

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

January 1980

LEVEL

12

ADA 086359

THE COMPUTER MODERN FAMILY OF TYPEFACES

by

Donald E. Knuth

Research sponsored by

National Science Foundation
Office of Naval Research
IBM Corporation
Xerox Corporation

DTIC
COLLECTED
JUL 10 1980

COMPUTER SCIENCE DEPARTMENT
Stanford University

THIS DOCUMENT IS BEST QUALITY PRACTICABLE.
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

This document has been approved
for public release and sale; its
distribution is unlimited.



DDC FILE COPY

80 7 8 023

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER STAN-CS-80-780	2. GOVT ACCESSION NO. AD-A086359	3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) The Computer Modern Family of Typefaces		5. TYPE OF REPORT & PERIOD COVERED technical, January 1980	
7. AUTHOR(s) Donald E. Knuth		6. PERFORMING ORG. REPORT NUMBER STAN-CS-80-780	
		8. CONTRACT OR GRANT NUMBER(s) ✓ N00014-76-C-0330	
9. PERFORMING ORGANIZATION NAME AND ADDRESS ✓ Department of Computer Science Stanford University Stanford, California 94305 USA		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Department of the Navy Arlington, Virginia 22217		12. REPORT DATE January 1980	13. NO. OF PAGES 406
14. MONITORING AGENCY NAME & ADDRESS (if diff. from Controlling Office) ONR Representative - Philip Surra Durand Aeromautics Building, Room 165 Stanford University		15. SECURITY CLASS. (of this report) Unclassified	
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this report) Approved for public release; distribution unlimited.			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from report)			
18. SUPPLEMENTARY NOTES			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) (see other side)			

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19. KEY WORDS (Continued)

20 ABSTRACT (Continued)

✓

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 definitions 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.

↑

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

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

12
11
January 1980
(first printing)

12/207

DTIC
ELECTE
JUL 10 1980
S D C

6
THE COMPUTER MODERN FAMILY OF TYPEFACES
10 Donald E. Knuth

9 Technical rept.

15 N00014-76-C-0330

094120

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.

This document has been approved
for public release and sale; its
distribution is unlimited.

AB

Accession For	
NTIS GRA&I	☒
DDC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail. for special
A	23

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 definitions 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 have 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 Computer 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 be 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 rpens 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

Preface	2
Contents	4
Introduction	6
The file cmbase	14
Computer Modern Roman definitions	25
The file roman	25
The file romitu	27
The file roman1	69
The file romand	93
The file romans	105
The file romitp	109
The file romita	119
The file romlig	129
The file romitl	133
The file romits	137
Computer Modern Italic definitions	147
The file italic	147
The file itall	149
The file itald	165
The file greek1	175
The file italms	189
The file italig	193
The file itals	197
Computer Modern Symbol definitions	201
The file symbol	201
The file script	251
Computer Modern Math Extension definitions	271
The file mathex	271
The file mathd1	271
The file mathop	295
The file mexext	313

Computer Modern Roman fonts	325
The file cmr10	325
The file cmr9	327
The file cmr8	329
The file cmr7	331
The file cmr6	333
The file cmr5	335
The file cmb10	337
The file cmb9	339
The file cmb8	341
The file cms10	343
The file cms9	345
The file cms8	347
The file cms7	349
The file cmtt	351
The file cmtt9	353
The file cmtt8	355
The file cmss8	357
The file cmsss8	359
The file cmssb	361
The file cmss12	363
The file cmtitl	365
The file cmc10	367
The file cmdunh	369
The file cmfib	371
The file cmff	373
Computer Modern Italic fonts	375
The file cmti10	375
The file cmti9	377
The file cmu10	379
The file cmi10	381
The file cmi9	383
The file cmi8	385
The file cmi7	387
The file cmi6	389
The file cmi5	391
Computer Modern Symbol fonts	393
The file cmsy10	393
The file cmsy9	395
The file cmsy8	397
The file cmsy7	399
The file cmsy6	401
The file cmsy5	403
Computer Modern Math Extension font	405
The file cmathx	405
Index to font codes	406

In this report we shall study the Computer Modern fonts by working our way in from the outermost level, then going back out again. File `cmr10.mf` looks like this:

```
"Computer Modern Roman 10 point";
ph = 238; px = 190; pe = 30; pd = 30;
pb = 38; po = 38; ps = 38; pa = .5(ph - pd);
pw = 2; pwi = 2; pwii = 2; pwiii = 38;
pwiv = 38; pwv = 38; aspect = 1.0;
pu = 20; lcs = 1.075; ucs = 1.7; sc = 0; ls = 0;
slant = 0; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 0; varg = 0; lowast = 0; ligs = 1.
input embase; 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 at which serif bracketing is tangent to 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 are seven parameters affecting the pen sizes.

- pw* is the hairline width, used in the thinnest parts of letters
- pwi* is the stem width, used for the vertical strokes in an "i".
- 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, used in an "O" at its widest point.
- aspect* is the ratio of a hairline pen's height to its width.

INTRODUCTION

A complete font design is a complex system, so there are 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 METAFONT is applied to that file it will produce the "Computer Modern Roman 10 point" font. Another file `"cmss8.mf"` produces "Computer Modern Slanted Sans Serif 8 point," and so on. But if we actually look up the values of certain parameters and input the file `"roman.mf"`, which specifies the actual METAFONT programs for individual letters. Therefore 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 `embase.mf`.

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

Next come five parameters concerning horizontal dimensions:

pu is the unit width, 1/18 of an em.
les is the amount by which serifs of lower-case letters project from the stems, in units of *pu*.
ucs is the amount 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 multiples 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 letter 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*, used to slant letters either forwards or backwards.
sqrtwo is used to control the ellipticity of the bowls of letters, as explained in Chapter 8 of the METAFONT manual.
halfd is nonzero if certain characters like "i" are to descend only half as far 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 lowered so that it is centered on the axis.
lgs 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 *cmst10.mf* ("Computer Modern Slanted 10 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 *cmi10.mf* ("Computer Modern Bold 10 point") are nearly identical to those of *cmr10.mf*, except that the pens are bigger:

```
pw = 13; pwi = 10; pwj = 10; pwk = 10;
pwl = 10; pwm = 10; pwn = 10;
pwo = 10; pwq = 10; pwr = 10;
pws = 10; pwt = 10; pwu = 10;
pww = 10; pwx = 10; pwy = 10;
pww = 10; pwx = 10; pwy = 10;
```

furthermore serifs are shorter (*les* = .85, *ucs* = 1.5).

File *cmr5.mf* generates 5-point type, but it is not simply obtained by halving the parameters of *cmr10*. The eight vertical dimensions *ph*, *px*, ..., *ps* are exactly half as large as before, but the pen sizes and the horizontal dimensions get smaller at different rates ...

```
pwi = 10; pwi = 10; pwi = 10; pwi = 10;
pwi = 10; pwi = 10; pwi = 10; pwi = 10;
pwi = 10; pwi = 10; pwi = 10; pwi = 10;
pwi = 10; pwi = 10; pwi = 10; pwi = 10;
```

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" (*cmstb*). It uses the same vertical dimensions and miscellaneous

settings as *cmr10*, and gets its other characteristics from the following parameter values:

```
pwi = pwi - pwj - pwn = 10;
pwi = pwi - pwj - pwn = 10;
pwi = pwi - pwj - pwn = 10;
pwi = pwi - pwj - pwn = 10;
```

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

```
ph = 10; px = 10; pe = 10; pd = 10;
ps = 10; po = 10; ps = 10; pa = 10;
pw = pwi - pwi - pwi = 10;
pwi = 10; pwi = 10; pwi = 10;
pwi = 10; pwi = 10; pwi = 10;
pwi = 10; pwi = 10; pwi = 10;
pwi = 10; pwi = 10; pwi = 10;
pwi = 10; pwi = 10; pwi = 10;
```

By making stronger settings of the parameters you can also get typewriter fonts, but the font variations are not as pronounced as those for the possible combinations of parameters in *cmr10*, but they do try to handle all reasonable variations.

The programs for Computer Modern can be used in several ways. The general procedure is to run METAFONT and type

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

the routines will act differently depending on the specified mode. At present mode 0 generates proof sheets and shows the letters as they are being drawn, with a resolution of 16 pixels per point; mode 1 generates a font for the X(11) or Versaler or Varian, etc., with a resolution of 3.6 pixels per point; mode 2 generates a font for the CTR with a resolution of 7.3 pixels per point; displaying the titles of the letters as they are being drawn; mode 3 is like mode 1 but for a Doyler, and other modes cause a file mode *mf* to be read in, where this file specifies METAFONT's mode of operation. 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 = 1 input (font name)", etc.

All of the illustrations of individual characters in this report were generated with 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;
ph = 10; ... (set up for cmr10) ...; call fontbegin;
input test;
```

```

new pw, ... (set up for cmh10) ...; call fontbegin.
input test;
new pw, ... (set up for cmab) ...; call fontbegin.
input test;
new ph, ... (set up for cm1) ...; call fontbegin.
input test;
new ph, ... (set up for cmab8) ...; call fontbegin.
input test;
end.

```

Thus, it runs your test file against five 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 cm1). Then comes the italic group, which is somewhat similar, but it either makes text fonts with ligatures (like cm1i10) or italic fonts for mathematical formulas (like cm10). The third group is called symbol, and it makes math symbols (like cm10). Finally there is the mathex group, for extended math symbol fonts (like cm10x).

Let's go one level deeper and take a look at the programs for individual letters. 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 , pz , etc., have corresponding raster-oriented variables, satisfying the approximate relation

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

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

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

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 to stand for the x-height, since a line at 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 $y\text{-coordinate} = 0$. The top pixel of an "l" is in row h , which is always an integer. (Note that there are actually $h+1$ occupied rows, not h , although h is called the h-height.) The top pixel of an "n" is in row m , and the bottom pixels of the descender letters (g, j, p, y) appear in row $-d$. All three of these variables (h, m, d) are integers, and so is the overshoot variable e (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 open height, since the bar of an "e" is drawn with an open 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

integer 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 are generally expressed in terms of the following variables, each of which has a positive integer value intended to approximate the "true" infinite-resolution value (and slightly increased in order to look right on the output device, depending on the current mode):

```

w0, the hairline width;
w1, the stem width;
w2, the curve width;
w3, the dot diameter;
w4, the upper-case stem width;
w5, the upper-case curve width;
w6, the hairline height;
w7, the stem height;
w8, the curve height;
w9, the upper-case stem height;
w10, the rule thickness for math symbols.

```

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

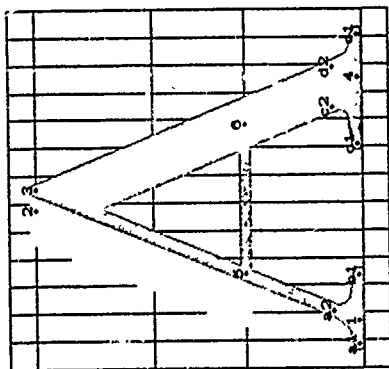
$$w_4/w_0 \approx w_7/w_1, w_5/w_2 \approx w_8/w_3, w_9/w_4 \approx w_{10}/w_5, w_{10} \approx 25w_0, w_2.$$

The height, print, and repeat are w_0 , and the y -pen is w_0 . Thus, an open of size w_0 is equivalent to a open of size w_1 ; we may call it the "hairline 1-p" for the font.

In the horizontal dimension, the Computer Modern programs make frequent use of a variable u , the approximate unit width when there are 18 units to an em. The width of a character is expressed in terms of units (e.g., an "l" is 10u wide, unless there is a serif correction $sc \neq 0$ or some additional letter spacing $l \neq 0$), and key positions can be specified as a certain number of units from the left (e.g., the stems of an "l" are centered at $2.5u$ and $7.5u$). The vertical guidelines in the illustrations of this report are one unit apart.

If the character is l units wide, variable u has been calculated so that l times u is an integer, the rightmost column of the character. (The value of u itself is usually not an integer, nor need l 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 white 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 (l in pixels), the 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 " $-2.5u$ " as a point $2\frac{1}{2}$ units from the right edge, etc.

Variable oo is an integer approximation to one-half of u , it is used for "half of an overshoot," when curved lines approach the bottom or the top of a character. Variable do is like d except it is only half as large, when $half$ is nonzero. A few other variables like this are defined for use in several different characters, either in file encase 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 more 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 give eight equations to define the locations of points 1, 2, 3, and 4. First point 1 is positioned so that, using an $hpen$ of size u_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_0 . (The upper-case curve width w_0 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_2 , and another from 3 to 1 with an $hpen$ of width u_0 . First we define y_1 and y_2 so that the top occurs at the h -height h , plus the "overshoot" o that gives this letter a touch of class. Then we state that $x_3 - x_1 = x_2 - x_1$, so that the two diagonal strokes will have the same slopes (the same amount of change in the x direction). Finally we stipulate that $rt_2x_2 = rt_0x_3$, 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 information 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). Before erasing anything,

```

%The letter "A":
call charbegin("A, 13, 2sc, 2sc, ph, 0, 0);
hpen;
if_0x1 = round 1.5u; b_0y1 = 0;
rt_2x1 = round(r - 1.5u); bot_2y1 = 0;
top_0y1 = top_2y1 = h + o;
x_3 - x_1 = x_2 - x_1; rt_2x2 = rt_0x3;
u_0 draw 2...4;
y_1 = h_0 = c;
new aa, bb;
x_2 - 1 = aa[x_1, y_1]; y_2 = aa[y_1, y_1];
x_0 + 1 = bb[x_1, x_2]; u_0 = bb[y_1, y_2];
u_0 draw 5...6;
hpen#; u_0 draw 3...5;
if ucs # 0.
  call a serif(1, 0, 3, -5ucs);
  call b serif(1, 0, 3, +ucs);
  call c serif(4, 5, 2, -ucs);
  call d serif(4, 5, 2, +5ucs);
fi
% right diagonal stroke
% auxiliary variables for intersection of lines
% bar line
% crase excess at upper left
% left diagonal stroke
% left serifs
% right serifs

```

A METAFONT program for upper-case "A"

however, we may as well draw the bar line. Computer Modem fonts place this line at the e -height, the same level as the bar line in an "e", hence $y_2 = u_0 = e$. The calculation of x_2 and x_0 is slightly trickier, x_2 lies between x_1 and x_0 , and the ratio of its distance is the same as the ratio for y_2 with respect to y_1 and y_0 . The equations " $x_2 = aa[x_1, x_0]$, $y_2 = aa[y_1, y_0]$ " would almost surely work to define a suitable point, but the program actually uses $x_2 = 1$ instead of x_2 , 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 `hpen#` is used to 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 necessary 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 to attach fancy serifs at points 1 and 4, these serifs extend 5ucs 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 "W", as well as for the Greek letter "A", and you will be well on your way to having "M" and "W". Similarly, the code below for "i" leads to "j" and "p" with little further ado.

We shall now plunge into the deepest level, the subroutines in embage `mf` that take 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 = 0.00314159;
if mode = 0: proofmode; drawdisplay; titlctrace;
  pixels = 18; blacker = 0;
else: if mode = 1: fntmode; txfmode; no modtrace;
  pixels = 3.6; blacker = 1.2;
  else: if mode = 2: crsmode; txfmode; titlctrace; no modtrace;
  pixels = 73.7973; blacker = 1;
  else: if mode = 3: fntmode; txfmode; no modtrace;
  pixels = 3.6 * (1.1/1.3) * (1/3); blacker = 1.2;
  else: input mode;
  fi;
fi;
fi;

```

```

subroutine fontbegin:
no eqtrace;
new typesize;
new h, d, dd, m, e, o, co, b, s, a;
new w, w1, w2, w3, w4, w5, w6, w7, w8, w9, w10, w11;
new delaw, bold;
new armic, lcic;
new less;
new prt;
w0 = round(pixels*pw + blacker);
w1 = round(pixels*pw + blacker);
w2 = round(pixels*pw + blacker);
w3 = round(pixels*pw + blacker);
w4 = round(pixels*pw + blacker);
w5 = round(pixels*pw + blacker);
w6 = round(pixels*pw + blacker);
w7 = round(pixels*pw + blacker);
w8 = round(pixels*pw + blacker);
w9 = round(pixels*pw + blacker);
prt = .25*pw, pw11;
w10 = round(pixels*prt + blacker);
w11 = round(pixels*(2*pw, pw11)*aspect + blacker);
delaw = pixels*(pw11 - pw);
bold = .5[pw11, pw11]*pixels + blacker;

```

```

% The following corrections are for l-w resolution:
if w0/w1 > { (pw11/pw1): new w2, w1, w2 = w3 = w1;
fi;
if w2/w1 > { (pwv/pwiv): new w3; w3 = w1;
fi;
if w3/w1 > { (pw11/pw1): new w4; w4 = w1;
fi;
hpenht w1; vpenwd w1; lpenht w1; rpenht w1;
typesize = ph + pd + 2pb; cf-typesize = pixels*typesize - 1;
h = round cf-ph; d = round cf-pd;
new pdd; pdd = (1 - .5*half)pd; did = round cf-pdd;
m = round cf-px;
o = round cf-po; oo = round .5cf-po; s = cf-ps,
a = 5*round 2cf-pa;
b = - round( 5(h + d - typesize*pixels));
lpen; e = good cf-pe;
maxht h + b + 2;
trxy slant;
if ucs < 0: armic = ph slant + (sc - 1)pu,
else: armic = ph slant + (sc - 5)pu,
fi;
if pw11 > 1.5pu: lcic = -.25pu;
else: lcic = .5pw11 - pu;
fi;
if pw < pw11: less = lcs;
else: less = .5lcs;
fi.

```

```

subroutine charbegin(var charno)
  (var charuw)
  (var lfcorr, var rfcorr)
  (var charh, var charw, var chari)
  no eqtrace; no calltrace; no drawdisplay;
  new uw, moduw;
  new u;
  new tu;
  new lfcorr;
  new lcorr, rcorr;
  if chari >= 0: italcorr = chari; else: italcorr = 0;
  fi;
  if danger ≠ 0:
    lcorr = danger-round((lfcorr - ls)/danger);
    rcorr = danger-round((rfcorr - ls)/danger);
  else: lcorr = lfcorr - ls; rcorr = rfcorr - 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 charh > 0: chardp charh;
  else: chardp 0;
  fi;
  charwd moduw-pu; charwd moduw-tu;
  incx round(-lcorr-u);
  if mode = 0: call box(-round lcorr u);
  fi.

```

```

subroutine box(var offset);
  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; open, 1 draw 1..2;
  y3 = y4 = e; draw 3..4;
  y5 = y6 = m; draw 5..6;
  y7 = y8 = k; draw 7..8;
  y9 = y10 = topp; draw 9..10;
  y11 = y12 = -d; draw 11..12;
  y13 = y14 = bott; draw 13..14;
  trxy 0;
  y15 = y16 = topp; y17 = y18 = bott;
  draw 15..17, draw 16..18;
  if italcorr > 0: x19 = x20 = right + italcorr pixels,
    y19 = topp; y20 = 0; draw 19..20;
  fi;
  trxy slant;
  pos = 0; call unitlines.

