x0)/(y1 — y0)  =  .5(x1 — x2)/(y1 - - y2);
w0 draw  1.. 0;                                                      % short diagonal
d r a w 0{x0 — x1, y0 -- y1}|w0#|2 . . |2/3[w0, w2]|3{--1, O}.  . |w2#|4{0, 1} .
      |w0#|5{x5 — x1, y5 -- 1/3[y3, y7]} . .
      6{0,1} . . 7{—1, O};                       % lower bowl and right part of upper bowl
call `a arc(7, 8, w1);                                        % left part of upper bowl
d r a w |w1|8{0, —1} . . 9 . . |w1#|10 . . |w0#|11{1, 0} . 12{0, 1};         % long diagonal
if Jcs ≠ 0:   call ` b serif (I, 0, 0, —lcs);
      call ` c serif (1, 0, 0, Jcs);                                               % serif
fi.

"The numeral 0";
call charbegin(`O, 9, 0, 0, ph, 0, ph·slant — .5pu);
if fixwidth = 0: new save;   save = sqrttwo;   new sqrttwo;
      sqrttwo == sqrt(1.23114413save);                          % the constant is 2^(3/10)
fi;
hpen;
i f w2 > 1.5u:   lft2x2 = round .75u;
else:   x2 == good2 1.5u;
fi;
x1 = r --- x1;                                            % axis of left-right symmetry
x3 = r — x2;   top0y1 = h  + oo;   bot0y2 = —oo;   y3 = y2;
call ` a darc( I, 2, w2);   call . b darc( I, 3, w2);                              % bowl
i f fixwidth == 0: n e w sqrttwo;   sqrttwo == save;
fi.
```



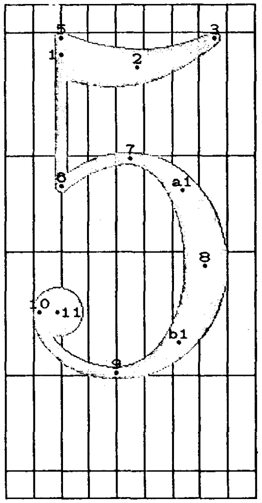


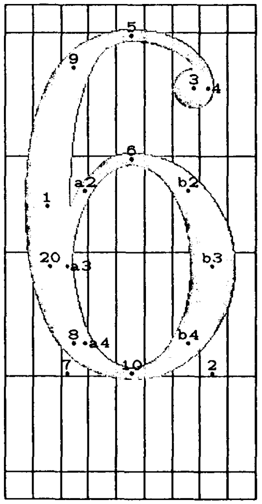
```
"The numeral 1";
call charbegin(`1, 9, 0, 0, ph, 0, ph·slant — .5pu);
hpen;   x1 = x2 = good2 .5r;   top2y1 == h;   bot2y2 = 0;
w2 draw 1 . . 2;                                                        % stem
call ` a serif (2, 2, 1, -2);
call ` b serif (2, 2, 1, 2);                                            % serif
if w0 = w2:   if lcs == 0:   top0y3 == .8[m, h];   x3 = x1 — 2u — cps;
        clsc:   top0y3 == .2[m, h];   x3 == lft2x1 — 2u — eps;
        fi;
else:   top0y3 = .8[m, h];   x3 = x1 — 2.5u — eps;
fi;
y4 == y1;   rt2x1 == rt0x4;   y5 == 1.5[m, h];   x4 == x5;
lpen#;   w2 draw (5. . )4 . . 3{—1, 0};                                  % erase excess
hpcn;   w0 draw (5. . )4 . . 3{—1, 0}.                                  % point

"The numeral 2";
call charbegin(`2, 9, 0, 0, ph, 0, ph·slant — .5pu);
vpen;   new w99;
if lcs = 0:   w99 = w4;
else:   w99 = w8;
fi;
rt99x1 == round(r — u);   lft99x2 == round u;   bot99y1 = 0;   y1 == y2;
w99 draw 1 . . 2;                                                       % bar
if lcs ≠ 0:   x4 = x1;   top6y4 = top99y1 + ucs·aspect·u + eps;   bot6y3 == 0;
        if w7 == w8:   x3 == x1;
        else:   x3 + .5u = x1;   rpen#;   w0 + .5u draw 3 . . 4;         % erase excess
        fi;
        hpen;   w0 draw 3 . . 4;                                        % serif
fi;
cpcn;   bot3y5 = m;   lft 3x5 == round u;
w3 draw 5;                                                              % bulb
hpen; lft3x5 == lft0x6; y5 == y6;
rt2x20 = rt0x9 == round(r -- u);   lft2x20 == lft0x8;   x7 == .5[x6, x9];   x11 == x2;
x10 = x7;   top0y7 = h + oo;   y8 == y9 = .7h;   bot0y1 I == 0;   y10 = .4[y1 I, y9];
w0 draw 6{0, 1} . . 7{1, 0};                                            % shoulder
lpen#;   2u draw 9{0, --I} . 10{x10 — x9, .5(y10 -- y9)}
        . 11{x11 --- x10, 2(y1 I — y10)};                                % erase excess
hpcn;   w0 ddraw 7{ I ,  0 } 8{0, - - I } 10{x10 - x9, .5(y10 -- y9)},
        7{1, 0} . 9{0, —1} . . 10{x10 — x9, .5(y10 — y9)};               % stroke
draw 10{x10 — x9, .5(y10 .   . y9)} 11{x11 — x10, 2(y11 — y10)}.      % link
```



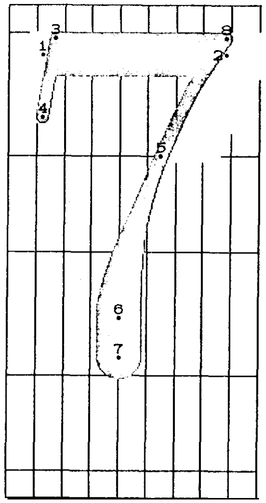


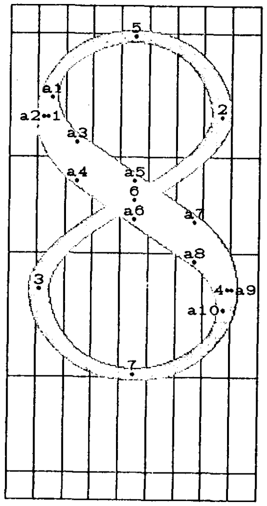
```
“The numeral 3”;
call charbegin(`3, 9, 0, 0, ph, 0, ph·slant -- .5pu);
new $w_{99}$;   $w_{99}$ = round .75[$w_0$, $w_3$];
cpen;   bot$_{99}y_1$ = .75h;
i f top$_{99}y_1$ > .9h: new $y_1$; top$_{99}y_1$ = .9h;
fi;
lft$_{99}x_1$ = round 1.5u; $w_{99}$ draw 1;                      % upper bulb
top$_3y_2$ = .25h;
i f bot$_3y_2$ < .1h: new $y_2$;   bot $_3y_2$ = .1h;
fi;
lft$_3x_2$ = round u;   $w_3$ draw 2;                             % lower bulb
hpen; lft$_0x_3$ = lft$_{99}x_1$; lft$_0x_4$ = lft $_3x_2$; $y_3$ = $y_1$; $y_4$ = $y_2$;
$x_5$ = .5[$x_3$, $x_7$];   $x_6$ = .5[$x_4$, $x_8$];   top$_0y_5$ = h + oo;   bot$_0y_6$ = −oo;
rt$_1x_7$ = round(r − 1.5u);   rt$_2x_8$ = round(r − u);   $y_7$ = .5[$y_9$, $y_5$];   $y_8$ = .5[$y_{10}$, $y_6$];
$y_9$ = $y_{10}$ = good, .52h;   $x_{10}$ + u = $x_9$ = .5r;
$w_0$ draw 3{0,1} . . 5{1, O};                                    % shoulder
call ‘a arc(5, 7, $w_1$);   call `b arc(9, 7, $w_1$);             % upper howl
$w_0$ draw 9 . IO;                                                % bar
call `c arc(9, 8, $w_2$);   call `d arc(6, 8, $w_2$);             % lower bowl
$w_0$ draw 6{−1, 0} . 4{0, 1}.                                    % tail

“The numeral 4”;
call charbegin(`4, 9, 0, 0, ph, 0, ph·slant — .5pu);
hpen;   $x_1$ = $x_2$;
if lcs ≠ 0:   if $w_2$ > 1.5u:   rt$_2x_1$ = round(r — 1.75u);
    else:   $x_1$ = good$_2$ 6.5u;
    fi;
else:   $x_1$ = good$_2$ 6.5u;
fi;
top$_2y_1$ = h;   bot$_2y_2$ = 0;
rt$_2x_1$ = rt$_0x_3$;   $y_1$ = $y_3$;   $y_4$ = $y_5$ = good, $\frac{5}{6}$e;   lft$_0x_4$ = round u;
$w_2$ draw 1 . 2;                                                 % stem
lpen#; $w_2$ draw 3 . . 4 ;                                       % erase excess at upper left
hpen; $w_0$ draw 3 . . 4;                                         % diagonal
if Jcs ≠ 0:   rt$_0x_5$ = rt$_2x_2$ + 1.5u + eps;
    call ‘a serif(2, 2, 1, --1.5);
    call `b serif(2, 2, 1, 1.5);                                  % serif
else:   rt$_0x_5$ = round(r - - .75u);
fi;
$w_0$ draw 4 . . 5.                                               % bar
```



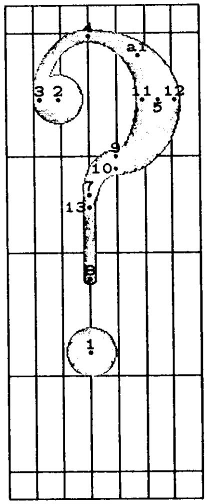


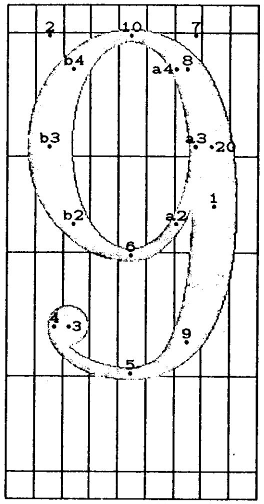
```
"The numeral 5";
call charbegin(`5, 9, 0, 0, ph, 0, ph·slant — .5pu);
vpen;   x_1 = good,, 2u;   top_8 y_1 = h;   rt_6 x_3 = round(r — 1.25u);   top_6 y_3 = h;
x_2 = .5[x_1, x_3];   new w_99;   w_99 = round .75[w_6, w_8];   top_99 y_2 = round .95h;
x_0 = —.5u;   x_4 = r + 1.5u;   y_0 = y_4 = 1.5h;
draw (|w_8|0 . .)I.. |w_99|2 . . |w_6|3( . . 4);                          % arm
hpen;   x_5 = x_6 = x_1;   top_0 y_5 = h;   top_0 y_6 = .75[e, m];
x_7 = .5r;   top_0 y_7 = m + oo;   x_9 = x_7 — .5u;   bot_0 y_9 = —oo;
rt_2 x_8 = round(r — u);   y_8 = .5[y_7, y_9];
w_0 draw 5 . . 6;                                                       % stem
draw (9 . )6 . . 7{1, 0};                                   % left part of bowl
call `a arc(7, 8, w_2);   call `b arc(9, 8, w_2);          % right part of bowl
lft_0 x_10 = lft_3 x_11 = round u;   y_10 = y_11 = 1/3 y_6;
w_0 draw 9{—1, 0} . 10{0, 1};                                           % tail
cpen;   w_3 draw 11.                                                    % bulb

"The numeral 6";
call charbegin(`6, 9, 0, 0, ph, 0, ph·slant — .5pu);
if w_2 > 1.5u:   lft_2 x_1 = round .75u;   rt_2 x_2 = round(r — .75u);
else:   x_1 = good_2 1.5u;   x_2 = good_2(r — 1.5~);
fi;
new w_99;   w_99 = round .75[w_0, w_3];
cpen;   top_3 y_3 = h — .25e;
if y_3 < .5[m, h]:   new y_3;   y_3 = .5[m, h];
f-i;
rt_99 x_3 = rt_0 x_4 = round(r — 1.5u);   y_4 = y_3;
w_99 draw 3;                                                            % bulb
hpcn;   x_20 = good_2(x_1 + .1u);   x_5 = x_6 = x_10 = .5[x_20, x_2];   top_0 y_5 = h + 00;
w_0 draw 4{0, 1} . . 5{ -1, O};                                     % shoulder
bot_0 y_2 = —oo;   top_0 y_6 = m + 0 0 ;   y_20 = .5[y_2, y_6];
Y 7 = y_2 = y_10;   rt_0 x_7 = rt_2 x_20;
call 'a darc(6, 7, w_0);   call `b darc(6, 2, w_2);                      % bowl
new w_99;   w_99 = 2/3 [w_0, w_2];         % intermediate width used in darc routine
x_8 = x_9;   rt_99 x_8 = rt_0(1/sqrttwo[x_6, x_7]);
y_8 = 1/sqrttwo[y_20, y_7];   y_5 — y_9 = y_8 — y_7;   y_1 = .5[y_5, y_10];
draw |w_0#|5{—1, 0} . . |w_99|9{x_7 — x_6, y_7 — y_20} . |w_2#|1{0, —1}
      . |w_99|8{x_6 - - x_7, y_7 — y_20} |w_0#| 10{ 1, 0}.                 % stroke
```



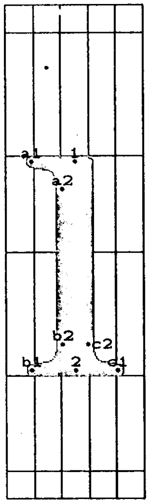


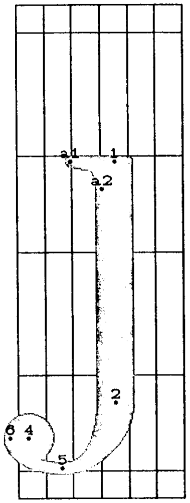
```
The numeral 7";
call $charbegin$(`7, 9, 0, 0, $ph$, 0, $ph \cdot slant$ — .5$pu$);
vpen;    **new** $w_{99}$;
if $lcs$ = 0:    $w_{99} = w_4$;
else:    $w_{99} = w_8$;
**fi**;
**lft**$_{99}x_1$ = round u;    **rt**$_{99}x_2$ = **round**($r - u$);    **top**$_{99}y_1$ = ***h*** ;    $y_2 = y_1$;
$w_{99}$ draw 1 . . 2;                                                      % bar
if $lcs \neq 0$:    $x_4 = x_1$;    bot$_6 y_4$ = bot$_{99} y_1 - ucs \cdot aspect \cdot u$ -- $eps$;    top$_6 y_3 = h$;
    if $w_7 = w_8$:    $x_3 = x_4$;
    else:    $x_3 - .5u = x_4$;    lpen#;    $w_0$ + .5$u$ draw 3 . 4;          % erase excess
    **fi**;
    hpen;    $w_0$ draw 3. 4;                                               % serif
**fi**;
cpen;    bot$_2 y_7 = -oo$;    bot$_2 y_6 = \frac{1}{6}m$;    $x_6 = x_7$ = **good**$_2$ 4$u$;
$w_2$ draw 7 . . 6;                                                         % bulb
**hpen**;    x j = 5.5$u$;    $y_5$ = $m$;    top$_0 y_8$ = ***h***;    $x_8 = x_2$;
**rpen#**;    2$u$ **draw** 8{$x_5 - x_8$, $\frac{5}{7}$($y_5$ ·  · $y_8$)} . . 5( . 6);        % erase excess
**hpen**;    **draw** |$w_0$|8{$x_5 - x_8$, $\frac{5}{7}$($y_5$ - - $y_8$)} . |$w_0$#|5 . . |$w_2$#|6{0, —1}.   % stroke

"The numeral 8";
call $charbegin$(`8, **9**, 0, 0, $ph$, ***0***, $ph \cdot slant$ — .5$pu$);
new $w_{98}, w_{99}$, ***ss***;    $w_{99}$ = round $\frac{1}{3}[w_0, w_1]$;
hpen;    lft$_{99} x_1$ = round $u$;    $x_2 = r - x_1$;    $y_1 = y_2$;
lft$_{99} x_3$ = round .75$u$;    $x_4 = r - x_3$;    yj = $y_4$;    $x_5 = r - x_5 = x_6 = x_7$;
top$_0 y_5$ = ***h*** + $oo$;    $y_6$ = .52$h$;    bot$_0 y_7 = -oo$;
$w_{98} = 2[w_7, w_8]$;
if $w_6 = w_{98}$:    $ss$ = 0;
else:  $ss$ = $h/(18u)$;                                                   % slope
**fi**;
call `a $sdra$ w(5, 1, 6, 4, 7, $w_{99}, w_{98}, -ss$);                % upper left and lower right strokes
$w_{99}$ draw 5{1, 0} . . 2{0, —1} . . 6{—1, —.75$ss$} . .
    3{0, **--*I*}.** . 7{1, 0}.                                          % upper right and lower left strokes
```

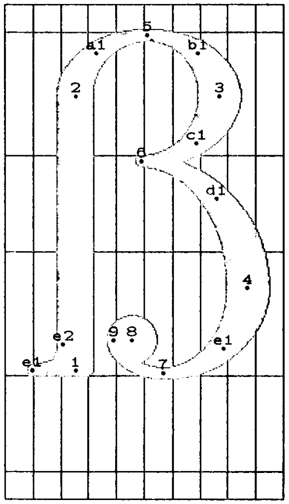




```
"The numeral 9";
call charbegin(`9, 9, 0, 0, ph, 0, ph·slant — .5pu);
if w2 > 1.5u:   rt2x1 = round(r — .75u);   lft2x2 == round .75u;
else:   x1 = good2(r — 1.511);   x2 = good2 1.5u;
fi;
new w99;   w99 = round .75[w0, w3];
cpen;   bot99y3 = .25e;
if y3 > .5e:   new y3;   y3 = .5e;
fi;
lft99x3 == lft0x4 = round 1.5u;   y4 = y3;
w99 draw 3;                                                        % bulb
hpcn;   x20 == good2(x1 - - .1u);   x5 =: x6 = x10 == .5[x20, x2];   bot0y5 = —oo;
w0 draw 4{0, —1} .. 5{1, 0};                                       % tail
top0y2 = h + oo;   y6 = e — oo;   y20 == .5[y2, y6];
y7 == y10 = y2;   lft0x7 = lft2x20;
call 'a darc(6, 7, w0);   call `b darc(6, 2, w2);                 % bowl
new w99;   w99 = 2/3[w0, w2];                   % intermediate width in darc routine
x8 == x9;   lft99x8 == lft0(1/sqrttwo[x6, x7]);
y8 == 1/sqrttwo[y20, y7];   y5 -- y9 == y8 — y7;   y1 == .5[y5, y10];
draw |w0#|5{1, 0} .. |w99|9{x7 — x6, y7 — y20} .. |w2#|1{0, 1}
      . |w99|8{x6 — x7, y7 . . y20} . |w0#| 10{—1, 0}.             % stroke

"Question mark";
call charbegin( '077, 7, 0, 0, ph, 0, .8ph·slant + .5pwii — pu);
new w99, w98;   w98 = round .4[w0, w2];
i f w3 < w0 sqrt 2:   w99 == w0 sqrt 2;
else:   w99 == w3;
fi;
cpen;   bot99y1 = 0;   x1 = good0 .5(r — u);
w99 draw 1;                                                        % dot
lft3x2 = lft0x3 = round u;   y2 == y3 = .8[top99y1, bot6y4];   top6y4 = h + oo;
w3 draw 2;                                                         % bulb
hpen;   x4 == .5(r — u);   x5 == good2(r — 1.5u);   y5 == y2;
w0 draw 3{0, 1} .. 4{1, 0};   call `a arc(4, 5, w2);              % bowl
lft0x11 = lft2x5;   rt0x12 = rt2x5;   y11 = y12 == y5;
cpen;   top6y9 == top98y6;   bot6y10 == bot98y6;
x9 == x10 == x7 + u;   x7 == x8 = x13 = x1;   y13 — y7 == y10 . . y9;
y4 - - y2 == y2 — y6;   top6y7 == top98 1/3[y6, y8];   bot6y8 == .25[top99y1, m] + 1;
hpen;   w0 ddrnw 1 1{0, —1} .. 9{—1, 0} .. 7{0, —1},
      12{0, —1} . 10{—1, 0} . . 13{0, —1};                        % link and stem
draw 13.. 8.                                                       % point
```

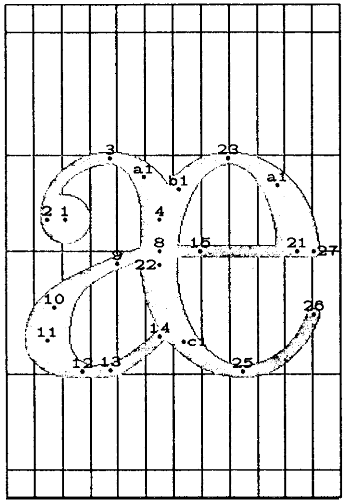




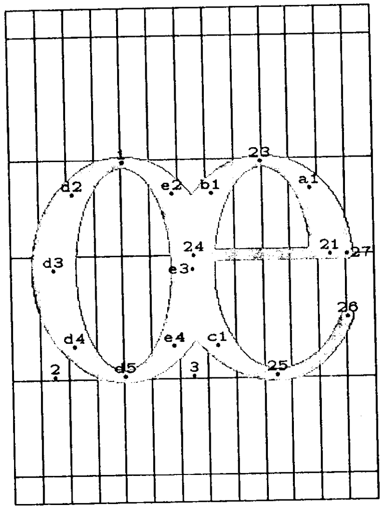


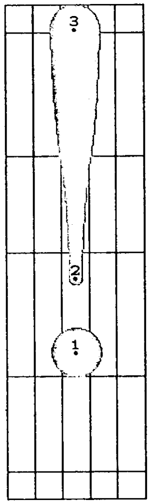
The file romans. mf

```
% This tile contains special letters and letter combinations,
% compatible with the alphabet"romanl".
% Codes '013, '014, '033-'035 are used.

"Dotlcss letter i";
call charbegin( '013, 5, sc, sc, px, 0, px·slant+.5pwi+ (sc — 2)pu);
hpen;   x_1 = x_2 = good, .5r;   top_1y_1 = m;   bot_1y_2 = 0;
w_1 draw 1.. 2;                                                    % stem
if lcs ≠ 0:   call 'a serif(1, 1, 2, —lcs);                        % upper serif
      call ` b serif(2,1,1,—lcs);
      call ` c serif(2, 1, 1, lcs);                                % lower serif
fi.

"Dotless letter j";
call charbegin( '0 14, 6, sc, 0, px, pd,px·slant+.5pwi -- 2pu);
hpen;   x_1 = x_2 == good_1(r — 2.5~);
if fixwidth = 0:   lft_3x_4 == round(—.5u);
else:   lft_3x_4 == round u;
fi;
cpen;   bot_3y_4 = —.9d; w_3 draw 4;                                 % bulb
hpen; top_1y_1 = m; bot_1y_2 = —1/3d;
bot_0y_5 = - d — oo;   y_4 = y_6;   lft_0x_6 = lft_3x_4;   xj = .5[x_2, x_6];
draw |w_1|1..|w_1#|2{0, --1).|w_0#|5{—1,0}..6{0,1};            % stem and tail
if lcs ≠ 0:   call 'a serif(l, I, 2, —lcs);                        % serif
fi.

"The German letter ss";
call charbegin( '033, 10, sc, 0, ph, 0, ph·slant—pu);
hpen;   x_1 = x_2 == good, 2.5u;   rt_2x_3 = round(r — 1.5u);   rt_2x_4 == round(r — .5u)
bot_1y_1 = 0;   y_2 == .5[m, y_5];
top_0y_5 = h + 00;   x_5 = .5[x_2, x_3];   y_3 == .5[m, y_5];   y_4 = .4m;
w_1 draw 1.. 2;                                                    % stem
call 'a arc(5, 2, w_1);                                            % shoulder
top_0y_6 == m;   lft_0x_6 == round(rt_1x_2 + 1.521);
call ` b arc(5, 3, w_2);   call ` c arc(6, 3, w_2);                % upper bowl
cpen;   lft_3x_8 == round(rt_1x_1 + .5u);   bot_3y_8 = 1e;
w_3 draw 8;                                                        % bulb
hpen; lft_0x_9 == lft_3x_8; y_9 == y_8;   x_7 == l[lft_0x_9, x_1]; bot_0y_7 == —oo;
call `d arc(6, 4, w_2);   call `e arc(7, 4, w_2);                  % lower bowl
w_0 draw 7{—1, 0). . 9{0,1};                                       % link
if lcs ≠ 0:   call `e serif(1,1,2,—lcs);                           % serif
fi.
```



```
The ligature ae";
call charbegin( '034, 12, 0, 0, px, 0, pe·slant + Jcic);
cpcn; lft_3x_1 == round 1.25u;
if top_3(top_3top_6e + 2) > .9[e, m]:    top_3y_1 = .9[e, m];
else:   y_1 = top_3top_6e + 2;
fi;
w_3 draw 1;                                                        % bulb
hpen; lft_0x_2 = lft_3x_1;   y_2 = y_1;   x_3 = 3.75u;   top_0y_3 = m + 00;
x_4 = good, 5.521;   y_4 = 1/3[e, m];
w_0 draw 2{0, 1}..3{ 1, O};   call 'a arc(3, 4, w_1);               % shoulder
x_8 = x_4;   y_8 = e;   w_1 draw 4.. 8;                             % stem
x_9 = 4u;   y_9 == .9[w_6, y_8];   x_10 = x_11 + .25u;   y_10 == .5[w_6, y_8];
x_11 = good_2 1.5u;   y_11 = .2[w_6, y_8];
x_12 = 2.75u;   bot_0y_12 = —oo;   x_13 = 3.75u;   y_13 = .015[y_12, y_8];
x_14 == x_8;   y_11 = .3[y_12, y_8];   x_15 = 7u;   y_15 = e;
draw |w_0|8{—1, 0}.|w_0#|9..|.8[w_0, w_2]|10 |w_2#|11{0, —1}..
    |.7[w_0, w_2]|12{1, 0}.|w_0#|13..14(.. 15);                   %left bowl
if w_2 > 1.5u:   rt_2x_21 = round(r — .75u);
clsc:   x_21 = good_2(r — 1.5u);
fi;
x_22 = x_4;   x_23 = r — 4u;   top_0y_23 = m + oo;   y_21 == e;   y_22 = .5[y_25, y_23];
bot_0y_25 = -00;   x_25 == x_23 + .5u;
call `a arc(23, 21, w_2);                                          % right bowl
call `b arc(23, 22, w_1);   call `c arc(25, 22, w_1);               % stroke
w_0 draw 8.. 21;                                                    % bar
if w_0 = w_1:   x_26 == x_21;   x_27 — x_25 = x_21 — x_23;   y_27 == y_22;
    new aa;   x_26 == aa[x_25, x_27];   y_26 == (sqrt(1 —aa·aa))[y_27, y_25];
else:   rt_0x_26 == rt_2x_21;   x_27 = x_26;   y_26 == .5e — 1;   y_27 == e;
fi;
w_0 draw 25{ 1, 0}..26(..27).                                      % point
```

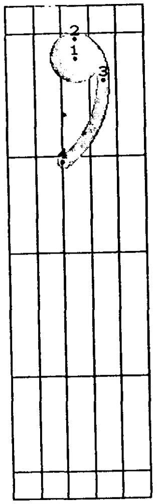


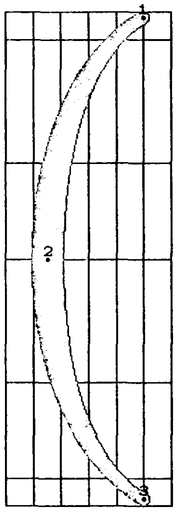


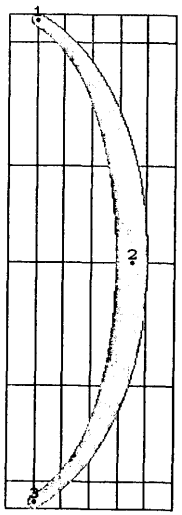
```
"The ligature oe";
call charbegin( '035, 13, 0, 0, px, 0, pe·slant + lcic);
hpcn;   x3 = good2 .5r;
if w2 > 1.5u:   lft2x2 = round .75u;   rt2x21 = round(r — .75u);
else:   x2 = good2 1.5u;   x21 = good2(r — 1.5u);
fi;
x1 — x2 = x3 — x1;
top0y1 = m + 00;   bot0y2 = -00;   y2 = y3;
call `d darc(1, 2, w2);                                    % left part of left bowl
call `e darc(1, 3, w2);                                   % right part of right bowl
x22 = x3;   x23 = r — 4u;   top0y23 = m + oo;   y21 = e;   y22 = .5[y25, y23];
bot0y25 = -00;   x25 = x23 + .5u;
call `a arc(23, 21, w2);                                            % right bowl
call `b arc(23, 22, w2);   call `c arc(25, 22, w2);                   % stroke
new aa;   y21 = y2 I = aa[y22, y23];   x21 — 1 = (sqrt(1 — aa·aa))[x23, x22];
w0 draw 24.. 21;                                                         % bar
if w0 = w1:   x26 = x21;   x27 -- x25 = x21 -- x23;   y27 = y22;
      new aa;   x26 = aa[x25, x27];   y26 = (sqrt(1 — aa·aa))[y27, y25];
else:   rt0x26 = rt2x21;   x27 = x26;   y26 = .5e — 1;   y27 = e;
fi;
w0 draw 25{1, 0} . 26( . 2 7 ) .                                      % point
```

The file romitp m f

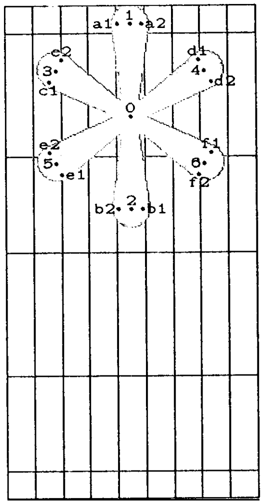
```
% This file contains punctuation marks common to roman and
% italic styles, including math italic and fixed-width fonts.
% Codes '041, '0 17-'054, '056, '057, '072 -'076, '133, '135, '140 are used

"Exclamation point";
call charbegin('041, 5, 0, 0, ph + ph, 0, (ph + pb)·slant + .5pwiii — 2pu);
new w99;
if w3 < w0 sqrt 2:   w99 = round w0 sqrt 2;
else:   w99 = w3;
fi;
cpen;   x1 = x2 = x3 = good,,,, 2.5u;   bot99y1 = 0 ; w99 draw 1 ;        % dot
top99y3 = h + b;   bot6y2 = .25[top99y1 + 1, m];
w99 draw 3;                                                     % top of stem
hpen;   d r a w |w99|3 |w0|2.                                           % stem
```

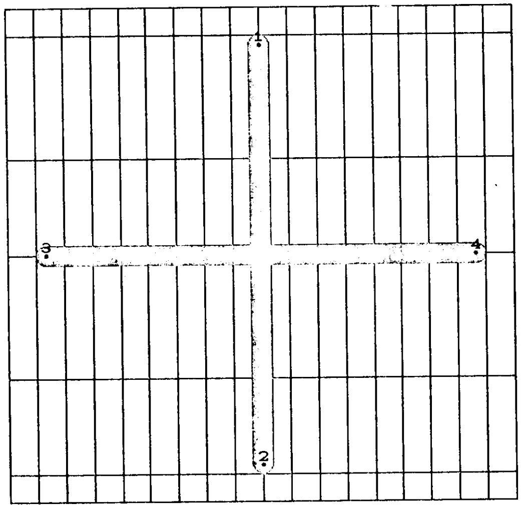


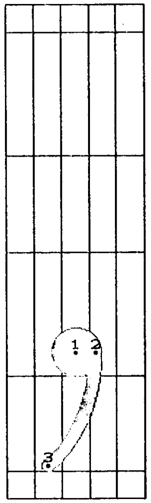




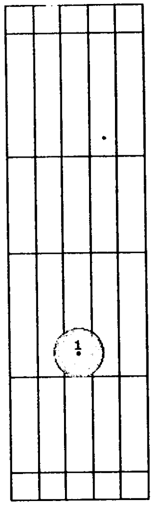
```
"Apostrophe";
call charbegin('047, 5, 0, 0, ph, 0, 2/3[px, ph]·slant + .5pw — pu);
% There is rotational symmetry with respect to reverse apostrophe.
new v;   v = fixwidth[u, 5/9u];            % unit width adjusted to agree with closing quotes
cpen;  top3y1 = top6y2 = h;  top6y4 = m;  y3 = 2/3[y4, y2];
x1 = x2 = good, .5r;  x3 = good0(x1 + v + eps);  x4 = x1 — .5v — eps;
w3 draw 1;                                                                     % bulb
hpen;  w0 draw 2{1, 0} .. 3{0, —1} . 4{3(x4 — x3), y4 — y3}.                   % tail

"Left parenthesis";
call charbegin('050, 6, 0, 0, ph + pb, ph + pb — 2pa, (ph + pb)·slant + .5pw — .5pu);
% There is left-right symmetry with respect to right parenthcsis.
hpen;  x1 = x3 = good0(r — u);
top0y1 = h + b;  y2 = a = .5[y1, y3];  y0 = y1;  y4 = y3;
new v;                                          % unit width corrected for fixed width option
if fixwidth = 0:   v = u;
else:   v = 6/9u;
fi;
new w99;  w99 = round .75[w0, w1];  lft99x2 = round(x1 — 4v);
x0 = x4 = x1 + 7.5v;
draw (0 ..)|w0|1 . . |w99#|2 . . |w0|3( .. 4).                                 % stroke

"Right parenthcsis";
call charbegin('051, 6, 0, 0, ph + pb, ph + pb — 2pa, pa·slant — .5pu);
% Thcrc is left-right symmetry with respect to left parenthesis.
hpen;  x1 = x3 = good0u;
top0y1 = h + b;  y2 = a = .5[y1, y3];  y0 = y1;  y1 = y3;
new v;                                          % unit width corrected for fixed width option
if fixwidth = 0:   v = u;
else:   v = 6/9u;
fi;
new w99;  w99 = round .75[w0, w1];  rt99x2 = round(x1 + 4v);
x0 = x4 = x1 — 7.5v;
draw (0. .)|w0|1 . . |w99#|2 . . |w0|3( .. 4).                                 % stroke
```

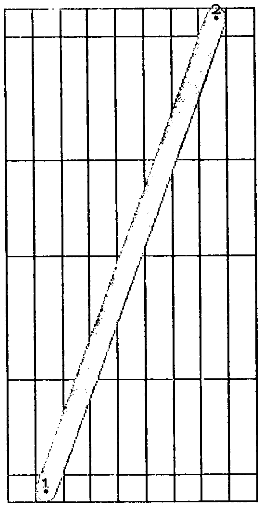


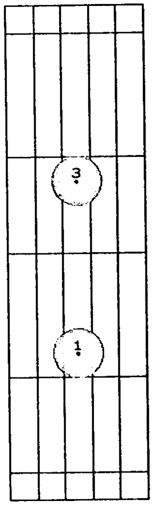


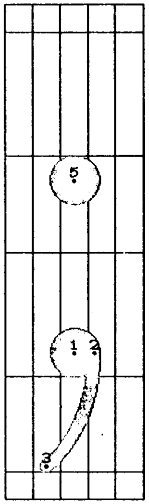


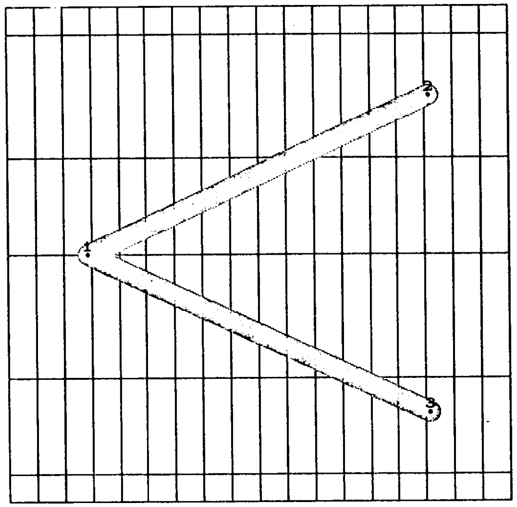
```
Asterisk";
call charbegin( '052, 9, 0, 0, lowast[ph + pb, pa + .5px], 0,
                        (lowast[ph + pb, pa + .5px] — .25px)·slant — (4 — 1.875 sqrt 3)pu);
cpen;   top_1 y_1 = lowast[h + b, round(a + .5m)];   top_1 y_1 — bot_1 y_2 = m;
y_0 = .5[y_1, y_2];   x_0 = r — x_0;   x_1 = x_2 = x_0;
x_3 = r — x_4;   x_5 = r — x_6;   x_3 = x_5;                               % left-right symmetry
y_3 = y_4;   y_5 = y_6;   y_3 — y_5 = y_1 — y_0;   .5[y_3, y_5] = y_0;
x_1 — x_0 = (.5 sqrt 3)lft_1 3.75u;          % asterisk will have 60-degree angles if m = 7.5u
call 'a cdraw(1, 0, 1, 0);                                                      % upper arm
call `b cdraw(2, 0, 1, 0);                                                      % lower arm
call `c cdraw(3, 0, 1, 0);                                                 % upper left arm
call `d cdraw(4, 0 , 1, 0 ) ;                                             % upper right arm
call ` e cdraw (5, 0, 1, 0);                                               % lower left arm
call ` f cdraw(6, 0,1,0).                                                 % lower right arm

"Plus sign";
cpen;
if fixwidth = 0:   if pa + 8pu > ph:
            call charbegin('053, 18, 0, 0, ph, ph — 2pa, pa·slant -- .5pu);   top_10 y_1 = h;
      else:   call charbegin( '053, 18, 0, 0, pa + 8pu, 8pu — pa, pa·slant — .5pu);
            top_10 y_1 = a + 8u;
      fi;
else:   call charbegin('053, 9, 0, 0, 3.5pu + pa, 3.5pu -- pa, 0);   top_10 y_1 = a + 3.5u;
fi;
.5[y_1, y_2] = a;   x_1 = x_2 = .5r;
lft_10 x_3 = round u;   x , = r — x_3;   y_3 = y_4 = a ;
w_10 draw 1.. 2;                                                                    % stem
draw 3 . . 4.                                                                        % bar

"Comma";
call charbegin( '054, 5, 0, 0, I .5pwiii, pdd, 0);
cpen;   new w_99;
if w_3 < w_0 sqrt 2:   w_99 = round w_0 sqrt 2;
else:   w_99 = w_3;
fi;
x_1 = goody,5r;   bot_99 y_1 = 0; w_99 draw 1;                                      % bulb
y_2 = y_1;   rt_99 x_1 = rt_0 x_2;
if fixwidth = 0:   x_3 = good,, 1.5u;
else:   x_3 = good_0 35/18 u;
fi;
hpen;   bot_0 y_3 = — dd;
w_0 draw 2{0, —1} . . 3{3(x_3 — x_2), y_3 -- y_2}.                                % point
```



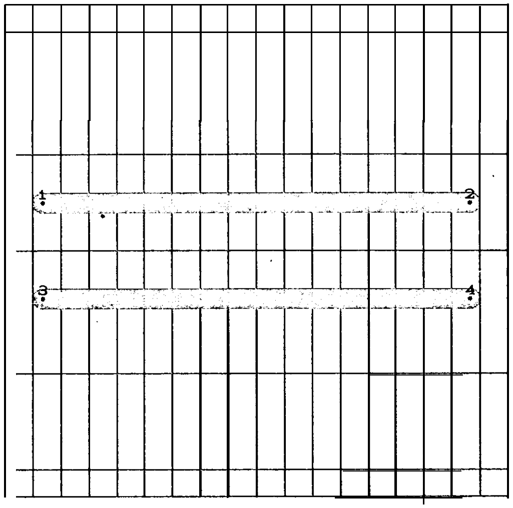


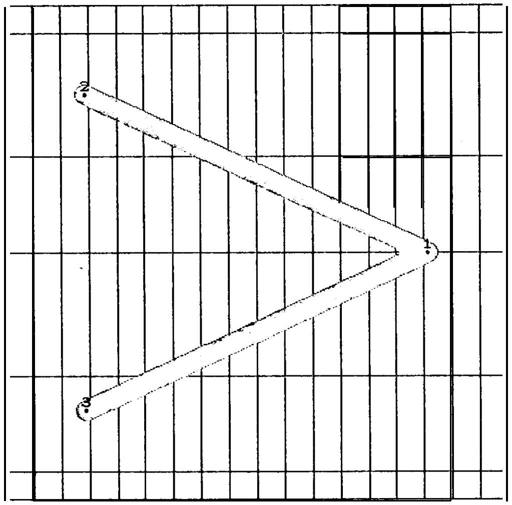


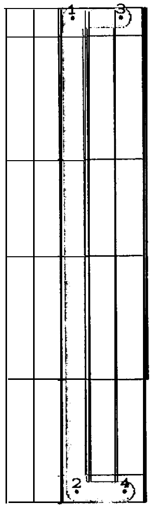


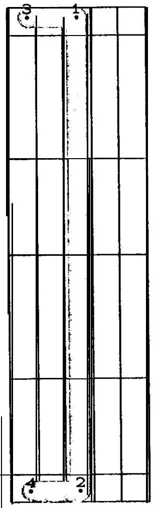


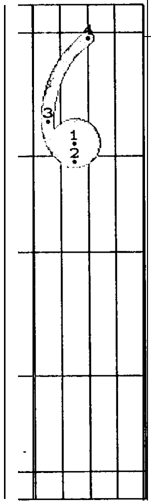
```
"Period";
call charbcgin( ´056, 5, 0, 0, 1.5pwiii, 0, 0);
cpcn;   new w99;
if w3 < w0 sqrt 2:   w99 == round w0 sqrt 2;
else:   w99 == w3;
fi;
x1 = good,, .5r; bot99y1 = 0; w99 draw 1.                                % dot

"Virgule (slash)";
call charbcgin(´057, 9, 0, 0, ph + pb, ph + pb − 2pa, (ph + pb)·slant − .5pu);
cpen;   lft10x1 = round u;   x2 = r − x1;
top10y2 = h + b;   .5[y1, y2] = a;
w10 draw 1 . . 2.                                                        % diagonal

"colon" ;
call charbcgin( ‘072, 5, 0, 0, px, 0, px·slant + .5pwiii sqrt 2 − 2pu);
cpen;   new w99;
if w3 < w0 sqrt 2:   w99 = round w0 sqrt 2;
else:   w99 = w3;
fi;
x1 == good99 .5r;   bot99y1 = 0; w99 draw 1;                             % upper dot
x3 == x1;   top99y3 = m; w99 draw 3.                                     % lower dot

"Semicolon";
call charbegin( ‘073, 5, 0, 0, px, pdd, px·slant + .5pwiii sqrt 2 − 2pu);
cpen;   new w99;
if w3 < w0 sqrt 2:   w99 == round w0 sqrt 2;
else:   w99 = w3;
fi;
x1 = good99 .5r;   bot99y1 == 0;   w99 draw 1;                           % bulb
x5 = x1;   top99y5 = m; w99 draw 5;                                      % dot
y2 = y1;   rt99x1 == rt0x2;
if fixwidth == 0:   x3 == good,, 1.5u;
else:   x3 = good0 35/18 u;
fi;
hpen; bot0y3 == −dd;
w0 draw 2{0, −1} . 3{3(x3 --- x2), y3 − y2}.                             % point

"Less than sign";
call charbegin( ‘074, 18, 0, 0, .5[px, ph] + prt/2,
                    .5[px, ph] + prt/2 − 2pa, .5[px, ph]·slant -- 2pu);
cpen;   lft10x1 = round 2.5u;   x2 == x3 == r -- x1;
y2 = good 10 .5[m, h];   .5[y2, y3] = y1 = good10a;
w10 draw 2.. 1.. 1.. 3.                                                  % diagonals
```



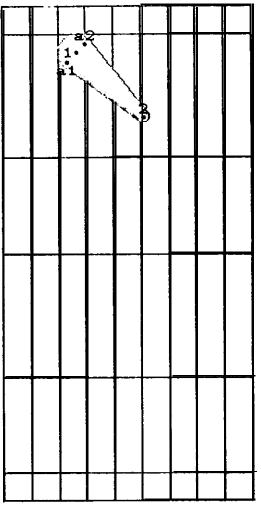


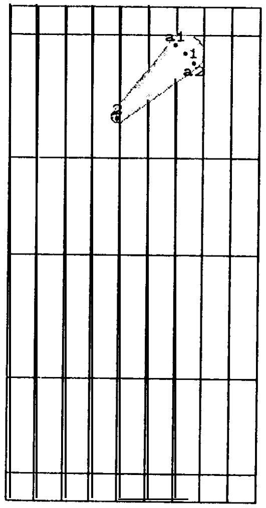


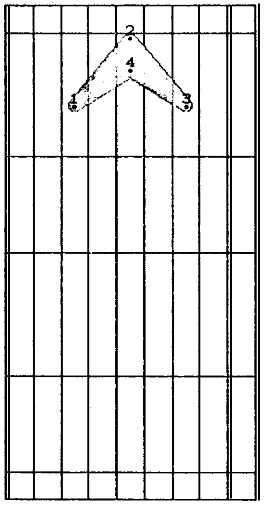


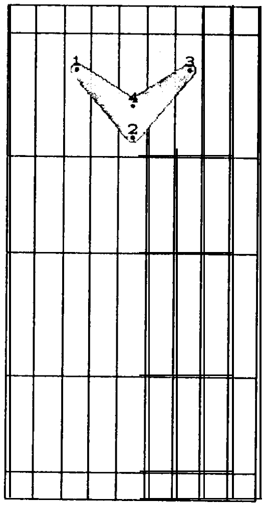


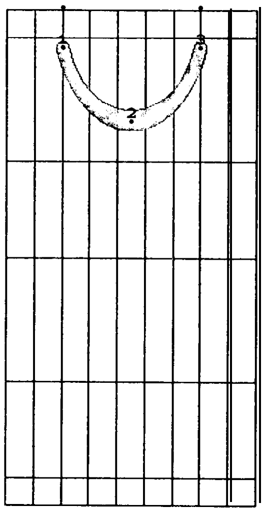
```
"Equal sign";
call charbegin('075, 18, 0, 0, pa + .5(px — pe) + prt/2, 0, (pa + .5(px -- pe))·slant — .5pu);
cpen;   lft10x1 = round u;   x3 = x1;   x2 = x.1 = r — x1;
y1 = y2;   y3 = y4;   y1 — y3 = round(m — e);   .5[y1, y3] = a;
w10 draw 1.. 2;                                                        % upper bar
draw 3. . 4.                                                           % lower bar

"Greater than sign";
call charbegin( '076, 18, 0, 0, .5[px, ph] + prt/2,
                  .5[px, ph] + prt/2 -- 2pa, pa·slant — 2pu);
cpen;   lft10x2 = round 2.5u;   x2 = x3 = r — x1;
y2 = good 10 .5[m, h];   .5[y2, y3] = y1 = good10a;
w10 draw 2.. 1 .. 1.. 3.                                               % diagonals

"Left bracket";
call charbegin(' 133, 5, 0, 0, ph + pb, ph + pb — 2pa,
                  (ph + pb)·slant + prt/2 — .25pu);
cpcn;   x1 = x2 = good 10 .5r;   x3 = x4 = x1 + 1.75u + eps
top10y1 = h + b;   .5[y1, y2] = a;   yj = y1;   y4 = y2;
w10 draw 3.. I.. 1 .. 2.. 2.. 4.                                       % stem and points

"Right bracket";
call charbegin('135, 5, 0, 0, ph + pb, ph + pb — 2pa,
                  (ph + pb)·slant + prt/2 — 2pu);
cpen;   x1 = x2 = good10 .5r;   x3 = x4 = x1 — 1.75~ -- eps;
top10y1 = h + b;   .5[y1, y2] = a;   y3 = y1;   y4 = y2;
w10 draw 3.. 1.. l .. 2.. 2.. 4.                                       % stern and points

"Reverse apostrophe";
call charbegin('140, 5, 0, 0, ph, 0, ph·slant + .5pw — 1.5pu);
% There is rotational symmetry with respect to apostrophe.
new v;   v = fixwidth[u, 5/9 u];            % unit width adjusted to agree with opening quotes
cpen;   bot3y1 = bot6y2;   top6y1 = h;   y3 = 2/3 [y1, y2];   y2 — y4 = m — h;
x1 = x2 = good3 .5r;   x3 = good0(x1 — v — eps);   x4 = x1 + .5v + eps;
w3 draw 1;                                                              % bulb
hpen;   w0 draw 2{—1, 0} .. 3{0, 1} .. 4{3(x1 -- x3), y4 — y3}.         % tail
```











% This file contains accents common to text roman and italic fonts.
% It also contains the ‘Æ’, ‘Œ’, and ‘%’, since these symbols are common
% to the same fonts that the accents are common to.
% Character codes ´015–´032,´036, ‘037, ‘045 are represented.
% (Actually the accents in positions ‘025, ´026, ‘031, ‘032 are
% not generated unless $ligs \neq 0$, since other symbols arc substituted
% for those accents in non-ligature fonts.)

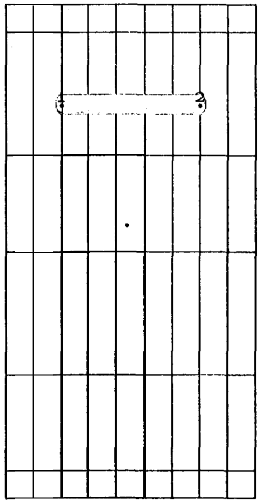
“Grave accent”;
call $charbegin$(´015, 9, 0, 0, ph, 0, $\frac{2}{3}[ph, px]\cdot slant + .5pw - pwi/6 - \frac{19}{6}pu$);
cpen; Ift $_1x_1$ = round $2u$; $x_2 = \frac{2}{3}[x_1, r$ -- x,];
top$_1y_1 = h$; $y_2 = \frac{2}{3}[h, m]$;
call ‘a $cdraw(1, 2, 1, 0)$. % diagonal

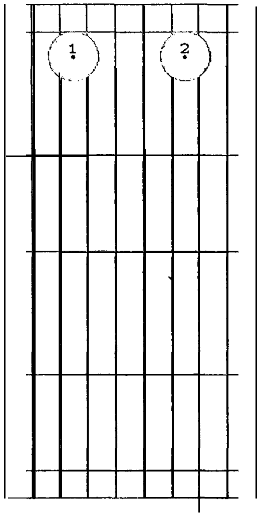
“Acute accent”;
call $charbegin$(´016, 9, 0, 0, ph, 0, $ph\cdot slant - 1.5pu$);
cpen; rt$_1x_1$ = round($r - 2u$); $x_2 = \frac{2}{3}[x_1, r - x_1]$;
top$_1y_1 = h$; $y_2 = \frac{2}{3}[h, m]$;
call ‘a $cdraw(1, 2, 1, 0)$. % diagonal

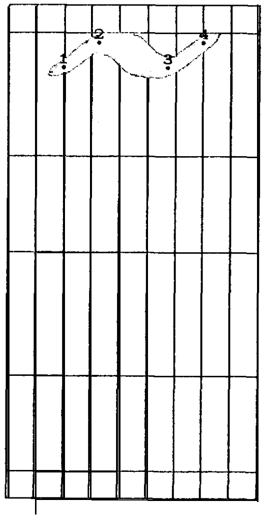
“Circumflex (hat) accent”;
call $charbegin$(´017, 9, 0, 0, ph, 0, $.5[px, ph]\cdot slant + .5pw - 2pu$);
x_1 = good,, $2.5u$; $x_2 = x_4 = r - x_2$; $x_3 = r - x_1$; % left-right symmetry
vpen; $y_1 = y_3 = .5[m, y_0]$; top$_8y_0$ = top$_6y_2 = h$; bot$_8y_0$ = bot$_6y_4$;
w_6 dtlraw 2.. 1, 4 . . 1; % left point
ddraw 2. . 3, 4 . . 3. % right point

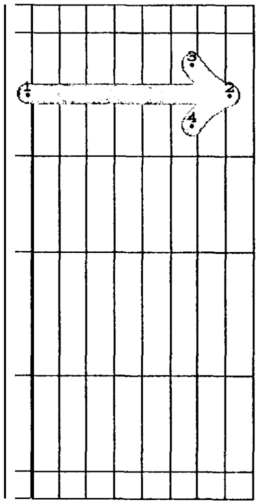
“Hachek (hooklet) accent”;
call $charbegin$(´020, 9, 0, 0,.75[px, ph], 0, $.75[px, ph]\cdot slant$ + $.5pw$ -- $2pu$);
x_1 = good$_0$ $2.5u$; x_2 = x.1 = $r - x_2$; x_3 = r $- x_1$; % left-right symmetry;
vpcn; top$_6y_1$ = round $.75[m, h]$; $y_1 = y_3$;
top$_8y_{20} = h$; $y_{21} = .5[m, y_{20}]$; $y_1 - y_0 = y_{20} - y_{21}$; % inverted circumflex
top$_8y_0$ = top$_6y_1$; bot$_8y_0$ = bot$_0y_2$;
w_6 ddraw 2 . . 1, 4 . . 1; % left point
ddraw 2. . 3, 4 . . 3. % right point

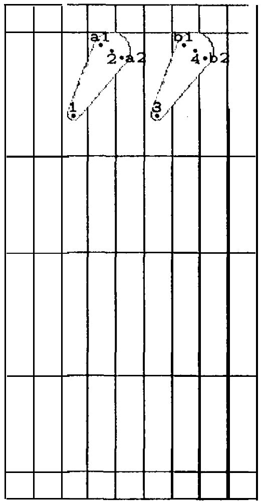
“Breve accent”;
call $charbegin$(‘0% I, 9, 0, 0, ph, 0, $ph slant$ +· .5pw 1.5pu);
$x_0 = x_1$ = good,, $2u$; $x_2 = r - x_2$; $x_3 = x_1 = r - x_1$; % left-right, symmetry
new w_{99}; w_{99} = round $.25[w_6, w_8]$;
$y_0 = y_4 = 1.25[m, h]$; $y_1 = y_3$; $y_2 = \frac{1}{3}[m, h]$;
vpcn; top$_{99}y_1 = h$;
draw $|w_{99}\#|2\{-1, 0\}$ . . $|w_6\#|1($ . . $0)$; % left point
d r a w $|w_{99}\#|2\{1, 0\}$ . . $|w_6\#|3($ . . $4)$. % right point



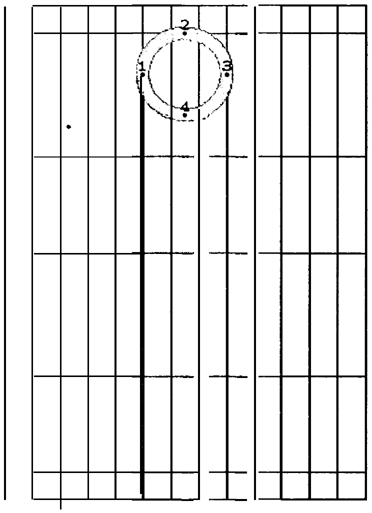


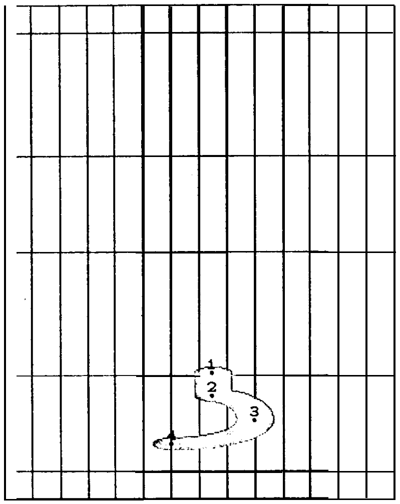


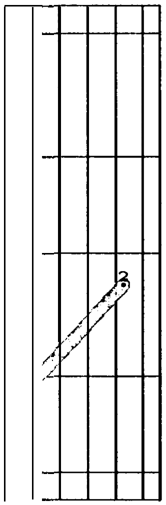


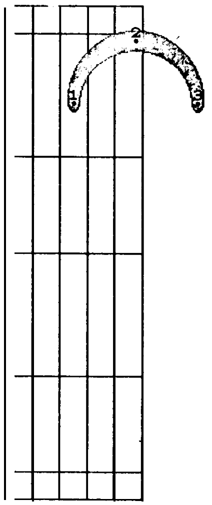


```
"Macron (bar) accent";
call charbcgin( '022, 9, 0, 0, 1/3[px, ph] + prt·aspect, 0, 1/3 [px, ph]·slant + .5pw — 1.5pu);
x1 = good0 2u;   x2 = r — x1;   y1 = y2;
vpen;  new w99;   w99 = round .25[w6, w8];   bot99y1 = 1/3[m, h];
w99 draw 1 . 2.                                                          % bar

"Umlaut (double dot) accent";
call charbcgin( '023, 9, 0, 0, ph, 0, ph·slant + .5pwiii -- 2pu);
x1 = good3 2.521;   x2 = r — x1;   y1 = y2;
cpen;   top3y1 = h;
w3 draw 1;                                                          % left dot
draw 2.                                                             % right dot

"Tilde(squiggle) accent";
callcharbegin('024, 9, 0, 0, ph, 0, ph·slant +  5pwi - 1 .5pu);
x1 = 2u;   x4 = r — 2u;   x2 = .25[x1, x4];   x3 = .75[x1, x4];
new aa, bb, rr, costh, sinth;
aa = 1/3(x4 -- x1);   bb = .2(h -- m);   rr = sqrt(aa·aa + bb·bb);
costh = aa/rr;   sinth = bb/rr;
spen(4(costh costh/w1/w1 + sinth·sinth/w0/w0),
     8costh·sinth(1/w1/w1 -- I /w0/w0),
     4(costh·costh/w0/w0 + sinth·sinth/w1/w1),
     0, 0, 0, 0);                          %oblique pen in direction {(x4 -- x1)/3, y2 — y1}
top0y1  =  .8[m, h]; top0y2  =  h; y3  =  y1; y4 = y2;
w0 draw 1.. 2{ I) 0}. 3{ I) 0). . 4.                                  % points and stroke

if ligs ≠ 0:   "Arrow (vector) accent";
     call charbegin( '025, 9, 0, 0, .75[px, ph] + prt·aspect/2, 0, .5[px, ph]slant);
     cpen;   lft10x1 = round .5u;   x2 = r — x1;   y1 = y2 = .5[m, h];
     w10  draw  1  2;                                                   % bar
     rt10x3 = round(x2 — u);   x1 = x3;
     y3 == y2 + (h — m)/4;   y1 = y2 — (h --- m)/4;
     d r a w 3{.5(x2 -- x3), y2 — y3} 2{x2 -- x3, .5(y2 -- y3)};           % upper point
     d r a w 4{ .5(x2 — x4), y2 — y4} 2{x2 -- x4, .5(y2 — y4)};            % lower point
fi.

if ligs ≠ 0:   "Long Hungarian umlaut accent";
     call charbegin('026, 9, 0, 0, ph, 0, ph·slant — pu);
     cpen;   x1 = good,, 2.5u;   rt 1x2 = round .5r;
     x3 -- x1 = x4 -- x2;   rt1x4 = round(r — I 5u);
     y1 = y3 = 1/3 [m, h];   top1y2 = h;   y2 = y4;
     call `a cdraw(2, 1, 1, 0 ) ;                                      % left diagonal
     call ` b cdraw(4, 3, 1, 0);                                       % right diagonal
fi.
```

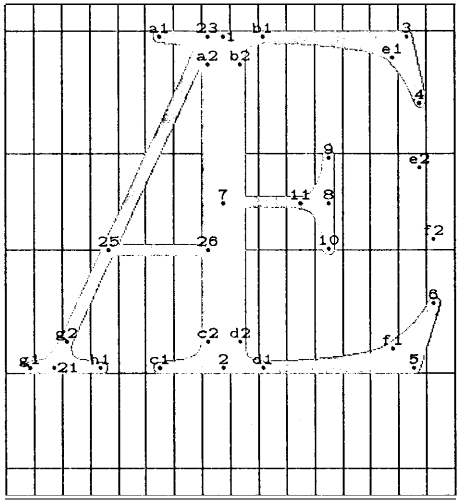




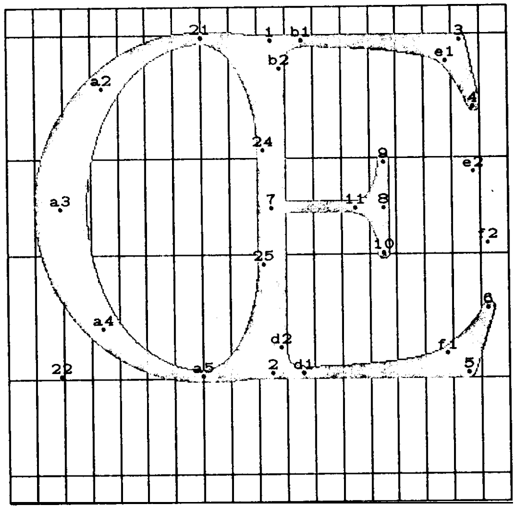




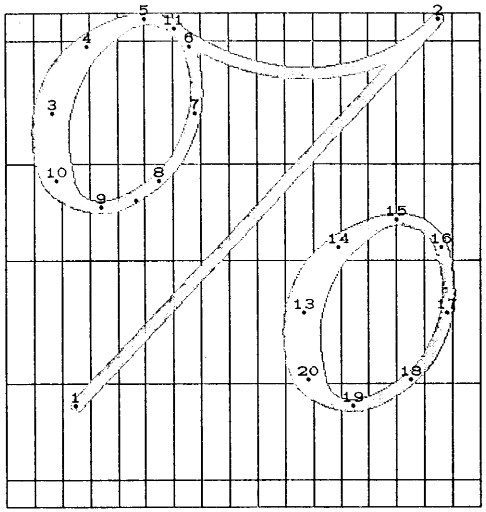
```
"Scandinavian circle accent";
% This character is designed to be used as an ordinary accent on an 'a'
% or to bc raised $\frac{2}{3}(ph - px)$ points and superimposed on an 'A'.
call charbegin( '027, 13, 0, 0, ph, 0, 0);
$x_1 = good_{,}\ 5u$;   $x_2 = x_4 = r - x_2$;   $x_3 = r - x_1$;
hpen;   $top_0 y_2 = h + o$;   $top_0 y_4 = round\ \frac{1}{3}[m,h] + o$;   $y_1 = y_3 = .5[y_2, y_4]$;
$w_0$ draw 4{—1,0} . . 1{0, I}. . 2{1,0};                          % left half of bowl
draw 4{1,0} . . 3{0,1} . . 2{—1,0}.                                % right half of bowl

"Cedilla accent";
% This character is designed to be used as an ordinary accent on a 'c'
% or to be superimposed on a 'C'.
call charbegin( '030, 14, 0, 0, 0, .75pd, 0);
$x_1 = x" = 7.5u$;   $x_3 = x_2 + 1.5u$;   $x_4 = x_2$ -- 1.5,;
new $w_{99}$;   $w_{99} = round\ .5[w_0, w_1]$;
hpen;   $bot_1 y_1 = -oo$;   $bot_1 y_2 = round(-.25d - oo)$;
$w_1$ draw 1 . . 2;                                                % stem
$bot_1 y_4 = round(-.75d - 00)$;   $y_3 = .5[y_2, y_4]$;
draw 2{1,0} . . 3{0,—1} . . 4{—1,0}.                               % hook

if $ligs \neq 0$:   "Cross for Polish l and L";
% This character is designed to bc used as an ordinary accent on an 'l'
% or to be raised $ph -$ px points and superimposed on an 'L'.
    call charbegin('031, 4 + pwiv/pu, 2sc, 2sc, px — (ph — px), 0, 0);
    lft , $.5[x_1, x_2] = 2u$;   $x_2 - x_1 = 3u$;
    $top_0 y_2 + (h - m) = m$;   $y_1 + (h - m) = e$;
    $w_0$ draw 1 . . 2;                                            % diagonal
fi.

if $ligs \neq 0$:   "Tic accent";
% This character is tuned to work best with a dotless roman i.
    call charbegin('032, 5, sc, s c, ph, 0, px·slant + .5pwi + (sc — 2)pu);
    $x_1 = good_{,}\ .5r$;   $y_1 = y_3 = \frac{4}{9}[m, h]$;   $x_2 = .5[x_1, x_3]$;   $x_3 = good_0(r + 2u)$;
    vpen;   $top_{10} y_2 = h + oo$;
    $w_{10}$ draw 1{0,1} . . 2{1,0};                               % left point
    draw 3{0, 1} . . 2{— I, 0};                                    % right point
fi.
```

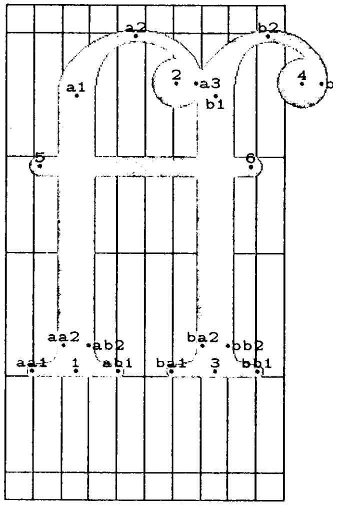


```
 Upper case ligature  AI:";
, all charbegin('036, 16, 2sc, sc, ph, 0, armic);
hpen;
lft1 x1 = round 7u;    x2 = x1;    top1 y1 = h;    bot1 y2 = 0;
w1  draw  1  2;                                                   % stem
if ucs ≠ 0:
     call ` a serif(1, 4, 2, —ucs);
     call . b serif(1, 4, 2, .5ucs);                              % upper stem serif
     call ` c serif(2, 4, 1, - - ucs);
     call ` d scrif(2, 4, 1, .5ucs);                              % lower stem serif
fi;
new ss;   ss = 1 4aspect ucs·u + eps;
if ss + w6 > .25h:   new ss;   ss = .25h — w6 + eps;
fi;
rt0 x3 = round(r — 1.5u);   x4 = x3 + 5u;   y3 = y1;   y4 = y3 - ss;
rt0 x5 = round(r — 1.25u);   x6 = x4 + .5u;   y5 = y2;   y6 = y5 + ss;
call ` e arm(1, 3, 4);                                            % upper arm and serif
call ` f arm(2, 5, 6);                                            % lower arm and serif
x7 = x1;   y7 = y8 = .5[y1, y2];   x8 = good,, 11.5u;
w0 draw 7 . 8;                                                    % middle arm
if ucs ≠ 0:   x9 = x10 = x8;   y9 = y8 + .7ss;   y10 = y8 — .7ss;
     if w0 = w4:   w0 draw 9 10;                                  % middle arm serif
     else:   x11 = x8 — u;   y11 = y8;
          minvr 0;   minvs 0;
          w0 ddraw 11{1, 0} . 10{0, —1}, 8 . . 10;
          ddraw 11{1, 0} . . 9{0, 1}, 8 . . 9;                    % middle arm serif
          minvr .5;   minvs .5;
     fi;
fi;
lft0 x21 = round 1.5u;   bot0 y21 = 0;   lft0 x23 = lft1 x1;   top0 y23 = h;
w0 draw 23 . . 21;                                                % left diagonal stroke
y25 - y26 = e;
x25 - - 1 = (y25 - - y21)/(y23 — y21)[x21, x23];   x26 = x23;
w0 draw 25. . 26;                                                 % bar line
if ucs ≠ 0:   call ` g serif(21, 0, 23, —.5ucs);
     call ` h serif(21, 0, 23, ucs);                              % left serif
fi.
```



```
"Upper case ligature OE";
call charbegin( '037, 18, 0, sc, ph, 0, armic);
hpen;   new w99;   w99 = round .5[w0, w4];
lft99x1 = round 9u;   x2 = x1;   top99y1 = h;   bot99y2 = 0;
w99 draw 1.. 2;                                                        % stem
if ucs ≠ 0:
      call `b serif(1, 99, 2, .5ucs);                      %upper stem serif
      call `d serif(2, 99, 1, .5ucs);                      % lower stem serif
fi;
new ss;   ss = 1.4aspect·ucs·u + cps;
if ss + w6 > .25h:   new ss;   ss = .25h — w0 + eps;
fi;
rt0x3 = round(r — 1.5u);   x4 = x3 + .5u;   y3 = y1;   y4 = y3 — ss;
rt0x5 = round(r— 1.2521);   x6 = x4 + .5u;   y5 = y2;   y6 = y5 + ss;
call ` e arm(1,3, 4);                                    % upper arm and serif
call ` f arm(2, 5, 6);                                   % lower arm and serif
x7 = x1;   y7 = y8 = .5[y1, y2];   x8 = good,, 13.511;
w0 draw 7. .8;                                                  % middle arm
if ucs ≠ 0:   x9 = x10 = x8;   y9 = y8 + .7ss;   y10 = y8 — .7ss;
      if w0 = w4:   w0 draw 9.. 10;                          %middle arm serif
      else:   x11 = x8 — u ;   y11 = y8;
            minvr 0;   minvs 0;
            w0 ddraw 11{1,0} . . 10{0,—1}, 8.. 10;
            ddraw 11{1, O}. . 9{0, 1}, 8 . . 9;                % middle arm serif
            minvr .5;   minvs .5;
      fi;
fi;
if fixwidth ≠ 0: new save;   save = sqrttwo;
      new sqrt t wo;   sqrttwo = sqrt save;                   % super-supcrcllipse
      lft5x22 = round 1.5u;
else:   lft5x22 = round u;
fi;
x21 = 7u;   top0y21 = h + oo;   bot0y22 = —oo;
call ` a darc(2 1, 22, w5);                                % left part of bowl
x23 = x21;   y23 = y22;
lft0x24 = lft99x1;   x24 = x25;   y24 = ⅔h;   y25 = ⅓h;
w0 draw 21{1, 0} . . 24{0, —1};                      %upper right part of bowl
draw 23{ 1,0} . 25{0,1};                             % lower right part of bowl
i f ucs ≠ 0: if w0 ≠ w4:
      ddraw 21{1, 0} . . 24{0, —1}, 21{1, 0} . . 1{1, 0};            % fill in
      ddraw 23{1, 0} . . 25{0,1}, 23{1, 0} . . 2{1, O};              % fill in
      fi;
fi;
if fixwidth ≠ 0: new sqrttwo;   sqrttwo = save;
fi.
```

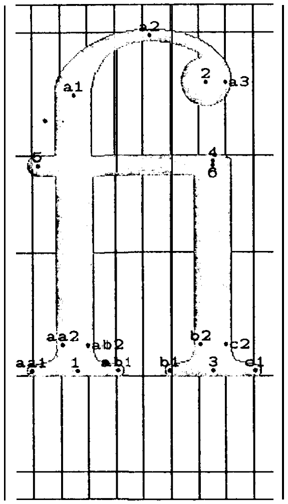


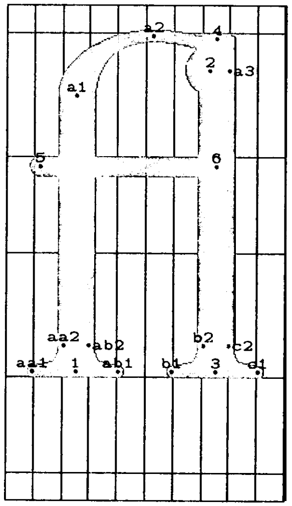


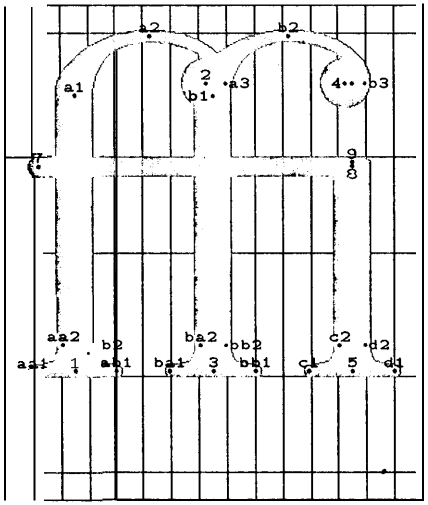
```
"Per cent sign";
call charbegin('045,17,0,0, ph + pb,pb, (ph + pb)·slant + .5pw — pu);
hpen;   x_1 = 2.5u;   x_2 = r — 1.5u;   bot_0 y_1 =  - 6 ;   top_0 y_2 = h  +  6;
w_0 draw 1 . . 2;                                                          % diagonal
lft_1 x_3 = round  u;   rt_0 x_7 = round .5(r — 3u);
new ss, st ;   ss  =  .25(x_2 — x_1)/(y_2 — y_1);                % reciprocal slope for ellipses
y_3 = y_7 = .5[y_5, y_9];   top_0 y_5 = h + b;   bot_0 y_9 = round .5h;
st = 1/(sqrt 2);   y_4 = y_6 = st[y_3, y_5];   y_8 = y_10 = st[y_3, y_9];
ss(y_5 — y_9) = x5  — x_9;   ss(y_4 — y_10) = x_4 — x_10 = x_6 — x_8;
x_37  =  .5[x_3, x_7]  =  .5[x_5, x_9];   .5[x_4, x_10]  =  st[x_37, x_3];   .5[x_6, x_8] = st[x_37, x_7];
draw |w_1#|3{ss, I}. |5/6[w_0, w_1]|4{x_5 — x_3 + ss(y_5 — y_3), y_5 — y_3} .
     |w_0#|5{1, 0} . 6{x_7 — x_5 + ss(y_7 — y_5), y_7 — y_5} . 7{—ss, —1} .
     8{x_9 - - x_7 +  ss(y_9 - - y_7), y_9 — y_7} . 9{—1, 0} .
     |5/6[w_0, w_1]| 10{x_3 - - x_9 -I- ss(y_3 — y_9), y_3 — y_9} |w_1#|3{ss, 1};        % upper bowl
y_1 I = .9[y_7, y_5];   x_11 = (sqrt(1 — (.9)(.9)))[x_37, x_7] + ss(y_1 , -- y_7);      % p·point or, ellipse
y_12 = .901[y_7, y_5];   x_12 = (sqrt(1 -- (.901)(.901)))[x_37, x_7]-+ ss(y_12 — y_7);   % nearby point
w_0 draw  (12. )11 . 2{x_2 — x_1, y_2 — y , };                                          % link
x_13 — x_3 = x_11 — x_4 = x_1 j  —  Xj  = x_16 — x_6 = x_17 —  x_7 =
     x_18 — x_8 = x_19 — x_9 = x_20 -   x_10 = round .5(r + u);
y_13 — y_3 =  y_14 — y_4 = y_15 .  . y_5 = y_16 — y_6 =  y_17 — y_7 = y_18 .  . y_8 = y_19 — y_9 = y_20 .  . y_10;
bot_0 y_19  =  —b;
draw |w_1#|13{ss, 1} |5/6[w_0, w_1]|14{x_5 —  x_3  +  ss(y_5 — y_3), y_5 — y_3} .
     |w_0#|15{1, 0}16{x_7 — Xj  + ss(y_7  —  y_5), y_7  --  y_5} 17{—ss, —1} .  .
     J 8{x_9 — x_7 + ss(y_9 — y_7),y_9 - y_7}.19{—1, 0}   .   .
     |5/6[w_0, w_1]|20{x_3 — x_9  +  ss(y_3 .   . y_9), y_3 — y_9} . |w_1#|13{ss, 1}.        % lower bowl
```

The file roml ig. mf

```
"The ligature R';
call charbegin('173, 10, 0, 0, ph, 0, ph slant + 2pu);
hpcn;   x_1  =  good, 2.5u;   x_3 = good_1(r — 2.5u);
if w_0  =  w,:   rt_3 x_2  =  round 6.5u;   x_4 — x_3 = x_2 — x_1;
else:   r t _3 x_2 = round(.5r  +  2u);   r t _3 x_1  =  round(r  +  1.5u);
fi;
cpen;   top_3 y_2 = .8[m, h];   y_4 = y_2;
call `a fstroke(2, 1);                                  % left bulb, shoulder, stem, and serif
call `b fstroke(4, 3);                                  % right bulb, shoulder, stem, and serif
lft_10 x_5 = lft_1 x_1 - - u — eps;   rt_10 x_6 = rt_1 x_3 -+ U  +  eps;   top_10 y_5 = m;   y_6 = y_5;
w_10 draw 5.. 6.                                                                  % bar link

lig 'f:   ` i  =  '174,  ` f =' 173,  ` 1 ='  175;
```

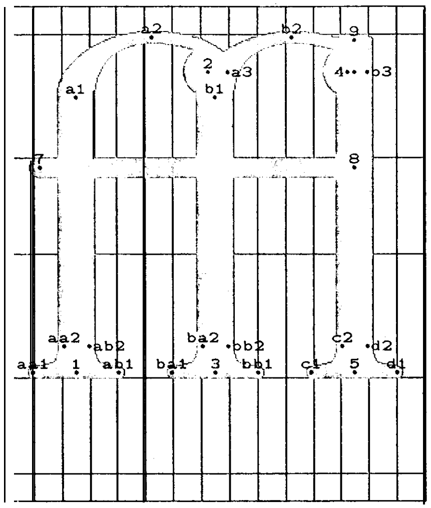




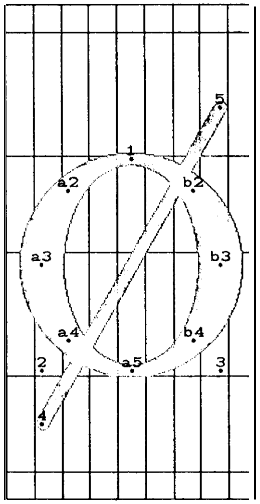


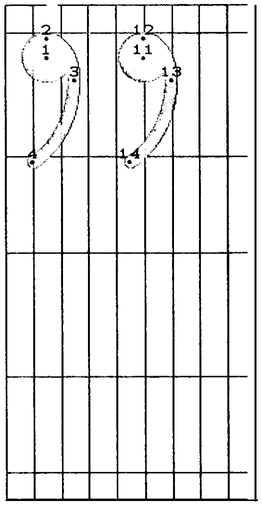
6

```
"The ligature fi";
call charbegin( '174, 10, 0, 2sc, ph, 0, ph·slant + .5pwi + (2sc — 2)pu);
hpen;   x1 = good, 2.5u;   x3 = x4 = good1(r — 2.5u);
rt3x2 = rt1x3;
cpen;   top3y2 = .8[m, h];
call `a fstroke(2, 1);                                   % bulb, shoulder, left stem, left serif
hpen; top0y1 = m; bot0y3 = 0; w1 draw 3..4;                                         % right stem
cpen;   lft10x5 = lft1x1 — u — eps;   x6 = x3;   top10y5 = m;   y6 = y5;
w10 draw 5.. 6;                                                                       % bar link
if lcs ≠ 0:   call ` b serif (3, 1, 4, — lcs);
     call ` c serif (3, 1, 4, lcs);                                                 % right serif
fi.                                                                 % appropriate end lo "fi"

"The ligature fl";
call charbegin('175, 10, 0, 2sc, ph, 0, ph·slant + .5pwi + (2sc -- 2)pu);
hpen;   x1 = good, 2.5u;   x3 = x4 = good1(r — 2.5u);
rt3x2 = rt1x3;
cpen;   top3y2 = .9[m, h];
call `a fstroke(2, 1);                                    % bulb, shoulder, left stem, left serif
hpcn; top0y1 = h ; bot0y3 = 0; w1 draw 3.. 4;                                       % right, stem
cpen;   lft10x5 = lft1x1 -- u — eps;   x6 = x3;   top10y5 = m;   y6 = y5;
w10 draw 5.. 6;                                                                       % bar link
if lcs ≠ 0:   call ` b serif(3, 1, 4, —lcs);
     call ` c serif (3, 1, 4, lcs);                                                 % right serif
fi.