  % Draw guidelines and box around a character:
  % baseline
  % e-height
  % h-height
  % mean line (x-height)
  % top of character
  % descender line
  % bottom of character
  % temporarily turn off the slant
  % left and right edges
  % show italic correction
  % restore slanted transformation
  % draw the unit guidelines

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

```

```

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

subroutine serif(index i)
(index k)
(index j)
(var al):
  y1 = yi;
  if y1 < yj: y2 = y1 + s;
  else: y2 = y1 - s;
  fi;
  hpen;
  if al < 0: lft0x1 = lft0x1 + sl*u - epc;
  lft0x2 = lft0x1(y2 - y1)/(y1 - y1)[x1, xj];
  else: rft0x1 = rft0x1 + sl*u + epc;
  rft0x2 = rft0x1(y2 - y1)/(y1 - y1)[x1, xj];
  fi;
  no proofmode;
  x3 = 1/2[x1 - sl*u, 1/2[x1, xj]];
  y3 = 1/2[y1, 1/2[y1, y2]];
  minvr 0; minvs 0;
  u0 ddraw 1{x1 - x1, 0}..3..2{xj - x1, yj - y1}, 1..1..i;
  minvr 0.5; minvs 0.5;

subroutine darc(index i)
(index j)
(var maxwidth):
  x3 = x1; x2 = x4 = 1/sqrt(2)[x1, xj]; x3 = xj;
  y3 = yj; y2 = 1/2[y1, yj];
  y1 = 1/sqrt(2)[y1, yj]; y1 = 1/sqrt(2)[y1, yj];
  hpen; draw lft0{x3 - x1, 0}..1/2[3]{u0, maxwidth}2{x1 - x1, y1 - y1}..
  1/2[3]{u0, maxwidth}3{0, y2 - y1}..
  1/2[3]{u0, maxwidth}4{x3 - x1, y3 - y1}..lft0{x3 - x2, 0}.

subroutine arc(index i)
(index j)
(var maxwidth):
  x1 = 1/sqrt(2)[x1, xj]; y1 = 1/sqrt(2)[y1, yj];
  hpen; draw lft0{x2 - x1, 0}..1/2[3]{u0, maxwidth}1{xj - x1, yj - y1}..
  1/2[3]{u0, maxwidth}2{0, y2 - y1}.

```

```

subroutine arm(index i)
(index j)
(index k):
  if ucs < 0:
    if u0 = w1: x1 = x2 = x1; y1 = yj;
    if y1 < yj: y2 = yj - aspect*ucs*u;
    else: y2 = yj + aspect*ucs*u;
    fi;
    u0 draw i..1; draw i..2;
    else: minvr 0; minvs 0;
    x1 = xj - (x1 - xj); y1 = 1/2[yj, y1];
    x2 = x1; y2 = 2[yj, y1];
    u0 ddraw i..j..k, i{xj - x1, 0}..1..k(..2);
    minvr 0.5; minvs 0.5;
    fi;
    else: x1 = x1; y1 = yj; u0 draw i..1;
    fi.

subroutine scomp(index i)
(index p)
(index j)
(index k)
(var slope):
  % This subroutine computes y1, x1, y2 so that y1 - x1 = slope(x1 - x1)
  % and so that the following curve is consistent with an ellipse:
  % i{x1 - x1, 0}..p{0, y1 - y1}..j{x1 - x1, slope(x1 - x1)}.
  y1 - yj = slope(x1 - x1);
  new aa, bb; aa = slope(x1 - x1); bb = y1 - yj - slope(x1 - x1);
  xj - x1 = -2aa*bb/(x1 - x1)/(aa*aa + bb*bb);
  yj - y1 = .5(bb*bb - aa*aa)/bb.

```

```

subroutine sdraw(index i)
  (index p)
  (index k)
  (index q)
  (index j)
  (var penwd)
  (var penht)
  (var slope)

  new w18, w19; w18 = penht; w19 = penwd;
  open; top0y = top0y; bot0y = bot0y;
  if x2 < x1; rt0x = rt0x; lt0x = lt0x;
  else; lt0x = lt0x; rt0x = rt0x;
  fi;
  y1 = y0; y2 = y1;
  call scomp(i, 1, 3, 5, slope);
  call scomp(i, 2, 4, 6, slope);
  call scomp(j, 9, 7, 5, slope);
  call scomp(j, 10, 8, 6, slope);
  hpen; w1 draw i {x1 - x0, 0} .. 1 {0, y1 - y0} .. 3 {x2 - x1, slope(x2 - x1)} ..
    7 {x2 - x1, slope(x2 - x1)} ..
    9 {0, y1 - y0} .. 2 {0, y1 - y0} .. 4 {x2 - x1, slope(x2 - x1)} ..
    8 {x2 - x1, slope(x2 - x1)} ..
    10 {0, y1 - y0} .. j {x2 - x1, 0}.
  % the s-curve

subroutine zcomp(index i)
  (index p)
  (index j)
  (index k)
  (var slope)

  % This subroutine is dual to scomp.
  % It computes x0, x1, y1 so that x2 - x1 = slope(y2 - y1)
  % and so that the following curve is consistent with an ellipse:
  % i {0, y0 - y1} .. j {x0 - x1, 0} .. k {slope(y2 - y1), y2 - y1}
  x1 - x0 = slope(y2 - y1);
  new aa, bb; aa = slope(y2 - y1); bb = x2 - x1 - slope(y2 - y1);
  y1 - y0 = -2aa*bb/(y2 - y1)/(aa*aa + bb*bb);
  x0 - x1 = 5(bb*bb - aa*aa)/bb.

```

```

subroutine zdraw(index i)
  (index p)
  (index k)
  (index q)
  (index j)
  (var penht)
  (var penwd)
  (var slope)

  % This subroutine is dual to sdraw.
  new w18, w19; w18 = penwd; w19 = penht;
  open; rt0x = rt0x; lt0x = lt0x; y1 = y0; y2 = y1;
  if y0 > y1; top0y = top0y; bot0y = bot0y;
  top0y = top0y; bot0y = bot0y;
  else; bot0y = bot0y; top0y = top0y;
  bot0y = bot0y; top0y = top0y;
  fi;
  x2 = x0; x3 = x1;
  call zcomp(i, 1, 3, 5, slope);
  call zcomp(i, 2, 4, 6, slope);
  call zcomp(j, 9, 7, 5, slope);
  call zcomp(j, 10, 8, 6, slope);
  hpen; w1 draw i {0, y1 - y0} .. 1 {x1 - x0, 0} .. 3 {slope(y2 - y1), y2 - y1} ..
    7 {slope(y2 - y1), y2 - y1} ..
    9 {x1 - x0, 0} .. 2 {0, y1 - y0} .. 4 {slope(y2 - y1), y2 - y1} ..
    8 {slope(y2 - y1), y2 - y1} ..
    10 {x1 - x0, 0} .. j {0, y1 - y0}.
  % the s-curve

subroutine bar(index i, index j)
  % This subroutine is similar to "vpen; w1 draw i..j",
  % but the vpen slants with italic.
  no proofmode,
  vpen, top0y = top0y; bot0y = bot0y;
  top0y = top0y; bot0y = bot0y;
  x1 = x2 = x0; x2 = x1 = x1;
  w1 draw 1..3, 2..4.

subroutine lstroke(index i)
  (index j)
  (index k)
  (var penwd)
  (var penht)
  (var slope)

  hpen; x1 = x0; bot0y = 0; y1 = 5{m, h};
  rt0x = rt0x; y1 = y1;
  x2 = 5{x1, x1}; top0y = h + oo;
  draw w1; w1 {0, 1} .. w2 {2, 1, 0} .. 3 {0, -1};
  open; w2 draw i;
  if lcs < 0; call "a serif(j, 1, 1, -lcs);
  call "b serif(j, 1, 1, lcs);
  fi.

```

```

subroutine hstroke(index i)
  (index j)
  (index k):
  % x-coordinate of left stem
  % x-coordinate of right stem
  % will be set to base of right stem
  hpen;  $x_i = x_j$ ; botl $y_i = 0$ ;
  rll $y_i = r_{l,i}$ ;  $y_i = \{e, m\}$ ;
   $x_2 = .5(x_i, x_j)$ ; topl $y_i = m + oo$ ;
   $x_3 = 1/\sqrt{2}(x_i, x_j)$ ;  $y_i = 1/\sqrt{2}(y_i, y_j)$ ;
  draw [wl]{0, 1} .. [wl]{2{1, 0} .. [6{wl, wl]}{3{xj - xi, yj - yi}} ..
  [wl]{j}{0, -1} .. k.
  % shoulder and stem

subroutine cdraw(index i, index j)
  (index p, index q):
  % given points
  % given widths,  $w_p \geq w_q$ 
  % An implementation of the forbidden "open; draw [wp]{i} .. [wq]{j}"
  open; wp draw i;
  new aa; (aa + cps) sqrt((xj - xi)(xj - xi) + (yj - yi)(yj - yi)) = wp - wq;
  % plot the bigger dot
   $x_2 - x_1 = aa(y_i - y_j)$ ;  $y_2 - y_1 = aa(x_j - x_i)$ ;
  % perpendicular points
   $x_1 = .5(x_i, x_j)$ ;  $y_1 = .5(y_i, y_j)$ ;
  wq ddraw 1..j, 2..j.
  % fill in the rest

subroutine qcire(index i)
  (index j)
  (index k)
  (var size):
  % size of open that draws a quarter circle
  open;  $x_j = 1/\sqrt{2}(x_i, x_j)$ ;  $y_j = 1/\sqrt{2}(y_i, y_j)$ ;
  size draw i(xi - xj, 0) .. j(xi - xj, yj - yi) .. k(0, yk - yj).

subroutine hcire(index viii, index i, index ii, index iii, index iv, var size):
   $x_{iv} = x_{viii}$ ;  $y_{iv} = .5(y_{viii}, y_{viii})$ ;
  call qcire(viii, i, ii, size); call qcire(iv, iii, ii, size).

subroutine circle(index i, index ii, index iii, index iv,
  index v, index vi, index vii, index viii, var size):
   $x_{iv} = x_{viii} = .5(x_{viii}, x_{viii})$ ;  $y_{iv} = y_{viii} = .5(y_{viii}, y_{viii})$ ;
  call qcire(viii, i, ii, size); call qcire(iv, iii, ii, size);
  call qcire(iv, v, vi, size); call qcire(viii, vii, vi, size).

subroutine entry(var z)
  (index j):
  % x-coordinate for downward stroke
  % This subroutine draws a little hook at the beginning left of an italic character,
  % ending with the pen traveling vertically at point j with size wl.
  hpen;  $x_1 = good_0$ ;  $y_1 = \{m\}$ ;  $y_2 = x_j - 1.5u$ ; topl $y_i = m + oo$ ;
  draw [wl]{1{(xj - 2.5u) - x1, m} .. [wl]{2{1, 0} .. [wl]{j}{0, -1}}.

```

```

subroutine skewentry(var z)
  (index j):
  % x-coordinate for upward stroke
  % 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;  $x_1 = good_0$ ;  $y_1 = \{m\}$ ;  $y_2 = \{m\}$ ;
   $x_2 = x_j - 1.25u$ ; topl $y_i = m + oo$ ;
  draw [wl]{1{0, 1} .. [wl]{2{1, 0} .. [wl]{j}{-u, -m}}.
  % x-coordinate for downward stroke (yl will be set)
  subroutine exit(index i)
    (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 wl.
    hpen;  $x_2 = good_0$ ;  $y_2 = \{m\}$ ;  $y_1 = \{m\}$ ;  $x_1 = x_i + 1.5u$ ; botl $y_i = -oo$ ;
    draw [wl]{i}{0, -1} .. [wl]{1{1, 0} .. 2{x2 - (xi + 2.5u), m}}.
    % x-coordinate for downward stroke (yl will be set)
    subroutine skewexit(index i)
      (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 vertically
      hpen;  $x_2 = good_0$ ;  $y_2 = \{m\}$ ;  $y_1 = \{m\}$ ;  $x_1 = x_i + 1.25u$ ; botl $y_i = -oo$ ;
      draw [wl]{i}{-u, -m} .. [wl]{1{1, 0} .. 2{0, 1}}.
      % starting point
      subroutine italstroke(index i)
        (index j):
        % x-coordinate of right stem (yl will be set)
        hpen;  $x_1 = 6(x_i, x_j)$ ;  $x_2 = x_j - .4u$ ; topl $y_i = m + oo$ ;  $y_2 = 75(x_i, y_i)$ ;
         $y_j = .3(e, m)$ ;
        draw [wl]{i}{0, 1} .. [wl]{1{1, 0} .. 75{wl, wl]}{2 .. [wl]{j}{0, -1}}.
        % makes the bar of p, t, u, variant omega
        subroutine pstroke:
          vpen;  $x_1 = good_0(0)$ ;  $y_1 = m - m/3 14159$ ;
           $x_2 = 2u$ ; topl $y_i = m$ ;  $y_1 = y_2$ ;  $x_1 = r - 1.5u$ ;
          draw [wl]{1{x2 - x1, 3 14159(y2 - y1)}} [wl]{2{1, 0} .. 3;
          open; wl draw 3.
          % make the end point round
          subroutine endv(index i)
            open;  $x_1 = x_i - u$ ;  $x_2 = x_i - 6u$ ; topl $y_i = m + oo$ ;  $y_2 = y_1$ ;
            hpen; draw [wl]{i}{0, 1} .. [wl]{1{1, 2}};
            open; wl draw 1
            % draws final bulb starting at this point
            % stroke
            % bulb
            subroutine max(var a, var b)
              new acc;
              if a > b: acc = a;
              else: acc = b;
              fi.
              % sets acc = max(a, b)

```

ROMAN CHARACTER DESIGNS

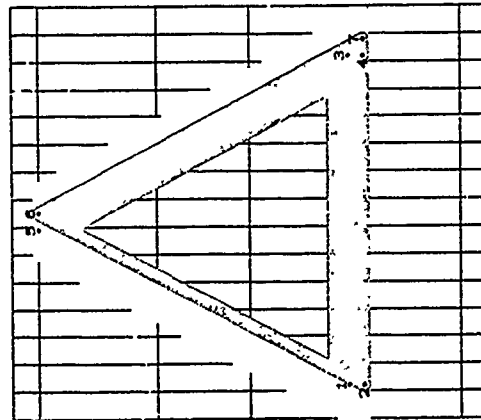
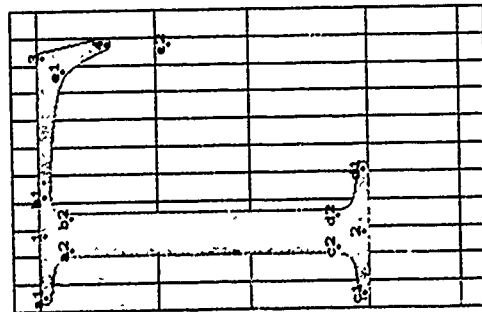
The file roman mf

[illegible]

24

25





The file romitu mf

% These upper-case Roman and Greek alphabets were prepared by D. E. Knuth in November, 1979, inspired by the Monotype alphabets used in *The Art of Computer Programming*.
 % For text spacing, set $mi = 0$; for math spacing, set $mi = 1$.
 % Character codes '000'-'012' and '101'-'132' are generated.

new mc, bowl, rbowl, rstem, rv, hic;
 mc = mi/pt;
 bowl = .3ph-slant + .5pu;
 rbowl = 7ph-slant - .5pu;
 rstem = ph-slant + (ucs + 2sc - 1.5)pu;
 rv = ph-slant + (.5ucs + 2sc - 1)pu;
 hic = 1 - .5mi;
 % quantities used to compute spacing
 % converts to relative units when $mi = 1$
 % used at left of upper-case bowl
 % used at right of upper-case bowl
 % used at right of tall stem
 % used at right of tall diagonal
 % used when half the italic correction goes into *it* or *err*

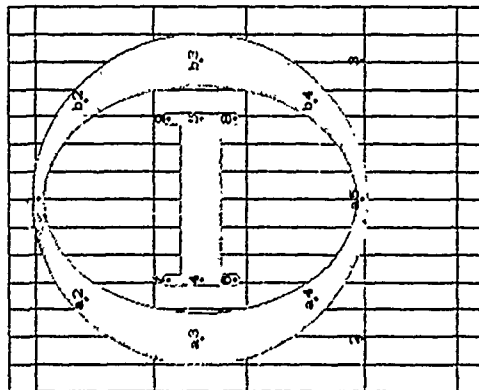
"Upper case Greek Gamma";
 call charbegin('000, 11, 2sc, sc - mc(armic - 2.5pu), ph, 0, mi(armic, 2.5pu));
 hpen; ift, x1 = round 2u; x2 = x1; top, y1 = h; bot, y2 = 0;
 w1 draw 1..2;
 if ucs $\neq$ 0;

call 'a serif(1, 4, 2, -ucs);
 call 'b serif(1, 4, 2, .5ucs);
 call 'c serif(2, 4, 1, -ucs);
 call 'd serif(2, 4, 1, ucs);
 % stem
 % upper stem serif
 % lower stem serif

fi;
 new ss; ss = 1.4aspect-ucs+u+eps;
 if ss + w1 > .25h; new ss; ss = .25h - w1 + eps;
 fi;
 r0x2 = round(r - 1.5u); x1 = x1 + 5u; y1 = y1;
 r0y2 = round(r - 1.5u); x1 = x1 + 5u; y1 = y1 - ss;
 call 'e arm(1, 3, 4);
 % upper arm and serif

"Upper case Greek Delta";
 call charbegin('001, 15, 0, 0, ph, 0, 0);
 hpen; ift, x2 = round u; r1, x1 = round(r - u); bot, y2 = 0; bot, y1 = 0;
 top, y5 = h + o; y4 = y5;
 vpen; bot, y1 = 0; y3 = y1;
 w1 draw 1..3;
 rpen; w1 draw 5..4;
 hpen; w1 draw 5..4;
 lpen; w1 draw 6..2;
 hpen; w1 draw 6..2;
 y1 = y1; r0x2 = r1(x1);
 w1 draw 2..7..7..6.

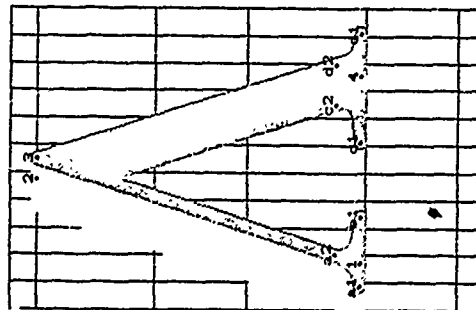
% bar line
 % erase excess at lower right
 % right diagonal
 % erase excess at left
 % left diagonal
 % sharpen lower right corner



```

"Upper case Greek Theta";
call charbegin('002, 14, mc lbowl, -mc rhowl, ph, pdd, m[rbowl, 0]);
hpen;
if fixwidth ≠ 0: new save; save = sqrttwo;
  new sqrttwo; sqrttwo = sqrt save;
  lf: x2 = round 1.5u;
  else lf: x2 = round u;
fi;
x1 = r - x1;
topdy = h + oo;
botdy = -oo; y1 = y1; x1 = r - x1;
call "a dare(1, 2, w5);
call "b dare(1, 3, w5);
lf: x1 = round(r; x2 + u); x3 = r - x1; y1 = y1 = 5[y1, y1];
vpen;
if ucs = 0: w1 draw 4..5;
else: call bar(4, 5);
  x4 = x1 = x1; x4 = x1 = x1; y1 = y1 = 6h; y1 = y1 = 4h;
  w1 draw 6..7;
  draw 8..9;
fi;
if fixwidth ≠ 0: new sqrttwo; sqrttwo = save;
fi.

```



```

"Upper case Greek Lambda";
call charbegin('003, 11, 2sc, 2sc, ph, 0, 0);
hpen; lf: x1 = round 1.5u; botdy = 0;
rt: x1 = round(r - 1.5u); botdy = 0;
topdy = topdy = h + oo;
x3 = x1 = x1 = x1; rt: x2 = rt: x1;
w5 draw 2..4;
lpen; w1 draw 3..1;
hpen; w1 draw 3..1;
if ucs ≠ 0:
  call "a serif(1, 0, 3, -5ucs);
  call "b serif(1, 0, 3, +ucs);
  call "c serif(3, 5, 2, -ucs);
  call "d serif(4, 5, 2, +5ucs);
fi.

```

```

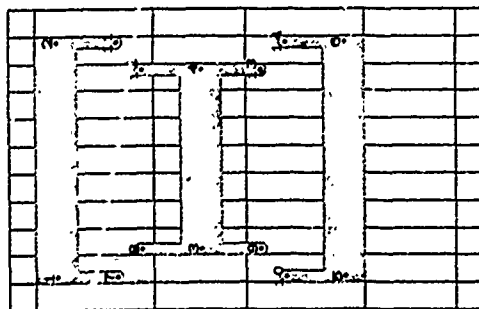
% right diagonal stroke
% erase excess at upper left
% left diagonal stroke

```

```

% left serifs
% right serifs

```



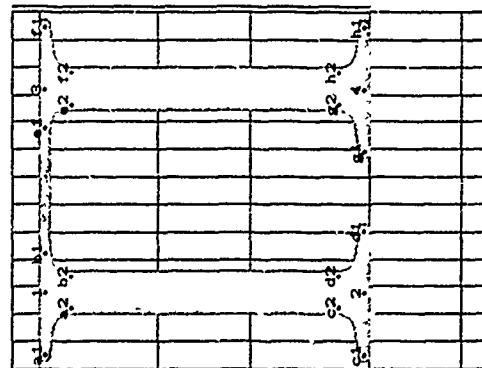
```

"Upper case Greek Xi";
call charbegin("004,11,0,-5mc(ph-slant-5pu),ph,0,hic(ph-slant-5pu));
vpen; top4y1 = h, y1 = y2; y1 = y2 = good8 5h; bot3y1 = 0, y2 = y3;
lt4x1 = round u; r4x2 = round(r-u); x3 = x1 x4 = x2;
if ucs = 0: w1 draw 1..2;
draw 3..4;
draw 5..6;
else call bar(1,2);
call bar(3,4);
call bar(5,6);
new ss; ss = urs-aspect 1 + eps;
if ss > 2h new ss; ss = .2h + eps;
fi;
x7 = x10 = x1; x8 = x1 = x1, x11 = x1 = x2; x12 = x1 = x1;
bot4y1 = bot3y1 - ss; top4y1 = top3y1 + ss;
bot4y1 = bot3y1 - ss; top4y1 = top3y1 + ss;
y1 = y2; y12 = y3; y1 = y2, y1 = y3;
w1 draw 7..1; draw 11..2;
draw 8..9, draw 12..13;
draw 10..5, draw 14..6;
fi.

```

% upper bar
 % middle bar
 % lower bar
 % upper bar
 % middle bar
 % lower bar

% upper serifs
 % middle serifs
 % lower serifs

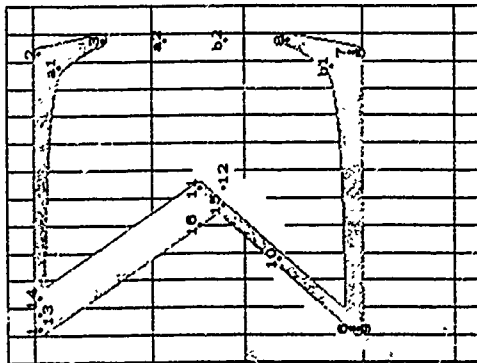