"The ligature ffi";
call charbegin('176, 15, 0, 2sc, ph, 0, ph·slant + .5pwi + (2sc — 2)pu);
hpen;   x1 = good, 2.5u;   x3 = good, .5r;
x j — x3 = x1 — x2 = x3 — x1;   x6 = x5 = x9;   y2 = y1 = y6;
cpen;   top3y2 = .8[m, h];   rt 3x1 = rt1x6;
call 'a fstroke(2, 1);                                    % left bulb, shoulder, stem, and serif
call ` b fstroke(4, 3);                       % right bulb and shoulder, middle stem and serif
hpen; top0y9 = m ; bot0y5 = 0; w1 draw 5.. 9;                                       % right stem
cpen;   lft10x7 = lft1x1 -- u — eps;   x8 = x5;   top10y7 = m;   y8 = y7;
w10 draw 7.. 8;                                                                       % bar link
if lcs ≠ 0:   call `c serif(5, 1, 6, —lcs);
     call `d serif(5, 1, 6, lcs);                                                   % right serif
fi.

lig '173:   `i = '176, `1 = '177;
```



6

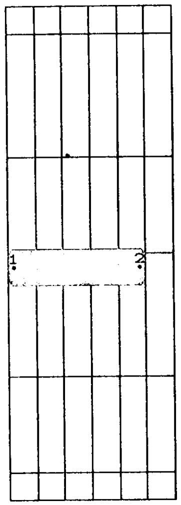


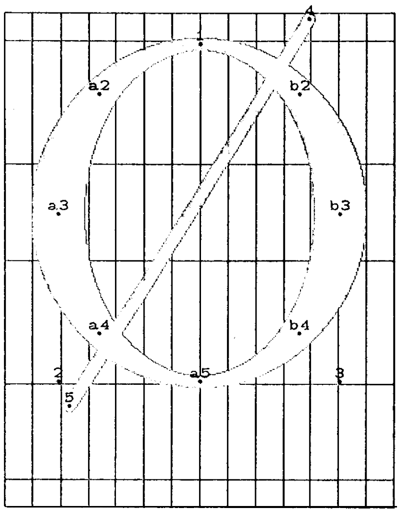


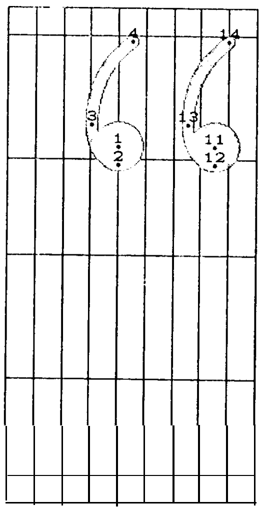
```
"The ligature ffl";
call charbegin(´177, 15, 0, 2sc, ph, 0, ph·slant + .5pwi + (2sc — 2)pu);
hpen;   x_1 = good_1 2.5u;   x_3 = good, .5r;
x_5 — x_3 = x_4 — x_2 = x_3 — x_1;   x_6 = x_5 = x_9;   y_2 = y_1 = y_6;
cpen;   top_3 y_2 = .9[m, h];   rt_3 x_1 = rt_1 x_6;
call `a fstroke(2, 1);                         % left bulb, shoulder, stem, and serif
call `b fstroke(4, 3);               % right bulb and shoulder, middle stem and serif
hpen; top_0 y_9 = h; bot_0 y_5 = 0; w_1 draw 5.. 9;                      % right stem
cpen;   lft_10 x_7 = lft_1 x_1 — u — eps;   x_8 = x_5;   top_10 y_7 = m;   y_8 = y_7;
w_10 draw 7.. 8;                                                           % bar link
if lcs ≠ 0:   call `c serif(5, 1, 6, —lcs);
     call `d serif(5, 1, 6, lcs);                                        % right serif
fi.
```

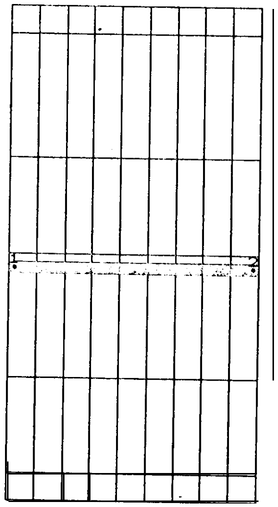
The file `romitl mf`

```
% This file defines characters common to roman and italic text fonts
% that do not appear in non-ligature fonts.
% Character codes ‘040, ‘0 12, ‘055, ´100, ‘134, ´ 136, ´ 137 are used.

"The Scandinavian letter o-slash";
call charbegin( ‘040, 9, 0, 0, px + .5pd, .5pd, (px + .5pd)·slant + .5(pw — pwii));
hpen;   x_1 = r — x_1;                                % axis of left-right symmetry
if fixwidth = 0:   lft_2 x_2 = round .5u;
else:   lft_2 x_2 = round 1.5u;
fi;
x_1 — x_2 = x_3 — x_1;   top_0 y_1 = m + oo;   bot_0 y_2 = --00;   y_2 = y_3;
call `a darc(1, 2, w_2);                                        % left part of bowl
call `b darc(1, 3, w_2);                                       % right part of bowl
x_4 = x_2;   x_5 = x_3;   y_4 = —.5d;   y_5 = m + .5d;
w_0 draw 4..5.                                                         % diagonal

"Closing quotes";
call charbegin(´042, 9, 0, 0, ph, 0, 2/3[px, ph]·slant + .5pw — 2.5pu);
% There is rotational symmetry with respect to opening quotes.
cpen;   top_3 y_1 = top_6 y_2 = h;   top_6 y_1 = m;   y_3 = 2/3[y_1, y_2];
x_1 = x_2 = good:, 1.5u;   x_3 = good_0(x_1 + u + eps);   x_4 = x_1 — .5u — cps;
x_11 --- x_1 = x_12 -- x_2 = x_13 -  x_3 = x_1 , --- x_1 = round 3.5u;
y_11 = y_1;   y_12 = y_2;   y_13 = y_3;   y_14 = y_1;
w_3 draw 1;                                                           % left bulb
draw 11;                                                             % right bulb
hpen;   w_0 draw 2{1, 0} 3{0, —1} .. 4{3(x_4 — x_3), y_4 — y_3};        % left tail
draw 12{1, 0} .13{0, —1} .. 14{3(x_14 — x_13), y_14 — y_13}.           % right tail

lig ` ': ` ' = ´042;
```

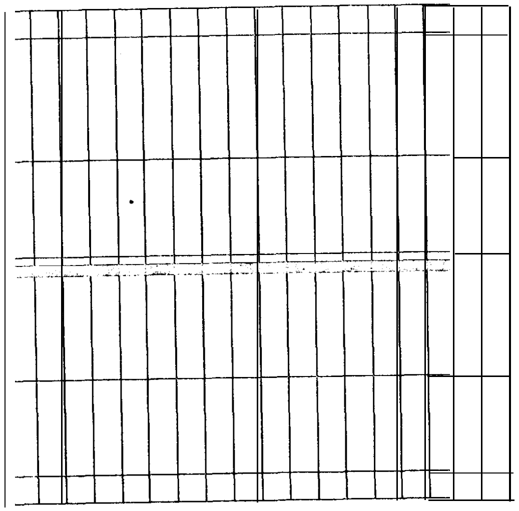


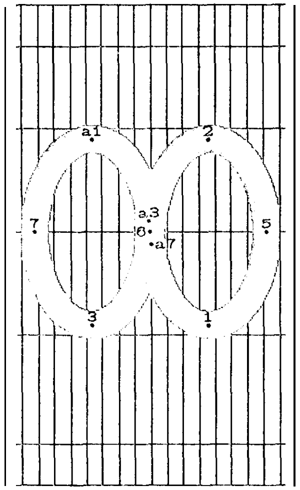


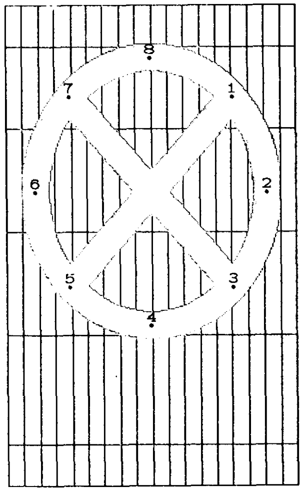




```
"Hyphen";
call charbegin('055, fixwidth[6, 9], 0, 0, px, 0, .5px·slant — .5pu);
vpen;   y1 = y2 = .5m;
if fixwidth = 0:   lft7x1 = 0;   rt7x2 = r - u ;
else : lft7x1 = 1.5u; x2 = r — x1;
fi;
w7 draw 1..2.                                                        % bar

"The Scandinavian letter O-slash";
call charbegin( '100, 14, 0, 0, ph + ph, pb, .5ph·slant — .5pu);
hpen;
if fixwidth ≠ 0 : new save; save = sqrttwo;
      new sqrttwo;    sqrttwo = sqrt save;                 % super-superellipse
      lft5x2 = round 1.5u;
else:   lft5x2 = round u;
fi;
x1 = r — x1;                                   % axis of lrft-right symmetry
top0y1 = h + 00;
bot0y2 = --00;   y3 = y2;   x3 = r — x2;
call `a darc( 1, 2, w5);                                  % left part of bowl
call `b darc( 1, 3, w5);                                  %rightpart of bowl
if fixwidth ≠ 0: new sqrttwo; sqrttwo = save;
fi;
top0y4 = h + b;   bot0y5 = —b;   rt0x4 = lft5x3;   lft0x5 = rt0x2;
w0 draw 4..5;                                                    % diagonal
fi.

"Opening quotes";
call charbegin('134, 9, 0, 0, ph, 0, ph·slant + .5pw — .5pu);
% There is rotational symmetry with respect to closing quotes.
cpen;   bot3y1 = bot6y2;   top6y1 = h;   y3 = 2/3[y1, y2];   y2 — y4 = m — h;
x1 = x2 = good3 4u;   x3 = good0(x1 — u - - eps);   x4 = x1 + .5u + eps;
x11 — x1 = x12 -- x2 = x13 — x3 = x14 — x4 = round 3.5u;
y11 = y1;   y12 = y2;   y13 = y3;   y14 = y4;
w3 draw 1;                                                      %left bulb
draw 11;                                                       % right bulb
hpen;   w0 draw 2{—1, 0 } 3{0, 1} 4{3(x4 — x3), y4 — y3};          %left tail
draw 12{—1, O}. . 13{0,1}.14{3(x14 -- x13), y14 — y13}.          % right tail
lig `': `' = '134;

"En dash";
call charbegin('136, 9, 0, 0, .5px + 5pw·aspect, 0, .5px·slant + .5pu);
hpen; lft0x1 = 0; rt0x2 = r;   y1 = y2 = .5m;
w0 draw 1.. 2.                                                      % bar
lig `-: -- = '136;
```



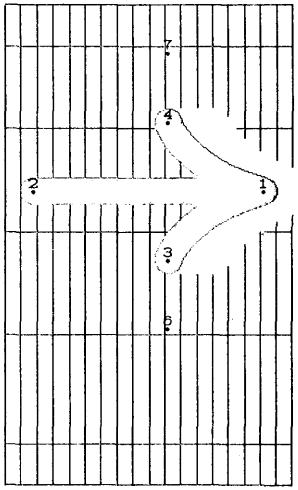


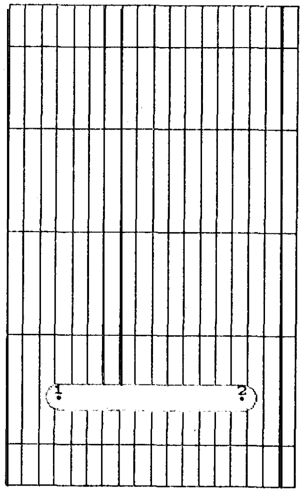


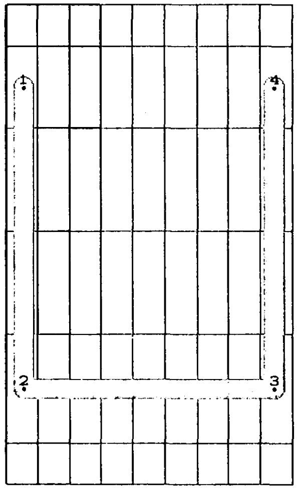
```
"Em dash";
call charbegin('137, 18, 0, 0, 5px + .5pw·aspect, 0, .5px·slant + .5pu);
hpen; lft0x1 = 0 ; rt0x2 = r ;   y1 = y2 = .5m;
w0 draw 1 . 2.                                                   % bar
lig '136: -- = '137;
```

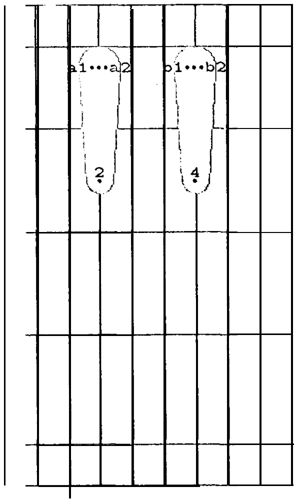
The file romits mf

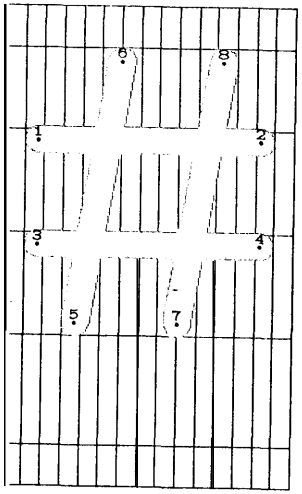
```
% This file contains the characters substituted into a text font when ligs = 0 is set.
% Character codes '025, '026, '031, '032, '040, '042 '044, '055,
% '100, '134, '136, '137, and '173-'177 are affected.
"Infinity";
call charbegin( '025, 18, 0, 0, px, 0, .5px·slant — .5pu);
new w98, w99;   w99 = round .25[w6, w7];   w98 = 2[w1, w2];
vpen;   top99y2 = m  +  oo; bot99y1 = —oo; y2 = y4;   y1 = y3;
x1 = x2;   x3 = x4;   y5 = y6 = y7 = .5[y1, y2];
lft0x7 = round u;   rt0x5 = round(r — u);   x6 = 5[x7, x5];
new ss, mss;   ss = —6u/m;                       % reciprocal slope at center
if w0 = w98:   mss = ss;
else:   mss = .75ss;
fi;
call 'a zdraw(5, 1, 6, 4, 7, w99, w98, ss);        % lower right and upper left strokes
w99 draw 5{ 0, 1}. 2{ -- 1, 0}. 6{ mss, —1} .
     3{—1, 0} . . 7{0, 1}.                        % upper right and lower left strokes

"Circle-times operator";
call charbegin('026, 18, 0, 0, ph, ph -- 2pa, pa·slant -- .5pu);
cpen;   lft0x6 = round u;   y6 = a;   x8 = r -- x8;   top0y8 = h + oo;
call circle(1 2 3, 4, 5, 6, 7, 8, w0);                                    % bowl
w0 draw 7 . . 3';'                                % upper left to lower right diagonal
draw 5 . . 1.                                     % lower left to upper right diagonal
```



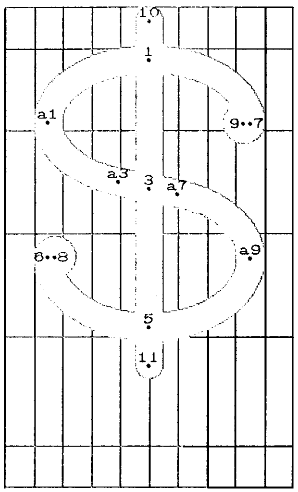


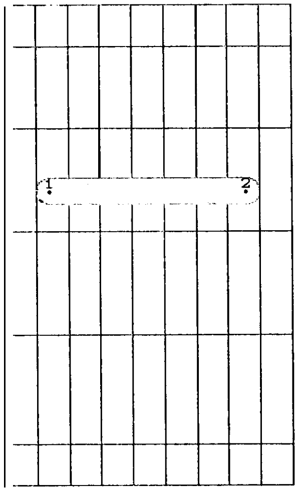


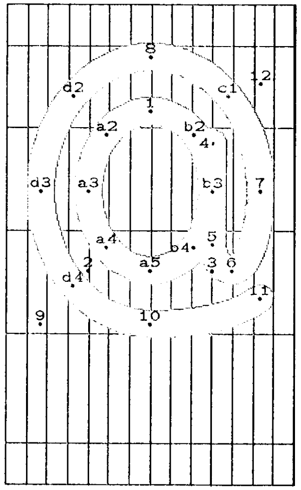




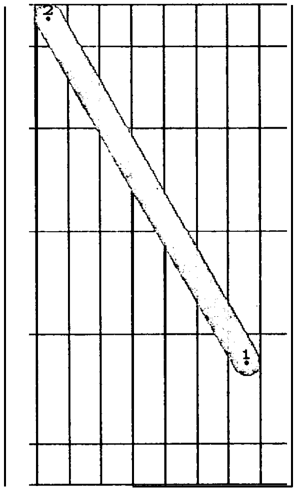
```
“Rightward arrow”;
call charbegin(´031,18,0,0,.24ph + .5prt + pa, .24ph + .5prt— pa, pa·slant—.5pu);
cptn;   lft_10 x_2 = round u;   rt_10 x_1 = x_0 = round(r -- u);
y_1 = y_2 = y_5 = y_8 = good_10 a;
w_10 draw 1 .. 2;                                                  % bar
hpen;   rt_1 x_8 = x_0;
x_5 — x_8 = x_8 — x_3 = fix width[3u, 6u] + eps;   x_3 = x_1 = x_6 = x_7;
y_3 — y_6 = y_1 — y_3 = y_4 — y_1 = y_7 -- y_4 = .24h + eps;
rpen#;   w_10 + w_1 draw (5.. )8 .. 3( .. 6);          % erase excess at lower right
hpen;   draw (|w_1|5 .. )8 .. |w_0|3( .. 6);                    %lower point
rpen#;   w_10 + w_1 draw (5. )8 .. 4( .7);             %erase excess at upper right
hpen;   draw (|w_1|5 .. )8 .. |w_0|4( .. 7).                    %upper point

“Underbar suitable for < and >”;
call charbegin(´032,18,0,0,0,.5[px,ph] + prt/2 -- 2pa + (px — pe), 0);
y_22 = good_10 .5[m,h];   5[y_22, y_2 3] = good_10 a;   y_1 = y_2 = y_2 1 — (m — e);
cpen;   lft_10 x_1 = round 2.5u;   x_2 = r — x_1;
w_10 draw 1 .. 2.                                                  % bar

“Blank-space sign”;
call charbegin( ‘040, 9,0,0,.5[px,ph],.5pd,.5[px,ph]slant + .25pu);
cpen;   new w_99;   w_99 = round(75pixels pw + blacker);   %smaller than hairline
lft_99 x_1 = round .25u;   rt_99 x_4 = round(r — .25u);   x_1 = x_2;   xj = x_4;
y_1 = y_4 = .5[m,h];   y_2 = y_3 = good_99(—.5d);
w_99 draw 1 .. 2 .. 2 .. 3 . . 3 .. 4 .               % left stem, bar, right stem

“Straight double quotes”;
call charbegin(´042,9,0,0, ph, 0, ph·slant + .5pwiii -- 2.5pu);
new w_99;
i f w_3 < w_0 sqrt 2:   w_99 = round w_0 sqrt 2;
else:   w_99 = w_3;
fi;
x_1 = x_2 = good_99 3u;   x_3 = x_4 = r — x_1;
cpen;   top_99 y_1 = h;   y_2 = .5[e, m];   y_3 = y_1;   y_4 = y_2;
call ` a cdraw( 1, 2, 99, 0);                                      % left stem
call `b cdraw(3, 4, 99, 0).                                        % right stem

“Sharp symbol(number sign or hash mark)“;
call charbegin( ‘043, 15, 0, 0, ph, ph—2pa, 0);
cpen; lft_10 x_1 = round u;   x_3 = x_1;   x_2 = x_1 = r — x_1;
y_1 = y_2;   y_3 = y_4;   y_1 — y_3 = round(m -- e);   .5[y_1, y_3] = a;
w_10 draw 1 .. 2;                                                  %upper bar
draw 3.. 4;                                                        % lower bar
x_5 — 2u = x_1;   x_8 + 2u = x_2;   x_6 — x_5 = x_8 — x_7;   x_6 — x_7 = fixwidth[0, --3u];
y_5 = y_7;   y_6 = y_8;   top_10 y_6 = h;   .5[y_5, y_6] = a;
draw 5.. 6;                                                        % left diagonal
draw 7.. 8.                                                        % right diagonal
```

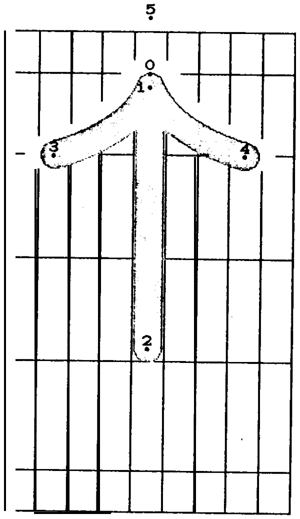


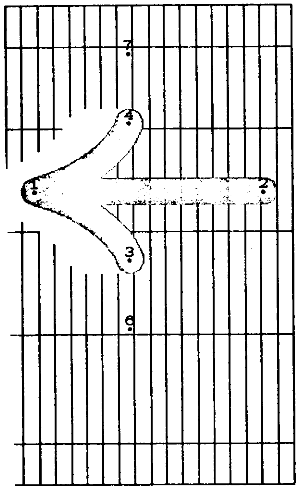


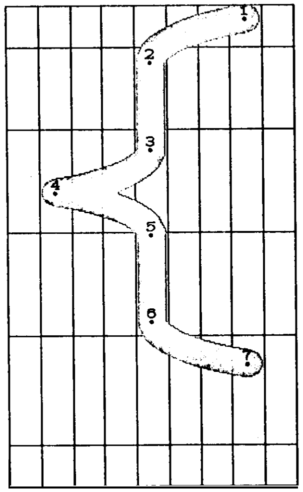


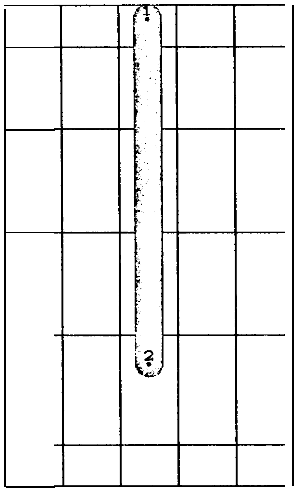
```
"Dollar sign";
call $charbegin$( '044, 10, 0, 0, $ph + pb, pb, ph\cdot slant - .5pu$);
hpen;  $top_0 y_1 = h + oo$;  $bot_0 y_5 = -oo$;
$x_3 = good_{10}\,.5r$;  $y_3 = .52h$;  $lft_{11} x_2$ = round $u$;  $x_4 = r - x_2$;
if $ucs = 0$:   $x_1 = x_5 = x_3$;  $y_6 = .5[y_5, y_3]$;  $y_7 = .5[y_1, y_3]$;
else:  if W(j $= w_9$:  $x_1 = x_5 = x_3$;  $y_6 = .5[y_5, y_3]$;  $y_7 = .5[y_1, y_3]$;
    else:  $x_1 + .5u = x_5 - .5u = x_3$;  $y_6 = h/4 - 1$;  $y_7 = .8h + 1$;
    fi;
fi;
$y_8 = y_6$;  $y_9 = y_7$;  cpen; $lft_3 x_8 = lft_0 x_6$; $rt_3 x_9 = rt_0 x_7$; $x_6 = x_2$; x   7 $= x_4$;
$w_3$ draw 8 ;                                            % lower bulb
draw 9;                                                   % upper bulb
hpen;  $w_0$ draw 6{0, --I} 5{1, O } ;                    % lower left stroke
draw 7{0,1}.1{—1, O };                                    %upper right stroke
call ˋa $sdraw$(l, 2, 3, 4, 5, $w_{11}, w_9, -h/(50u)$);  % middle stroke
cpen;  $x_{10} = x_{11} = x_3$;  $top_{10} y_{10} = h + b$;  $bot_{10} y_{11} = -b$;
$w_{10}$ draw 10.. 11.                                    % stem

"Minus sign";
cpen;
if $fixwidth = 0$:  if pa +- $8pu$ > ph:
        call $charbegin$( '055, 18, 0, 0, ph, ph $- 2pa, pa\cdot slant - .5pu$);
    else:  call $charbegin$( '055, 18, 0, 0, $8pu$ + pa, $8pu$ -- pa, $pa\cdot slant - .5pu$);
    fi;
else:  call $charbegin$('055, 9, 0, 0, $3.5pu + pa, 3.5pu -$ pa, 0);
fi;
$lft_{10} x_1$ = round $u$;  $x_2 = r - x_1$;  $y_1 = y_2 =$ a ;
$w_{10}$ draw 1.. 2.                                      % bar

"At sign";
call $charbegin$( '100, 14, 0, 0, $ph$, 0, $.5ph\cdot slant + .5pw - 5pu$);
hpen;  $x_1 = x_8 = x_{10} = .5r$;  $x_2$ = good, $4u$;  $x_3 = x_4 = x_5 = r - - x_2$;
$lft_0 x_9 =$ round $u$;  $x_7 = x_{11} = x_{12} = r - - x_9$;  $x_6 = \frac{5}{12}[x_5, x_7]$;
$top_0 y_8 = h + oo$;  $bot_0 y_{10} = -oo$;  $y_9 = y_{10}$;
$y_1 = good_6\,.8[y_{10}, y_8]$;  $y_2 = y_3 = y_6 = good_6\,.2[y_{10}, y_8]$;
$y_7 = .5[y_{10}, y_8]$;  $y_{11} = good_6\,.1[y_{10}, y_8]$;  $y_{12} = .9[y_{10}, y_8]$;
$y_4 = 8[y_2, y_1]$;  $y_5 = \frac{2}{3}[y_7, y_6]$;
call ˋa $darc$(1, 2, $w_1$);                              %left part of inner bowl
call . b $darc$(1, 3, $w_0$);                             % right part of inner bowl
draw |$w_1$|4 . . |$w_1$#|5{0, - 1 )  |$w_0$#|6{1, 0). 7{0, 1);   % stem and link
call ˋ c $arc$(8, 7, $w_0$);                              % right part of outer bowl
call ˋd $darc$(8, 9, $w_0$);                              %left part of outer bowl
$w_0$ draw LO{ 1, O}. 11( . . 12).                        % point
```



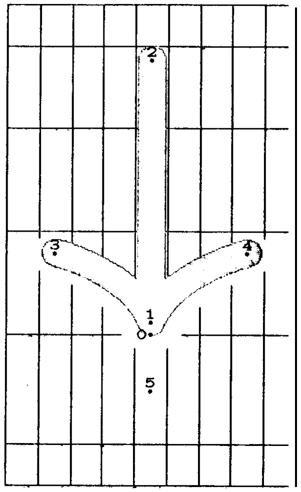


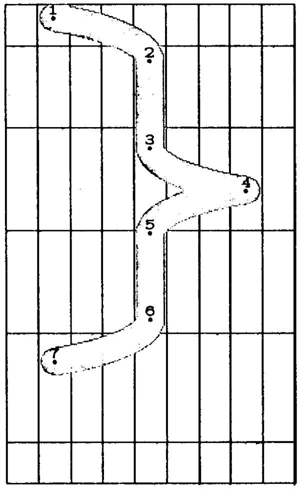


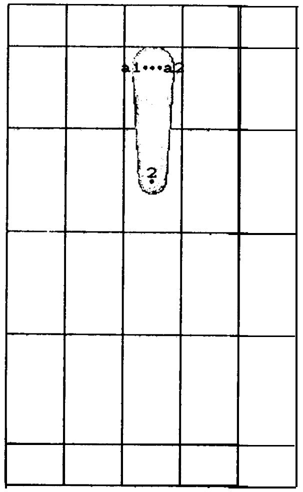




```
'Reverse slash";
call charbegin('134, 9, 0, 0, ph + pb, ph + pb — 2pa, 0);
cpen;  lft10x2 = round u;  x2 = r — x1;
top10y2 = h + b;  .5[y1, y2] = a;
w10 draw 1 . . 2.                                          % diagonal

"Upward arrow";
call charbegin('136, 9, 0, 0, ph, ph — 2pa, .75ph·slant + .5pw — pu);
cpen;  top10y1 = y0 = h;  .5[y1, y2] = a;
x0 = x1 = x2 = x5 = x8 = good,,, .5r;
w10 draw 1 . . 2;                                          % stem
vpen;  top7y8 = y0;
lpen#;  w10 draw 0.. 8; rpen#; w10 draw 0. .8;             % clean the top
y5 — y8 = y8 — y3 = .24h + eps;  y3 = y4 = y6 = y7;
x3 — x6 = x1 — x3 = x4 — x1 = x7 — 54 = 3u + eps;
lpen#;  w10 draw (5. .)8 . 3( . . 6);                      % erase excess at left
vpen;  draw (|w7|5 . . )8 . . |w6|3( . . 6);               % left point
rpen#;  w10 draw (5. . )8 . . 4( . . 7);                   % erase excess at right
vpen;  draw (|w7|5 . . )8 . . |w6|4( . . 7).               % right point

"Leftward arrow";
call charbegin('137, 18, 0, 0, .24ph + .5prt + pa, .24ph + .5prt — pa, pa·slant — .5pu);
cpen;  lft10x1 = x0 = round u;  rt 10x2 = round(r — u);
y1 = y2 = y5 = y8 = good10a;
w10 draw 1.. 2;                                            % bar
hpen; lft1x8 = x0;
x5 — x8 = x8 — x3 = —fixwidth[3u, 6u] — eps;  x3 = x4 = x6 = x7;
y3 — y6 = y1 — y3 = y4 — y1 = y7 — y4 = .24h + eps;
lpen#;  w10 + w1 draw (5. . )8 . . 3( . . 6);              % erase excess at lower left
hpen;  draw (|w1|5 . . )8 . . |w0|3( . . 6);               % lower point
lpen#;  w10 + w1 draw (5 . )8 . . 4( . . 7);               % erase excess at upper left
hpen;  draw (|w1|5 . . )8 . . |w0|4( . . 7).               % upper point

"Left brace";
call charbegin('173, 9, 0, 0, ph + pb, ph + pb — 2pa, (ph + pb)·slant + .5pw — pu);
hpen;  x2 = x3 = x5 = x6 = good1 .5r;  x1 — x2 = x2 — x4 = 3u + eps;  x1 = x7;
top0y1 = h + b;  y4 = .5[y1, y7] = .5[y2, y6] = .5[y3, y5] = good6a;
y1 - - y2 = y3 - - - y4 = (y1 - - y4)/4;
draw |w0#|1{3(x2 — x1), y2 — y1} . . |w1#|2{0, —1} . . |w1#|3{0, —1} . .
     |w0#|4{3(x1 — x3), y4 — y3};                          % upper stem
draw |w0#|7{3(x6 — x7), y6 — y7} . |w1#|6{0, 1} . |w1#|5{0, 1} . .
     |w0#|4{3(x4 — x5), y4 — y5}.                          % lower stem

"Vertical line";
call charbegin('174, 5, 0, 0, ph + pb, ph + pb — 2pa, (ph + pb)·slant + prt/2 — 2pu);
cpen;  x1 = x2 = good10 .5r;  top10y1 = h + b;  .5[y1, y2] = a;
w10 draw 1.. 2.                                            % stem
```





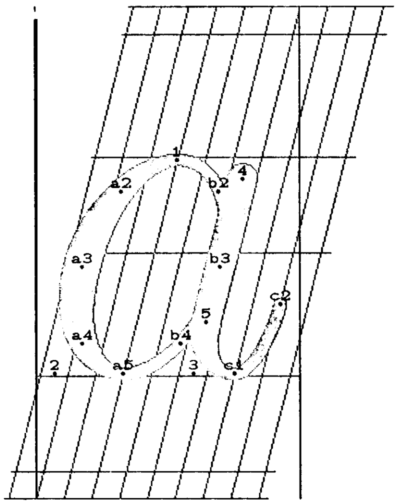


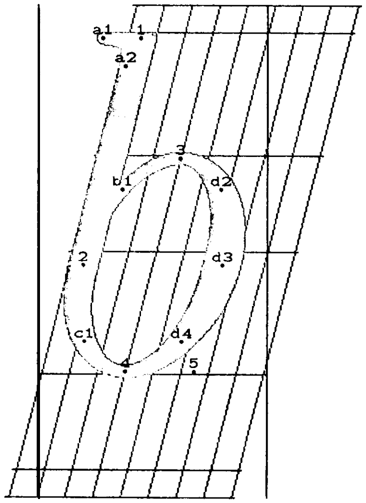
```
"Downward arrow";
call charbegin('175,9,0,0,ph,ph—2pa, 0);
cpen;  top_{10}y_2 = h;  .5[y_1, y_2] = a;  y_0 = bot_{10}y_1;
x_0 = x_1 = x_2 = x_5 = x_8 = good_{10} .5r;
w_{10} draw 1..2;                                              % stem
vpen;  bot_7y_8 = y_0;
lpen#;  w_{10} draw 0.. 8; rpen#;w_{10} draw 0.. 8;            % clean the top
y_5 — y_8 = y_8 — y_3 = —.24h — eps;  y_3 = y_4 = y_6 . y_7;
x_3 — x_6 = x_1 — x_3 = x_4 — x_1 = x_7 — x_4 = 3u + eps;
lpen#;  w_{10} draw (5 . )8..3(..6);                    % erase excess at left
vpcn;  draw (|w_7|5..)8..|w_6|3(..6);                           % left point
rpen#;  w_{10} draw (5.. )8..4(..7);                   % erase excess at right
vpen;  draw (|w_7|5..)8..|w_6|4(..7).                          % right point

"Right brace";
call charbegin('176,9,0,0,ph + pb, ph + pb — 2pa,(ph + pb)·slant + .5pwi — 4pu);
hpen;  x_2 = x_3 = x_5 = x_6 = good_1 .5r;  x_1 — x_2 = X',', — x_1 = —3u — eps;  x_1 = x7;
top_0y_1 = h + b;  y_4 = .5[y_1, y_7] = .5[y_2, y_6] =.5[y_3,y_5]= good_6a;
y_1 — y_2 = y_3 — y_4 = (y_1 — y_4)/4;
draw |w_0#|1{3(x_2 — x_1), y_2 — y_1}..|w_1#|2{0, -1).|w_1#|3{0, —1}. .
     |w_0#|4{3(x_4 — x_3), y_4 — y_3};                         % upper stern
d r a w |w_0#|7{3(x_6 — x_7), y_6 — y_7}|w_1#|6{0,1}.|w_1#|5{0,1}. .
     |w_0#|4{3(x_4 — x_5), y_4 — y_5}.                          % lower stem

"Straight single quote";
call charbegin( '177,5,0,0,ph,0,0);
new w_{99};
if w_3 < w_0 sqrt 2:   w_{99} = round w_0 sqrt 2;
else:   w_{99} = w_3;
fi;
x_1 = x_2 = good,, .5r;
cpen;  top_{99}y_1 = h;  y_2 = .5[e, m];
call ` a cdraw(1,2,99,0).                                             % stem
```

ITALIC CHARACTER DESIGNS

The Ale italic.mf

```
% The Compuler Modern Italic family of fonts (by D. E. Knuth, 1979).
danger = mi/8;
input romitu;                                              % upper case (majuscules)
input ital 1;                                              % lower case (minuscules)
input itald;                               % numerals, ampersand, and question mark
input romitp;                       % punctuation symbols common to all roman and italic
if mi ≠ 0:                            % the following is the "easy" case, a math italic font
    input greekl;                                          % lower case Greek letters
    input italms;                                     % special symbols for math italic
    input itmcxt;                   % nonstandard characters (codes '134, '136, '137)
    texinfo slant, 6pu,3pu,2pu, px, 18pu,2pu;
else:                                 % the following is the "hard" case, a text italic font
    input rom i ta;       %accents and other symbols common to roman and italic text
    if Jigs ≠ 0:  input ital ig;                       % letter ligatures (codes '173- '177)
        input itals;                                   % miscellaneousletter combinations
        input romitl;                                        % ligatures common with italic
        input i ttext;                       % nonstandard characters (codes '043, '044)
    else:  input rom i ts;                                     % substitutes for ligatures
    fi;
    if fixwidth = 0:  new k, kk, kkk;                          % three degrees of kerning
        k = —.5pu;  kk = —1.5pu;  kkk = —2.5pu;  uk = +pu;
        lig 'd: `w:`  1:  '1  kern uk;
        Iig  'F:  'V:
            'A kern kkk, `o kern kk, `e kern kk,
            'a kern kk, `u kern kk, `r kern kk,
        'K: 'X:
            '0 kern k,`C kern k,`G kern k, `Q kern k;
        Iig 'T:  `y kern kk,
        `Y:  `o kern kk, `e kern kk,
            'a kern kk, ` u kern kk, `r kern kk,
        'P:  `W:  'A kern kk;
        lig `O:  'A kern k,`W kern k,`Y kern k, . V kern k,`X kern k;
        if ucs ≠ 6: Iig `R:;
        fi;
        lig 'A:  `t kern k,`b kern k,`h kern k,` k kern k,` 1 kern k,
            `m kern k,` n kern k,`r kern k,`v kern k,` w kern k,
            'L:  ` i kern k,`u kern k,
            `T kern kk, `0 kern k,`U kern k,`C kern k,
            `W kern kk, `Y kern kk,`G kern k,`V kern kkk,
            `Q kern k,
            'b:     `c: `e: `o: `p: `r:
            ` a kern k,` c kern k,` d kern k,` e kern k,` g kern k,
            ` o kern k,` q kern k;
```

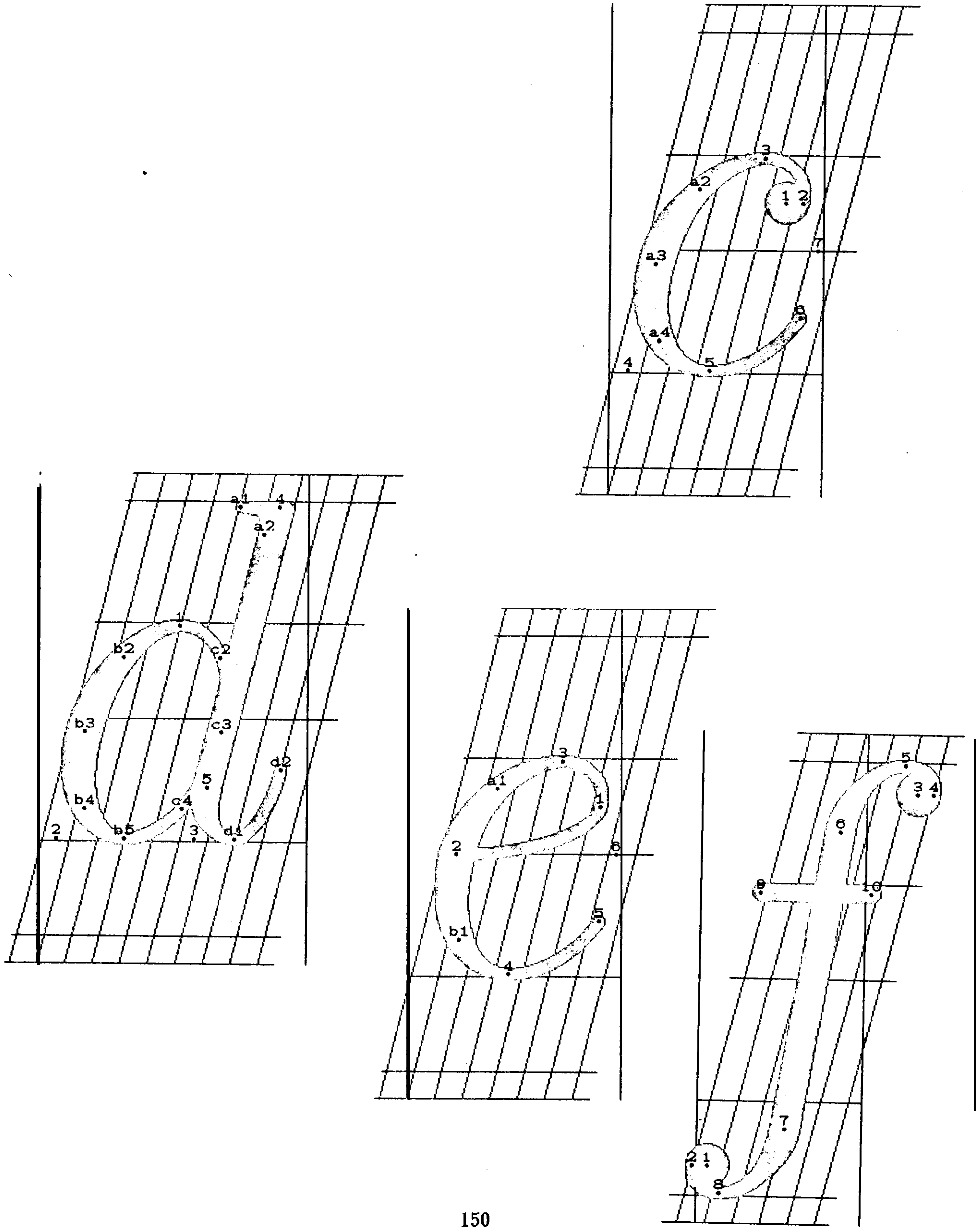




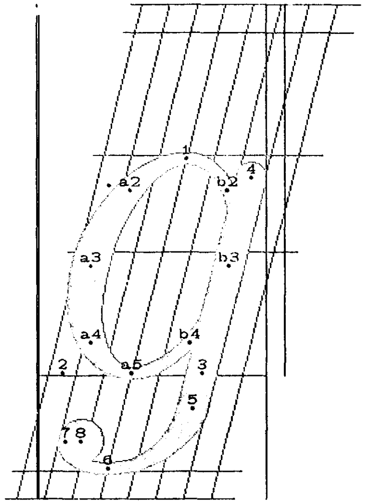
```
        texinfo slant, 6pu,3pu,2pu, px, 18pu,2pu;
    else: texinfo slant,9pu, 0, 0, px, 9pu,9pu;
    fi;
fi.
```

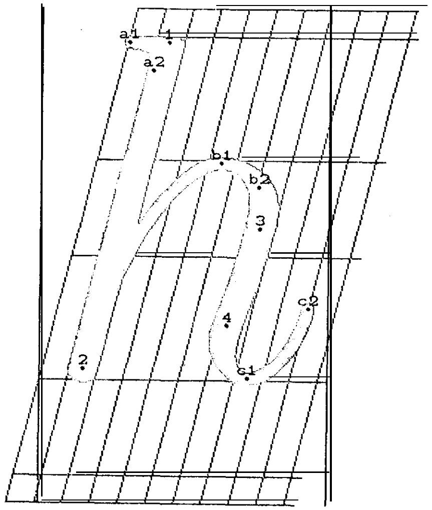
The file itall.mf

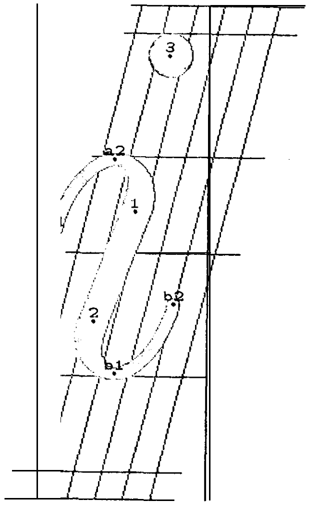
```
% This lower-case italic alphabet was prepared by D. E. Knuth in December, 1979,
% inspired by the Monotype alphabet used in The Art of Computer Programming.
% Math spacing is obtained by setting mi = 1, otherwise set mi = 0.
% Character codes '141-'172 are generated.

new mc, lbowl, lhook, rbowl, rhook, rstem;      % quantities used in spacing corrections
mc = mi/pu;
rhook = 1/3px·slant + .5pw + .5pu;
lbowl = .3px·slant - .5pwii + pu;
rbowl = .7px·slant + .5pwii - pu;
lhook = 2/3px·slant - .5pw - .5pu;
rstem = px slant + .5pwi - pu;

"Italic letter a";
call charbegin('a, 9, mc·lbowl, -mc·rhook, px, 0, mi[rhook,0]);
cpen;  x1 = .5[x2, x3];  x2 = good2 1.5u;  x3 = x4 = x5 = good1(r - 2.5u);
top6y1 = m + oo;  bot6y2 = -oo;  y3 = y2;  top1y4 = m - .02h;
call 'a darc(1, 2, w2);  call 'b darc(1, 3, w0);                      % bowl
w1 draw 4;                                                  % make end point round
call 'c exit(5, r);                                               % closing hook
hpen;  w1 draw 4..5.                                                      % stem

"Italic letter b";
call charbegin('b, 8, mc·lbowl, -mc·rbowl, ph, 0, mi[rbowl,0]);
hpen;  x1 = x2 = good1 1.5u;  x5 = good2(r - 1.5u);  x3 = x4 = .5[x2, x5];
top1y1 = h;  y2 = .5[y3, y4];  top0y3 = m + oo;  bot0y4 = -oo;  y5 = y4;
call 'a serif(1, 1, 2, -lcs);                                             % serif
w1 draw 1..2;                                                              % stem
call 'b arc(3, 2, w0);  call 'c arc(4, 2, w1);                    % left part of bowl
call 'd darc(3, 5, w2).                                          % right part of bowl
```

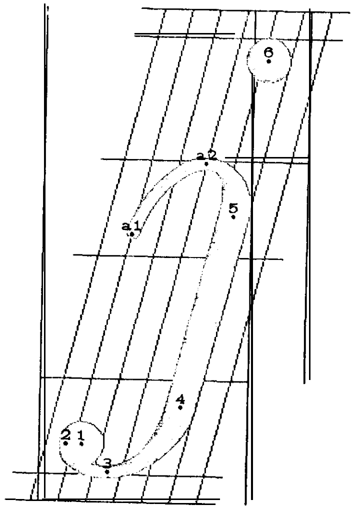


```
"Italic letter c";
call charbegin(` c, 8, mc·lbowl, —1/3 mc·px·slant, px, 0, mi[px·slant — pu, 0]);
hpen;   rt_3 x_1 = rt_0 x_2 = round(r — 1.521);   x_3 = x_5 = .5(r + u);   x_4 = good_2 1.5~;
rt_0 x_6 = r — .5u;   top_0 y_6 = .5e;   x7 = x_6;   y_7 = e;
y_1 = y_2 = .5[e, m ];   top_0 y_3 = m + oo;   bot_0 y_4 = —oo;   y_5 = y_4;
w_0 draw 2{0, 1} .. 3{—1, 0};                                         % shoulder
call ` a darc(3, 4, w_2);                                                 % bowl
w_0 draw 5{1, 0} .. 6( .. 7);                                             % point
cpen;   w_3 draw 1.                                                       % bulb

"Italic letter d";
call max(ph·slant + .5pwi -- 2pu, rhook);
call charbegin(`d, 9, mc·lbowl, —mc·acc, ph, 0, mi[acc, 0]);
hpen; x_1 = .5[x_2, x_3]; x_2 = good_2 1.5u;   x_3 = x_4 = x_5 = good_1(r — 2.5u);
top_1 y_4 = h;   top_0 y_1 = m + oo;   bot_0 y_2 = —oo;   y_3 = y_2;
call `b darc(1, 2, w_2);   call `c darc(1, 3, w_0);                       % bowl
call `d exit(5, r);                                                % closing hook
call 'a serif(4, 1, 5, —lcs);                                             % serif
w_1 draw 4 .. 5.                                                          % stem

"Italic letter e";
call charbegin(`e, 8, mc·lbowl, —1/3 mc·px·slant, px, 0, mi[px·slant — pu, 0]);
hpen;   rt_0 x_1 = round(r — 1.5~);   x_2 = good_2 1.5u;
x_3 = x.1 = .5(r + u);   rt_0 x_5 = r — .5u;   x_6 = x_5;
y_1 = .5[e, m ];   y_2 = e;   top_0 y_3 = m + 00;   bot_0 y_4 = —oo; top_0 y_5 = .5e; y_6 = e;
w_0 draw 2{1, 0} .. 1{0, 1} .. 3{—1, 0} .. 3{—1, O};               % bar and shoulder
call 'a arc(3, 2, w_2);   call `b arc(4, 2, w_2);                         % howl
draw 4{1, 0} .. 5( .. 6).                                                 % point

"Italic letter f";
call charbegin(` f, 7, 1 — fixwidth — mc(pd·slant + pu),
                1 — fixwidth — mc(px·slant), ph, pd, ph·slant + .75pu — mi·px·slant);
cpen;   lft_3 x_1 = lft_0 x_2 = round .75u;   rt_3 x_3 = rt_0 x_4 = round(r — .75u);
x_5 = r — 2.25u;   x_6 = good_1(.5r — .5u);   x_7 = good_1(.5r + .5u);   x_8 = 2.25u;
bot_3 y_1 = —.9d;   y_2 = y_1;   y_3 = y_4;   y_5 — y_3 = y_2 — y_8;
top_6 y_5 = h + 00;   y_5 — y_6 = y_7 — y_8;   bot_6 y_7 = —1/3 d;   bot_6 y_8 = —d -- oo;
x_9 = x_6 — 2.25u — eps;   x_10 = x_6 + 1.75u + eps;   top_10 y_9 = m;   y_9 = y_10;
w_3 draw I; draw 3;                                                       % bulbs
w_10 draw 9.. 10;                                                         % bar
hpen;   draw |w_0|4{0, I}   |w_0#|5{—1, 0} |w_1#|6{x_7 — x_6, y_7 -- y_6} . .
     |w_1#|7{x_7 — x_6, y_7 — y_6} . |w_0#|8{—1, 0} . . 2{0, 1}.          % stem
```

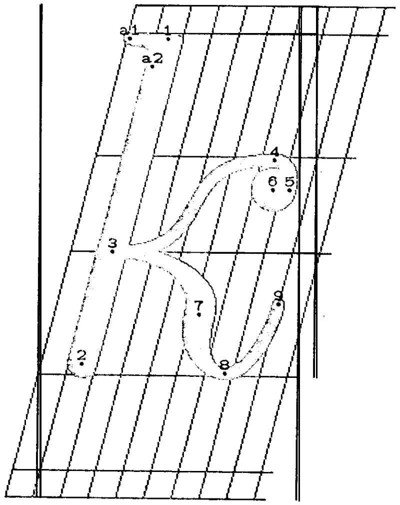


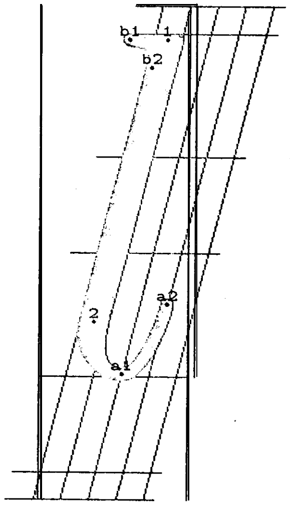


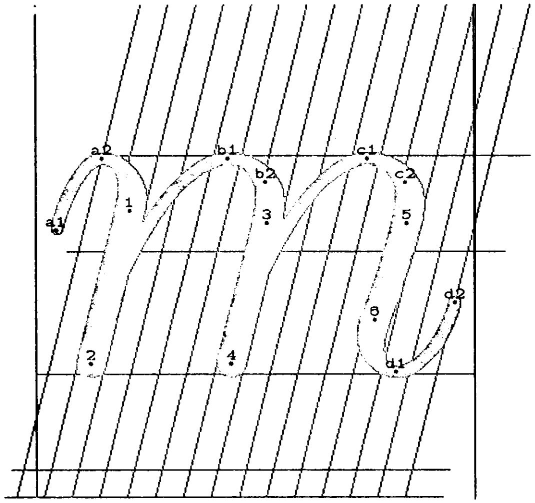


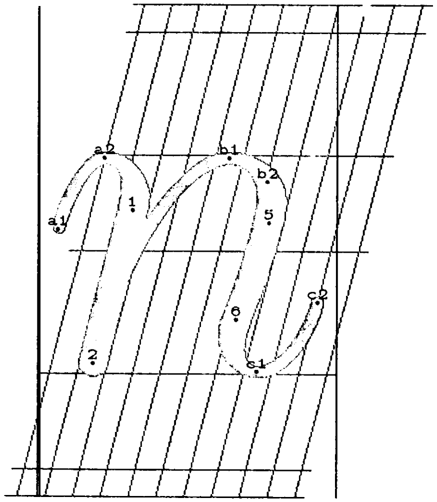


```
"Italic letter g";
call max(—lbowl, pd·slant — 1.5pu);
call charbegin(ˋ g, 8, —mc·acc, -mc( rstem — 1/3 px·slant), px, pd,
                        mi[rstem, 1/3 px·slant]);
cpen;  x1 = x6 = .5r;  x2 = good2 1.5u;  x3 = x4 = x5 = good1(r — 1.5u);
lft3x8 = lft0x7 = round 2u;
top6y1 = m + oo;  bot6y2 = —oo;  y3 = y2;  top1y4 = m — .02h;  y5 = — 1/3 d;
bot6y6 = --d — oo;  bot3y7 = —.9d;  y8 = y7;
w3 draw 8;                                                    % bulb
w1 draw 4;                                     %make end point round
call ˋa darc( I, 2, w2);                          % left part of bowl
call ˋb darc(1, 3, w0);                          % right part of bowl
hpen;  draw |w1|4..|w1#|5{0,—1}.|w0#|6{—1,0}..7{0,1}.  % stem and tail

"Italic letter h";
call charbegin(ˋ h, 9, 0, —mc·rhook, ph, 0, mi[rhook,0]);
cpen;  x1 = x2 = good, 1.5u;  x3 == good1(r — 2.5u);  x4 = x3— .25u;
top6y1 = h;  bot1y2 = —oo;
w1 draw 2;                                    % make endpoint round
call ˋa serif (1, 1, 2, —lcs);                              % serif
hpen;  w1 draw 1..2;                                         % stem
call ˋ b italhstroke(2, 3);                                % stroke
call ˋc skewexit(4, r);                              % closing hook
draw 3{0,—1}..4{—-u, -m}.                                    % link

"Italic letter i";
call max(rhook, ph·slant + .5pwiii — 2pu);
call charbegin(ˋi, 7, 1 -- fixwidth + mc·lhook, 1 — fixwidth —- mc·acc, ph, 0, mi[acc, 0]);
x1 = .5r + .25u;  x2 = .5r — .25u;
call ˋa skewentry(u, 1);                             % opening hook
call ˋ b skewexit(2, r — u);                         % closing hook
hpen; w1 draw 1..2;                                          % stem
cpcn;  top3y3 = h ; x3 == .5r; w3 draw 3.                    % clot

"Italic letter j";
call charbegin(ˋ j, 7.5, I — fixwidth -- mc(pu + pd·slant),
                        1 — fixwidth -- mc(pe·slant + .5pwi -- 1.5pu), ph, pd,
                        mi[ph·slant + .5pwi --- 1.5pu, (ph -- pe)·slant]);
cpen;  lft3x1 = lft0x2 = round .5u;  x3 = 2.5u;  x4 = xj = good1 4.5u;  rt3x6 = rt1x5;
top3y6 = h;  bot3y1 = —.9d;  y2 = y1;  bot6y3 = —d — oo;  bot6y4 = — 1/3 d;
w3 draw 1;                                                   % bulb
draw 6;                                                       % dot
call ˋ a entry(u, 5);                                % opening hook
hpen; draw |w1|5..|w1#|4{0, —1}..|w0#|3{—1, 0}..2{0, 1}.  % stern and tail
```

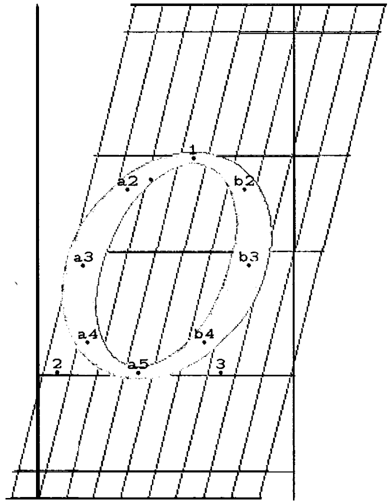


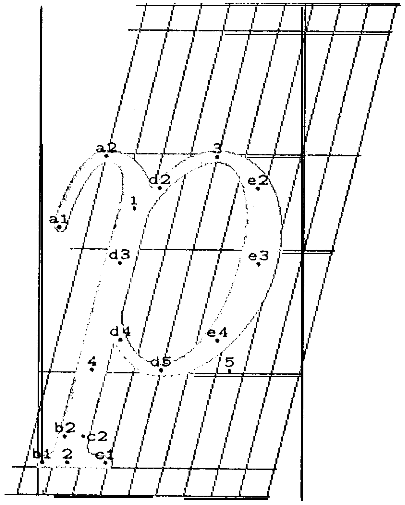


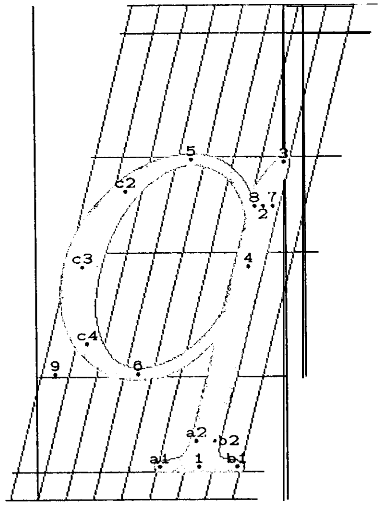




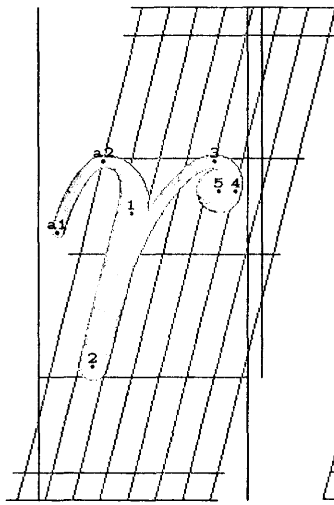
```
'Italic letter k";
call charbegin(`k, 8, 0, —mc·rhook, ph, 0, mi[px·slant, px·slant — rhook]);
cpen;  x1 = x2 = x3 = good, 1.5u;  x4 = r — 1.5~;
rt3x6 = rt0x5 = round(r — .5u);  x7 = good1(r — 2.75~);
x8 = r — 1.2571;  x9 = good0r;
top6y1 = h;  bot1y2 = -00;  y3 = good0e;  top6y4 = m + 00;  y5 = y6 = 2/3[e, m];
y7 = .5e;  bot6y8 = —oo;  y9 = 1/3m;
w1 draw 2;                                          % make end point round
w3 draw 6;                                                          % bulb
call 'a serif(1, 1, 2, —lcs);                                      % serif
hpen;  w1 draw 1..2;                                                % stem
w0 draw 3{1, 0}..4{1, 0}..5{0, —1};                      % upper diagonal
draw |w0#|3{1, 0}.|w1#|7{0, —1}..
     |w0#|8{1, 0}..9{0, 1}.              % lower diagonal and closing hook

"Italic letter I";
call max(rhook, ph·slant + .5pwi — 2pu);
call charbegin(` 1, 6, 1 — fixwidth, 1 — fixwidth -- mc·rhook, ph, 0, acc — mi·rhook);
hpen;  x1 = x2 = good, 2.5u;  top1y1 = h;
call ` a exit(2, r — u);                                    % closing hook
call ` b serif (1, 1, 2, — Jcs);                                   % serif
w1 draw 1..2.                                                       % stem

"Italic letter m";
call charbegin(`m, 15, mc·lhook, —mc·rhook, px, 0, mi[rhook, 0]);
cpcn;  x1 = x2 = good, 2.5u;  x3 = x4 = good1 .5r;  x5 — x3 = x3 — x1;  x6 = x5 — .25u;
bot1y2 = -00; y1 = y2;  w1 draw 2; draw 4;          % make end points round
call ` a entry(0, 1);                                       % opening hook
hpen; w1 draw 1.. 2;                                           % left stem
call ` b italhstroke(2, 3);                                  % left stroke
draw 3.. 4;                                                  % middle stem
call ` c italhstroke(4, 5);                                 % right stroke
call `d skewexit(6, r);                                     % closing hook
draw 5{0, —1}.6{—u, -m}.                                      % right stem

"Italic letter n";
call charbegin(`n, 10, mc·lhook, --mc·rhook, px, 0, mi[rhook, 0]);
cpen;  x1 = x2 = good, 2.511;  x5 : good1(r — 2.5u);  x6 = x5 — .25u;
bot1y2 = —oo; w1 draw 2;                             % make end point round
call ` a entry(0, I);                                       % opening hook
hpen;  w1 draw 1.. 2;                                          % left stem
call . b italhstroke(2, 5);                                       % stroke
call `c skewexit(6, r);                                     % closing hook
draw 5{0, -1)..6{—u, -m}.                                     % right stem
```

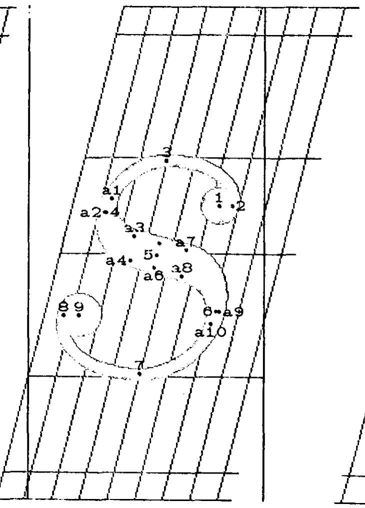


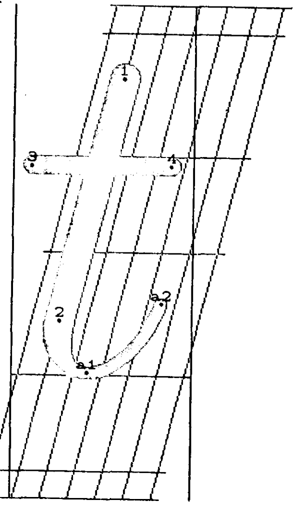




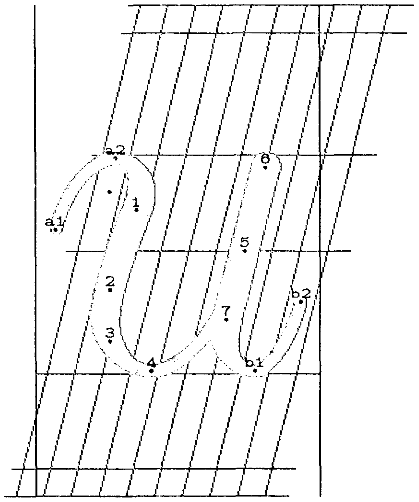
```
"Italic letter o";
call charbegin(`o, 9, mc·lbowl, —mc·rbowl, px, 0, mi[rbowl,0]);
hpen;   x1 = r — x1;                                % axis of left-right symmetry
x2 = good2 1.5u;
x1 — x2 = x3 — x1;   top0y1 = m + oo;   bot0y2 = —oo;   y2 = y3;
call `a darc(1, 2, w2);                                      % left part of bowl
call `b darc(1, 3, w2).                                      % right part of bowl

"Italic letter p";
call max(-Jhook, pd·slant + .5pwi — 2pu);
call charbegin(`p, 9, —mc·acc, —mc·rbowl, px, pd, mi[rbowl,0]);
x1 = x2 = x4 = good, 2.5u;   x3 = .5[x4, x5];   x5 = good2(r — 1.5u);
hpen; bot1y2 = —d;   top0y3 = m + oo;   bot0y4 = —oo;   y5 = y4;
call `a entry(0, 1);                                           %opening hook
w1 draw 1.2;                                                           % stem
call `b serif(2,1, 1, —.5lcs);   call `c serif(2, 1, 1, Jcs);          % serif
call `d darc(3, 4, w0);                                      % left part of bowl
call `e darc(3, 5, w2).                                      % right part of bowl

"The letter q";
call charbegin(`q, 10, mc·lbowl, 1 — fixwidth -- mc(rstem — 1/3 px·slant),
                  px, pd, mi[rstem, 1/3 px·slant]);
hpen;   x1 = good1(r — 2.5~);   x5 = .5(r — u);   x2 = x1 = x4;   bot1y1 = —d;
rt0x7 = rt1x1;   x3 = x7;   x5 = x6;   lft0x8 = lft1x2;
top0y3 = m;   y4 = .5[y5, y6];   top0y5 = m + 00;   bot0y6 = -00;
new aa;   lft1x2 = aa[x6, x4];
y2 = y7 = y8 = (sqrt(1 — aa·aa))[y4, y5];                % where bowl meets stem
if Jcs ≠ 0:   call 'a serif(1,1,2,—lcs);
    call ` b serif(1,1,2, Jcs);                                    % lower serif
fi;
w1 draw 1 .. 2;                                                        % slem
w0 ddraw 7.. 3, 8{0, 1}. . 3{x3 — x8, .5(y3 — y8)};                    % spur
w0 draw 6{1, 0} .. 4{0,1}. .5{—1, O};                      % right part of bowl
if w2 > 1.5u:   lft2x9 = round .75u;
else:   x9 = good2 1.5u;
fi;
y9 = y6;   call ` c darc(5, 9, w2).                            % left part of bowl
```

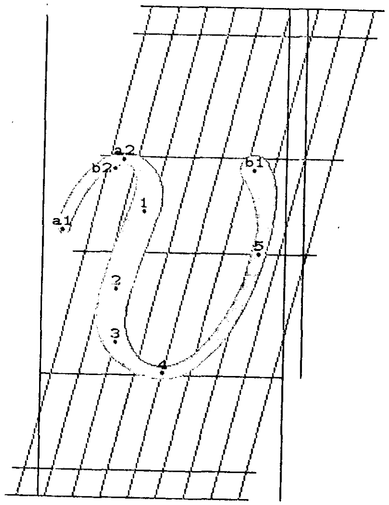


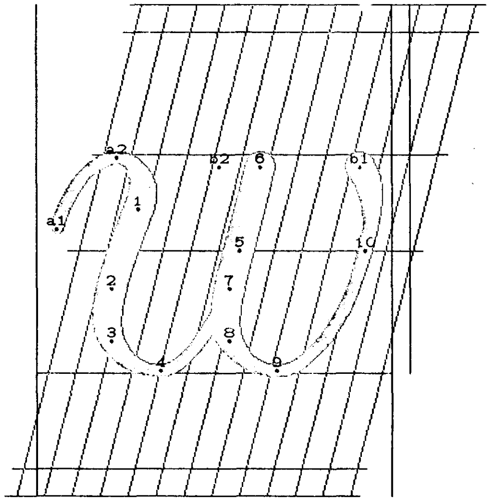




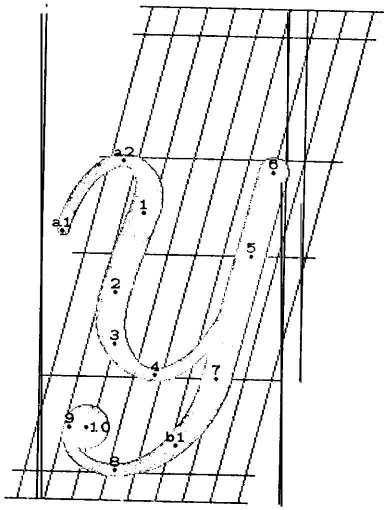
```
"Italic letter r";
call charbegin(`r, 7, mc·lhook, —mc(px·slant — .75pu),
                    px, 0, mi[px slant — .25pu, .5pu]);
cpen;  x1 = x2 = good1 2.5u;  x3 = r — 2u;  rt3x5 = rt0x4 = round(r — .75u);
bot1y2 = —oo;  top6y3 = m + 00;  y4 = y5 = 2/3 [e, m] ;
w1 draw 2;                                               % make end point round
w3 draw 5;                                                               % bulb
call 'a entry(O, I);                                             % opening hook
hpen;  w1 draw 1 .. 2;                                                   % stem
w0 draw 2{0, 1} .. 3{1, 0} .. 4{0, -1).                                % stroke

"Italic letter s";
call charbegin(`s, 7, 0, —mc(px·slant — .5pu), px, 0, mi[px·slant — .5pu, 0]);
cpcn;  rt2x1 = rt0x2 = round(r — u);  x3 = 3u;  lft11x1 = round u;
x5 = .5r;  rt11x6 = round(r — .5u);  x7 = r — 3u;  lft0x8 = lft3x9 = round .5u;
y1 = y2 = .5[e, m ];  top6y3 = m + oo;  bot6y7 = —oo;  y8 = y9 = .5e;
(y5 -- w6 — .5w8)/(m + 0 -- 2w6 — w8) = e/m;                % whitespace ratio
w2 draw 1;  w3 draw 9;                                                  % bulbs
hpen;  w0 draw 2{0, 1} . 3{—1, O};  draw 8{0, —1} .. 7{ I, 0};          % links
new aa;
if m — e > e:  aa = m - e ,
else:  aa = e;
fi;
call `a sdraw(3, 4, 5, 6, 7, w11, w8, —aa/(18u)).                      % stroke

"Italic letter t";
cpen;  call max(rhook, px·slant — .5pu + .5pw);
if px — pe < .75(ph — px):  top1y1 = 2m — e;
     call charbegin(`t, 7.5, 1 — fixwidth + mc(px·slant — pu — .5pw),
                         1 — fixwidth -- mc·acc, 2px — pe, 0, mi[acc, 0]);
else: top1y1 = .75[m, h];
     call charbegin(`t, 7.5, 1 -- fixwidth + mc(px·slant -- pu — .5pw),
                         1 — fixwidth — mc acc, .75[px, ph], 0, mi[acc, 0]);
fi;
x1 = x2 = good1 3u;  x3 = x1 — 2.5u — eps;  x4 = x1 + 2.5u + eps;
w1 draw 1;                                               % make end point round
top10y3 = m ; y4 = y3; w10 draw 3 .. 4;                                   % bar
call `a exit(2, r — u);                                          % closing hook
hpen;  w1 draw 1 .. 2.                                                   % stem
```

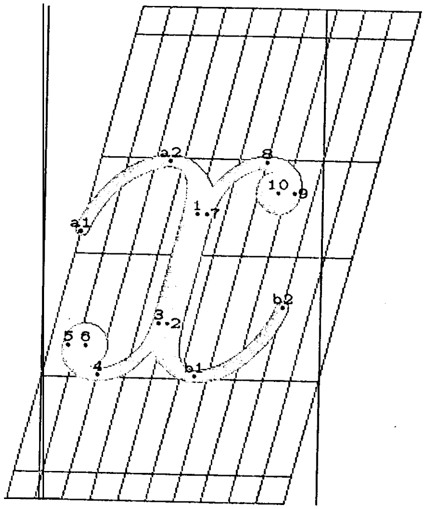


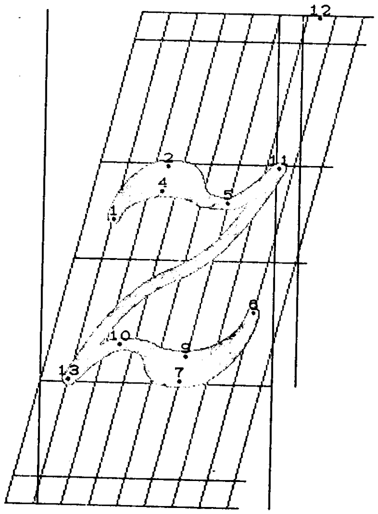




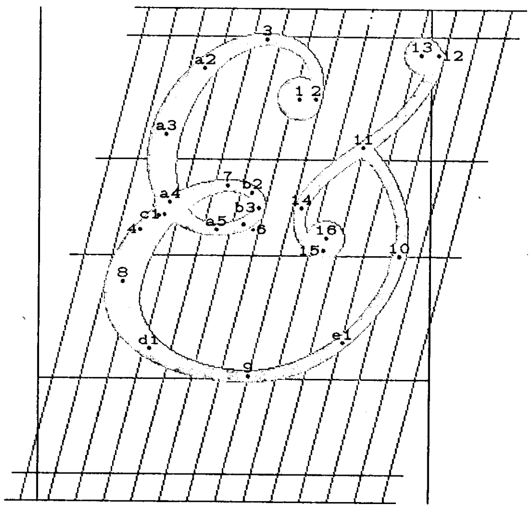
```
"Italic letter u";
call charbegin(`u, 9.5, mc·lhook, —mc·rhook, px, 0, mi[rhook, 0]);
cpen;   x1 — .25u = x2 = good1 2.5u;   x3 = x2 + .5u;   x4 = .5[x2, x5];
x5 = x6 = x7 = good1(r — 2.5u);
y2 = .7e;   y3 = .25[y4, e];   bot6y4 = —oo;   y5 = e;   top1y6 = m + oo;
w1 draw 6;                                                % make end point round
call `a skewentry(0, 1);                                        % opening hook
hpen;   draw |w1|1{—u, —m} . . |w1#|2{0, —1} . . |.75[w0, w1]|3 . .
     |w0#|4{1, 0} . . 5{0, 1};                                        % stroke
call `b exit(7, r);                                            % closing hook
w1 draw 6 . . 7.                                                        % stem

"Italic letter v";
call charbegin(`v, 8, mc·lhook, —2/3mc·px·slant, px, 0, mi[px·slant, 1/3px·slant]);
x2 = good1 2.5u;   x1 = x2 + .25u;   x3 = x2 + .5u;
call `a skewentry(0, 1);                                        % opening hook
hpen;   y2 = .7[y4, e];   y3 = .25[y4, e];   bot0y4 = —oo;   x4 = 5u;
rt0x5 = round(r — .5u);   y5 = e;
draw |w1|1{—u, —m} . . |w1#|2{0, —1} . . |.75[w0, w1]|3 . .
     |w0#|4{1, 0} . . 5{0, 1};                                        % stroke
call `b endv(5).                                              % closing bulb

"Italic letter w";
call charbegin(`w, 12, mc·lhook, —2/3mc·px·slant, px, 0, mi[px·slant, 1/3px·slant]);
cpen;   x1 — .25u = x2 = good1 2.5u;   x3 = x2 + .5u;   x4 = .6[x2, x5];
x5 = x6 = x7 = good1 6.75u;   x8 = x7 + .5u;   x9 = r — 2.75u;   rt0x10 = round(r — .5u);
y2 = .7e;   y3 = .25[y4, e];   bot6y4 = —oo;   y5 = e;   top1y6 = m + oo;
w1 draw 6;                                                % make end point round
y7 = y2;   y8 = y3;   y9 = y4;   y10 = y5;
call `a skewentry(0, 1);                                        % opening hook
hpen;   draw |w1|1{—u, —m} . . |w1#|2{0, —1} . . |.75[w0, w1]|3 . .
     |w0#|4{1, 0} . . 5{0, 1};                                   % left stroke
draw |w1|6 . . |w1#|7{0, —1} . . |.75[w0, w1]|8 . . |w0#|9{1, 0} . . 10{0, 1};   % right stroke
call `b endv(10).                                              % closing hook
```

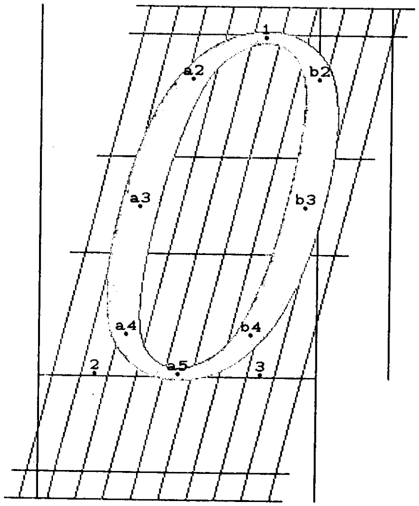






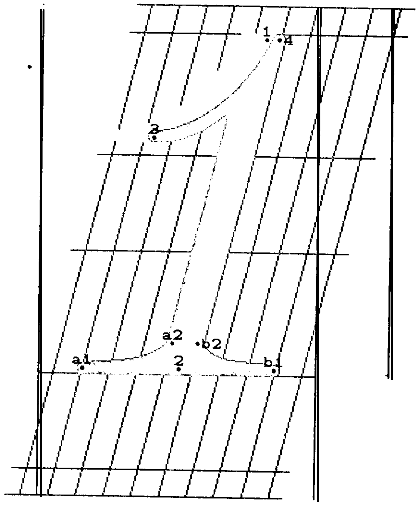
```
"Italic letter x";
if px·slant > rhook:   call charbegin(`x, 8, 0, —mc·px·slant, px, 0, mi[px·slant, 0]);
else:   call charbegin(` x, 8, mc·lhook, —mc·rhook, px, 0, mi[rhook, 0]);
fi;
cpen;   x_1 = x_2 = good, .5r;   lft_1x_2 = lft_0x_3;   x_4 = 2u;   lft_0x_5 = lft_3x_6 = round .5u;
rt_1x_1 = rt_0x_7;   x_8 = r — 2u;   rt_0x_9 = rt_3x_10 = round(r — .5u);
y_3 = y_2;   bot_6y_4 = —oo; y_5 = y_6;   y_7 = y_1;   top_6y_8 = m + oo;   y_9 = y_10;
y_8 — y_9 = y_5 — y_4;  y_9 = 2/3[e, m];
call `a entry(0, 1);                                                  % opening hook
call `b exit(2, r);                                                   % closing hook
w_3 draw 6;   draw 10;                                                % bulbs
hpen;   w_0 draw 3{0, —1} .. 4{—1, 0} .. 5{0, 1};                     % lower left link
draw 7{0, 1} .. 8{1, 0} .. 9{0, -1);                                  % upper right link
w_1 draw 1 .. 2.                                                      % stern

"Italic letter y";
call charbegin(`y, 8.5, mc·lhook, —mc(rstem — 1/3 px·slant),
                    px, pd, mi[rstem, 1/3 px·slant]);
cpcn;   x_1 — .25u = x_2 = good, 2.5u;   x_3 = x_2 + .5u;   x_4 = .5[x_2, x_5];
x_5 = x_6 = x_7 = good_1(r — 1.5u);   x_8 = .5r;   lft_0x_9 = lft_3x_10 = round 2u;
y_2 = .7e;   y_3 = .25[y_1, e];   bot_6y_4 = —oo;   y_5 = e;   top_1y_6 = m + oo;
w_1 draw 6;                                                           % make end point round
Y7 = 0;   bot_6y_8 = - d — oo; y_9 = y_10;   bot_3y_10 = —.75d;   w_3 draw 10;      % bulb
call `a skewentry(0, 1);                                              % opening hook
hpen; draw |w_1|1{—u, —m} .. |w_1#|2{0, —1} .. |.75[w_0, w_1]|3 ..
     |w_0#|4{1, 0} .. 5{0, 1};                                        % stroke
w_1 draw 6. . 7;                                                      % stem
call ` b arc(8, 7, w_1);                                              % tail
w_0 draw 8{—1, 0} .. 9{0, 1}.                                         % link

"Italic letter z";
call charbegin(`z, 7, 0, —mc·rhook, px, 0, px·slant + .5pw — mi·rhook);
vpen;   x_1 = good_0u;   x_2 = x_4 = 2.5u;   x_5 = 5u;
x_6 = good_0r;   x 7 = x_9 = 5u;   x_10 = 2.5u;
x_11 = x_12 = good_0(r — .5u);   x_13 = x_14 = x_1;
y_1 = .75m;   top_6y_2 = top_8y_3 = m + oo;   bot_8y_3 = bot_6y_4;   y_5 = .825m;
y_6 = 1/3 m;   bot_6y_7 = bot_8y_8 = —oo;   top_8y_8 = top_6y_9;   y_10 = .175m;
y_11 = y_2;   y_12 = y_11 + (h + b — m);   y_13 = y_7;   y_14 = y_13 — (h + b — m);
w_6 ddraw 1{0, 1} .. 2{1, 0} .. 5{1, 0}, 1{0, 1} .. 4{1, 0} .. 5{1, 0};          % upper bar
ddraw 10{1, 0} .. 9{1, 0} .. 6{0, 1}, 10{1, 0} .. 7{ I, 0} .. 6{0, 1};          % lower bar
draw 5{1, 0} .. 11( 12);   draw (14 )13 .. 10{1, O};                             % links
draw 11{x_13 — x_11, 2(y_13 — y_11)) 13{x_13 — x_11, 2(y_13 — y_11)}.            % diagonal
```

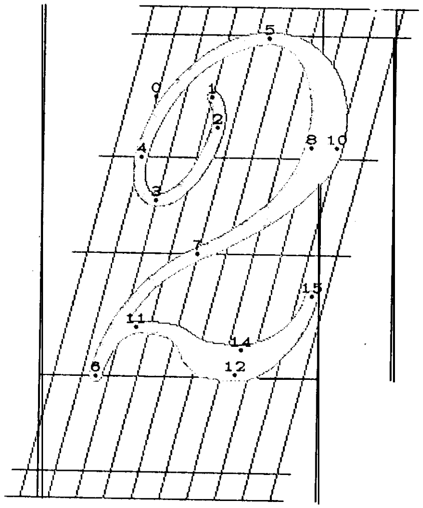


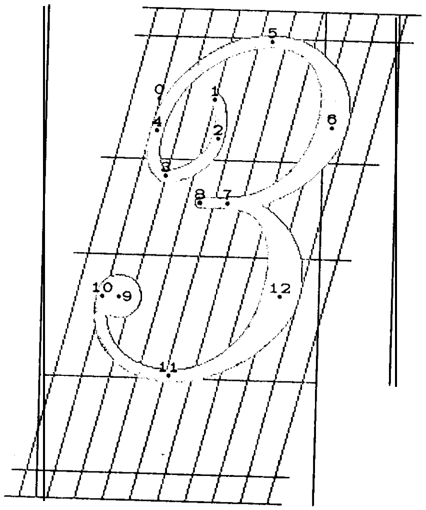


The file itald.mf

```
% This file contains the ten digits, as well as '&' and '?',
% in the so-called italic style.
% Codes '046, '060-'071, and '077 are used.

"Italic ampersand";
call charbegin( '046, 13, 0, 0, ph, 0, 0);
hpen;   rt_3 x_1 = rt_0 x_2 = round .5(r + u);   y_1 = y_2 = .5[m, h];
x_3 = 4.5u;   top_0 y_3 = h + 00;   ;
lft_1 x_4 = round 1.25u;   top_0 y_4 = round 1/3[e, m];   x_7 = x_3;
rt_0 x_6 = round 6u;   y_6 = y_4;   bot_0 y_7 = round 2/3[e, m];
lft_2 x_8 = round u;   x_9 = 7u;   y_8 = .5[y_7, y_9];   bot_0 y_9 = -oo;
rt_0 x_10 = round(r - 1.5~);   y_10 = e;
w_0 draw 2{0, I}..3{-1, 0};                                     % shoulder
call `a darc(3, 4, w_1);                                     % upper bowl
call `b darc(7, 6, w_0);                                           % loop
call `c arc(7, 8, w_2);   call `d arc(9, 8, w_2);            % lower bowl
call `e arc(9, 10, w_0);                                           % link
x_11 = 9u;   y_11 = .5[e, h];
rt_3 x_13 = rt_0 x_12 = round(r - 2u);   y_12 = y_13;
rt_0 x_14 = round(r - 5.5~);   y_14 = .5[e, m];
x_15 = x_16 = r - 4.5u;
cpen;   top_2 y_13 = h;   bot_2 y_16 = bot_6 y_15 = round e;
w_3 draw 1; w_2 draw 13; draw 16;                                 % bulbs
hpen;   w_0 draw 10{0, 1}..11{2(x_11 - x_10), y_11 - y_10};        % stem
draw 12{0, -1}.11{2(x_14 - x_12), y_14 - y_12}.14{0, -1}..15{1, 0}.  % arms

"Italic numeral 0";
% (Same as in the roman font, except for spacing.)
call charbegin(`0, 9, 0, 0, ph, pd, ph·slant - .5pu);
if fixwidth = 0:   new save;   save = sqrttwo;   new sqrttwo;
    sqrttwo = sqrt(1.23114413save);           % the constant is 2^(3/10)
fi;
hpen;   x_2 = good_2 1.5u;
x_1 = r - x_1;                            % axis of left-right symmetry
x_3 = r - x_2;   top_0 y_1 = h + oo;   bot_0 y_2 = -oo;   y_3 = y_2;
call 'a darc(1, 2, w_2);   call .b darc( I, 3, w_2);               % bowl
if fixwidth = 0: new sqrttwo; sqrttwo = save;
fi.
```



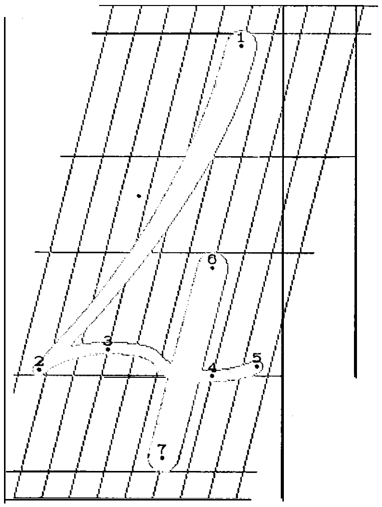


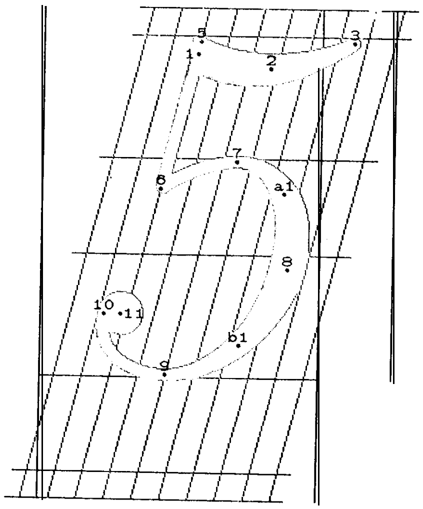


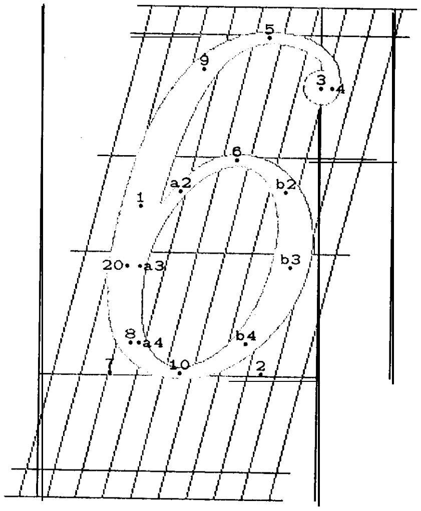
```
''ltalic numeral 1";
callcharbegin(`1,9, 0, 0, ph, pd, ph·slant—.5pu);
hpen;  x1 = x2 = good2 .5r;  top2y1 = h ;  bot2y2 = 0;
w2 draw 1..2;                                                   % stem
call ` a serif (2, 2, 1, -3);
call ` b serif(2,2,1,3);                                        % serif
top0y3 == .2[m, h];  x3 = lft2x1 — 2.5u — eps;
y4 == y1;  rt2x1 = rt0x4;  y, == I.5[m, h];  x4 == x5;
lpen#;  w2 draw (5. )4..3{ -1, O};                       %erase excess
hpcn; w0 draw (5..)4..3{—1,0}.                                 % point
```

```
"ltalic numeral 2";
call charbegin(` 2, 9,0, 0, ph,pd,ph·slant—.5pu);
hpcn;  x0 == u  ; y0 = .5[m,h]; x1 == 3u;  y1 = y0;  x2 == good0 3.5u; y2 = .25[m,h];
x3 = 2u;  bot0y3 == round .5[e,m];
x.1 ==good0u;  y4 = m;  x5 = .5r;  top0y5 == h + oo;
x9 = good2(r — 1.51,);  lft2x9 == lft0x8;  rt2x9 == rt0x10;  y8 == y10 == .5[y7, y5];
x7 == 4u;  y7 = e;  x6 == good,, 1.5u;  bot0y6 == —o;
x11 =: 2.5u;  y11 == .4e;
vpen;  bot8y13 = --oo = bot6y12;  top8y13 = top6y11;  x12 == x14 == 6.5u;
x1 j == good0(r -- .5u);  y1 j == 2/3e;
hpen;  w0 draw (0. .)1..2{0,—1}..3{—1, 0} .4{0,1}..5{1,0};      % shoulder
ddraw 5{1,0}.8{0,—1}.7{2(x7 -- x10), y7 — y10},
    5{1, 0} . . 10{0,—1}..7{2(x7 -- x10), y7 — y10};                % bowl
draw 7{2(x7 — x10), y7 — y10}.6{0,--1};                            % link
draw 6{0,1}..11{1,0};                                   % left part of bar
ddraw 11{1,0}..14{1,0}..15{0,1},11{1,0}..12{1,0}.15{0,1}.           % bar
```

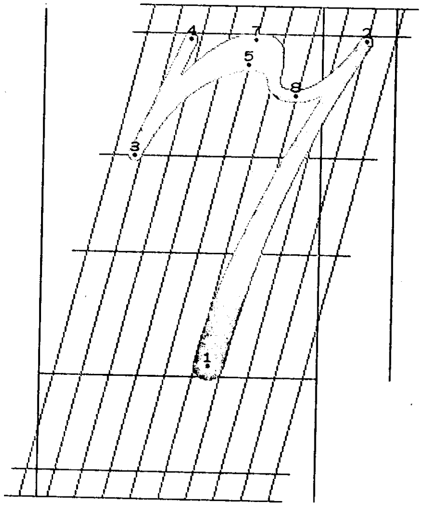
```
"Italic numeral 3";
call charbegin(`3,9,0,0, ph, pd,ph·slant—.5pu);
hpen; x0 == u;  y0 = .5[m,h]; x1 == 3u; y1 = y0;  x2 = good0 3.5u;  y2 = .5[y1, y3];
y7 == y8 = .52h;  x3 == 2u;  bot0y3 == round .125[y7,h];
lft0x4 == round u;  y4 == 1/3 [y3,y5];
x5 = .5r;  top0y5 == h+oo; rt1x6 == round(r—u); y6 = .75h; x7 == .5r; x8 == x7—u;
lft3x9 = lft0x10 == round .75u;  y9 == y10;
cpen;  top3y9 == .3h; w3 draw 9;                                   % bulb
hpen; bot0y11 , == --oo;  x11 == .5[x10, x12];  rt2x12 == round(r — u);  y12 == .25h;
w0 draw (0..)1.2{0,—1}..3{—1,0}..4{0,1}..5{1,0};               % shoulder
draw |w0#|5{1,0} . |w1#|6{0, -1). . |w0#|7{—1, 0};           % upper bowl
draw 7..8;                                                         % bar
draw |w0#|7{1,0}..|w2#|12{0,—1}..|w0#|11{—1,0};               % lower bowl
draw 11{—1,0}.10{0,1}.                                            % link
```

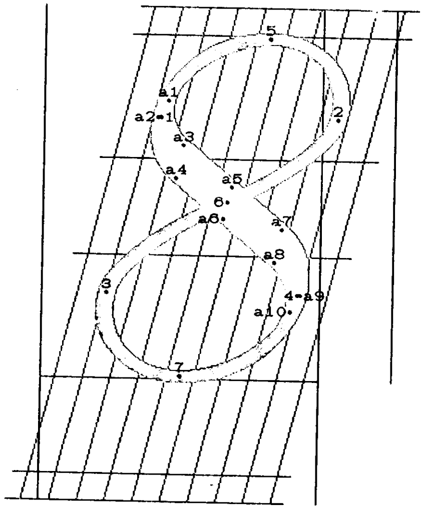




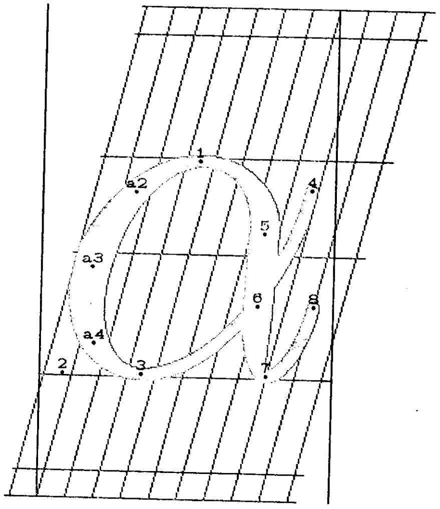


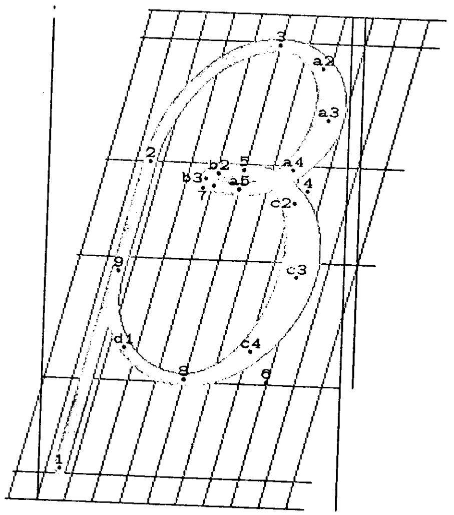
```
"Italic numeral 4";
call charbegin(`4, 9, 0, 0, $ph, pd, ph\cdot slant — .5pu$);
cpen;   top$_1y_1 = h +$ 00;   $x_1 =$ good, 5u;   lft$_0x_2 =$ round $.5u$;   bot$_6y_2 =$ 0;
$x_3 = 3u$;   $y_3 = .08h$;   $x_4 = r — 2u$;   $y_1 = 0$;   rt$_0x_5 =$ round$(r — .25u)$;   $y_5 = .03h$;
$x_6 = x_7 = \frac{2}{3}r$;   top$_1y_6$ = e;   bot$_1y_7 = -d$;
$w_1$ draw 1;                                                          % make end point round
draw 6..7;                                                                      % stem
hpen;   draw $|w_1\#|1\{0, —1\}.|w_0\#|2\{2(x_2 — x_1), y_2 — y_1\}$;                    % diagonal
$w_0$ draw $2\{2(x_1 — x_2), y_1 — y_2\}..3\{1, 0).4\{1, 0\}.5\{x_5 — x_4, 2(y_5 — y,)\}$.          % bar

"Italic numeral 5";
%(Same as in thr rornnn font, except for spacing.)
call charbegin(`5, 9, 0, 0, $ph, pd, ph\cdot slant - - .5pu$);
vpen;   $x_1 =$ good$_0 2u$;   top$_8y_1 = h$,   rt$_6x_3 =$ round$(r — 1.25u)$;   top$_6y_3 = h$;
$x_2 = .5[x_1, x_3]$;   new $w_{99}$;   $w_{99}$ = round $.75[w_6, w_8]$;   top$_{99}y_2 =$ round $.95h$;
$x_0 = —.5u$;   $x_4 = r + 1.5u$;   $y_0 = y_4 = 1.5h$;
draw $(|w_8|0.)1\ |w_{99}|2.|w_6|3(.4)$ ;                                            % arm
hpen;   $x_5 = x_6 = x_1$;   top$_0y_5 = h$;   top$_0y_6 = .75[e, m]$;
$x_7 = .5r$;   top$_0y_7 = m + oo$;   $x_9 = x_7 — .5u$;   bot$_0y_9 = —oo$;
rt$_2x_8$ = round$(r -- u)$;   $y_8 = .5[y_7, y_9]$;
$w_0$ draw 5..6;                                                                % stem
draw (9.)6..7{1,0};                                                 % left, part of bowl
call `a arc$(7, 8, w_2)$;   call `b arc$(9, 8, w_2)$;                         % right part of bowl
lft$_0x_{10}$ = lft$_3x_1$, = round $u$;   $y_{10} = y_{11} = \frac{1}{3}y_6$;
$w_0$ draw 9{—1, O}..10{0,1};                                                    % tail
cpen;   $w_3$ draw 11.                                                            % bulb

"Italic numeral 6";
%(Same as in the roman font, except for spacing.)
call charbegin(`6, 9, 0, 0, $ph, pd, ph\cdot slant — .5pu$);
$x_1 =$ good$_2$ $1.5u$;   $x_2 =$ good$_2(r — 1.5u)$;
new $w_{99}$;   $w_{99}$ = round $.75[w_0, w_3]$;
cpen;   top$_3y_3 = h — .25e$;
i f $y_3 < .5[m, h]$: new $y_3$;   $y_3 = .5[m$, h];
fi;
rt$_{99}x_3 =$ rt$_0x_1 =$ round$(r —$ 1.57~);   $y_1 = y_3$;
$w_{99}$ draw 3;                                                                % bulb
hpen;   $x_{20} =$ good$_2(x_1 + .1u)$;   $x_5 = x_6 = x_{10} = .5[x_{20}, x_2]$;   top$_0y_5 = h + oo$;
$w_0$ draw 4{0, 1}.5{—1, O};                                                 % shoulder
bot$_0y_2 = —oo$;   top$_0y_6 = m + oo$;   $y_{20} = .5[y_2, y_6]$;
$y_7 = y_2 = y_{10}$;   rt$_0x_7 =$ rt$_2x_{20}$;
call `a darc$(6, 7, w_0)$;   call `b darc$(6, 2, w_2)$;                                 % bowl
new $w_{99}$;   $w_{99} = \frac{2}{3}[w_0, w_2]$;                  % intermediate width used in darc routine
$x_8 = x_9$;   rt$_{99}x_8 =$ rt$_0$( I /sqrttwo$[x_6, x_7]$);
$y_8 = 1/$sqrttwo$[y_{20}, y_7]$;   $y_5 — y_9 = y_8 — y_7$;   $y_1 = .5[y_5, y_{10}]$;
draw $|w_0\#|5\{—1, 0\}..|w_{99}|9\{x_7 — x_6, y_7 — y_{20}\}..|w_2\#|1\{0, -1)$
    $..|w_{99}|8\{x_6 —$ x7, $y_7 — y_{20}\}..|w_0\#|10\{1, 0\}$.                              % stroke
```





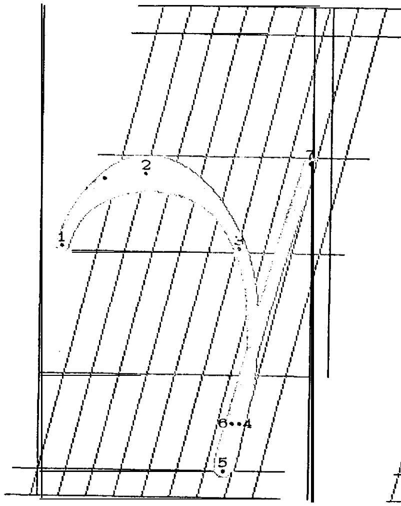
```
“Italic numeral 7”;
call $charbegin($`$7, 9, 0, 0,$ ph, $pd, ph\cdot slant —.5pu)$;
cpen;   $x_1 =$ good$_1 5.5u$;   bot$_1 y_1$ = -00;   $x_2 =$ good$_0(r — u)$;   top$_6 y_2 = h$;
$x_3 =$ good$_1 .75u$;   $y_3 = m$;   $y_8 =$ good$_1 .5[m,h]$;   $y_4 = y_7 = y_2$;
$(x_4 — x_3)/(y_4 — y_3) = (x_2 . x_1)/(y_2 — y_1)$;
top$_8 y_6$ = top$_6 y_7$;   bot$_8 y_6$ = bot$_6 y_5$;   $x_5 = x_7 = 4u$;   $x_8 = \frac{2}{3}r$;
$w_1$ draw 1;                                          % make end point round
hpen;   draw $|w_0\#|2\{2(x_1 — x_2), y_1 — y_2\} . |w_1\#|1\{0, -1\}$;     % stem
$w_0$ draw 4..3;                                                   % serif
draw $8\{1, 0\} . . 2\{2(x_2 — x_1), y_2 — y_1\}$;                      % link
ddraw $3\{x_4 — x_3, y_4 — y_3\} . 7\{ I, 0\} . . 8\{1, 0\}$,
      $3\{x_4 — x_3, y_4 — y_3\} . . 5\{1, 0\} . . 8\{1, 0\}$.            % bar

“Italic numeral 8”;
% (Same as in the roman font, except for spacing.)
call $charbegin($`$8, 9, 0, 0, ph, pd, ph\cdot slant —.5pu)$;
new $w_{98}, w_{99}, ss$;   $w_{99}$ = round $\frac{1}{3}[w_0, w_1]$;
hpen;   lft$_{99} x_1$ = round $u$;   $x_2 = r — x_1$;   $y_1 = y_2$;
lft$_{99} x_3$ = round $.75u$;   $x_4 = r — x_3$;   $y_3 = y_1$;   $x_5 = r -- x_5 = x_6 = x_7$;
top$_0 y_5 = h + oo$;   $y_6 = .52h$;   bot$_0 y_7 = —oo$;
$w_{98} = 2[w_7, w_8]$;
if $w_6 = w_{98}$:   $ss = 0$;
else:   $ss = h/(18u)$;                                             % slope
fi;
call `a $sdraw(5, 1, 6, 4, 7, w_{99}, w_{98}, —ss)$;    %upper left and lower right strokes
$w_{99}$ draw $5\{1, 0\} . 2\{0, —1\} . . 6\{—1, —.75ss\}$ . .
      $3\{0, —1\} . . 7\{1, 0\}$.                 % upper right and lower left strokes
```

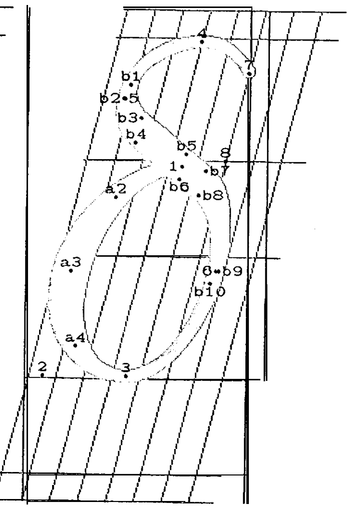


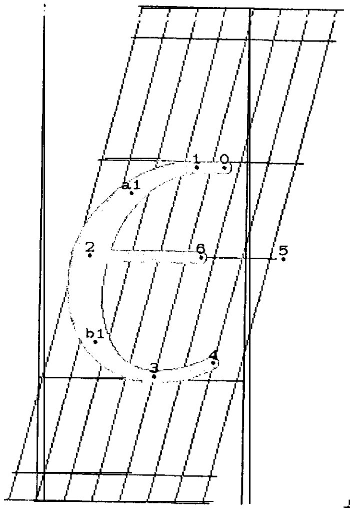


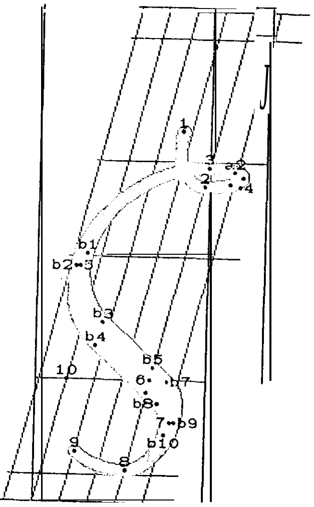
T!:e file greek1.mf

```
% This lower-case Greek alphabet was prepared by D. E. Knuth in December, 1979,
% inspired by the Monotype alphabet used in The Art of Computer Programming.
% Its spacing is intended for math formulas only.
% Character codes ´013–´037 and ´173–´177 are used.

new mc, lbowl, lhook, rbowl, rhook, rstem;          % quantities used in spacing corrections
mc = 1/pu;
rhook = 1/3 px·slant + .5pw + .5pu;
lbowl = .3px·slant — .5pwii + pu;
rbowl = .7px·slant + .5pwii -- pu;
lhook = 2/3 px·slant — .5pw — .5pu;
rstem = px·slant + .5pwi - - pu;

"Lower case Greek alpha";
call charbegin( ‘013, 10, mc·lbowl, —mc·rhook, px, 0, 0);
hpen;  x_1 = x_3 = 4.5u;  lft_2 x_2 = round u;
top_0 y_1 = m + oo;  bot_0 y_3 = —oo;  y_2 = y_3;
call ` a darc( 1, 2, w_2);                                                   % left of bowl
rt_0 x_1 = round(r — u);  top_0 y_4 = .75[e, m];
w_0 draw 3{1, 0} . . 4{0, 1};                                              % upper diagonal
x_5 = r — 2.5u;  x_6 = r — 2u;  y_5 = 2/3 m;  y_6 = 1/3 m;
bot_0 y_7 = -00;  x_7 = r — u;  x_8 = good_0 r;  y_8 = y_6;
draw |w_0#|1{1, 0} |w_1#|5{x_6 — x_5, y_6 — y_5} . .
     |w_1#|6{x_6 — x_5, y_6 — y_5} . . |w_0#|7{1, 0} . . 8{0, 1}.    % lower diagonal and tail

"Lower case Greek beta";
call charbegin( ‘0 14, 9.5, mc(pu — pd·slant — .5pw),
                    —mc·rbowl, ph, pd, .5[px, ph]·slant — pu — rbowl);
hpen;  x_1 = x_2 = x_9 = good,, 1.5u;  bot_0 y_1 = —d;  y_2 = m;
x_3 = x_5 = x_8 = 5u;  rt_1 x_4 = round(r — 1.521);
x_6 = good_2(r — 1.5u);  lft_0 x_7 = round 3.5u;
top_0 y_3 = h + oo;  bot_0 y_4 = m — w_7;  top_0 y_5 = m;  bot_0 y_6 = --oo;  y_4 = y_7;  y_6 = y_8;
y_9 = .5m;
w_0 draw 1.. 2{0,1} . . 3{ I, 0};                                     % stem and shoulder
call ‘a darc(3, 4, w_1);                                                % upper bowl
call ` b darc(5, ‘7, w_0);                                                      % loop
call ` c darc(5, 6, w_2);                                               % lower bowl
call ` d nrc(8, 9, w_0).                                                        % link
```







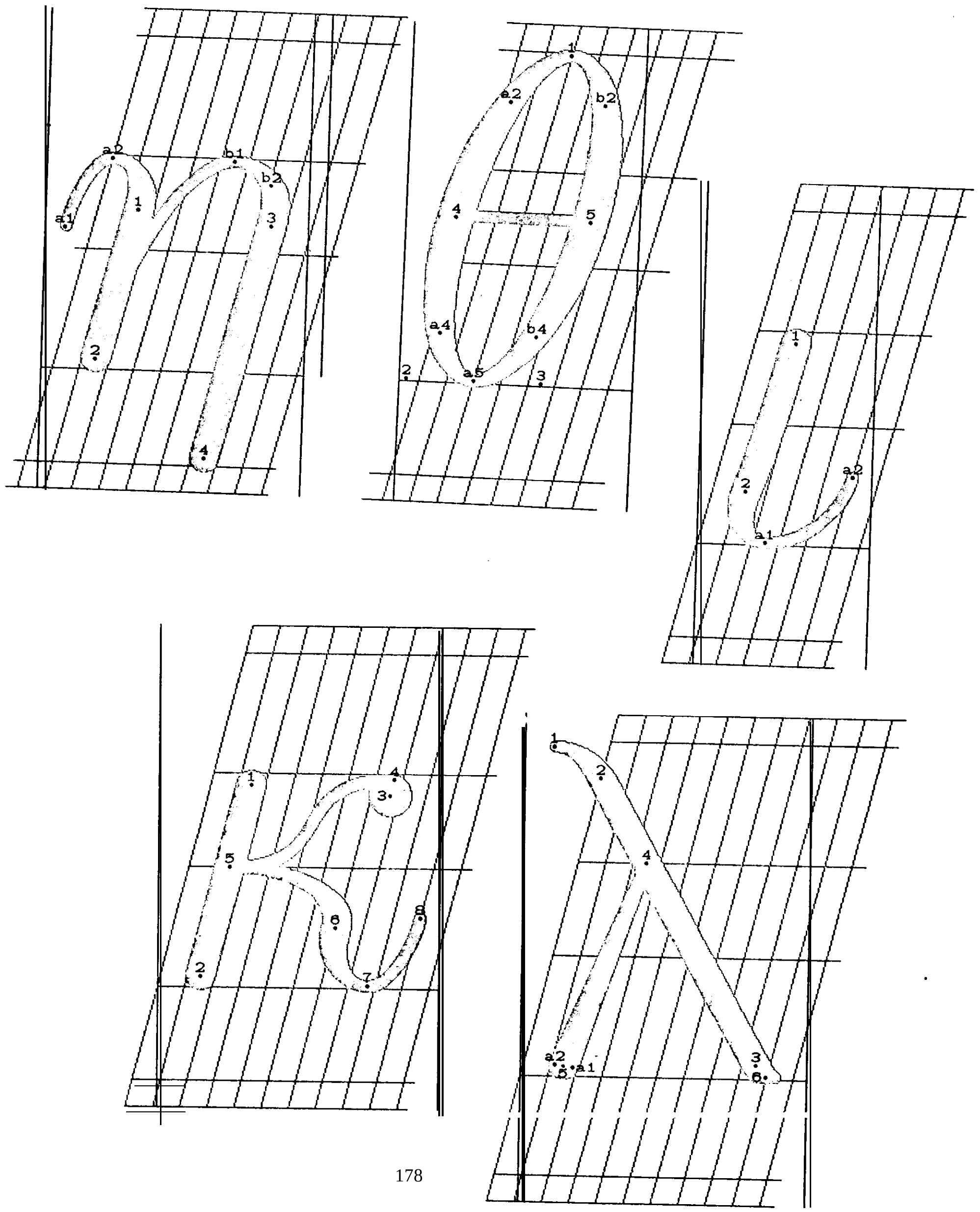


“Lower case Greek gamma”;
call *charbegin*(‘015, 10, *mc·pe·slant*, -mc($\frac{2}{3}$*px·slant* -- .5*pu*), px, pd, $\frac{1}{3}$*px·slant*);
vpen; $\mathrm{lft}_6 x_1$ == round .5*u*; $\mathrm{bot}_6 y_1$ = *e*; $\mathrm{top}_8 y_2$ == ***m*** ; x_2 = 3*u*;
x_3 = 1/*sqrttwo*[x_2, x_4]; y_3 == 1/*sqrttwo*[y_1, y_2];
$\mathrm{rt}_6 x_4$ = **round**(*r* -- *u*); y_4 = —.5*d*; x_5 == .5[x_1, x_6]; $\mathrm{bot}_6 y_5$ == —*d* — *o*;
$\mathrm{lft}_6 x_6$ == **round**(*r* — u — $\frac{1}{3}$[w_0, w_1]); y_6 = y_4; x_7 == x_4; $\mathrm{top}_6 y_7$ == m;
d r a w |w_6#|1{0, 1} . |w_8#|2{1, 0} . |w_6#|3{x_4 — x_2, y_4 - - y_2} . . 4{0, —1} .
5{—1, 0} . . 6{0, 1} . . 7{2(x_7 -- x_6), y_7 — y_6}. % stroke

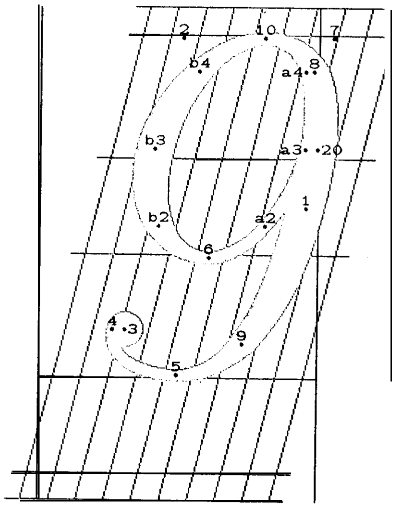
“Lower case Greek delta”;
call *charbegin*(´016, 8, *mc·lbowl*, --*mc*(.7*px·slant* — .5*pu*), *ph*, 0,
.9*ph·slant* + .5*pw* — 1.5*pu* — (7*px·slant* — .5*pu*));
hpen; x_1 == 4.5*u*; $\mathrm{top}_0 y_1$ = m ; x_2 == good$_2$ 1.5*u*; y_2 = = y_3;
x_3 == x_1; $\mathrm{bot}_0 y_3$ = --00; x_4 == 4*u*; $\mathrm{top}_0 y_4$ == *h* + 00;
$\mathrm{lft}_{11} x_5$ == **round**(1.5*u*); $\mathrm{rt}_{11} x_6$ == **round**(*r* — **u**);
x 7 = x_8 = *r* — *2u*; y_7 == .9*h*; y_8 = m;
call ˋ a darc(1, 2, w_2); % left of bowl
call ˋ **b** *sdraw*(4, 5, 1, 6, 3, w_{11}, w_8, --*h*/(18*u*)); % shoulder and right of bowl
w_0 draw 4{1, 0} . . 7(. . 8). % point

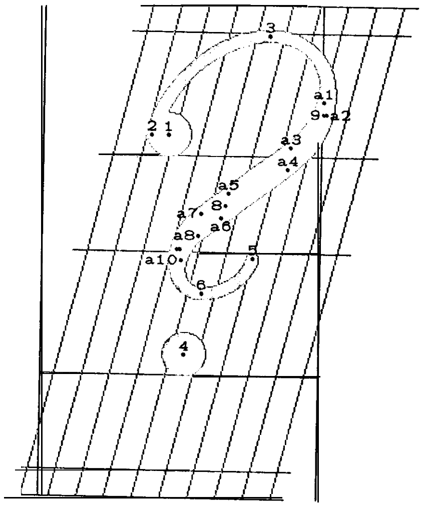
“Lower case Greek epsilon”;
call *max*(*px·slant* + .5*pw* -- 2*pu*, —.5*pu*);
call *charbegin*(‘0 17, 8, *mc·lbowl*, —*mc·acc*, px, 0, 0);
hpen; x_0 — .5*u* == x_3 == *r* — 3*u*; x_1 == .5(*r* + u); $\mathrm{top}_0 y_1$ = m; y_0 = y_1; x_6 = x_0;
x_2 == **good**$_2$ 1.5~; y_2 = y_5 = y_6 = ***e*** ; x_4 = *r* — *u*; x_5 == ***r*** + .5*u*; $\mathrm{bot}_0 y_3$ == - 0 0 ;
new *aa*; x_4 == aa[x_6, x_5]; y_4 = (sqrt(1 — aa·aa))[y_6, y_3];
w_0 draw 0.. I; % strengthen upper point
call ˋ a *arc*(1, 2, w_2); % upper left of bowl
call ˋ **b** *arc*(3, 2, w_2); % lower left of bowl
d r a w 3{1, 0} 4(. . 5); %lower point
draw 2 . . 6. % bar

“Lower case Greek zeta”;
call *charbegin*(‘020, 7, *mc·lbowl*, 0, .25[*px*, *ph*], pd, *px·slant*);
hpen; $\mathrm{lft}_0 x_1$ = **round** 3.5*u*; y_1 = .25[*m*, *h*];
x_2 = x_3 == x_6 = x_8 = 5*u*; $\mathrm{bot}_0 y_2$ = m -- w_7; y_2 == y_1; $\mathrm{top}_0 y_3$ = ***m*** **;**
$\mathrm{rt}_0 x_4$ == **round**(*r* — .5*u*); $\mathrm{lft}_{10} x_5$ -- **round** *u*; y_6 == 0, $\mathrm{rt}_{10} x_7$ == **round**(*r* — .5*u*);
$\mathrm{bot}_0 y_8$ == —*d*; x_9 == 3*u*; x_{10} == ***2u;*** y_9 == **-- *75d;*** y_{10} == ***0;***
w_0 draw 1{0, --1} . . 2{1, O}; % flourish
call ˋ a *darc*(3, 4, w_0); % loop
call ˋ **b** *sdraw*(3, 5, 6, 7, 8, w_{10}, w_8, - *e*/(6*u*)); % stroke
draw 8{--1, 0} . . 9(. . IO). % point

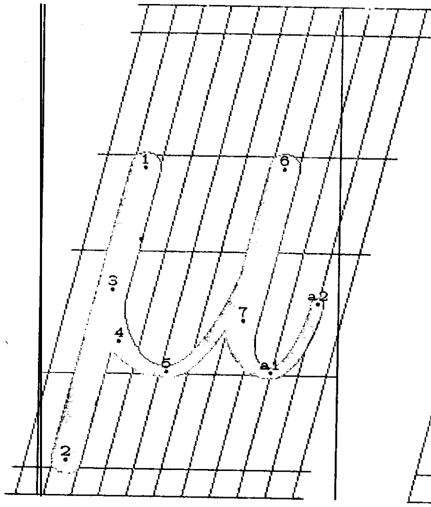


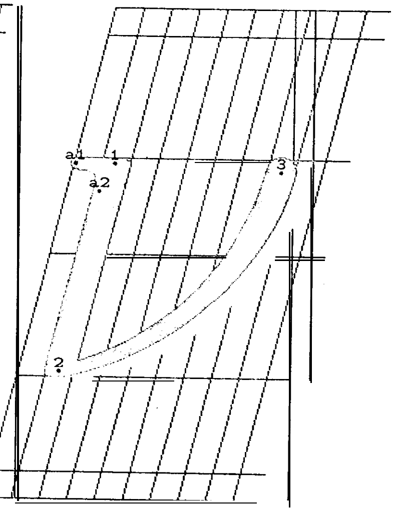
```
'Lower case Greek eta";
call charbegin( '021, 10, mc·lhook,—mc(rstem—⅔px·slant), px, pd,⅓px·slant);
xI = good, 2.5u;   call 'a entry(0,1);                                  % opening hook
cpen;  x2 = x1;  bot1y2 = —oo; w1 d r a w 1..2;                           % left stem
x3 == good 1(r — 2.5u);   call ` b italhstroke(2, 3);                       % shoulder
x4 = x3;  bot1y4 = - d — o;  draw 3.. 4.                                   % right stem

"Lower case Greek theta";
call charbegin('022, 8, mc(.3ph·slant + pu — .5pwii),
                 —mc(.7ph·slant -- pu + .5pwii), ph, 0, 0);
x1 = r — x1;                                              % axis of left-right symmetry
x2 == x4 == good2 1.5,;   x3 == x5 = r — x4;
top0y1 = h + oo;  bot0y2 = —oo;  y3 == y2;  y4 = y5 = .5[y1, y2];
call `a darc(1, 2, w2);                                            % left part of bowl
call `b darc(1, 3, w2);                                           % right part of bowl
w0 draw 4..5.                                                                    % bar

"Lower case Greek iota";
call charbegin('023, 5, 0, --mc·rhook, px, 0, 0);
cpen;  x1 = good, 1.5u;  x2 == x1 - - .25u;  top1y1 = m + oo;
call 'a skewexit(2, r);                                                % closing hook
w1 d r a w 1{0,—1}..2{—u, - m } .                                              % stem

"Lower case Greek kappa";
call charbcgin('024, 9, 0,—mc·rhook, px, 0, 0);
cpen;  x1 == x2 == x5 == good1 1.5u;  top1y1 = m + oo;  bot1y2 = —oo;
top3y3 == top6y4 = m ;  x3 == x1;  rt3x3 = round(r — 1.5u);
y5 == e;  x6 == 6u;  x7 = 7.7511;  x8 == good0r;
y6 = .5e;  bot0y7 = —oo;  y8 == ⅓m;
w1 d r a w 1..2;                                                                % stem
w3 draw 3;                                                                      % bulb
hpen;  w0 d r a w 5{1,0} 4{1, O } ;                                   % upper diagonal
draw |w0#|5{ 1, 0} . . |w1#|6{0, - 1 ) . |w0#|7{ 1, 0} .8{0, 1}.      % lower diagonal

"Lower case Greek lambda";
call charbcgin( '025, 10.5, 0, 0, ph, 0, 0);
cpcn;  top6y1 == h; x1 == - 2 U ;  x2 == 0;  y2 == .7[m, h];  x3 = r — 2u;  y3 = .125e;
ncw aa;  x4 == aa[x2, x3];  y4 == aa[y2, y3];  y4 == m;  x5 == 1.5u; bot1y5 == -oo;
hpen; x6 = r — 1 . 5 ~ ;  bot1y6 = - 0 0 ;
draw |w0#|1 {1, 0} |w1 #|2{x3 — x2, y3 — y2} 3{x3 - - x2, y3 — y2}
     6{2(x6 -- x3), y6 — y3};                                         % long diagonal
call 'a cdraw(5, 4, 1, 0).                                          % short diagonal
```

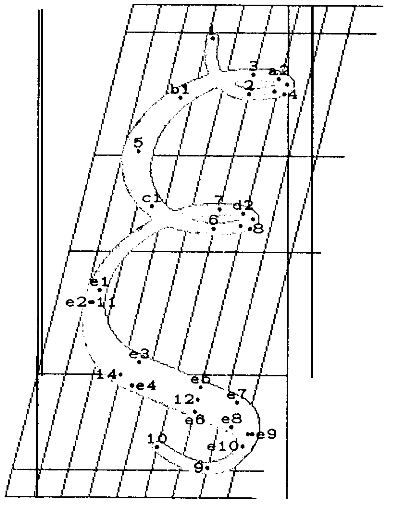


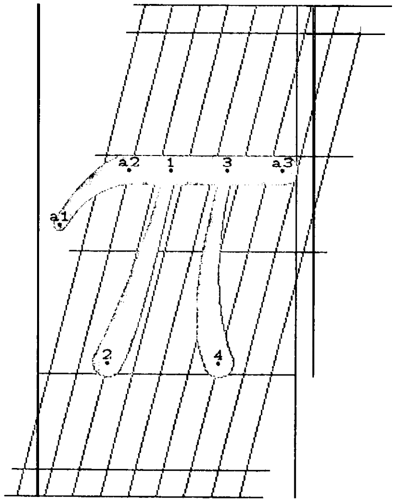


```
' Italic numeral 9";
% (Same as intheroman font, except for spacing.)
call charbegin(`9,9,0, 0,ph,pd, ph slant—.5pu);
x_1 == good_2(r — 1.5u);   x_2 = good_2 1.5u;
new w_99;   w_99 == round .75[w_0, w_3];
cpen;   bot_99 y_3 == .25e;
if y_3 > .5e:   new y_3;   y_3 == .5e;
fi;
lft_99 x_3 == lft_0 x_4 = round 1.5u;   y_4 = y_3;
w_99 draw 3;                                                          % bulb
hpcn;   x_20 == good_2(x_1 - - .1u);   x_5 == x_6 == x_10 == 5[x_20, x_2];   bot_0 y_5 == --oo;
w_0 draw 4{0,—1}..5{1,0};                                             % tail
top_0 y_2 = h + 0 0;   y_6 == e — oo;   y_20 == .5[y_2, y_6];
y_7 == y_10 == y_2;   lft_0 x_7 == lft_2 x_20;
call ` a darc(6, 7, w_0);   call ` b darc(6, 2, w_2);                   % bowl
new w_99;   w_99 == 2/3 [w_0, w_2];               % intermediate width in darc routine
x_8 = x_9;   lft_99 x_8 = lft_0(1/sqrttwo[x_6, x_7]);
y_8 == 1/sqrttwo[y_20, y_7];   y_5 - - y_9 == y_8 — y_7;   y_1 == .5[y_5, y_10];
draw |w_0#|5{1,0} . |w_99|9{x_7 -- x_6, y_7 -- y_20}|w_2#| 1{0, 1}
     |w_99|8{x_6 -  - x_7, y_7 — y_20} . |w_0#|10{—1,0}.                % stroke

"Italic question mark";
call charbegin('077, 9, 0, 0, ph, 0,0);
cpen;   lft_3 x_1 == lft_0 x_2 = round u;   y_1 == y_2;   bot _3 y_1 == m;
x_3 = x_1 = x_6 == x_8 == .5r;   top_6 y_3 = h + 0 0 ;   y_8 == .5h;   y_6 == good_6 2/3 e;
rt_10 x_9 = round(r — 1.5~);   lft_10 x_7 == round 3u;
x_5 == 2/3 r; top_6 y_5 = e; bot_3 y_4 == 0;
w_3 draw 4;                                                           % dot
draw 1;                                                               % bulb
hpen;   w_0 draw 2{0,1}3{1, O} ;                                      % shoulder
call 'a sdraw(3, Y, 8, 7, 6, w_10, w_8,(y_3— y_6)/(10u));              % stroke
draw 6{1,0}.5{0, 1).                                                  % point
```

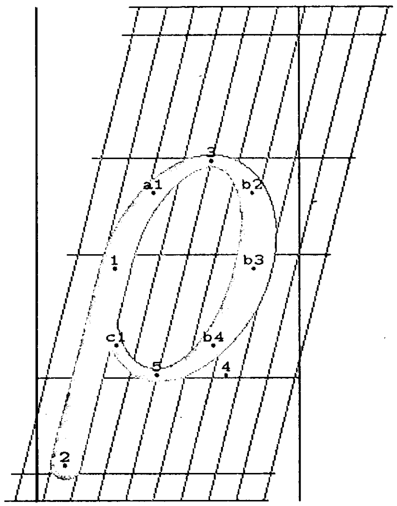


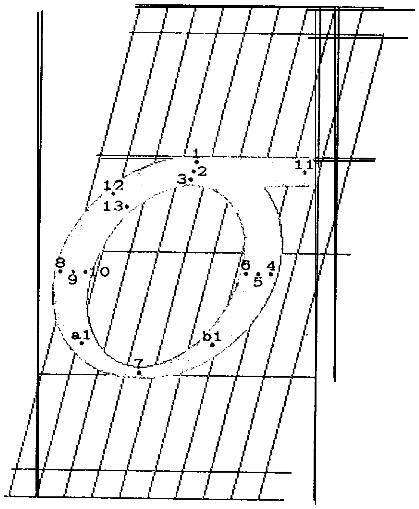


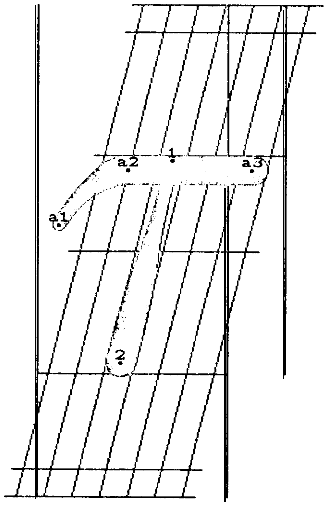


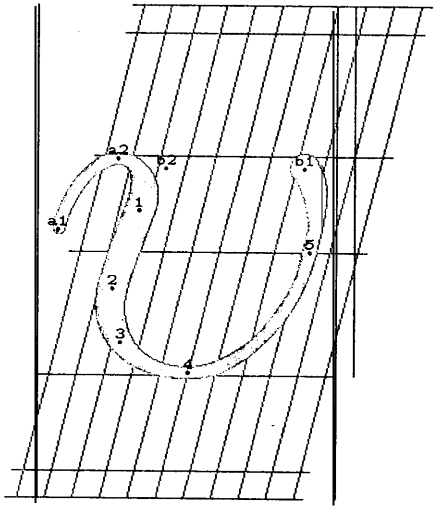


```
"Lower case Greek mu";
call charbegin(´026, 9, mc(pu — pd·slant — .5pwi), — mc·rhook, px, pd, 0);
cpen;  x1 = x2 = x3 = good, 1.5u;
x4 = x3 + .75u;  x5 = .55[x3, x6];  x6 = x7 = good1 6.5u;
top1y1 = m + 00;  bot1y2 = -  d —  o;  y3 = .7e; y4 = .25[y5, e]; bot6y5 = —oo;
y6 = y1;
call `a exit(7, r);                                              % closing hook
w1 draw 1 . . 2; draw 6. . 7;                                         % stems
hpen;  draw |w1#|3{0, -1). . |.75[w0, w1]|4 . . |w0#|5{1, 0} . . 6{0, 1}.      % stroke

"Lower case Greck nu";
call charbegin( ‘027, 9, 0, — mc(rstem — 1/3 px·slant), px, 0, 1/3 px·slant);
cpen;  x1 = x2 = good, 1.5u;  x3 = r — 1.5u;
top6y1 = m;  bot6y2 = 0;  top1y3 = m + 00;
hpen; w1 draw 1 . . 2;                                                % stem
call ‘a serif(1, 1, 2, —lcs);                                         % serif
rpen#; w1 draw 2{36u, m } . 3{0, 1};                  % erase excess at bottom
cpen;  w1 draw 3;                                                     % bulb
hpen;  draw |w0|2{36u, m}. |w1#|3{0, 1}.                           % diagonal

"Lower case Greek xi";
call charbegin( ‘030, 9, 0, 0, ph, pd, .75[px, ph]·slant -- 2pu);
hpen;  top0y1 = h;  lft0x1 = round 3u;
x2 = x3 = 5u;  bot0y2 = top0y3 — w7;  top0y3 = round(y1 -- .25(h — m));  y1 = y2;
rt0x1 = round x2 + 1.5u;  lft1x5 = round u;  y5 = .5[y3, y6];
x6 = x7 = 5u;  bot0y6 = top0y7 — w7;  top0y7 = round .5h;  y8 = y6;
rt0x8 = round x6 + 1.5u;
Ift 10x11 = round u;  x12 = 7 — 3u;  y12 = --- 1/4 d;  rt10x13 = round(r — .5u);
x9 = r — 2u;  bot0y9 = - d — oo;  x10 = x9 — 2u;  y10 = —.75d;  x1, = x10 --- 2u;
y14 = 0;
w0 draw 1{0, —1} . . 2{1, 0};                                     % flourish
call `a darc(3, 4, w0);                                          % upper loop
call `b arc(3, 5, w1);                                          % top of bowl
call `c arc(6, 5, w1);                                       % bottom of bowl
call `d darc(7, 8, w0);                                          % lower loop
call `e sdraw(7, 11, 12, 13, 9, w10, w8, -- e/(12u));                % stroke
draw 9{ —1, 0} . 10( 14).                                             % point

"Lower case Greek pi";
call charbegin(´031, 9, mc(7px·slant — .5pw -- .5pu),
                 — mc(2/3 px·slant -- pu + .5pwi·aspect), px, 0, 1/3 px·slant);
call ‘a pistroke;                                                       % bar
cpen;  bot1y2 = - 0 0 ; top7y1 = m; y3 = y1; y4 = y2;
x1 = good0 3.5u;  x2 = good, 3u;  x3 = good,, 5.5u;  x4 = good, 7u;
hpen;  draw |w0#|1{0, —1} . |w1#|2{3.14159(x2 — x1), y2 — y1};      % left stem
draw |w0#|3{0, —1} . . |w1#|4{3.14159(x4 — x3), y4 — y3};          % right stem
cpen;  w1 draw 2;  draw 4.                          % make the end points round
```

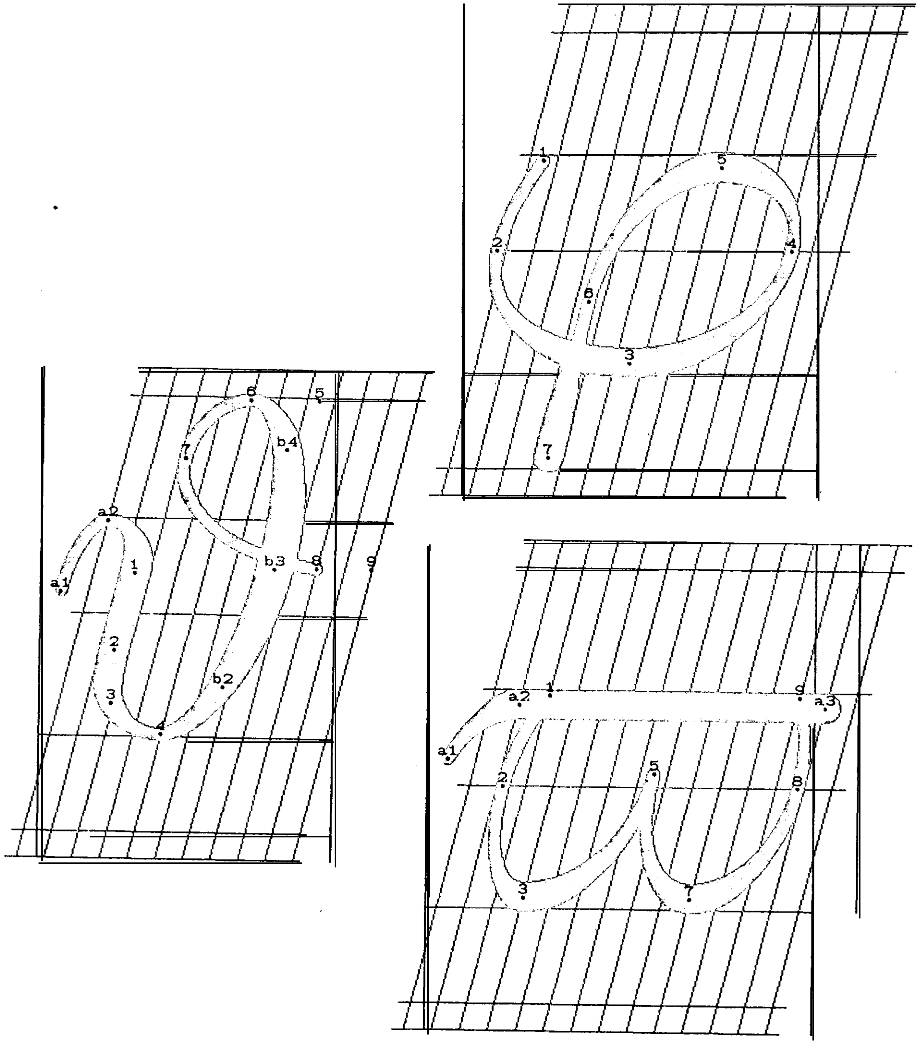




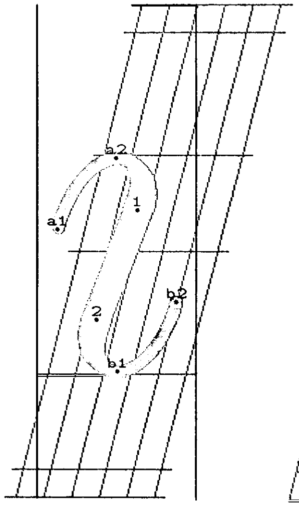


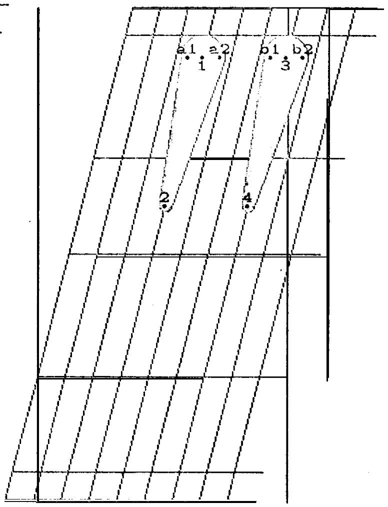


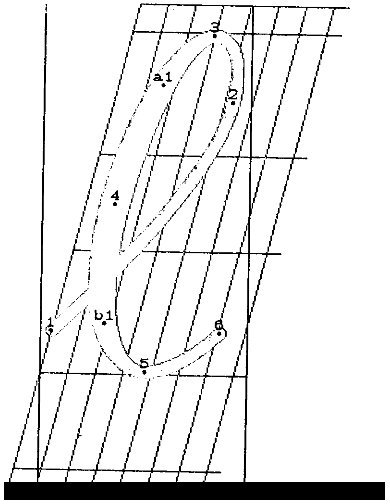
```
"Lower case Greek rho";
call charbegin('032, 8, mc(pu — pd·slant — .5pwi), —mc·rbowl, px, pd, 0);
cpen;  x1 = x2 == good, 1.5u;  x4 = good2(r -- 1.5u);  x3 = x5 == .5[x1, x4];
bot1y2 = -d — o;  y1 = .5[y3, y5];  top6y3 = m + oo;  bot6y5 = —oo;  y4 == y5;
w1 draw 2..1;                                                    % stem
call ` a arc(3, 1, w1);                           % upper left part of bowl
call `b darc(3, 4, w2);                                 % right part of bowl
call `c arc(5, 1, w0).                            % lower left part of bowl

"Lower case Greek sigma";
call charbegin('033, 10, mc·lbowl, — mc(2/3px·slant — .5pu), px, 0, 1/3px·slant);
cpen;  top7y2 == top6y1 == m ;  bot7y2 == bot6y3;  y11 == y2;
y8 == y9 = y10 = y6 = y5 = y4 = .5[y2, y7];  bot6y7 = —oo;
x1 = x2 = x3 = x7 == .5(r — u);  rt7x11 == r — u;
lft0x8 == lft2x9 = round .5u;  rt0x10 == rt2x9;
lft0x6 = lft2x5;  rt0x4 = rt2x5 = round(r — 1.5u);
x12 = 1/sqrttwo[x1, x8];  y12 == 1/sqrttwo[y8, y1];
x13 = 1/sqrttwo[x3, x10];  y13 = 1/sqrttwo[y10, y3];
w7 draw 2.. 11;                                                   % bar
hpen;  w0 ddraw 8{0, 1}.12{x1 — x8, y1 -- y8}.1{1, 0}..4{0, —1},
     10{0, 1)..13{x3 — x10, y3 — y10}..3{1, 0}..6{0, —1};  % upper part of bowl
call ‘a arc(7, 9, w2);                            % lower left part of bowl
call ` b arc(7, 5, w2).                          % lower right part of bowl

"Lower case Greek tau";
call charbegin('034, 8, mc(.7px·slant -- .5pw -- .5pu), .5,
     px, 0, px·slant — .5pu +- .5pwi·aspect);
call ‘a pistroke;                                                 % bar
cpen;  x1 = x2 = good, 3.5u;  top6y1 = m;  bot1y2 == --oo;
w1 draw 2;                                  % make lower end point round
hpen; draw |w0#|1..|w1#|2.                                       % stem

"Lower case Greek upsilon";
call charbegin( '035, IO, mc·lhook, —2/3mc·px·slant, px, 0, 1/3px·slant);
x2 = good1 2.5u;  x1 = x2 + .25u;  x3 = x2 + .8u;
call ‘a skcwentry(0, 1);                                 % opening hook
hpen; y2 = .7[y1, e]; y3 = .25[y1, e]; bot0y4 = --oo; x4 = 6u;
rt0x5 == round(r --- .5u);  y5 == e;
draw |w1|1{—u, —m}. . |w1#|2{0, -1}..|.75[w0, w1]|3..
     |w0#|4{1, 0}..5{0, 1};                                    % stroke
call `b endv(5).                                         % closing bulb
```



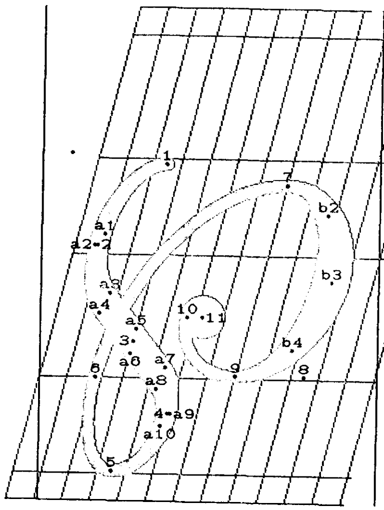
```
'Variant lower case Creek phi";
call charhegin('l75, 12, .3mc·px·slant,—.7mc·px·slant, ph, pd, 0);
hpen;  x1 = good,, 1.5u;  lft0x2 = round .5u;  x3 = .5(r + u);
rt0x4 = round(r — .5u);  x5 = 8u;  x6 = x7 = good,, 4.5~;
vpen,  top6y1 = m ;  y2 = y4 = e;  bot7y3 = -00;  top7y5 = m + 00; y6 = 1/3m;
bot1y7 = -d — oo;
draw |w6|1{2(x2 — x1), y2 — y1}. |w6#|2{0,—1}..|w7#|3{1,0}.|w6#|4{0,1}..
    |w7#|5{—1,0}..|w6#|6{0,—1};                                  % stroke
hpen; draw |w0#|6.|w1#|7;                                         % stern
cpen;  w1 draw 7.                                 % make the end point round

"Variant lower case Greek theta";
call charbegin('176, 9, mc·lhook,—mc(.5[pe,px]·slant + .5pw + .5pu), ph, 0, 0);
x1 == good, 2.5u; call `a entry(0, 1);                         % opening hook
hpen;  x2 = x1;  y2 = .7[y4, e];  x3 = x2 + .4u;  y3 = .25[y4, e];
x4 = .5[x1,x5];  bot0y4 = —oo;  x5 = good2(r — 1.5u);  top0y5 = h + oo;
x6 = x4;  y6 = y5;  lft0x7 == round 3u;  y7 = .5[m, h];
x8 = r;  y8 = y9 = good6 .5[e, m];  x9 = r + 2u;
draw |w1|1.|w1#|2{0, -1)..|.75[w0, w1]|3..|w0#|4{1,0};    % lower left part of bowl
call`b darc(4, 5, w2);                                    % right part of bowl
w0 draw 6{—1, O}..7{0,—1}..8(..9).                                % loop

"Variant lower case Greek omega";
call charbegin( '177, 14.5, mc(.7px·slant -- .5pw—.5pu), 0,
                px, 0, px·slant — pu + .5pwi·aspect);
call `a pistroke;                                                  % bar
hpen;  x1 = good0 3u;  lft0x2 = round 2u;  x3 = 4u;  x5 = good2 7.5u;
x7 = r — 4.5u;  rt0x8 = round(r -- 1.5u);  x9 = r — 2.5u;
vpen;  top6y1 = m; y2 = y8 == e; bot7y3 = —oo; y7 = y3; y9 = y1;  bot1y5 = e;
draw |w6|1{2(x2 — x1), y2 — y1}..|w6#|2{0, —1}..|w7#|3{1,0}..
    |w6#|5{0, 1};                                             % left bowl
draw |w6#|5{0, —1}.|w7#|7{1,0} |w6#|8{0, 1}.
    9{2(x9 — x8), y9 -- y8}.                                   % right bowl
```

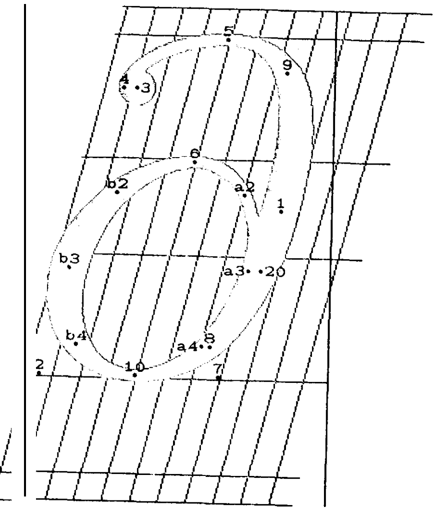


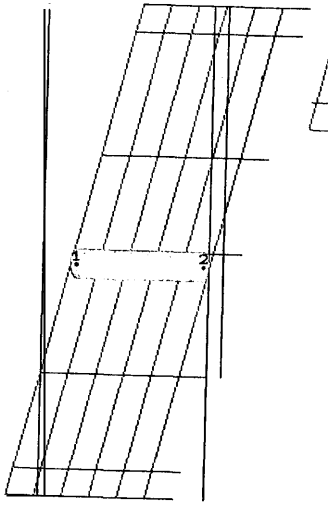




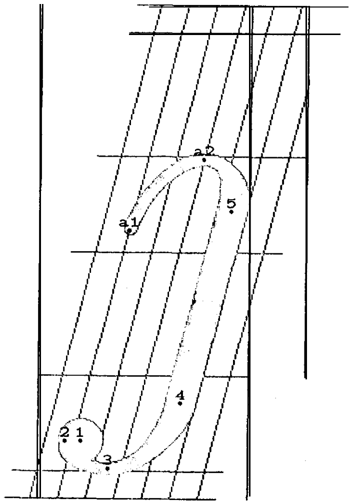
```
% This file contains miscellaneous syntbots of math italic fonts.
% Math spacing is assurned.
% The character codes are '040, '042-'045, '055, and '100.

new mc, lbowl, Jhook, rbowl,rhook,rstem;            % quantities used in spacing corrections
mc = 1/pu;
rhook = 1/3px·slant + .5pw + .5pu;
lbowl = .3px·slant — .5pwii + pu;
rbowl = .7px·slant + .5pwii — pu;
lhook = 2/3px·slant — .5pw — .5pu;
rstem = px·slant + .5pwi — pu;

"Doltcss italic letter i";
call charbegin('040, 7, 1 + mc·lhook, 1 — mc·rhook, px, 0, 0);
x1 = .5r + .25u;   x2 = .5r -- .25u;
call 'a skewentry(u, 1);                                        % opening hook
call 'b skewexit(2, r - - u);                                   % closing hook
hpen; w1 draw 1..2.                                                    % stem

"Straight double quotes";
call charbegin( '042, 9, 0, 0, ph, 0, ph·slant + .5pwiii — 2.5pu);
new w99;
if w3 < w0 sqrt 2:   w99 = round w0 sqrt 2;
else:   w99 = w3;
fi;
x1 = x2 = good99 3u;   x3 = x1 = r — x1;
cpen;   top99y1 = h; y2 = .5[e, m]; y3  =  y1; y1 = y2;
call 'a cdraw( 1, 2, 99, 0);                                       % left stem
call 'b cdraw(3, 4, 99, 0).                                        % right stem