```

"Upper case Greek Pi";
call charbegin("005,13,2sc,2ac-5mc rstem,ph,0,hir rstem);
hpen; lt1x1 = round 2u; x2 = x1;
rt1x2 = round(r-2u); x1 = x2;
top1y1 = h; y1 = y2;
bot1y1 = 0; y2 = y1;
w1 draw 1..2;
w1 draw 3..4;
if ucs ≠ 0: call "a serif(1,4,2,-ucs);
call "b serif(1,4,2,5ucs);
call "c serif(2,4,1,-ucs);
call "d serif(2,4,1,ucs);
call "e serif(3,4,4,-5ucs);
call "f serif(3,4,4,ucs);
call "g serif(4,4,3,-ucs);
call "h serif(4,4,3,ucs);
fi;
w1 draw 1..3.

```

% left stem
 % right stem
 % upper left serif
 % lower left serif
 % upper right serif
 % lower right serif
 % bar

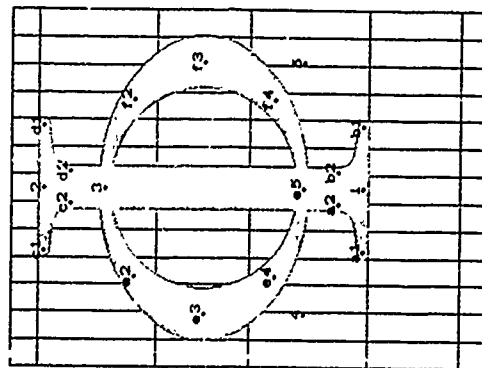
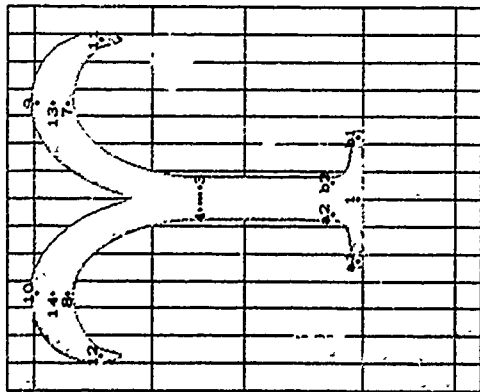


```

"Upper case Greek Sigma";
call charbegin(006,13,0,sc -.5mc armic,ph,0,hic,armic);
new w99, w99 = round 25[w0,w0];
hpen, lt0x1 = round u; r0x2 = round(r - 1.5u); x1 = x2 + .5u;
new ss; ss = 1.4aspect-ucs-u + eps;
if ss + u0 > 25h; new ss; ss = .25h - u0 + eps;
fi;
top0y1 = h, y2 = y1; y1 = y2 - ss;
vpen; bot0y1 = 0; y5 = y1; top0x6 = top0y1, y7 = y0, y8 = y7 + ss;
x1 = x0 = x1, x2 = x1 = x2; x8 = x3;
call "a arm(1,2,3);
call "b arm(6,7,8);
w99 draw 4..5;
if ucs = 0: draw 5..8;
else if u0 ≠ w1: draw [w0]5 [u0]8;
else: draw 5..8;
fi;
fi;
hpen; x9 = x1, x10 = .5[x0,x1]; x16 = round 5u, lt0x12 = lt0x1;
lt0x9 = x16; r0x9 = r0x1; lt0x13 = lt0x1, r0x13 = r0x1;
bot0y9 = 0; y10 = .5[y0,y1], y11 = y10 + .5h, y12 = y10, y13 = y1;
new aa,bb; lt0x15 = aa[lt0x1,x10], y15 = aa[y1,y10];
lt0x15 = bb[x0,x1]; y15 = bb[y0,y1];
w1 draw 13..12;
w0 draw 1..15;
rpen4; w1 draw 10..11;
hpen4; w99 draw 9..10;
hpen; w0 draw 9..11;
draw 14..11

```

% upper diagonal
 % sharpen upper left corner
 % erase excess at right
 % erase excess at left
 % lower diagonal
 % sharpen middle corner

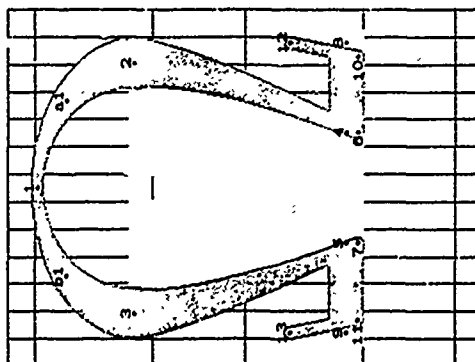
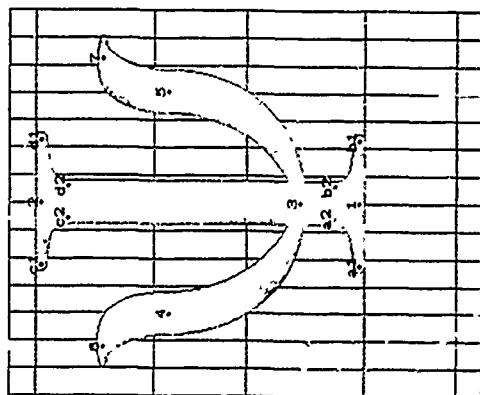


```

"Upper case Greek Upsilon";
call charbegin('007,14,mc(8ph-slant+.5pu),-mc(8ph-slant-.3pu),
    ph,0,mf(8ph-slant-.5pu,2.5pu));
hpen; x1=good,.5r; bot.y1=0; x2=x1; y2=5h;
w1 draw 1..2;
if ucs != 0; call 'a serif(1,4,2,-ucs);
    call 'b serif(1,4,2,ucs);
fi;
new w99; w99=round.5w1;
r1.g9x1=r1.x2; if r99x1=ft.x2;
    lft.g9x1=x2-.25u; x7=x2; x1=x1+10.5u;
vpen; r1.g9x1=round(r-u); y1=8h;
top.g9y1=top.g9h=h+oo; bot.g9y1=bot.g9h;
y1=y1; y2=y2; y3=y3; y4=y4; y5=y5; y6=y6; y7=y7; y8=y8; y9=y9; y10=y10; y11=y11; y12=y12; y13=y13; y14=y14; y15=y15; y16=y16; y17=y17; y18=y18; y19=y19; y20=y20; y21=y21; y22=y22; y23=y23; y24=y24; y25=y25; y26=y26; y27=y27; y28=y28; y29=y29; y30=y30; y31=y31; y32=y32; y33=y33; y34=y34; y35=y35; y36=y36; y37=y37; y38=y38; y39=y39; y40=y40; y41=y41; y42=y42; y43=y43; y44=y44; y45=y45; y46=y46; y47=y47; y48=y48; y49=y49; y50=y50; y51=y51; y52=y52; y53=y53; y54=y54; y55=y55; y56=y56; y57=y57; y58=y58; y59=y59; y60=y60; y61=y61; y62=y62; y63=y63; y64=y64; y65=y65; y66=y66; y67=y67; y68=y68; y69=y69; y70=y70; y71=y71; y72=y72; y73=y73; y74=y74; y75=y75; y76=y76; y77=y77; y78=y78; y79=y79; y80=y80; y81=y81; y82=y82; y83=y83; y84=y84; y85=y85; y86=y86; y87=y87; y88=y88; y89=y89; y90=y90; y91=y91; y92=y92; y93=y93; y94=y94; y95=y95; y96=y96; y97=y97; y98=y98; y99=y99; y100=y100;
x1+x1=x2+x2; x3=x3; x4=x4; x5=x5; x6=x6; x7=x7; x8=x8; x9=x9; x10=x10; x11=x11; x12=x12; x13=x13; x14=x14; x15=x15; x16=x16; x17=x17; x18=x18; x19=x19; x20=x20; x21=x21; x22=x22; x23=x23; x24=x24; x25=x25; x26=x26; x27=x27; x28=x28; x29=x29; x30=x30; x31=x31; x32=x32; x33=x33; x34=x34; x35=x35; x36=x36; x37=x37; x38=x38; x39=x39; x40=x40; x41=x41; x42=x42; x43=x43; x44=x44; x45=x45; x46=x46; x47=x47; x48=x48; x49=x49; x50=x50; x51=x51; x52=x52; x53=x53; x54=x54; x55=x55; x56=x56; x57=x57; x58=x58; x59=x59; x60=x60; x61=x61; x62=x62; x63=x63; x64=x64; x65=x65; x66=x66; x67=x67; x68=x68; x69=x69; x70=x70; x71=x71; x72=x72; x73=x73; x74=x74; x75=x75; x76=x76; x77=x77; x78=x78; x79=x79; x80=x80; x81=x81; x82=x82; x83=x83; x84=x84; x85=x85; x86=x86; x87=x87; x88=x88; x89=x89; x90=x90; x91=x91; x92=x92; x93=x93; x94=x94; x95=x95; x96=x96; x97=x97; x98=x98; x99=x99; x100=x100;
w1 draw 13{1,0}.11{0,-1};
draw 14{-1,0}..12{0,-1};
hpen; w99 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}

"Upper case Greek Phi";
call charbegin('010,13,mc(5ph-slant+.5pu),-mc(5ph-slant-.5pu),
    ph,0,mf(5ph-slant-.5pu,0));
hpen; x1=good,.5r; bot.y1=0; x2=x1; top.y2=h;
w1 draw 1..2;
if ucs != 0; call 'a serif(1,4,2,-ucs);
    call 'b serif(1,4,2,ucs);
    call 'c serif(2,4,1,-ucs);
    call 'd serif(2,4,1,ucs);
fi;
x3=x1; y3=y1; y4=y1; y5=y1; y6=y1; y7=y1; y8=y1; y9=y1; y10=y1; y11=y1; y12=y1; y13=y1; y14=y1; y15=y1; y16=y1; y17=y1; y18=y1; y19=y1; y20=y1; y21=y1; y22=y1; y23=y1; y24=y1; y25=y1; y26=y1; y27=y1; y28=y1; y29=y1; y30=y1; y31=y1; y32=y1; y33=y1; y34=y1; y35=y1; y36=y1; y37=y1; y38=y1; y39=y1; y40=y1; y41=y1; y42=y1; y43=y1; y44=y1; y45=y1; y46=y1; y47=y1; y48=y1; y49=y1; y50=y1; y51=y1; y52=y1; y53=y1; y54=y1; y55=y1; y56=y1; y57=y1; y58=y1; y59=y1; y60=y1; y61=y1; y62=y1; y63=y1; y64=y1; y65=y1; y66=y1; y67=y1; y68=y1; y69=y1; y70=y1; y71=y1; y72=y1; y73=y1; y74=y1; y75=y1; y76=y1; y77=y1; y78=y1; y79=y1; y80=y1; y81=y1; y82=y1; y83=y1; y84=y1; y85=y1; y86=y1; y87=y1; y88=y1; y89=y1; y90=y1; y91=y1; y92=y1; y93=y1; y94=y1; y95=y1; y96=y1; y97=y1; y98=y1; y99=y1; y100=y1;
lft.x1=round u; x1-x1=x2-x2; x3-x3=x4-x3;
call 'e darc(3,4,w1); call 'f darc(3,5,w1).

```

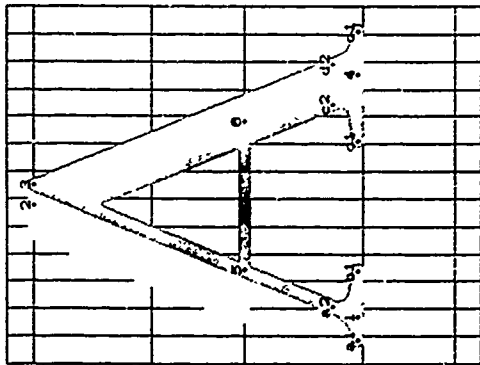


```

"Upper case Greek Psi",
call charbegin('011,1,4,mc(8pt slant + 5pu),-5mc(8pt slant -.5pu),
  ph,0,hic(8pt slant -.5pu));
hpen; x1 = good,5r; bot,y1 = 0; x2 = x1; top,y2 = h;
w, draw 1..2;
if ucs ≠ 0: call 'a serif(1,4,2,-ucs);
  call 'b serif(1,4,2,ucs);
  call 'c serif(2,4,1,-ucs);
  call 'd serif(2,4,1,ucs);
fi;
x2 = x1; y2 = .2h; lft,x3 = u; x4 = good,3u; y3 = 6h; y4 = 8h
y1 = y2; y4 = y2; x1 = x1; x2 = x1; x3 = x1; x4 = x1; x5 = x1;
w, draw 6{1,0}..4{0,-1}..3{1,0};
draw 7{-1,0}..5{0,-1}..3{-1,0};

"Upper case Greek Omega",
call charbegin('012,13,mc(4pt slant + 5pu),-5mc(7.5pt slant -.5pu),
  ph,0,hic(7.5pt slant -.5pu));
hpen; top,y1 = h + oo; y2 = 7h; rlx,x2 = rlx,x3 = round(r - u);
vpen; bot,y4 = bot,y5 = 0; y1 = y4; x1 = x4 = good,8.5u;
x1 + x1 = x2 + x1 + x1 + x1; x2 = x1; x3 = x1 + x1; x4 = x1;
y2 = y1; y1 = y2; y3 = y2; y4 = y2;
w, draw 4..8;
draw 5..9;
lpen; x0 = x1 draw 2{0,-1}..6{x0 - x2, 2(y4 - y2)};
rpen; x0 = x1 draw 3{0,-1}..7{x1 - x1, 2(y1 - y1)};
call 'a arc(1,2,w); call 'b arc(1,3,w);
hpen; draw [w]2{0,-1}..[w]6{x0 - x2, 2(y4 - y2)};
draw [w]3{0,-1}..[w]7{x1 - x1, 2(y1 - y1)};
if ucs ≠ 0: x12 = x3; bot,y10 = 0; top,y12 = 4e;
  x10 + x11 = x12 + x13 = r; y10 = y11; y12 = y11;
  if w1 = w2: x10 = x12;
  else: x10 + .5u = x12;
  rpen; .5u + w1 draw 10..12;
  lpen; .5u + w2 draw 11..13;
fi;
hpen; w1 draw 10..12, draw 11..13;
fi

```



```

"The letter A";
call charbegin("A, 13, 2sc, 2ac, ph, 0, 0);
hpen;
lt,x1 = round 1.5u;  bol,y1 = 0;
rt,x1 = round(r - 1.5u);  bot,y1 = 0;
top,y1 = top,y1 = h + o;
x1 - x1 = x1 - x1;  rt,y2 = rt,y2;
w1 draw 2..4;
y1 = w1 = e;
new az, bb;
x1 - 1 = aa[x1, x1];  y1 = aa[y1, y1];
x1 + 1 = bb[x1, x1];  y1 = bb[y1, y1];
w1 draw 5..6;
hpen;  w1 draw 7..8;
hpen;  w1 draw 9..10;
if ucs < 0;
  call "a serif(1, 0, 3, - 5ucs);
  call "b serif(1, 0, 3, 4 ucs);
  call "c serif(4, 5, 2, - 5ucs);
  call "d serif(4, 5, 2, 4 5ucs);
fi.

```

% right diagonal stroke

% auxiliary variables for intersection of lines

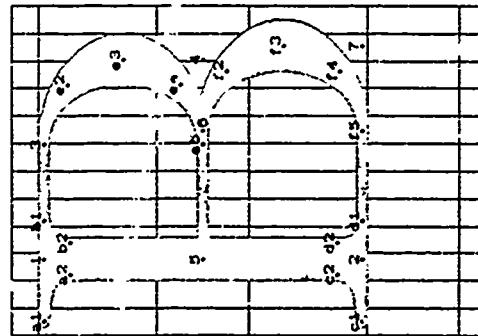
% bar line

% erase excess at upper left

% left diagonal stroke

% left serifs

% right serifs



```

"The letter B";
call charbegin("B, 12, 2sc, - 5mc(75ph-slant - 5pu), ph, 0, lu( 75ph slant - 5pu));
hpen;  lt,x1 = lt,x1 = round 2u,  top,y1 = h;
bot,y1 = 0;
w1 draw 1..2;
if ucs < 0;
  call "a serif(1, 4, 2, - 5ucs);
  call "b serif(1, 4, 2, 5ucs);
  call "c serif(2, 4, 1, - 5ucs);
  call "d serif(2, 4, 1, 5ucs);
fi;
x1 = 1/2(2u, r);  y1 = y1;
rt,x1 = round(r - u);  y1 = good,y1,h;
w1 draw 3;
call "e darc(3, 4, w1);
x1 = x1;  x1 = x1 + 1/2u,  y1 = y1;  w1;
rt,x1 = round(r - 1/2u);  bot,y1 = 0;
w1 draw 5..6;
call "f darc(6, 7, w1);
x1 = x1,  y1 = y1;  w1 draw 7..8.

```

% stem

% upper serif

% lower serif

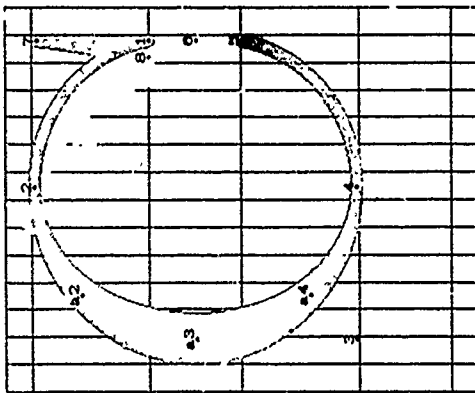
% upper bar line

% upper bowl

% middle bar line

% lower bowl

% lower bar line

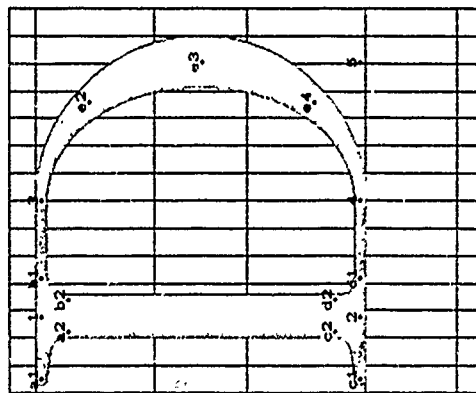


```

The letter C":
if ucs ≠ 0: call charbegin("C, 14, mc lbowl, - 5mc(ph slant - 5pu),
    ph, 0, lie(ph slant - 5pu));
else: call charbegin("C, 11, mc lbowl, - 5mc(ph slant - 5pu),
    ph, 0, lie(ph slant - 5pu));
fi;
hpen;
rt0x1 = round(r - u); x3 = x1; lft0x3 = round u; x2 = x1 = 7.5u;
top0y2 = h + oo; bot0y2 = -oo; y1 = y1, y2 = 5[y2, y1];
if ucs = 0: x0 = 13u; new aa, x1 = aa[x2, x0],
    y1 = (sqrt(1 - aa.aa))[y0, y1];
else: if m < .6h y1 = good0, 7h, else y1 = good0, m;
fi;
x0 = x1; x2 = x1, top0y1 = h, lft0x2 = lft x1; y1 = y1,
u0 draw 1. 7, 8..7;
ipen; w1 draw (6) 1 2(-1, 0);
fi;
hpen, y1 = h - y1;
u0 draw (6) 1..2(-1, 0);
call "a darc(2, 3, w2);
u0 draw 3(1, 0) 5(.6).

```

% upper serif
% erase spurious part
% upper right stroke
% main stroke
% lower right stroke

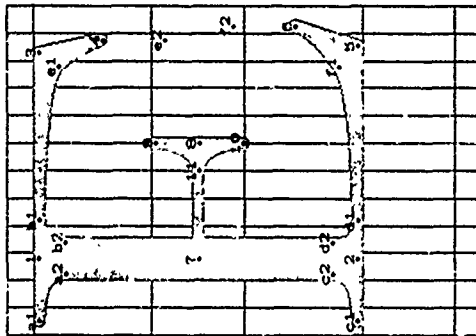


```

The letter D":
call charbegin("D, 14, 2sc, -mc rhowl, ph, 0, m[rhowl, 0]);
hpen; lft x1 = round 2u; x2 = x1, top y1 = h, bot y1 = 0,
w1 draw 1 2;
if ucs ≠ 0:
    call "a serif(1, 4, 2, -ucs);
    call "b serif(1, 4, 2, 5ucs);
    call "c serif(2, 4, 1, -ucs);
    call "d serif(2, 4, 1, 5ucs);
fi;
x3 = x1 = 7u, rt x3 = round(r - u),
y1 = y1; y1 = y1 = y1;
u0 draw 1..3;
call "a darc(3, 5, w2);
u0 draw 4 2

```

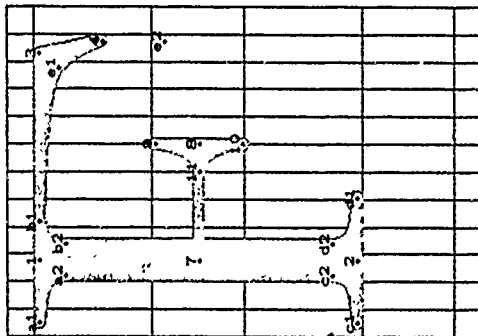
% stem
% upper serif
% lower serif
% upper bar line
% bowl
% lower bar line



```

"The letter E";
call charbegin(-E, 12, 2sc, sc = .5mc armic, ph, 0, hic armic);
lipen; ift, x1 = round 2u; x2 = x1; top, y1 = h; bot, y2 = 0,
w1 draw 1..2;
if ucs >= 0:
  call ~a serif(1, 4, 2, -ucs);
  call ~b serif(1, 4, 2, 5ucs);
  call ~c serif(2, 4, 1, -ucs);
  call ~d serif(2, 4, 1, 5ucs);
fi;
new ss, ss = 1.4aspect ucs-u + eps;
if ss + u0 > .25h, new ss; ss = .25h - u0 + eps;
fi;
r4oz1 = round(r - 1.5u), x1 = x1 + 3u; y1 = y1, y1 = y1 - ss;
r4oz2 = round(r - 1.25u), x0 = x1 + 5u; y1 = y1, y1 = y1 + ss;
call ~e arm(1, 3, 4);
call ~f arm(2, 5, 6);
x7 = x1; y7 = y8 = .5(y1, y2); x8 = 200d, 7u;
u0 draw 7..8;
if ucs < 0: x9 = x10 = x8; y9 = y8 + 7ss, y10 = y8 - 7ss,
  if u0 = w1: u0 draw 9..10;
  else: x11 = x8 - u; y11 = y8;
    minvr 0; minvr 0;
    u0 ddraw 11{1, 0}..10{0, -1}, 8 10;
    ddraw 11{1, 0}..9{0, 1}, 8 9;
    minvr .5; minvr .5;
fi;
fi;