"Lower case italic script I";
call charbegin( '043, 6, 0, —mc(.8ph·slant — pu), ph, 0, 0);
hpen;   x1 = good0 0;   rt0x2 = round(r — 1.5u);   x3 = .5r;   lft1x4 = round .5u;
x5 = r — 2.25u;   x6 = good0(r + 2);                     % (two in a row will connect)
y1 = y6 = .125h;   y2 = .8h;   top0y3 = h + oo;   y4 = .5[y3, y5];   bot0y5 = —oo;
w0 draw 1{10u,h}.2{0,1}..3{--1, 0};                                  %right of bowl
call 'a arc(3, 4, w1);   call 'b arc(5, 4, w1);                      %left of bowl
draw 5{1, 0}..6{10u, h}.                                                % link
```

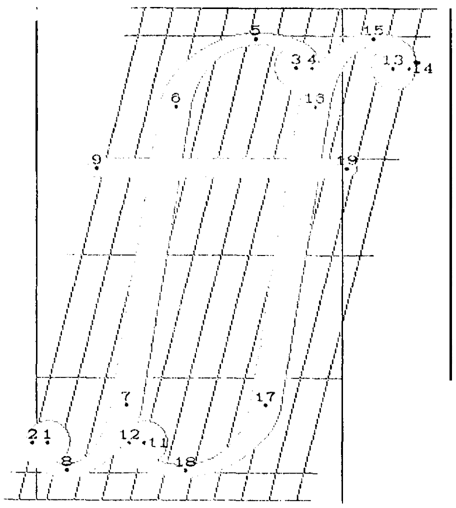


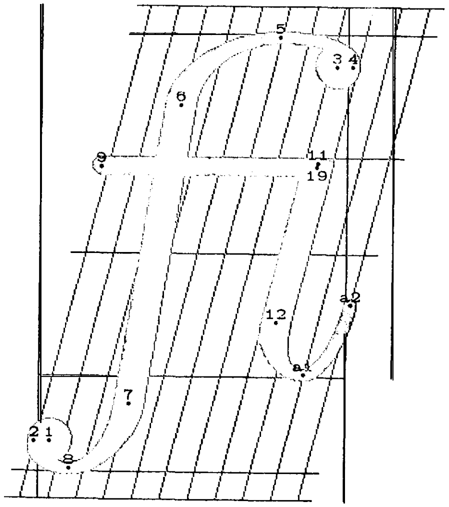




```
''Weierstrass p";
call charbegin( '044, 11, 0, —mc·rbowl, px, pd, 0);
hpen;  x1 = 2.5u;  top0y1 = m;  lft10x2 == round .5u;
x3 == 3u;  y3 = .5[e, —.5d];  rt10x4 = round 5.25u;
x5 = 3.571;  bot0y5 = -d -- oo;
call 'a sdraw(1, 2, 3, 4, 5, w10, w8, —m/(8u));                % flourish
x6 == good0 2u; y6 == 0; x7 == 7u;  top0y7 == round .8[e, m];
w0 draw 5{—1, 0} .. 6{0, 1} .. 7{1, 0};                    % stem and shoulder
x8 = good2(r — 1.521);  bot0y8 = -00;  x9 == x7;  y9 = y8;
call `b darc(7, 8, w2);                                         % howl
lft3x11 = lft0x10 == round 4.5u;  y10 = y11 = .5e;
draw 9{—!, 0} 10{0, 1};                                         % link
cpen;  w3 draw 11.                                              % bulb

"Partial differential sign";
call charbegin('045, 10, mc·lbowl, —mc(.7ph·slant + .5pwii — pu), ph, 0, 0);
% (This is a mirror-reflected number 6, one unit wider.)
x1 == good2(r — 1.5u);  x2 == good2 1.5u;
new w99;  w99 == round .75[w0, w3];
cpen;  top3y3 == h — .25e;
if y3 < .5[m, h]: new y3;  y3 == .5[m, h];
fi;
lft99x3 == lft0x4 == round 1.5u;  y4 == y3;
w99 draw 3;                                                     % bulb
hpen;  x20 = good2(x1 — .1u);  x5 == x6 == x10 == .5[x20, x2];  top0y5 = h + 00;
w0 draw 4{0, 1} .. 5{ 1, O};                                    % shoulder
bot0y2 == --oo;  top0y6 == m +· oo;  y20 == .5[y2, y6];
y7 == y2 == y10;  lft0x7 == lft2x20;
call 'a darc(6, 7, w0);  call `b darc(6, 2, w2);                % bowl
new w99;  w99 == 2/3[w0, w2];            % intermediate width used in darc routine
x8 = x9;  lft99x8 = lft0(1/sqrttwo[x6, x7]);
y8 == 1/sqrttwo[y20, y7];  y5 - y9 == y8 -- y7;  y1 == .5[y5, y10];
draw |w0#|5{1, 0} .. |w99|9{x7 -- x6, y7 - y20} .. |w2#|1{0, —1}
      .. |w99|8{x6 — x7, y7 — y20} . |w0#|10{—1, 0}.                 % stroke

"Hyphen";
call charbegin( '055, 6, 0, 0, px, 0,5px·slant — .5pu);
vpen;  y1 = y2 == .5m;
if fixwidth == 0:  lft7x1 == 0;  rt7x2 == r — u;
else: lft7x1 == 1.521; x2 == r -- x1;
fi;
w7 draw 1.. 2.                                                  % bar
```





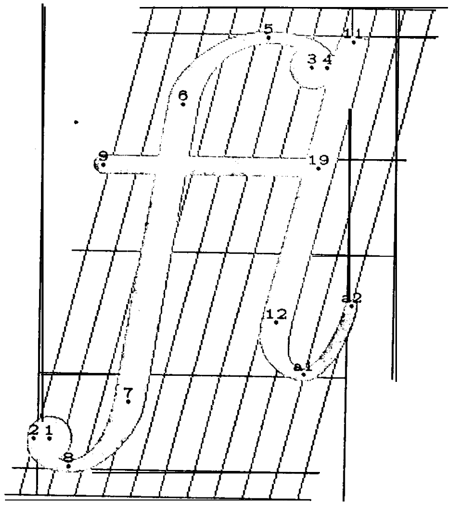


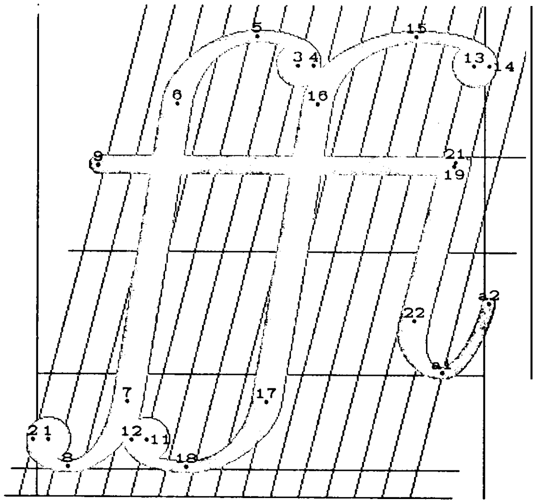
```
"Dotlcss italic letter j";
call charbegin('100, 7.5, 1 -- mc(pu + pd·slant), 1 — mc(pe·slant + .5pwi — 1.5pu),
      ph, pd, (ph -- pe)·slant);
cpen;   lft_3 x_1 == lft_0 x_2 = round .5u;   x_3 = 2.5u;   x_4 = x_5 == good_1 4.5u;
bot_3 y_1 = —.9d;   y_2 == y_1;   bot_6 y_3 = -d --- oo;   bot_6 y_4 == - 1/3 d;
w_3 draw 1;                                                              % bulb
call `a entry(u, 5);                                                     %opening hook
hpen;   draw |w_1|5 . . |w_1#|4{0, —1} . |w_0#|3{—1, 0} . . 2{0, 1}.       %stem and tail
```

The file ital ig. mf

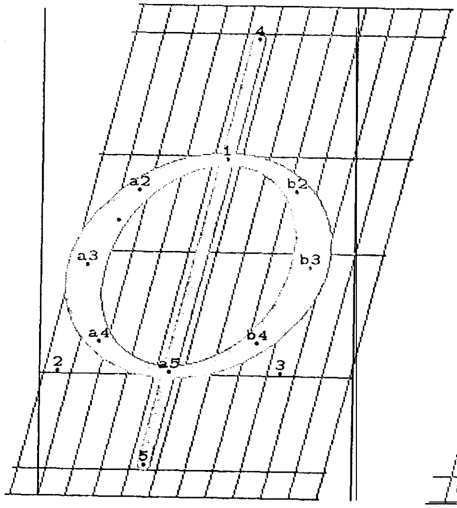
```
"Italic ligature ff";
call charbegin(' 173, 10, 0, 0, ph, pd, ph·slant + .75pu);
cpen;   lft_3 x_1 == lft_0 x_2 = round(—.25u);   rt_3 x_3 == rt_0 x_4 == round(.5r + I .75u);
x_5 == .5[x_6, x_4];   x_6 == good_1(.25r -- .5u);   x_7 = good_1(.25r + .5u);   x_8 == .5[x_2, x_7];
bot_3 y_1 = —.9d;   y_2 = y_1;   y_3 == y_4;   y_5 — y_3 = y_2 — y_8;
top_6 y_5 == h + oo;   y_5 — y_6 == y_7 -- y_8;   bot_6 y_7 == — 1/3 d;   bot_6 y_8 == --d — oo;
lft_3 x_11 == lft_0 x_12 = round(.5r -- 1.75u);   rt_3 x_13 = rt_0 x_14 = round(r + 25u);
x_15 = .5[x_16, x_14];   x_16 == good_1(.75r — .5u);   x_17 = good_1(.75r + .5u);   x_18 == .5[x_12, x_17];
!/I 1 == y_1;   y_12 == y_2;   y_13 == y_3;   y_14 = y_4;   y_15 == y_5;   y_16 == y_6;   y_17 == y_7;   y_18 == y_8;
x_9 == x_6 -- 2.25u — eps;   x_19 = x_16 + 1.75u + cps;   top_10 y_9 = m;   y_9 == y_19;
w_3 draw 1;   draw 3; draw 11; draw 13;                                   % bulbs
w_10 draw 9 . . 19;                                                       % bar
hpen;   draw |w_0|4{0, 1} |w_0#|5(--I, 0} . |w_1#|6{x_7 -- x_6, y_7 — y_6} . .
      |w_1#|7{x_7 — x_6, y_7 - - y_6} |w_0#|8{—1, 0} . 2{0, 1};             % left stem
draw |w_0|14{0, 1} . . |w_0#|15{—1, 0} . . |w_1#|16{x_17 — x_16, y_17 — y_16} . .
      |w_1#|17{x_17 — x_16, y_17 — y_16} . |w_0#|18{—1, 0} . 12{0, 1}.     % right stem

lig 'f:   `i = ' 174,  `f = ' 173,  `1 == '175;
```

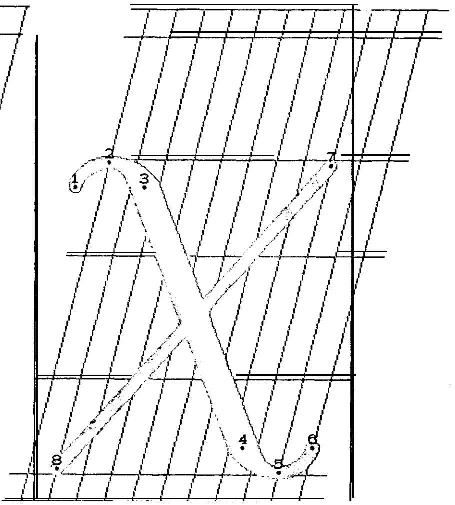
```
"Italic ligature fi";
call max(rhook, ph·slant + .5pwi — 2pu);
call charbegin( '174, 10, 0, 0, ph, pd, acc);
cpen;   lft_3 x_1 == lft_0 x_2 = round(— .25u);   rt_3 x_3 == rt_0 x_1 == rt_1 x_1 , ;
x_5 == .5[x_6, x_1];   x_6 == good_1(.25r -- .5u);   x_7 == good_1( 25r + .5u);   x_8 == .5[x_2, x_7];
bot_3 y_1 == —.9d;   y_2 == y_1;   y_3 == y_4;   y_5 -- y_3 == y_2 -- y_8;
top_6 y_5 = h + oo;   y_5 -- y_6 == y_7 — y_8;   bot_6 y_7 == — 1/3 d;   bot_6 y_8 == —d — oo;
x_11 == x_12 == good, .75r;   top_6 y_11 == m;
x_9 == x_6 — 2.25u -- eps;   top_10 y_9 == m;   x_19 == x_11;   y_19 == y_9;
w_3 draw 1; draw 3;                                                       % bulbs
w_10 draw 9.. 19;                                                         % bar
hpen;   draw |w_0|4{0, 1} . |w_0#|5{—1, 0} . . |w_1#|6{x_7 — x_6, y_7 — y_6} . .
      |w_1#|7{x_7 — x_6, y_7 — y_6} . |w_0#|8{—1, 0} . . 2{0, 1};           % left stem
call 'a exit(12, r);                                                      % closing hook
w_1 draw 11 . . 12.                                                       % right stem
```

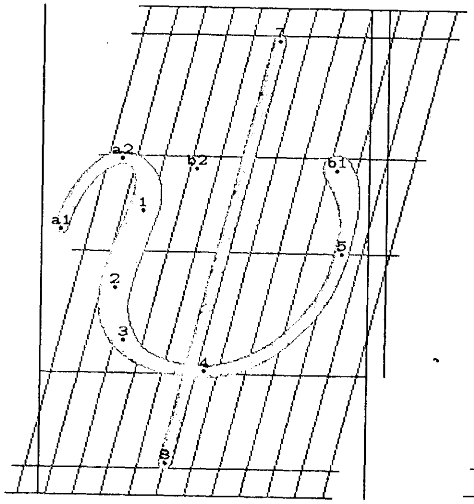


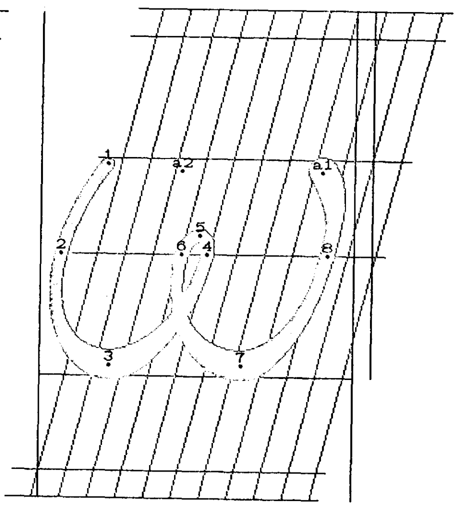


```
"Italic ligature fl";
call max(rhook, ph·slant + .5pwi — 2pu);
call charbegin('175, 10, 0, 0, ph, pd, acc);
cpcn;  lft_3 x_1 = lft_0 x_2 = round(—.25u);  rt_3 x_3 = rt_0 x_4 = round(.5r + 2u);
xj = .5[x_6, x_4];  x_6 = good_1(.25r — .5u);  x_7 = good_1(.25r + .5u);  x_8 = .5[x_2, x_7];
bot_3 y_1 = —.9d;  y_2 = y_1;  y_3 = y_4;  y_5 — y_3 = y_2 — y_8;
top_6 y_5 = h + oo;  y_5 — y_6 = y_7 — y_8;  bot_6 y_7 = —1/3 d;  bot_6 y_8 = —d — oo;
x_11 = x_12 = good, .75r;  top_6 y_11 = h;
x_9 = x_6 — 2.25u -- eps;  top_10 y_9 = m;  x_19 = x_11;  y_19 = y_9;
w_3 draw 1; draw 3;                                                   % bulbs
w_10 draw 9.. 19;                                                     % bar
hpen;  draw |w_0|4{0, 1} . |w_0#|5{—1, 0} . . |w_1#|6{x_7 — x_6, y_7 — y_6} .
    |w_1#|7{x_7 — x_6, y_7 — y_6} |w_0#|8{--1, 0} . 2{0, 1};          % left stem
call 'a exit(12, r);                                                  % closing hook
w_1 draw 11 . . 12.                                                   % right stem

"Italic ligature ffi";
call max(rhook, ph·slant + .5pwi — 2pu);
call charbegin(' 176, 15, 0, 0, ph, pd, acc);
cpen;  lft_3 x_1 = lft_0 x_2 = round(—.25u);  rt_3 x_3 = rt_0 x_4 = round(1/3 r + 1 75u);
x_5 = .5[x_6, x_4];  x_6 = good, 2u;  x_7 = good_1 3u;  x_8 = .5[x_2, x_7];
bot_3 y_1 = —.9d;  y_2 = y_1;  y_3 = y_4;  y_5 — y_3 = y_2 - - y_8;
top_6 y_5 = h + oo;  y_5 — y_6 = y_7 — y_8;  bot_6 y_7 = —1/3 d;  bot_6 y_8 = —d — oo;
lft_3 x_11 = lft_0 x_12 = round(1/3 r — 1.75u);  rt_3 x_13 = rt_0 x_14 = rt_1 x_21;
x_15 = .5[x_16, x_14];  x_16 = good_1 7u;  x_17 = good_1 8u;  x_18 = .5[x_12, x_17];
y_11 = y_1;  y_12 = y_2;  y_13 = y_3;  y_1 / = y_4; y_15 = y_5; y_16 = y_6; y_17 = y_7; y_18 = y_8;
x_21 = x_22 = good_1(r -- 2.5u);  top_6 y_21 = m;
x_9 = x_6 — 2.25u — eps;  x_19 = x_21;  top_10 y_9 = m;  y_9 = y_19;
w_3 draw 1; draw 3;  draw 11; draw 13;                               % bulbs
w_10 draw 9.. 19;                                                     % bar
hpen;  draw |w_0|4{0, 1} . |w_0#|5{—1, 0} . |w_1#|6{x_7 — x_6, y_7 — y_6} . .
    |w_1#|7{x_7 - - x_6, y_7 — y_6} . |w_0#|8{ - 1, 0} . 2{0, 1};     % left stem
draw |w_0|14{0, 1} . |w_0#|15{—1, 0 } |w_1#|16{x_17 — x_16, y_17 — y_16} .
    |w_1#|17{x_17 — x_16, y_17 — y_16} . . |w_0#|18{—1, 0} . . 12{0, 1};   % right stem
call 'a exit(22, r);                                                  % closing hook
w_1 draw 21 . . 22.                                                   % right stem

lig '173:  `i = '176, `l = '177;
```

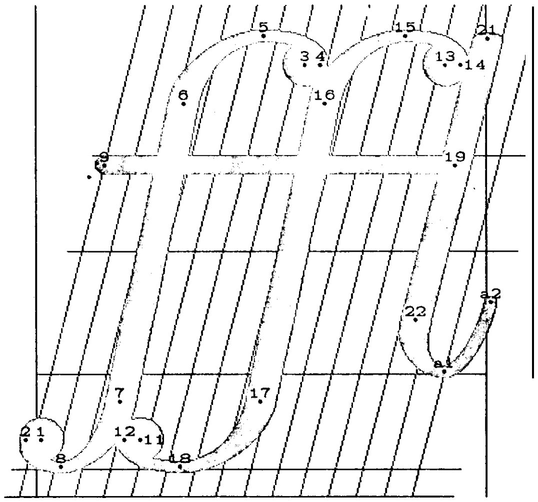


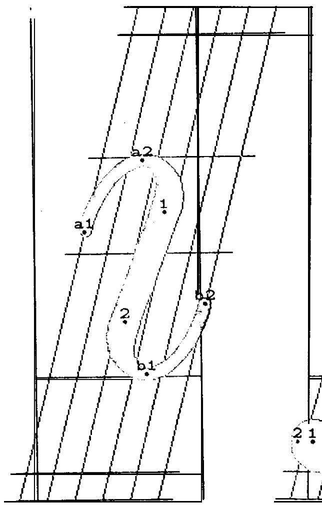


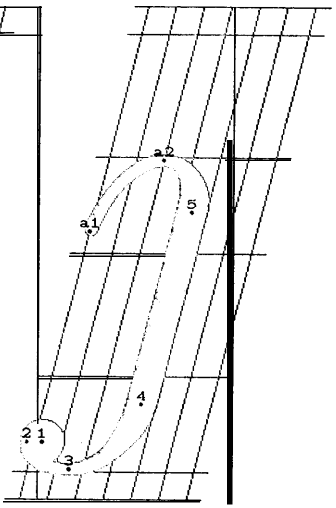




```
'.ower case  Creek  phi";
call charbegin( '036, 11, mc·lbowl,—mc·rbowl, ph, pd, 0);
x1 = r — x1;                                              % axis of left-right symmetry
x2 = good2 1.5u;   x3 = r — x2;
hpen; top0y1 = m +  0 0 ; bot0y2 =  - 0 0 ; y3 = y2;
top0y1 = h;  bot0y5 = —d;  x4 = x5 = x1;
call ` a darc(1, 2, w2);                                         %left half of bowl
call ` b darc(1, 3, w2);                                        % right half of bowl
w0 draw 4..5.                                                              % stem

"Lower case Greek chi";
if (px + pd)·slant > 2pu:
    call charbegin('037,11, mc( t .5pu — pd·slant — .5pw),
                         —mc(px slant +.5pw—1.5pu),px, pd, 0);
else:  call charbegin( '037, 1 1,mc(px·slant -- .5pw -- .5pu),
                         —mc(.5pu — pd·slant +.5pw),px,pd,0);
li;
hpen;  x1 = good0 0;  x2 = u ;  x3 == 2.5u;  x4 = r — x3;  x5 = r — x2;  x6 = r — x1;
x7 = good0(r -- 2u);  x8 = r — x7;
y1 = y3 = .875m;  top0y2 = m + oo;  bot0y5 = -d — o o;  y2 — y1 = y4 — y5;  y1 = y6;
top0y7 = m;  bot0y8 = —d;
draw |w0| 1{0, 1} . |w0#|2{ 1,  0} . |w1#|3{x4 — x3, y4 — y3} |w1#| 1{x4 - - x3, y1 — y3} .
    |w0#|5{1,0}..6{0,1};                          %upper left to lower right diagonal
w0 draw 7..8.                                     %upper right to lower left diagonal

"Lower case Greek psi";
call charbegin('173, 1 1, mc·lhook, — 2/3 mc·px·slant, ph, pd, 1/3 px·slant);
x2 = good, 2%;  x1 = x2 + 25u;  x3 = x2 + .8u;
call ` a skewentry(0,1);                                            % opening hook
hpen;  y2 = .7[y4, e]; y3 = 25[y1, e]; bot0y4 = —oo;  x4 = 6.5u;
rt0x5 = round(r — .5u);  y5 = e;
draw |w1|1{—u, —m} . |w1#|2{0, —1} |.75[w0, w1]|3 ..
    |w0#|4{1, 0} .. 5{0, 1};                                          % stroke
call ` b endv(5);                                                % closing bulb
x7 = x8 = .5(r + u);  top0y7 = h;  bot0y8 = —d;  w0 draw 7.. 8.          % stem

"Lower case Greek omega";
call charbegin(' 174, I l, mc pe slant, - 2/3 mc·px·slant, px, 0, 1/3 px slant);
hpen;  x1 = good0 1.5u;  lft0x2 = round .5u;  x3 = 3.5u;  x5 = good2 .5r;
rt0x4 = rt2x5;  lft0x6 = lft2x5;  x7 = r — 2.75u;  rt0x8 = round(r — .5u);
vpen;  top6y1 = m;  y2 = y4 = y6 = y8 = e;  bot7y3 = —oo;  y5 = good6 .2[c, m];
y7 = y3;
draw |w6|1{2(x2 — x1), y2 --- y1} .. |w6#|2{0, —1} .. |w7#|3{1, 0} .. |w6#|4{0, 1} ..
    5{—1, 0} .. |w6#|6{0,—1} .. |w7#|7{1, 0} . |w6#|8{0, 1};            % strokes
call ` a endv(8).                                                % closing bulb
```







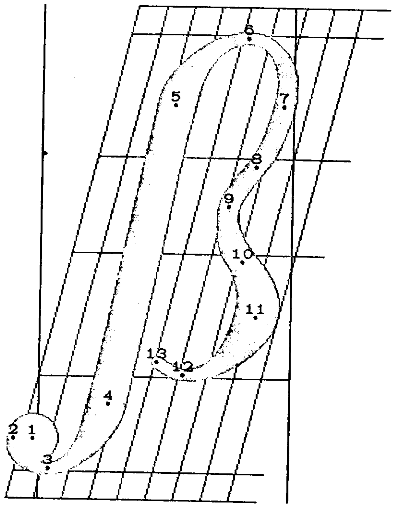
'Italic ligature ffl";
call $max(rhook, ph{\cdot}slant$ +- $.5pwi$ — $2pu$);
call $charbegin$('177, 15, 0, 0, ph, pd, acc);
cpen; lft_3x_1 = lft_0x_2 = round(—.5u); $\mathrm{rt}_3x_3 = \mathrm{rt}_0x_4$ = round($\frac{1}{3}r$ + $2u$);
$x_5 = .5[x_6, x_4]$; x_6 = good, 2.25u; x_7 = good, 2.75u; $x_8 = .5[x_2, x_7]$;
bot_3y_1 = —.9d; $y_2 = y_1$; $y_3 = y_4$; y_5 -- $y_3 = y_2 - y_8$;
top_6y_5 = IL +oo; $y_5 - y_6 = y_7 - y_8$; bot_6y_7 = —$\frac{1}{3}d$; bot_6y_8 = —d — oo;
$\mathrm{lft}_3x_{11} = \mathrm{lft}_0x_{12}$ = round($\frac{1}{3}r - 2u$); rt_3x_{13} = rt_0x_{14} = round($\frac{2}{3}r + 2u$);
$x_{15} = .5[x_{16}, x_{14}]$; x_{16} = good_1 7.25u; x_{17} = good, 7.75u; $x_{18} = .5[x_{12}, x_{17}]$;
$y_{11} = y_1$; $y_{12} = y_2$; $y_{13} = y_3$; y_1 I = y_4; $y_{15} = y_5$; $y_{16} = y_6$; $y_{17} = y_7$; $y_{18} = y_8$;
$x_{21} = x_{22}$ = good_1(r --- 2.5~); $\mathrm{top}_6y_{21} = h$;
$x_9 = x_6 - 2.25u - eps$; $x_{19} = x_{21}$; $\mathrm{top}_{10}y_9$ = m; $y_9 = y_{19}$;
w_3 **draw** 1; **draw** 3; draw 11; draw 13; % bulbs
w_{10} draw 9.. 19; % bar
hpen; draw $|w_0|4\{0, 1\}$ $|w_0\#|5\{-1, 0\}$ $|w_1\#|6\{x_7 - x_6, y_7 - y_6\}$.
$|w_1\#|7\{x_7 - x_6, y_7 - y_6\}$ . . $|w_0\#|8\{-1, 0\}$. 2{0, 1}; % left stem
draw $|w_0|14\{0, 1\}$. $|w_0\#|15\{-1, 0\}$. $|w_1\#|16\{x_{17} - x_{16}, y_{17} - y_{16}\}$.
$|w_1\#|17\{x_{17} - x_{16}, y_{17} - y_{16}\}$. $|w_0\#|18\{-1, 0\}$ . . 12{0, 1}; % right stem
call 'a exit(22, r); %closing hook
w_1 draw 21.. 22. % right stem

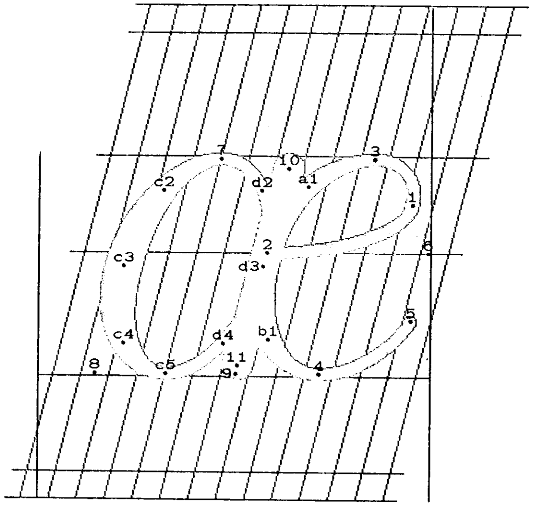
The file **itals**. mf

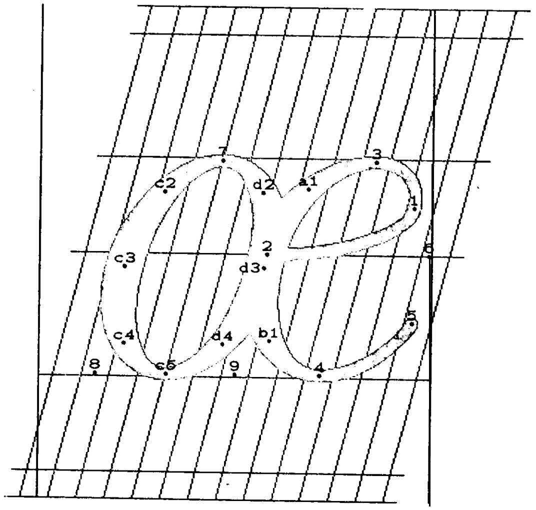
% This file contains special letters and letter combinations,
%compatible with the alphabet "itall".
%Codes '013, '014, '033 '035 are used.

"Dotless italic letter i";
call $charbegin$('0 13, 7, 1—$fixwidth$,1 --- $fixwidth, px$, 0, 0);
x_1 = .5r +- .25u; x_2 = .5r — .25u;
call 'a $skewentry(u$, 1); %opening hook
call ` b skewexit(2, r— u); % closing hook
hpcn; w_1 draw 1.. 2. % stem

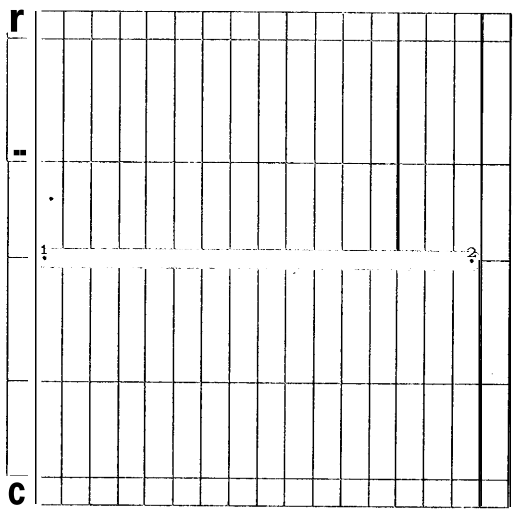
"Dotless italic letter j";
call $charbegin$('0 14, 8, 1—$fixwidth$, 1—$fixwidth$, px, pd, 0);
cpen; $\mathrm{lft}_3x_1 = \mathrm{lft}_0x_2$ = round .5u; x_3 = 2.5u; $x_4 = x_5$ = good, .5(r + u);
bot_3y_1 -- -- .9d; y_2 -- y_1; bot_6y_3 = d - 00; bot_6y_4 = — $\frac{1}{3}d$;
w_3 draw 1; % bulb
call 'a $entry(u$, 5); %opening hook
hpen; draw $|w_1|5$ . . $|w_1\#|4\{0, -1\}$ . . $|w_0\#|3\{-1, 0\}$ . . 2{0, 1}. %stem and tail

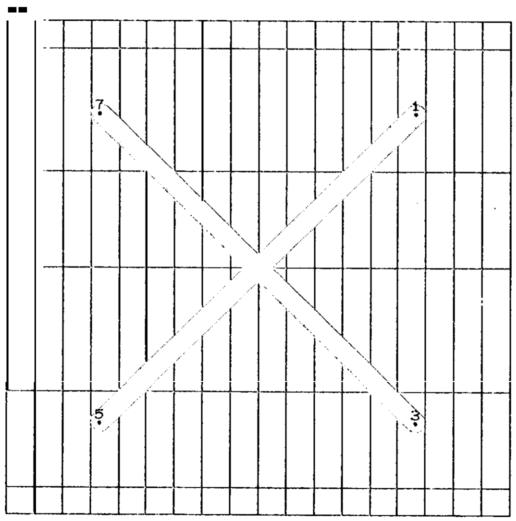


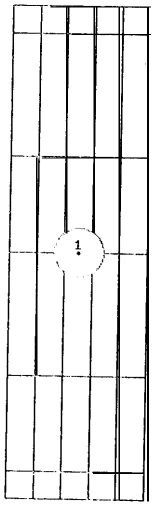




```
"Italic German letter ss";
call charbegin('033, 9, 0, 0, ph, pd, 0);
cpen;  lft_3x_1 = lft_0x_2 = round(—.5u);   x_3 = .5[x_2, x_4];
x_4 = good, 2.75u;   x_5 = good_1 2.25u;
x_6 = .5[x_5, x_7];   rt_10x_7 = round(r — 2.5~);   x_8 = .6[x_9, x_7];   rt_10x_9 = round(r — 3.521);
x_10 = .5[x_9, x_11];   rt_2x_11 = round(r — u);   x_12 = x_10 — u;   lft_0x_13 = rt_1x_4 + .5u;
bot_3y_1 = —.9d;   y_1 = y_2;   bot_6y_3 = -d — oo;   bot_6y_4 = —1/3d;   y_6 -- y_5 = y_4 — y_3;
top_6y_6 = h + oo;   y_5 = y_7;   y_8 = .6[y_7, y_9];   y_9 = .5[e, m];   y_10 = 5[y_9, y_11];
y_11 = .5e;   bot_6y_12 = —oo; y_13 = .125e;
w_3 draw 1;                                                              % bulb
hpen;  draw |w_0|2{0, —1} . . |w_0#|3{1, 0} . . |w_1#|4{x_5 — x_4, y_5 — y_4} . .
    |w_1#|5{x_5 — x_4, Y/i — y_4} . |w_0#|6{1, O} . |w_10#|7{0, -1). . 8{1.5(x_9 — x_8), y_9 -- y_8} .
    |w_10#|9{0, —1} . |.5[w_10, w_2]|10{2(x_10 — x_9), y_10 — y_9} . . |w_2#|11{0, -1). .
    |w_0#|12{—1, 0} . 13( . . 9).                                        % stroke

"Italic ligature ae";
call charbegin('034, 13, 0, 0, px, 0, 0);
hpen;  rt_0x_1 = round(r — 1.571);   x_2 = good, .5r;
x_3 = x_4 = .5(r + 6u);   rt_0x_5 = r — .5u;   x_6 = x_5;
y_1 = .5[e, m];   y_2 = e;   top_0y_3 = m + oo;   bot_0y_1 = - o o ;   top_0y_5 = .5e;   y_6 = e;
w_0 draw 2{l, O}. . l{0,l} . . 3{--1, 0} . . 3{—1, O};              % bar and shoulder
call 'a arc(3, 2, w_1);   call `b arc(4, 2, w_1);                      %right bowl
draw 4{ 1, 0} 5( .6);                                                   % point
x_7 = .5[x_8, x_9];   x_8 = good_2 l.5u;   x_9 = x_2;   y_i = y_3;   Y8 = y_9 = y_4;
call `c darc(7, 8, w_2); call `d darc(7, 9, w_0);                      % left bowl
cpen;  x_10 = x_11 = x_2;  top_1y_10 = m + oo;  bot_1y_11 = —oo;
w_1 draw 10.. 11.                                                       % stern

"Italic ligature oe":
call charbegin('035, 13, 0, 0, px, 0, 0);
h p c n ; rt_0x_1 = round(r — 1.5u); x_2 = good_2 .5r;
x_3 = x_4 = .5(r + G u);   rt_0x_5 = r -- .5u;   x_6 = x_5;
y_1 = .5[e, m];   y_2 = e;   top_0y_3 = m + oo;   bot_0y_4 = —oo; top_0y_5 = .5e; y_6 = e;
w_0 draw 2{1, 0} . . 1{0, 1} . . 3{—1, 0} . . 3{—1, 0};            % bar and shoulder
call 'a arc(3, 2, w_2);   call . b arc(4, 2, w_2);                     % right bowl
draw 4{ 1, 0} . . 5( . . 6);                                            % point
x_7 = .5[x_8, x_9];   x_8 = good, 1.5u;   x_9 = x_2;   y_7 = y_3;   y_8 = y_9 = y_4;
call `c darc(7, 8, w_2);   call . d darc(7, 9, w_2).                   %left bowl
```



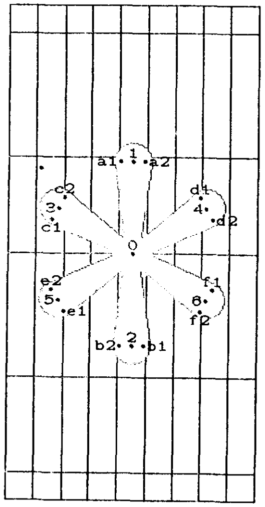


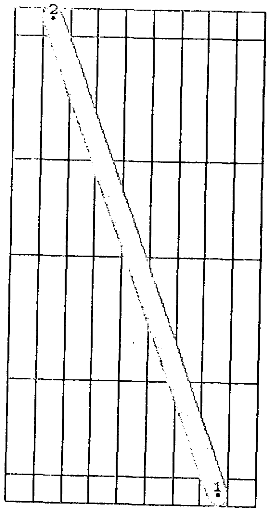


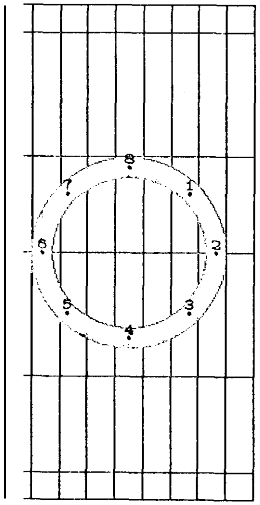
SYMBOL CHARACTER DESIGNS

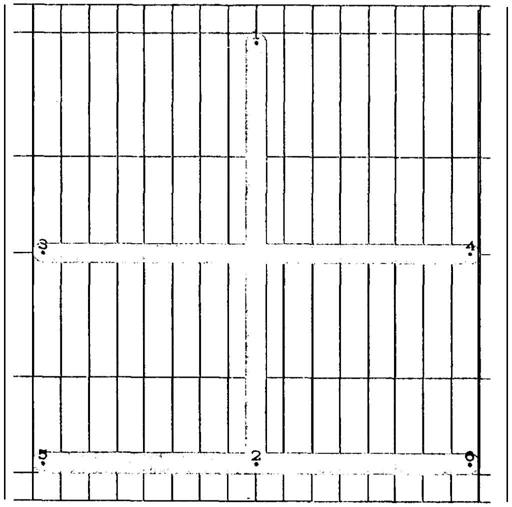
The file symbol mf

```
% The Computer Modern Symbols family of fonts (by D. E. Knuth, 1979).
danger = 0;
mi = 1;   input script;                              % upper case script alphabet
tcxinfo slant, 6pu, 3pu, 2pu, px, 18pu;
% (The calling file should give the rest of the texinfo.)
new slaut;   slant = 0; trxy 0;              % the non-script characters are unslanted

"Minus sign";
cpen;
if fixwidth = 0:   if pa |- 8pu > ph:
          call charbegin( '000, 18, 0,0, ph, ph -- 2pa, 0);
     else:   callcharbegin('000,18, 0,0, 8pu+pa,8pu-- pa,   0);
     fi;
else:   call charbegin( '000, 9, 0, 0, 3.5pu +- pa, 3.5pu  - pa, 0);
fi;

lft10x1 = round u;   x2 = r -- x1;   y1 = y2 = a;
w10 draw 1 .. 2.                                                     % bar

"Period raised to axis height";
callcharbegin( '00 1, 5, 0, 0, pa + .75pwiii, 0, 0);
cpcn;   new w99;
if w3 < w0 sqrt 2:   w99 = round u0 sqrt 2;
else:   w99 = w3;
fi;
x1 = good99 .5r;   y1 = a;   w99 draw 1.                                % dot

"Times operator";
call charbegin( '002, 18, 0, 0,1/sqrttwo[pa,ph],1/sqrttwo[pa, ph -- 2pa], 0);
cpen;   x7 = 1/sqrttwo[.5r, u];   y7 = 1/sqrttwo[a, h];
x5 = x7;   x1 = x3 = r - x7;   y1 = y7;   y3 = y5;   .5[y1, y3] = a;
w10 draw 7.. 3;                                  % upper left t o lower right diagonal
draw 5.. 1.                                      % lower left 1 o upper right diagonal
```

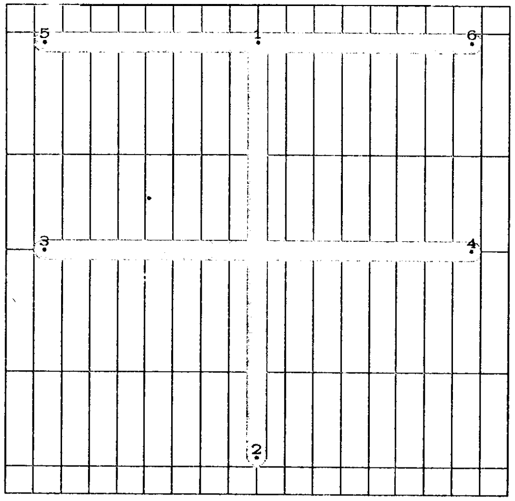


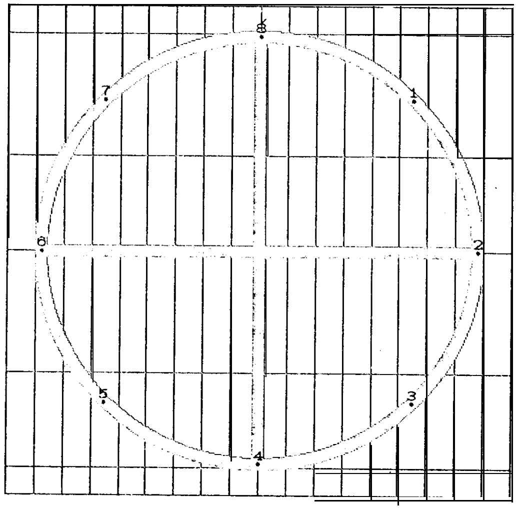


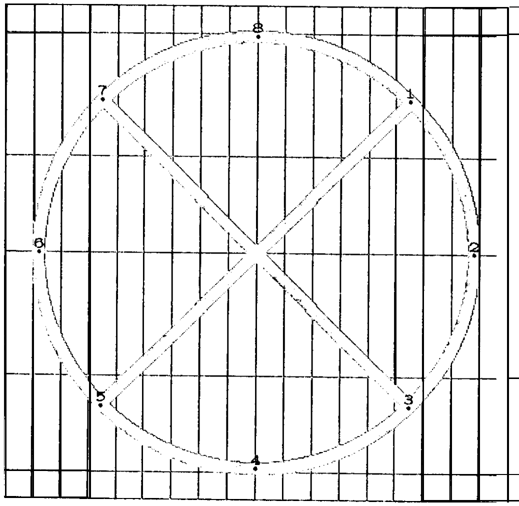


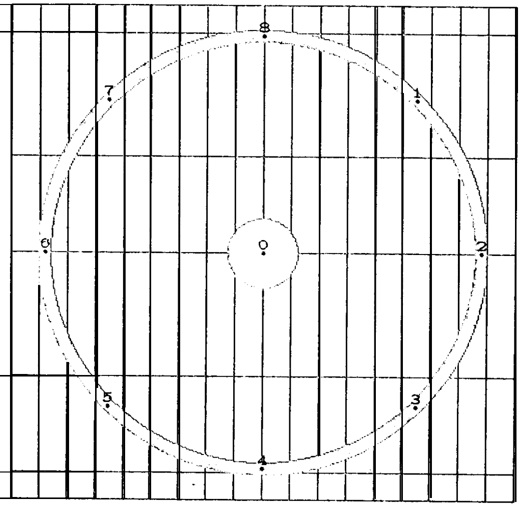


```
'Asterisk at the axis",
call charbegin('003, 9, 0, 0, .5px + pa, .5px - pa, 0);
cpen;   top_1 y_1 = round(a + .5m);   top_1 y_1 -- bot_1 y_2 = m;
y_0 = .5[y_1, y_2]; x_0 = r - x_0; x_1 = x_2 = x_0; x_3 = r - x_4; x_5 = r - x_6; x_3 =
x_5;                                                     % left-right symmetry
y_3 = y_4;   y_5 = y_6;   y_3 -- y_5 = y_1 -- y_0;   .5[y_3, y_5] = y_0;
x_4 - x_0 = (.5 sqrt 3)lft_1 3.75u;          % asterisk will have 60-degree angles if m = 7.5u
call 'a cdraw(1, 0, 1, 0);                                          % upper arm
call 'b cdraw(2, 0, 1, 0);                                          % lower arm
call 'c cdraw(3, 0, 1, 0);                                     % upper left arm
call 'd cdraw(4, 0, I, 0);                                    % upper right arm
call 'e cdraw(5, 0, 1, 0);                                     % lower left arm
call 'f cdraw(6, 0, 1, 0).                                    % lower right arm

"Reverse slash";
call charbegin('004, 9, 0, 0, ph + pb, ph + pb - 2pa, 0);
cpen;   lft_10 x_2 = round u;   x_2 = r - x_1;
top_10 y_2 = h + b;   .5[y_1, y_2] = a;
w_10 draw 1 . . 2.                                                   % diagonal

"Circle operator";
call charbegin('005, 9, 0, 0, 35pu + pa, 3.5pu - pa, 0);
cpcn;   x_8 = .5r;   lft_10 x_6 = round u;   top_10 y_8 = round(a + 3.5u);   y_6 = a;
call circle(1, 2, 3, 4, 5, 6, 7, 8, w_10).                               % bowl

"Plus or minus sign";
cpen;
if fixwidth = 0:    if pa + 8pu > ph:
            call charbegin( '006, 18, 0, 0, ph, ph - 2pa, 0);    top_10 y_1 = h;
      else:   call charbegin('006, 18, 0, 0, 8pu + pa, 8pu - pa, 0),    top_10 y_1 = a + 8u;
      fi;
else:   call charbegin( '006, 9, 0, 0, 3.5pu + pa, 3.5pu --- pa, 0);    top_10 y_1 = a --- 3.5u;
fi;
.5[y_1, y_2] = a;   x_1 = x_2 = .5r;
lft_10 x_3 = round u;   x_4 = r - x_3;   y_3 = y_4 = a;
w_10 draw 1 . . 2;                                                       % stem
draw 3 . . 4;                                                        % plus bar
x_5 = x_3; x_6 = x_4; y_5 = y_6 = y_2;    draw   5 . . 6.            % minus bar
```

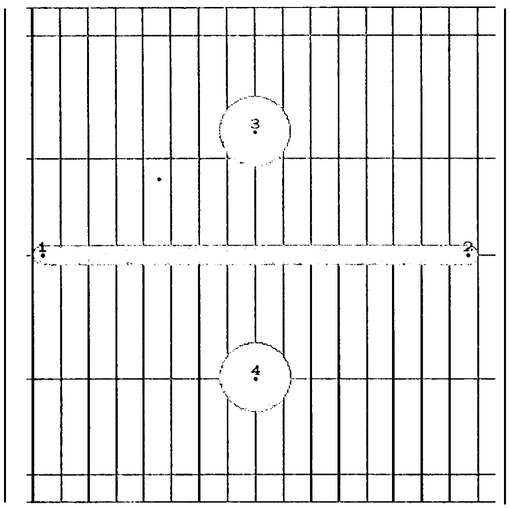


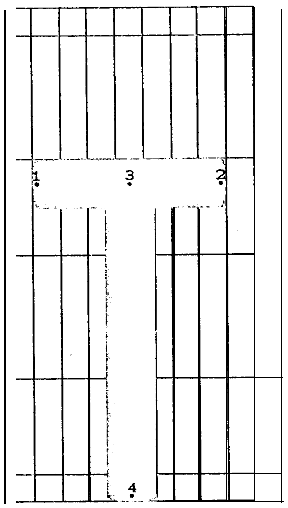


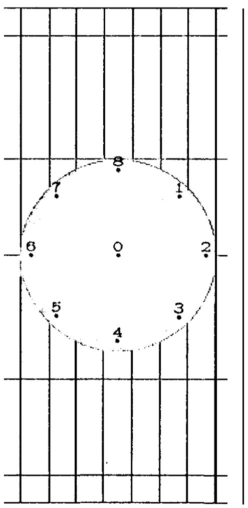


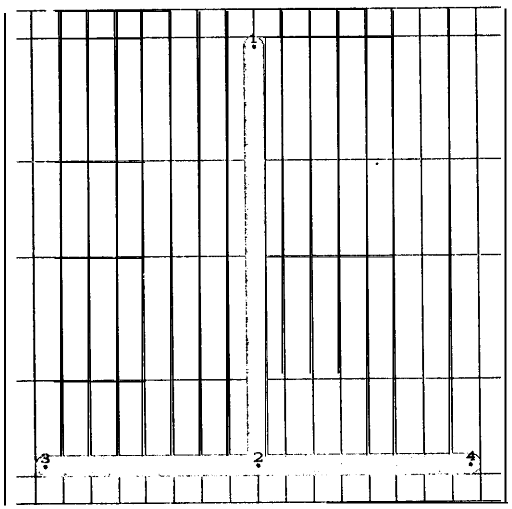


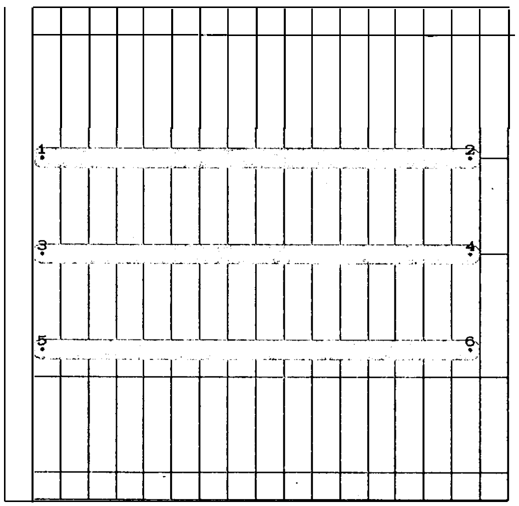
```
"Minus or plus sign";
cpen;
if fixwidth = 0:  if pa + 8pu > ph:
          call charbegin('007, 18, 0, 0, ph, ph — 2pa, 0);    top_{10}y_1 = h;
     else:  call charbegin('007, 18, 0, 0, 8pu + pa, 8pu -- pa, 0);   lop _{10}y_1 = a + 8u;
     fi;
else:  call charbegin('007, 9, 0, 0, 3.5pu + pa, 3.5pu -- pa, 0);   top_{10}y_1 = a + 3.5u;
fi;
.5[y_1, y_2] = a;   x_1 = x_2 = .5r;
lft_{10}x_3 = round u;   x_4 = r — x_3;   y_3 = y_4 = a;
w_{10} draw 1 . . 2;                                                    % stem
draw 3 . . 4;                                                       % plus bar
x_5 = x_3; x_6 = x_4;   y_5 = y_6 = y_4;   draw 5 . . 6.               % minus bar

"Circle-plus operator";
call charbegin( '010, 18, 0, 0, ph, ph - 2pa, pa·slant — .5pu);
cpen;  lft_0x_6 = round u;   y_6 = a;   x_8 = r — x_8;   top_0y_8 = h + 00;
call circle( 1, 2, 3, 4, 5, 6, 7, 8, w_0);                               % bowl
w_0 draw 6. 2;                                                           % bar
draw 4 . . 8.                                                           % stem

"Circle-minus operator";
call charbegin('011, 18, 0, 0, ph, ph — 2pa, pa·slant — .5pu);
cpen;  lft_0x_6 = round u;   y_6 = a;   x_8 = r -- x_8;   top_0y_8 = h + 00;
call circle(1, 2, 3, 4, 5, 6, 7, 8, w_0);                                % bowl
w_0 draw 6.. 2.                                                          % bar

"Circle-times operator";
call charbegin( '012, 18, 0, 0, ph, ph — 2pa, pa·slant — .5pu);
cpen;  lft_0x_6 = round u;   y_6 = a;   x_8 = r -- x_8;   top_0y_8 = h + 00;
call circle( I, 2, 3, 4, 5, 6, 7, 8, w_0);                               % bowl
w_0 draw 7.. 3;                             % upper left to lower right diagonal
draw 5 . . 1.                               % lower left to upper right diagonal

"Circle-divide operator";
call charbegin('013, 18, 0, 0, ph, ph — 2pa, pa slant — .5pu);
cpen;  lft_0x_6 = round u;   y_6 = a;   x_8 = r -- x_8;   top_0y_8 = h + 00;
call circle( I, 2, 3, 4, 5, 6, 7, 8, w_0);                               % bowl
w_0 draw 5 . 1.                             % lower left to upper right diagonal

"Circle-dot operator";
call charbegin('014, 18, 0, 0, ph, ph — 2pa, pa·slant — .5pu);
cpen;  lft_0x_6 = round u;   y_6 = a;   x_8 = r — x_8;   top_0y_8 = h + oo;
call circle(1, 2, 3, 4, 5, 6, 7, 8, w_0);                                % bowl
x_0 = x_8;   y_0 = y_2;
cpen;  new w_{99};   w_{99} = w_3 sqrt 2;   w_{99} draw 0.                     % dot
```



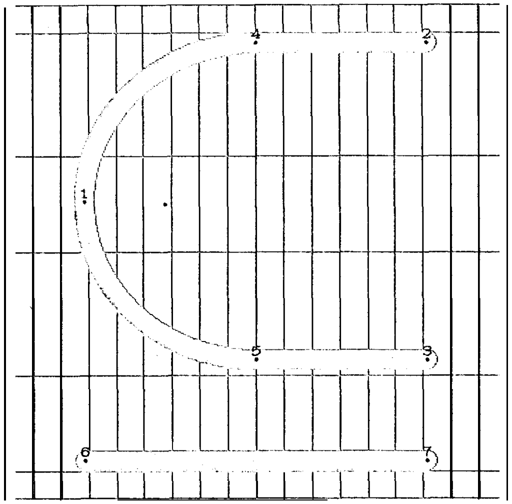


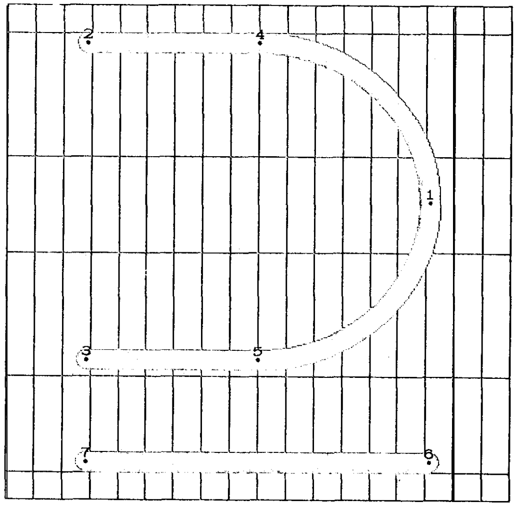


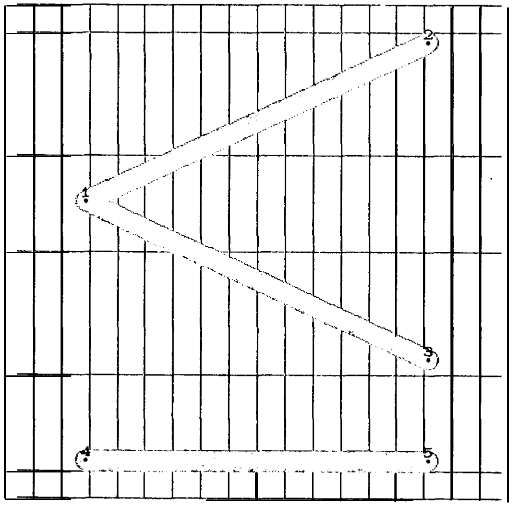


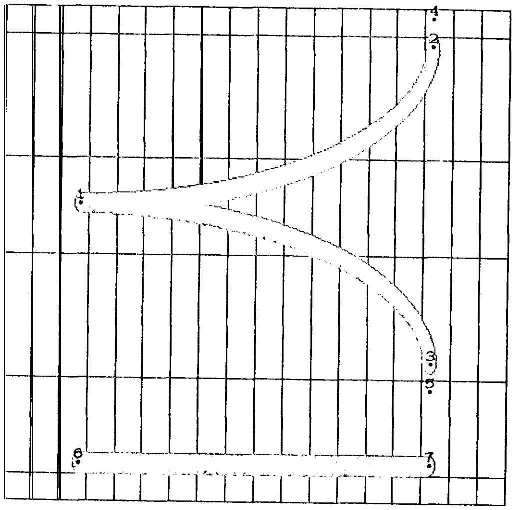


```
"Elementary division operator";
call charbegin('015, 18, 0, 0, .5[px, ph], .5[px, ph] — 2pa, 0);
cpen; lft_10 x_1 = round u;   x_2 = r — x_1;   y_1 = y_2 = a;
win draw 1 . . 2;                                                        % bar
new w_99;   w_99 = w_3 sqrt 2;
top_99 y_3 = .5[m, h];   .5[y_3, y_4] = a;
x_3 = x_4 = .5r;
w_99 draw 3;   draw 4.                                                   % dots

"Intercalation product operator";
call charbegin('016, S, 0, 0, px, pd + pb, 0);
vpen; lft_3 x_1 = u; rt_3 x_2 = r — u;   x_3 = x_4 = .5r
top_3 y_1 = m;   y_1 = y_2 = y_3;   bot_6 y_4 = --d — b;
w_3 draw 1 . 2;                                                          % bar
hpen;   w_3 draw 3.. 4.                                                  % stem

"Bullet";
call charbegin('017, 9, 0, 0, 3.5pu + pa, 3.5pu — pa, 0);
cpen;   x_8 = .5r;   lft_10 x_6 = round u;   top_10 y_8 = round(a + 3 5u);   y_6 = a;
call circle(1, 2, 3, 4, 5, G, 7, 8, win);                     % outer boundary
x_0 = x_8;   y_0 = y_2;
cpcn;   w_10 ddraw 1{x_2 — x_8, y_2 — y_8} . 2{0, —1} . . 3{x_4 — x_2, y_4 — y_2} . . 4{—1, 0} . .
     5{x_6 — x_4, y_6 — y_4} . . 6{0, 1} . . 7{x_8 — x_6, y_8 — y_6} . 8{1, 0} . 1{x_2 — x_8, y_2 — y_8},
     0 . . 0 . . 0 . . 0 . . 0 . . 0 . . 0 . . 0 . . 0.                        % fill it in

"Perpendicular sign or lattice bottom";
cpen;
if fixwidth = 0:   if pa + 8pu > ph:
           call charbegin( '020, 18, 0, 0, ph, ph — 2pa, 0);   top_10 y_1 = h;
     else:   call charbegin('020, 18, 0, 0, 8pu + pa, 8pu — pa, 0);   top_10 y_1 = a + 8u;
     fi;
else:   call charbegin('020, 9, 0, 0, 3.5pu + pa, 3.5pu — pa, 0);   top_10 y_1 = a + 3.5u;
fi;
.5[y_1, Ml = a;   x_1 = x_2 = .5r;
lft_10 x_3 = round u;   x_4 = r — x_3;   y_3 = y_1 = y_2;
w_10 draw 1.. 2;                                                         % stem
draw 3.. 4.                                                              % bar

"Equivalence or congruence sign";
call charbegin('021, 18, 0, 0, px — pe + prt/2 + pa, px — pe + prt/2 — pa, 0);
cpen;   lft_10 x_1 = round u;   x_3 = x_5 = x_1;   x_2 = x_4 = x_6 = r — x_1;
y_1 = y_2;   y_3 = y_4 = u;   y_5 = y_6;   y_1 — y_3 = y_3 — y_5 = round(m -- e);
w_10 draw 1.. 2;                                                   % upper bar
draw 3.. 4;                                                        % middle bar
draw 5.. 6.                                                         % lower bar
```

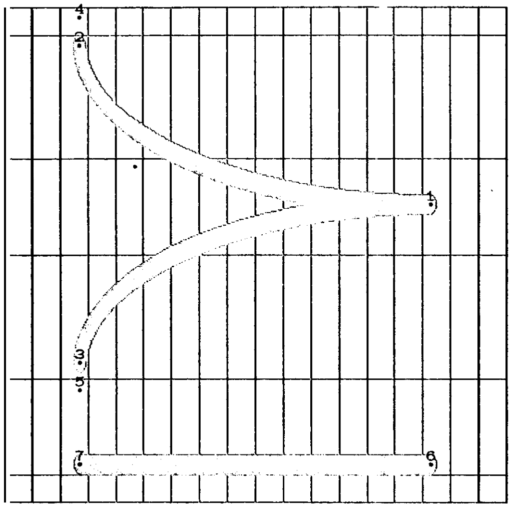


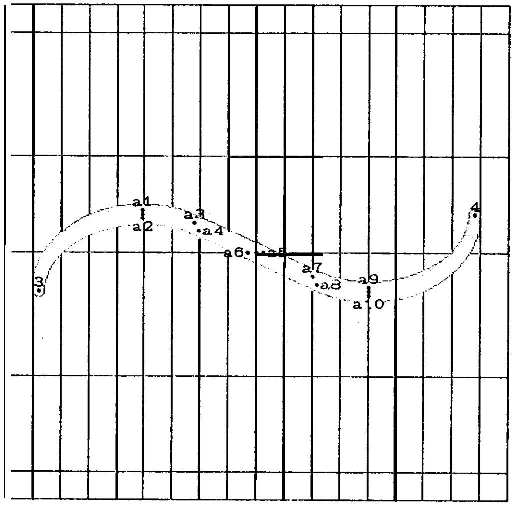


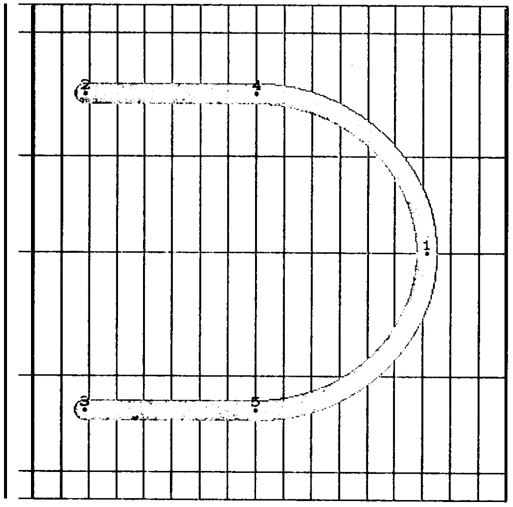


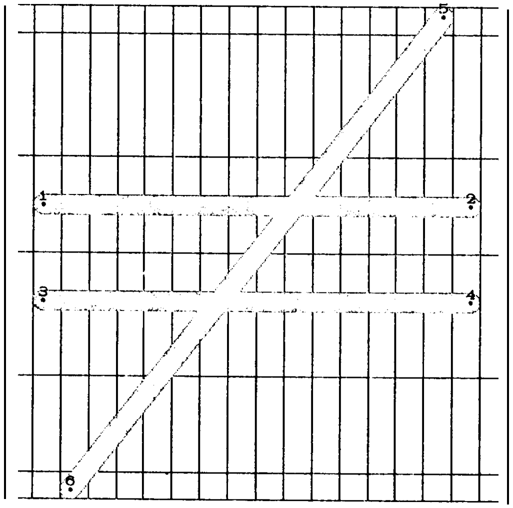


```
'Reflexive subse t sign";
call charbegin( '022, 18, 0, 0, ph, ph —2pa, 0);
cpcn;   lft10x1 = round 2.5u;   x2 = x3 = r — x1;
top10y2 = h;   .5[y2, y3] = y1;   y2 — y1 = (good10 .5[m, h]) — (good10a);
x4 = x5 = .5r;   y4 = y2;   y5 = y3;
w10 draw 2 .. 4{—1, 0} . 1{0, —1} . 5{1, 0} .. 3;                   % stroke
x6 = x1;   x7 = x2;   y6 = y;;   bot10y6 = 2a — h;
draw 6 .. 7.                                                          % bar

"Rcflcxive superset sign";
call charbegin( '023, 18, 0, 0, ph, ph -- 2pa, 0);
cpen;   lft10x2 = round 2.5u;   x2 = x3 = r — x1;
top10y2 = h;   .5[y2, y3] = y1;   y2 — y1 = (good10 .5[m, h]) — (good10a);
x4 = x5 = .5r;   y1 = y2;   y5 = y3;
w10 draw 2 .. 4{1, 0} . 1{0, —1} . 5{ —1, 0} 3 ;                    % stroke
x6 = x1;   x7 = x2;   y6 = y7;   bot10y6 = 2a — h;
draw 6 .. 7.                                                          % bar

"Less than or equal lo sign";
call charbegin( '024, 18, 0, 0, ph, ph —2pa, 0);
cpen;   lft10x1 = round 2.511;   x2 = x3 = r — x1;
top10y2 = h;   .5[y2, y3] = y1;   y2 — y1 = (good10 .5[m, h]) — (good10a);
w10 draw 2 1 .. 1 .. 3 ;                                              % diagonals
x4 = x1;   x5 = x2;   y4 = y5;   bot10y4 = 2a — h;
draw 4 .. 5.                                                          % bar

"Greater than or equal to sign";
call charbegin( '025, 18, 0, 0, ph, ph -- 2pa, 0);
cpen;   lft10x2 = round 2.5u;   x2 = x3 = r — x1;
top10y2 = h;   .5[y2, y3] = y1;   Y : — y1 = (good10 5[m, h]) -- (good10a);
w10 draw 2.. 1 .. 1 .. 3;                                             % diagonals
x4 = x1;   x5 = x2;   y4 = y5;   bot10y4 = 2a — h;
draw 4.. 5.                                                           % bar

"Precedes or equals sign";
call charbegin( '026, 18, 0, 0, ph, ph -- 2pa, 0);
vpen;   lft10x1 = round 2.5u;   x2 = x3 = r — x1;
top10y2 = h;   5[y2, y3] = y1;   y2 -- y1 = (good 10 5[m, h]) -- (good 10a);
x4 = x5 = x2;   y4 = y2 + b;   y5 = y3 — b;
w10 draw (4 )2 .. 1{—1, 0 )   1{1, 0} .. 3( .. 5);                  % diagonals
x6 = x1 ;   x7 = x2;   y6 = y7;   bot10y6 = 2a — h;
draw  6  7.                                                           % bar
```

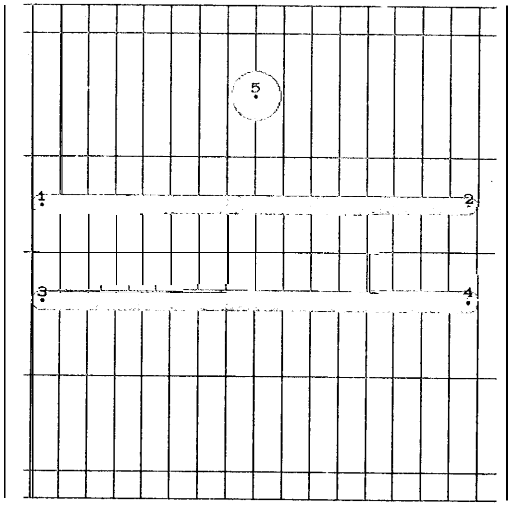


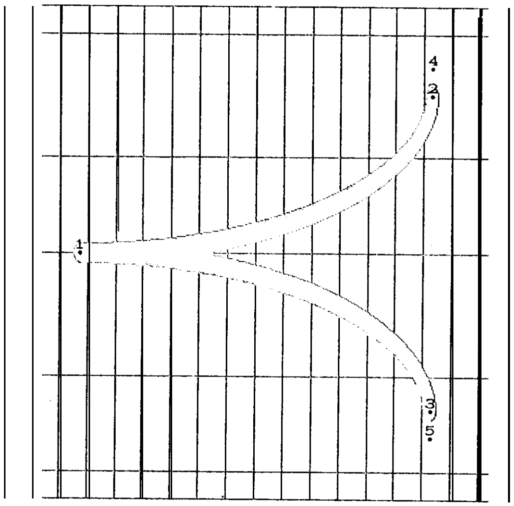


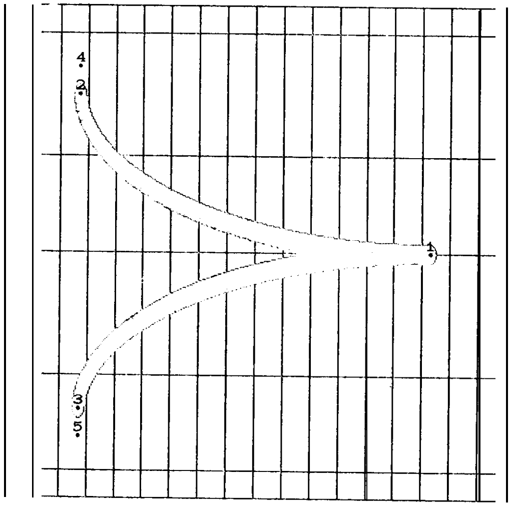


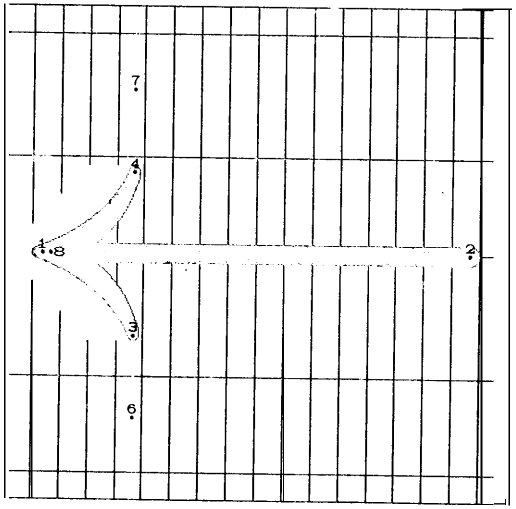


```
"Follows or equals sign";
call charbegin( '027, 18, 0, 0, ph, ph — 2pa, 0);
vpen;  lft10x2 = round 2.5u;  x2 = x3 = 7 — x1;
top10y2 = h;  .5[y2, y3] = y1;  y2 — y1 = (good10 .5[m, h]) — (good10a);
x4 = x5 = x2;  y4 = y2 + b;  y5 = y3 — b;
w10 draw (4. )2 . . 1{1, 0} . . 1{—1, 0} . 3( . . 5);                    % diagonals
x6 = x1;  x7 = x2;  y6 = y7;  bot10y6 = 2a — h;
draw 6 . . 7.                                                             % bar

"Similarity sign";
call charbegin('030, 18, 0, 0, .5(px — pe) + pa, 0, 0);
vpen;  top10y1 = round(a + .5(m — e) + eps);  top10y1 — hot 10y2 = round(m — e);
lft10x3 = round u;  y5 = .5[y1, y2]; y3 = y2; y4 = y1;  x4 = r — x3;  x5 = 7 — x5;
call 'a zdmw(3, 1, 5, 2, 4, w10, w10 + deltaw, 7.5u/(e — m)).            % stroke

"Approximate equality sign";
call charbegin('031, 18, 0, 0, 1.1(px — pe) + pa, 1.1(px — pe) — pa, 0);
vpen;  top10y1 = round(a + 1.1(m -- e) + eps);  top10y1 — bot10y2 = round(m — e);
Ift 10x3 = round u;  y5 = .5[y1, y2];  y3 = y2; y1 = y1;  x4 = r — x3; x5 = 7 — x5;
x8 = x3;  x9 = x4;  x10 = x5;
y1 — y6 = y2 — y7 = y3 — y8 = y4 — y9 = y5 — y10 = round 1.2(m — e);
call 'a zdraw(3, 1, 5, 2, 4, w10, w10 + delta w, 7.5u/(e — m));         %upper stroke
call `b zdraw(8, 6, 10, 7, 9, w10, w10 + deltaw, 7.5u/(e — m)).          % lower stroke

"Proper subset sign";
call charbegin( '032, 18, 0, 0, .5[px, ph] + prt/2, .5[px, ph] + prt/2 — 2pa, 0);
cpen;  lft10x1 = round 2.5u;  x2 = x3 = 7 — x1;
y2 = good 10 .5[m, h];  .5[y2, y3] = y1 = good10a;
x4 = x5 = .5r;  y4 = y2;  y5 = y3;
w10 draw 2. . 4{—1, O}. 1{0, -1). . 5{1, O}. . 3.                        % stroke

"Proper superset sign";
call charbegin( '033, 18, 0, 0, .5[px, ph] + prt/2, .5[px, ph] + prt/2 — 2pa, 0);
cpen;  lft10x2 = round 2.5u;  x2 = x3 = 7 — x1;
y2 = good10 .5[m, h];  .5[y2, y3] = y1 = good10a;
x4 = x5 = .5r;  y4 = y2;  y5 = y3;
w10 draw 2. . 4{1, 0} . . 1{0, —1} . . 5{—1, 0} . . 3.                   % stroke