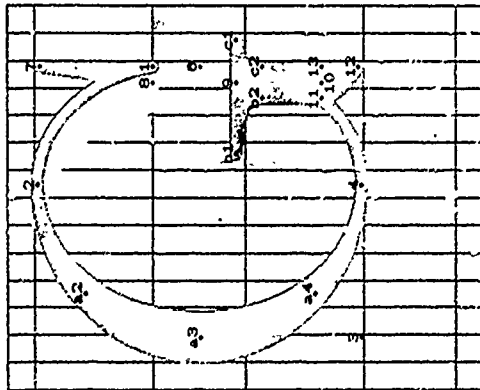
```

```

"The letter F";
call charbegin("F", 12, 25, sc = mc(armic - 2.5pt), pl, 0, ml(armic, 2.5pt));
hpen; ft_x1 = round 2u; x2 = x1; top_y1 = h; bot_y2 = 0;
w1 draw 1..2;
if ucs ≠ 0:
  call "a serif(1, 4, 2, -ucs);
  call "b serif(1, 4, 2, 5ucs);
  call "c serif(2, 4, 1, -ucs);
  call "d serif(2, 4, 1, ucs);
  % stem
  % upper stem serif
  % lower stem serif
fi;
new ss, ss = 1.4aspect·ucs·u + eps;
if ss + w1 > .25h: new ss; ss = .25h - w1 + eps;
fi;
rt_x1 = round(x - 1.5u); x1 = x1 + .5u, y1 = y1, y1 = y1 - ss;
call "e arm(1, 3, 4);
x1 = x1, y1 = y1, x2 = 5[y1, y1], x3 = good_7u,
w1 draw 7..8;
if ucs ≠ 0: x1 = x10 = x1, y1 = y1 + .7ss, y10 = y1 - 7ss;
  if w1 = w1: w1 draw 9..10,
  else: x11 = x1 - u; y11 = y1;
  minvr 0; minvs 0;
  w1 ddraw 11{1, 0} 10{0, -1}, 8..10;
  ddraw 11{1, 0} 9{0, 1}, 8..9;
  minvr 5, minvs .5;
  % lower arm serif
  % lower arm serif
fi;
fi;

```

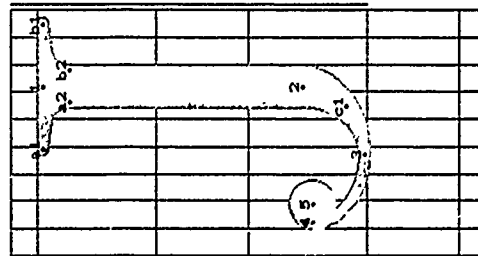
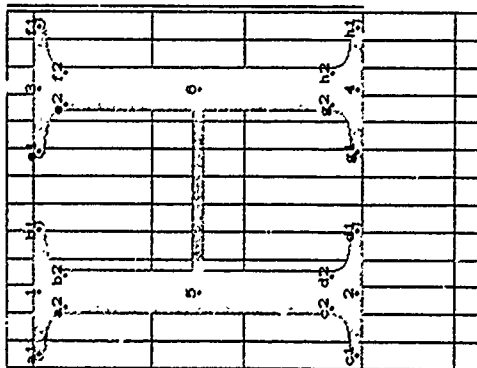


```

The letter "G";
if ucs = 0: if m < .6h: call charbegin("G, 14, mc-lbowl,
    sc - mc(1/3 ph-slant + (sc - 1.5)pu),
    ph, 0, (1 - m)(1/3 ph-slant + (sc - 1.5)pu));
else: call charbegin("G, 14, mc-lbowl,
    1 - mc(px-slant + (sc - 1.5)pu),
    ph, 0, (1 - m)(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 - m)(ph-slant + (sc - 1.5)pu));
fi;
hpen;
rt0x1 = rt(x0) = round(r - 2u); if x1 = round u, x2 = x1 = 7.5u,
top0y1 = h + oo, bot0y1 = -oo, y1 = y0; y2 = 5[y1, y0];
if m < .6h: y1 = good0, 3h; else: y1 = good0, m,
fi;
x0 = x1, y1 = good0, 1[e, m]; x0 = x10;
if ucs != 0: x1 = x1; top0y1 = h; if ucs = lf x1, y2 = y1,
w0 ddraw 1..7, 8..7;
lpen#; w; draw (6..1) 1..2{-1, 0},
fi;
hpen; w; draw (6..1) 2{-1, 0};
call "a dare(2, 3, w);
if u1 = w: w; draw 4{1, 0}..9{0, 1};
bot0y10 = 0; draw 9..10;
else: y1 = y10 = y1 = 3y;
if0x11 = lf(x0);
rt0x12 = rt0x11 = rt(x0); bot0y12 = 0;
w; draw 4{1, 0}..11..9{0, 1};
w; draw 9..10;
w; ddraw 13..12, 11{0, -1}..12{2(x12 - x1), y12 - y1};
fi;
if ucs < 2: call "b serif(9, 4, 10, -2);
else: call "b serif(9, 4, 10, -ucs);
fi;
if ucs > 0: call "c serif(9, 4, 10, +1),
fi

```

% upper serif
 % erase spurious part
 % upper right stroke
 % main stroke
 % lower right stroke
 % stem and spur
 % lower right stroke
 % stem
 % spur
 % lower serif



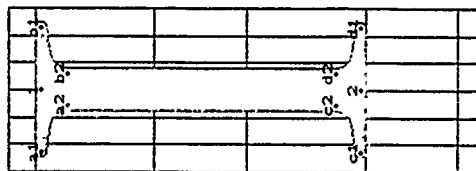
```

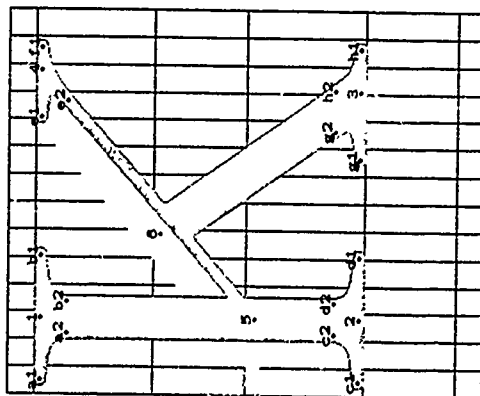
"The letter H";
call charbegin(1,13,2sc,2sc-.5mc:stem,ph,0,hic:stem);
hpen; if(x1==round 2u, x2==x1,
rt,x3==round(r-2u); x4==x3;
top,y1==h; y2==y1;
bot,y2==0; y3==y1;
w1 draw 1..2;
w1 draw 3..4;
if ucs!=0: call `a serif(1,4,2,-ucs),
call `b serif(1,4,2,ucs);
call `c serif(2,4,1,-ucs);
call `d serif(2,4,1,ucs);
call `e serif(3,4,4,-ucs);
call `f serif(3,4,4,ucs);
call `g serif(4,4,3,-ucs);
call `h serif(4,4,3,ucs);
fi;
x5==x1; x6==x1; y5==y6==5h;
w1 draw 5..6.

"The letter I";
call charbegin(1,6,0,-5mc(ph:slant-5pu),ph,0,hic(ph slant-5pu));
hpen; x1==x2==good,5r; top,y1==h; bot,y2==0;
w1 draw 1..2;
new ss;
if ucs!=0: ss==ucs; else ss==2-5w1/u;
fi;
call `a serif(1,4,2,-ss); call `b serif(1,4,2,ss);
call `c serif(2,4,1,-ss); call `d serif(2,4,1,ss).

"The letter J";
call charbegin(1,9,0,2sc-.5mc:stem,ph,0,hic:stem);
hpen; rt,x1==round(r-2u); top,y1==h; y2==2h; x2==x1,
w1 draw 1..2;
if ucs!=0: call `a serif(1,4,2,-ucs),
call `b serif(1,4,2,ucs);
fi;
if(x1==round 2u, y1==y2==h,
x3==5[x1,x2]; bot,y3==0;
call `c arc(3,2,w1);
w1 draw 3{(-1,0)..4(0,1)}; open; w1 draw 5.

```



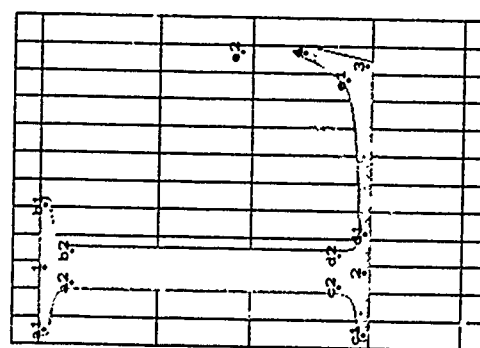


```

"The letter K";
call charbegin("K, 14, 2sc, 2sc - .5mc(ph slant + (.5ucs + 2sc - 1.5)pt),
  ph, 0, hic(ph slant + (.5ucs + 2sc - 1.5)mu));
lpen; lft, x1 = round 2u; x1 = x2 = x;
rt, x2 = round(r - 2u);
top, y1 = round(r - 2u);
new aa, bb;
rt, x1 = aa[rt, x1, rt, x1]; y1 = aa[y1, y1];
rt, x2 = bb[rt, x2, rt, x2]; y2 = bb[y2, y2];
w1 draw 6..3;
lpen; w1 draw 4..5;
w1 draw 1..2;
if ucs < 0: call "a serif(1, 4, 2, -ucs);
  call "b serif(1, 4, 2, ucs);
  call "c serif(2, 4, 1, -ucs);
  call "d serif(2, 4, 1, ucs);
  call "e serif(4, 0, 5, -ucs);
  call "f serif(4, 0, 5, .5ucs);
  call "g serif(3, 5, 1, -ucs);
  call "h serif(3, 5, 1, .5ucs);
fi.

```

% lower diagonal
 % crane excess
 % stem
 % upper diagonal
 % upper stem serif
 % lower stem serif
 % upper diagonal serif
 % lower diagonal serif

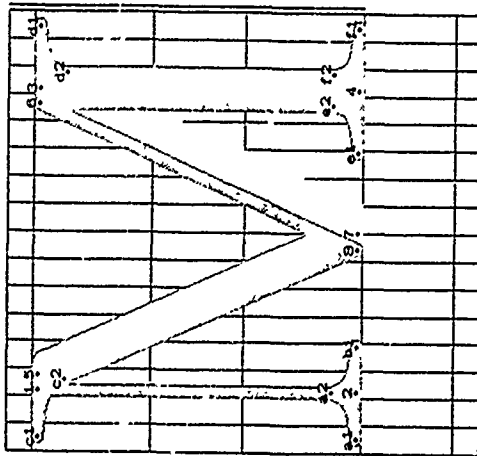


```

"The letter L";
call charbegin("L, 12, 2sc, sc, ph, 0, 0);
lpen; lft, x1 = round 2u; x2 = x1; top, y1 = h; bot, y2 = 0;
w1 draw 1..2;
if ucs < 0: call "a serif(1, 4, 2, -ucs);
  call "b serif(1, 4, 2, ucs);
  call "c serif(2, 4, 1, -ucs);
  call "d serif(2, 4, 1, .5ucs);
fi;
rt, x1 = round(r - 1.5u); x1 = x1 + .5u; y1 = y1 + 1.5u;
call "e arm(2, 3, 4).

```

% stem
 % upper stem serif
 % lower stem serif
 % arm and arm serif



```

"The letter M",
call charbegin("M, 16, 2sc, 2sc - 5mc rstem, ph, 0, hic rstem);
hpen; lft0x1 = round 2u; x1 = x2; top0y1 = h; bot0y2 = 0,
w1 draw 1 2,
rt0x1 = round(r - 2u); x1 = x1; top0y1 = h; bot0y1 = 0,
lft0x2 = lft0x1; lft0x3 = lft0x1; y1 = y1; y2 = y1;
x1 - x2 = x3 - x2; lft0x1 = lft0x3; bot0y1 = 0; y2 = y1,
w1 draw 5 .7;
rpen#; w1 draw 8..6;
hpen; w0 draw 8..6;
w1 draw 3..4;
if ucs # 0: call "a serif(2, 0, 1, --ucs),
call "b serif(2, 0, 1, ucs);
call "c serif(5, 4, 1, --ucs),
call "d serif(3, 4, 4, ucs),
call "e serif(4, 4, 3, --ucs);
call "f serif(4, 4, 3, ucs);
fi.

```

% left stem

% left diagonal

% erase excess at lower right

% right diagonal

% right stem

% lower left serif

% upper left serif

% upper right serif

% lower right serif

```

"The letter N",
call charbegin("N, 14, 2sc, 2sc - 5mc rstem, ph, 0, hic rstem);
hpen; lft0x1 = round 2u; x1 = x2; top0y1 = h; bot0y2 = 0;
w1 draw 1 .2;
rt0x1 = round(r - 2u); x1 = x1; top0y1 = h; bot0y1 = 0;
lft0x2 = lft0x1; lft0x3 = lft0x1; y1 = y1; y2 = y1,
w1 draw 5 .6;
rpen#; w1 draw 4..3;
hpen; w0 draw 4..3;
if ucs # 0: call "a serif(2, 0, 1, --ucs);
call "b serif(2, 0, 1, ucs);
call "c serif(5, 5, 6, --ucs);
call "d serif(3, 0, 4, --ucs);
call "e serif(3, 0, 4, ucs);
fi.

```

% left stem

% diagonal

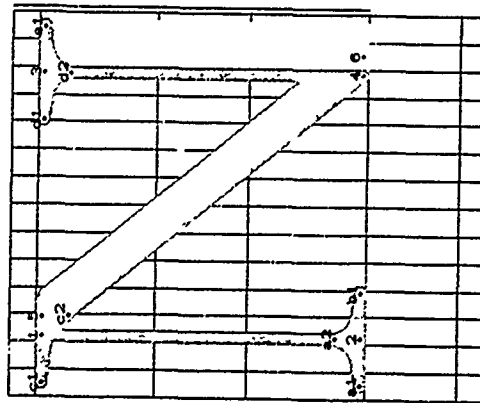
% erase excess at lower right

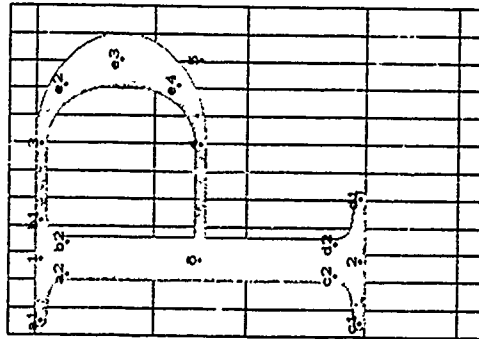
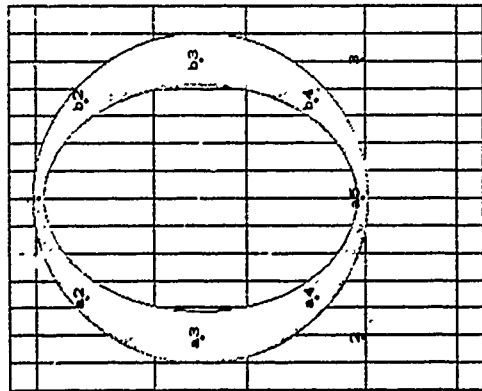
% right diagonal

% lower left serif

% upper left serif

% upper right serif



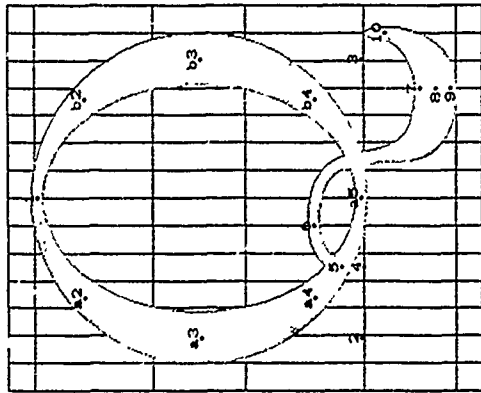


```

"The letter O";
call charbegin("O, 14, mc: bowl, --mc: bowl, ph, 0, ml: bowl, 0));
hpen;
if fixwidth < 0: new save; save = sqrttwo;
    new sqrttwo; sqrttwo = sqrt save;
    if fixwidth < 0: round 1.5u;
    else. if fixwidth < 0: round u;
fi;
x1 = r - x1;
topy1 = h + oo;
boty1 = -oo; y1 = y1; x1 = r - x1;
call "a dare(1, 2, w1);
call "b dare(1, 3, w2);
if fixwidth < 0: new sqrttwo; sqrttwo = save;
fi.

"The letter P";
call charbegin("P, 12, 2sc, --mc(.75ph slant -- 3pu), ph, 0, ml(.75ph slant -- 5pu, 2 5pu));
hpen, if fixwidth < 0: round 2u; x2 = x1, topy1 = h; boty1 = 0;
if ucs < 0:
    call "a serif(1, 4, 2, -ucs);
    call "b serif(1, 4, 2, 5ucs);
    call "c serif(2, 4, 1, -ucs);
    call "d serif(2, 4, 1, ucs);
fi;
x1 = x1 = 7u; rtx1 = round(r - u); x0 = x1,
y1 = y1, y0 = y1 = 5u = goodh, 5h;
w1 draw 1..3;
call "e dare(3, 5, w3);
w1 draw 4..6.
% super-supercilipse
% axis of left-right symmetry
% left part of bowl
% right part of bowl
% stem
% upper serif
% lower serif
% upper bar line
% bowl
% lower bar line

```



```

"The letter Q";
call charbegin("Q, 14, mc-lbowl, -mc-rbowl, ph, pdd, ml(rbowl, 0));
hpen;
if fixwidth ≠ 0: new save; save = sqrttwo;
    new sqrttwo, sqrttwo = sqrt save,
    lft, x2 = round 1.5u;
else: lft, x2 = round u;
fi;

x1 = r - x1;
topy1 = h + oo;
boty2 = -oo; y1 = y2; x1 = r - x2;
call "a darel(1, 2, w);
call "b dar(3, w);
x1 = x3 = 4.5u; y1 = 0;
new aa; x2 = aa(x1, x2); y2 = (sqrt(1 - aa.aa))(5y1 + .5y2, y2);
x0 = 6u; y0 = 1h + 1.5u;
w0 draw (4 .) 5 6{1, 0};
vpen; x7 = x8 = x0 = 11u; boty8 = boty0 = -dd, topy8 = topy0;
x10 = round 13u; topy10 = 0;
w0 gldraw 6{1, 0}..7{1, 0}, 6{1, 0}. 9{1, 0};
w0 draw 8{1, 0}..10{0, 1};
if fixwidth ≠ 0: new sqrttwo; sqrttwo = save,
fi;

```

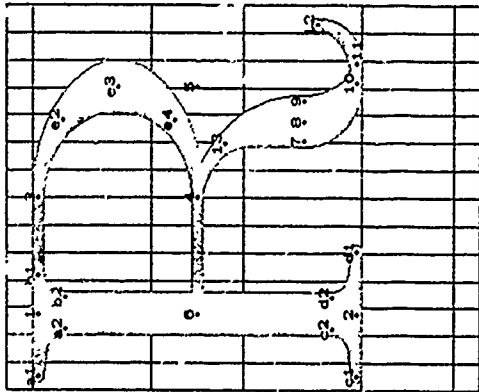
% super-supercellpse

% axis of left-right symmetry

% left part of bowl
% right part of bowl

% left part of tail

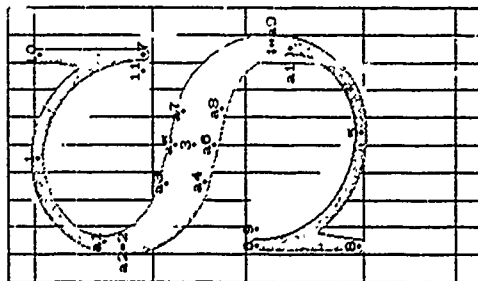
% middle part of tail
% right part of tail



```

The letter R";
lipen;
if ucs == 0 call charbegin(R, 12.5, 2sc, -mc(.75ph slant - .75pu),
    ph, 0, ml(.75ph slant - .75pu, 0));
    rt z, - round(r - .75u);
else call charbegin(R, 14, 2sc, 0, ph, 0, .75ph slant - 1.5pu);
    rt z3 = round(r - 2u);
fi;
ift, x1 = round 2u; x2 = x1; top y1 = h, bot y2 = 0;
w1 draw 1..2;
if ucs != 0
    call a serif(1, 4, 2, -ucs);
    call b serif(1, 4, 2, .5ucs);
    call c serif(2, 4, 1, -ucs);
    call d serif(2, 4, 1, ucs);
fi;
x3 = x1 = 7u; x4 = x1;
y1 = y1; y6 = y1 = y5 = good0.5h;
w1 draw 1..3;
call e darc(3, 5, w3);
w1 draw 4..6;
if ucs == 0 rt x7 = round(r - u); bot y7 = 0;
    w1 draw 4..7;
else x3 = good0(x3 - 1.25u); lft x8 = lft x7; rt x8 = rt x7;
    y7 = y3 = y5 = y1;
    y13 = y1; x13 = 1/sqrt(2)*x1;
    draw w1{1, 0}..1.25[w1, w1]13{x3 - x1, y3 - y1}..
        w1{8, 0, -1};
    y10 = y1 = y2; y12 = .25[y2, y1];
    x10 = 5[x7, x12], x11 = 5[x7, x12]; rt x12 = round(r - .5u);
    w1 ddraw 7{0, -1}..13{1, 0}..12{0, 1},
        9{0, -1}, 11{1, 0}..12{0, 1};
fi;
% stem
% upper serif
% lower serif
% upper bar line
% bowl
% lower bar line
% diagonal stroke
% upper tail
% lower tail and hook