"Unequal sign";
call charbegin('034, 18, 0, 0, ph + pb, ph + pb — 2pa, 0);
cpcn;  lft10x1 = round u;  x3 = x1;  x2 = x4 = 7 — x1;
y1 = y2;  y3 = y4;  y1 — y3 = round(m — e);  .5[y1, y3] = a ;
w10 draw 1 . . 2;                                                         % upper bar
draw 3. . 4;                                                              %lower bar
rt 10x5 = round(r — 2u);  lft 10x6 = round 2u;
top10y5 = h + b;  bot 10y6 = - d — b;
draw 5. . 6.                                                              % diagonal
```

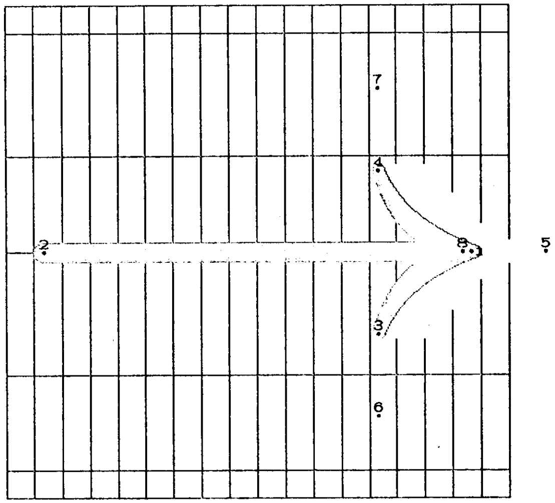


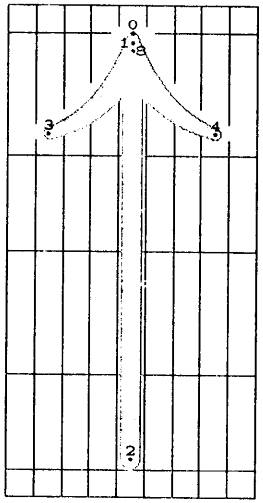


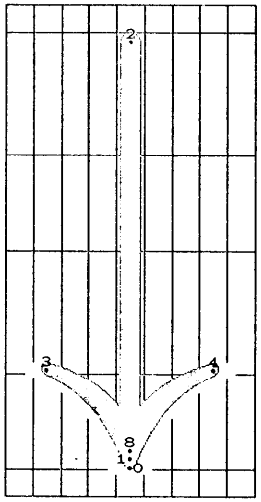




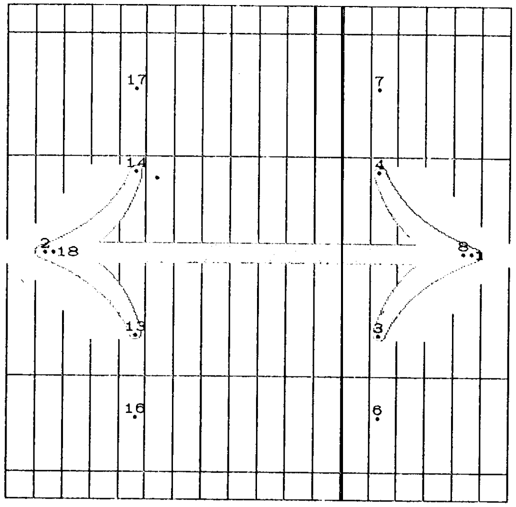
```
“Dot over equal sign”;
Call $charbegin($ ‘035, 18, 0, 0, $ph$, 0, 0);
cpen;   lft$_{10}x_1$ = round $u$;   $x_3 = x_1$;   $x_2 = x_4 = r - x_1$;
$y_1 = y_2$;   $y_3 = y_4$;   $y_1 - y_3$ = round($m$ − e);   $.5[y_1, y_3]$ = a;
$w_{10}$ draw 1 . . 2;                                                % upper bar
draw 3.. 4;                                                           % lower bar
new $w_{99}$;
if $w_3 < w_0$ sqrt 2:   $w_{99}$ = round $w_0$ sqrt 2;
else:   $w_{99} = w_3$;
fi;
$x_5 = .5r$;   $y_5 = 5[m, h]$; $w_{99}$ draw 5 .                              % dot

“Precedes sign”;
call $charbegin($’036, 18, 0, 0, $.5[px, ph] + prt/2, .5[px, phI] + prt/2 - 2pa$, 0)
vpen;   Ift $_{10}x_1$ = round 2 5$u$;   $x_2 = x_3 = r - - x_1$;
$y_2$ = good $_{10}5[m, h]$;   $.5[y_2, y_3] = y_1$ = good$_{10}a$;
$x_4 = x_5 = x_2$;   $y_1 = y_2 + b$;   $y_5 = y_3 - b$;
$w_{10}$ draw (4. . )2 . . 1{−1, 0} .  1{1, 0} . 3( . . 5).                 % diagonals

“Follows sign”;
callcharbegin( ‘037, 18, 0, 0, $.5[px, ph] + prt/2, .5[px, ph] + prt/2 - 2pa$, 0);
vpen;   lft$_{10}x_2$ = round 2.5$u$;   $x_2 = x_3$ = 7 − $x_1$;
$y_2$ = good$_{10}$ .5[$m, h$];   $.5[y_2, y_3] = y_1$ = good$_{10}a$;
$x_4 = x_5 = x_2$;   $y_1 = y_2 + b$;   $y_5 = y_3 - b$;
$w_{10}$ draw (4. )2 . . 1{1, 0} . . 1{−1, 0} . . 3( . . 5).                % diagonals

“Leftward arrow”;
call charbegin( ‘040, 18, 0, 0, $.24ph + .5prt$ + pa, $.24ph + .5prt -$ pa, 0);
cpen;   lft$_{10}x_1 = x_0$ = round $u$;   rt$_{10}x_2$ = round($r$ − u);
$y_1 = y_2 = y_5 = y_8$ = good$_{10}a$;
$w_{10}$ draw 1.. 2;                                                        % bar
hpen; lft$_1 x_8 = x_0$;
$x_5 - x_8 = x_8 - x_3$ = - - fixwidth[3$u$, 6$u$] − eps;   $x_3 = x_4 = x_6 = x_7$;
$y_3 - y_6 = y_1 - y_3 = y_4 - y_1 = y_7 - y_4$ = .24$h$ + eps;
lpen#;   $w_{10} + w_1$ draw (5. )8 . 3( . . 6);              % erase excess at lower left
hpen;   draw (|$w_1$|5 . . )8 . . |$w_0$|3( . . 6);                            % lower point
lpen#;   $w_{10} + w_1$ draw (5. )8 . . 4( . . 7);             % erase excess at upper left
hpen;   draw (|$w_1$|5 )8 . |$w_0$|4( . 7 ) .                                  %upper point
```

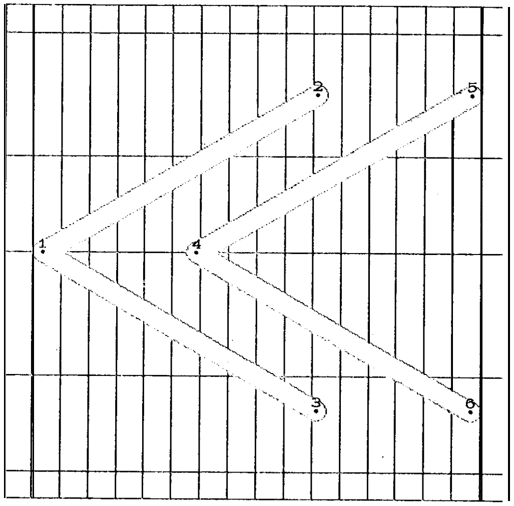


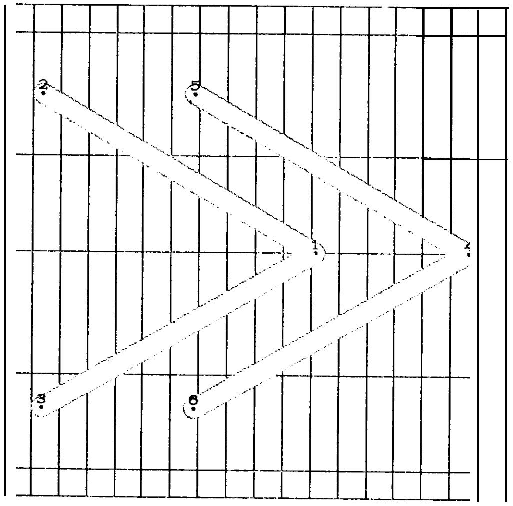


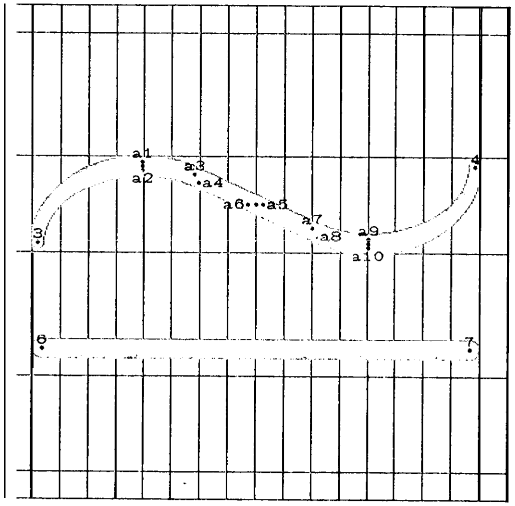


```
‘Rightward arrow”;
call charbegin(´041, 18, 0, 0, .24ph + .5prt + pa, .24ph + .5prt − pa, 0);
cpen;   lft_10 x_2 = round u;   rt_10 x_1 = x_0 = round(r − u);
y_1 = y_2 = y_5 = y_8 = good_10 a;
w_10 draw 1 . . 2;                                                      % bar
hpen;   rt_1 x_8 = x_0;
x_5 − x_8 = x_8 − x_3 = fixwidth[3u, 6u] + eps;   x_3 = x_4 = x_6 = x_7;
y_3 − y_6 = y_1 − y_3 = y_4 − y_1 = y_7 − y_4 = .24h + eps;
rpen#;   w_10 + w_1 draw (5 . . )8 . . 3( . . 6);     % erase excess at lower right
hpen;   draw (|w_1|5 . . )8 . . |w_0|3( . . 6);                     % lower point
rpen#;   w_10 + w_1 draw (5 . . )8 . . 4( . . 7);     % erase excess at upper right
hpen;   draw (|w_1|5 . . )8 . . |w_0|4( . . 7).                     % upper point

“Upward arrow”;
call charbegin(´042, 9, 0, 0, ph, ph − 2pa, 0);
cpen;   top_10 y_1 = y_0 = h;   .5[y_1, y_2] = a;
x_0 = x_1 = x_2 = x_5 = x_8 = good_10 .5r;
w_10 draw 1 . . 2;                                                     % stem
vpen;   top_7 y_8 = y_0;
lpen#;   w_10 draw 0 . . 8;   rpen#;   w_10 draw 0 . . 8;       % clean the top
y_5 − y_8 = y_8 − y_3 = .24h + eps;   y_3 = y_4 = y_6 = y_7;
x_3 − x_6 = x_1 − x_3 = x_4 − x_1 = x_7 − x_4 = 3u + eps;
lpen#;   w_10 draw (5 . . )8 . . 3( . . 6);                % erase excess at left
vpen;   draw (|w_7|5 . . )8 . . |w_6|3( . . 6);                       % left point
rpen#;   w_10 draw (5 . . )8 . . 4( . . 7);               % erase excess at right
vpen;   draw (|w_7|5 . . )8 . . |w_6|4( . . 7).                      % right point

“Downward arrow”;
call charbegin(´043, 9, 0, 0, ph, ph − 2pa, 0);
cpen;   top_10 y_2 = h;   .5[y_1, y_2] = a;   y_0 = bot_10 y_1;
x_0 = x_1 = x_2 = x_5 = x_8 = good_10 .5r;
w_10 draw 1 . . 2;                                                     % stem
vpen;   bot_7 y_8 = y_0;
lpen#;   w_10 draw 0 . . 8;   rpen#;   w_10 draw 0 . . 8;       % clean the top
y_5 − y_8 = y_8 − y_3 = −.24h − eps;   y_3 = y_4 = y_6 = y_7;
x_3 − x_6 = x_1 − x_3 = x_4 − x_1 = x_7 − x_4 = 3u + eps;
lpen#;   w_10 draw (5 . . )8 . . 3( . . 6);                % erase excess at left
vpen;   draw (|w_7|5 . . )8 . . |w_6|3( . . 6);                       % left point
rpen#;   w_10 draw (5 . . )8 . . 4( . . 7);               % erase excess at right
vpen;   draw (|w_7|5 . . )8 . . |w_6|4( . . 7).                      % right point
```







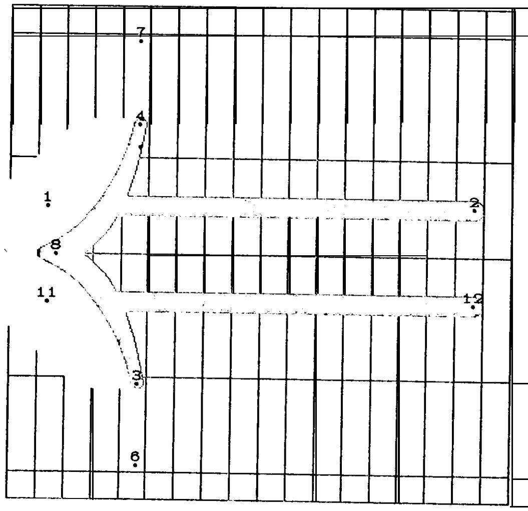


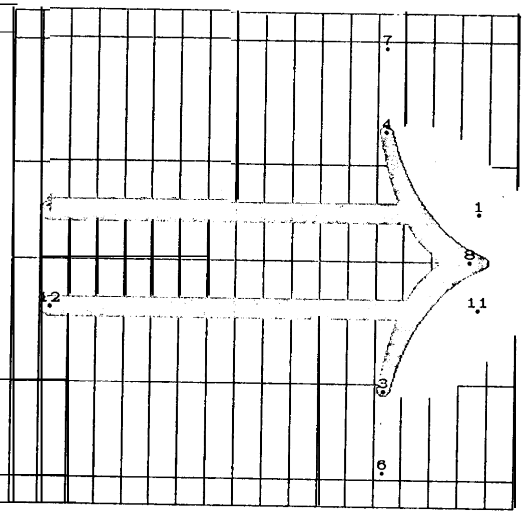
"Left-and-right arrow";
call *charbegin*('044, 18, 0, 0, .24*ph* + .5*prt* + pa, 24*ph* + .5*prt* — *pa*, 0);
cpcn; lft $_{10}x_2 = x_{10}$ = round u; rt $_{10}x_1 = x_0$ = round$(r - u)$;
$y_1 = y_2 = y_5 = y_8$ = good$_{10}a$;
w_{10} draw 1 2; % bar
hpen; rt $_1x_8 = x_0$; lft $_1x_{18} = x_{10}$;
$x_5 - x_8 = x_8 - x_3 = x_{13} - x_{18} = x_{18} - x_{15} =$ *fixwidth*$[3u, 6u]$ + *eps*;
$x_3 = x_1 = x_6 =$ x7; $x_{13} = x_{14} = x_{16} = x_{17}$;
$y_3 - y_6 = y_1 - y_3 = y_4 - y_1 = y_7 - y_4 = .24h$ + *eps*;
$y_{13} = y_3$; $y_{14} = y_4$; $y_{15} = y_5$; $y_{16} = y_6$; $y_{17} = y_7$; $y_{18} = y_8$;
rpen#; $w_{10} + w_1$ draw (5.)8 . . 3(. . 6); % erase excess at lower right
hpen; draw ($|w_1|$5 . .)8 . $|w_0|$3(. . 6); % lower right point
rpen#; $w_{10} + w_1$ draw (5.)8 . 4(. . 7); % erase excess at upper right
hpen; draw ($|w_1|$5 . .)8 . $|w_0|$4(. 7); %upper right point
lpen#; $w_{10} + w_1$ draw (15.)18 . . 13(.16); % erase excess at lower left
hpen; draw ($|w_1|$15 . .)18 . $|w_0|$13(. . 16); % lower left point
lpen#; $w_{10} + w_1$ draw (15 . .)18 . . 14(. . 17); % erase excess at upper left
hpen; draw ($|w_1|$15 . .)18 . . $|w_0|$14(. . 17). %upper left point

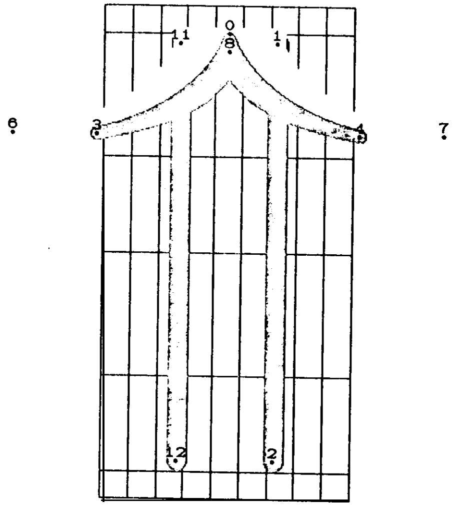
"Much less sign";
call *charbegin*('045, 18, 0, 0, .5[*px*, *ph*] + *prt*/2, 5[*px*, ph] + *prt*/2 — 2*pa*, 0);
cpen; lft $_{10}x_1$ = round u; rt $_{10}x_2$ = round I I .5u; $x_3 = x_2$;
y_2 = good $_{10}$.5[m, h]; .5[y_2, y_3] = y_1 = good$_{10}a$;
w_{10} draw 2.. 1 . . 1. . 3; % left diagonals
rt$_{10}x_5$ = round$(r - u)$; $x_6 = x_5$; $x_4 - x_1 =$ X5 $- x_2$; $y_4 = y_1$; $y_5 = y_2$; $y_6 = y_3$;
draw 5.. 4.. 4.. 6. % right diagonals

"Much greater sign";
call *charbegin*('046, 18, 0, 0, .5[*px*, *ph*] + *prt*/2, .5[*px*, *ph*] + *prt*/2 — 2*pa*, 0);
cpen; lft$_{10}x_2$ = round u; rt $_{10}x_1$ = round 1 1.5~; $x_3 = x_2$;
y_2 = good $_{10}$.5[m, h]; .5[y_2, y_3] = y_1 = good$_{10}a$;
w_{10} draw 2 . . 1 . . 1 . . 3; % left diagonals
rt$_{10}x_4$ = round$(r - u)$; $x_6 = x_5$; $x_4 - x_1 = x_5 - x_2$; $y_4 = y_1$; $y_5 = y_2$; $y_6 = y_3$;
draw 5.. 4.. 4.. 6. % right diagonals

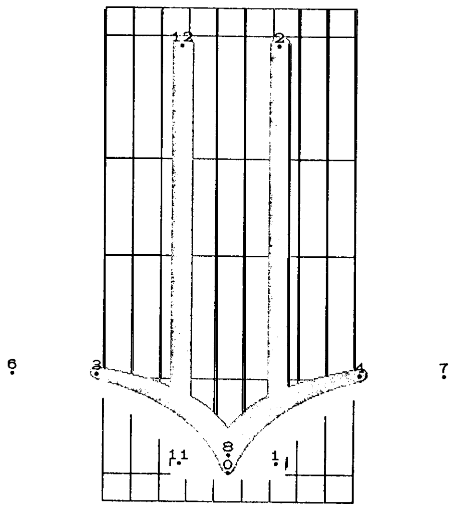
"Similar or **equal** sign";
call *charbegin*('047, 18, 0, 0, *px* — *pe* + *prt*/2 + *pa*, *px* — *pe* + *prt*/2 — *pa*, 0);
vpen; top$_{10}y_1$ = round(a + (m — e) + *eps*); top$_{10}y_1$ -- bot $_{10}y_2$ = round(m — e);
lft$_{10}x_3$ = round u; $y_5 = .5[y_1, y_2]$; $y_3 = y_2$; $y_4 = y_1$; $x_4 = r - x_3$; $x_5 = r - x_5$;
call 'a *zdraw*(3, 1, 5, 2, 4, w_{10}, w_{10} + *deltaw*, 7 5u/(e — m)); % stroke
cpen; $y_6 = y_7$; a $- y_6$ = round(m — e); Ift $_{10}x_6$ = round u; $x_7 = r - - x_6$;
w_{10} draw 6.. 7. % bar

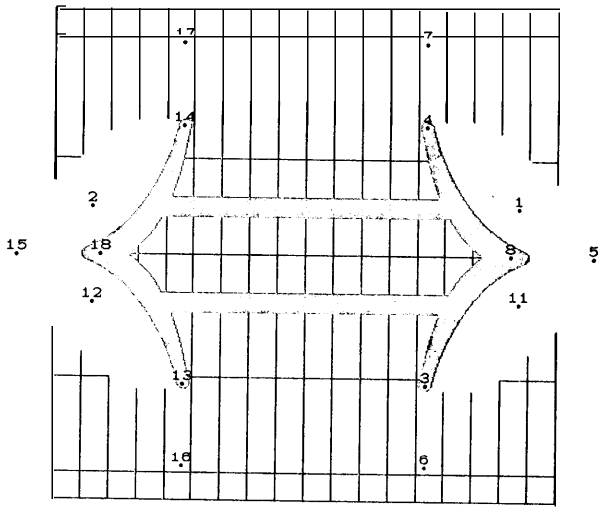




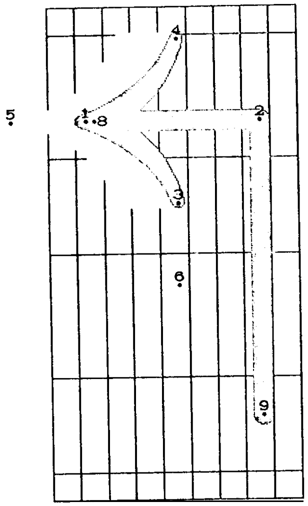


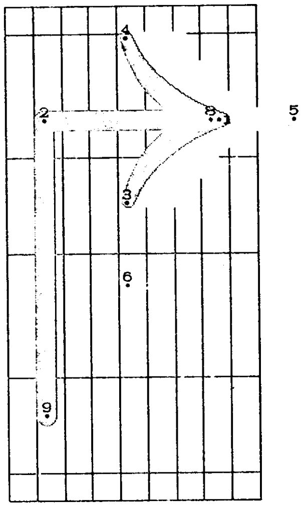
```
"Double leftward arrow";
call charbegin( '050, 18, 0, 0,.24ph + .5prt + .5(px — pe) + pa,
                        .24ph + .5prt + .5(px — pe) — pa, 0);
cpen;  lft10x1 = x0 = round u;  rt10x2 = round(r — u);  x11 = x1;  x12 = x2;
y5 = y8 = good10a;  y1 = y2;  y11 = y12;  .5[y1,y1 I] = y5;  y1 — y11 = round(m — e);
w10 draw 1..2; draw 11.. 12;                                          % bars
hpen; lft1x8 = x0;
x5 — x8 = x8 — x3 = —fixwidth[3u, 6u] — eps;  x3 = x4 = x6 = x7;
y3 — y6 = y1 I — y3 = y4 — y1 = y7 — y4 = .24h + eps;
lpen#;  w10 + wI ddraw (5. .)8..3(..6),11 .. 11;         % erase excess at lower left
hpen;  draw (|w1|5..)8..|w0|3(..6);                                  % lower point
lpen#;  w10 + w1 ddraw (5.. )8..4(..7), 1. . 1;          % erase excess at upper left
hpen;  draw (|w1|5..)8..|w0|4(..7).                                  % upper point

"Double rightward arrow";
call charbegin('051, 18, 0, 0, .24ph + .5prt + .5(px — pe) + pa,
                        .24ph + .5prt + .5(px — pe) — pa, 0);
cpen;  lft10x2 = round u;  rt10x1 = x0 = round(r — u);  x11 = x1;  x12 = x2;
y5 = y8 = good10a;  y1 = y2;  y11 = y12;  .5[y1, y11] = y5;  y1 -- y11 = round(m — e);
w10 draw 1..2; draw 11.. 12;                                          % bars
hprn;  rt1x8 = x0;
x5 — x8 = x8 — x3 = fixwidth[3u, 6u] + cps;  x3 = x4 = x6 = x7;
y3 — y6 = y11 — y3 = y4 — y1 = y7 — y4 = .24h + eps;
rpen#;  w10 + w1 ddraw (5. .)8..3(..6), 11.. 11;        % erase excess at lower right
hpcn;  draw (|w1|5..)8..|w0|3(..6);                                  % lower point
rpen#;  w10 + w1 ddraw (5. .)8..4(..7), 1.. 1;          % erase excess at upper right
hpen;  draw (|w1|5..)8..|w0|4(..7).                                  % upper point

"Double upward arrow";
call charbegin('052, 9, 0, 0, ph, ph — 2pa, 0);
cpen;  top10y1 = y0 = h;  .5[y1, y2] = a;  y11 = y1;  y12 = y2;
x0 = x5 = x8 = good10 .5r;  x1 = x2;  x11 = x12;
.5[x1, x11] = x5;  x1 — x11 = round 3.5u;
w10 draw 1..2; draw 11..12;                                          % stems
vpen;  top7y8 = y0;
lpen#;  2u draw 0.. 8; rpen#;  221 draw 0.. 8;                       % clean the top
y5 — y8 = y8 — y3 = .24h + eps;  y3 = y4 = y6 = y7;
x3 — x6 = x12 — x3 = x4 — x1 = x7 — x4 = 3u + eps;
lpen#;  2u draw (5.. )8..3(..6);                                % erase excess at left
vpen;  draw (|w7|5..)8..|w6|3(..6);                                  % left point
rpen#;  2u draw (5.. )8..4(..7);                               % erase excess at right
vpen;  draw (|w7|5..)8..|w6|4(..7).                                 % right point
```

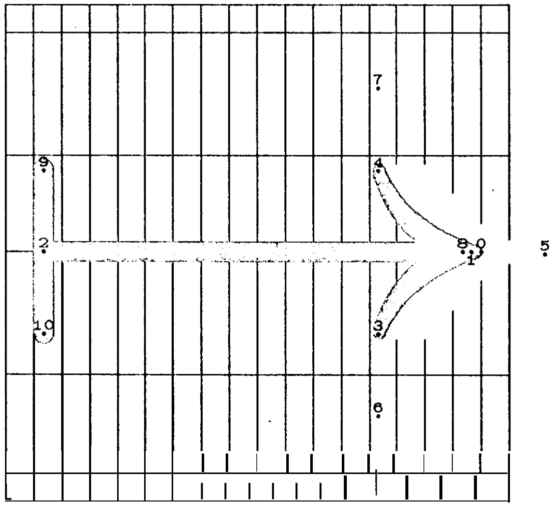




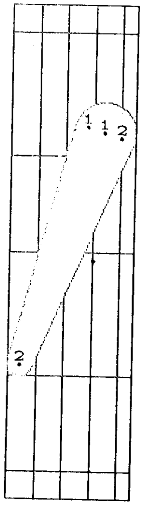
```
'Ɔouble downward arrow";
call charbegin( '053, 9, 0, 0, ph, ph—2pa, 0);
cpen;   top_10 y_2 == h;   .5[y_1, y_2] == a;   y_0 = bot_10 y_1;   y_11 = y_1;   y_12 == y_2;
x_0 = x_5 == x_8 == good,,, .5r;   x_1 = x_2;   x_11 = x_12;
.5[x_1, x_1 1] = x_5;   x_1 — x_11 == round 3.5u;
w_10 draw 1.. 2; draw 11.. 12;                                         % stems
vpen;   bot_7 y_8 = y_0;
lpen#; 2u draw 0..8; rpen#; 2u draw 0.. 8;                             % clcnn the top
y_5 — y_8 = y_8 — y_3 == —.24h — eps;   y_3 = y_4 == y_6 = y_7;
x_3 — x_6 = x_12 --- x_3 == x_1 — x_1 = x7 — x_1 = 3u + eps;
lpen#;   2u draw (5. )8..3(. 6);                                       % erase excess at left
vpen;   draw (|w_7|5..)8..|w_6|3(..6);                                % left point
rpen#;   2u draw (5 . )8..4(..7);                                      % crasc excess at right
vpen;   draw (|w_7|5..)8..|w_6|4(..7).                                % right point

"Double left-and-right arrow";
call charbegin('054,18, 0, 0, .24ph +·.5prt + .5(px — pe) + pa,
                    .24ph + 5prt + .5(px -- pc) — pa, 0);
cpen;   lft_10 x_2 == x_10 == roundu;   rt_10 x_1 = x_0 == round(r -- u);   x_11 == x_1;   x_12 == x_2;
y_5 == y_8 == good_10 a;   y_1 = y_2;   y_11 == y_12;   .5[y_1, y_1 1] ==y_5;   y_1 — y_11 == round(m — e);
w_10 draw 1.. 2; draw 11.. 12;                                         % bars
hpen;   rt_1 x_8 = x_0;   lft_1 x_18 == x_10;
x_5 — x_8 == x_8 — x_3 == x_13 -- x_18 == x_18 - -x_15 == fixwidth[3u, 6u] + eps;
x_3 = x_4 = x_6 = x 7;   x_13 = x_14 = x_16 = x_17;
y_3 — y_6 = y_11 — y_3 = y_1 — y_1 == y_7 — y_4 = .24h + eps;
y_13 == y_3;   y_1 1 == y_4;   y_15 == y_5; y_16 == y_6; y_17 == y_7; y_18 == y_8;
rpen#;   w_10 +- w_1 ddraw (5. . )8..3(..6),11.. 11;                  % erase excess at tower right
hpen;   draw (|w_1|5..)8..|w_0|3(..6);                                % lower right point
rpen#;   w_10 + w_1 ddraw (5. . )8..4(..7), 1.. 1;                     % crasc excess at upper right
hpen;   draw (|w_1|5.)8.|w_0|4( 7 );                                   % upper right, point
lpen#;   w_10 + w_1 ddraw (15. . )18..13(.16),12.. 12;                 % erase excess at tower left
hpen;   draw (|w_1|15..)18..|w_0|13(..16);                            % lower left point
lpen#;   w_10 |- w_1 ddraw (15. )18.. 14(.17), 2. . 2;                 % crasc excess at upper left
hpen;   draw (|w_1|15..)18..|w_0|14(. . 17).                          % upper left point
```

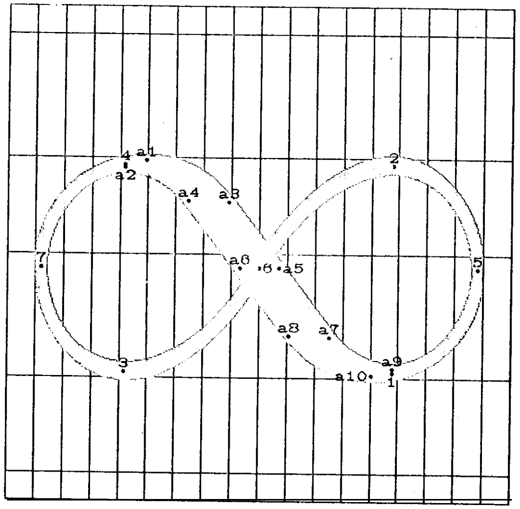


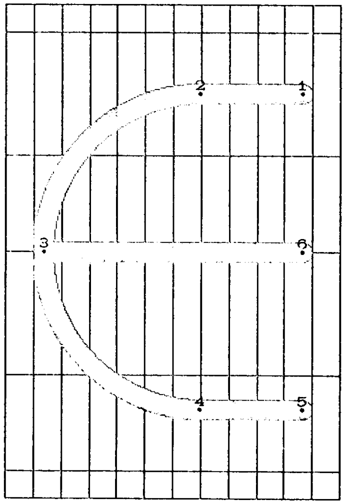


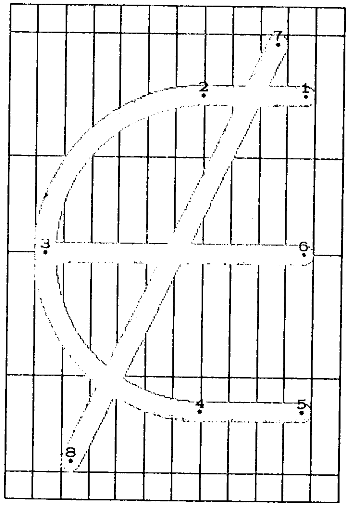


```
"Left shift sign";
call charbegin('055, 9, 0, 0, ph, .5pd, 0);
cpen;   Ift 10x1 = x0 = round u;   rt 10x2 = round(r - u);
y1 = y2 = y5 = y8 = good,,, .75h;
w10 draw 1 .. 2;                                                    % bar
x9 = x2;  bot10y9 = -.5d; draw 2.. 9;                               % stem
hpen; lft1x8 = x0;
x5 - x8 = x8 - x3 = -3u - eps;   x3 = x4 = x6 = x7;
y3 - y6 = y1 - y3 = y 1 - y1 = y7 - y4 = .24h + eps;
lpen#;   w10 + w1 draw (5.. )8 .. 3( .. 6);            % erase excess at lower left
hpen;   draw (|w1|5 .. )8 .. |w0|3( .. 6);                      % lower point
lpen#;   w10 + w1 draw (5. . )8 .. 4( .. 7);           % erase excess at upper left
hpen;   draw (|w1|5 .. )8 .. |w0|4( .. 7).                      % upper point

"Right shift sign";
call charbegin('056, 9, 0, 0, ph, .5pd, 0);
cpen;   lft10x2 = round u;   rt 10x1 = x0 = round(r - u);
y1 = y2 = y5 = y8 = good10 .75h;
w10 draw 1.. 2;                                                     % bar
x9 = x2;  bot10y9 = -.5d;  draw 2.. 9;                              % stem
hpen;   rt1x8 = x0;
x5 - x8 = x8 - x3 = 3u + eps;   x3 = x4 = x6 = x7;
y3 - y6 = y1 - y3 = y4 - y1 = y7 - y4 = .24h + eps;
rpen#;   w10 + w1 draw (5. )8 .. 3( .. 6);            % erase excess at lower right
hpcn;   draw (|w1|5 .. )8 .. |w0|3( .. 6);                      % lower point
rpen#;   w10 + w1 draw (5.. )8 .. 4( .. 7);           % erase excess at upper right
hpen;   draw (|w1|5 .. )8 .. |w0|4( .. 7).                      % upper point

"Maps-to relation";
call charbegin( '057, 18, 0, 0, .24ph + .5prt + pa, .24ph + .5prt -- pa, 0);
cpen; lft10x2 = round u;   rt 10x1 = x0 = round(7 - u);
y0 = y1 = y2 = y5 = y8 = good10a;
w10 draw 1 .. 2;                                                    % bar
hpen;   rt1x8 = x0;
x5 - x8 = x8 - x3 = fixwidth[3u, 6u] + eps;   x3 = x4 = x6 = x7;
y3 - y6 = y1 - y3 = y4 -- y1 = y7 - y1 = .24h + eps;
rpen#;   w10 + w1 draw (5 . )8 .. 3( 6);              % erase excess at lower right
hpen;   draw (|w1|5 )8 |w0|3( .. 6);                            % lower point
rpen#;   w10 + w1 draw (5. )8 .. 4( .. 7);            % erase excess at upper right
hpen;   draw (|w1|5 )8 . |w0|4( .. 7);                          % upper point
x9 = x10 = x2;   y9 = y4;   y10 = y3;
cpen;   w10 draw 9.. 10.                                            % stem
```

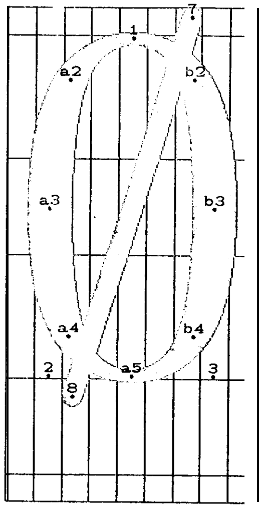


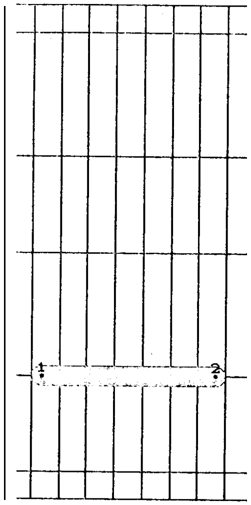


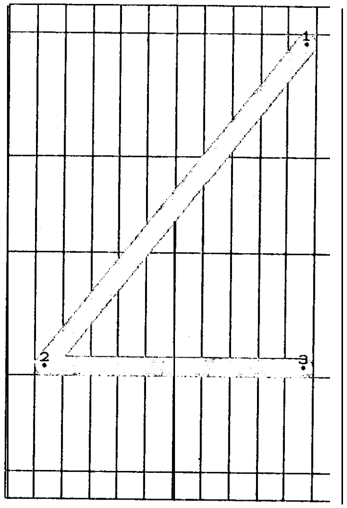


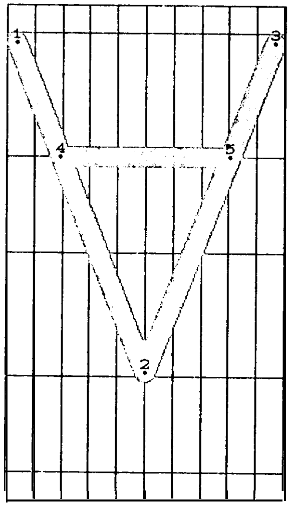


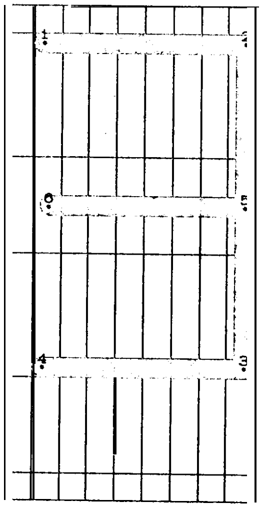
```
"l'rime symbol (intended as superscript only)";
call charbegin( '060, 4 5, 0, 0, .8ph, 0, 0);
new w98, w99;   w98 = round .5[w0, w1];   w99 = round( hold + 2deltaw);
cpen;   top99y1 = round .8h; rt99x1 = r; bot98y2 = 0 ; lft98x2 = 0 ;
call cdraw(1, 2, 99, 98).                                                % diagonal

"lnfnity";
call charbegin('061, 18, 0, 0, px, 0, .5px·slant—.5pu);
new w98, w99;   w99 = round .25[w6, w7];   w98 = 2[w1, w2];
vpen;   top99y2 = m + 00;   bot99y1 = - 0 0 ;   y2 = y4;   y1 = y3;
x1 = x2;   x3 = x4;   y5 = y6 = y7 = .5[y1, y2];
lft0x7 = round u;   rt0x5 = round(r -- u);   x6 = .5[x7, x5];
new ss, mss;                                         % reciprocal slope at center
if w0 = w98:   mss = ss;   ss = --u/m;
else:   mss = .75ss;   ss = —6u/m;
fi;
call ` a zdraw(7, 4, 6, 1, 5, w99, w98, ss);          % lower right and upper left slrokes
w99 draw 5{0, 1} . . 2{—1, 0} . 6{mss, —1} . 3{—1, 0} . .
    7{0, 1}.                                          % upper right and lower left strokes

"Element  sign";
call charbegin('062, 12, 0, C, 5[px, ph] + prt/2, .5[px, ph] + prt/2 -- 2pa, 0);
cpen;   rt10x1 = round(r — u);   lft10x3 = round u;
x5 = x6 = x1;   x2 = x4 = .5(r + 2u);
y1 = y2 = good10(.5[m, h]);   y3 = y6 = a;   y4 = y5;   .5[y5, y1] = y6;
w10 draw 1.. 2{—1, 0} . . 3{0, —1} . . 4{1, 0} . . 5;                    % bowl
draw 3 . . 6.                                                           % bar

"Nonclement sign";
call charbegin('063, 12, 0, 0, ph, ph -- 2pa, 0);
cpen;   rt10x1 = round(r -- u);   lft10x3 = round u;
x j = x6 = x1;   x2 = x4 = .5(r + 2u);
y1 = y2 = good10(.5[m, h]);   y3 = y6 = a;   y4 = y5;   5[y5, y1] = y6;
w10 draw 1.. 2{—1, 0} . . 3{0, —1} . . 4{1, 0} . . 5;                    % bowl
draw 3.. 6;                                                             % bar
rt10x7 = round(r — 2u);   lft10x8 = round 2u;   top10y7 = h;   .5[y7, y8] = a;
draw 7.. 8.                                                             % diagonal
```

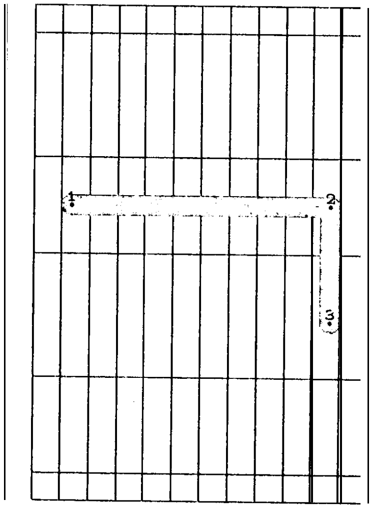




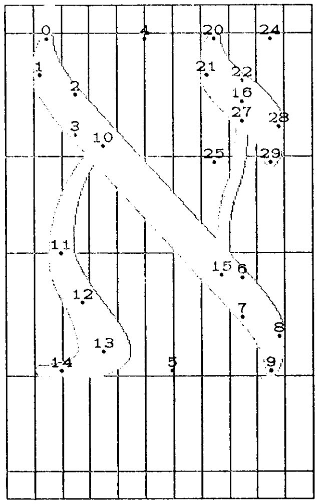




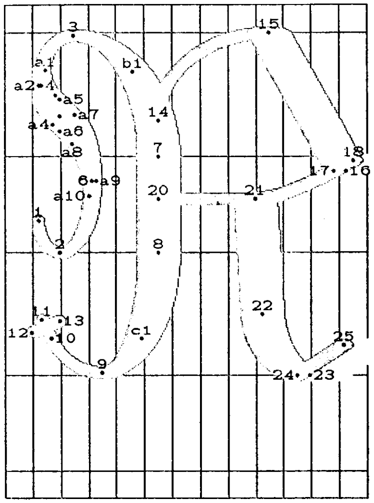




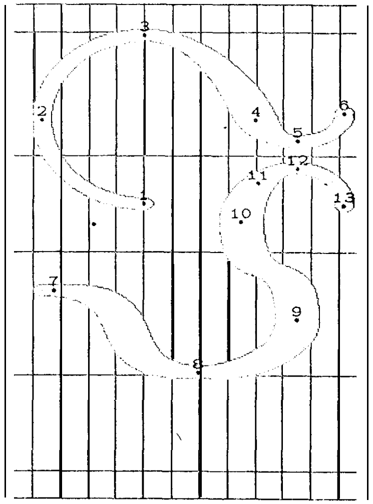
```
'Empty set symbol";
call $charbegin$('064, 9, 0, 0, ph + pb, pb, 0);
if $fixwidth = 0$:   new save;   $save$ = sqrttwo ; new $\mathbf{sqrttwo}$;
     $\mathbf{sqrttwo}$ = sqrt(1.231144413$save$);                   % the constant is $2^{3/10}$
fi;
hpen;
if $w_2 > 1.5u$:   $\mathrm{lft}_2 x_2 =$ round $.75u$;
else:   $x_2$ = good,! $1.5u$;
fi;
$x_1 = r - x_1$;                                                  % axis of left-right symmetry
$x_3 = r - x_2$;   $\mathrm{top}_0 y_1 = h + oo$;   $\mathrm{bot}_0 y_2 = -oo$;   $y_3 = y_2$;
call ` a $darc(1, 2, w_2)$;   call ` b $darc(1, 3, w_2)$;                  % bowl
if $fixwidth$ = 0: new sqrttwo;   $\mathbf{sqrttwo} = \mathbf{save}$;
fi;
cpen;   $\mathrm{rt}_{10} x_7 =$ round$(r - 221)$;   $\mathrm{lft}_{10} x_8 =$ round $2u$;   $\mathrm{top}_{10} y_7 = h + b$;   bot $_{10} y_8 = -b$;
$w_{10}$ draw 7.. 8.                                                  % diagonal

"Underline";
call $charbegin$( '065, 9, 0, 0, 0, 0, 0);
cpcn;   $\mathrm{lft}_{10} x_1 =$ round $u$;   $x_2 = r - x_1$;   $y_1 = y_2 = 0$;
$w_{10}$ draw 1 . . 2.                                                 % bar

"Angle sign";
call $charbegin$('066, 12, 0, 0, ph, 0, 0);
cpcn;   lft $_{10} x_2 =$ round $u$;   $x_1 = x_3 = r - x_2$;   $\mathrm{top}_{10} y_1 = h$ ; $\mathrm{bot}_{10} y_2 = 0$ ;   $y_3 = y_2$;
$w_{10}$ draw 1.. 2.. 2.. 3.                                          % diagonal and bar

"Universal quantifier";
call $charbegin$( '070, 10, 0, 0, $ph$, 0, 0);
cyan;   $\mathrm{lft}_{10} x_1 = 0$ ; $x_2 =$ good,,, $.5r = 5[x_1, x_3]$;
$\mathrm{top}_{10} y_1 = h$;   $y_3 = y_1$; $\mathrm{bot}_{10} y_2 = -o$;   $y_4 = y_5$ = $\mathrm{good}_{10} m$;
new $aa, bb$;   $x_4 = aa[x_1, x_2]$;   $y_4 = aa[y_1, y_2]$;   $x_5 = bb[x_3, x_2]$;   $y_5 = bb[y_3, y_2]$;
$w_{10}$ draw 1.. 2.. 2.. 3;                                          % diagonals
draw 4 . . 5.                                                         % bar

"Existential quantifier";
call $charbegin$( '071, LO, 0, 0, ph, 0, 0);
cpen;   lft $_{10} x_1 =$ round $u$;   $\mathrm{rt}_{10} x_2 =$ round$(r - u)$;   $x_3 = x_5 = x_2$;   $x_4 = x_6$ - - $.25u = x_1$ ;
$\mathrm{top}_{10} y_1 = h$;   $\mathrm{bot}_{10} y_4 = 0$;   $y_2 = y_1$   $y_5 = y_6 = .5[y_1, y_3]$;   $y_3 = y_4$;
$w_{10}$ draw 1.. 2 . . 2.. 3.. 3 . . 4;                              % upper bar. stem, lower bar
draw 5. . 6.                                                          % middle bar

"Logical NOT";
call $charbegin$( '072, 12, 0, 0, px, 0, 0);
cpen;   $\mathrm{lft}_{10} x_1$ = round $u$;   $x_2 = x_3 = r - x_1$;
$y_1 = y_2$ = good,, $.5[e, m]$;   $y_2 - y_3 = 1.2(m - e)$;
$w_{10}$ draw 1 . . 2.. 2.. 3.                                          % bar and stem
```

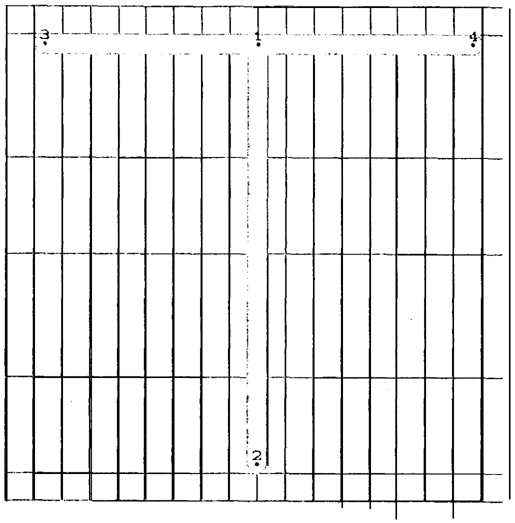


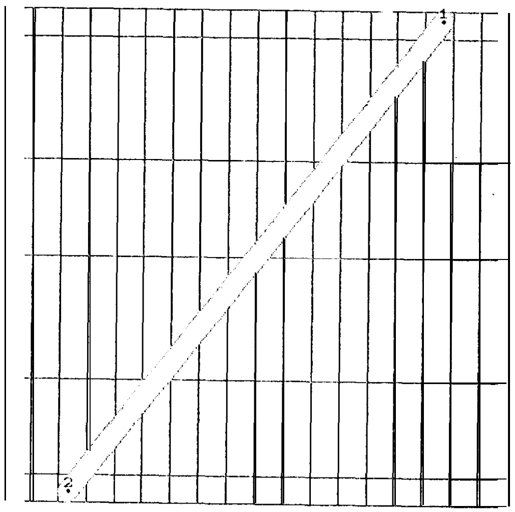
```
"Hebrew letter aleph";
call charbegin( '073, 11, 0, 0, ph,0,0);
vpen;  lft_6 x_1 == round u;  y_1 == 2/3[m,h];
x_4 + Xj == x_3 + x_6 = x_2 + x_7 == x_0 + x_9 == x_1 + x_8 = r;    x_2 = x_3 == 2.5u;
new aa;  x_2 = aa[x_1, x_8];  y_50 = aa[y_1, y_8];
top_5 y_50 = top_6 y_2;  bot_5 y_50 = bot_6 y_3;
x_0 = 1.5u;  top_6 y_0 = h;  y_4 == y_0;  x_4 == 5u;
y_4 + y_5 == y_3 + y_6 = y_2 + y_7 = y_1 + y_8 == y_0 + y_9 = h;
w_6 ddraw 0 0{0, —1}..2{x_8 — x_1, y_8 — y_1} 6{x_8 — x_1, y_8 — y_1}.8{0, - - 1}  .9(. 5),
    (4. .)0..1{0,—1}..3{x_8 --- x_1, y_8 – y_1}.7{x_8 — x_1, y_8 — y_1}. .
        9{0,—1}..9;                                                       % long diagonal
hpen;  x_10 == 3 5u;  new aa;  x_10 = aa[x_1, x_8];  y_10 == aa[y_1, y_8];
new w_99;  w_99 = round .5[w_0, w_1];
x_11 = good,,, 2u;  y_11 = e;
x_12 == .5[x_1 I, x_13];  y_12 == .5[y_1 I, y_13];
x_13 == good, 3.5u;  y_13 == .2e;
x_14 = good_5 2u;  bot_5 y_14 =0;
draw |w_0#|10{y_8 — y_1, x_1 — x_8}. |w_99|11{0, —1}..
    |5[w_99, w_5]|12{ 1.5(x_13 — x_11), y_13 — y_11}..
    |w_5#|13{0, - 1 ) . . 14{—1, 0};                                   %lower left stroke
x_15 == 7.75u;  new aa;  x_15 = aa[x_1, x_8];  y_15 == aa[y_1, y_8];
x_16 = .5[x_21, x_28];  y_16 = .5[y_21, y_28];
vpen; lft_6 x_21 = round 7u; y_21 = y_1;
x_20 + x_29 = x_21 + x_28 = x_22 + x_27 == x_24 + x_25;  x_28 == x_8;
y_20 + y_29 = y_2 I + y_28 == y_22 + y_27 = y_2 I + y_25;  y_28 == 25[m, h];
top_5 y_16 == top_6 y_22;  x_22 = x_27;
x_20 == 7.5u;  x_24 == 9.5u;  y_20 = y_24 = y_0;
w_6 ddraw 2 0 . 20{0, -I}. .22{x_28 -- x_21, y_28 — y_21}..28{0, -- 1}..29(..25),
    (24..)20..21{0, —1}..27{x_28 — x_21, y_28 — y_21}..
        29{0,—1}..29;                                                     % short diagonal
d r a w 15{0, 1}.16{0, 1}.                                                         % link
```

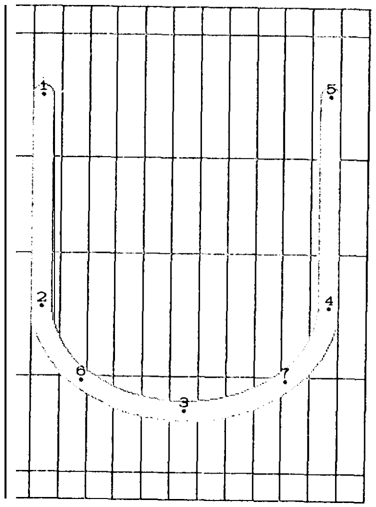


```
"Upper case Fraktur R.";
call charbegin( '074, 13, 0, 0, ph, 0, 0);
new w99;   w99 = round .25[w0, w1];
hpcn; lft0x1 = round u;   y1 = 1/3[e, m]; x2 = 2u; y2 = e ;
w0 draw 1{0, -1} . 2{1, 0};                                % point of upper lefl flourish
x3 = 2.5u;   top0y3 = h + oo;   lft99x4 = round u;
x5 = 2u;   y5 = 1/3[m, h];   rt99x6 = round 3.5u;
call ` a sdraw(3, 4, 5, 6, 2, w99, w8, -(h - e)/(8u));                            % flourish
x7 = x8 = good, 5.5u;   y7 = m;   y8 = e;
call `b arc(3, 7, w4);                                                         % upper link
w4 draw 7. .8;                                                                       % stem
x9 = 3.5u;   bot0y9 = -oo;   call . c arc(9, 8, w4);                           % lower link
x10 = 5/3u;   y10 = .3e;   x11 = 4/3u;
x12 = u;   y12 = .35e;   x13 = 2u;   y13 = .45e;
new aa;   x11 = x10 + aa(y12 - y13);
y11 = y10 + aa(x13 - x12);                                          % perpendicular to 12..13
w0 draw 9{-1, 0} 10{x11 - x10, y11 - y10} . . 11;                                    % tail
draw 12 . . 13;                                                                     % cross
x14 = x7;   y14 = .75h;   x15 = 9.5u;   top99y15 = h + o;
draw 14{0, 1} . 15{1, 0};                                                     % top of bowl
lft4x16 = 11.5u;   y16 = .6h;
w4 draw 15. 16;                                                          % diagonal of bowl
lft99x17 = lft4x16;   y17 = y16;
new aa, hb;   rt99x18 = aa[rt4x15, rt4x16];   y18 = aa[y15, y16];
x18 = x17 + bb(y15 - y16);   y18 = y17 + bb(x16 - x15);         % perpendicular to 15..16
rpen#; w4 draw 17.. 18;                                                      % crase excess
x20 = x7;   y20 = y21 = .52h;   x21 = 9u;
hpen;   w99 draw 20 . . 21{1, 0} . 17{x18 - x17, y18 - y17} . 18;            % bottom of bowl
x22 = 9.25u;   y22 = .5e;   lft99x21 = lft4x23;   bot4y23 = -o;
x24 = 10.5u;   y24 = y23;   rt99x25 = 12.5u;   y25 = e/4;
w4 draw 21 . 22{x22 - - x21, y22 - y21} 23{1, 0};                          % lower diagonal
rpen#;   w4 draw 24. 25;                                       % erase excess at lower right
hpen;   w99 draw 24.. 25.                                     % point of lower right diagonal
```

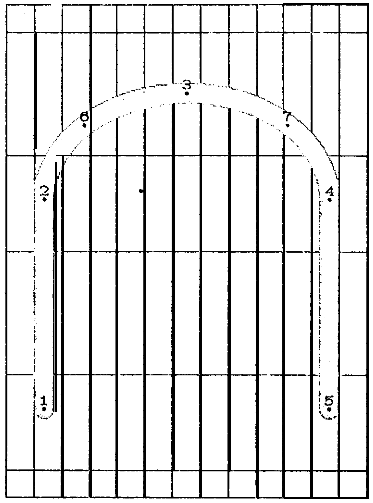


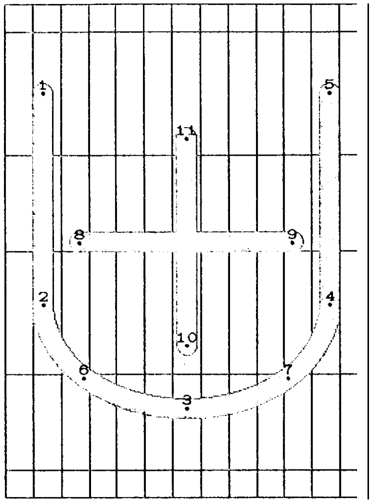


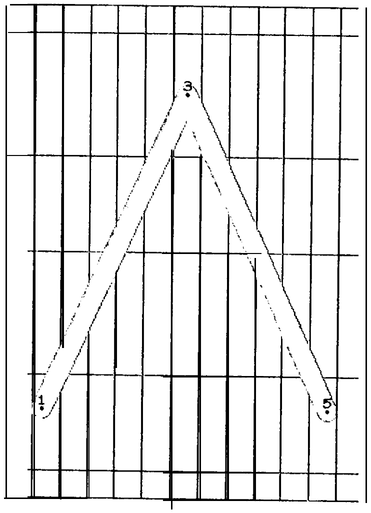


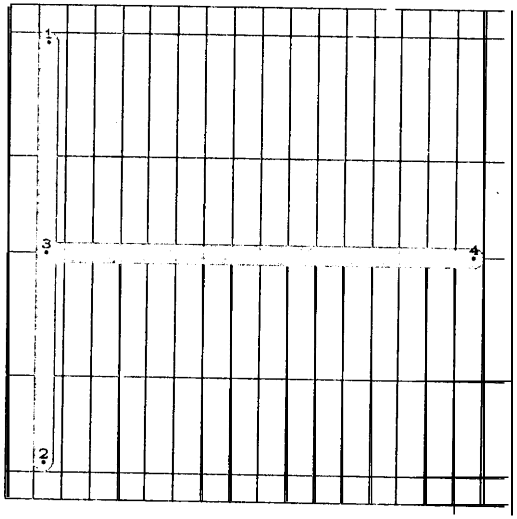


```
'Upper case Fraktur I";
call charbegin('075, 13, 0, 0, ph, 0, 0);
% (this letter extended to be same width as the R)
new w_99;   w_99 == round .25[w_0, w_1];
hpen;   x_1 == 5u;   y_1 == good_6 .5h;   lft_99 x_2 == round u;   y_2 == .75h;
x_3 == x_1;   top_99 y_3 == h + oo;
w_99 draw 1{--1, 0} . . 2{0, 1} . . 3{1, 0};                               % shoulder
x_4 == 9u;   y_4 == .75h;   x_5 == 10.5u;   bot_99 y_5 == 2/3 h + 1;
rt_99 x_6 == 12.5u;   bot_99 y_6 == .75h + 1;
draw |w_99#|3{1, 0} . . |w_1#|4{6u, --h} . . |w_99#|5{1, 0} . . 6{0, 1};      % upper stroke
lft_1 x_7 == round u;   y_7 == good_6 .25h;
x_8 == 7u;   bot_1 y_8 == --oo;
x_9 == good_1 10.5u;   y_9 == h/6;
x_10 == good_1 8.5u;   y_10 == .45h;
x_11 == 1/sqrttwo[x_12, x_10];   y_11 == 1/sqrttwo[y_10, y_12];
x_12 == x_9;   top_99 y_12 == .625h;
rt_99 x_13 == round 12.5u;   y_13 == .5h;
draw |w_1|7{1, 0} . . 8{1, 0} . . 9{0, 1} . . |w_1#|10{0, 1} . .
     |2/3[w_99, w_1]|11{x_12 -- x_10, y_12 -- y_10} . . |w_99#|12 . . 13{0, --1}.   % lower stroke

"Lattice Lop";
cpen;
if fixwidth== 0:   i f pa +- 8pu > ph:
          call charbegin('076,18, 0, 0, ph, ph - -2pa, 0);   top_10 y_1 == h;
     else:   call charbegin('076,18, 0, 0, 8pu + pa, 8pu - pa, 0);   top_10 y_1 = a +- 8u;
     fi;
else:   call charbegin('076, 9, 0, 0,3.5pu + pa,35pu - pa, 0);   top_10 y_1 == a +- 3.5u;
fi;
.5[y_1, y_2] = a,   x_1 == x_2 == .5r;
lft_10 x_3 == round u;   x_4 = r -- x_3;   y_3 == y_4 == y_1;
w_10   draw   I   2;                                                      % stem
draw 3 . . 4.                                                             % bar

"Zero-width slash to negate a relation";
call charbegin( '100, 18, 0, 0, ph +- pb, ph + pb --2pa, 0);
charwd 0 ; chardw 0 ;
cpen;   rt_10 x_1 = round(r - - 2u);   lft_10 x_2 == round 2u;
top_10 y_1 == h +- b;   5[y_1 , y_2] = a;
w_10   draw   1   2.                                                      % diagonal

"Set union sign";
call charbegin('133, I 3, 0, 0, ph, ph -- 2pa, 0);
cpen;   lft_10 x_1 == round u;   x_2 == x_1;   x_3 == r -- x_1;   x_4 == x_5 == r -- x_1;
y_1 == good_10(.5[m, h]);   .5[y_1, y_3] = a ;   y_2 == y_4 == 2/3[y_1, y_3];   y_5 == y_1;
call qcirc(3, 6, 2, w_10);   call qcirc(3, 7, 4, w_10);                      % cup
w_10 draw 1 . . 2; draw 4 . . 5.                                          % sterns
```

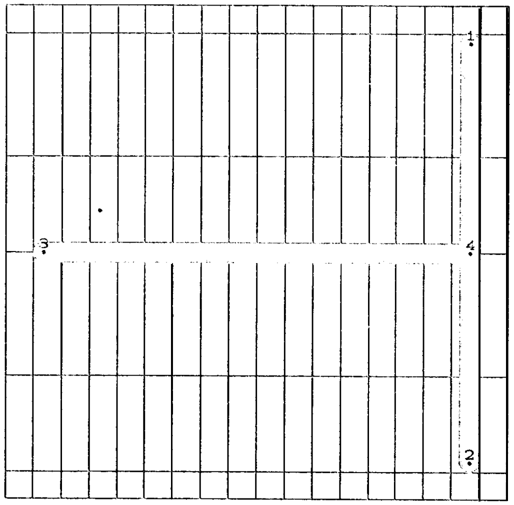


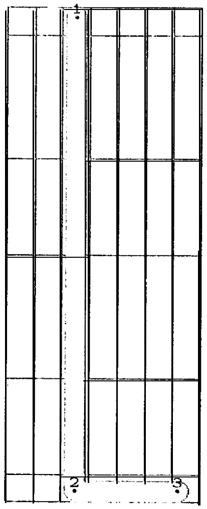


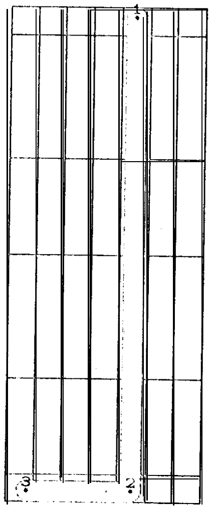


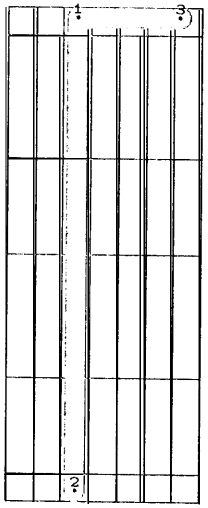


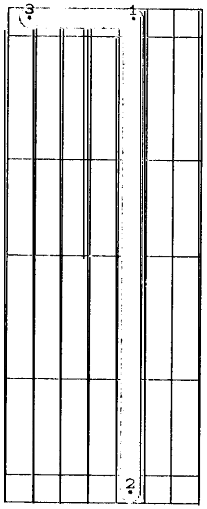
```
“Set intersection sign”;
call $charbegin$(´ 134, 13, 0, 0, ph, ph $-2pa$, 0);
cpen;   lft $_{10}x_1 =$ round $u$;   $x_2 = x_1$;   $x_3 = r - x_3$;   $x_4 =$ xj $= r - x_1$;
$y_3 =$ good$_{10}$(.5$[m, h]$);   .5$[y_1, y_3] = a$;   $y_2 = y_4 = \frac{2}{3}[y_1, y_3]$;   $y_5 = y_1$;
call $qcirc(3, 6, 2, w_{10})$; call $qcirc(3, 7, 4, w_{10})$;                      % cap
$w_{10}$ draw 1 . . 2;   draw 4.. 5.                                                % stems

“Multiset union sign”;
call $charbegin$(´135,13, 0, 0, ph, ph $-2pa$, 0);
cpcn;   lft$_{10}x_1 =$ round $u$;   $x_2 = x_1$;   $x_3 = r - x_3$;   $x_4 = x_5 = r - x_1$;
$y_1 =$ good$_{10}$(.5$[m, h]$);   .5$[y_1, y_3] = a$;   $y_2 = y_4 = \frac{2}{3}[y_1, y_3]$;   $y_5 = y_1$;
call $qcirc(3, 6, 2, w_{10})$;   call $qcirc(3, 7, 4, w_{10})$;                      % cup
$w_{10}$ draw 1 . 2;   draw 4 . . 5;                                                % stems
$y_8 = y_9 = .47[y_1, y_3]$;   $x_8 = r - x_9 = x_1 + 1.75w_{10} - eps$;
$x_{10} = x_{11} = x_3$;   .5$[y_{10}, y_{11}] = y_8$;   $y_{11} -- y_{10} = 5 -- x_8$;
draw 8 . . 9; draw 10 . . 11.                                             % enclosed plus sign

“Lattice infimum (logical AND) sign”;
call $charbegin$(´136, 13, 0, 0, ph, $ph - 2pa$, 0);
cpen;   lft$_{10}x_1 =$ round $u$;   $x_3 = r - x_3$;   $x_5 = r - x_1$;
$y_3 =$ $good$ $_{10}$(.5$[m, h]$);   .5$[y_1, y_3] = a$; $y_5 = y_1$;
$w_{10}$ draw 1 . . 3 . . 3 . . 5.                                                  % diagonals

“Lattice supremum (logical OR) sign”;
call $charbegin$( ‘137, 13, 0, 0, ph, $ph - 2pa$, 0);
cpen;   Ift $_{10}x_1 =$ round $u$;   $x_3 = r - x_3$;   $x_5 = r - x_1$;
$y_1 =$ good$_{10}$(.5$[m, h]$);   .5$[y_1, y_3] = a$;   $y_5 = y_1$;
$w_{10}$ draw 1 . . 3 . . 3 . . 5.                                                  % diagonals

“Left turnstile”;
cpen;
if $fixwidth$ = 0:   if pa $+ 8pu$ > ph:
            call $charbegin$(´140,18, 0, 0, $ph$, ph $-2pa$, 0);    top$_{10}y_1 = h$;
      else:   call $charbegin$( ‘140, 18, 0, 0, $8pu$ + $pa$, $8pu - pa$, 0);   top$_{10}y_1 = a + 8u$;
      fi;
else:   call $charbegin$(´140, 9, 0, 0, $3.5pu + pa$, $3.5pu - pa$, 0);   top$_{10}y_1$ = a $+ 3.5u$;
fi;
.5$[y_1, y_2] = a$;   $x_1 = x_2 = x_3$;
lft $_{10}x_3 =$ round $u$;   $x_4 = r -- x_3$;   $y_3 = y_4 =$ a ;
$w_{10}$ draw 1 . . 2;                                                              % stem
draw 3.. 4.                                                                         % bar
```

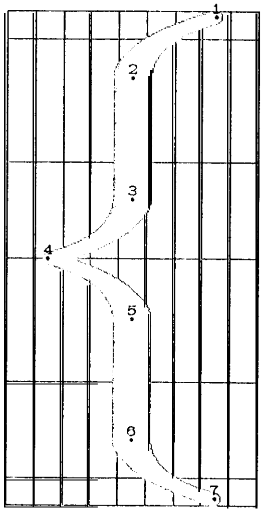




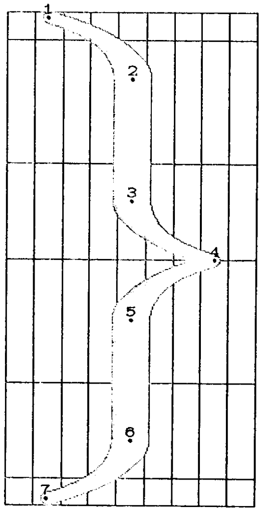


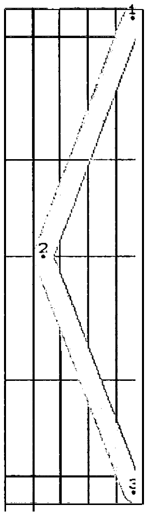


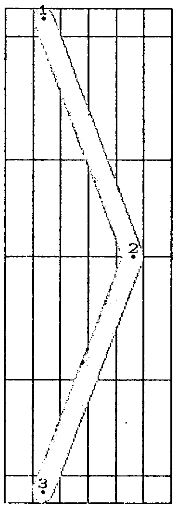


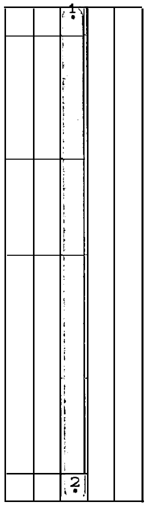


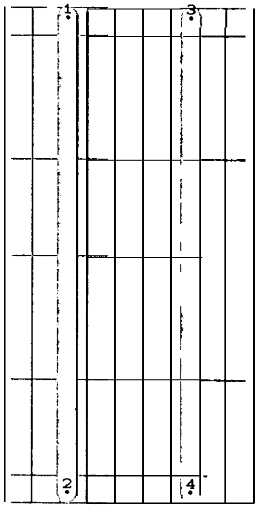
```
"Right  turnstile";
cpcn;
if $fixwidth$ = 0:   if $pa + 8pu > ph$:
            call $charbegin$('141, 18, 0, 0, $ph, ph - 2pa$, 0);    $top_{10}y_1 = h$ ;
      else:   call $charbegin$('141,18, 0, 0, $8pu +$ pa, $8pu$ -- $pa$, 0);    $top_{10}y_1 = a + 8u$;
      fi;
else:   call $charbegin$('141, 9, 0, 0, $3.5pu + pa, 3.5pu -$ pa, 0);    $top_{10}y_1 = a + 3.5u$;
fi;
$.5[y_1, y_2] = a$;    $x_1 = x_2 = x_4$;
$lft_{10}x_3 =$ round $u$;    $x_4 = r - x_3$;    $y_3 = y_4 =$ a;
$w_{10}$ draw 1 . . 2;                                                         % stem
draw 3 . 4.                                                                    % bar

"Left floor bracket";
call $charbegin$(' 112, 7, 0, 0, $ph + pb, ph + pb$ -- $2pa$, 0);
cpen;   $x_1 = x_2 =$ good $_{10}2.5u$;    $x_3 = x_1 + 3.75u + eps$;
$top_{10}y_1 = h + b$;    $5[y_1, y_2] = a$;    $y_3 = y_2$;
$w_{10}$   draw  1  2.  2.  3.                                                 % stem and bar

"Right floor bracket";
call $charbegin$('143, 7, 0, 0, $ph + pb, ph + pb - 2pa$, 0);
cpen;   $x_1 = x_2 = good_{10}(r - 2.5u)$;    $x_3 = x_1$ -- $3.75u -$ eps;
$top_{10}y_1 = h + b$;    $5[y_1, y_2] = a$;    $y_3 = y_2$;
$w_{10}$ draw 1 . . 2..  2..  3.                                               % stern and bar

"Left ceiling bracket";
call $charbegin$(' 144, 7, 0, 0, $ph + pb, ph + pb - 2pa$, $0$);
cpen;   $x_1 = x_2 =$ good $_{10}$ 2.5,;    $x_3 = x_1 + 3.75u + cps$;
$top_{10}y_1 = h + b$;    $.5[y_1, y_2] = a$;    $y_3 = y_1$;
$w_{10}$ draw 3 . . 1 . . 1..  2.                                              % bar and stem

"Right  ceiling  bracket";
call $charbegin$('145, 7, 0, 0, $ph + pb, ph + pb$ -- $2pa$, $0$);
cpen;   $x_1 = x_2 =$ good $_{10}(r$ -- $2.5u)$;    $x_3 = x_1 - 3.75u -$ eps;
$top_{10}y_1 = h + b$;    $.5[y_1, y_2] =$ a;    $y_3 = y_1$;
$w_{10}$ draw  3..  1..  1 . . 2.                                              % bar and stem