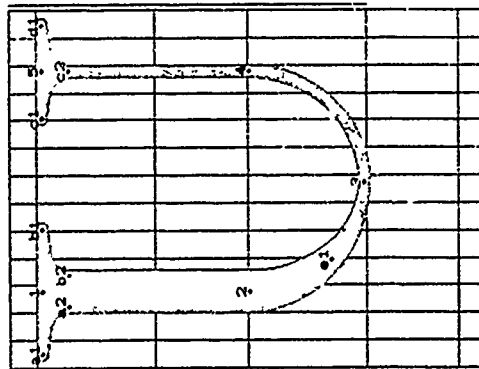
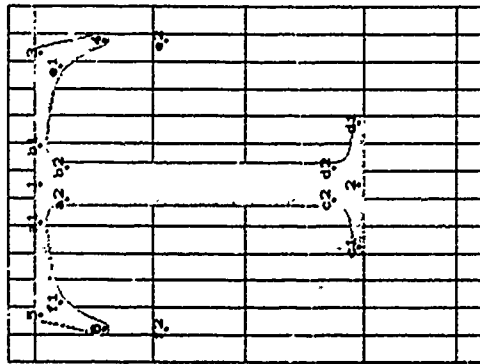
```

```

"The letter S";
call charbegin('S', 10, 0, -5mc(ph-slant - .5pu), ph, 0, hic(ph-slant - .5pu));
new w0; w0 = round 2{w0, w1};
hpen; top0y1 = h + .00; bot0y1 = -0.0;
x1 = .5r; y1 = .52h; lft0x2 = round u; r00x1 = round(r - u);
if ucs = 0: x1 = x2 = x3; x0 = x2; x7 = x1;
w0 draw 6{x2 - x0, 3{y1 - y0}} .5{1, 0};
draw 7{x1 - x7, 2{y1 - y7}} 1{-1, 0};
else: if u0 = w1; x1 = x2 = x3; x0 = x2; x7 = x1;
w0 = {top0, y1}; y1 = {bot0, y1};
w0 draw 6{x2 - x0, 3{y1 - y0}} .5{1, 0};
draw 7{x1 - x7, 2{y1 - y7}} 1{-1, 0};
x8 = x0 = x6; y8 = y0 + ucs-aspect u + eps; bot0y8 = 0;
x10 = x1 = x2; y11 = y1 - ucs-aspect u - eps; top0y10 = h;
w0 draw 8. 9;
draw 10...11;
else: x1 = x2 = .5u; x3 = x1 + .5u; x0 = x2; r00x2 = round(r - 1.5u);
w0 = good0 {h - 1; y1 = good0 {h + 1;
bot0y8 = 0; y1 = y0; x8 = x0; r00x0 = r00x0;
top0y10 = h; y11 = y7; x10 = x1; lft0x7 = lft0x11;
w0 ddraw 6. 8, 9, 8;
ddraw 7. 10, 11...10;
rpen#; w1 draw 6{0, -1} .5{1, 0};
hpen#; w1 draw 7{0, 1}...1{-1, 0};
hpen; w0 draw 6{0, -1} .5{1, 0};
draw 7{0, 1}...1{-1, 0};
fi;
fi;
call ~a sdraw(1, 2, 3, 4, 5, w0, round(pixels-pwv aspect + blacker),
-h/(50u)).

```



```

"The letter T";
call charbegin("T, 13, sc + .75mc-ph-slant, sc -- mc(armic -- 2.5pu),
    ph, 0, mi(armic, 2.5pu));
hpen; x1 == good, 6.5u, x2 == x1;
top y1 == h; bot y2 == 0;
w1 draw 1..2;
if ucs != 0: call "a serif(1, 4, 2, --.5ucs),
    call "b serif(1, 4, 2, .5ucs);
    call "c serif(2, 4, 1, -- ucs);
    call "d serif(2, 4, 1, ucs);
fi;
new ss; ss = 1.4aspect-ucs u + eps,
if ss + u1 > .25h: new ss; ss == .25h -- u1 + eps;
fi;
r1(x1) == round(r -- 1.5u); x1 == x1 + .5u; y1 == y1, y2 == y2 -- ss;
l1(x1) == round 1.5u; x1 == x1 -- .5u; y1 == y1, y2 == y2 -- ss;
call "o arm(1, 3, 4);
call "f arm(1, 5, 6).

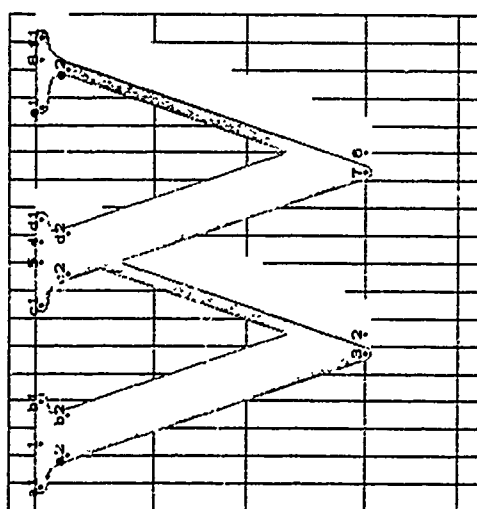
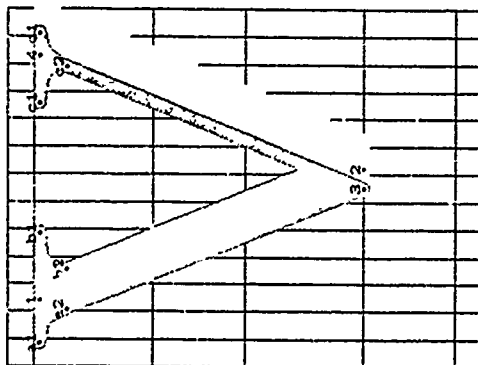
"The letter U";
call charbegin("U, 13, 2sc(1 -- mi) + mc(3ph slant + pu), 2sc -- 5mc rstem,
    ph, 0, huc-rstem);
hpen;
if fixwidth != 0: l1(x1) == round(1.5u), r1(x1) == round(r -- 1.5u);
else: l1(x1) == round 2u, r1(x1) == round(r -- 2u);
fi;
x2 == x1; x1 == x1, x2 == .5[x2, x1];
top y1 == h; y2 == y1; y2 == y1 == 36h, bot y1 == --oo;
if ucs != 0: call "a serif(1, 4, 2, --ucs);
    call "b serif(1, 4, 2, ucs);
    call "c serif(5, 0, 4, --ucs);
    call "d serif(5, 0, 4, ucs);
fi;
w1 draw 1..2;
call "e arc(3, 2, w);
w1 draw 3{1, 0}..4{0, 1};
draw 4..5.

```

```

% stem
% upper bracketing
% lower serif
% left serif
% right serif
% upper left stroke
% lower left stroke
% upper right stroke
% lower right stroke
% upper right arm and serif
% upper left arm and serif

```



```

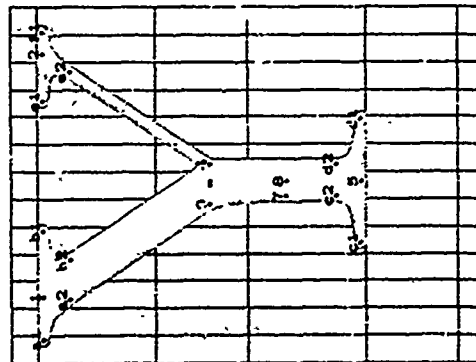
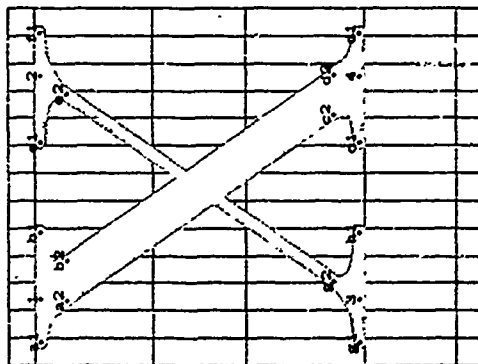
The letter V",
call charbegin("v, 13, 2sc + mc(ph:slant + 5pu), 2sc - mc(rv - 2.5pu),
    ph, 0, m[rv, 2.5pu]);
hpen; lt(x1 == round 1.5u, rt(x1 == round(r - 1.5u),
x2 == x1 == x1 - x1, lt(x1 == lt(x1,
top(y1 == h; y1 == y1; bot(y2 == -o, y1 == y1,
w; draw 1..2;
rpen#; w; draw 3..4;
hpen; w; draw 3..4;
if ucs != 0: call "a serif(1, 5, 2, - 5ucs);
    call "b serif(1, 5, 2, ucs);
    call "c serif(1, 0, 3, - 5ucs);
    call "d serif(1, 0, 3, 5ucs);
fi.

"The letter W",
call charbegin("w, 18, 2sc - 1 mc(ph:slant + 5pu), 2sc - mc(rv - 2.5pu),
    ph, 0, m[rv, 2.5pu]);
hpen; lt(x1 == round 1.5u;
rt(x1 == rt(x1, x1 == x1 == x1, rt(x1 == r - 1.5u,
% x1, x1, and x1 are approximations to x1, x1, and x1
x1 == x1 == x1 - x1 == x1 == x1 == x1 == round(x1 - x1),
% The idea is to draw two V's displaced by an integer amount.
top(y1 == h; bot(y2 == -o; y1 == y1 == y1 == y1 == y1 == y1;
rt(x1 == rt(x1, lt(x1 == lt(x1, x2 == x1 == x1 - x1;
w; draw 1..2;
rpen#; w; draw 3..4;
hpen; w; draw 3..4;
w; draw 5..6;
rpen#; w; draw 7..8;
hpen; w; draw 7..8;
if ucs != 0: call "a serif(1, 5, 2, - 5ucs);
    if w == w; call "b serif(1, 5, 2, ucs);
    else: call "c serif(1, 5, 2, 5ucs);
fi;
call "e serif(5, 5, 6, - 5ucs);
call "d serif(5, 5, 6, 5ucs);
call "a serif(8, 0, 7, - 5ucs);
call "f serif(8, 0, 7, 5ucs);
fi.

% first diagonal
% erase excess
% second diagonal
% third diagonal
% erase excess
% fourth diagonal

% left serif
% middle serif
% right serif

```



```

"the letter X";
call charbegin("X, 13, 2sc, 2sc -- .5mc rv, ph, 0, huc:rv);
hpen; lt:xt1 = round 1.5u; rt:xt1 = round(r -- 1.5u);
top:y1 = h; bot:y1 = 0; y2 = y1; y3 = y1;
w5 draw l...4;
w6 draw 3..2;
if it uses ≠ 0: call "a serif(1, 5, 4, --.5ucs);
call "b serif(1, 5, 4, ucs);
call "c serif(4, 5, 1, --ucs);
call "d serif(4, 5, 1, .5ucs);
new ss; ss = 5(w5 - w6)/w1;
call "a serif(2, 0, 3, --ucs -- ss);
call "f serif(2, 0, 3, .5ucs + ss);
call "g serif(3, 0, 2, .5ucs - ss);
call "h serif(3, 0, 2, ucs + ss);

ff.

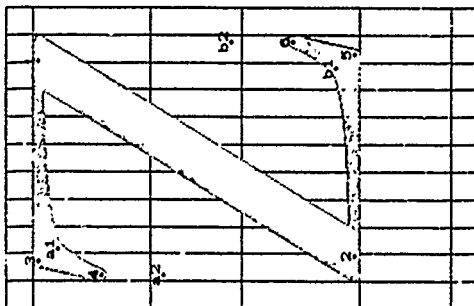
"the letter Y";
call charbegin("Y, 13, 2sc + mc(ph slant + .5pu), 2sc - mc(rv - 2.5pu),
ph, 0, mlrv, 2.5pu);
hpen; lt:xt1 = round 1.5u, rt:xt2 = round(r -- 1.5u),
top:y1 = top:y2 = h;
rt:xt1, xt1 = rt:xt2, xt2 = xt1 -- xt1 -- xt1, lt:xt2 = lt:xt1;
x1 = good:xt1; rt:xt1 = rt:xt2 -- rt:xt2, y1 = y1 -- y1 -- y1, y2 = y2 -- y2 -- y2, bot:w, w = 0,
x3 = x2 = x4; lt:xt2 = lt:xt2;
w5 draw l...3;
w6 draw 0; m1w5 0; m1w6 0;
w7 draw 9(x1 -- x1, y1 -- y1) .7{0, --1}, 6 8;
m1w5 .5; m1w6 .5;
w1 draw 6..5;
w2 draw 4..2;
if it uses ≠ 0: call "a serif(1, 5, 3, --.5ucs);
call "b serif(1, 5, 3, ucs);
call "c serif(5, 4, 6, --ucs);
call "d serif(5, 4, 6, ucs);
rt:xt1 = serif(2, 0, 4, -w);
call "f serif(2, 0, 4, .5ucs);
call "f serif(2, 0, 4, .5ucs);

% the letter X
% upper left serif
% lower right serif
% correction to ucs makes up like w
% upper right serif
% lower left serif

% upper left serif
% lower right serif
% correction to ucs makes up like w
% upper right serif
% lower left serif

% left diagonal
% transition
% stem
% right diagonal
% upper left serif
% lower serif
% upper right serif

```



```

%The letter 'z'
call charbegin('z', 11, 0, -5mc(ph slant - 5pu), ph, 0, tie(ph slant - 5pu)),
hpen; lift; x2 = round u; rt; x1 = round(r - u);
top; y1 = h; bot; y2 = 0;
new ss; ss = 1.4aspect ucs u + eps;
if ss + u0 > 25h new ss; ss = 25h - u0 + eps;
fi;
lift; x1 = round 1.5u; x2 = x1 - 5u; y1 = y1; y2 = y1 - ss;
rt; x2 = round(r - 1.5u); x0 = x2 + 5u; y2 = y2; y0 = y2 + ss;
call 'a arm(1, 3, 4);
w; draw 1 2;
call 'b arm(2, 5, 6).
% upper arm and serif
% diagonal
% lower arm and serif

```

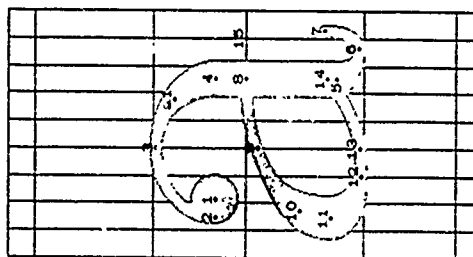
The file roman1.mf

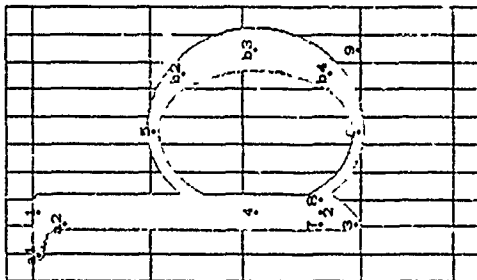
% This lower-case roman alphabet was prepared by D. V. Knuth in November, 1979,
 % inspired by the Monotype alphabet used in *The Art of Computer Programming*
 % Its spacing is intended for text only.
 % Character codes '141'-'172' are generated

```

%The letter 'a'
call charbegin('a', 9, 0, sc, px, 0, {jpc, px} slant + 5pwt + {sc - 2}pu),
open; lift; x1 = round 1.25u;
if top {top, topc + 2} > 9[c, m] top; y1 = 9[c, m];
else y1 = top, topc + 2;
fi;
w; draw 1;
hpen; lift; x2 = lift, x1; y2 = y1; x3 = 4u; top; y3 = m + oo;
x1 = x2 = good(r - 2.5u); y1 = 1[c, m];
u; draw 2{0, 1} 3{1, 0}; call 'a arc(3, 4, w);
if les = 0; bot; y2 = 0; w; draw 4 5;
else: y2 = 2{top, 0, d}, rt; x3 = 5{rt, x2, rt; x2};
bot; x3 = 0; rt; x2 = round(r - 5u); y2 = c/3;
w; draw 4 5;
draw {w; #1}{0, -1} {w; #1}{0, 1} 7{0, 1};
fi;
x0 = x2; y0 = c; x1 = 4u; y1 = 9[w, y0]; x10 = x1 + 2.5u; y10 = 5[w, y0];
x11 = good, 1; u; y11 = 2[w, y1];
x2 = 3u; bot; y2 = -oo; x3 = 4u; y3 = 0.15[y1, y0];
x11 = x0; y11 = 3[y1, y0]; x12 = r - u; y12 = c;
draw {w; #1}{-1, 0} {w; #1}{9 18[w, w]}10 {w; #1}{0, -1}
1.7[w, w]}12{1, 0} {w; #1}{1.3 1.4} 15).
% bulb
% shoulder
% stem
% stem
% hook
% bowl

```





```

The letter b:
call charbegin("b, 10, sc, 0, ph, 0, 5px slant + hrc),
hpen, x1 = x2 = x1 = good, 2.5u; top, y1 = h;
lft, x1 = lft, x1; x1 = x2; x1 = x0 = 5(r + u); r1, x1 = r1, x1;
bot, y1 = 0; y1 = 5(y, y1); top, y1 = m + oo; bot, y1 = -oo,
new aa; r1, x1 = aa(x0, x1);
y1 = y1 = y1 = (sqrt(1 - aa-aa))(y1, y1);
if lcs ≠ 0. call "a serif(1, 1, 2, -lcs);
fi;
w1 draw 1..2;
w1 draw 7 3, 8(0, -1) 3(x1 - x0, 5(y1 - y1));
w1 draw 6(-1, 0) 4(0, 1) 5(1, 0);
if w1 > 1.5u r1, x1 = round(r - .75u);
else: x1 = good, 2(r - 1.5u);
fi;
y1 = y1; call "b darc(5, 9, w1)

```

```

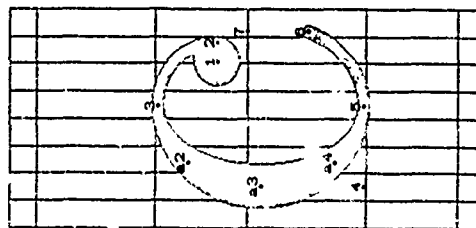
The letter c:
call charbegin("c, 8, 0, 0, px, 0, px slant - pu),
open; r1, x1 = round(r - u);
if top, (top, top, c + 2) > 9[e, m] top, y1 = 9[e, m];
else: y1 = top, top, c + 2;
fi;

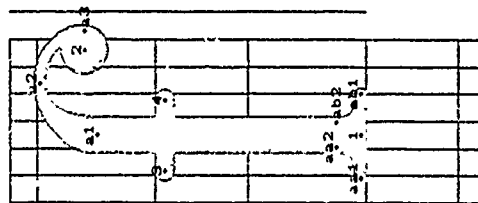
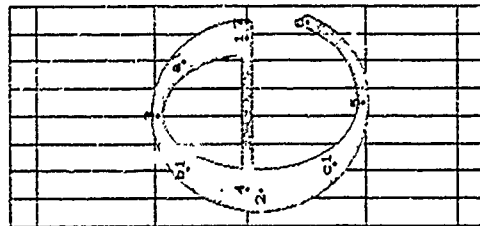
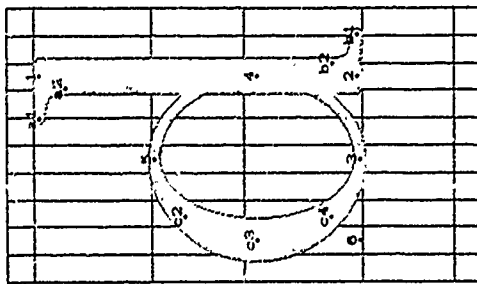
```

```

w1 draw 1;
hpen; r1, x1 = r1, x1; y1 = y1; x1 = x1 = 5(r + u); top, y1 = m + oo,
w1 draw 2(0, 1) 3(-1, 0);
if w1 > 1.5u lft, x1 = round(.75u);
else: x1 = good, 1.5u;
fi;
y1 = y1; bot, y1 = -oo,
call "a darc(3, 4, w1);
if w1 = w1: x1 = x1; x1 = x1 = x1 - x1; y1 = 5(y1, y1);
new aa; x1 = aa(x1, x1); y1 = (sqrt(1 - aa-aa))(y1, y1);
else: lft, x1 = r1, x1; y1 = .5e - 1; x1 = x1, y1 = e;
fi;
w1 draw 5(1, 0) 6( 7).

```





```

"The letter q",
call charbegin(4, 10, 0, sc, ph, 0, ph slant + 5pwt + (sc - 2)pt),
hpen; x1 = x2 = good1(r - 2.5u), top1y1 = h, bot1y2 = 0;
w1 draw 1..2,
if lcs > 0: call ~a serif(1, 1, 2, -lcs);
call ~b serif(2, 1, 1, lcs);
fi;
bot0y1 = -oo; top0y1 = m + oo, y1 = 5[y1, y1],
x1 = x2 = 3(r - u), x1 = x2,
w1 draw 3{1, 0}..4{0, 1}..5{-1, 0};
if w2 > 1.5u then x6 = round 75u;
else x6 = good1 1.5u;
fi;
y6 = y1, call ~c darc(5, 6, w2)

"The letter e",
call charbegin(5, 8, 0, 0, px, 0, pe slant + kuc + 23pt);
hpen; x1 = 5r, top1y1 = m + oo, y1 = c, y2 = 5[y1, y1],
if w2 > 1.5u then x2 = round 5u, rt1x1 = round(r - 5u);
else x2 = good1 1.25u, x1 = good1(r - 1.25u);
fi;
bot0y1 = -oo, x3 = x1 + 5u,
call ~a arc(3, 1, w1);
call ~b arc(3, 2, w2), call ~c arc(5, 2, w2);
new aa; y1 = y1 = aa[y1, y1]; x1 = 1 = (sqrt(1 - aa aa))(x1, x1);
w1 draw 4..1;
if w1 = w1: x0 = x1; x1 = x1 = x1, y1 = y1,
new aa; x0 = aa[x1, x1]; y0 = (sqrt(1 - aa aa))(y1, y1),
else, rt0x0 = rt1x1; x1 = x0, y0 = 5c - 1, y1 = c;
fi,
w1 draw 5{1, 0}..6{.7}.