"Left brace";
call $charbegin$('146, 9, 0, 0, $ph + pb, ph + pb - 2pa, (ph + pb)\cdot slant$ + $.5pw$ - - $pu$);
hpen;   $x_2 = x_3 =$ x-, $= x_6 = good_1 5r$;    $x_1 - x_2 = x_2 - x_1 = 3u + eps$;    $x_1 = x_7$;
$top_0y_1 = h + 6$;    $y_1 = .5[y_1, y_7] = .5[y_2, y_6] = .5[y_3, y_5] = good_6a$;
$y_1 - y_2 = y_3$ - - $y_4 = (y_1 - y_1)/4$;
draw $|w_0\#|1\{3(x_2 - x_1), y_2 - y_1\}$  $|w_1\#|2\{0, -1\}$ .  $|w_1\#|3\{0, -1\}$ .
      $|w_0\#|4\{3(x_1 - x_3), y_1$ -- $y_3\}$;                                 % upper stem
draw $|w_0\#|7\{3(x_6$ - - $x_7), y_6 - y_7\}$ . $|w_1\#|6\{0, 1\}$ $|w_1\#|5\{0, 1\}$ . .
      $|w_0\#|4\{3(x_4 - x_5), y_4 - y_5\}$.                                    % lower stem
```

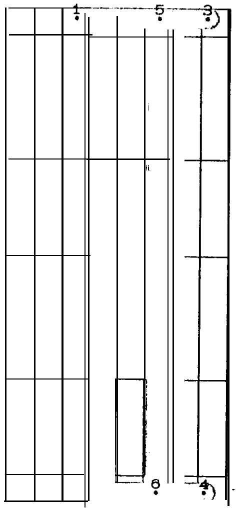


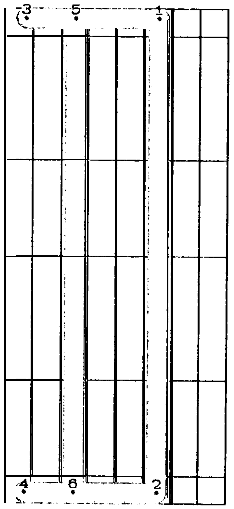












'Right brace";
call *charbegin*('147, 9, 0, 0, ph $+ pb, ph +$ pb $- 2pa$, (ph $+ pb)\cdot slant$ + $5pwi - 4pu$);
hpen; $x_2 = x_3 = x_5 = x_6 =$ good$_1$ $.5r$; $x_1 - x_2 = x_2 - x_4 = -3u - eps$; $x_1 = x_7$;
$top_0 y_1 = h + 6$; $y_4 = .5[y_1, y_7] = .5[y_2, y_6] = .5[y_3, y_5] =$ good$_6 a$;
$y_1 - y_2 = y_3 - y_4 = (y_1 - y_4)/4$;
draw $|w_0\#|1\{3(x_2 - x_1), y_2 - y_1\}$. $|w_1\#|2\{0, -1\}$ . . $|w_1\#|3\{0, -1\}$.
    $|w_0\#|4\{3(x_4 - x_3), y_4 - y_3\}$; % upper stern
draw $|w_0\#|7\{3(x_6 - x_7), y_6 - y_7\}$. $|w_1\#|6\{0, 1\}$. $|w_1\#|5\{0, 1\}$.
    $|w_0\#|4\{3(x_4 - x_5), y_4 - y_5\}$. % lower stem

"Left angle bracket";
call *charbegin*('150, 6, 0, 0, $ph + pb, ph + pb - 2pa, 0$);
cpen; $rt_{10}x_1$ = round($r - u$); $x_1 = x_3$; $lft_{10}x_2$ = round *fixwidth*$[u, \frac{7}{3}u]$;
$top_{10}y_1 = h + 6$; $.5[y_1, y_3] = y_2 =$ good $_{10}a$;
w_{10} *draw* 1 . . 2.. 2.. 3. % diagonals

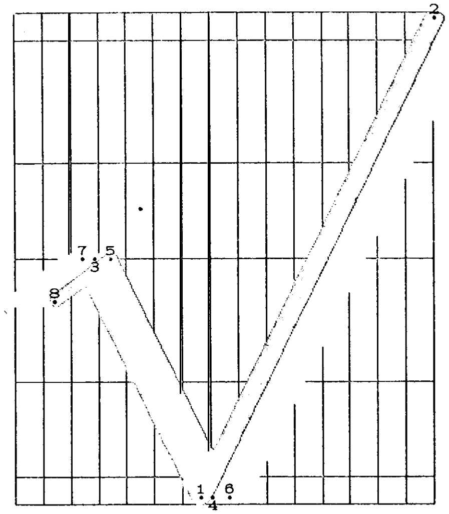
"Right angle bracket";
call *charbegin*('15 1,6, 0, 0, $ph + pb$, ph + pb $- 2pa$, 0);
cpcn; $rt_{10}(r - x_1) =$ round$(r - u)$; $x_1 = x_3$; $lft_{10}(r - x_2) =$ round *fixwidth*$[u, \frac{7}{3}u]$;
$top_{10}y_1 = h + b$; $.5[y_1, y_3] = y_2 =$ good$_{10}a$;
w_{10} draw 1 . . 2 . . 2 . . 3. % diagonals

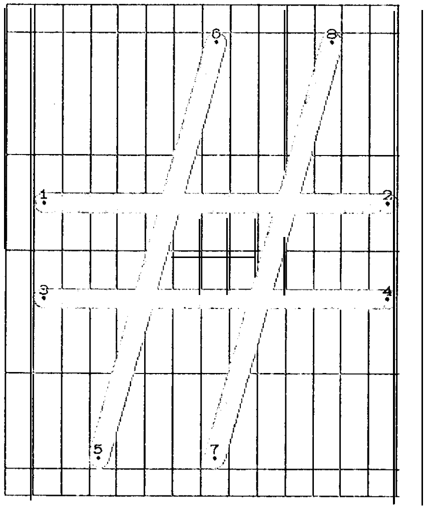
"Vertical line (absolute value or length)";
call *charbegin*('152, 5, 0, 0, $ph +$ pb, ph + $pb - 2pa$, 0);
cpen; $x_1 = x_2 =$ good $_{10}.5r$; $top_{10}y_1 = h + b$; $.5[y_1, y_2] = a$;
w_{10} draw 1 . . 2. % stem

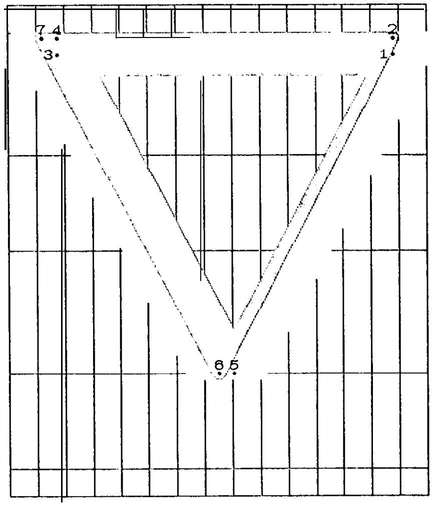
"Double vertical line (norm or cardinality)";
call *charbegin*('153, 9, 0, 0, $ph + pb, ph +$ pb $- 2pa$, 0);
cpen; $x_1 = x_2 =$ *good* $_{10}.25r$; $top_{10}y_1 = h + b$; $.5[y_1, y_2] = a$;
$x_3 = x_4 = r - x_1$; $Y.l = y_1$; $y_4 = y_2$;
w_{10} draw 1.. 2; draw 3.. 4. % stems

"Double left bracket";
call *charbegin*('154, 8, 0, 0, ph $+ pb$, ph + pb $- 2pa$, 0);
cpen; $x_1 = x_2 =$ good$_{10}2.5u$; $x_3 = x_4 = x_1 + 4.75u$ + eps; $x_5 = x_6 = x_1 +$ round $3u$;
$top_{10}y_1 = h + 6$; $.5[y_1, y_2] = a$; $y_3 = y_5 = y_1$; $Y\ I = y_6 = y_2$;
w_{10} draw 3. 1 l 2. 2 . 4; % bars and left stem
draw 5.. 6. % right stem

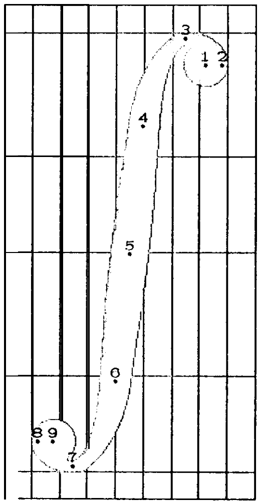
"Double right bracket";
call *charbegin*('155, 8, 0, 0, $ph +$ pb, $ph +$ pb $- 2pa$, 0);
cpen; $x_1 = x_2 =$ good$_{10}(r - 2.5u)$;
$x_3 = x_4 = x_1 - 4.75u -$ eps; $X\ j = x_6 = x_1 -$ round $3u$;
$top_{10}y_1 = h + b$; $.5[y_1, y_2] = a$; $y_3 = y_5 = y_1$; $y_4 = y_6 = y_2$;
w_{10} draw 3.. 1 . . 1 . . 2.. 2.. 4; % bars and right stem
draw 5 . . 6. % left stem

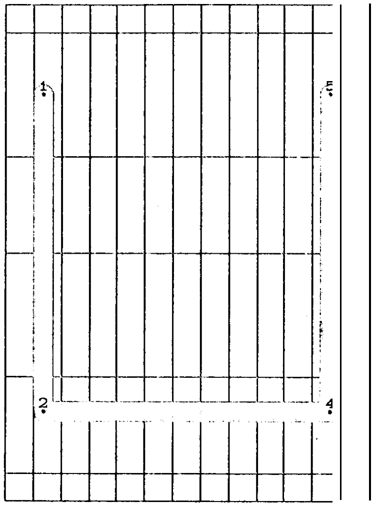


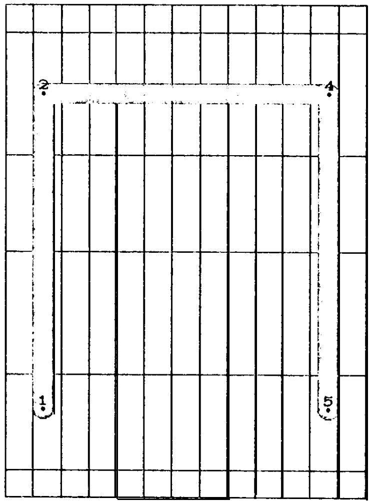


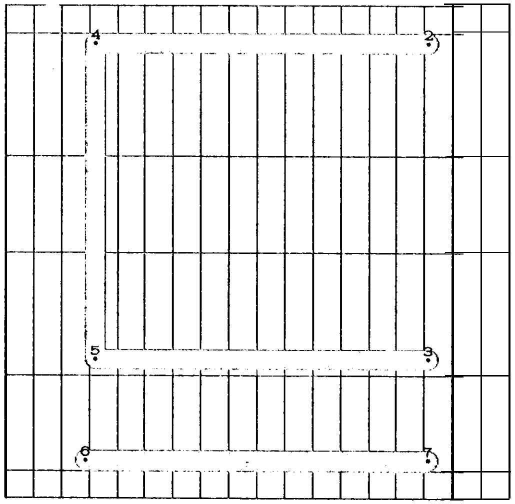


```
"Radical sign";
call charbegin(' 160, 15, 0, 0, ph |-pb, ph+pb—2pa, 0);
hpen;   x1 = good10(20/3 u);   x2 = r + 1;   top10y2 = h + 6;   .5[y1, y2] = a;
y3 = y5 = y7 = good6a;   y4 = y6 = y1;
x7 = 1.5[x2, x1];   lft10x7 = lft2x3;   rt2x3 = rt0x5;
lft10x1 = lft2x4;   rt2x4 = rt0x6;
x8 = x7 — u;   new aa;   x8 = aa[x5, x2];   y8 = aa[y5, y2];
hpen;   w2 draw 3. . 4;                                         % left diagonal
w10 draw 7.. 4;   w0 draw 5.. 6;                         % sharpen the corners
lpen#;   w2 draw 8..5;                               % erase excess at upper left
hpen; w0 draw 8.. 5;                                                   % serif
rpen#;   w2 draw l . . 2;                          % erase excess al lower right
hpcn;   w10 draw 1 . . 2.                                      % right diagonal

"Sharp symbol (number sign or hash mark)";
call charbegin( '16 l, 15, 0, 0, ph, ph—2pa, 0);
cpen;   Ift 10x1 = round u;   x3 = x1;   x2 = x4 = r — x1;
y1 = y2;   y3 = y4;   y1 -- y3 = round(m — e),   5[y1, y3] = a;
w10   draw   l   2;                                                 % upper bar
draw 3.. 4;                                                         % lower bar
x5 — 2u = x1;   x8 + 2u = x2;   x6 — x5 = x8 — x7;   x6 -- x7 = fixwidth[0, —3u];
y5 = y7;   y6 = y8;   top 10y6 = h;   .5[y5, y6] = a;
draw 5..6;                                                      % left diagonal
draw 7..8.                                                     % right diagonal

"Nabla or backwards-difference operator";
call charbegin('162,15, 0, 0, ph, 0, 0);
hpen;   rt0x2 = round(r — u);   lft 1x4 = round u;   top0y2 = h;   top 1y1 = h;
bot 1y5 = —o;   y6 = y5;
x6 — x2 = x4 — x5;   lft 1x5 = lft0x6;   x1 = x2;   x3 = x4;
vpcn;   top8y1 = h;   y3 = y1;
w9   draw   1   3;                                                   % bar line
lpen#;   w9 draw 5. . 4;                             % erase excess at upper left
hpen;   w4 draw 5.. 4;                                          % left diagonal
rpen#; w4 draw 6..2;   w9 draw 6.. 2;                  % erase excess at right
hpen;   w0 draw 6.. 2;                                         % right diagonal
y7 = y4;   lft0x7 = Ift 1x 1;
w0 draw 2..7.7..6.                                % sharpen upper left corner
```

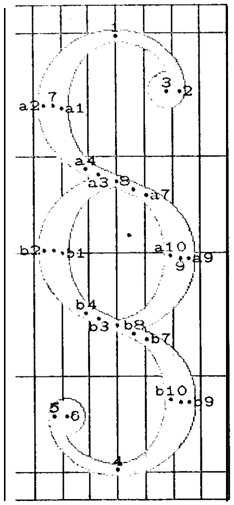


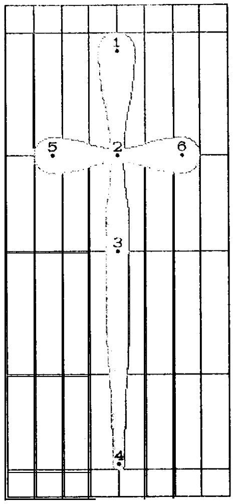


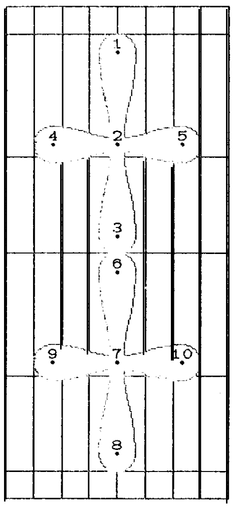


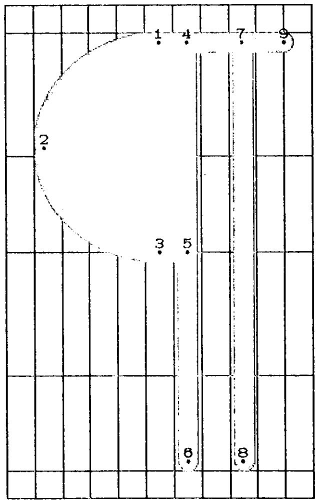


```
"Integral sign";
call charbegin( '163, 9, 0, 0, ph, pd, 0);
hpen;   rt_2 x_1 = rt_0 x_2 = round(r — u);   y_1 = y_2;
lft_2 x_9 == lft_0 x_8 == round u;   y_8 = y_9;
cpen;   bot_2 y_9 == —.9d;   y_8 -- y7 = y_3 — y_1;
hpen;   x_3 == r —  2.5u;   x_7 == 2.5u;
top_0 y_3 = h  ;   bot_0 y_7 = —d;
x_5 == good, .5r;   y_5 = .5[y_3, y_7];
x_4 = x_5 + .5u;   y_4 == y_5 + .3(y_3 — y_7);
x_6 = x_5 — .5u;   y_6 == y_5 — .3(y_3 -- y_7);
cpen;   w_2 draw 1;                                                % upper bulb
draw 9;                                                            % lower bulb
hpen;   draw |w_0|2{0, 1 } |w_0#|3{—1, 0 } |.75[w_0, w_1]|4{x_6 — x_4, y_6 -- y_4}. .
     |w_1 #|5 . |.75[w_0, w_1]|6{x_6 -- x_4, y_6 -- y_4} . . |w_0#|7{—1, 0} . 8{0, 1}.   % stem

"Square set union sign (Scott lub)";
call charbegin(' 164, 13, 0, 0, ph, ph — 2pa, 0);
cpen;   lft_10 x_1 == round u;   x_2 == x_1;   x_4 == x_5 == r — x_1;
y_1 == good_10(.5[m, h]);   .5[y_1, y_2] = a;   y_2 == y_4;   y_5 == y_1;
w_10 draw 1 . . 2.. 2.. 4.. 4 . . 5.                                % stroke

"Square set intersection sign (Scott glb)";
call charbegin('165,13, 0, 0, ph, ph -- 2pa, 0);
cpen;   Ift _10 x_1 == round u;   x_2 == x_1;   x_4 == x_5 == r — x_1;
y_2 == good_10(.5[m, h]);   .5[y_1, y_2] == a;   y_2 == y_4;   y_5 == y_1;
w_10 draw 1 . . 2 . . 2 . . 4 . . 4 . . 5.                          % stroke

"Square reflexive subset sign";
call charbegin(' 166, 18, 0, 0, ph, ph --- 2pa, 0);
cpcn;   lft_10 x_6 == round 2.5u;   x_2 = x_3 == x_7 == r -- x_6;   lft_10 x_1 = x_6;
top_10 y_2 = h;   .5[y_2, y_3] == y_1;   y_2 -- y_1 == (good,,, .5[m, h]) --- (good_10 a);
x_4 == x_5;   y_4 = y_2; y_5 == y_3;
w_10 draw 2 . . 4.. 4.. 5.. 5.. 3;                                  % slroke
y_6 == y_7;   bot_10 y_6 == 2a — h;
draw 6.. 7.                                                         % bar
```

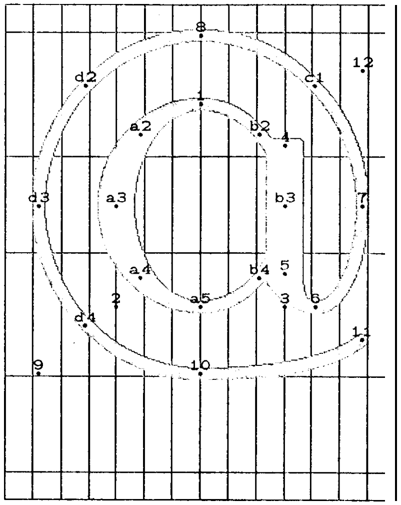


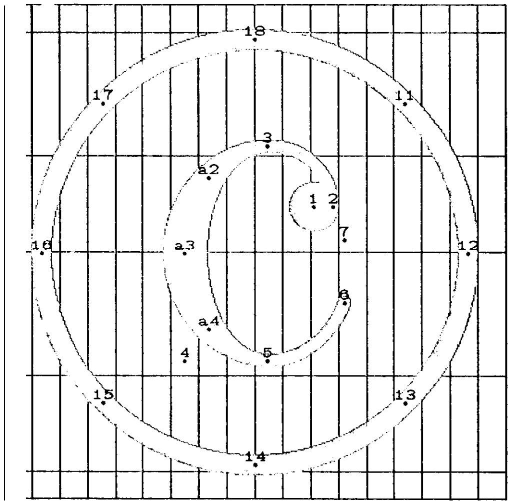




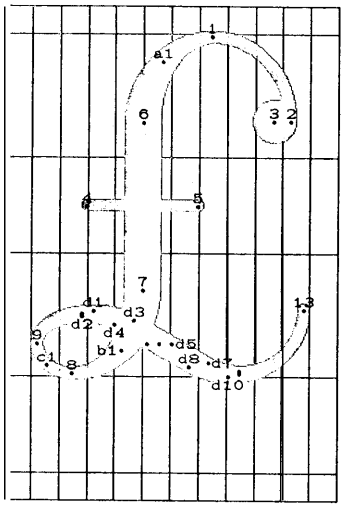


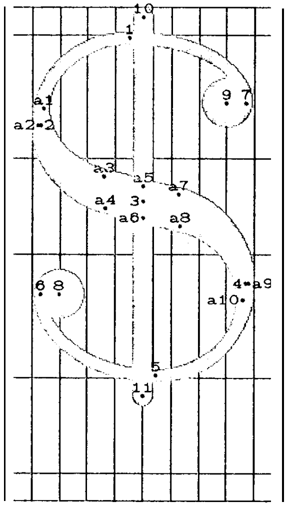
```
'Section sign";
call $charbegin$('170, 8, 0, 0, $ph$, $pd$, 0);
hpen;   $x_1 = x_4 = .5r$;   $\mathrm{rt}_0 x_2 = \mathrm{rt}_1 x_3 = \mathrm{round}(r - 1.5u)$;   $\mathrm{lft}_0 x_5 = \mathrm{lft}_1 x_6 = \mathrm{round}\,1.5u$;
$\mathrm{top}_0 y_1 = h + oo$;   $\mathrm{bot}_0 y_4 = -d - oo$;
$y_2 = y_3 = .125[y_1, y_4]$;   $y_5 = y_6 = .125[y_4, y_1]$;
$w_0$ draw 1{1, 0} . 2{0, -1};                                   % upper link
draw 4{-1, 0} . 5{0, 1};                                         % lower link
cpen;   $w_1$ draw 3;   draw 6;                                  % bulbs
$x_7 = x_{11} = x_5$;   $x_9 = x_{12} = x_2$;   $x_8 = x_{10} = x_1$;
$y_8 = \frac{1}{3}[y_1, y_4]$;   $y_{10} = \frac{1}{3}[y_4, y_1]$;
call 'a $sdraw(1, 7, 8, 9, 10, w_1 - deltaw, w_0, -(h + d)/(36u))$;   % upper stroke
call 'b $sdraw(8, 11, 10, 12, 4, w_1 - deltaw, w_0, -(h + d)/(36u))$.   % lower stroke

"Dagger mark";
call $charbegin$('171, 8, 0, 0, $ph$, $pd$, 0);
cpen;   $x_1 = x_2 = x_3 = x_1 = \mathrm{good}_1(.5r)$;   $\mathrm{lft}_1 x_5 = \mathrm{round}\,u$;   $x_6 - x_2 = x_2 - x_5$;
$\mathrm{top}_1 y_1 = h$;   $\mathrm{bot}_6 y_1 = -d$;   $y_2 = y_5 = y_6 = \mathrm{good}_1 m$;   $y_3 = e$;
$w_1$ draw 1;   draw 5;   draw 6;                                % bulbs
hpen;   draw $|w_1|1$ $|w_0|2$;                                  % top stem
draw $|w_0|2$ . . $|(\mathrm{round}\,.5[w_0, w_1]) - eps\#|3$ . $|w_0|4$;   % bottom stem
vpen;   draw $|w_1|5$ . $|w_0|2$;   draw $|w_1|6$ $|w_0|2$.      % bars

"Double dagger mark";
call $charbegin$('172, 8, 0, 0, $ph$, $pd$, 0);
cpen;   $x_1 = x_2 = x_3 = x_6 = x_7 = x_8 = \mathrm{good}_1(.5r)$;   $\mathrm{lft}_1 x_4 = \mathrm{round}\,u$;   $x_5 - x_2 = x_2 - x_4$;
$x_9 = x_4$;   $x_{10} = x_5$;
$\mathrm{top}_1 y_1 = h$;   $\mathrm{bot}_1 y_3 = .5[y_1, y_8] = \mathrm{top}_1 y_6$;   $\mathrm{bot}_1 y_8 = -d$;
$y_2 = y_4 = y_5 = \mathrm{good}_1 .5[y_1, y_3]$;   $y_7 = y_9 = y_{10} = \mathrm{good}_1 .5[y_6, y_8]$;
$w_1$ draw 1;   draw 3;   draw 4;   draw 5;   draw 6;   draw 8;   draw 9;   draw 10;% bulbs
hpen;   draw $|w_1|1$ $|w_0|2$;   draw $|w_1|3$ $|w_0|2$;
     draw $|w_1|6$ . . $|w_0|7$;   draw $|w_1|8$ . $|w_0|7$;     % stems
vpen;   draw $|w_1|4$ $|w_0|2$;   draw $|w_1|5$ . $|w_0|2$;
     draw $|w_1|9$ $|w_0|7$;   draw $|w_1|10$ . . $|w_0|7$.      % bars

"Paragraph mark";
call $charbegin$('173, 11, 0, 0, $ph$, $pd$, 0);
cpen;   $\mathrm{top}_{10} y_1 = h$;   $\mathrm{bot}_{10} y_6 = -d$;   $y_4 = y_7 = y_9 = y_1$;
$y_8 = y_6$,   $y_3 = y_5 = \mathrm{good}_{10} .5[y_1, y_6]$;
$\mathrm{lft}_{10} x_2 = \mathrm{round}\,u$;   $y_2 = .5[y_1, y_3]$;   $x_1 = x_3 = .5r$;   $x_4 = x_5 = x_6 = \mathrm{good}_{10} .5(r + 2u)$;
$x_7 = x_8 = x_6 + 2u$;   $x_9 = r - u$;
$w_{10}$ ddraw 4 . . 1{-1, 0} . . 2{0, -1} . . 3{1, 0} . 5, 4 4 . . 4 . . 5 . . 5;   % filled bowl
draw 6 . . 4 . 4 . . 9;                                          % left stem and upper serif
draw 7 . . 8.                                                    % right stem
```



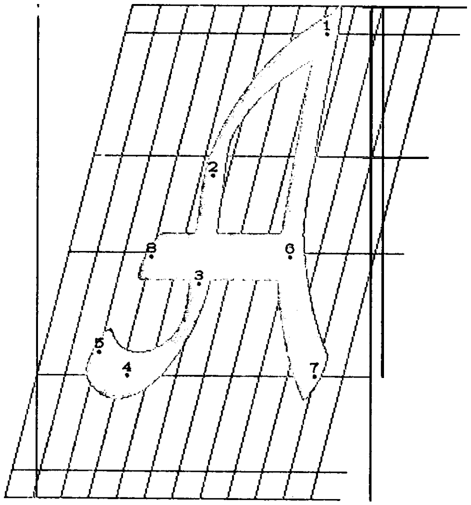


```
“At sign”;
call $charbegin$(´174, 14, 0, 0, $ph$, 0, 0);
hpen;  $x_1 = x_8 = x_{10} = .5r$;  $x_2 =$ good$_1$ 4$u$;  $x_3 = x_4 = x_5 = r - x_2$;
lft$_0x_9$ = round $u$;  x 7 $= x_{11} = x_{12} =$ 7 $- x_9$;  $x_6 = \frac{5}{12}[x_5, x_7]$;
top$_0y_8 = h + oo$;  bot$_0y_{10} = -oo$;  $y_9 = y_{10}$;
$y_1 =$ good$_6$ .8$[y_{10}, y_8]$;  $y_2 = y_3 = y_6 =$ good$_6$ .2$[y_{10}, y_8]$;
$y_7 = .5[y_{10}, y_8]$;  $y_{11} =$ good$_6$ 1$[y_{10}, y_8]$;  $y_{12} = .9[y_{10}, y_8]$;
$y_4 = .8[y_2, y_1]$;  $y_5 = \frac{2}{3}[y_7, y_6]$;
call ` a $darc$(1, 2, $w_1$);                                  % left part of inner bowl
call ` b $darc$(1, 3, $w_0$);                                 % right part of inner bowl
draw $|w_1|4\,..\,|w_1\#|5\{0, -1\}\,.\,|w_0\#|6\{1, 0\}\,.\,7\{0, 1\}$;              % stem and link
$call$ ` c arc(8, 7, $w_0$);                                  % right part of outer bowl
call ` d $darc$(8, 9, $w_0$);                                  % left part of outer bowl
$w_0$ draw 10{1, 0} . 11( . 12).                                               % point

“Copyright symbol”;
call $charbegin$(´175, 18, 0, 0, $ph$, $pd$, 0);
new $up$;  $up = .5((h - m) - d)$;                % amount to raise baseline of lowercase c
cpen;  rt$_3x_1$ = round$(r - 6u)$;
if top$_3$(top$_3$top$_6e + 2) > .9[e, m]$:  top$_3y_1 = .9[e, m] + up$;
else:  $y_1 =$ top$_3$top$_6e + 2 + up$;
fi;
$w_3$ draw 1;                                                              % bulb
hpen;  rt$_0x_2$ = rt$_3x_1$;  $y_2 = y_1$;  $x_3 = x_5 = .5(r + u)$;  top$_0y_3 = m + oo + up$;
$w_0$ draw 2{0, 1} . . 3{−1, 0};                                           % shoulder
if $w_2 > 1.5u$:  lft$_2x_4$ = round(5.75$u$);
else:  $x_4 =$ good$_2$ 6.5$u$;
fi;
$y_4 = y_5$;  bot$_0y_5 = -oo + up$;
call ` a $darc$(3, 4, $w_2$);                                                % stroke
if $w_0 = w_1$:  $x_6 = x_2$;  X 7 - - $x_5 =$ X j $- x_4$;  $y_7 = .5[y_3, y_5]$;
    new $aa$;  $x_6 = aa[x_5, x_7]$;  $y_6 =$ (sqrt(1 − $aa{\cdot}aa$))$[y_7, y_5]$;
else:  lft$_0x_6$ = rt$_0x_2$;  $y_6 = .5e - 1 + up$;  X 7 $= x_6$;  $y_7 = e$ + $up$;
fi;
$w_0$ draw 5{1, 0} . . 6( . . 7);                                            % point
cpen;  top$_{10}y_{18} = h + oo$;  bot$_{10}y_{14} = -d - oo$;
lft$_{10}x_{16}$ = round $u$;  rt$_{10}x_{12}$ = round$(r - u)$;
call $circle$(11, 12, 13, 14, 15, 16, 17, 18, $w_{10}$).                          % enclosing cirle
```





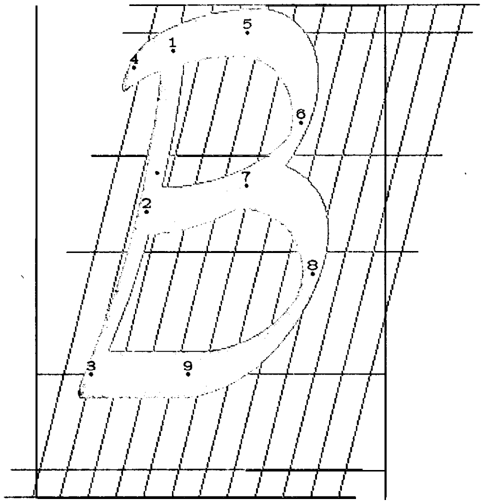
```
"Sterling sign";
call charbegin('176, 12, 0, 0, ph, 0, 0);
hpen;   x1 = 7.5u;   top0y1 = h + oo;   rt0x2 = rt2x3 = round(r — 1.5u);
y2 = y3 = .75h;   y4 = y5 = .5h;   x4 = 3u — eps;   Xj = 7u + eps;
x6 = x7 = good1 5u;   y6 = .75[y8, y1];   y7 = .25[y8, y1];   bot0y8 = —oo;
x8 = 2.5u;   lft0x9 = round u;   y9 = .1h;
w0 d r a w 1{1,0}. 2{0, —1};                                        % link
cpen;   w2 draw 3;                                                  % bulb
call `a arc(1, 6, w1);                                          % shoulder
hpen;   w1 draw 6 . . 7;                                            % stem
w0  draw  4  5;                                                      % bar
call `b arc(8, 7, w1);   call `c arc(8, 9, w0);                     % loop
top11y10 = round .2h;   bot11y12 = —oo;
x11 = .5[x8, x13] — u;   y11 = y9;
rt0x13 = round(r — u);   y13 = .2h;
call `d zdraw(9, 10, 11, 12, 13, w11, w7, —(x13 — x8 — 4u)/(.2h)).   % arm

"Dollar sign";
call charbegin('177, 10, 0, 0, ph + pb, pb, ph·slant — .5pu);
hpen;   top0y1 = h + oo;   bot0y5 = —oo;
x3 = good10 .5r;   y3 = .52h;   lft11x2 = round u;   x4 = r — x2;
if ucs = 0:   x1 = x5 = x3;   y6 = .5[y5, y3];   y7 = .5[y1, y3];
else:   if w6 = w9:   x1 = x5 = x3;   y6 = .5[y5, y3];   y7 = .5[y1, y3];
      else:   x1 + .5u = x5 — .5u = x3;   y6 = h/4 — 1;   y7 = .8h + 1;
      fi;
fi;
y8 = y6;   y9 = y7;   cpen;   lft3x8 = lft0x6;   rt3x9 = rt0x7;   x6 = x2;   x7 = x4;
w3 draw 8;                                                    % lower bulb
draw 9;                                                       % upper bulb
hpen;   w0 draw 6{0, —1} . 5{ 1, 0};                      % lower left stroke
draw 7{0, 1} . . 1{—1, 0};                               % upper right stroke
call `a sdraw( 1, 2, 3, 4, 5, w11, w9, —h/(50u));           % middle stroke
cpen;   x10 = x11 = x3;   top10y10 = h + b;   bot10y11 = —b;
w10 draw 10 . . 11.                                                 % stem
```

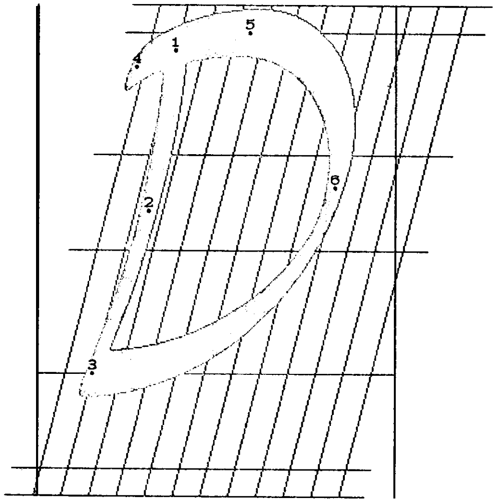


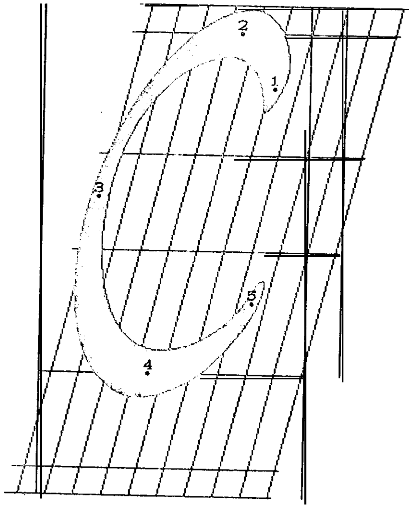
```
i.put symext;                          % possible characters ´067, ´077, ´156, ´157, ´167
```

The file script.mf

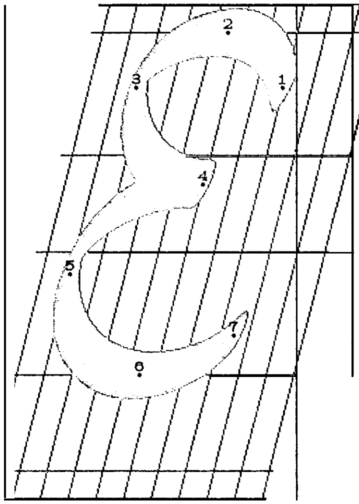
```
% This script alphabet is based on traditional Chancery style, except
% that descenders are eliminated, and the letters I, J, Q, T have been
% modified to accord with modern practice suitable for use with
% mathematics. All the letters are drawn with a single pen, whose
% dimensions are pw by pwv, rotated to the angle whose tangent is 2.
new aa, bb. aaa, bbb, spa, spb, spc;                   % variables to specify the spen
aa = (sqrt 1.25)(pwv·pixels + blacker);
bb = (sqrt 1.25)(pw·pixels + blacker);
aaa = 1/aa/aa;   bbb = 1/bb/bb;
spen(aaa + 4bbb, 4aaa − 4bbb, 4aaa + bbb, 0, 0, 0, 0);
no penreset;
new cor, corr;                                     % commonly used spacing corrections
cor = ph·slant/pu;
if cor > 1:   corr = 1;
else:   corr = cor ;
fi;

"Script  A";  spen;
call charbegin(`A,12, 0, 0, ph, 0, ph·slant − (3.75 − mi)pu);
x1 = 7.25u;   y1 = h;
x2 = 4.5u;   y2 = .59h;
x3 = 5u;   y3 = .27h;
x4 = 3.25u;   y4 = 0;
x5 = 2u;   y5 = .07h;
x6 = 8u;   y6 = .35h;
x7 = 10u;   y7 = 0;
x8 = 3u;   y8 = .35h;
draw 1{3(x2 − x1), y2 − y1} . . 2{x3 − x2, y3 − y2} . . 3{x3 − x2, y3 − y2}
     4{−1, 0} . . 5{x5 − x4, 3(y5 − y4)};                          % left diagonal
draw 1{0, 1} . . 6{x7 − x6, 1.5(y7 − y6)} . . 7{1.5(x7 − x6), y7 − y6};    % right diagonal
draw 8 . . 6.                                                              % bar
```

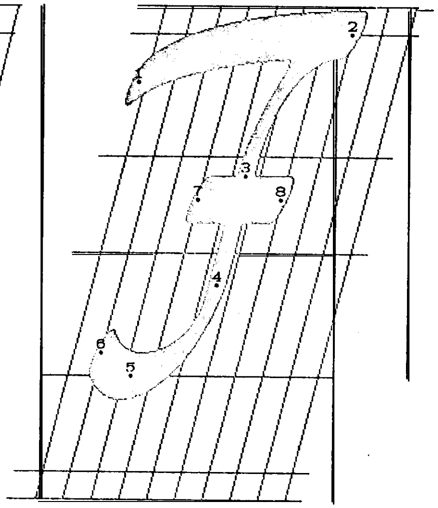


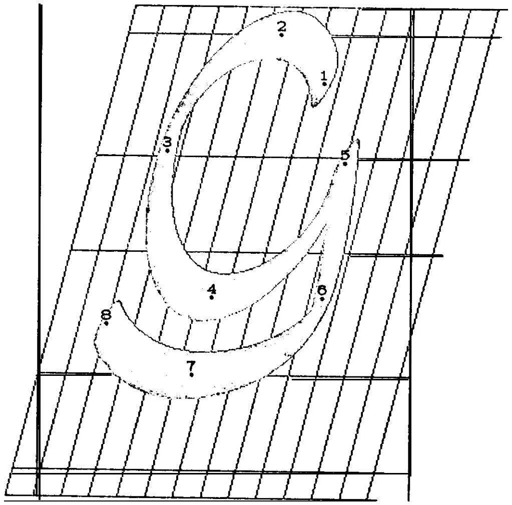




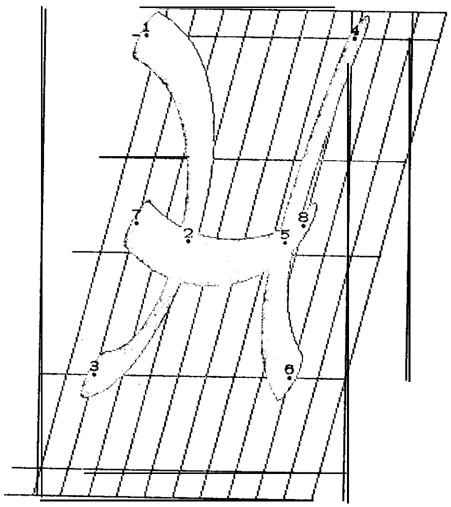
```
"Script B"; spen;
call charbegin(`B, 12, mi·corr, —.5mi·cor, ph, 0, mi[.5ph·slant — pu, 0]);
x1 = 3u;    y1 = .95h;
x2 = 3.5u;  y2 = .48h;
x3 = 3u;    y3 = 0;
x4 = 1.75u; y4 = .9h;
x5 = 5.5u;  y5 = h;
x6 = 8.25u; y6 = .74h;
x7 = (sqrt .5)[x2, x6];  y7 = (sqrt .5)[y6, y2];
x8 = 10u;   y8 = .3h;
x9 = 6.5u;  y9 = 0;
draw  1{2(x2 — x1), y2 — y1} . . 2{0, —1} . . 3{2(x3 — x2), y3 — y2};         % stem
draw 4{x5 — x4, 3(y5 — y4)} . 5{1, 0} . . 6{0, —1} . .
     7{x2 — x6, y2 — y6} . . 2{—1, 0};                                      % upper bowl
draw 7{1, 0} . . 8{0, —1} . . 9{—1, 0} . . 3.                               % lower bowl

"Script C"; spen;
call charbegin(`C, 10.5, .5mi·cor, —.2mi·cor, ph, 0,
               mi[ph·slant — 2.25pu, .8ph·slant — 1.25pu]);
x1 = 7.25u;  y1 = .84h;
x2 = 5.5u; y2 = h;
x3 = 2u;   y3 = .52h;
x4 = 5.5u; y4 = 0;
x5 = 8.5u; y5 = .21h;
draw 1{x2 — x1, 4(y2 — y1)} . . 2{—1, 0} . . 3{0, —1} . . 4{1, 0} . .
     5{x5 — x4, 4(y5 — y4)}.                                                % bowl

"Script D"; spen;
call charbegin(`D, 12, mi·corr, —.6mi·cor, ph, 0, mi[.6ph·slant — pu, 0]);
x1 = 3u;    y1 = .95h;
x2 = 3.5u;  y2 = .48h;
x3 = 3u;    y3 = 0;
x4 = 1.75u; y4 = .9h;
x5 = 5.5u;  y5 = h;
x6 = 10u;   y6 = .55h;
draw 1{2(x2 — x1), y2 — y1} . . 2{0, —1} . . 3{2(x3 — x2), y3 — y2};          % stem
draw 4{x5 — x4, 3(y5 — y4)} . . 5{1, 0} . . 6{0, —1} . . 3{—1, 0}.            % bowl
```

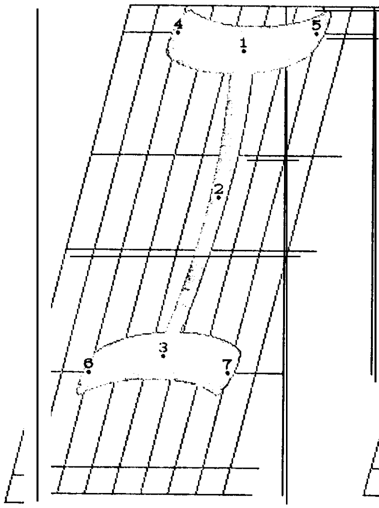


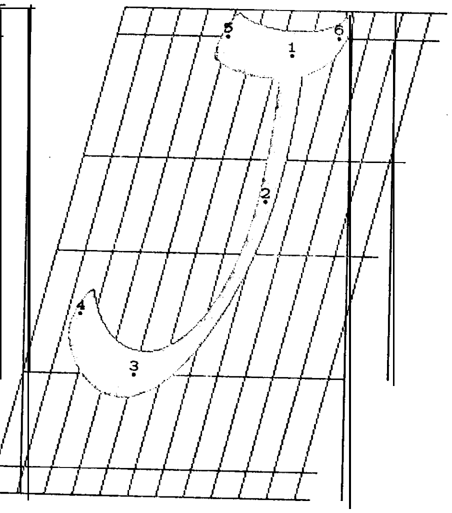




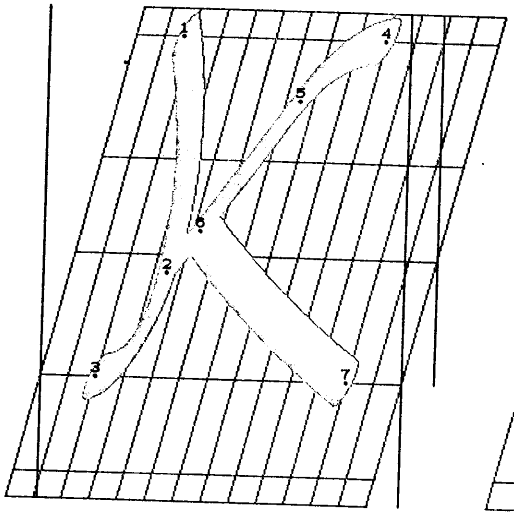
```
"Script E"; spen;
call charbegin(`E, 10.5, .2mi·cor, —.2mi·cor, ph, 0, mi[ph·slant — 1.5pu, .8ph·slant — .5pu]);
x1 = 8u;   y1 = .84h;
x2 = 5.5u;   y2 = h;
x3 = 2.75u;   y3 = .84h;
x4 = 6u;   y4 = .56h;
x5 = 2u;   y5 = .3h;
x6 = 5.5u;   y6 = 0;
x7 = 8.5u;   y7 = .12h;
draw 1{x2 — x1, 4(y2 — y1)} . . 2{—1, 0} . . 3{0, -1 ). . 4{1, 0};                % upper bowl
draw 4{—1, 0} . . 5{0, —1} . . 6{1, 0} . . 7{x7 — x6, 3(y7 — y6)}.                 % lower bowl

"Script F"; spen;
call charbegin(`F, 11, mi·corr, mi(1 — .5cor), ph, 0, mi[ph·slant — pu, .5ph·slant + pu]);
x1 = 1.75u;   y1 = .86h;
x2 = 9u;   y2 = h;
x3 = 6.5u;   y3 = .59h;
x4 = 6.5u;   y4 = .27h;
x5 = 4.25u;   y5 = 0;
x6 = 3u;   y6 = .07h;
x7 = 5u;   y7 = .52h;
x8 = 8u;   y8 = .52h;
draw 1{x2 — x1, 5(y2 — y1)} . . 2{1, 0};                                            % shoulder
draw 2{—1, 0} . . 3{0, -1). . 4{0, -1). . 5{—1, 0} . . 6{x6 — x5, 3(y6 — y5)};      % stem
draw 7 . . 8.                                                                       % bar

"Script G"; spen;
call charbegin(`G, 11.5, 0, mi(-. 75cor + .5corr), ph, 0, mi[.75ph·slant — 1.5pu, 0]);
x1 = 7.5u;   y1 = .86h;
x2 = 5.5u;   y2 = h;
x3 = 2.5u;   y3 = .66h;
x4 = 5.5u;   y4 = .23h;
x5 = 9u;   y5 = .63h;
x6 = 9.5u;   y6 = .23h;
x7 = 5.5u;   y7 = 0;
x8 = 2u;   y8 = .15h;
draw 1{x2 — x1, 4(y2 — y,)} . . 2{—1, 0). . 3{0, -1). : 4{1, 0} . . 5{0, 1};        % upper bowl
draw 5{2(x6 — x5), y6 — y5} . . 6{0, —1} . . 7{—1, 0} . . 8{x8 — x7, 3(y8 — y7)}.   % tail
```

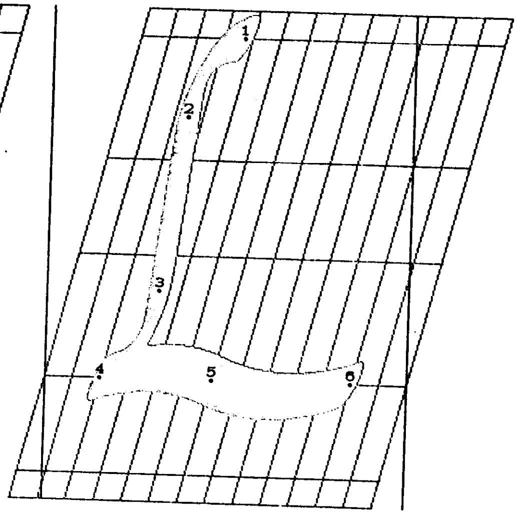


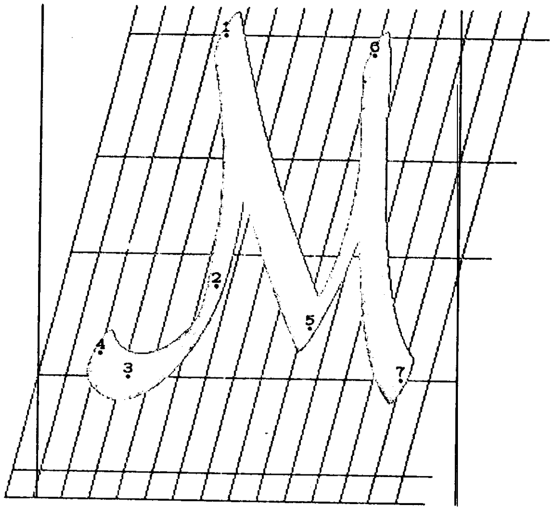




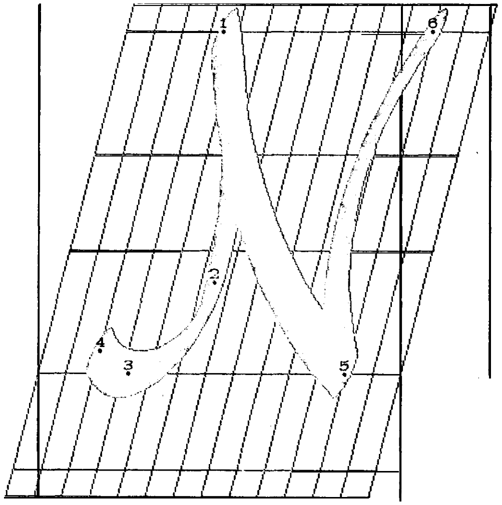
```
'Script H"; spen;
call charbegin(`H, 12, mi·corr, 0, ph, 0, ph·slant — (2 — mi)pu);
x1 = 1.5u;   y1 = h;
x2 = 5u;   y2 = .4h;
x3 = 3u;   y3 = 0;
x4 = 9u;   y4 = h;
x5 = 8.5u;   y5 = .4h;
x6 = 10u;   y6 = 0;
x7 = 3u;   y7 = .45h;
x8 = 9u;   y8 = .45h;
draw 1{5(x2 — x1), y2 — y1} .. 2{0, —1} .. 3{—1, 0};                    % left stem
draw 4{2(x5 — x4), y5 — y4} .. 5{0, —1} .. 6{3(x6 — x5), y6 — y5};     % right stem
draw 7{7u — x7, .15h — y7} .. 8{x8 — 7u, y8 — .15h}.                    % bar

"Script I"; spen;
call charbegin( `I, 9, 0, 0, ph, 0, ph·slant — (1 — mi)pu);
x1 = 4.5u;   y1 = .95h;
x2 = 5u;   y2 = .52h;
x3 = 4.5u;   y3 = .05h;
x4 = 2u;   y4 = h;
x5 = 7u;   y5 = h;
x6 = 2u;   y6 = 0;
x7 = 7u;   y7 = 0;,
draw 1{2(x2 — x1), y2 — y1} . 2{0, —1} .. 3{2(x3 — x2), y3 — y2};      % stem
draw 4{x1 — x4, 3(y1 — y4)} . . 1{ 1,0} . 5{x5 — x1, 3(y5 — y1)};     % upper serif
draw 6{x3 — x6, 3(y3 — y6)} . . 3{1, 0} .. 7{x7 — x3, 3(y7 — y3)}.     % lower serif

"Script J"; spen;
call charbegin(`J, 10, 0, —.5mi·cor, ph, 0, mi[ph·slant — pu, .5ph·slant]);
x1 = 6.5u;   y1 = .95h;
x2 = 7u;   y2 = .52h;
x3 = 4u;   y3 = 0;
x4 = 1.521;   y4 = .18h;
x5 = 4u;   y5 = h;
x6 = 8u;   y6 = h;
draw 1{2(x2 — x1), y2 — y1} . 2{0, —1} . .
      3{— 1, 0} .. 4{x4 — x3, 3(y4 — y3)};                               % slem and tail
draw 5{x1 — x5, 3(y1 — yj)} 1{1, 0} 6{x6 — x1, 3(y6 — y1)}.              % serif
```

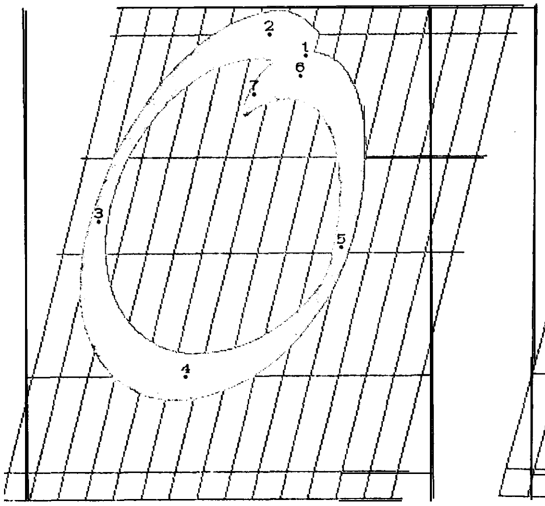


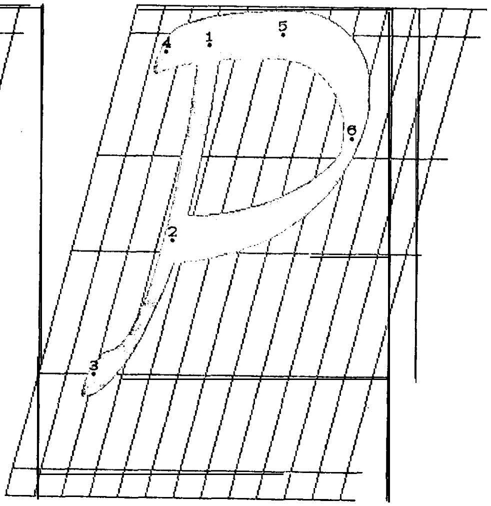




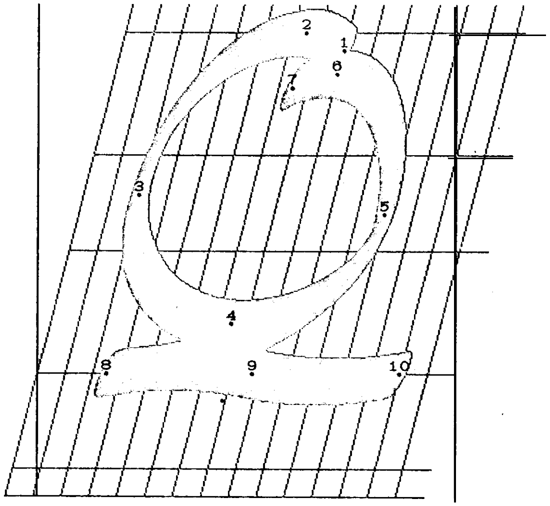
```
"Script K"; spen;
call charbegin(`K,13,0,0, ph, 0, ph·slant — (3 — mi)pu);
x1 = 1.75u;   y1 = h;
x2 = 3.5u;   y2 = .31h;
x3 = 2u;   y3 = 0;
x4 = 9u;   y4 = h;
x5 = 6.5u;   y5 = .82h;
x6  =  4.25u; new aa;   x6 = aa[x2, x5];   y6 = aa[y2, y5];
x7 = 11u;   y7 = 0;
draw 1{2(x2 — x1), y2 — y1}..2{0, —1}..3{—1, 0};                          % stem
draw 4{—1, 0}..5{x2 — x5, y2 — y5}..2;                          % upper diagonal
draw 6{x7 — x6, 1.25(y7 — y6)}..7{1.25(x7 — x6). y7 -- y6}.            % lower diagonal

"Script L"; spen;
call charbegin(`L,13,0,0, ph, 0, 0);
x1 = 3.75u;   y1 = h;
x2 = 2.5u;   y2 = .77h;
x3 = 3.25u;   y3 = .26h;
x4 = 2u;   y4 = 0;
x5 = 6u;   y5 = 0;
x6 = 11u;   y6 = 0;
draw 1{—1, 0}..2{x3 — x2, y3 — y2}..3{x3 — x2, y3 — y2}..4{—60u, —h};     % stem
draw 4{30u, h}..5{30u, -h}..6{30u, h}.                                      % bar

"Script M"; spen;
call charbegin(`M,15,0,0, ph, 0, 0);
x1 = 3.5u;   y1 = h;
x2 = 5.5u;   y2 = .27h;
x3 = 3.25u;   y3 = 0;
x4 = 2u;   y4 = .07h;
x5 = 9.25u;   y5 = .15h;
x6 = 9u;   y6 = .95h;
x7 = 13u;   y7 = 0;
draw 1{2(x2 — x1), y2 — y1}..2{0, -1}. .
     3{—1, 0}..4{x4 — x3, 3(y4 — y3)};                             % first diagonal
draw 1{x5 — x1, 1.25(y5 — y1)}..5{I .25(x5 — x1), y5 — y1};         % second diagonal
draw 5{4u, h}..6{—4u, h};                                            % third diagonal
draw 6{x7 — x6, 1.5(y7 — y6)}..7{3(x7 -- x6), y7 — y6}.             % fourth diagonal
```

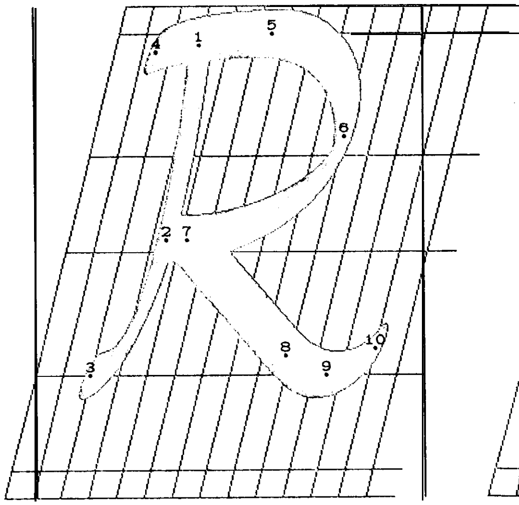


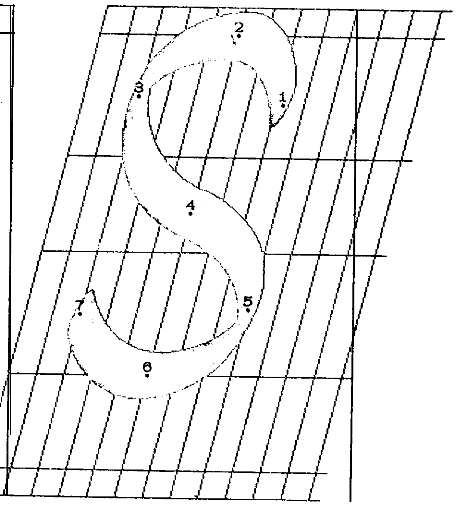




```
"Script N"; spen;
call charbegin( `N, 13, 0, 0, ph, 0, ph·slant— (1 —mi)pu);
x1 = 3.5u;   y1 = h;
x2 = 5.5,;   y2 = .27h;
x3 = 3.25u;   y3 = 0;
x4 = 221;   y1 = .07h;
x5 = 11u;   y5 = 0;
x6 = llu;   y6 = h;
draw 1{2(x2 -- x1), y2 — y1} . 2{0, —1} . . 3{—1, 0} . . 4{x4 — x3, 3(y4 — y3)};     % left stem
draw 1{x5 — x1, 2(y5 -- y1)} . 5{2(x5 — x1), y5 — y1};                               % diagonal
draw 5{—6u, h} .. 6{6u, h}.                                                          % right stem

"Script O";   spen;
call charbegin(`O, 13, .25mi·cor, —.75mi·cor, ph, 0, mi[.75ph·slant — pu, 0]);
x1 = 8u;   y1 = .94h;
x2 = 6.5u;   y2 = h;
x3 = 2u;   y3 = .46h;
x4 = 6.5u;   y4 = 0;
x5 = llu;   y5 = .38h;
x6 = 8u;   y6 = .88h;
x7 = 6 5u;   y7 = .83h;
draw 1{x2 -- x1, 2(y2 — y1)} . . 2{—1, 0} .. 3{0, —1} . . 4{1, 0} .. 5{0, 1} . .
     6{—1, 0} . . 7{x7 — x6, 4(y7 — y6)}.                                            % bowl

"Script P"; spen;
call charbegin(`P, 11, 0, —.5mi·cor, ph, 0, mi[.8ph·slant — pu, .3ph·slant]);
x1 = 3u;   y1 = .97h;
x2 = 3.5u;   y2 = .4h;
x3 = 2u;   y3 = 0;
x4 = 1.5u;   y4 = .95h;
x5 = 5.5u;   y5 = h ;
x6 = 9u;   y6 = .7h;
draw 1{2(x2 — x1), y2 — y1} . . 2{0, -l}. . 3{—1, 0};                                % stem
draw 4{x5 — x4, 4(y5 — y4)} . . 5{1, 0} . . 6{0, -1). . 2{—1, 0}.                    % bowl
```



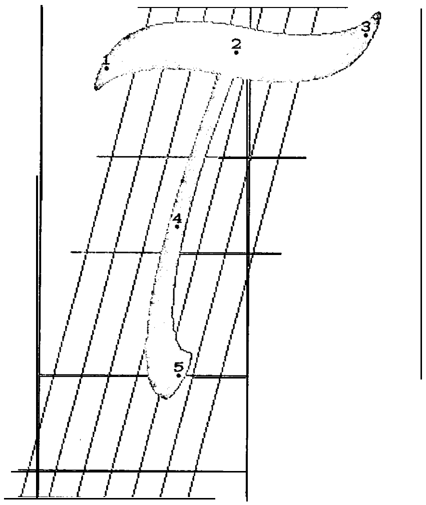


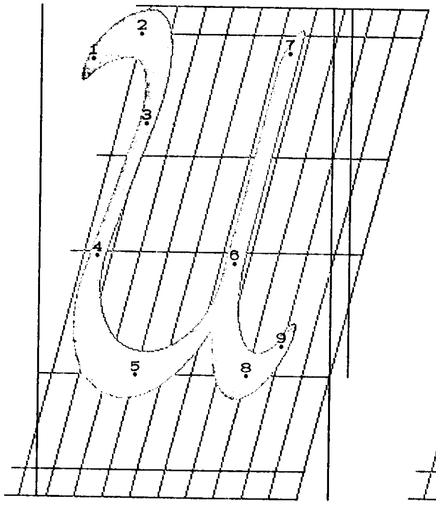


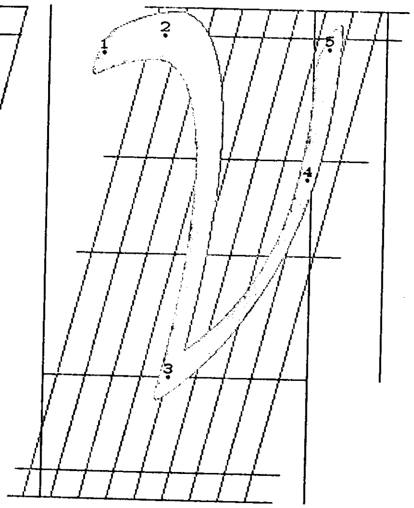
"Script Q"; spen;
call *charbegin*(`Q, 15, 0, 0, *ph*, 0, 0);
$x_1 = 8u$; $y_1 = .95h$;
$x_2 = 6.5u$; $y_2 = h$;
$x_3 = 2u$; $y_3 = .53h$;
$x_4 = 6.5u$; $y_4 = .15h$;
$x_5 = 11u$; $y_5 = .47h$;
$x_6 = 8u$; $y_6 = .88h$;
$x_7 = 6.5u$; $y_7 = .84h$;
$x_8 = 2.5u$; $y_8 = 0$;
$x_9 = 7.75u$; $y_9 = 0$;
$x_{10} = 13u$; $y_{10} = 0$;
draw $1\{x_2 -$ XI, $2(y_2 - y_1)\}.2\{-1,0\}..3\{0,-1\}..4\{1,0\}..5\{0,1\}..$
$6\{-1,0\}..7\{x_7 - x_6, 4(y_7 - y_6)\}$; % howl
draw $5\{0,-1\}..8\{-120u,-h\}$; % link
draw $8\{60u, h\}.9\{60u, -h\} 10\{60u, h\}$. % bar

"Script It"; spen;
call *charbegin*(`R, 14, 0, 0, ph, 0, 0);
$x_1 = 3u$; $y_1 = .97h$;
$x_2 = 3.5u$; $y_2 = .4h$;
$x_3 = 2u$; $y_3 = 0$;
$x_4 = 1.5,$; $y_4 = .95h$;
$x_5 = 5.5u$; $y_5 = h$;
$x_6 = 9u$; $y_6 = .7h$;
$x_7 = 4.25u$; $y_7 = .4h$;
$y_8 = .06h$; % x_8 will be defined shortly
$x_9 = 10.5u$; $y_9 = 0$;
$(x_9 -- x_8)/(y_9 - y_8) = 2(x_8 - x_7)/(y_8 - y_7)$;
$x_{10} = 12u$; $y_{10} = .08h$;
draw $1\{2(x_2 -- x_1), y_2 - y_1\}.2\{0,-1\}..3\{-1,0\}$; % stem
draw $4\{x_5 - x_4, 4(y_5 - y_4)\}..5\{1,0\}..6\{0,-1\}..2\{--1, O\}$; % bowl
draw 7. $.8\{x_8 -- x_7, y_8 - y_7\}.9\{1, 0\} . 10\{x_{10} - x_9, 3(y_{10} - y_9)\}$. % diagonal

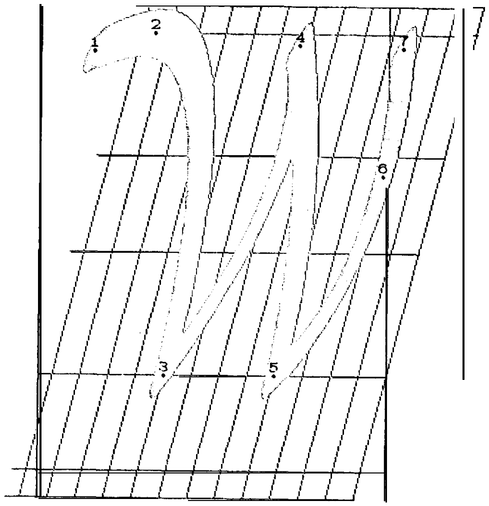
"Script S"; spen;
call *charbegin*(`S, 10, 0, $-mi(cor - .75corr)$, ph, 0, $mi[ph{\cdot}slant - 1.75pu, 0]$);
$x_1 = 7.25u$; $y_1 = .8h$;
$x_2 = 5u$; $y_2 = h$;
$x_3 = 2u$; $y_3 = .82h$;
$x_4 = 5u$; $y_4 = .48h$;
$x_5 = 8u$; $y_5 = .2h$;
$x_6 = 5u$; $y_6 = 0$;
$x_7 = 2u$; $y_7 = .18h$;
draw $1\{x_2 - x_1, 4(y_2 - y_1)\}..2\{-1,0\}..3\{0,-1\}..4\{32u,--h\}..5\{0,-1\}..$
$6\{-1,0\}..7\{x_7 - x_6, 3(y_7 - y_6)\}$. % stroke

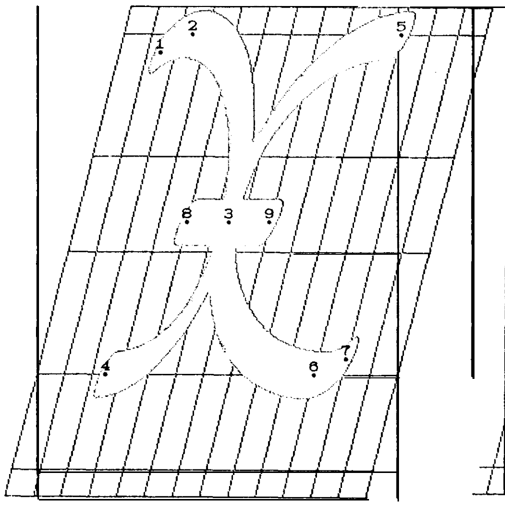


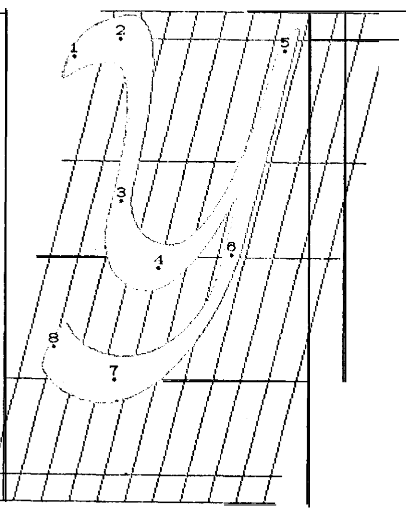




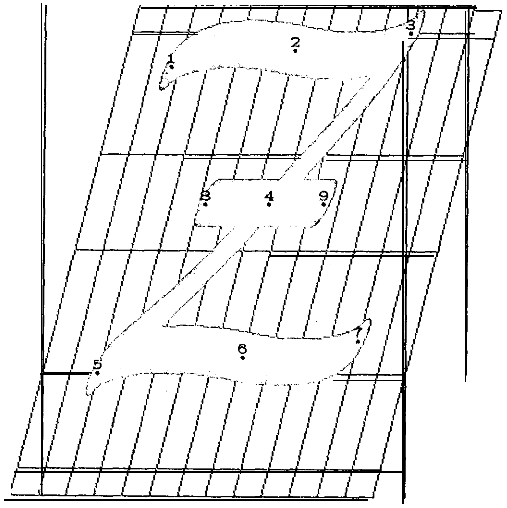
```
"Script T";   spen;
call charbegin(`T, 13, .8mi·cor,3mi,ph, 0, ph·slant + (4mi — l)pu);
x1 = 2u;   y1 = .9h;
x2 = 6.5u;   y2 = .95h;
x3 = llu;   y3 = h;
x4 = 6u;   y4 = .44h;
x5 = 7.5u;   Y5 = 0;
draw 1{6u,h} . . 2{60u, -h}. . 3{6u, h};                             % bar
draw 2{2(x4 — x2), y4 — y2} . . 4{0, -1). . 5{1, O}.                 % stem

"Script U";  spen;
call charbegin(`U, 13, .8mi·cor, 0, ph, 0, ph·slant — (3.5 — mi)pu);
x1 = 1.5u;   y1 = .93h;
x2 == 3u;   y2 = h;
x3 = 4u;   y3 == .74h;
x4 = 3.5u;   y4 = .35h;
x5 = 6u;   y5 = 0;
x6 = 8.5u;   y6 = .33h;
x7 = 8.5u;   y7 = .95h;
x8 = 10u;   y8 = 0;
x9 = llu;   y9 = .09h;
draw l{x2 — x1, 3(y2 — y1)} . . 2{1, 0} . 3{x4 — x3, y4 — y3} . . 4{x4 — x3, y4 — y3} . .
    5{1, 0} . . 6{0, l} . . 7;                                       % stroke and stem
draw 6{0, -1). . 8{1, 0} . . 9{x9 — x8, 4(y9 — y8)}.                 % point

"Script V"; spen;
call charbegin(`V, 12, .8mi·cor, 0, ph, 0, ph·slant — (1.5 — mi)pu);
x1 = l.5u;   y1 = .95h;
x2 = 3.521;  y2 = h;
x3 = 7u;   y3 = 0;
x4 = 10u;   y4 = .59h;
x5 = 9.5u;   y5 = .97h:
draw 1{x2 — x1, 2(y2 — y1)} . . 2{1, 0} . . 3{0, —1};                % left diagonal
draw 3{3(x4 — x3), y4 — y3} . . 4{0, l} . . 5{2(x5 — x4), y5 — y4}.  % right diagonal
```







```
"Script W";    spen;
call charbegin(`W, 15, .8mi·cor, 0, ph, 0, ph·slant -- (1.5 — mi)pu);
x1 = 1.5u;   y1 = .95h;
x2 = 3.5u;   y2 = h;
x3 = 7u;   y3 = 0;
x4 = 8.75u;   y4 = .97h;
x5 = llu;   y5 = 0;
x6 = 13u;   y6 = .59h;
x7 = 12.5u;   y7 == .96h;
draw 1{x2 — x1, 2(y2 — y1)} . 2{1, O}. 3{0, —1};                     % first diagonal
draw 3{x7 — x3, y7 — y3} . 4{2(x4 — x5), y4 -- y5};                 % second diagonal
draw 4{2(x5 — x4), y5 — y4} 5{0, —1};                                % third diagonal
draw 5{3(x6 — x5), y6 — y5} . . 6{0, 1} . 7{2(x7 — x6), y7 -- y6}.  % fourth diagonal

"Script X"; spen;
call charbegin(`X, 13, 0, 0, ph, 0, ph·slant — (2 — 1.5mi)pu);
x1 == 1.5u;   y1 = .95h;
x2 == 2.5u;   y2 = h;
xj == 5.5u;   y3 = .45h;
x4 == 2.5u;   y4 == 0;
x5 == 10u;   y5 == h;
x6 = 10u;   y6 = 0;
x7 = 11u;   y7 = .05h;
x8 == 4u;   y8 == .45h;
x9 = 7u;   y9 = .45h;
draw 1{x2 — x1, 3(y2 — y1)} . . 2{1, O}. . 3{0, —1} . . 4{—1, 0};          % left stroke
draw 5{—1, 0} . . 3{0, —1} . . 6{1, 0} . . 7{x7 — x6, 3(y7 — y6)};         % right stroke
draw 8 . . 9.                                                              % bar

"Script Y"; spen;
call charbegin(`Y, 11.5, .8mi·cor, —.6mi·cor, ph, 0, mi[ph·slant — pu, .4ph·slant]);
x1 = 2u;   y1 = .95h;
x2 = 3.5u;   y2 = h;
x3 = 5u;   y3 = .53h;
x4 = 721;   y4 = .33h;
x5 = 9.5u;   y5 = .97h;
x6 = 9.5u;   y6 = .37h;
x7 = 6.5u;   y7 = 0;
x8 = 4u;   y8 = .1h;
draw 1{x2 — x1, 3(y2 — y1)} , . 2{1, 0} . . 3{0, —1} . 4{1, 0} . . 5{0, 1};   % stroke
draw 5.. 6{0, —1} . . 7{—1, 0} . . 8{x8 -- x7, 3(y8 --- y7)}.                  % stem and tail
```



```
"Script Z";" spen;
call charbegin(`Z, 13, 0, 0, ph, 0, ph·slant — (2 — mi)pu);
x1 = 1.75u; y1 = .9h;
x2 = 6u;   y2 = .95h;
x3 = 10u;   y3 = h;
x4 = 6.5u;   y4 = .5h;
x5 = 2u;    y5 = 0;
x6 = 7u;   y6 = .05h;
x7 = 11u;   y7 = .1h;
x8 = 4.25,;    y8 = .5h;
x9 = 8.5u;   y9 = .5h;
draw 1{6u, h} . 2{60u, -h} . 3{6u,h};                                        % upper bar
draw 3{x4 — x3, 2(y4 — y3)} 4{x5 — x3, y5 — y3} . . 5{x5 — x4, 2(y5 — y4)};   % diagonal
draw 5{6u, h} . 6{60u, —h} . . 7{6u,h};                                      % lower bar
draw 8. . 9.                                                                 % middle bar
```

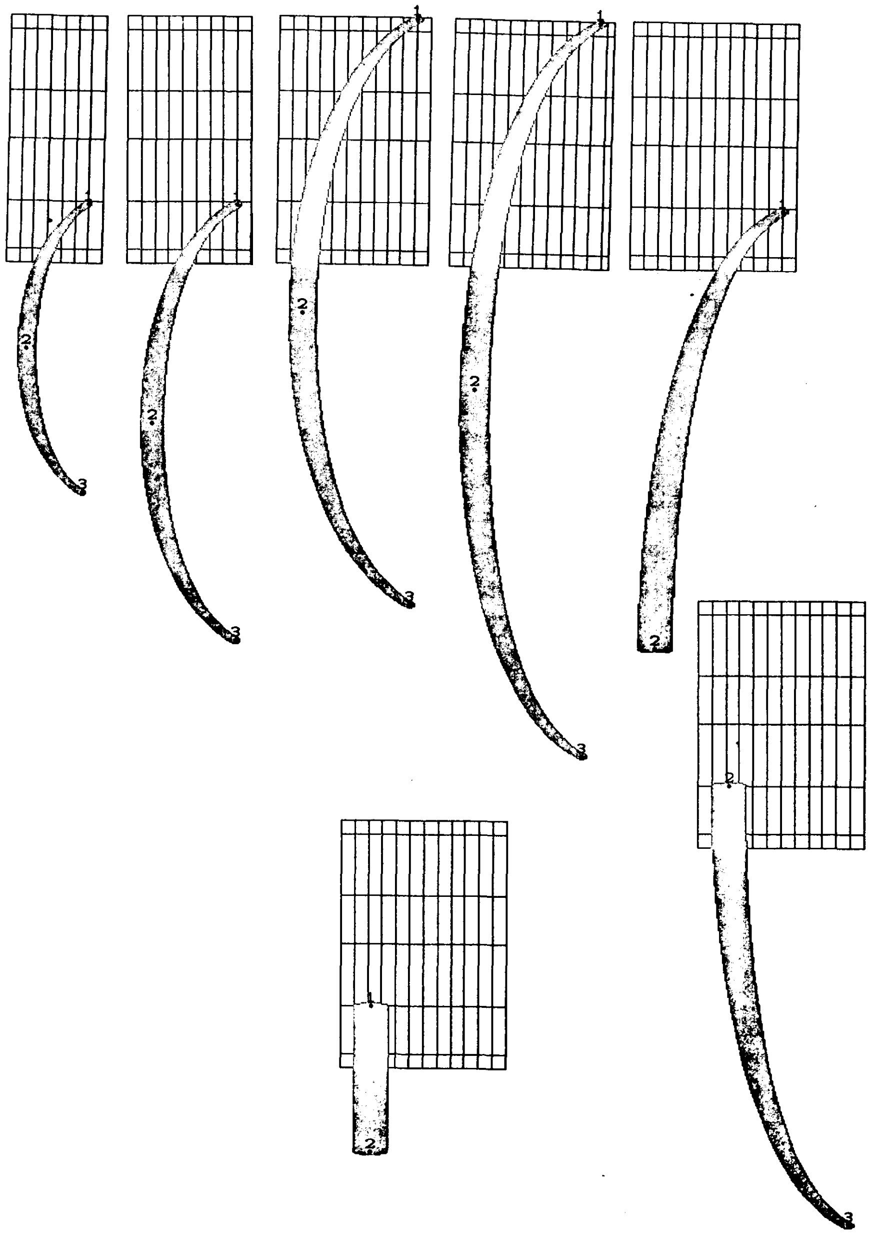
MATHEX CHARACTER DESIGNS

The file mathex.mf

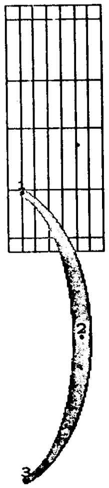
```
% The Computer Modern Math-Extension family of fonts (by D. E. Knuth, 1979).
danger = 0;
new pt;                                             % one virtual point
pt = typesize/10;                         % assumes 10 point specifications
% The following subroutines break up the large characters on an Alphatype CRS,
% assuming that 10pt equals 10 points.
subroutine eighteen :
if mode = 2:   crsbreak —9pt·cf;
fi.

subroutine twentyfour:
if mode = 2:   crsbreak (ph + pb —12pt)·cf;
fi.

subroutine thirty:
if mode = 2:   crsbreak (ph + pb —12pt)·cf;   crsbreak (ph + pb —18pt)·cf;
fi.

input mat hd 1;                                    % the large delimiters
input mathop;                                      % the large operators
input mexext;                  % nonstandard characters (any of the codes
          % ‘017, ‘037, ‘0555’057, ‘077, ´104–´105,´140–´157, ‘167, ‘176, ‘177)
texinfo slant, 6pu,3pu,2pu, px, 18pu,2pu, prt;
% (The calling tile should supply the remaining tcxinfo.)
```

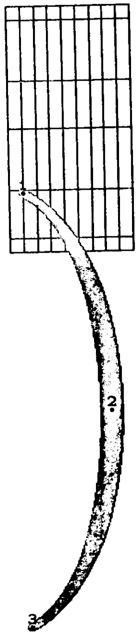
The file mathd 1.mf

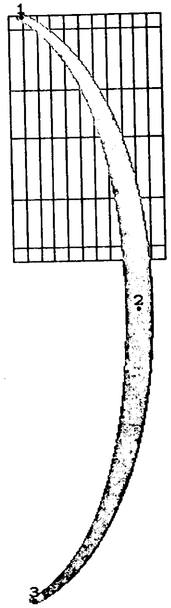
```
% Left parentheses (left-right symmetric with right ones)
charlist ‘000, ‘020, ‘022, ‘040, ‘060, 0;
subroutine biglp(var code, var units, var minps, var maxps, var height, var depth):
call charbegin(code, units, 0, 0, height, depth, 0);
new w48,w49;   w48 = round minps;   w19 = round maxps;
hpen;   x1 = x3 = good48(r — u);
top48y1 = round pixels·height;   bot48y3 = 1 — round pixels·depth;   y2 = .5[y1, y3];
lft49x2 = round u;
y0 = y1;   y4 = y3;   x0 = x4 = x1 + 1.875(units — 2)u;
draw (0..)|w48|1..|w49#|2{0,—1}..|w48|3(..4).                        % stroke
```

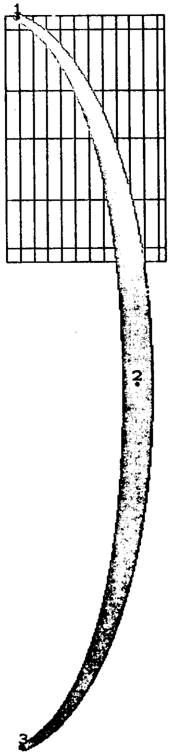


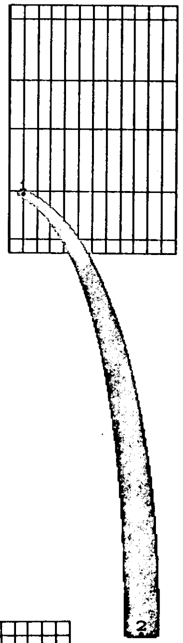
```
"12 point left parenthesis";
call biglp('000, 7, w0, w1, 0, 12pt).

"18 point left parenthesis";
call biglp('020, 9, w10, bold, 0, 18pt);   call eighteen.

"24 point left parenthesis";
call biglp( '022, 11, w10 + .2deltaw, bold + deltaw, ph + pb, 24pt — ph — pb);
call twentyfour.

"30 point left parenthesis";
call biglp('040, 11.5, w10 + .4deltaw, bold + 2deltaw, ph + pb, 30pt -- ph — pb);
call thirty.

subroutine biglp: .                                       % free up METAFONT's memory

new w98, w99;   w98 = round(w10 + .6deltaw);   w99 = round(bold + 3deltaw);
"Extensible left parenthesis- top";
call charbegin( '060, 12, 0, 0, 0, 18pt, 0);
varchar '060, 0, ' 100, '102;                             %extensible left parenthesis
hpen;   x1 = good98(r — u);   top98y1 = 0;
lft99x2 = round u;   y2 = round(.5 — 18pt·pixels);
x0 = x1 + 1.875(10u);   y0 = y1;
draw (0. . )|w98|1|w99#|2{0, -1).                          % upper part of stroke

"Extensible left parenthesis- bottom";
call charbegin( '100, 12, 0, 0, 0, 18pt, 0);
hpen;   x3 = good98(r — u);   bot98y3 = round(.5 -- 18pt·pixels);
lft99x2 = round u;   y2 = 0;
x4 = x3 + 1.875(10u);   y4 = y3;
draw |w99#|2{0, —1} . |w98|3( . . 4).                       %lower part of stroke

"Extensible left parenthesis extension module";
call charbegin( '102, 12, 0, 0, 0, 6pt, 0);
varchar 0, 0, 0, '102;                          % left parenthesis extension modules only
hpen ; lft99x1 = round u; y1 = 0 ;
x2 = x1;   y2 = round(.5 — 6pt·pixels);
w99 draw 1 . . 2.                                             % middle part of stroke
```

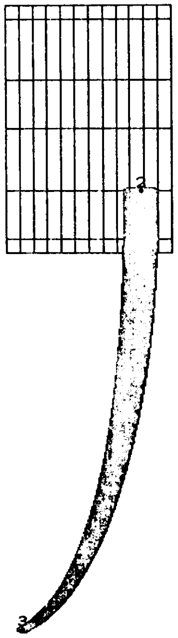




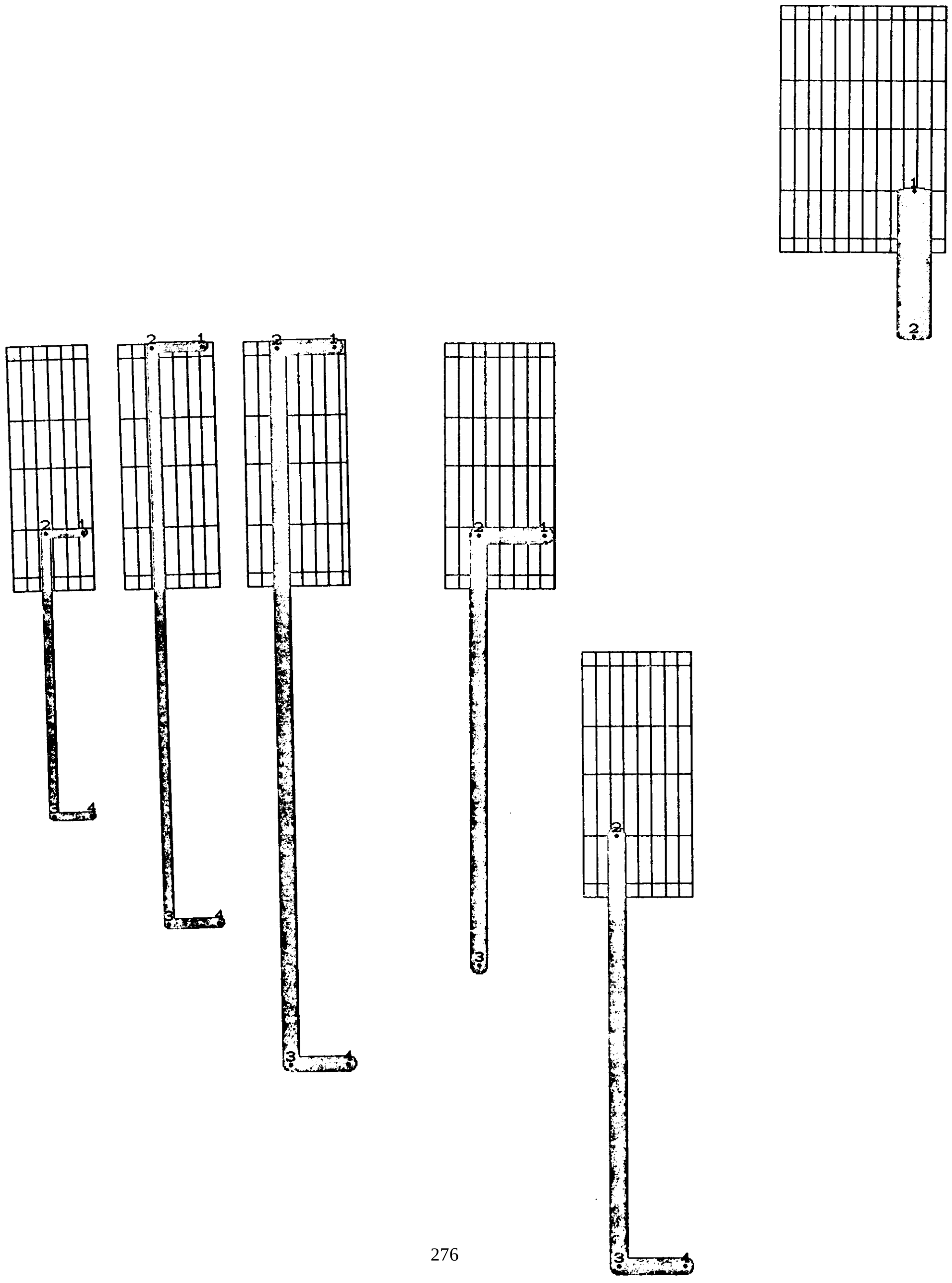




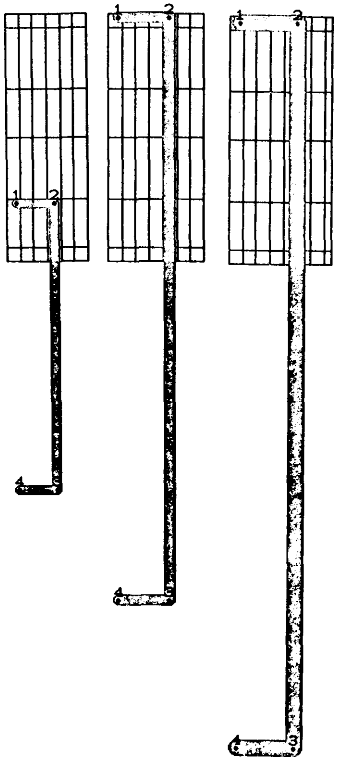


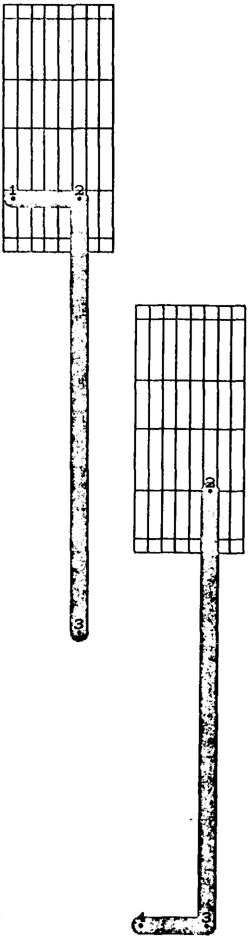


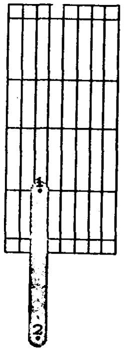
```
Ⅱ Right parentheses (Icft-right symmetric with left ones)
charlist ‘001, ‘021, ‘023, ‘041, ‘061, 0;
subroutine bigrp(var code, var units, var minps, var maxps, var height, var depth):
call charbegin(code, units, 0, 0, height, depth, 0);
new w48, w49;   w18 = round minps;   w49 = round maxps;
hpen;   x1 = x3;   r — x1 = good 18(r — u);
top18y1 = round pixels·height;   bot 18y3 = l — round pixels·depth;   y2 = .5[y1, y3];
lft49(r — x2) = roundu;
y0 = y1;   y4 = y3;   x0 = x4 = x1 — 1.875(units — 2)u;
draw (0.. )|w18| 1. . |w49#|2{0, -1) . |w18|3( .. 4).                                   % stroke

“12 point right parenthesis”;
call bigrp( ‘00 1, 7, w0, w1, 0, 12pt).

“18 point right parenthesis”;
call bigrp(´021, 9, w10, bold, 0, 18pt);   call eighteen.

“24 point right parenthesis”;
call bigrp(´023, 11, w10 + .2deltaw, bold + deltaw, ph + pb, 24pt — ph — pb);
call twentyfour.

“30 point right parenthesis”;
call bigrp( ‘041, 11.5, w10 + .4deltaw, bold + 2deltaw, ph + pb, 30pt — ph — pb);
call thirty.

subroutine bigrp: .                                   % free up METAFONT’s memory

new w98, w99;   w98 = round(w10 + .6deltaw);   w99 = round(bold + 3deltaw);
“Extensible right parenthesis-top”;
call charbegin( ‘06 1, 12, 0, 0, 0, 18pt, 0);
varchar ‘061, 0, ‘10 1, ‘103;                          % extensible right pnrcnthesis
hpen;   r — x1 = good98(r -- u);   top98y1 = 0;
lft99(r — x2) =.round u;   y2 = round(.5 -- 18pt·pixels);
x0 = x1 — 1.875(10u);   y0 = y1;
draw (0 . . )|w98|1 . . |w99#|2{0, -1).                        % upper part of stroke

“Extensible right parenthesis-bottom”;
call charbegin(´101, 12, 0, 0, 0, 18pt, 0);
hpen;   r — x3 = good98(r -- u);   bot98y3 = round(.5 — 18pt·pixels);
lft99(r — x2) = round u;   y2 = 0;
x1 = x3 -- 1.875(10u);   y4 = y3;
draw |w99#|2{0, —1} . . |w98|3( .. 4).                         % lower part of stroke
```



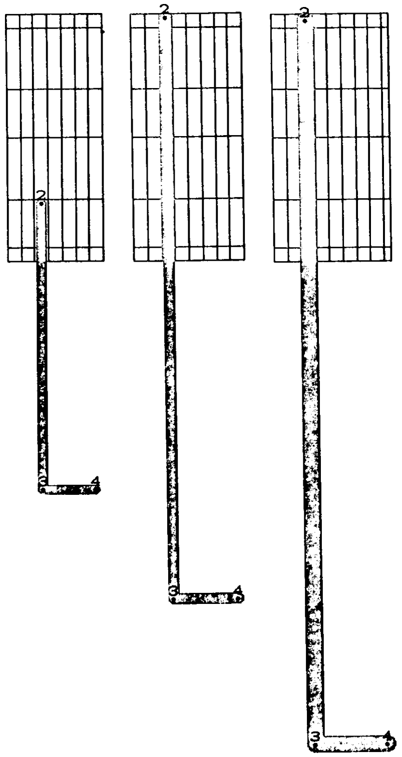
```
"Extensible right parenthesis extensi n module";
call charbegin('103, 12, 0, 0, 0, 6pt, 0);
varchar 0, 0, 0, '103;                          % right parenthesis extension modules only
hpen;   lft99(r -- x1) = round u;   y1 = 0;
x2 = x1;   y2 = round(.5 --- 6pt·pixels);
w99 draw 1..2.                                   % middle part of stroke

% Left brackets (left-right symmetric with right ones)
charlist '002, '024, '042, '062, 0;
subroutine biglb(var code, var units, var psize, var height, var depth):
call charbegin(code, units, 0, 0, height, depth, 0);
new w19;   w19 = round psize;
cpen;   x1 = x1 = good19(r — .75u);   x2 = x3 = good19(2.5u);
top19y1 = round pixels height;   hot19y3 = 1 — round pixels·depth;   y1 = y2;   y3 = y4;
w19 draw 1..2..2..3..3..4.                                              % stroke

"12 point left bracket";
call biglb('002, 6, w10, 0, 12pt).

"24 point left bracket";
call biglb('024, 7, w10 + 2deltaw, ph + pb, 24pt — ph — pb);
call twentyfour.

"30 point left bracket";
call biglb('042, 7.5, w1 -- .8deltaw, ph + pb, 30pt — ph — pb);
call thirty.

subroutine biglb:                                 % free up METAFONT's memory

new w99;   w99 = round(w1 - - .6deltaw);
"Extensible left bracket top";
call charbegin('062, 8, 0, 0, 0, 18pt, 0);
varchar '062, 0, '064, '066;                     % extensible left bracket
cpen;   x1 = good99(r — .75u);   top99y1 = 0;
x2 = good,, 2.5u;   y2 = y1;
x3 = x2;   y3 = round(.5 — 18pt·pixels);
w99 draw 1.. 2.. 2.. 3.                           % upper part of stroke

"Extensible left bracket-bottom";
call charbegin('064, 8, 0, 0, 0, 18pt, 0);
varchar 0, 0, '064, '066;                        % extensible left floor bracket
cpen;   x1 = good99(r — .75u);   bot99y1 = round(.5 — 18pt·pixels);
x3 = good99 2.5u;   y3 = y1;
x2 = x3;   y2 = 0;
w99 draw 2.. 3.. 3.. 4.                           % lower part of stroke
```

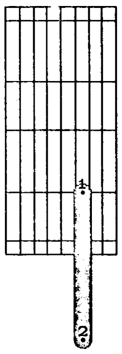


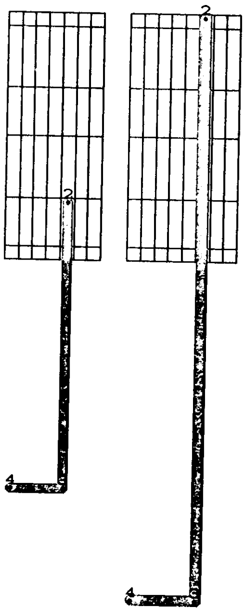




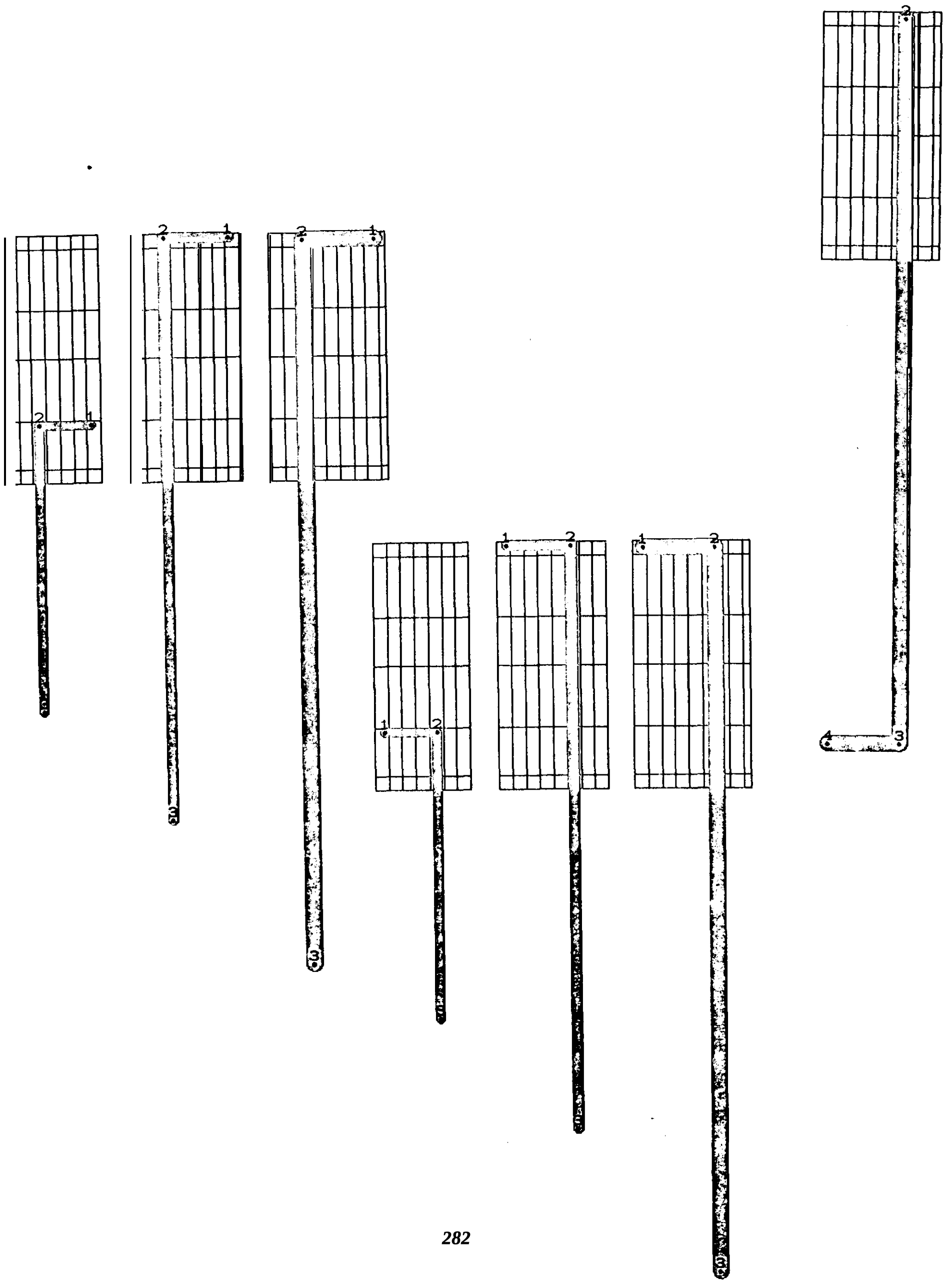
```
"Extensible left bracket-extension module";
call charbegin('066, 8, 0, 0, 0, 6pt, 0);
varchar '062, 0, 0, '066;                                   % extensible left ceiling bracket
cpen;  x1 = good99 2.5u;  y1 = 0;
x2 = x1;  y2 = round(.5 — 6pt·pixels);
w99 draw 1.. 2.                                             % middle part of stroke

% Right brackets (left-right symmetric with left ones)
charlist '003, '025, '043, '063, 0;
subroutine bigrb(var code, var units, var psize, var height, var depth):
call charbegin(code, units, 0, 0, height, depth, 0);
new w19;  w19 := round psize;
cpen;  x1 := x1;  r — x1 = good19(r — .75u);  x2 := x3;  r — x2 := good 19(2.5u);
top19 y1 := round pixels·height;  bot19 y3 := 1 -- round pixels·depth;  y1 = y2;  y3 = y4;
w19 draw 1 . . 2 . . 2 . . 3 . . 3 . . 4.                    % stroke

"12 point right bracket";
call bigrb('003, 6, w10, 0, 12pt).

"24 point right bracket";
call bigrb('025, 7, w10 + .2deltaw, ph + pb, 24pt — ph -- pb);
call twentyfour.

"30 point right bracket";
call bigrb('043, 7.5, w1 — .8deltaw, ph + pb, 30pt — ph — pb);
call thirty.

subroutine bigrb: .                                         % free up METAFONT's memory

new w99;  w99 := round(w1 — .6deltaw);
"Extensible right bracket-top";
call charbegin('063, 8, 0, 0, 0, 18pt, 0);
varchar '063, 0, '065, '067;                                % extensible right bracket
cpen;  r — x1 := good99(r — .75u);  top99 y1 = 0;
r — x2 = good99 2.5u;  y2 = y1;
x3 = x2;  y3 := round(.5 — 18pt·pixels);
w99 draw 1 . . 2 . . 2 . . 3.                               % upper part of stroke

"Extensible right bracket--bottom";
call charbegin('065, 8, 0, 0, 0, 18pt, 0);
varchar 0, 0, '065, '067;                                   % extensible right floor bracket
cpen;  r — x1 = good99(r — .75u);  bot99 y4 = round(.5 -- 18pt·pixels);
r — x3 = good99 2.5u;  y3 = y4;
x2 = x3;  y2 := 0;
w99 draw 2.. 3.. 3.. 4.                                     % lower part of stroke
```



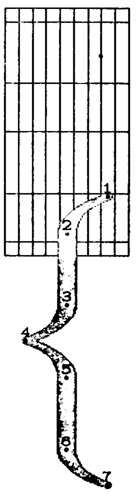


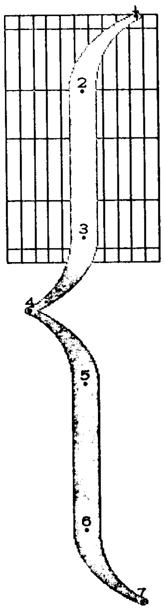


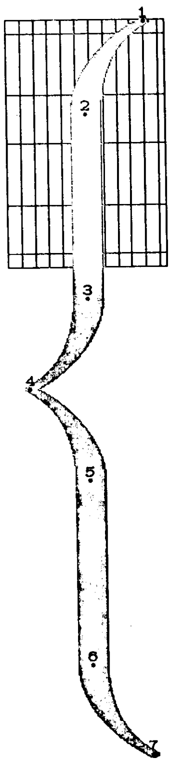
```
Extensible right bracket-extension module";
call charbegin('067, 8, 0, 0, 0, 6pt, 0);
varchar '063, 0, 0, '067;                                    % extensible right ceiling bracket
cpen;  r — x_1 = good_99 2.5u;  y_1 = 0;
x_2 = x_1;  y_2 = round(.5 — 6pt·pixels);
w_99 draw 1..2.                                              % middle part of stroke

% Left floor brackets (left-right symmetric with right ones)
charlist '004, '026, '044, '064, 0;
subroutine biglfb(var code, var units, var psize, var height, var depth):
call charbegin(code, units, 0, 0, height, depth, 0);
new w_19;  w_19 = round psize;
cpen;  x_4 = good_19(r — .75u);  x_2 = x_3 = good_19(2.5u);
top_19 y_2 = round pixels height;  bot_19 y_3 = 1 — round pixels·depth;  y_3 = y_4;
w_19 draw 2..3..3..4.                                        % stroke

"12 point left floor bracket";
call biglfb( '004, 7, w_10, 0, 12pt).

"24 point left floor bracket";
call biglfb('026, 8, w_10 + .2deltaw, ph + pb, 24pt -- ph — pb);
call twentyfour.

"30 point left floor bracket";
call biglfb('044, 8.5, w_1 — .8deltaw, ph + pb, 30pt — ph -- pb);
call thirty.

subroutine biglfb: .                                         % free up METAFONT's rnemory

% Right floor brackets (left-right symmetric with left ones)
charlist '005, '027, '045, '065, 0;
subroutine bigrfb(var code, var units, var psize, var height, var depth):
call charbegin(code, units, 0, 0, height, depth, 0);
new w_49;  w_19 = round psize;
cpen;  r — x_1 = good_49(r — .75u);  x_2 = x_3;  r — x_3 = good_19(2.5u);
top_49 y_2 = round pixels·height;  hot_19 y_3 = 1 — round pixels·depth;  y_3 = y_4;
w_19 draw 2.. 3.. 3.. 4.                                     % stroke

"12 point right floor bracket";
call bigrfb('005, 7, w_10, 0, 12pt).

"24 point right floor bracket";
call bigrfb('027, 8, w_10 + .2deltaw, ph + pb, 24pt — ph — pb);
call twentyfour.
```

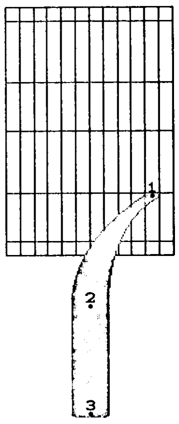


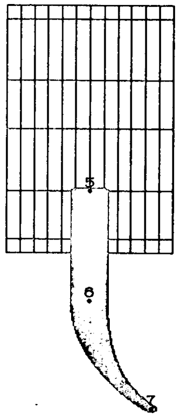
```
"30 point right floor bracket";
call $bigrfb('045, 8.5, w_1 - .8deltaw, ph + pb, 30pt - ph - pb)$;
call thirty.

subroutine bigrfb: .                                   % free up METAFONT's memory

% Left ceiling brackets (left-right symmetric with right ones)
charlist '006, '030, '046, '066, 0;
subroutine biglcb(var code, var units, var psize, var height, var depth):
call charbegin(code, units, 0, 0, height, depth, 0);
new $w_{19}$;   $w_{19}$ = round psize;
cpen;   $x_1 = good_{19}(r - .75u)$;   $x_2 = x_3 = good_{19}(2.5u)$;
$top_{19}y_1$ = round pixels·height;   $bot_{19}y_3 = 1$ − round pixels·depth;   $y_1 = y_2$;
$w_{19}$ draw 1 . . 2. . 2. . 3.                                                    % stroke

"12 point left ceiling bracket";
call $biglcb('006, 7, w_{10}, 0, 12pt)$.

"24 point left ceiling bracket";
call $biglcb('030, 8, w_{10} + .2deltaw, ph + pb, 24pt - ph - pb)$;
call twentyfour.

"30 point left ceiling bracket";
call $biglcb('046, 8.5, w_1 - .8deltaw, ph + pb, 30pt - ph - pb)$;
call thirty.

subroutine biglcb: .                                   % free up METAFONT's memory

% Right ceiling brackets (left-right symmetric with left ones)
charlist '007, '031, '047, '067, 0;
subroutine bigrcb(var code, var units, var psize, var height, var depth):
call charbegin(code, units, 0, 0, height, depth, 0);
new $w_{19}$;   $w_{19}$ = round psize;
cpen;   $r - x_1 = good_{19}(r - .75u)$;   $x_2 = x_3$;   $r - x_3 = good_{19}(2.5u)$;
$top_{19}y_1$ = round pixels·height;   $bot_{19}y_3 = 1$ − round pixels·depth;   $y_1 = y_2$;
$w_{19}$ draw 1..2..2..3.                                                      % stroke

"12 point right ceiling bracket";
call $bigrcb('007, 7, w_{10}, 0, 12pt)$.

"24 point right ceiling bracket";
call $bigrcb('031, 8, w_{10} + .2deltaw, ph + pb, 24pt - ph - pb)$;
call twentyfour.

"30 point right ceiling bracket";
call $bigrcb('047, 8.5, w_1 - .8deltaw, ph + pb, 30pt - ph - pb)$;
call thirty.
```



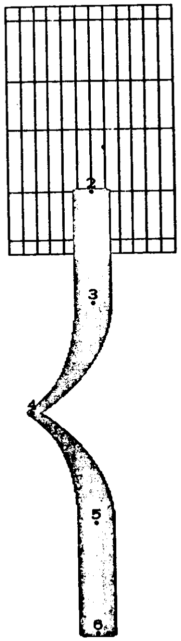


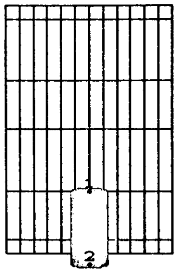


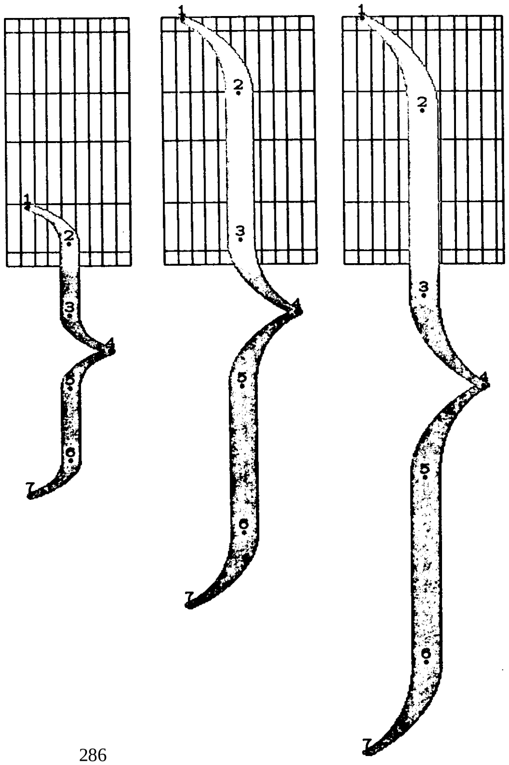




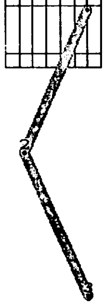
```
subroutine bigrcb: .                                            % free up METAFONT's memory

% Left braces (left-right symmetric with right ones)
charlist '010, '032, '050, '070, 0;
subroutine biglbr(var code, var units, vnr minps, var maxps, var height, var depth):
call charbegin(code,units, 0, 0, height, depth, 0);
new w_IX, w_19;   w_18 = round minps;   w_19 = round maxps;
hpen;   x2 = x3 = x5 = x6 = good_19(.5r);
x1 — x2 = x2 — x1 = .5(units — 3)u + eps;   x1 = x7;
top_19 y1 = round height·pixels;   bot_19 y7 = 1 — round pixels·depth;
y4 = .5[y3, y5] = good_6(.5[y1, y7]);
y1 -- y2 = y3 — y4 = y6 — y7 = (y1 -- y4)/4;
draw |w18#|1{3(x2 — x1), y2 -- y1} . |w19#|2{0, —1} |w19#|3{0, —1} . .
     |w18#|4{3(x4 — x3), y4 -- y3};                                             % upper stem
draw |w18#|7{3(x6 — x7), y6 -- y7} . |w19#|6{0,1} . |w19#|5{0,1} . .
     |w18#|4{3(x4 — x5), y4 — y5}.                                              % lower stem

"12 point left brace";
call biglbr( '0 10, 9, w0, w1, 0, 12pt).

"24 point left brace";
call biglbr( '032, 11, w10 + .2deltaw, bold + deltaw, ph + pb, 24pt — ph — pb);
call twentyfour.

"30 point left brace";
call biglbr( '050, 11.5, w10 + .4deltaw, bold + 2deltaw, ph + pb, 30pt — ph — pb);
call thirty.

subroutine biglbr: .                                            % free up METAFONT's memory

new w98, w99;   w98 = round(w10 + .6deltaw);   w99 = round( bold + 4deltaw);
"Extensible left brace-top";
call charbegin( '070, 12, 0, 0, 0, 9pt, 0);
varchar '070, '074, '072, '076;                                 % extensible left brace
hpen;   x2 = x3 = good_99(.5r);   x1 — x2 = 4.5u + eps;
top_99 y1 = 0;   y2 = .5[y1, y3];   y3 = round(.5 — 9pt·pixels);
draw |w98#|1{3(x2 -- x1), y2 — y1} |w99#|2{0, —1} . .
     3{0, —1}.                                                  % top of upper stem

"Extensible left brace bottom";
call charbegin( '072, 12, 0, 0, 0, 9pt, 0);
varchar '070, 0, '073, '076;                   % top left, bottom right combination
hpcn;   x6 = x5 = good_99(.5r);   x7 — x6 = 4.5u + eps;
y5 = 0;   y6 = .5[y5, y7];   bot_99 y7 = round(.5 — 9pt·pixels);
draw|w98#|7{3(x6 — x7), y6 — y7} |w99#|6{0, 1} . .
     5{0, 1}.                                                   % bottom of lower stem
```

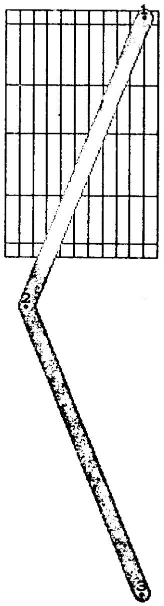


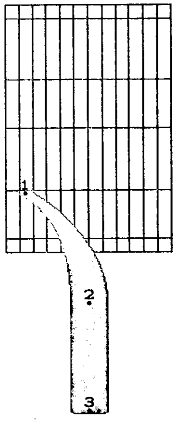


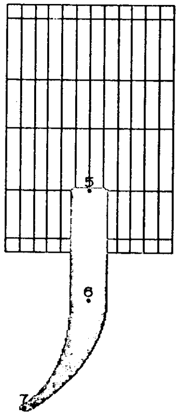


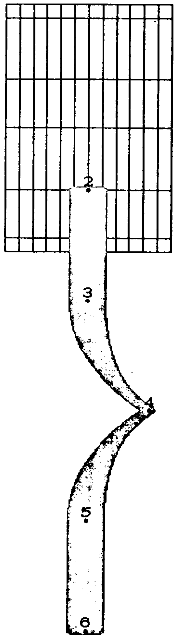
```
'Extensible left brace-middle";
call charbegin( '074, 12, 0, 0, 0,18pt, 0);
varchar 0, 0, 0, '066;                                  % left bracket extension modules only
hpen;  x2 = x6 = x3 = x5 = good99(.5r);  x3 — x4 = 4.5~ + cps;
y2 = good6 .25;  y6 = good6(.5 — 18pt·pixels);  y4 = good6(.5[y2, y6]);
y3 = .5[y2, y4]; y3 — y4 = y4 — y5;
draw |w99#|2 . . |w99#|3{0, —1} . .
     |w98#|4{3(x4 — x3), y4 — y3};                              % bottom of upper stem
draw |w99#|6 . . |w99#|5{0,1} . .
     |w98#|4{3(x4 -- x5), y4 -- y5}.                              % top of lower stem

"Extensible braces--extension module";
call charbegin('076, 12, 0, 0, 0, 3pt, 0);
varchar 0, 0, 0, '076;                                  %brace extension modules only
hpen;  x1 = x2 = good99(.5r);  y1 = 0;  y2 = round(.5 — 3pt·pixels);
w99 draw 1.. 2.                                                  %middle part of stroke

% Right braces (left-right symmetric with left ones)
charlist '011, '033, '051, '071, 0;
subroutine bigrbr(var code, var units, var minps, var maxps, var height, var depth):
call charbegin(code, units, 0, 0, height, depth, 0);
new w48, w49;  w48 = round minps;  w49 = round mnxps;
hpen;  x2 = x3 = x5 = x6 = good 19(.5r);
x1 — x2 = x2 — x4 = —.5(units --- 3)u + eps;  x1 = x7;
top 19y1 = round height·pixels;  hot 19y7 = 1 — round pixels·depth;
y4 = .5[y3, y5] = good6(.5[y1, y7]);
y1 — y2 = y3 — y4 = y6 — y7 = (y1 — y4)/4;
draw |w48#|1{3(x2 — x1), y2 -- y1} . |w49#|2{0, —1} . . |w49#|3{0, —1} . .
     |w48#|4{3(x4 — x3), y4 — y3};                                    % upper stem
d r a w |w48#|7{3(x6 — x7), y6 — y7} . |w49#|6{0, 1} . |w49#|5{0, 1} . .
     |w18#|4{3(x4 — x5), y4 — y5}.                                    % lower stem

"12 point right brace";
call bigrbr('011, 9, w0, w1, 0, 12pt).

"24 point right brace";
call bigrbr('033, 11, w10 + .2deltaw, bold + deltaw, ph + pb, 24pt — ph — pb);
call twentyfour.

"30 point right brace";
call bigrbr( '051, 11.5, w10 + .4deltaw, bold + 2deltaw, ph + pb, 30pt — ph — pb);
call thirty.

subroutine bigrbr: .                                    % free up ME TAFONT's memory
```



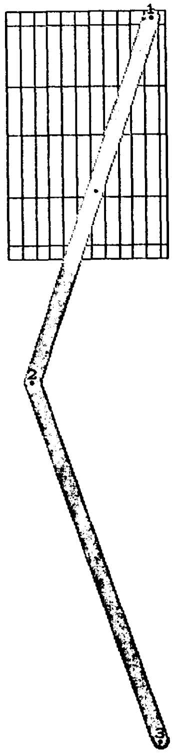


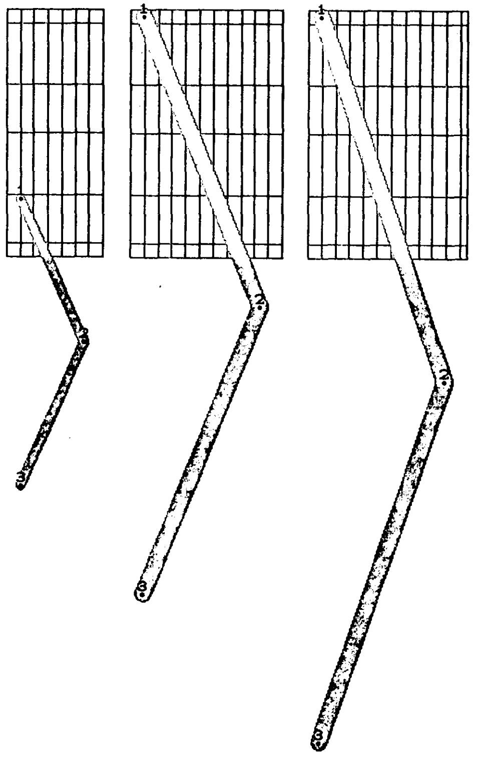


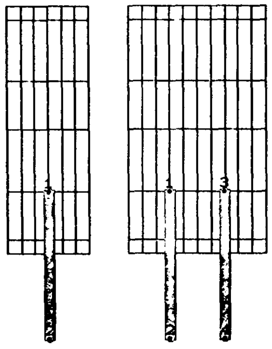




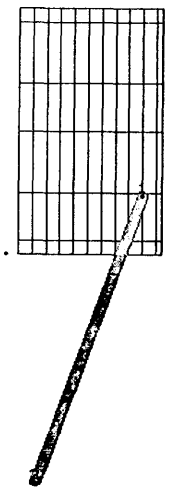
```
new w98, w99;   w98 == round(w10 + .6·leltaw);   w99 == round( bold +4deltaw);
"Extensible right brace top";
call charbegin('071,12, 0, 0, 0, 9pt, 0);
varchar '071, '075, '073, '076;                                %extensible right brace
hpen;   x2 = x3 == good99(.5r);   x1 -- x2 = --4.5u + eps;
top99y1 = 0;   y2 == .5[y1, y3];   yj == round(.5 — 9pt·pixels);
draw |w98#|1{3(x2 — x1), y2 — y1} · |w99#|2{0, —1} . .
     3{0, -1) .                                                % top of upper stem

"Extensible right brace-bottom";
callcharbegin( '073, 12, 0, 0, 0, 9pt, 0);
varchar '071, 0, '072, '076;                   % top right, bottom left combination
hpen;   x6 = x5 = good99( 5r);   x7 — x6 = —4.5u + eps;
y5 == 0;   y6 = .5[y5, y7];   bot99y7 == round(.5 — 9pt·pixels);
draw |w98#|7{3(x6 — x7), y6 — y7} · |w99#|6{0, 1} . .
     5{0, 1}.                                                % bottom of lower stem

"Extensible right brace- middle";
call charbegin( '075, 1 2, 0, 0, 0, 18pt, 0);
varchar 0, 0, 0, '067;                         % right bracket extension modules only
hpen;   x2 == x6 = x3 == x5 == good99(.5r);   x3 — x4 == - 4 5u + eps;
y2 = good6 .25;   y6 == good6(.5 -- 18pt·pixels);   y4 == good6(.5[y2, y6]);
y3 = .5[y2, y4];   Y I — y I == y4 — y5;
draw |w99#|2 . . |w99#|3{0, —1} . .
     |w98#|4{3(x4 — x3), y4 — y3};                            % bottom of upper stem
draw |w99#|6 . . |w99#|5{0, 1} . .
     |w98#|4{3(x4 — x5), y4 — y5}.                              % top of lower stem

% Left angle brackets (left-right symmetric with right ones)
charlist '012, '034, '052;
subroutine biglab(var code, var units, var psize, var height, var depth):
call charbegin(code, units, 0, 0, height, depth, 0);
new w49;   w 19 == round psize;
cpen;   x1 = x3 == good 19(r — u);   lft49x2 = round u;
top49y1 = round pixels·height;   bot49y3 = 1 — round pixels·depth;
y2 = good49(.5[y1, y3]);
w49 draw 1 . . 2 . . 2 . . 3.                                            % stroke

"12 point left angle bracket";
call biglab( '0 12, 7, w10, 0, 12pt).

"24 point left angle bracket";
call biglab('034, I I, w1 — deltaw, ph + pb, 24pt — ph — pb);
call twentyfour.
```

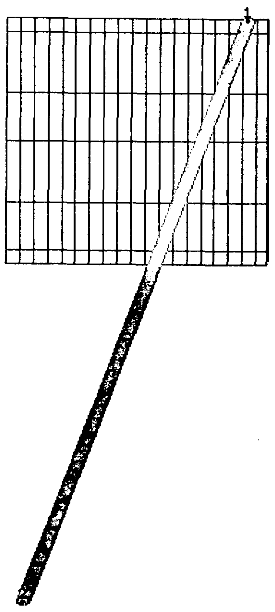


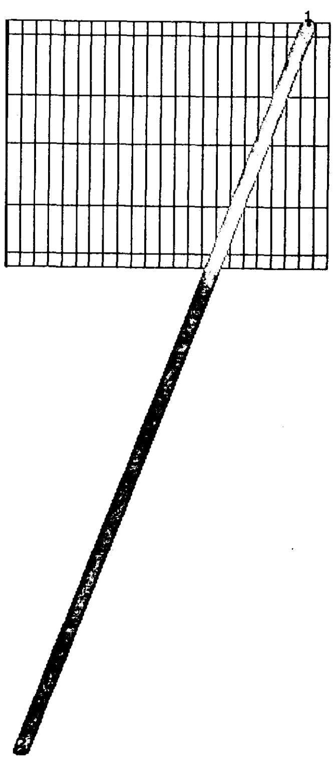


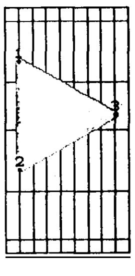


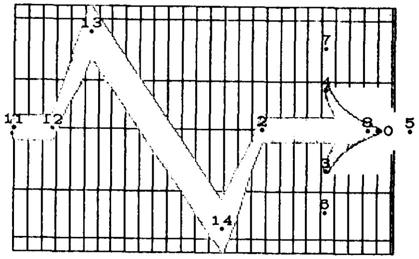
```
"30 point left angle bracket";
call biglab('052, 1 1.5, w1 -- .8deltaw, ph + pb, 30pt — ph — pb);
call thirty.

subroutine biglab:  .                                % free up METAFONT's memory

% Right angle brackets (left-right symmetric with left ones)
charlist '013, '035, '053;
subroutine bigrab(var code, var units, var psize, var height, var depth):
call charbegin(code, units, 0, 0, height, depth, 0);
new w19;   w19 = round psize;
epen;   x1 == x3;   r --- x1 == good19(r -- u);   lft19(r -- x2) == round u;
top19y1 == round pixels·height;   hot19y3 = 1 — round pixels·depth;
y2 == good19(.5[y1, y3]);
w19 draw 1 . 2.. 2.. 3.                                                 % stroke

"1% point right angle bracket";
call bigrab('013, 7, w10, 0, 12pt).

"24 point right angle bracket";
call bigrab('035, 1 1, w1 — deltaw, ph + pb, 24pt — ph -- pb);
call twentyfour.

"30 point right angle bracket";
call bigrab('053, 11.5, w1 -- .8deltaw, ph + pb, 30pt -- ph — pb);
call thirty.

subroutine bigrab:  .                                % free up METAFONT's memory

% Vertical lines
new w99;   w99 == round(w10 + .2deltaw);
"Extensible vertical line extension module";
chnrlist '014, 0;
call charbegin('014, 6, 0, 0, 0, 6pt, 0);
varchnr 0, 0, 0, '014;                                 % extensible verticnl line
hpen;   x1 = x2 == good99 .5r;   y1 == 0;   y2 == round(.5 — 6pt·pixels);
w99 draw 1 . . 2.                                                         % stem

"Extensible double vertical line extension module";
chnrlist '015, 0;
call charbegin('015, LO, 0, 0, 0, 6pt, 0);
varchar 0, 0, 0, '015;                         % extensible double vertical line
hpen;   x1 = x2 = good99 .3r;   x3 == x4 == r — x1;
y1 == y3 = 0;   y2 = y4 == round(.5 -- 6pt·pixels);
w99 draw 1. . 2; draw 3 . . 4.                                           % stems
```



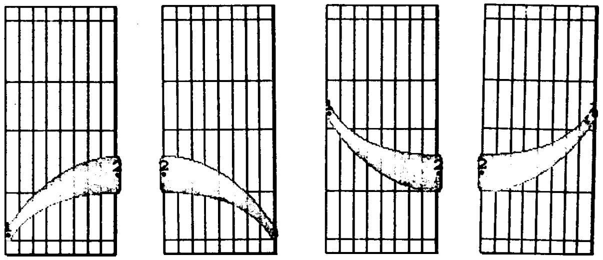


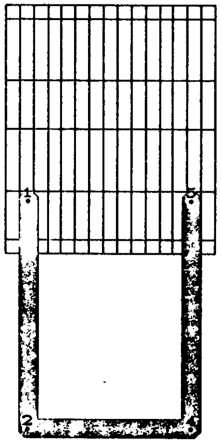






```
% Slashes
charlist´016, ‘036, ‘054;
subroutine bigslash(var code, var units, var psize,
      var height, var depth):
call charbegin(code, units, 0, 0, height, depth, 0);
new $w_{19}$;   $w_{19}$ = round psize;
hpen;   $rt_{19}x_1 = round(r - u)$;   $lft_{19}x_2 = round\ u$;
lop $_{19}y_1 =$ round pixels·height;   hot $_{19}y_2 = 1 -$ round pixels·depth;
$w_{19}$ draw 1.2.                                                      % diagonal

"12 point stash";
call bigslash(´016,10 4, $w_{10} + .2deltaw$, 0, 12pt)

"24 point slash";
call bigslash(´036, 18.8, $w_{10} + deltaw, ph + pb, 24pt - ph -- pb$);
call twentyfour.

"30 point slash";
call bigslash(´054, 23, $w_{10} + 1.5deltaw, ph +$ pb, $30pt -- ph -$ pb);
call thirty.

% l'hr bigslash subroutine is used also to make extrabold stashes


"Black triangle";
call charbegin(´170, 9, 0, 0, 6.25pt, 0, 0);
hpen;   $x_1 = x_2 = good_0 u$;   $x_3 = r - x_1$;   $y_3 = good_6 3.125pt·pixels$;
$y_1 - y_3 = y_3 - y_2 = (x_3 - x_1)/(sqrt 3)$;
$w_0$ ddraw 1 . . 3, 2 . . 3.                                    % fill in the triangle

"Arrow for errata lists";
call charbegin(´171, 27.2, 0, 0, ph + pb, pd + pb, 0);
vpen;   $x_2 = good_0 .65r$;   $rt_5 x_1 = x_0 = round(r - u)$;
$y_0 = y_1 = y_2 = y_5 = y_8 = y_{11} = y_{12} = good_{10} a$;
new $w_{99}$;   $w_{99} = 2w_5$;   $top_{99} y_{13} = h + b$;   $bot_{99} y_{14} = --d -- b$;
$x_{11} = 0$;   $x_{12} = .1r$;   $x_{13} = .2r$;   $x_{14} = .55r$;
draw $|w_5|$11 . . 12 . . $|w_5\#|$12 . . $|w_{99}\#|$13 . . 13 . . 14 . .
      $|w_{99}\#|$14 . . $|w_5\#|$2 . . 2 . . 1;                               % main arrow
hpen;   $rt_3 x_8 = x_0$;
$x_5 -- x_8 = x_8 -- x_3 = 3u + eps$;   $x_3 = x_4 = x_6 = x_7$;
$y_3 -- y_6 = y_1 -- y_3 = y_4 - y_1 = y_7 - y_4 = .24h + eps$;
rpen#;   $w_5 + w_3$ draw (5 . . )8 . . 3( . . 6);           % erase excess at lower right
hpen;   draw ($|w_3|$5 . . )8 . . $|w_0|$3( . . 6);                           % lower point
rpen#;   $w_5 + w_3$ draw (5 . . )8 . . 4( . . 7);           % erase excess at upper right
hpen;   draw ($|w_3|$5 . . )8 . . $|w_0|$4( . . 7).                           % upper point
```

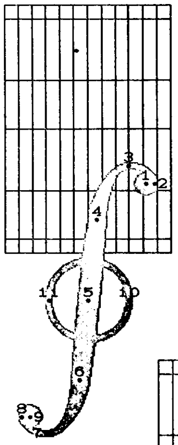


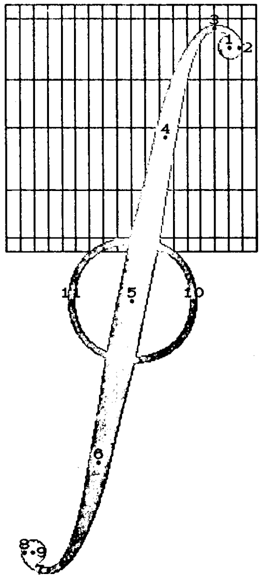


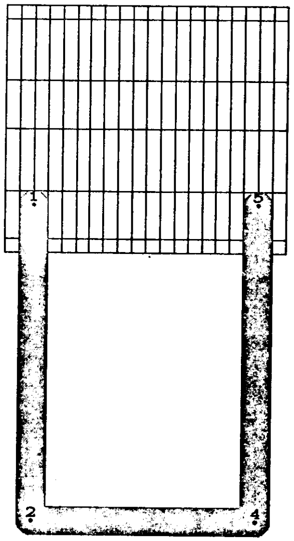
```
% Parts for extensible horizontal bra‹:s to match vertical ones
new rulcht, w98, w99;
ruleht = .5[pwii, pwiii] + 4(pwii — pwi);                    % height of extension rule
w98 == round(w10 + .6deltaw);
w99 == round( bold + 4dcJtaw);                              % corresponds to rule height
"Extensible downwards brace- left";
call charbegin('172,4.5pt/pu, 0, 0, ruleht,0,0);
vpen; lft98x1 = 0; x2 = r + 1;
bot99y2 = 0;   y1 = y2 — 4.5u — cps;
draw |w98#|1{x2 — x1, 3(y2 — y1)} . . |w99#|2{ 1,O}.                    % point

"Extensible downwards brace-right,";
call charbegin('173, 4.5pt/pu,0,0,ruleht,0,0);
vpen;   rt98x1 == r + 1;   x2 = 0;
bot99y2 == 0;   y1 == y2 — 4.5u — eps;
draw |w98#|1{x2 — x1, 3(y2 — y1)} . . |w99#|2{—1, O}.                   % point

"Extensible upwards brace-left.";
call charbegin( '174, 4.5pt/pu, 0, 0, ruleht, 0, 0);
vpen;   lft98x1 = 0;   x2 = r + 1;
bot99y2 == 0;   y1 == y2 + 4.5u + eps;
draw |w98#|1{x2 — x1, 3(y2 - - y1)} . |w99#|2{1, 0}.                    % point

"Extensible upwards brace right";
call charbegin( '175, 4.5pt/pu, 0, 0, ruleht, 0, 0);
vpen;   rt98x1 = r + 1;   x2 = 0;
bot99y2 = 0;   y1 = y2 + 4.5u + eps;
draw |w98#|1{x2 — x1, 3(y2 — y1)} . |w99#|2{—1, 0}.                     % point
```

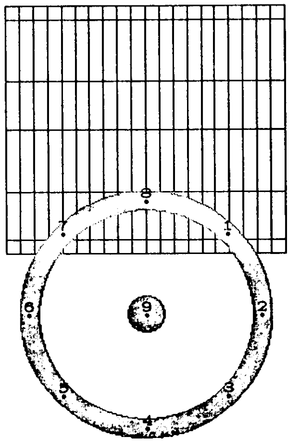
The file mathop. mf

```
% Square union signs
charlist' 106, '107;
subroutine bigsqun(var code, var units, var size, var depth):
call charbegin(code, units, 0, 0, 0, depth, 0);
new w19;   w19 = round size;
cpcn;   lft19x1 = round u;   x2 == x1;   x.1 = x5 = r — x1;
top19y1 = 0;   bot19y2 = 1 — round depth·pixels;
y2 = y4;   y5 = y1;
size draw 2.. 4;                                                         % bar
draw 1 . . 2; draw 4 . . 5.                                              % stems

"12 point square union sign";
call bigsqun('106, 15, w1, 10pt).
```

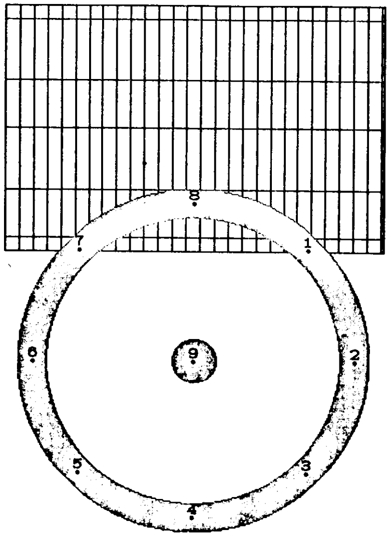


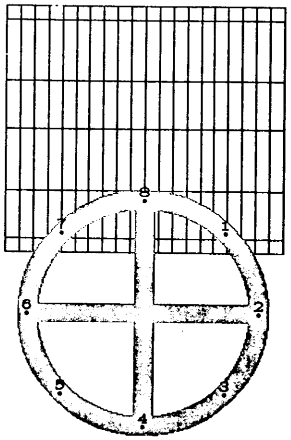


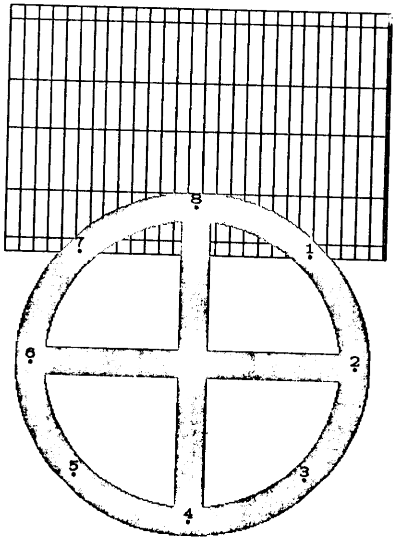


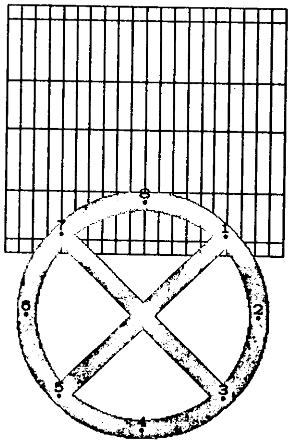


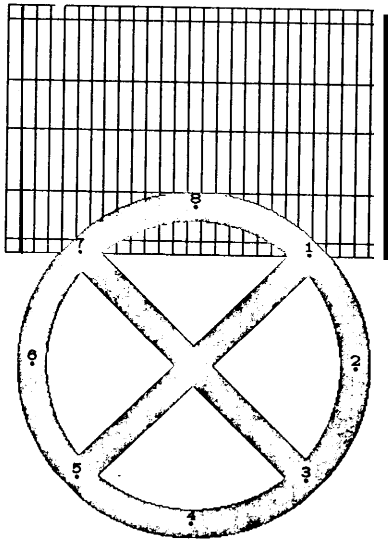
```
"18 point square union sign";
call bigsqun(' 107, 20, w3 + deltaw, 14pt);
call eighteen.

subroutine bigsqun : .                                          % free up METAFONT's memory

% Contoiir integral signs
charlist '110, '111;
subroutine bigoint(var code, var units, var keyps, var maxps,
                   var dotps, var height, var depth, var kerncorr):
call charbegin(code, units, 0, 0, height, depth, kerncorr);
new w48, w 19;   w48 = round maxps;   w49 = round dotps;
hpen;   rt19x1 = rt0x2 = round(r - u);   y1 = y2 = y3 - .75w49;
lft19x9 = lft0x8 = round u;   y8 = y9 = y7 + .75w49;
x3 = r - 3u;   x7 = 3u;
top0y3 = round pixels·height;   bot0y7 = 1 - round pixels·depth;
x5 = good18.5r;   y5 = .5[y3, y7];
x4 = x5 + .3(r - 10u);   y4 = y5 + .3(y3 - y7);
x6 = xj - .3(r - 10u);   y6 = y5 - .3(y3 - y7);
cpen;   w49 draw 1;                                                          % upper bulb
draw 9;                                                                      % lower bulb
hpen;   draw |w0|2{0, 1} . . |w0#|3{-1, 0} . . |keyps|4{x6 -- x4, y6 - y4} |w48#|5 .
     |keyps|6{x6 - x4, y6 - y4} . |w0#|7{-1, 0} . . 8{0, 1};                  % stem
x10 = good0 .75r;   x5 = .5[x10, x11];   y10 = y11 = y5;
w0 draw 10{0, 1} . . 11{0, -1} . 10{0, I} .                                  % bowl

"12 point contour integral sign";
call bigoint(' 110, 12, w1, w2, bold, 10pt/9, 10pt, 3.5pu).

"24 point contour integral sign";
call bigoint( '111, 18, w2, bold + 2deltaw, w3, ph + pb - 8/9pt,
              24pt - ph -- pb - 8/9pt, 8pu);
call twentyfour.

subroutine bigoint: .                                           % free up METAFONT's memory

% Circle-dot operators
charlist '112, '113;
subroutine bigodot(var code, var units, var size, var dotsize, var depth):
call charbegin(code, units, 0, 0, 0, depth, 0);
new w49;   w49 = round size;
cpen;   lft49x6 = round u;   top19y8 = 0;   x2 = r - x6;   bot19y4 = 1 - round depth·pixels;
call circle(1, 2, 3, 4, 5, 6, 7, 8, size);                                   % circle
x9 = .5[x6, x2];   y9 = .5[y8, y4];   dotsize draw 9.                          % dot

"12 point circle-dot operator";
call bigodot('112, 20, w1, bold + 4deltaw, 10pt).
```



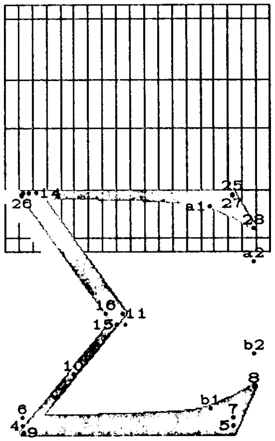


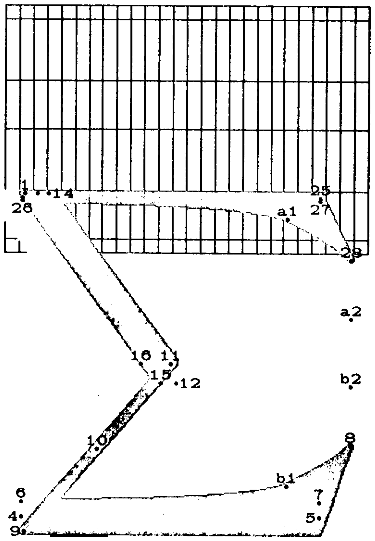






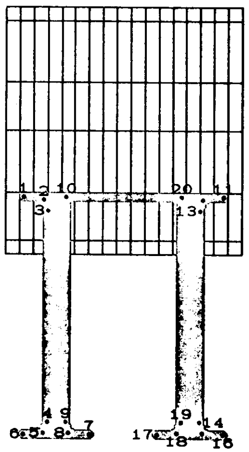
```
´ 18 point circle-dot operator";
call bigodot(´113, 27.2, w3 + deltaw, bold + 6deltaw, 14pt);
call eighteen.

subroutine higodot: .                               % free up METAFONT's memory

% Circle-plus operators
charlist '114, '115;
subroutine bigoplus(var code, var units, var size, var depth):
call charbegin(code, units, 0, 0, 0, depth, 0);
new w19;  w19 = round size;
cpen;  lft 19x6 = round u;  top 19y8 = 0;  x2 = r - x6;  bot 19y4 = 1 -round depth·pixels;
call circle(1, 2, 3, 4, 5, 6, 7, 8, size);                                  % circle
w19 draw 2..6; draw 4..8.                                                   % plus

"12 point circle-plus operator";
call ˙bigoplus(´114, 20, w1, 10pt).

"18 point circle-plus operator";
call bigoplus(´115, 27.2, w3 + deltaw, 14pt);
call eighteen.

subroutine bigoplus: .                              % free up METAFONT's memory

% Circle-times operators
charlist '116, '117;
subroutine bigotimes(var code, var units, var size, var depth):
call charbegin(code, units, 0, 0, 0, depth, 0);
new w19;  w19 = round size;
cpen;  lft 19x6 = round u;  top 19y8 = 0;  x2 = r - x6;  bot 19y1 = 1-round depth·pixels;
call circle(1, 2, 3, 4, 5, 6, 7, 8, size);                                  % circle
w19 draw 1.. 5; draw 3.. 7.                                                 % times

"12 point circle-times operator";
call bigotimes(´116, 20, w1, 10pt).

"18 point circle-times operator";
call bigotimes(´117, 27.2, w3 + deltaw, 14pt);
call eighteen.

subroutine bigotimes: .                             % free up METAFONT's memory
```

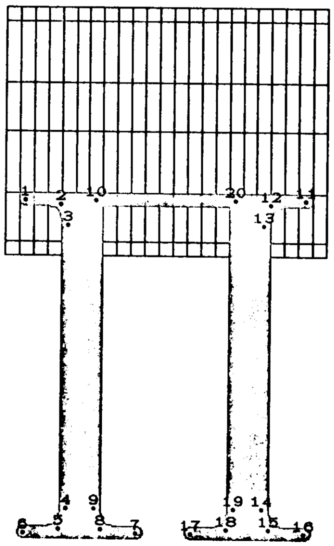




1 3
2 4

```
% Summation signs
charlist '120, '130;
subroutine bigsum(var code, var units, var minps, var maxps, var serif, vnr depth):
call charbegin(code, units, 0, 0, 0, depth, 0);
new w18, w19;   w18 = round minps·aspect;   w19 = round maxps·aspect;
hpen;  lft0x26 = round u;   rt0x28 = r - u;   x27 = good0(x28 - (r - 2u)/11);
new ss;   ss = 1.4aspect·serif·u + eps;
if ss + w6 > .25depth·pixels:   new ss;   ss = .25depth·pixels - w6 + eps;
fi;
vpen;   top18y21 = top6y1 = 0;   y25 = y21;   bot6y26 = bot18y21;
y27 = y26;   y28 = y27 - ss;
bot19y4 = bot6y9 = 1 - round(depth·pixels);   y5 = y4;
top6y6 = top19y1;   y7 = y6;   y8 = y7 + ss;
x24 = x26 = x1 = x6;   x25 = x27 = x5 = x7;   x8 = x28;
call 'a arm(26, 27, 28);                                    % upper arm and serif
call `b arm(6, 7, 8);                                       % lower arm and serif
w18 draw 24 . . 25;   w19 draw 4. 5;                        % thicken bars of arms
if ucs = 0: draw 5.. 8;   w18 draw 25.. 28;                 % thicken bars of arms
else:   if w0 ≠ w1: draw |w19|5 |w6|8;                      % thicken lower serif
            draw |w18|25 . . |w6|28;                        % thicken upper serif
    else: draw 5 . 8; w18 draw 25.. 28;                     % thicken bars of arms
    fi;
fi;
new w18, w19;   w18 = round minps;   w19 = round maxps;
hpen; lft18x1 = lft19x13 = lft0x24;   x9 = x1;   x10 = .5[x9, x11];
x16 = round(u + 1/11(r -- 2u));   lft19x12 = lft0x15;
lft19x99 = x16;   rt19x99 = rt18x11;   rt19x13 = rt18x14;
y10 = .5[y9, y11];   y11 = y16 = .5[y1, y9];   y12 = y15;   y13 = y11 = y1;
new aa, bb;   Ift18x15 = aa[lft18x1, x16];   y15 = aa[y1, y16];
lft18x15 = bb[x9, x11];   y15 = bb[y9, y11];
w19 draw 13. . 12;                                          % upper diagonal
lpen#;   w19 draw l . 15;                  % erase excess at upper left corner
hpen; w18 draw 1. . 15;                           % sharpen upper left corner
rpen#; w19 draw IO. . 11;                              % erase excess at right
lpen#;   w19 draw 9.. 10;                               % erase excess at left
hpen;   w18 draw 9.. 11;                                    % lower diagonal
draw 14.. 11.                                       % sharpen middle corner

"12 point summation sign";
call bigsum('120, 19, w10, w2, ucs, 10pt).

"18 point summation sign";
call bigsum('l30, 26, w10 + deltaw, bold + 4deltaw, 1.8 * ucs, 14pt);
call eighteen.

subroutine bigsum: .                           % free up METAFONT's memory
```



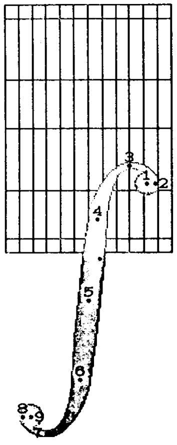


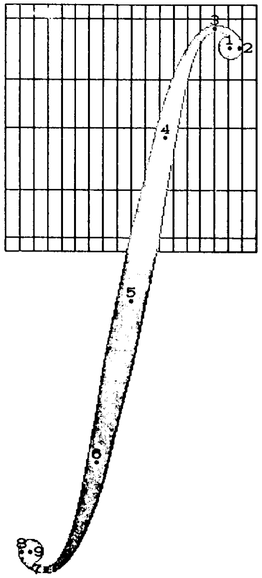
% Product signs
charlist ‘121, ‘131;
subroutine *bigprod*(**var** code, var units, var minps, var maxps,
  var serif, var *ss*, **var** *depth)*:
call *charbegin*(*code*, units, 0, **0**, **0**, *depth*, 0);
new w_{48}, w_{49}; w_{48} = round minps; w_{49} = round maxps;
cpen; $\mathbf{lft}_{48} x_1$ = round u; $x_3 - x_1$ = **round**(*serif*$\cdot u$ + *eps);*
$\mathbf{lft}_{48} x_3 = \mathbf{lft}_{49} x_{99}$; $\mathbf{rt}_{49} x_{99} = \mathbf{rt}_{48} x_{10}$;
$\mathbf{top}_{48} y_1 = 0$; $y_3 = y_1 - ss$; $y_{10} = y_1$;
$x_2 = \frac{1}{3}[x_3, \frac{1}{2}[x_1, x_3]]$; $y_2 = \frac{1}{3}[y_1, \frac{1}{2}[y_1, y_3]]$;
$\mathbf{bot}_{48} y_6 = 1 -$ **round**(*depth*$\cdot$*pixels*);
$y_6 + y_1 = y_5 + y_2 = y_4 + y_3$; $x_6 = x_1$; $x_5 = x_2$; $x_4 = x_3$;
$x_9 = x_{10}$; $x_4 + x_9 = x_5 + x_8 = x_6 + x_7$; $y_4 = y_9$; $y_5 = y_8$; $y_6 = y_7$;
$y_{11} = y_1$; $y_{12} = y_2$; $y_{13} = y_3$; YI I = y_4; $y_{15} = y_5$;
$y_{16} = y_6$; Y17 $= y_7$; $y_{18} = y_8$; $y_{19} = y_9$; $y_{20} = y_{10}$;
$r = x_{11} + x_1 = x_{12} + x_2 = x_{13} + x_3 = x_{14} + x_4 = x_{15} + x_5 =$
  $x_{16} + x_6 = x_{17} + x_7 = x_{18} + x_8 = x_{19} + x_9 = x_{20} + x_{10}$;
w_{48} ddraw 1{1, 0} . . 2 . . 3{0, —1} . . 4{0, —1} . . 5 . . 6{—1, 0},
  10 . . 10 . . 10{0, —1} . . 9{0, —1} . . 8 . . 7{1, 0}; % **lcft** stem and serifs
ddraw 11{—1, 0} . . 12.. 13{0, —1} . . 14{0, —1} . . 15 . . 16{1, 0},
  20.. 20 . . 20{0, —1} . . 19{0, —1} . . 18 . . 17{—1, 0}; % right stem and serifs
draw 10 . . 20. % bar

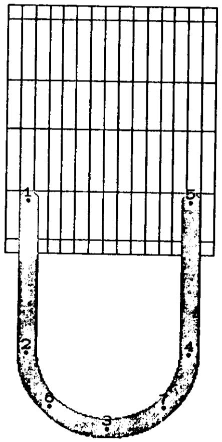
“12 point product sign”;
call bigprod(´121, 17, w_{10}, **bold** + *deltaw*, ucs, s, 10*pt*).

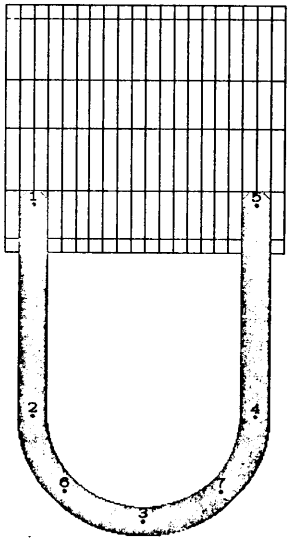
“18 point product. sign”;
call bigprod(‘131, 23, w_{10} + *dcltaw*, bold + 5*dcltaw*, *1.8* * ucs, 1.8*s*, 14pt);
call eighteen.

subroutine *bigprod*: . % free up **METAFONT’s** memory

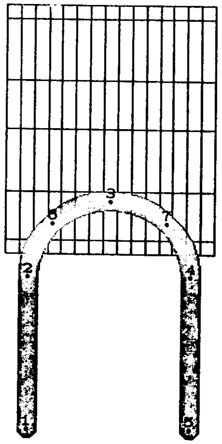


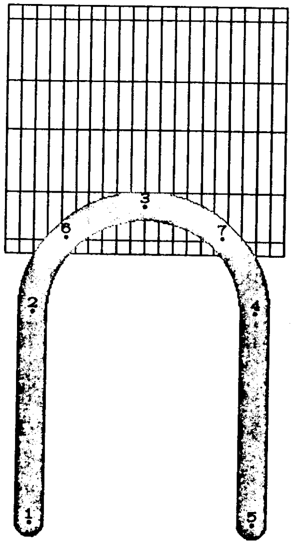


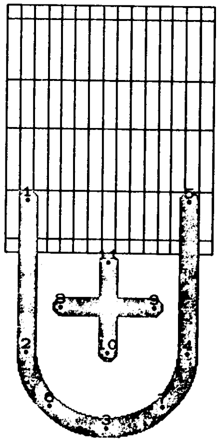


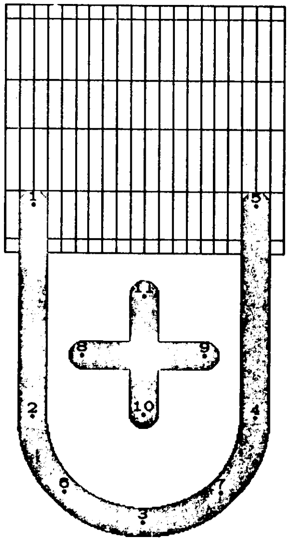


```
% Integral signs
charlist' 122, ' 132;
subroutine bigint(var code, var units, var keyps, var maxps,
                        var dotps, var height, var depth, var kerncorr):
call charbegin(code, units, 0, 0, height, depth, kerncorr);
new w IS, w49;   w48 = round maxps;   w49 = round dotps;
hpen;  rt 19x1 = rt0x2 = round(r — u);   y1 = y2 = y3 — .75w49;
lft 19x9 = lft0x8 = round u;   y8 = y9 = y7 + .75w49;
x3 = r — 3u;   x7 = 3u;
top0y3 = round pixels·height;   bot0y7 = 1 — round pixels·depth;
x5 = good,,, .5r;   y5 = .5[y3, y7];
x4 = x5 + .3(r -- 10u);   y4 = y5 + .3(y3 — y7);
x6 = x5 — .3(r — 10u);   y6 = y5 — .3(y3 — y7);
cpen;  w49 draw 1;                                                    % upper bulb
draw 9;                                                               % lower bulb
hpen;  draw |w0|2{0, 1} |w0#|3{—1, 0} . |keyps|4{x6 — x4, y6 — y4} . |w48#|5 . .
     |keyps|6{x6 -- x 1, y6 — y4} |w0#|7{—1, 0} . . 8{0, 1}.          % stem

"12 point integral sign";
call bigint( '122, 12, w1, w2, bold, 10pt/9, 10pt, 3.5pu).