"The letter f",
call charbegin(4, 6, 0, 0, ph, 0, ph slant + pt),
hpen; x1 = good1 2.5u,
if fix w1 lcs = 0: rt1x2 = round(r + 5u);
else rt1x2 = round(r - u);
fi,
open, top y2 = 8[m, h],
call ~a fstroke(2, 3);
open; top1y1 = m, y1 = y0, if 10x1 = lf1x1 - u - eps, rt1x1 + u + eps,
w1 draw 3..4

```

% stem
% upper serif
% lower serif

% right part of bowl

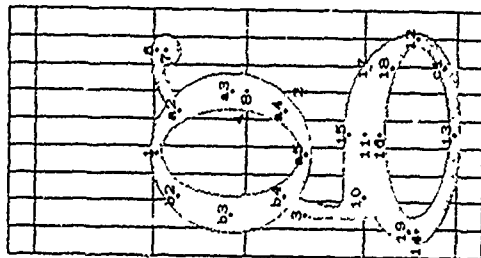
% left part of bowl

% bowl
% stroke

% bar

% point

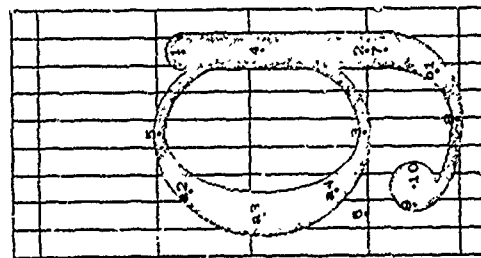
% bowl, shoulder, stem, and serif
% left part of bowl
% right part of bowl
% bar

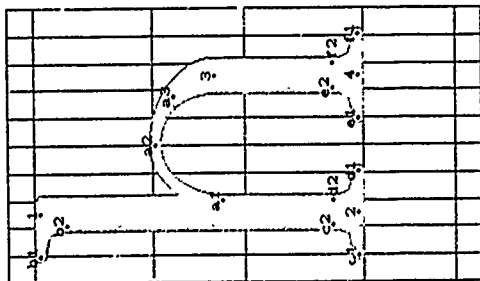


```

"The letter g";
if varg = 0;
    % the following program is for a 'classic' g shape
    call charbegin('g', 0, 0, 0, px, pd, px-slant + 5*px, -pu);
    lpen; x1 = good, 1.5u; x2 = good, 1.5r; x1 - x2 = x2 - x1;
    topdy = m + oo; y1 = y; y2 = round .5e;
    call 'a darc(1, 2, w); call 'a darc(1, 3, w);
    x1 = x2 - u; y1 = e;
    x2 = 1/sqrt(2)*x1; y2 = 1/sqrt(2)*x1 + 5*y1;
    x3 = x2 - 1.5u;
    new w; w1 = round .5(w, w);
    open; topdy = topdy + m + oo;
    w1 draw 7; w1 draw (e) 5 : {1, 0};
    ypen, x3 = x2; y3 = e;
    x1 = 1/sqrt(2)*x1; y1 = 1/sqrt(2)*x1 + 5*y1;
    x2 = x1; y2 = y1 = good, 0; x1 = x1 = 5r; x2 = good, (r - u);
    y2 = 5[botdy, topdy], botdy = -d - oo;
    draw (8..) w1, [w1] 10(1, 0) 11(1, 0);
    topdy = topdy; x1 = x1; botdy = botdy;
    x1 = 1/sqrt(2)*x1; y1 = 1/sqrt(2)*x1 + 5*y1;
    x2 = 1/sqrt(2)*x1; y2 = 1/sqrt(2)*x1 + 5*y1;
    lpen, w1 draw 15(1, 0) 17(x1 - x2, y2 - y1); 12(0, -1);
    16(1, 0) 18(x2 - x1, y2 - y1) 12(0, -1);
    call 'c arc(13, 12, w);
    new w; w1 = round .5(w, w);
    x1 = x1 = good, 0; y1 = 5[y1, y2], y2 = 5[y1, y2];
    w1 draw 13(-1, 0) 14(0, 1) 16(1, 0);
    13(-1, 0) 19(0, 1) 11(1, 0);
else;
    % the following program is for a 'simple' g shape
    call charbegin('g', 0, 0, 0, px, pd, 9*px-slant + 5*px, -pu);
    lpen; x1 = x2 = good, (r - 1.5u); x1 = x1; x2 = 5r;
    botdy = -oo; topdy = m + oo; y1 = 5[y1, y2];
    w1 draw 3(1, 0) 4(0, 1) 5(-1, 0);
    if w2 > 1.5u; lft, x1 = round 7.5u;
    else; x1 = good, 1.5u;
    fi;
    w1 = w1; call 'a darc(5, 6, w);
    open, topdy = 5[e, m]; y2 = e; w1 draw 1 2;
    lpen, x1 = x1; botdy = -2*d; w1 draw 2 7;
    x1 = good, 2.5u; x2 = 5[x1, x2], lft, x1;
    y1 = y1; botdy = -d - oo;
    open, botdy = -7.5d; w1 draw 10;
    lpen, w1 draw 9(0, -1) 8(1, 0); call 'b arc(8, 7, w);
    fi;

```





```

"The letter b";
call charbegin(1, 10, sc, 0, ph, 0, {pc, px} slant + .5pwi + (sc - 2)pu),
hpen; x1 = x2 = good1.25u; x3 = good1(r - 2.5u),
top1y1 = h; bot1y2 = 0;
w1 draw 1. 2,
call ~a hstroke(2, 3, 4);
if lcs ≠ 0: call ~b serif(1, 1, 2, -lcs);
call ~c serif(2, 1, 1, -lcs);
call ~d serif(2, 1, 1, lcs);
call ~e serif(4, 1, 3, -lcs);
call ~f serif(4, 1, 3, lcs);
fi

```

% left stem
 % shoulder and right stem
 % upper serif
 % lower left serif
 % lower right serif

```

"The letter v";
call charbegin(1, 5, sc, 0, ph, 0, ph slant + .5pwi + (sc - 2)pu),
hpen; x1 = x2 = good1.5r; top1y1 = m; bot1y2 = 0;
w1 draw 1. 2,
cpen; top3y1 = h; rt, x3 = rt, x1, w3 draw 3;
if lcs ≠ 0: call ~a serif(1, 1, 2, -lcs),
call ~b serif(2, 1, 1, -lcs),
call ~c serif(2, 1, 1, lcs),
fi.

```

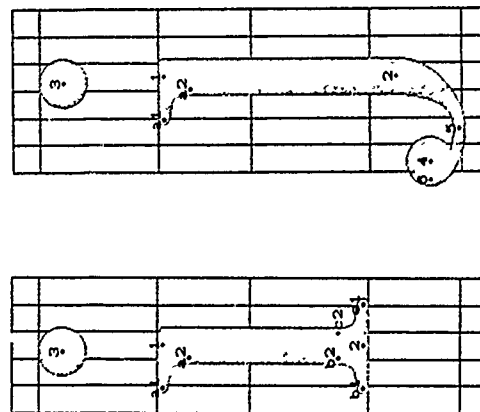
% stem
 % dot
 % upper serif
 % lower serif

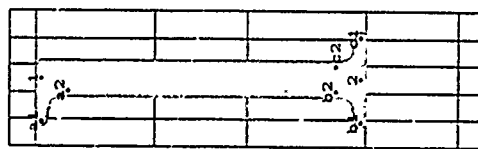
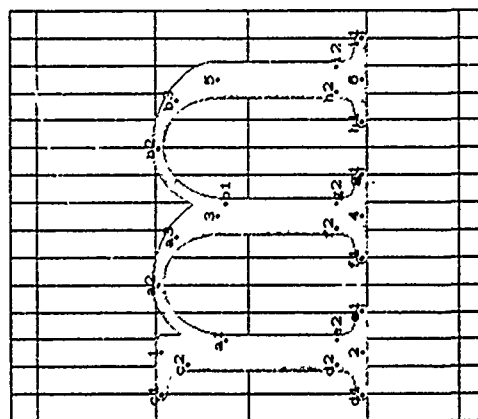
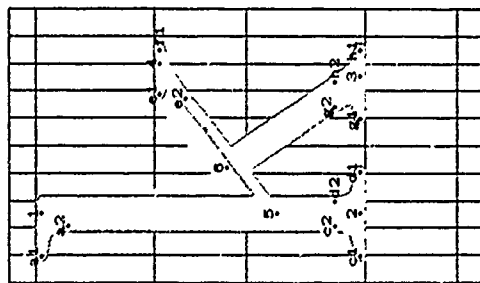
```

"The letter j";
call charbegin(1, 6, sc, 0, ph, pd, ph slant + .5pwi - 2pu)
hpen; x1 = x2 = good1(r - 2.5u);
if fixwidth = 0 lft, x1 = round(-.5u);
else lft, x1 = round u;
fi;
cpen; top3y1 = h; rt, x1 = rt, x1; w1 draw 3;
bot1y1 = -.9d; w3 draw 4;
hpen; top1y1 = m; bot1y2 = -.1d,
bot0y3 = -.d - .oo; y1 = h; lft, x1 = lft, x1; x3 = 5{c2, x0};
draw |w1| |w3| 2{0, -1} |w3| 5{-1, 0} . 6{0, 1};
if lcs ≠ 0: call ~a serif(1, 1, 2, -lcs),
fi.

```

% dot
 % bulb
 % stem and tail
 % serif





```

"The letter k":
call charbegin( k, 10, sc, sc, ph, 0, px slant + 5pwi + (lcs + sc - 2pu);
hpen, x1 = x2 = x3 = good, 2.5u,
x1 = good, (r - 2.5u); r1, x1 = r1, x1,
top1y1 = h; top1y1 = m, bot1y2 = bot1y3 = 0; y1 = .75e,
new aa, bb;
rt1, x1 = aa[rt1, x1, rt1, y1]; y1 = aa[bot1, m, y1];
rt1, x1 = bb[x1, x1]; y1 = bb[y1, y1];
w1 draw 6 3;
hpen#, w1 draw 4 .5;
w0 draw 4 .5;
if lcs ≠ 0: call ~a serif(1, 1, 2, -lcs);
call ~c serif(2, 1, 1, -lcs);
call ~d serif(2, 1, 1, lcs);
call ~e serif(4, 0, 5, -lcs);
call ~f serif(4, 0, 5, lcs);
call ~g serif(3, 1, 6, -lcs);
call ~h serif(3, 1, 6, lcs);
fi.

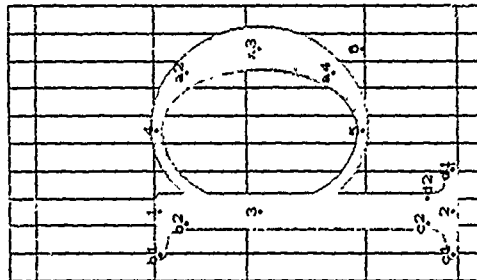
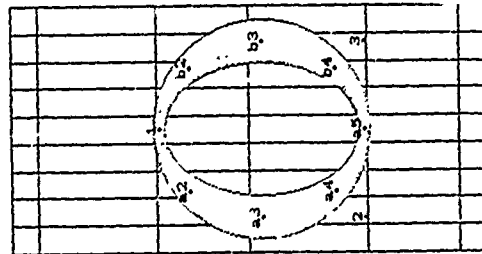
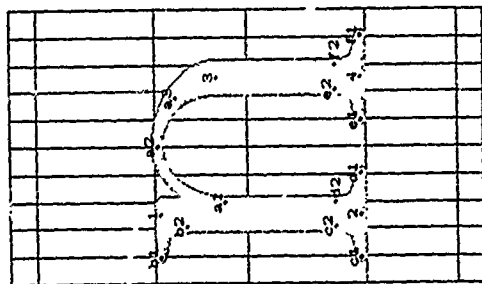
"The letter l":
call charbegin( l, 5, sc, sc, ph, 0, ph-slant + 5pwi + (sc - 2pu),
hpen, x1 = x2 = good, 5r; top1y1 = h, bot1y2 = 0,
w1 draw 1. 2;
if lcs ≠ 0: call ~a serif(1, 1, 2, -lcs);
call ~b serif(2, 1, 1, -lcs);
call ~c serif(2, 1, 1, lcs);
fi.

"The letter m":
call charbegin( m, 15, sc, sc, px, 0, [pr, px] slant + 5pwi + (sc - 2pu),
hpen, x1 = x2 = good, 2.5u, x1 = good, 5r, x1 = x1 = x1 - x1,
top1y1 = m, bot1y2 = 0;
w1 draw 1. 2;
call ~a hstroke(2, 3, 4);
if lcs ≠ 0: call ~c serif(1, 1, 2, -lcs);
call ~d serif(2, 1, 1, -lcs);
call ~e serif(2, 1, 1, lcs);
call ~f serif(4, 1, 3, -lcs);
call ~g serif(4, 1, 3, lcs);
call ~h serif(6, 1, 5, -lcs);
call ~i serif(6, 1, 5, lcs);
fi.

```

% lower diagonal
 % erase excess
 % stem
 % upper diagonal
 % upper stem serif
 % lower stem serif
 % upper diagonal serif
 % lower diagonal serif

% left stem
 % left shoulder and middle stem
 % right shoulder and right stem
 % upper serif
 % lower left serif
 % lower middle serif
 % lower right serif



```

"The letter n";
call charbegin( n, 10, sc, sc, px, 0, 1, pe, px, slant + 5pxw + (sc - 2)pu);
hpen, x1 = x2 = good, 2.5u, x1 = good, (r - 2.5u);
top, y1 = m; bot, y2 = 0;
w1 draw 1..2;
call a hstroke(2, 3, 4);
if les > 0: call b serif(1, 1, 2, -les);
    call c serif(2, 1, 1, -les);
    call d serif(2, 1, 1, les);
    call e serif(4, 1, 3, -les);
    call f serif(4, 1, 3, les);
fi.

"The letter o";
call charbegin( o, 9, 0, 0, px, 0, 5px, slant);
hpen, x1 = r - x1;
lft, x2 = round fixwidth( 5u, 1.5u);
x1 - x2 = x1 - x2, top, y1 = m + oo, bot, y2 = -oo, y1 = y1;
call a darc(1, 2, w);
call b darc(1, 3, w);

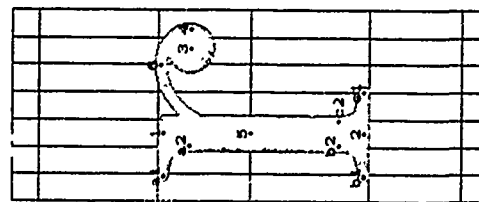
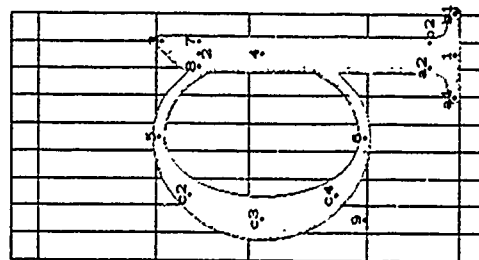
"The letter p";
call charbegin( p, 10, sc, 0, px, pd, 5px, slant + leu);
hpen, x1 = x2 = x3 = good, 2.5u, x1 = x3 = 5(r + u);
if w2 > 1.5u: rt, x4 = round(r - 75u);
else: x6 = good, (r - 1.5u);
fi,
top, y1 = m; bot, y2 = -d; top, y3 = m + oo, bot, y4 = -oo,
y1 = 5y1, y2 = y2, y3 = y3;
w1 draw 1..2;
w2 draw 5(-1, 0) .. 3(0, 1) 4(1, 0);
call a darc(4, 6, w);
if les > 0: call b serif(1, 1, 2, -les);
    call c serif(2, 1, 1, les);
    call d serif(2, 1, 1, les);

```

% left stem
 % shoulder and right stem
 % upper serif
 % lower left serif
 % lower right serif

% axis of left-right symmetry
 % left part of bowl
 % right part of bowl

% stem
 % left part of bowl
 % right part of bowl
 % upper serif
 % lower serif

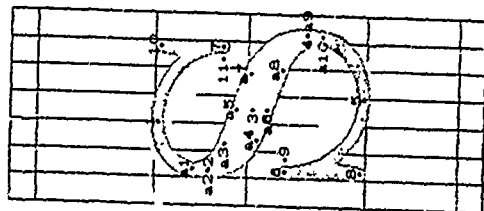


```

The letter q:
call charbegin( q, 10, 0, 1, px, pd, px slant + .5pxw - pu);
hpen; x1 = good( r - 2.5u); x2 = 5(r - u); z2 = x1 = x1, bot1y1 = -d;
rtot1 = rt1x1; x3 = x1; x5 = x1; lft1x2 = lft1x2;
top1y1 = m; y1 = 5(y1, u); top1y5 = m + oo; bot1y3 = -oo;
new aa; lft1x2 = aa(x1, x1);
y2 = y1 = y1 = (sqrt(1 - aa-aa))(y1, y1);
if lcs ≠ 0: call a serif(1, 1, 2, -lcs);
call b serif(1, 1, 2, lcs);
fi;
w1 draw 1..2;
w2 draw 7..3, 8{0, 1}..3{x1 - x1, 5(y1 - y1)};
w3 draw 6{1, 0}..4{0, 1}..5{-1, 0};
if w2 > 1.5u: lft1x2 = round 75u;
else x3 = good 1.5u;
fi;
y3 = y3; call c darc(5, 9, w2)

The letter r:
call charbegin( r, 7, sc, 0, px, 0, px slant);
hpen; x1 = x2 = good 2.5u; top1y1 = m; bot1y1 = 0;
w1 draw 1..2;
open; rt1x1 = rt1x1 = round(r - .5u); top1y1 = 9[e, m]; y1 = y1;
w1 draw 3;
hpen; x5 = x1; y5 = e; x4 = 5u; top1y3 = m + oo;
w2 draw 5{0, 1}..6{1, 0}..4{0, -1};
if lcs ≠ 0: call a serif(1, 1, 2, -lcs);
call b serif(2, 1, 1, -lcs);
call c serif(2, 1, 1, lcs);
fi.
% stem
% bulb
% shoulder
% upper serif
% lower serif
% left part of bowl
% right part of bowl
% lower serif
% stem
% bulb
% shoulder
% upper serif
% lower serif

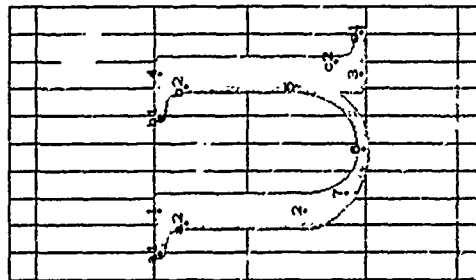
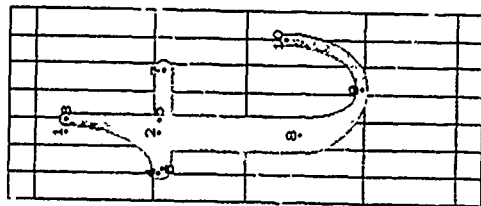
```



```

"The letter s",
call charbegin("s", 7, 0, 0, px, 0, px slant - .5pu);
hpen; topy1 = m + oo; boty1 = -oo;
x1 = .5r, (y1 - w1 - 5ux)/(m - 2w1 - w1) = e/m;
if lcs = 0: round u;
  x1 = x1 = x1; if 0.5u = round 75u, x1 = x1; rt1x1 = round(r - u);
  w1 = {top0, y1}, y1 = {bot0, y1},
  w1 draw 6{x1 - x0, 3(y1 - y0)} 5{1, 0};
  draw 7{x1 - x1, 2(y1 - y1)} 1{-1, 0};
  else if w1 = w1: x1 = x1 = x1; x1 = x1; rt1x1 = round(r - u);
  y1 = {top0, y1}; y1 = {bot0, y1};
  w1 draw 6{x1 - x0, 3(y1 - y0)} 5{1, 0};
  draw 7{x1 - x1, 2(y1 - y1)} 1{-1, 0};
  x1 = x1 = x1; y1 = y1 + lcs-aspect-u + eps, boty1 = 0,
  w1 draw 8..9;
  draw 10..11;
  else: x1 = x1 - .5u; x1 = x1 + 5u; x1 = x1; rt1x1 = round(r - u); rt1x1 =
  round(r - 5u);
  y1 = good0(y1) - 1, y1 = good0(y1, m) + 1,
  boty1 = 0; y1 = y1, x1 = x1, rt1x1 = rt1x1,
  topy1 = m, y1 = y1; x1 = x1; rt1x1 = rt1x1,
  w1 draw 6..8, 9..10;
  draw 7..10, 11..10;
  rpen; w1 draw 6{0, -1} 5{1, 0};
  lpen; w1 draw 7{0, 1} 1{-1, 0};
  hpen; w1 draw 6{0, -1} 5{1, 0},
  draw 7{0, 1} 1{-1, 0},
  fi,
fi;
new aa;
if m - e > e: aa = m - e,
else: aa = e,
fi,
call "a sdraw(1, 2, 3, 4, 5, w1, v1, -aa/(12u)).
% middle stroke e

```



```

"The letter v":
if px - pe < .75(ph - px): call charbegin( t, 7, .4, 0, 2px - pe, 0, px slant + .5pwi - 2pu);
y1 = 2m - e;
else: call charbegin( t, 7, 0, 0, .75(px, ph), 0, px slant + 5pwi - 2pu);
y1 = 75[m, h];

fi;
hpen, x1 = x2 = good, 2.5u; top0y2 = m;
if w1 = w1: w1 draw 1..2;
else: rt1x1 = rt0x2; lt0x1 = lt1x1 - u - eps; y1 = y2 = y2; y1 = y1; x2 = x1;
w1 ddraw 4{1, 0}..3{0, 1}, 4..5;
% left and upper terminals

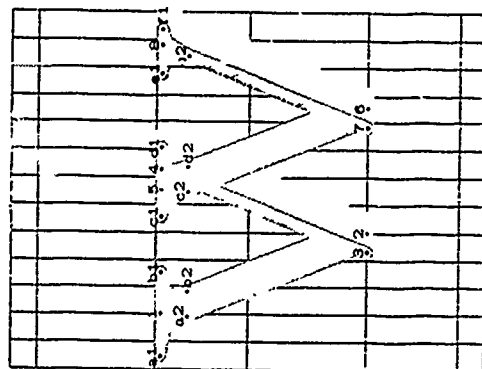
fi;
lt10x1 = lt1x1 - u - eps, rt10x2 = rt1x1 + 2u + eps;
open; top10y1 = m; y1 = y1, w1 draw 6..7;
hpen; x2 = x1; bot0y1 = .5e; w1 draw 2..8;
if w0 > 1.5u: bot0y1 = 0; x0 = r - 2.5u, rt0x10 = r - 5u, x11 = r + u, y11 = e;
new as; x10 = ad[x0, x1]; y10 = (sqrt(1 - ad ad))[y1, y1];
draw w1#8{0, --1}..w1#9{1, 0}..10{1, 1};
else: bot0y1 = -oo; y10 = {e; x1 = 5[cs, x10], x10 = good(r - u);
draw w1#8{0, --1}..w1#9{1, 0}..10{0, 1};
% hook

fi.

"The letter u":
call charbegin( u, 10, se, se, px, 0, px slant + 5pwi + (se - 2)pu);
hpen, x1 = x2 = good, 2.5u, x1 = x1 = good(r - 2.5u), lt0x1 = lt1x1,
top1y1 = m, m - y2 = {e, m}; m - y2 = {e, m}, % prepare for upside down stroke
x0 = 5[x2, x1]; bot0y1 = -oo;
x1 = 1/sqrt(2)[x0, x1]; y1 = 1/sqrt(2)[y2, y1];
draw w1#5{0, --1}..w1#6{-1, 0}..6[w1, w1]7{x1 - 2u, y2 - y1}
w1#12{0, 1}..1;
y1 = y1; bot1y1 = 0; w1 draw 3..4;
if les < 0: call a serif(1, 1, 2, --les);
call b serif(4, 1, 3, --les);
call c serif(3, 1, 4, les);
% stroke
% stem
% upper left serif
% upper right serif
% lower serif

fi.

```



```

"the letter w";
call charbegun(*w,l3,sc,sc,px,0,px slant + 5pw + (lc + 1 5)pw),
  lpen, x1 = good, 2u; r1(x) = r1(x)w, x1 = x1 = x1 - x1, x1 = x1 + (lc - lcs - 2)u,
  % v0, x1w, and x1 are approximations to x1, x2, and x0.
  x1 - x1 = x0 - x2 = x1 - x1 = x1 - x1 round(x1 - x1),
  % The idea is to draw two v's displaced by an integer amount
  top1y = m, bot1y = 0, y1 = y1 - y1, y1 = y1 - y1 - y1,
  r1(x) = r1(x); l1(x) = l1(x), x2 - x1 = x1 - x1,
  w1 draw 1, 2;
  rpen#, w1 draw 3, 4;
  lpen; w0 draw 3, 4;
  w; draw 5, 6;
  rpen#; w1 draw 7, 8;
  lpen, w0 draw 7, 8;
  if it is not 0; call *o serif(1, 1, 2, -lc);
  call *o serif(1, 1, 2, lcs);
  call *e serif(5, 1, 6, -lcs);
  call *d serif(5, 1, 6, lcs);
  call *o serif(8, 0, 7, lcs);
  call *f serif(8, 0, 7, lcs);

```

```

"The letter x":
call charbeg("x", 10, sc, sc, px, 0, px slant + .5pw + (sc + lcs - 2)pu),
hpen; x1 = x1 = good(2.5 - lcs + less)u, x2 = x1 = r - x1,
top, y1 = top, y2 = m; bot, y1 = bot, y1 = 0,
w1 draw 1..4;
w1 draw 3..2;
if lcs < 0, call "a serif(1, 1, 4, -lcs);
call "b serif(1, 1, 1, 2[less, lcs]);
call "c serif(4, 1, 1, -2[less, lcs]);
call "d serif(4, 1, 1, lcs);
new ss; ss = .5(w1 - w2)/w1;
call "e serif(2, 0, 3, -2[less, lcs] - ss);
call "f serif(4, 0, 3, lcs + ss);
call "g serif(3, 0, 2, -lcs - ss);
call "h serif(3, 0, 2, 2[less, lcs] + ss);
fi.

```

```

% upper left to lower right diagonal
% lower left to upper right diagonal
% upper left serif
% lower right serif
% correction to lcs makes w0 like w1
% upper right serif
% lower left serif

```

```

"The letter y":
call charbeg("y", 10, sc, sc, px, pd, px slant + .5pw + (lcs + sc - 1.5)pu),
hpen; x1 = good(2u, x1 = good(1 - lcs - 2)u);
x2 = x1 = x1 = x1; y1 = y1; y2 = y1; y1 = y1,
top, y1 = m; y1 = y1; bot, y2 = -o; y1 = y1,
w1 draw 1..2,
w1 draw 3..4;
open; lcs = lcs; w1 draw 2.5u;
bot, y1 = -1.5d; y1 = y1; w1 draw 8;
hpen; x1 = 2u; bot, y1 = -d - oo, bot, y1 = -5d;
new aa; x1 = aa[x1, x1]; y1 = aa[y1, y1];
w1 draw 4..3{x1 = x1, y1 = y1} . 6{-1, 0} . 7{0, 1};
if lcs < 0, call "a serif(1, 1, 2, -lcs);
call "b serif(1, 1, 2, lcs);
call "c serif(4, 0, 3, -lcs);
call "d serif(4, 0, 3, lcs);
fi.

```

```

% left diagonal stroke
% erase excess at lower right
% bulb
% right diagonal and tail
% left serif
% right serif

```

```

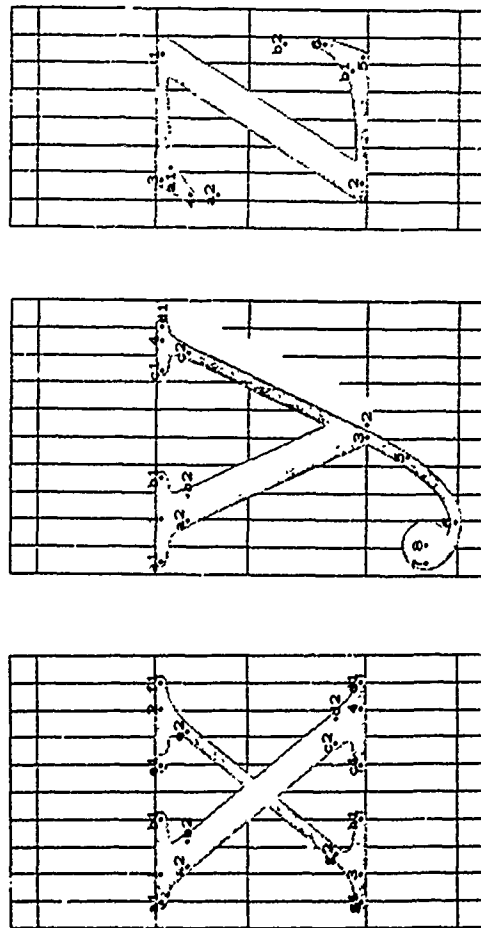
"The letter z":
call charbeg("z", 8, 0, 0, px, 0, px slant - .5pu);
hpen; lcs = lcs; w1 draw 1u, r1, x1 = round(1 - u),
top, y1 = m, bot, y2 = 0,
new ss; ss = 1.4aspect lcs u + eps;
if ss + w1 > 25m, new ss, ss = 25m - w1 + eps;
fi;
lcs = lcs; w1 draw 1.5u, x1 = x1 - 5u; y1 = y1 - ss/1.4;
r1, y1 = round(1 - 1.5u), x1 = x1 + .5u, y1 = y1 + ss;
call "a arm(1, 3, 4);
w1 draw 1..2;
call "b arm(2, 5, 6);

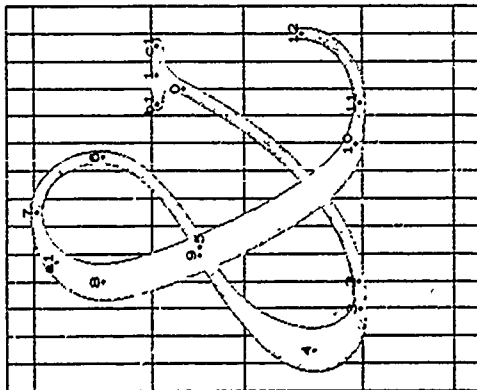
```

```

% upper arm and serif
% diagonal stroke
% lower arm and serif

```





The file romand mf

% This file contains the ten digits, as well as '&' and '?',
 % in the so-called roman style.

% Codes '046, '060-'07i, and '077 are used

"Amperсанд";

call charbegin(0.46, 1.4, 0, 0, ph, 0, 5pc-slant + .5pw -- 5pu),

hpen; x1 = r - 2.5u, topy1 = m;

x2 = 4u; y2 = .015[y1, e]; x3 = 3u, boty3 = -oo;

x4 = good, 1.5u; y4 = 1[y1, y1]; x5 = 5.25u, y5 = 5[y1, y1];

x6 = good, 8.5u; y6 = 8[y1, y1]; x7 = 6.5u, topy7 = h + oo;

x8 = good, 4u, y8 = 4u; x9 = x4 + u; y9 = y1; x10 = r - 5u, y10 = 0.15[y1, y1];

x11 = x10 + 1.5u; y11 = y1, x12 = good(r - e); y12 = 5[y1, e];

if s = 0: y0 = 1[y1, y2];

else: y0 = y1 - s;

fi;

(x1 -- x0)/(y1 -- y0) == 5(x1 -- x0)/(y1 -- y2),

w0 draw 1, 0;

draw 0(x0 -- x1, y0 -- y1) |w0#2 - 1|{u0, w0}[3{-1, 0} |w2#4(0, 1}

|w0#5(x1 -- x2, y1 -- y2) |y1, y1}

6{0, 1} . 7{-1, 0}.

call 'a arc(7.8, w0);

draw |w0|8{0, --1}..9 |w0#10 - |w0#11{1, 0} . 12{0, 1},

if lcs != 0: call 'b serif(1, 0, 0, -lcs);

call 'c serif(1, 0, 0, lcs),

fi.

"The numeral 0";

call charbegin(0.9, 0, 0, ph, 0, ph-slant -- .5pu);

if fixwidth = 0: new save, save = sqrtwo, new sqrtwo,

sqrtwo = sqrt(1.23114413save),

fi;

hpen;

if w0 > 1.5e: lt(x1, x2) = round.75u;

else: x2 = good, 1.5u,

fi,

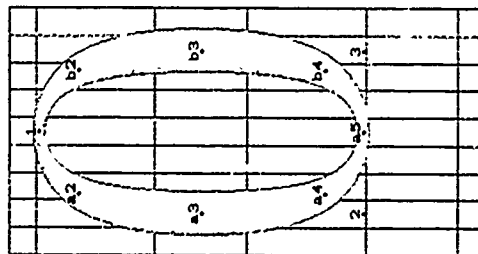
x1 = r - x1;

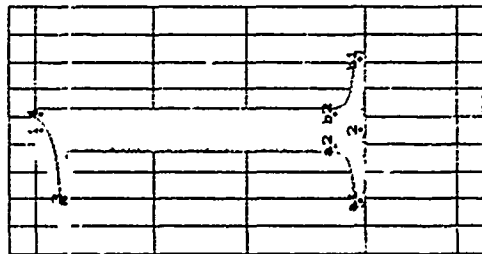
x3 = r - x2, topy3 = h + oo, boty3 = -oo, y3 = y1;

call 'a draw(1, 2, w2); call 'b draw(1.3, 2),

if fixwidth = 0: new sqrtwo, sqrtwo = save;

fi.





```

The numeral "1",
call charbegin(1, 0, 0, ph, 0, ph-slant - 5pt);
hpen; x1 = x2 = good2.5r; top2y = h, bot2y = 0,
w2 draw 1..2;
call a serif(2, 2, 1, -2);
if w2 = w1 if les = 0: top2y = 8[m, h], x1 = x1 - 2u - eps,
else: top2y = 2[m, h], x1 = 1t2x1 - 2u - eps,
fi;
else: top2y = 8[m, h]; x1 = x1 - 2.5u - eps,
fi;
y1 = y1; r1x1 = r1x1, y1 = 1.5[m, h], x1 = x1,
hpen; w2 draw (5..4) 3(-1, 0);
hpen; w3 draw (5..4) 3(-1, 0);

```

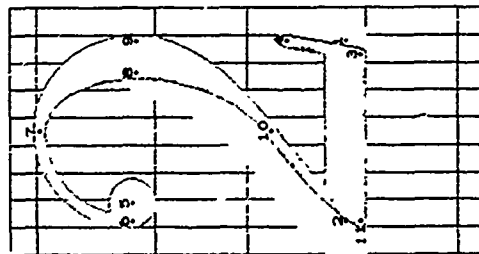
% stem
% serif

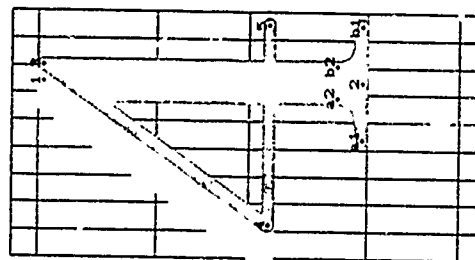
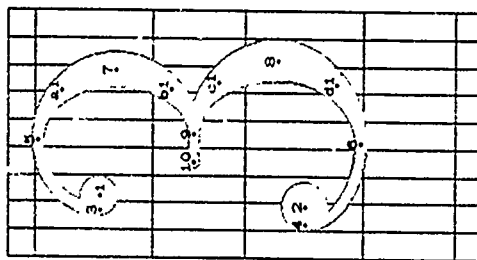
```

The numeral "2",
call charbegin(2, 0, 0, ph, 0, ph-slant - 5pt);
hpen; new w1;
if les = 0: w1 = w1;
else: w1 = w1;
fi;
r1x1 = round(r - u); 1t2x1 = round u, bot2y = 0; y1 = y1;
w1 draw 1..2;
if les < 0: x1 = x1, top2y = top2y + ucs aspect u + eps, bot2y = 0;
if w1 = w2 x1 = x1;
else: x1 = x1 + .5u = x1, hpen; w1 + 5u draw 3..4,
fi;
hpen; w3 draw 3..4;
fi;
hpen; bot2y = m; 1t2x1 = round u;
w1 draw 5;
hpen; 1t2x1 = 1t2x1, y1 = y1,
r1x1 = round(r - u), 1t2x1 = 1t2x1, x1 = 3[4u, x1], x1 = x1,
x1 = x1, top2y = h + .5u, y1 = y1, bot2y = 0; y1 = 4[y1, y1],
w1 draw 6(0, 1) 1(1, 0);
hpen; 2u draw 9(0, -1) 10(x1 - x2, 5(y1 - y2))
11(x1 - x2, 2(y1 - y2));
hpen, w1 draw 7(1, 0) 8(0, -1) 10(x1 - x2, 5(y1 - y2)),
7(1, 0) 9(0, -1) 10(x1 - x2, 5(y1 - y2)),
draw 10(x1 - x2, 5(y1 - y2)) 11(x1 - x2, 2(y1 - y2))

```

% bar
% serif excess
% point
% serif
% ball
% shoulder
% serif excess
% stroke
% link





```

"The numeral 3",
call charbegin(-3, 9, 0, 0, ph, 0, ph slant - 5pu);
new w0, w1, w2 = round 75[w0, w1],
open; bot y0 = 75h,
if tot y0 > 9h new y1; top y1 = 9h,
fi;
lt y1 = round 1.5u, w1 draw 1,
top y1 = 25h;
if bot y1 < 1h new y2; bot y2 = 1h,
fi;
lt y2 = round u; w1 draw 2,
open; lt y1 = lt y2, lt y1 = y1, y1 = y1,
x1 = 5[x1, x1], x1 = 5[x1, x1], top y1 = h + oo, bot y1 = oo,
rt y1 = round(r - 1.5u), rt y1 = round(r - 1), y1 = 5[y1, y1], is
y1 = y1 = good, 52h; x1 = x1 - 5r,
w1 draw 3(0, 1) 5(1, 0);
call "a arc(9, 8, w1), call "b arc(9, 7, w1);
w1 draw 9, 10;
call "c arc(9, 8, w2), call "d arc(6, 8, w2),
w1 draw 6(-1, 0) 4(0, 1).

```

% upper bulb

% lower bulb

% shoulder

% upper bowl

% bar

% lower bowl

% tail