"24 point, integral sign";
call bigint('132, 18, w2, bold + 2deltaw, w3, ph + pb — 8/9pt,
                   24pt — ph — pb — 8/9pt, 8pu);
call twentyfour.

subroutine bigint: .                          % free up METAFONT's memory

% Set union signs
charlist' 123, ' 133;
subroutine bigun(var code, var units, var size, var depth):
call charbegin(code, units, 0, 0, 0, depth, 0);
new w49;   w 49 = round size;
cpen;  lft 19x1 = round u;   x2 = x1;   x3 = r — x3;   x4 = x5 = r — x1;
top49y1 = 0;   bot49y3 = 1 — round depth·pixels;
y2 = y4 = 2/3[y1, y3];  y5 = y1;
call qcirc(3, 6, 2, size);   call qcirc(3, 7, 4, size);           % cup
size draw 1.. 2; draw 4.. 5.                                        % stems

"12 point set union sign";
call bigun( '123, 15, w1, 10pt).

"18 point set union sign";
call bigun( '133, 20, w3 + deltaw, 14pt);
call eighteen.

subroutine bigun: .                           % free up METAFONT's memory
```









% Set **intersection** signs
charlist ‘124, ‘134;
subroutine *bigin*(**var** code, var units, var *size*, var depth):
call charbegin(code, *units*, 0, 0, 0, depth, 0);
new w_{49}; w_{49} = round size;
cpen; $\textbf{lft}_{49}x_1$ = round u; $x_2 = x_1$; $x_3 = r - x_3$; $x_1 = x_5 = r - x_1$;
$\textbf{top}_{49}y_3 = 0$; $\textbf{bot}_{49}y_1 = 1 -$ round *depth·pixels*;
$y_2 = y_4 = \frac{2}{3}[y_1, y_3]$; $y_5 = y_1$;
call *qcirc*(3, 6, 2, size); call *qcirc*(3, 7, 4, size); % cap
size draw 1. . 2; draw 4. . 5. % stems

“12 point set intersection sign”;
call *bigin*(´124, 15, w_1, 10*pt*).

“18 point set intersection sign”;
call *bigin*(´134, 20, w_3 + *deltaw*, 14*pt*);
call eighteen.

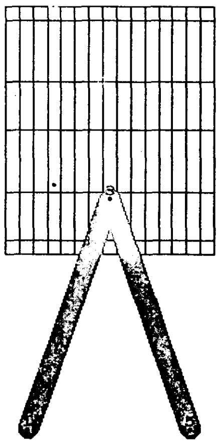
subroutine *bigin*: . % free up **METAFONT**’s memory

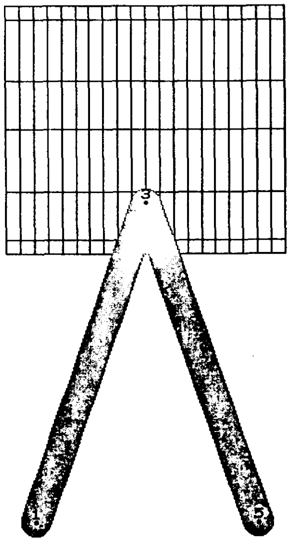
% **Multiset** union signs
charlist ‘125, ‘135;
subroutine *bigmun*(**var** code, var units, var size, var depth):
call *charbegin*(*code*, units, 0, 0, 0, depth, 0);
new w_{49}; w_{49} = round size;
cpen; $\textbf{lft}_{49}x_1$ = round u; $x_2 = x_1$; $x_3 = r - x_3$; $x_4 = x_5 = r - x_1$;
top $_{19}y_1$ = 0; $\textbf{bot}_{49}y_3$ = 1 — round *depth·pixels*;
$y_2 = y_4 = \frac{2}{3}[y_1, y_3]$; $y_5 = y_1$;
call *qcirc*(3, 6, 2, size); call *qcirc*(3, 7, 4, *size*); % cup
size draw 1 . . 2; draw 4.. 5; % stems
$y_8 = y_9 = .47[y_1, y_3]$; $x_8 = r - x_9 = x_1 + 1.75size -$ eps;
$x_{10} = x_1$ I $= x_3$; $.5[y_{10}, y_{11}] = y_8$; $y_{11} - y_{10} = x_9 - x_8$;
draw 8.. 9; draw 10.. 11. % **enclosed** plus sign

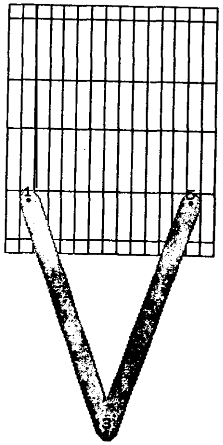
“12 point **multiset** union sign”;
call *bigmun*(´125, 15, w_1, 10*pt*).

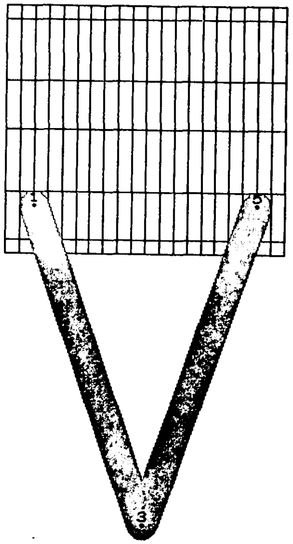
“18 point **multiset** union sign”;
call *bigmun*(´135, 20, w_3 + *deltaw*, 14*pt*);
call eighteen.

subroutine bigmun: . % free up **METAFONT**’s memory

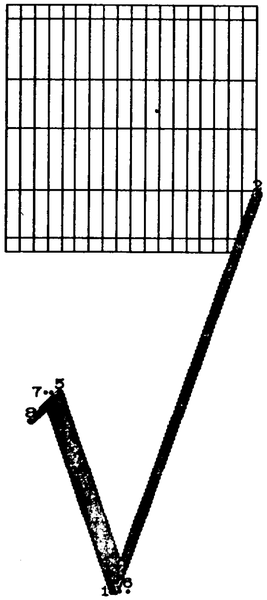


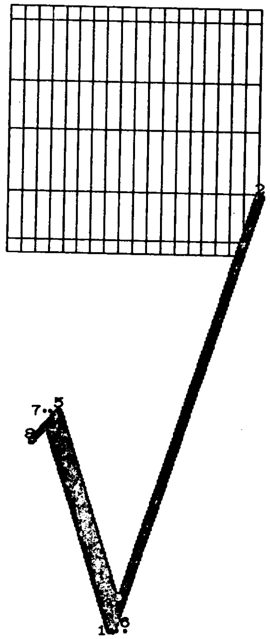


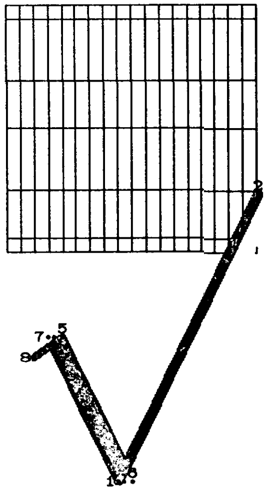


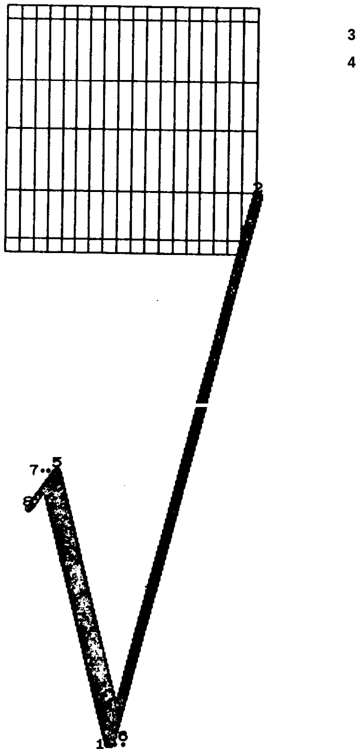


```
% Lattice infimum (logical AND) signs
charlist '126, '136;
subroutine bigmeet(var code, var units, var size, var depth):
call charbegin(code, units, 0, 0, 0, depth, 0);
new w_49;   w_49 = round size;
cpen;   lft_49 x_1 = round u;   x_3 = r - x_3;   x_5 = r - x_1;
top_49 y_1 = 0;   bot _19 y_3 = 1 - round depth·pixels - o;
y_5 = y_1;
size draw 1.. 3; draw 3.. 5.                                   % diagonals

"12 point lattice meet sign";
call bigmeet('126, 15, w_1, 10pt).

"18 point lattice meet sign";
call bigmeet('136, 20, w_3 + deltaw, 14pt);
call eighteen.

subroutine bigmeet: .                    % free up METAFONT's memory

% Lattice supremum (logical OR) signs
charlist '127, '137;
subroutine bigjoin(var code, var units, var size, var depth):
call charbegin(code, units, 0, 0, 0, depth, 0);
new w_49;   w_49 = round size;
cpen;   lft_49 x_1 = round u;   x_3 = r - x_3;   x_5 = r - x_1;
top_49 y_3 = o;   bot_49 y_1 = 1 - round depth·pixels;
y_5 = y_1;
size draw 1.. 3;   draw 3.. 5.                                 % diagonals

"12 point lattice join sign";
call bigjoin('127, 15, w_1, 10pt).

"18 point lattice join sign";
call bigjoin('137, 20, w_3 + deltaw, 14pt);
call eighteen.

subroutine bigjoin: .                    % free up METAFONT's memory
```

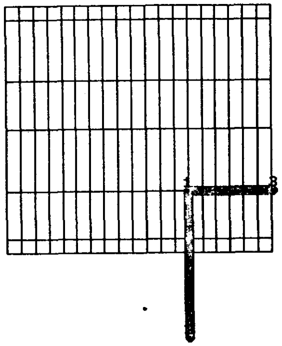


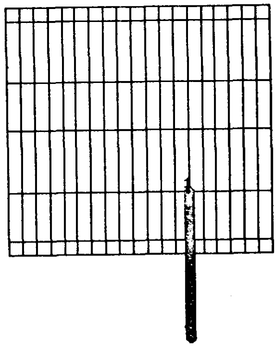


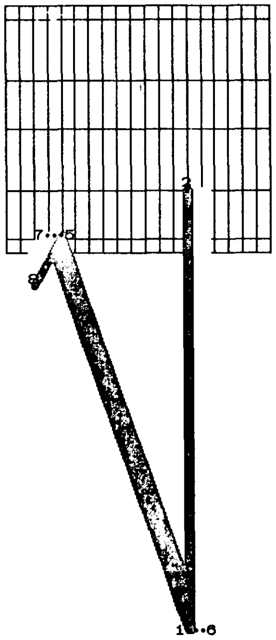


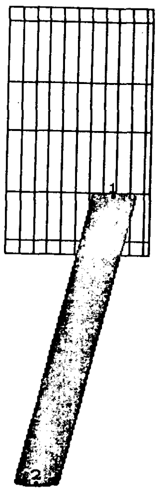


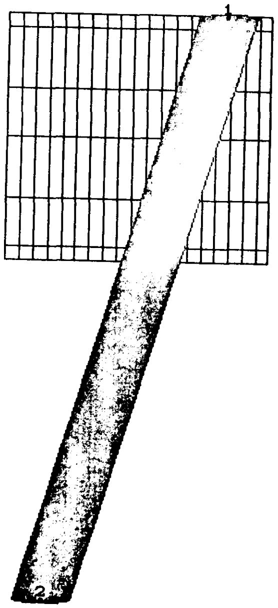
```
% Square root signs
charlist ‘160, ‘161, ‘162, ‘163, ‘164, 0;
subroutine bigroot(var code, var units, var height, var depth):
call charbegin(code, units, 0, 0, height, depth, 0);
hpen;   $x_1 = \mathbf{good}_{10}(\frac{8}{18}r)$;   $x_2 = r + 1$;   $\mathbf{bot}_{10}y_1 = 1 -$ round depth·pixels;
$\mathbf{top}_{10}y_2 = 0$;
$y_3 = y_5 = y_7 = \mathbf{good}_6(.5[y_1, y_2])$;   $y_4 = y_6 = y_1$;
$x_7 = 1.5[x_2, x_1]$;   $\mathbf{lft}_{10}x_7 = \mathbf{lft}_2x_3$;   $\mathbf{rt}_2x_3 = \mathbf{rt}_0x_5$;
$\mathbf{lft}_{10}x_1 = \mathbf{lft}_2x_4$;   $\mathbf{rt}_2x_4 = \mathbf{rt}_0x_6$;
$x_8 = x_7 -$ u;   new aa;   $x_8 = aa[x_5, x_2]$;   $y_8 = aa[y_5, y_2]$;
hpen;   $w_2$ draw 3.. 4;                                      % left diagonal
$w_{10}$ draw 7.. 4;   $w_0$ draw 5.. 6;                      % sharpen the corners
lpen#;   $w_2$ draw 8..5;                             % erase excess at upper left
hpen;   $w_0$ draw 8..5;                                                % serif
rpen#;   $w_2$ draw 1..2;                            % erase excess at lower right
hpen;   $w_{10}$ draw 1.. 2.                                   % right diagonal

“12 point radical sign”;
call bigroot( ‘160, 18, 0, 12pt).

“18 point radical sign”;
call bigroot( ‘161, 18, 0, 18pt);
call eighteen.

“24 point radical sign”;
call bigroot(‘162, 18, ph + pb, 24pt — ph — pb);
call twentyfour.

“30 point radical sign”;
call bigroot(‘163, 18, ph + pb, 30pt — ph — pb);
call thirty.

subroutine bigroot: .                               % free up METAFONT’s memory
```







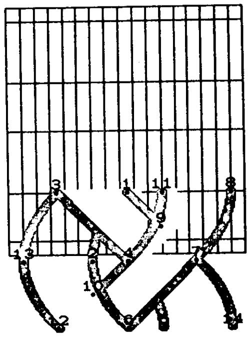


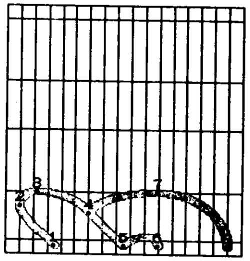


```
"Extensible radical sign-bottom";
call charbegin('164, 19, 0, 0, 0, 18pt, 0);
varchar '166, 0, '164, '166;                                  % extensible radical sign
hpen;  x1 = x2 = good10(r — 6u);  y2 = 0;  bot10y1 = round(.5 — 18pt·pixels);
y3 = y5 = y7 = good6(.1[y2, y1]);  y4 = y6 = y1;
x7 = 3u;  lft10x7 = lft2x3;  rt2x3 = rt0x5;
lft10x1 = lft2x4;  rt2x4 = rt0x6;
x8 = x7 — u;  new aa;  x8 = aa[x5, x2];  y8 = aa[y5, h + b];
hpen;  w2 draw 3.. 4;                                                  % left diagonal
w10 draw 7.. 4;  w0 draw 5.. 6;                                   % sharpen the corners
lpen#;  w2 draw 8.. 5;                                        % erase excess at upper left
hpen;  w0 draw 8.. 5;                                                            % serif
rpen#; w2 draw 1..2;                                         % erase excess at lower right
hpen;  w10 draw 1..2.                                                  % right diagonal

"Extensible radical sign-top";
call charbegin('166, 19, 0, 0, 0, 6pt, 0);
cpen;  x1 = x2 = good10(r — 6u);  x3 = r + 1 ;
y1 = y3 = 0;  y2 = round(.5 — 6pt·pixels);
w10 draw 3.. 1.. 1.. 2.                                                 % link and stem

"Extensible radical sign-extension module";
call charbegin( '165, 19, 0, 0, 0, 6pt, 0);
cpen;  x1 = x2 = good10(r — 6u);  y1 = 0;  y2 = round(.5 — 6pt·pixels);
w10 draw 1..2.                                                                    % stem
```

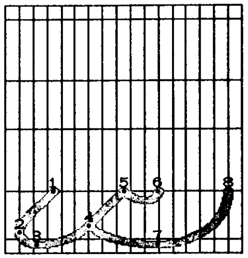
The file mexext . mf

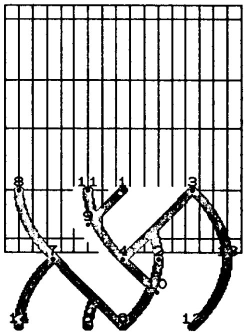
```
% Extrabold slashes
charlist '017, '037;
"12 point extrabold slash";
call bigslash('017, 10.4, bold + 6deltaw, 0, 12pt).

"24 point extrabold slash";
call bigslash('037, 18.8, bold + 10deltaw, ph + pb, 24pt — ph — pb);
call twentyfour.
```





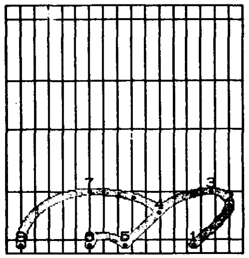
```
% Left pretzels (left-right symmetric with respect to right ones)
"Extensible left pretzel-extension module";
call charbegin('056, 17, 0, 0, 0, 50/9 pt, 0);
varchar '144, 0, '146, '056;                                % the left pretzel
hpen;  x13 = good10 u;  x3 = x12 = 3.5u;  x2 = good10 6u;  x1 = x4 = x6 = 8.521;
x11 = x5 = good,, 11u;  x7 = 13.5u;  x8 = x14 = good10 16u;
y1 = y3 = y8 = y11 = 0;
y4 = -25/9 pt·pixels;  y7 = Ye + 25/9 pt·pixels;  y2 = y13 = .5[y3, y12];
y5 = y6 = y12 = y14 = round(.5 - 50/9 pt·pixels);
x9 = 11u;  y9 = -25/18 pt·pixels;
x10 = 6u;  y10 = y12 + 25/18 pt·pixels;
w10 draw 1{x6 - x7, y6 - y7} .. 2{0, -1};                   % right of upper left strand
rpen#;  u ddraw 3.. 4, 1 .. 4;                   % erase part covered by upper middle strand
hpen;  w10 draw 4{x4 - x3, y4 - y3} .. 5{0, -1};              % left of lower right strand
lpen#;  u ddraw 6.. 7, 6 .. 4;                   % erase part covered by lower middle strand
hpen;  w10 draw 3.. 4;                                       % left of upper middle strand
draw 6.. 7{x7 - x6, y7 - y6} .. 8{0, 1};           % right of lower middle/upper right strand
draw 1 .. 9;                                                % right of upper middle strand
rpen#;  u ddraw 4{x4 - x10, y4 - y10} . 11{0, 1},
      9 .. 11;                                    % erase part covered by upper right strand
hpen;  w10 draw 10..
      4{x4 - x10, Y4 - y10} .. 11{0, 1};          % left of upper right/lower middle strand
lpen#;  u ddraw 2{0, -1). . 6{x9 - x1, y9 - y1},
      2.. 10;                                       % erase part covered by lower left strand
hpen;  w10 draw 12{x3 - x4, y3 - y4} ..
      13{0, 1} .. 3{x10 - x12, y10 - y12};             % left of lower left/upper left strand
draw 2{0, -1} .. 6{x9 - x1, y9 - y1};                         % right of lower left strand
draw 7{x7 - x9, y7 - y9} .. 14{0, -1).             % right of upper middle/lower right strand

"Extensible left pretzel-top";
call charbegin('144, 17, 0, 0, 0, 25/9 pt, 0);
hpen;  x1 = 3.521;  y1 = y5 = y6 = y8 = round(.5 - 20/9 pt·pixels);
x2 = good10 u; y2 = 1/4[y3, y1];
x3 = .5[x1, x2];  y3 = y7 = good6 0 ;
x4 = 6u;  Y4 = y5 + 25/18 pt·pixels;
x5 = 8.5u;  x6 = x7 = 11u;  x8 = good10 16u;
w10 draw 1{x4 - x5, y4 - y5} .. 2{0, 1} .. 3{1, 0} ..
      4{x5 - x.1, y5 - y4} .. 5;                                           % end piece
draw 5{x5 - x4, y4 - y5} .. 6{0, -1};                                   % bottom of twist
draw 4{x5 - x4, y4 - y5} .. 7{1, 0} .. 8{0, -1).                           % top of twist
```

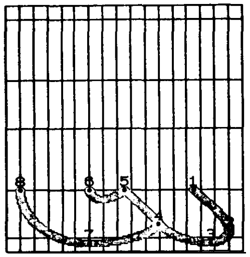


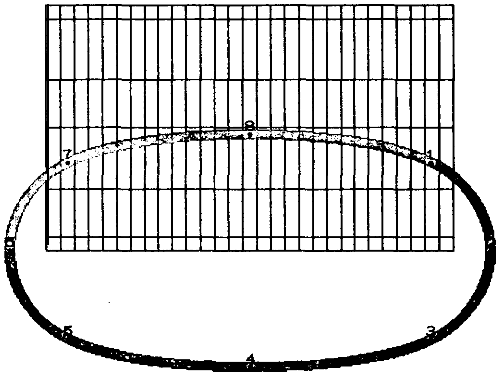


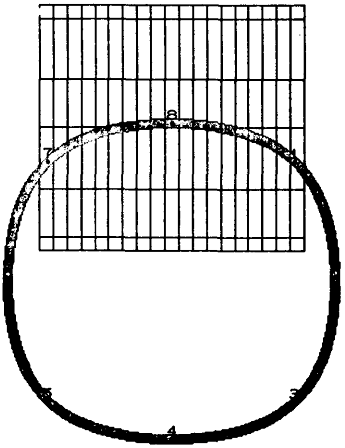
"Ↄxtensible left pretzel-bottom";
call *charbegin*('146, 17, 0, 0, 0, $\frac{25}{9}pt$, 0);
hpen; $x_1 = 3.5u$; $y_1 = y_5 = y_6 = y_8 = 0$;
$x_2 = \mathbf{good}_{10}u$; $y_2 = \frac{1}{4}[y_3, y_1]$;
$x_3 = .5[x_1, x_2]$; $y_3 = y_7 = \mathbf{good}_6(.5 - \frac{20}{9}pt\cdot pixels)$;
$x_4 = 6u$; $y_4 = y_5 - \frac{25}{18}pt\cdot pixels$;
$x_5 = 8.5u$; $x_6 = x_7 = 11u$; $x_8 = \mathbf{good}_{10}\,16u$;
w_{10} draw $1\{x_4 - x_5, y_4 - y_5\}\,.\,2\{0, -1\}\,..\,3\{1, 0\}\,..$
  $4\{x_5 - x_4$, Y5 $- y_4\}\,..\,5$; % end piece
draw $5\{x_5 - x_4, y_4 - y_5\}\,..\,6\{0, 1\}$; % top of twist
draw $4\{x_5 - x_4, y_4 - y_5\}\,..\,7\{1, 0\}\,..\,8\{0, 1\}$. % bottom of twist

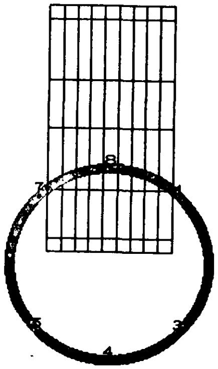
% Right pretzels (left-right symmetric with respect to left ones)
"Extensible right pretzel-extension module";
call *charbegin*('057, 17, 0, 0, 0, $\frac{50}{9}pt$, 0);
varchar '145, 0, '147, '057; % the right pretzel
hpen; $r - x_{13} = \mathbf{good}_{10}u$; $r - x_3 = r - x_{12} = 3.5u$; $r - x_2 = \mathbf{good}_{10}\,6u$; $x_1 = x_4 = x_6 = r - 8.5u$;
x_{11} = x5 = $r - \mathbf{good}_{10}\,11u$; $r - x_7 = 13.5u$; $x_8 = x_{14} = r -$ good,, $16u$;
$y_1 = y_3 = y_8 = y_{11} = 0$;
$y_4 = -\frac{25}{9}pt\cdot pixels$; $y_7 = y_6 + \frac{25}{9}pt\cdot pixels$; $y_2 = y_{13} = .5[y_3, y_{12}]$;
$y_5 = y_6 = y_{12} = y_{14} = \mathbf{round}(.5 - \frac{50}{9}pt\cdot pixels)$;
$r - x_9 = 11u$; $y_9 = -\frac{25}{18}pt\cdot pixels$;
$r - x_{10} = 6u$; $y_{10} = y_{12} + \frac{25}{18}pt\cdot pixels$;
w_{10} draw $1\{x_6 - x_7, y_6 - y_7\}\,..\,2\{0,$ -1); % left of upper right strand
lpen#; u ddraw $3\,..\,4, 1\,..\,4$; % erase part covered by upper middle strand
hpen; w_{10} draw $4\{x_4 - x_3, y_4 - y_3\}\,..\,5\{0, -1\}$; % right of lower left strand
rpen#; u ddraw $6\,..\,7, 6\,..\,4$; % erase part covered by lower middle strand
hpen; w_{10} draw 3.. 4; % right of upper middle strand
draw 6.. $7\{x_7 - x_6, y_7 - y_6\}\,.\,8\{0, 1\}$; % left of lower middle/upper left strand
draw 1. . 9; % left of upper middle strand
lpen#; u ddraw $4\{x_4 - x_{10}, y_4 - y_{10}\}\,..\,11\{0, 1\}$,
  9.. 11; % erase part covered by upper left strand
hpen; w_{10} draw 10..
  $4\{x_4 - x_{10}, y_4 - y_{10}\}\,..\,11\{0, 1\}$; % right of upper left/lower middle strand
rpen#; u ddraw $2\{0, -1\}\,.\,6\{x_9 - x_1, y_9 - y_1\}$,
  2 . . 10; % erase part covered by lower right strand
hpen; w_{10} draw $12\{x_3 - x_4, y_3 - y_4\}\,..$
  $13\{0, 1\}\,..\,3\{x_{10} - x_{12}, y_{10} - y_{12}\}$; % right of lower right/upper right strand
draw $2\{0, -1\}\,..\,6\{x_9 - x_1, y_9 - y_1\}$; % left of lower right strand
draw $7\{x_7 - x_9, y_7 - y_9\}\,..\,14\{0,$ -1). % left of upper middle/lower left strand



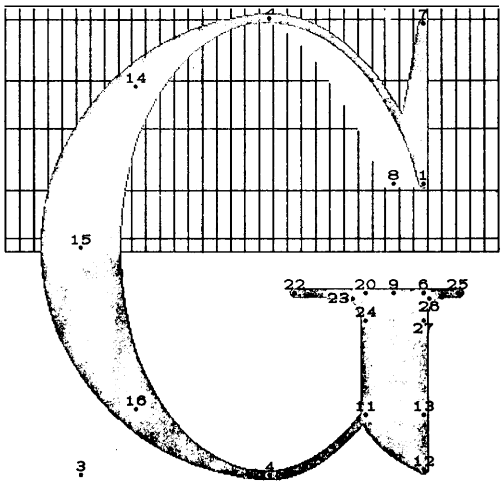




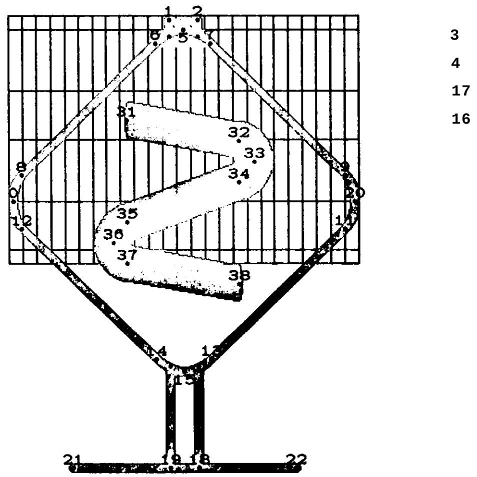




```
"Extensible right pretzel-top";
call charbegin('145, 17, 0, 0, 0, 25/9 pt, 0);
hpen;  r - x_1 = 3.5u;  y_1 = y_5 = ye = y_8 = round(.5 - 20/9 pt·pixels);
r - x_2 = good_10 u;  y_2 = 1/4[y_3, y_1];
x_3 = .5[x_1, x_2];  y_3 = y_7 = good_6 0;
r - x_4 = 6u;  y_4 = y_5 + 25/18 pt·pixels;
r - x_5 = 8.5u;  r - x_6 = r - x_7 = 11u;  r - x_8 = good_10 16u;
w_10 draw 1{x_4 - x_5, y_4 - y_5} . . 2{0, 1} . . 3{-1, 0} . .
     4{x_5 - x_4, y_5 - y_4} . . 5;                              % end piece
draw 5{x_5 - x_4, y_4 - y_5} . . 6{0, -1);                  % bottom of twist
draw 4{x_5 - x_4, y_4 - y_5} . . 7{-1, O}. . 8{0, -1).         % top of twist

"Extensible right pretzel-bottom";
call charbegin( '147, 17, 0, 0, 0, 25/9 pt, 0);
hpen;  r - x_1 = 3.5u;  y_1 = y_5 = y_6 = y_8 = 0;
r - x_2 = good_10 u;  y_2 = 1/4[y_3, y_1];
x_3 = .5[x_1, x_2];  y_3 = y_7 = good_6(.5 - 20/9 pt·pixels);
r - x_4 = 6u;  y 4 = y_5 - 25/18 pt·pixels;
r - x5 = 8.5u;  r - x_6 = r - x_7 = 11u;  r - x_8 = good_10 16u;
w_10 draw 1{x_4 - x_5, y_4 - y_5} . . 2{0, -1} . . 3{-1, 0} . .
     4{x_5 - x_4, Y5 - y_4} . . 5;                               % end piece
draw 5{x_5 - x_4, y_4 - y_5} . . 6{0, 1};                       % top of twist
draw 4{x_5 - x_4, y_4 - y_5} . . 7{-1, 0} . . 8{0, 1}.        % bottom of twist

% Circumscribed circles
subroutine bigcircle(var code, var units, var depth, var asp):
call charbegin(code, units, 0, 0, 0, depth, 0);
new aa;  aa = .5 sqrt(r·r + depth·depth·pixels·pixels/asp/asp);
x_8 = .5r;  y_2 = -.5 round depth·pixels;
x_2 = good_10(x_8 + aa);  y_8 = good_10(y_2 + asp·aa);
call circle(1, 2, 3, 4, 5, 6, 7, 8, w_10).                              % bowl

new sqrtsave;  sqrtsave = sqrttwo;
"Ellipse for floating-point mod operator";
new sqrttwo;  sqrttwo = 1.3195;
call bigcircle('140, 29, 5pt, .5).

"Circle to enclose two digits";
call bigcircle('141, 19, 7.5pt, 1).

"Circle to enclose an exponent of 2";
new sqrttwo;  sqrttwo = sqrtsave;
call bigcircle('142, 9, 6pt, 1).
```



```
“Large G for user manuals”;
call charbegin(´176, 35, 0, 0, ph, 0, 0);
% The depth of this character has been intentionally set to zero.
new w97, w98, w99;
w97 = round(1.5pw·pixels + blacker);
w98 = round(3pwiv·pixels + blacker);
w99 = round(3pwv·pixels + blacker);
cpen;  rt97x1 = rt98x9 = round(12/14 r);  lft99x3 = round r/14;
x2 = x4 = 7.5/14 r;  top97y2 = h + 2o;  bot97y3 = 1 — round(12pt·pixels — 20);
y3 = y4;  y6 = .6[y2, y4];  y1 = good6 9/25 [y2, y3];
x6 = x1;  y9 = good97 3/5[y2, y3];
x 7 = x1;  top97y7 = h;  lft97x8 = lft98x1;  y8 = y1;
hpen; w97 ddraw 1..7, 8..7;                                        % upper serif
lpen#;  w97 ddraw (6 . )1. 2{—1, 0},
     (6. . )8 .. 2{—1, O};                                         % erase spurious part
cpen;  w97 draw (6. . )1 . . 2{—1, 0};                              % upper right stroke
x15 = x3;  y15 = .5[y2, y3];
call hcirc(2, 14, 15, 16, 17, w97);                                % reinforce main stroke
call ` a darc(2, 3, w99);                                           % main stroke
crsbreak .5[y2, y3];
y11 = y10 = y13 = 1/3[y3, y9];
lft97x11 = lft98x9;  rt97x12 = rt97x13 = rt98x9;  y12 = Y J + 2o;
w97 draw 4{1, 0} . . 11 . . 9{0, 1};                                % lower right stroke
x20 = x11; y20 = y9; x21 = x12;  y21 = y9;
ddraw 20.. 11, 21 . . 13;                                           % stem
ddraw 13.. 12, 11{0, —1} . . 12{2(x12 — x11), y12 — y11};           % spur
y22 = y25 = y20;  y24 = y27 = y20 — 2s;
y23 = y26 = 1/3[y20, 1/2[y22, y24]];
x22 = x20 — 2/14 r;  x24 = x20;  x23 = 1/3[x20, 1/2[x22, x24]];
x25 = x21 + r/14;  x27 = x21;  x26 = 1/3[x21, 1/2[x27, x25]];
minvr 0;  minvs 0;
ddraw 22{1, 0} . . 23 . . 24{0, —1}, 25{—1, 0} . . 26 . . 27{0, —1};  % lower serif
minvr 0.5; minvs 0.5.
```



```
' Dangerous bend sign";
call charbegin( '177, 25, 0, 0, ph + pb, 0, 0);
% The depth of this character has been intentionally set to zero.
% The actual depth is 11pt, as desired in the user manuals.
minvr 0;   minvs 0;
cpen;   x1 = good10(.5r — u) = r — x2;                                    % left-right symmetry;
top10y1 = h + b; y2 = y1;
x3 = x, ;   x4 = x2;   x5 = r — x5;   x6 = r — x7;   x8 = r — x9;   x0 = r — x20;
x6 = x5 — 2u;   x8 = u;   lft10x0 = 0;
y5 = good10(25/27(h + b));   y6 = y7 = 23/27(h + b);   y8 = y9 = 4/27(h + b);   y0 = y20 = 0;
y11 + y9 = y12 + Y 8 = y13 + y7 = y14 + y6 =
     y15 + y5 = y16 + y4 = Y 1 7 + y3 = 0;                               % top-bottom symmetry
x11 — x9 = x,2 — x8 = x13 — x7 = x14 — x6 = x15 — x5 =
     x16 — x4 = x,7 — x3 = x,8-x2 = x19 — x1 = 0;
y18 = y19 = y21 = y22;   bot 10y21 = 1 — round(11pt·pixels);
x21 = r — x22 = good10(.5r — 8u);
y3 = Y I = .5[y5, y6];
w10 draw 1 . . 2;                                                        % top of signpost
draw 21.. 22;                                                            % ground level
draw 1. . 3;   draw 19.. 17;                                             % left edge of signpost
draw 2.. 4; draw 18.. 16;                                                % right edge of signpost
rpen#;   draw |1|5{—1, 0} . . |2u|6{x8 — x6, y8 — y6};
     draw |1|15{—1, 0} . . |2u|14{x12 — x14, y12 — y14};                 % erase hidden left edge
lpen#;   draw |1|5{1, 0} . . |2u|7{x9 — x7, y9 — y7};
     draw |1|15{1, 0} . . |2u|13{x11 — x13, y11 — y13};                  % erase hidden right edge
cpen;   w10 draw 5{—1, 0} . . 6{x8 — x6, y8 — y6} . .
     8{x8 — x6, y8 — y6} . . 0{0, —1};                                   % upper left edge of sign
draw 5{1, 0} . . 7{x9 — x7, y9 — y7} . .
     9{x9 — x7, y9 — y7} . . 20{0, —1};                                  % upper right edge of sign
draw 15{—1, 0} . . 14{x12 — x14, y12 — y14} . .
     12{x12 — x14, y12 — y14} . . 0{0,1};                                % lower left edge of sign
draw 15{1, 0} . . 13{XI I — x13, y11 — y13} . .
     11{x1 1 — x13, y11 — y13} . . 20{0,1};                              % lower right edge of sign
new w99;   w99 = round(bold + 4deltaw);
x3, = r -x38 = x5 — 4u;   x32 = x.14 = x38;
x35 =x37 =x3, ;   x33 = r — x36 = good99(x5 + 5u);
y31 = 12/27(h + b);   y32 = 9/27(h + b);   y33 = 6/27(h + b);   y34 = 3/27(h + b);
y31 + y38 = y32 + y J 7 = y33 + y36 = y34 + y35 = 0;
vpen;   w99 draw 31.. 32;                                                % upper bar
draw 38 . . 37;                                                          % lower bar
cpen;   w99 draw 32{x32 — x31, y32 — y31} . . 33{0, —1} . .
     34{x35 — x34, y35 — y34} . . 35{x35 — X34, y35 — y34} . . 36{0, —1} . .
     37{x38 — X37, y38 — y37};                                           % the dangerous bend
crsbreak 0;
minvr .5;   minvs .5.
```

	0	1	2	3	4	5	6	7
‘000	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ
‘010	Φ	Ψ	Ω	1	J		` ´	ˆ
‘020	ˇ	˘	¯	¨	~	→	˝	˚
‘030	¸	ˏ	ˆ	ß	æ	œ	Æ	Œ
‘040	ø	!	”	'	∞	%	&	’
‘050	(	)	*	+	,	-	.	/
'060	0	1	2	3	4	5	6	7
‘070	8	9	:	;	<	=	>	?
’100	0	A	B	C	D	E	F	G
‘110	H	I	J	K	L	M	N	0
’120	P	Q	R	S	T	U	V	W
’130	X	Y	Z	[	“	1	-	—
‘140	‘	a	b	c	d	e	f	g
.150	h	i	j	k	1	m	11	0
‘160	p	q	r	s	t	u	v	w
‘170	x	y	z	ff	fi	fl	ffi	ffl

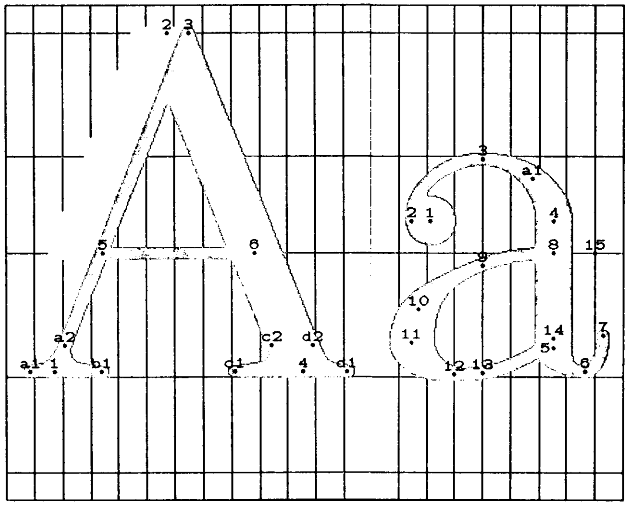
Mathematics booke and journals do not look as beautiful as they used to. It is not that their mathematical content is unsatisfactory, rather that the old and well developed traditions of **typesetting have become too expensive. Fortunately, it now appears that mathematics itself can be used to solve this problem, in spite of the** fact **that the flrat edition of Æsop's Œuvres is naïve about the efficient preparation of flawleae soufflés. This is a sample of the font when the resolution is 200 dote per inch and 3.6 dots per "point".**

STANDARD FONTS

The file `cmr10.mf`

“Computer Modern Roman 10 point”;
$ph = \frac{250}{36}$; $px = \frac{160}{36}$; $pe = \frac{90}{36}$; $pd = \frac{70}{36}$;
$pb = \frac{20}{36}$; $po = \frac{4}{36}$; $ps = \frac{20}{36}$; $pa = .5(ph - pd)$;
$p\ w = \frac{9}{36}$; pwi = $\frac{27}{36}$; pwii = $\frac{32}{36}$; pwiii = $\frac{36}{36}$;
$pwiv = \frac{32}{36}$; $pwv = \frac{38}{36}$; aspect = 1.0;
$pu = \frac{20}{36}$; Jcs = 1.075; *ucs* = 1.7; SC = 0; *ls* = 0;
slant = 0; *sqrttwo* = sqrt 2; ***fixwidth* = *0;***
halfd = 0; varg = 0; Jowast = 0; Jigs = 1.

input cmbase; call ***fontbegin;***
input roman;
end



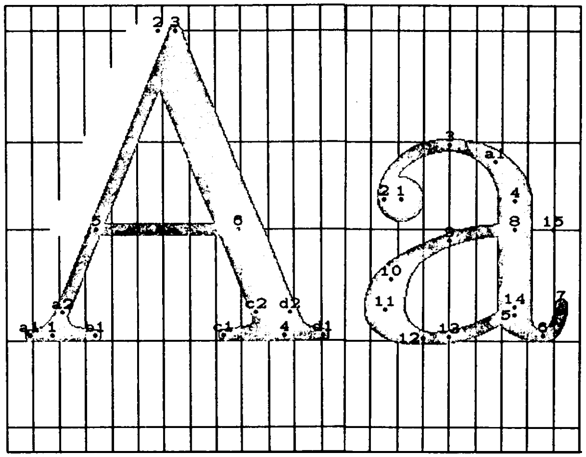
	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ
'010	Φ	Ψ	Ω	ı	ȷ	`	´	^
'020	ˇ	˘	-	¨	~	-	˝	•
'030	¸	,	°	ß	æ	œ	Æ	Œ
'040	ø	!	”	'	∞	%	&	'
'050	(	)	*	+	,	-	.	/
'060	0	1	2	3	4	5	6	7
'070	8	9		;	<	=	>	?
'100	0	A	B	C	D	E	F	G
'110	H	I	J	K	L	M	N	0
'120	P	Q	R	S	T	U	V	W
'130	X	Y	Z	[	“	]	–	—
'140	‘	a	b	c	d	e	f	g
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	ff	fi	fl	ffi	ffl

Mathematics books and **journals** do not look **as** beautiful **as** they **used** to. It **is** not that their mathematical content **is unsatisfactory,** rather that the old and well developed **traditions** of **typesetting** have become too expensive. Fortunately, it now **appears** that **mathematics itself can** be **used** to **solve** thir problem, in **spite** of the *fact* that the **first** edition of **Æsop's Œuvres is naïve** about the **efficient** preparation of **flawless soufflés.** Thir **is** a **sample** of the font when the **resolution is** 200 dotr per inch and 3.6 dotr per **"point".**

The file cmr9.mf

"Computer Modern Roman 9 point";
$ph = \frac{225}{36}$; $px = \frac{144}{36}$; $pe = \frac{81}{36}$; $pd = \frac{63}{36}$;
p b = $\frac{18}{36}$; $po = \frac{4}{36}$; $ps = \frac{18}{36}$; $pa = .5(ph - pd)$;
p w $= \frac{9}{36}$; $pwi = \frac{25}{36}$; p w i i = $\frac{29}{36}$; p w i i i = $\frac{33}{36}$;
p w i v $= \frac{29}{36}$; pwv = $\frac{33}{36}$; aspect = 1.0;
$pu = \frac{18.5}{36}$; $lcs = 1.05$; $ucs = 1.65$; s c = 0; ls = 0;
slant = 0; *sqrttwo* = sqrt 2; *fixwidth* = 0;
halfd = 0; v a r g = 0; *lowast* = 0; *ligs* = 1.

input cmbase; call *font* begin ;
input roman;
end



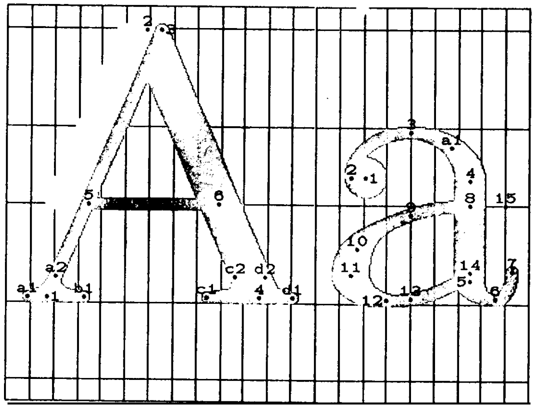
	0	1	2	3	4	5	6	7
‘000	I-	A	Θ	Λ	Ξ	Π	Σ	Υ
‘010	Q	Ψ	n	ı	ȷ	`	´	ˆ
‘020	ˇ	˘		''		-	˝	˚
’030	¸	´	ˆ	ß	æ	œ	Æ	Œ
‘040	ø	!	"	'	∞	%	&	’
‘050	(	)	•	+	,	-	.	/
’060	0	1	2	3	4	5	6	7
‘070	8	9	:	;	<	=	>	?
’100	Ø	A	B	C	D	E	F	G
‘110	H	I	J	K	L	M	N	O
‘120	P	Q	R	S	T	U	V	W
‘130	X	Y	Z	[	“	]	–	—
. 140	‘	a	b	C	d	e	f	g
‘150	h	i	j	k	I	m	n	0
’160	p	q	r	a	t	u	v	w
‘170	x	y	Z	ff	fi	fl	ffi	a ,

Mathematics books **and journals** do not look **as beautiful as** they **used** to. **It is** not that their **mathematical** content **is unsatisfactory, rather that** the old **and well** developed **traditions** of **typesetting have** become too **expensive. Fortunately,** it now **appears that mathematics itself can** be **used** to **solve this** problem, in **spite** of the fact **that** the **first édition** of **Æsop's Œuvres is naïve about** the **efficient preparation** of **flawless soufflés.** This **is** 8 **sample** of the font **when** the **resolution is 200** dotr per inch **and** 3.6 **dots** per ‘point*.

The file `cmr8.mf`

“Computer Modern Roman 8 point”;
$ph = \frac{200}{36}$; $px = \frac{128}{36}$; $pe = \frac{72}{36}$; $pd = \frac{56}{36}$;
$pb = \frac{16}{36}$; $po = \frac{3}{36}$; $ps = \frac{16}{36}$; $pa = .5(ph - pd)$;
$pw = \frac{9}{36}$; $pwi = \frac{23}{36}$; $pwii = \frac{27}{36}$; $pwiii = \frac{30}{36}$;
pwiv = $\frac{27}{36}$; $pwv = \frac{30}{36}$; **aspect = 1.0;**
$pu = \frac{17}{36}$; *lcs* **= 1.03; ucs = 1.56;** SC **= 0; Is = 0;**
slant = 0; *sqrttwo* = sqrt 2; *fixwidth* **= 0;**
halfd **= 0; varg = 0;** *lowast* **= 0;** *ligs* **= 1.**

input `cmbase`**; call fontbegin;**
input roman;
end



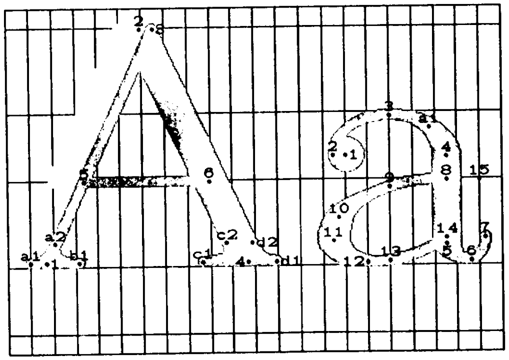
	0	1	2	3	4	5	6	7
'000	I·	A	Θ	Λ	8	Π	Σ	T
'010	Q	♥	n	ı	J	·	·	·
'020	·	˘			˜	¯	˝	·
'030	.	,	˜	ß	æ	a	Æ	Œ
'040	ø	!	"	I	∞	%	&	'
'050	(	)	•	✉	📪	-	ʃ	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	;	<	=	>	?
.100	Ø	A	B	C	D	E	F	G
● 110	II	I	J	K	L	M	N	O
.120	P	Q	R	S	T	U	V	W
'130	X	Y	Z	[	"	]		—
'140	'	a	b	c	d	e	f	g
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	Y	z	ff	fi	fl	ffi	ffl

Mathematics books 8nd **journals** do not look **as** beautiful **as** they **used** to. It **is** not that their mathematical content **is unsatisfactory,** rather that the old **and** well developed **traditions** of **typesetting** have become too **expensive.** Fortunately, it now **appears** that **mathematics itself can** be ured to **solve this** problem, in **spite** of the **fact that** the **first** edition of **Æsop's Œuvres is naïve** about the **efficient** preparation of **flawless soufflés.** Thin in **a sample** of the font when the **resolution** in **200** dotn per inch and 8.6 **dots** per **"point".**

The file cmr7.mf

"Computer Modern Roman 7 point";
$ph = \frac{175}{36}$; $px = \frac{112}{36}$; $pe = \frac{63}{36}$; $pd = \frac{49}{36}$;
$pb = \frac{14}{36}$; $po = \frac{3}{36}$; $ps = \frac{14}{36}$; $pa = .5(ph - pd)$;
pw = $\frac{8}{36}$; pwi = $\frac{21}{36}$; pwii = $\frac{24}{36}$; *pwiii* = $\frac{26}{36}$;
$pwiv = \frac{24}{36}$; $pwv = \frac{26}{36}$; $aspect = 1.0$;
pu = $\frac{10}{36}$; *lcs* = .97; ucs -- 1.44; SC = 0; Is = 0;
slant = 0; sqrttwo = sqrt 2; *fixwidth* = 0;
halfd = 0; varg = 0; *lowast* = 0; *ligs* = 1.

input cmbase; call fontbegin;
input roman;
end



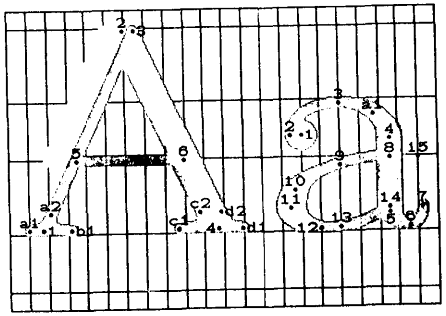
	0	1	2	3	4	5	6	7
‘000	Γ	A	Θ	Λ	Ξ	Π	Σ	Υ
‘010	Φ	Ψ	Ω	ı	ȷ	`	´	ˆ
‘020	ˇ	˘	-	-	-	-	˚	•
‘030	¸	ˏ	˜	ß	æ	œ	Æ	Œ
‘040	ø	!	”	’	∞	%	&	’
‘050	(	)	*	+	,	-	.	/
’060	0	1	2	a	4	5	6	7
‘070	8	9	:	;	<		>	?
‘100	Ø	A	B	C	D	E	F	G
’110	H	I	J	K	L	M	N	0
‘120	P	Q	R	S	T	U	V	W
‘130	X	Y	Z	[	•	]	–	—
‘140	‘	a	b	c	d	.	f	g
‘150	h	I	J	k	l	m	n	o
’160	p	q	r	s	t	u	v	w
‘170	x	Y	z	ff	fi	fl	ffi	ffl

Mathematics books and journals do not look as beautiful as they used to. It is not that their mathematical content is unsatisfactory, rather that the old and well developed traditions of typesetting have become too expensive. Fortunately, it now ● pp*wr that mathematics itself can be used to solve this problem, in spite of the fact that the first edition of Æsop's Œuvres is naïve about the efficient preparation of flawless soufflés. This is . sample of the font when the resolution is 200 dote per inch and 3.6 dote per ● point”.

The file cmr6. mf

“Computer Modern Roman 6 point”;
$ph = \frac{150}{36}$; $px = \frac{96}{36}$; $pe = \frac{54}{36}$; p d $= \frac{42}{36}$;
$pb = \frac{12}{36}$; $po = \frac{2}{36}$; $ps = \frac{12}{36}$; $pa = .5(ph - pd)$;
$pw = \frac{8}{36}$; $pwi = \frac{19}{36}$; $pwii = \frac{22}{36}$; pwiii $= \frac{23}{36}$;
pwiv $= \frac{22}{36}$; $pwv = \frac{23}{36}$; aspect = 1.0;
$pu = \frac{14}{36}$; $lcs = .89$; ucs = 1.43; sc = 0; $ls = 0$;
slant = 0; sqrttwo = sqrt 2; $fixwidth = 0$;
$halfd = 0$; $varg = 0$; $lowast = 0$; ligs = 1.

input cmbase; call fontbegin;
input roman;
end



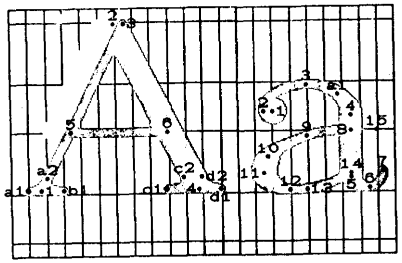
	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ
'010	Φ	Ψ	Ω	ı	ȷ	·	·	·
'020	·	·	-	-	~	~	·	·
'030	.	,	·	ß	æ	œ	Æ	Œ
'040	ø	!	"	'	∞	%	&	'
'050	(	)	*	+	,	-	.	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	;	<	=	>	?
'100	Ø	A	B	C	D	E	F	a
'110	H	I	J	K	L	M	N	0
'120	P	Q	R	S	T	U	V	W
'130	X	Y	Z	|	·	|		—
'140	'	a	b	c	d	e	f	g
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	ff	fi	fl	ffi	ffl

Mathematics books and journals do not look as beautiful as they used to. It is not that their mathematical content is unsatisfactory, rather that the old and well developed traditions of typesetting have become too expensive. Fortunately, it now appears that mathematics itself can be used to solve this problem, in spite of the fact that the first edition of Æsop's Œuvres is naïve about the efficient preparation of flawless soufflés. This is a sample of the font when the resolution is 200 dots per inch and 2.8 dots per "point".

The file cmr5. mf

"Computer Modern Roman 5 point";
$ph = \frac{125}{36}$; $px = \frac{80}{36}$; $pe = \frac{45}{36}$; $pd = \frac{35}{36}$;
$pb = \frac{10}{36}$; $po = \frac{2}{36}$; $ps = \frac{10}{36}$; $pa = .5(ph - pd)$;
$pw = \frac{7}{36}$; $pwi = \frac{17}{36}$; $pwii = \frac{19}{36}$; $pwiii = \frac{20}{36}$;
$pwiv = \frac{19}{36}$; $pwv = \frac{20}{36}$; aspect = 1.0;
$pu = \frac{12.5}{36}$; $lcs = .84$; ucs = 1.32; sc = 0; ls = 0;
slant = 0; $sqrttwo$ = sqrt 2; $fixwidth$ = 0;
$halfd$ = 0; varg = 0; $lowast$ = 0; $ligs$ = 1.

input cmbase; call fontbegin;
input roman;
end



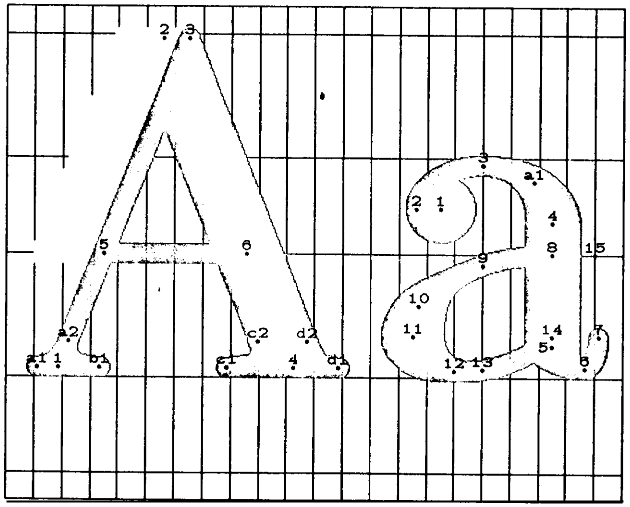
	0	1	2	3	4	5	6	7
‘000	Γ	Δ	Θ	Λ	Ξ	‖	Σ	Υ
‘010	Φ	Ψ	Ω	ı	ȷ	`	´	ˆ
‘020	ˇ	˘	¯	¨	˜	→	˝	•
‘030	¸	ʼ	ˆ	ß	æ	œ	Æ	Œ
‘040	ø	!	”	’	∞	%	&	’
‘050	(	)	•	▭	,	-	[illegible]	/
‘060	♋	1	2	3	4	5	6	7
‘070	8	9	:	;	<	=	>	?
‘100	□	A	B	C	D	E	F	G
‘110	H	I	J	K	L	M	N	0
‘120	P	Q	R S	T	U	V	W	
‘130	X	Y	Z	[	“	]	–	—
.140	‘	a	b	c	d	e	f	g
‘150	h	i	J	k	l	m	n	0
‘160	p	q	r	s	t	u	v	w
‘170	x	Y	z	ff	fi	fl	ffi	ffl

Mathematics books and Journal, do not look **as** beautiful **as they used** to. It **is** not that their mathematical content **is unsatisfactory,** rather that **the** old and **well** developed **traditions** of **typesetting have** become too txptnrivt. Fortunately, it now **appears** that mathtmaticr **itself** can **be used** to rolvt thir problem, in rpitt of **the** *fact* that the **first** edition of **Æsop's Œuvres is naïve** about the **efficient** preparation of **flawless soufflés. This is** *a* **sample** of **the** font **when the resolution is 200** dotr **per** inch and **3.6** dotr **per** ‘point”.

T:ie file cmb10. mf

“Computer Modern Bold Roman 10 point”;
$ph = \frac{250}{36}$; $px = \frac{160}{36}$; $pe = \frac{90}{36}$; $pd = \frac{70}{36}$;
$pb = \frac{20}{36}$; $po = \frac{4}{36}$; $ps = \frac{20}{36}$; $pa = .5(ph - pd)$;
pw = $\frac{15}{36}$; pwi = $\frac{40}{36}$; pwii = $\frac{45}{36}$; pwiii = $\frac{50}{36}$;
pwiv = $\frac{50}{36}$; *pwv* = $\frac{50}{36}$; aspect = 1.0;
$pu = \frac{20}{36}$; *lcs* = .85; ucs = 1.5; sc = 0; *ls* = 0;
slant = 0; *sqrttwo* = 1.375; *fixwidth* = 0;
halfd = 0; varg = 0; lowast = 0; *ligs* = 1.

input cmbase; call fontbegin;
input roman;
end



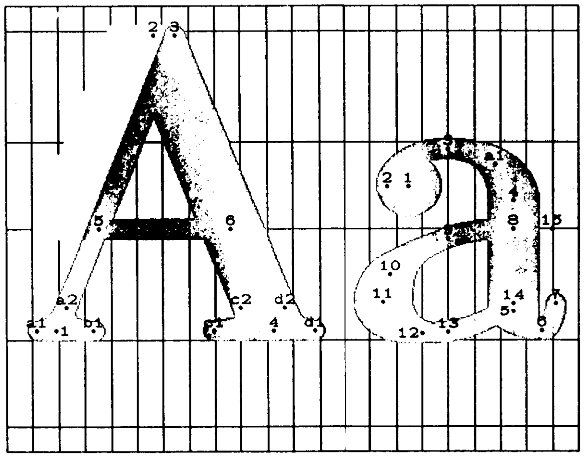
	0	1	2	3	4	5	6	7
‘000	Γ	A	Θ	Λ	Ξ	Π	Σ	Υ
‘010	●	●	Ω		ȷ	`	´	ˆ
‘020		˘		-	-	-	-	•
‘030	.	ˏ	ˆ	ß	æ	œ	Æ	Œ
‘040	ø	!	"	'	∞	%	&	'
‘050	(	)	•	+	,	-	.	/
’060	0	1	2	3	4	5	6	7
‘070	8	a	:	;	<	=	>	?
‘100	Ø	A	B	C	D	E	F	a
‘110	H	I	J	K	L	M	N	0
‘120	P	Q	a	S	T	U	V	W
‘130	X	Y	Z	[	"	]	-	—
‘140	‘	a	b	c	d	e	f	g
’150	h	i	j	k	l	m	n	o
’160	p	q	r	s	t	u	v	w
’170	x	y	z	ff	fi	fl	ffi	ffl

Mathematics bookr ● nd **journals do** not **look a** beautiful **as they used to.** It **is** not that their **mathematical** content **is unsatisfactory,** rather that the old and **well** developed **traditions** of **typesetting have become** too **exp nsive. Fortunately,** it **now** ● ppearr that **mathematics itself can be used** to **solve** thir p ɔ**blem,** in **spite** of **the fact that** the **first** edition of **Æsop's Œuvres is naïve about** the **fficient preparation** of **flawless soufflés. This is a sample** of the font when the **resoluti n is 200** dotr per **inch and 3.6** dotr per “point”.

The file cmb9. mf

“Computer Modern Bold Roman 9 point”;
$ph = \frac{225}{36}$; $px = \frac{144}{36}$; $pe = \frac{81}{36}$; $pd = \frac{63}{36}$;
$pb = \frac{18}{36}$; $po = \frac{4}{36}$; $ps = \frac{18}{36}$; $pa = .5(ph - pd)$;
$pw = \frac{15}{36}$; $pwi = \frac{37}{36}$; $pwii = \frac{42}{36}$; $pwiii = \frac{46}{36}$;
$pwiv = \frac{46}{36}$; pwv = $\frac{46}{36}$; aspect = 1.0;
$pu = \frac{18.5}{36}$; $lcs = .84$; ucs = 1.46; SC = 0; Is = 0;
slant = 0; sqrttwo = 1.375; *fixwidth* = 0;
halfd = 0; varg = 0; *lowast* = *0*; *ligs* = 1 .

input cmbase; call fontbegin;
input roman;
end



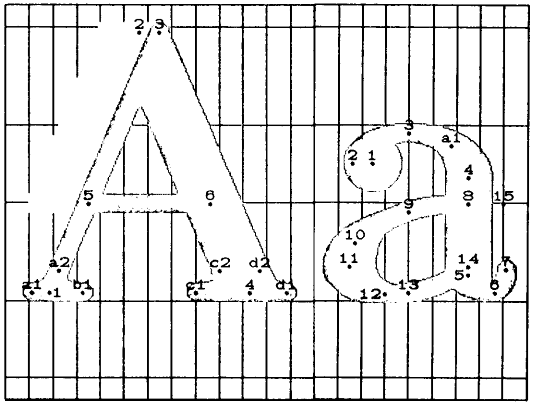
	0	1	2	3	4	5	6	7
‘000	Γ	A	Θ	A	Ξ	Π	Σ	Υ
‘010	●	●	Ω	[illegible]	ȷ	[illegible]	[illegible]	[illegible]
‘020	˘	˘	-	-	-	-	•	▪
‘030	¸	ˏ	ˆ	ß	æ	œ	Æ	Œ
‘040	ø	!	"	'	∞	%	&	'
‘050	(	)	•	+	,	•	.	/
’060	0	1	2	3	4	5	6	7
‘070	8	9	:	;	<	=	>	?
’ 100	ø	A	B	a	D	E	F	a
‘110	H	I	J	K	L	M	N	O
‘120	P	Q	R	S	T	U	V	W
’ 130	X	Y	Z	[	·	]	–	—
‘140	‘	a	b	c	d	0	f	g
‘150	h	i	j	k	l	m	n	o
’160	p	q	r	s	t	u	v	w
‘170	x	y	z	ff	fi	fl	ffi	ffl

Mathematics books ● md **journals** do not look **as beautiful as** thy **used** to. It **is** not **that their mathematical content is unsatisfactory,** rather that **the old** ● md mii **developed traditions** of **typesetting have become** too **expensive. Fortunately,** it **now appears that mathematics itself can be used** to **solve this** probkm, in **spite** of **the fact that the first** ● ditimn of **Æsop's Œuvres is naïve** ● bomt **the efficient preparation** of **flawless soufflés. This is** • **sample** of **the** font **when the resolution is 200** dotr **per inch and 3.6** dotr **per "point".**

The file cmb8.mf

“Computer Modern Bold Roman 8 point”;
$ph = \frac{200}{36}$; $px = \frac{128}{36}$; $pe = \frac{72}{36}$; $pd = \frac{56}{36}$;
$pb = \frac{16}{36}$; $po = \frac{3}{36}$; $ps = \frac{16}{36}$; ‘pa = .5($ph —$ pd);
$pw = \frac{14}{36}$; $pwi = \frac{34}{36}$; $pwii = \frac{38}{36}$; $pwiii = \frac{42}{36}$;
$pwiv = \frac{42}{36}$; pwv = $\frac{42}{36}$; aspect = 1.0;
pu = $\frac{17}{36}$; lcs = .82; ucs = 1.41; sc = 0; Is = 0;
slant = 0; sqrttwo = 1.375; $fixwidth$ = 0;
$halfd$ = 0 ; $varg$ = 0; $lowast$ = 0; $ligs$ = 1 .

input cmbase; call fontbegin;
input roman;
end



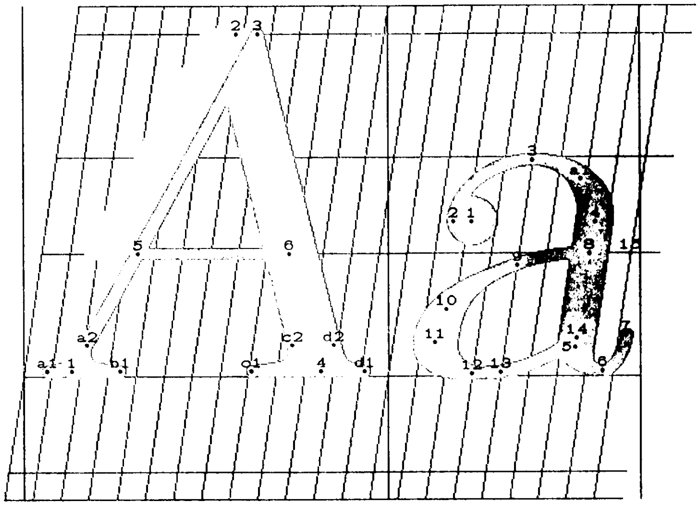
	0	1	2	3	4	5	6	7
´000	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ
´010	Φ	Ψ	Ω	ı	ȷ	`	.	^
´020	ˇ	˘	¯	¨	˜	→	˝	•
´030	¸	´	˚	ß	æ	œ	Æ	Œ
´040	ø	!	”	′	∞	%	&	’
´050	(	)	*	+	,	-	.	/
´060	0	1	2	3	4	5	6	7
´070	8	9	:	;	<	=	>	?
´100	Ø	A	B	C	D	E	F	G
´110	H	I	J	K	L	M	N	O
´120	P	Q	R	S	T	U	V	W
‘130	X	Y	Z	[	“	]	-	—
‘140	‘	a	b	c	d	e	f	g
‘150	h	i	j	k	l	m	n	o
‘160	p	q	r	s	t	u	v	w
‘170	x	y	z	ff	fi	fl	ffi	ffl

Mathematics books and journals **do** not ***look*** as beautiful as ***they*** used to. It is ***not that*** their ma tbema ***tical*** content is ***unsa*** tiefactory, ***ra ther tha*** t the old and ***well*** developed traditions of typesetting bave become too expensive. Fortunately, it now appears tbat matbemstics itself can **be** used to solve ***this*** problem, in spite of the fact tba t tbe ***first*** edition of hop’s ***Œuvres*** is ***naïve*** about tbe ***efficient*** preparation of ***flawless soufflés. This*** is a sample of tbe font wben the resolution is 200 ***dots*** per ***inch*** and 3.6 dots per “point”.

Tile file cms10. mf

“Computer Modern Slanted Roman 10 point”;
$ph = \frac{250}{36}$; $px = \frac{160}{36}$; $pe = \frac{90}{36}$; $pd = \frac{70}{36}$;
$pb = \frac{20}{36}$; $po = \frac{4}{36}$; $ps = \frac{20}{36}$; $pa = .5(ph - pd)$;
$pw = \frac{9}{36}$; $pwi = \frac{27}{36}$; $pwii = \frac{32}{36}$; $pwiii = \frac{36}{36}$;
$pwiv = \frac{32}{36}$; pwv = $\frac{38}{36}$; aspect = 1.0;
$pu = \frac{20}{36}$; *lcs* = 1.075; ucs = 1.7; SC = 0; Is = 0;
slant = .15; sqrttwo = sqrt 2; *fixwidth* = 0;
halfd = 0; varg = 0; *lowast = 0; ligs* = 1 .

input cmbase; call fontbegin;
input roman;
end



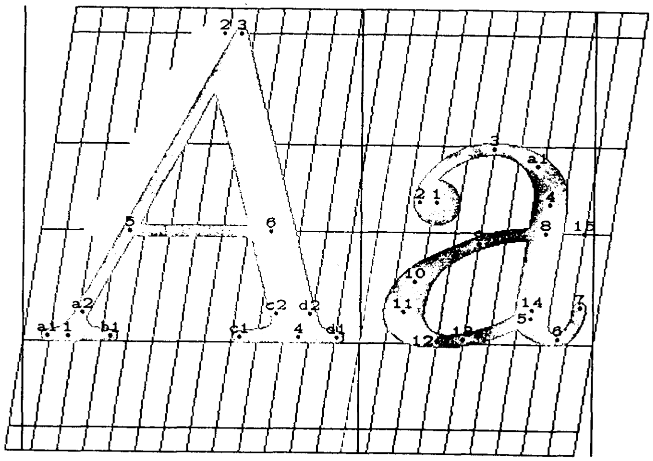
	0	1	2	3	4	5	6	7
‘000	**Γ**	A	**Θ**	A	**Ξ**	**Π**	**Σ**	**Υ**
‘010	**Φ**	**Ψ**	**Ω**	**ı**	*J*	·	**,**	·
‘020	ˇ	˘		··	˜	ˉ	˝	·
‘030	¸	´	ˆ	**ß**	**æ**	**œ**	**Æ**	**Œ**
‘040	**ø**	!	**”**	**’**	**∞**	**%**	**&**	**’**
‘050	**(**	**)**	*****	+	,	.	.	**/**
’060	*0*	**1**	*2*	*3*	*4*	*5*	*6*	*7*
‘070	*8*	*9*	:	**;**	**<**	**=**	>	**?**
‘100	0	*A*	**B**	*c*	**D**	*E*	F	G
‘110	**H**	I	**J**	K	L	**M**	**N**	0
’ 120	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**
. 130	X	Y	**Z**	**[**	**“**	**]**	–	—
‘140	**‘**	a	b	c	d	**e**	**f**	**g**
‘150	**h**	i	**j**	k	**l**	*m*	**n**	**o**
‘160	P	**q**	**r**	**s**	*t*	**u**	v	W
‘170	**x**	**Y**	**z**	**ff**	**fi**	**fl**	**ffi**	**ffl**

Mathematics **books** and journals do not **look** as beautiful **as** they **used** to. It is ***not*** that their mathematical ***content is*** unsatisfactory, rather tbat the old and well developed traditions of typesetting have become too expensive. Fortunately, it now **appears** that mathematics itself **can** be ***used*** to **solve this** problem, in spite of the fact that the **first** edition of ***Æsop's Œuvres is naïve*** about the **efficient** preparation of **flawless soufflés. *This is*** a **sample** of ***the*** font when the **resolution** is ***200*** dots per inch and 3.6 ***dots*** per **“point”.**

The file cms9. mf

“Computer Modern Slanted Roman 9 point”;
$ph = \frac{225}{36}$; $px = \frac{144}{36}$; $pe = \frac{81}{36}$; $pd = \frac{63}{36}$;
$pb = \frac{18}{36}$; $po = \frac{4}{36}$; $ps = \frac{18}{36}$; $pa = .5(ph - pd)$;
pw = $\frac{9}{36}$; pwi = $\frac{25}{36}$; pwii = $\frac{29}{36}$; pwiii = $\frac{33}{36}$,
pwiv = $\frac{29}{36}$; *pwv* = $\frac{33}{36}$; aspect = 1.0;
pu = $\frac{18.5}{36}$; *lcs* = 1.05; ucs = 1.65; SC = 0; Is = 0;
slant = .15; *sqrttwo* = sqrt 2; *fixwidth = 0;*
halfd = 0; varg = 0; *lowast* = 0; *ligs* = 1.

input cmbase; call *fontbegin;*
input roman;
end



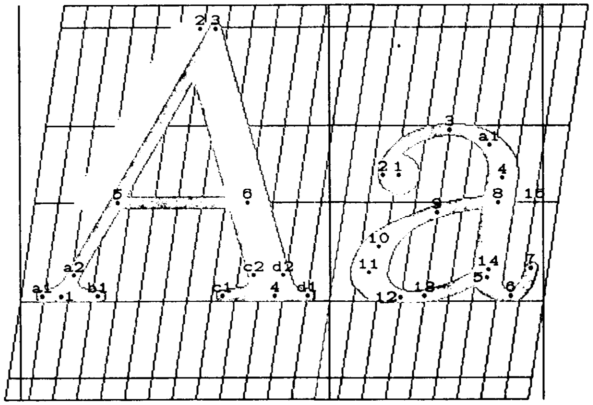
	0	1	2	3	4	5	6	7
´000	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ
´010	Φ	Ψ	Ω	ı	ȷ	`	´	^
´020	ˇ	˘	¯	¨	˜	-	˝	•
´030	¸	´	ˆ	ß	æ	œ	Æ	Œ
´040	ø	!	”	'	∞	%	&	'
´050	(	)	*	+	,	-	.	/
´060	0	1	2	3	4	5	6	7
´070	8	9	:	;	<	=	>	?
´100	Ø	A	B	C	D	E	F	G
´110	H	I	J	K	L	M	N	O
´120	P	Q	R	S	T	U	V	W
´130	X	Y	Z	[	“	]	-	—
´140	‘	a	b	c	d	e	f	g
´150	h	i	j	k	l	m	n	o
´160	p	q	r	s	t	u	v	w
´170	x	y	z	ff	fi	fl	ffi	ffl

Mathematics books and journals do not look as beautiful as they used to. It is not that their mathematical content is unsatisfactory, rather that the old and well developed traditions of typesetting have become too expensive. Fortunately, it now appears that mathematics itself can be used to solve this problem, in spite of the fact that the first edition of Æsop's Œuvres is naïve about the efficient preparation of flawless soufflés. This is a sample of the font when the resolution is 200 dots per inch and 3.6 dots per "point".

The file cms8.mf

"Computer Modern Slanted Roman 8 point";
$ph = \frac{200}{36}$; $px = \frac{128}{36}$; $pe = \frac{72}{36}$; $pd = \frac{56}{36}$;
$pb = \frac{16}{36}$; $po = \frac{3}{36}$; $ps = \frac{16}{36}$; $pa = .5(ph - pd)$;
$pw = \frac{9}{36}$; $pwi = \frac{23}{36}$; $pwii = \frac{27}{36}$; $pwiii = \frac{30}{36}$;
$pwiv = \frac{27}{36}$; $pwv = \frac{30}{36}$; $aspect = 1.0$;
$pu = \frac{17}{36}$; $lcs = 1.03$; $ucs = 1.56$; $sc = 0$; $ls = 0$;
$slant = .15$; $sqrttwo = \mathrm{sqrt}\,2$; $fixwidth = 0$;
$halfd = 0$; $varg = 0$; $lowast = 0$; $ligs = 1$.

input cmbase; call *fontbegin*;
input roman;
end



2 3
5
6
a1
1
b1
a2
c1
c2
d2
4
d1
3
a1
2 1
4
8
15
9
10
11
12
13
14
5
6
7