```

"The numeral 4",
call charbegin(-4, 9, 0, 0, ph, 0, ph slant - 5pu),
open, x1 = x2,
if les < 0 if w2 > 1.5u rt y1 = round(r - 1.75u),
else x1 = good, 6.5u,
fi;
else x1 = good, 6.5u;
fi;
top y1 = h, bot y1 = 0,
rt y1 = rt y1, y1 = y1, y1 = y1, good y1, lt y1 = round u,
w1 draw 1 2,
open, w2 draw 2 4;
if les < 0; rt y1 = rt y1 + 1.5u + eps,
call "a serif(2, 2, 1, 1.5),
call "b serif(2, 2, 1, 1.5),
else rt y1 = round(r - 7u),
fi;
w1 draw 4 5

```

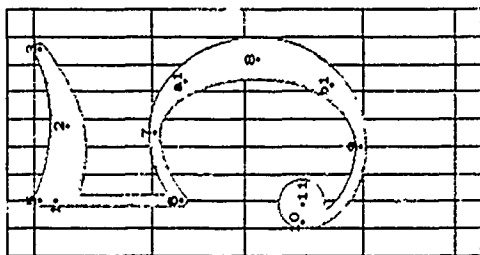
% stem

% excess at upper left

% diagonal

% serif

% bar

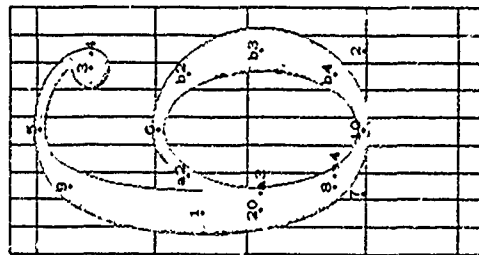


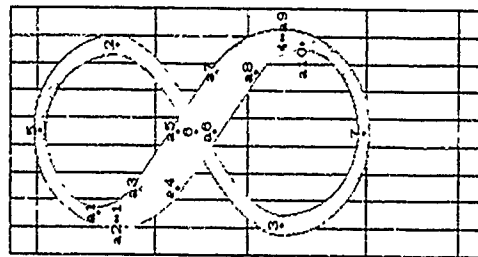
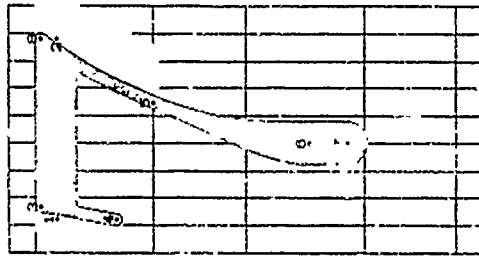
"The numeral 5";
 call charbegin(5, 0, 0, ph, 0, ph, slant - 5pu);
 vpen, $x_1 = \text{good}_1 2u$, $\text{top}_1 y_1 = h$; $\text{rt}_1 x_1 = \text{round}(r - 1.25u)$; $\text{top}_1 y_1 = h$,
 $x_2 = 5[x_1, x_1]$; new w_1 , $w_1 = \text{round } 75[u, u]$; $\text{top}_1 y_1 = \text{round } .95h$,
 $x_3 = -5u$, $x_4 = r + 1.5u$, $y_3 = y_4 = 1.5h$,
 draw $([u, 0] \dots [1.5u, 2.5u])$; $\text{top}_1 y_1 = 1.5h$,
 lpen; $x_5 = x_6 = x_1$, $\text{top}_1 y_1 = h$; $\text{top}_1 y_1 = 75[e, m]$,
 $x_7 = .5r$; $\text{top}_1 y_1 = m + .00$; $x_8 = x_7 - 5u$, $\text{bot}_1 y_1 = -.00$,
 $\text{rt}_1 x_1 = \text{round}(r - u)$; $y_1 = .5[y_1, y_1]$,
 w_1 draw 5, 6;
 draw $(9, .06, 7\{1, 0\})$,
 call a arc(7, 8, w_1); call b arc(9, 8, w_1);
 $\text{rt}_1 x_1 = \text{rt}_1 x_1$; $\text{top}_1 y_1 = \text{round } u$, $y_1 = y_1 + 1.5u$,
 w_1 draw 9(-1, 0) 10(0, 1);
 open; w_1 draw 11

% arm
 % stem
 % left part of bowl
 % right part of bowl
 % tail
 % bulb

"The numeral 6";
 call charbegin(6, 0, 0, ph, 0, ph, slant - 5pu),
 if $w_1 > 1.5u$; $\text{rt}_1 x_1 = \text{round } 75u$, $\text{rt}_1 x_2 = \text{round}(r - 75u)$;
 else $x_1 = \text{good}_1 1.5u$, $x_2 = \text{good}_2(r - 1.5u)$,
 fi,
 new w_1 ; $w_1 = \text{round } 75[u, w_1]$,
 open, $\text{top}_1 y_1 = h - .25e$,
 if $y_1 < 5[m, h]$ new y_1 , $y_1 = 5[m, h]$;
 fi,
 $\text{rt}_1 x_1 = \text{rt}_1 x_1$; $\text{round}(r - 1.5u)$; $y_1 = y_1$,
 w_1 draw 3;
 lpen; $x_3 = \text{good}_1(r + 1u)$, $x_4 = x_3 = x_1 = 5[x_3, x_2]$, $\text{top}_1 y_1 = h + .00$,
 w_1 draw 4(0, 1) 5(-1, 0);
 $\text{bot}_1 y_1 = -.00$; $\text{top}_1 y_1 = m + .00$, $y_2 = 5[y_2, y_1]$,
 $y_1 = y_2 = y_1$, $\text{rt}_1 x_1 = \text{rt}_1 x_1$;
 call a darc(6, 7, w_1), call b darc(6, 2, w_1),
 new w_1 ; $w_1 = \{[u, w_1]$;
 $x_5 = x_1$; $\text{rt}_1 x_1 = \text{rt}_1(1/\sqrt{2} \text{two}[x_5, x_1])$;
 $y_5 = 1/\sqrt{2} \text{two}[y_5, y_1]$; $y_5 - y_1 = y_5 - y_1$, $y_1 = 5[y_1, y_1]$;
 draw $[u, 5] 5(-1, 0) \dots [u, 9] 9(x_1 - x_5, y_1 - y_5) \dots [u, 1] 1(0, -1)$

% bulb
 % shoulder
 % bowl
 % intermediate width used in darc routine
 % roke



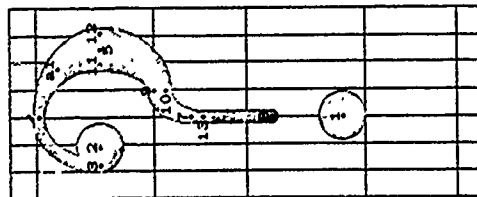
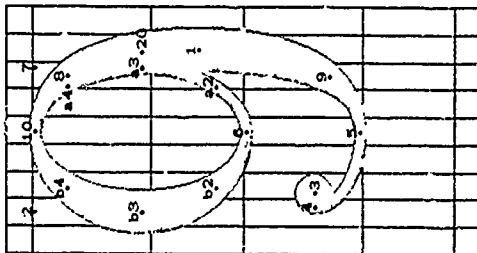


```

The numeral 7";
call charbegin(7, 0, 0, ph, 0, ph slant - 5pt);
vpen, new w0,
if lcs = 0 w0 = w1,
else w0 = w8,
fi;
ltopx1 = round u, rtopx2 = round(r - u), topy0 = h, y2 = y1,
w0 draw 1 2;
if lcs < 0 x1 = x1, boty0 = boty0 - ucs aspect u = eps, topy2 = h,
if w1 = w8 x1 = x1;
else x1 = x1, lpen#, w0 + 5u draw 3 4,
fi,
hpen, w0 draw 3 4;
fi;
open; boty2 = -oo, boty0 = l m, x0 = x - good2 4u;
w2 draw 7 6;
hpen; x2 = 5 5u, y2 = m, topy2 = h, x2 = x2,
rpen#, 2u draw 8{x2 - x0, y2 - y0} 5( 6),
hpen, draw boty0{x2 - x0, y2 - y0} 5 boty2{0, -1}

"The numeral 8";
call charbegin(8, 0, 0, ph, 0, ph slant - 5pt);
new w0, w0, ss, w0 = round 1{w0, w0};
hpen; ltopx1 = round u, x2 = r - x1, y2 = y1,
ltopx1 = round 7u, x1 = r - x0, y1 = y0, x1 = r - x2, x2 = x0,
topy2 = h + oo; y2 = 52u, boty2 = -oo,
w0 = 2{w1, w0};
if w0 = w0 ss = 0,
else ss = h/(18u),
fi;
call a draw(5, 1, 6, 4, 7, w0, w0, -ss),
w0 draw 5{1, 2} 2{0, -1} 6{-1, -75ss}
3{0, -1} 7{1, 0}

```



```

"The numeral 9";
call charbegin(9, 9, 0, 0, ph, 0, ph, slant - .5pu);
if w2 > 1.5u: r1x1 = round(r - 75u); lft, x2 = round 75u,
else: x1 = good2(r - 1.5u); x2 = good2(1.5u,
fi;
new w9; w9 = round .75(w3, w1);
open, bot9y1 = .25e;
if y1 > 5e: new y1, y1 = .5e;
fi;
lft9x1 = lft0x1 = round 1.5u; y1 = y1;
w9 draw 3;
lpen; x0 = good2(x1 - 1u); x1 = x0 = x0 = 5[x20, x1]; bot0y1 = - oo,
w9 draw 4(0, -1) .. 5(1, 0);
top0y1 = h + oo; y0 = e - oo; y0 = 5[y2, y0],
y1 = y10 = y2, lft0x1 = lft2x0,
call a dare(6, 7, w9); call b dare(6, 2, w2),
new w9; w9 = 3(w9, w1);
x0 = x1, lft0x1 = lft0(1/sqrt(2)(x0, x1));
y0 = 1/sqrt(2)(y0, y1); y1 = y1 = y1 = 5[y2, y0],
draw lft0(5(1, 0) {w9} 9{x1 - x0, y1 - y0} {w2} 1(0, 1)
, {w9} 8{x0 - x1, y1 - y0} {w2} 1(0, -1, 0),
% bulb
% tail
% bowl
% intermediate width in dare routine
% stroke

"Question mark";
call charbegin(077, 7, 0, 0, ph, 0, 8ph, slant + 5pw1 - pu),
new w9, w9, w9 = round .4(w9, w1);
if w1 < w9 sqrt 2: w9 = w9 sqrt 2,
else: w9 = w9,
fi;
open, bot9y1 = 0; x1 = good0(5(r - u));
w9 draw 1;
lft, x2 = lft0x1 = round u, y2 = y1 = 8[top0y1, bot0y1]; top0y1 = h + oo,
w1 draw 2;
lpen, x1 = .5(r - u); x2 = good2(r - 1.5u), y1 = y1;
w1 draw 3(0, 1) .. 4(1, 0); call a arc(4, 5, w2);
lft0x1 = lft2x1, r0x12 = r2x1; y1 = y12 = y1;
open; top0y1 = top0y1; bot0y1 = bot0y1,
x1 = x10 = x1; x2 = x2 = x1 = x1, y1 = y1 = y1 = y10 - y1,
y1 = y1 = y1; top0y1 = top0y1 {y0, y1}, bot0y1 = 2{top0y1, m} + 1,
lpen; w9 draw 4(0, -1) .. 9(1, 0) 7(0, -1),
12(0, -1) 1 -1, 0 13(0, -1);
draw 13..8.
% dot
% bulb
% bowl
% link and stem
% point

```

The file romans mf

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

"Dotless letter j";
 call charbegin('013, 3, sc, 0, px, 0, px slant + 5pxw + (sc - 2)pu),
 hpen; $x_1 = x_2 = \text{good}_1 \cdot 5r$; $\text{top}_1 y_1 = m$; $\text{bot}_1 y_1 = 0$;
 $w_1 \text{ draw } 1, 2$;
 if $\text{les} \neq 0$: call 'a serif(1, 1, 2, -les);
 call 'b serif(2, 1, 1, -les);
 call 'c serif(2, 1, 1, les);

% stem
 % upper serif
 % lower serif

fi.

"Dotless letter j";
 call charbegin('014, 6, sc, 0, px, 0, px, pd, px slant + 5pxw + 2pu),
 hpen; $x_1 = x_2 = \text{good}_1(r - 2.5u)$;
 if $\text{fixwidth} = 0$ $\text{fix}_1 r_1 = \text{round}(-5u)$;
 else $\text{fix}_1 x_1 = \text{round } u$;

% bulb

open; $\text{bot}_1 y_1 = -9d$; $w_1 \text{ draw } 4$;
 hpen; $\text{top}_1 y_1 = m$; $\text{bot}_1 y_2 = -1d$;
 $\text{bot}_1 y_1 = -d - oo$; $y_1 = u$; $\text{fix}_1 x_1 = 5[x_2, x_1]$;
 $\text{draw } [w_1] 1$ $[w_1] \# [2(0, -1)]$ $[w_1] \# [5(-1, 0)]$ $6(0, 1)$;
 if $\text{les} \neq 0$: call 'a serif(1, 1, 2, -les);

% stem and tail
 % serif

fi.

"The German letter ss";

call charbegin('033, 10, sc, 0, ph, 0, ph slant - pu),
 hpen; $x_1 = x_2 = \text{good}_1 2.5u$; $\text{rt}_1 x_1 = \text{round}(r - 1.5u)$; $\text{rt}_2 x_1 = \text{round}(r - 5u)$;
 $\text{bot}_1 y_1 = 0$; $y_1 = -5[m, y_1]$;
 $\text{top}_1 y_1 = h + oo$; $x_1 = 5[x_2, x_1]$; $y_1 = 5[m, y_1]$; $y_1 = 4m$;
 $w_1 \text{ draw } 1, 2$;

% stem
 % shoulder

call 'a arc(5, 2, w_1);

$\text{top}_1 y_1 = m$; $\text{fix}_1 x_1 = \text{round}(\text{rt}_1 x_2 + 1.5u)$;

call 'b arc(5, 3, w_1); call 'c arc(6, 3, w_1);

open; $\text{fix}_1 x_1 = \text{round}(\text{rt}_1 x_1 + 5u)$; $\text{bot}_1 y_1 = 1e$;

$w_1 \text{ draw } 8$;

hpen; $\text{fix}_1 x_1 = \text{fix}_1 x_2$; $y_1 = y_2$; $x_1 = \text{fix}_1 x_2$; $\text{bot}_1 y_1 = m$;

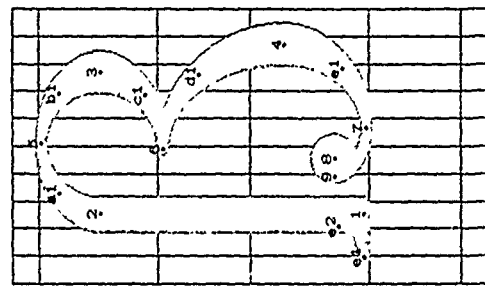
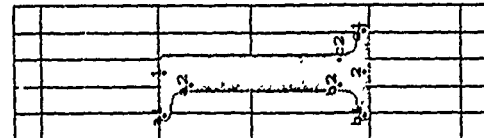
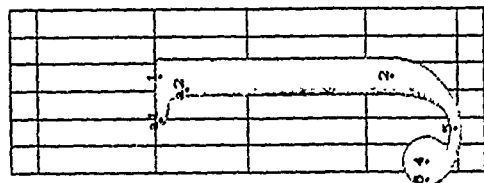
call 'd arc(6, 4, w_1); call 'e arc(7, 4, w_1);

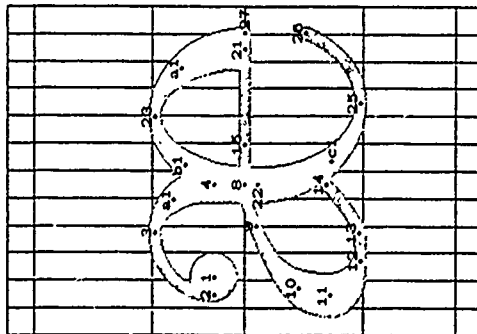
$w_1 \text{ draw } 7(-1, 0)$ $9(0, 1)$;

if $\text{les} \neq 0$: call 'e serif(1, 1, 2, -les);

fi.

% upper bowl
 % bulb
 % lower bowl
 % link
 % serif





```

The ligature 'ac';
call charbegin(034, 12, 0, 0, px, 0, pe-slant + leic);
open; lft, x1 = round 1.25u;
if top, (top, top, c + 2) > .9[e, m]. top, y1 = .9[c, m];
else. y1 = top, top, c + 2;
fi;

w, draw 1;
hpen; lft, x2 = lft, x1; y2 = y1; x1 = 3.75u; top, y2 = m + oo;
x1 = good, 5.5u; y1 = {e, m};
w, draw 2{0, 1}..3{1, 0}; call 'a arc(3.4, w);
x3 = x1, y3 = c, w, draw 4 8;
x1 = 4u; y1 = .9[u, y3]; x10 = x1 + 25u, y10 = .5[u, y3];
x11 = good, 1.5u, y11 = 2[u, y3];
x12 = 2.75u; bot, y12 = -oo; x13 = 3.75u; y13 = .015[y1, y3];
x14 = x3, y14 = 3[y12, y3]; x15 = 7u; y15 = c;
draw [w, w2]12{1, 0} [w, w2]13 14(-.15);
if w2 > 1.5u: r1, x21 = round(r - 7.5u);
else x21 = good, (r - 1.5u);
fi;

x22 = x1, x23 = r - 4u, top, y23 = m + oo; y24 = c, y25 = 5[y2, y1];
bot, y25 = -oo, x25 = x23 + .5u;
call 'a arc(23, 21, w2);
call 'b arc(23, 22, w1); call 'c arc(25, 22, w1);
w, draw 8..21;
if w1 = w1: x26 = x21; x27 = x23 - x24; y27 = y22;
new aa, x28 = aa[x25, x27], y28 = (sqrt(1 - aa aa))[y27, y25];
else r1, x28 = r1, x21; x27 = x26, y28 = 5c - 1, y27 = c;
fi;
w, draw 25{1, 0} 26{..27}

```

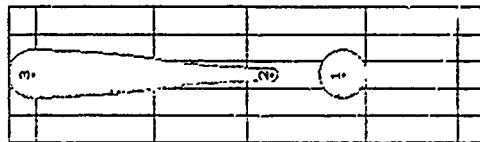
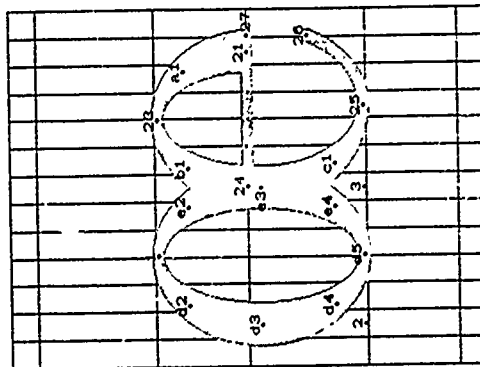
% bulb

% shoulder
% stem

% left bowl

% night bowl
% stroke
% bar

% point



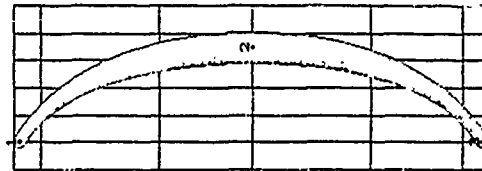
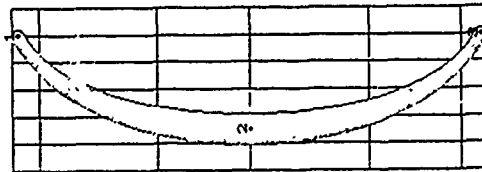
```

"The ligature oe"
call charbegin(035,13,0,0,px,0,pe,slant+leic);
hpen, x1=good,5r;
if w2>1.5u: fl,x2=round.75u: rt,x21=round(r-75u);
else: x2=good,1.5u: x21=good,(r-1.5u);
fi;
x1-x2=x1-r;
top,y1=m+e; bot,y2=-oo; y2=y1;
call ~d dare(1,2,w2);
call ~e dare(1,3,w2);
x22=x1; x23=r-4u; top,y21=m+oo; y21=e; u;
bot,y25=-oo; x25=x21+5u;
call ~a arc(23,21,w2);
call ~b arc(23,22,w2);
new aa, y21=y1; ad[y22,y21]; x21=1=(sqrt(1-aa aa))[r,x21];
u; draw 24 21;
if u0=w: x26=x21; x27=x25; x28=x21; y27=y21;
new aa, x26=ad[x21,x21]; y26=(sqrt(1-aa aa))[y27,y21];
else: r0,x26=r,x21; x27=x26; y26=x26-1; y27=e;
..b draw 25{1,0} 26{ 27)
% point

The file rom1tp mf

% This file contains punctuation marks common to roman and
% italic styles, including math italic and fixed-width fonts
% Codes '041, '047 '054, '056, '057, '072 '076, '133, '135, '140 are used
"Exclamation point";
call ~barbegin(041,5,0,px,0,(ph+pb) slant : 5pwm 2pu);
new wpp;
if w3<w0sqrt 2: wpp=round w0sqrt 2;
else: wpp=w3;
fi;
cpen; x1=x2=good,2.5u; bot,y1=-oo; m; draw 1,
top,y1=h+b; bot,y2=-25[top,y1+1 m];
wpp draw 3,
hpen, draw [wpp]3 [wpp]2
% dot
% top of stem
% stem

```

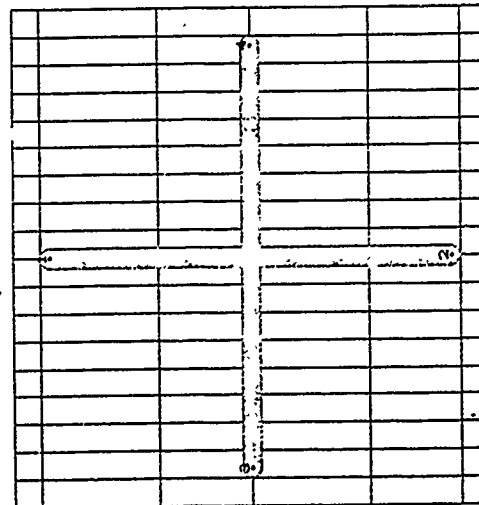
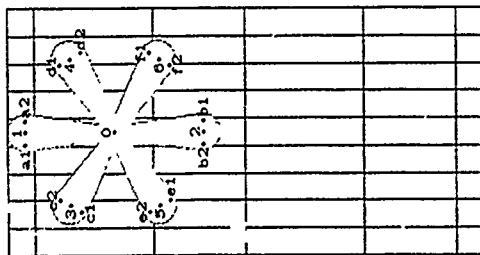


```

"Apostrophe";
call charbegin(0.047, 5, 0, 0, ph, 0, 3[px, ph] slant + 5pw - pu);
% There is rotational symmetry with respect to reverse apostrophe.
new v; v = fixwidth[u, 3u];
open, top_y1 = top_y2 = h; top_y3 = m; y1 = 3[y1, y2];
x1 = x2 = good_1.5r; x3 = good_0(x1 + v + eps), x1 = x1 - 5v - eps;
w1 draw 1;
hpen, w0 draw 2{1, 0} .3{0, -1} .4{3(x1 - x1), y1 - y1}.
"Left parenthesis";
call charbegin(0.050, 6, 0, 0, ph + pb, ph + pb - 2pa, (ph + pb) slant + 5pw - 5pu),
% There is left-right symmetry with respect to right parenthesis.
hpen; x1 = x1 = good_0(r - u);
top_y1 = h + b; y2 = a = 5[y1, y1]; w = y1, y1 = y1,
new v;
if fixwidth = 0: v = u;
else: v = 3u;
fi,
new w0, w0 = round 75[w0, w1], lt0, x2 = round(x1 - 4v),
x0 = x1 = x1 + 7.5v,
draw (0 .) [w0] 1 . [w0] 2 [w0] 3(. 4).
% stroke

"Right parenthesis";
call charbegin(0.051, 6, 0, 0, ph + pb, ph + pb - 2pa, pa slant - 5pu),
% There is left-right symmetry with respect to left parenthesis.
hpen; x1 = x1 = good_0(r);
top_y1 = h + b; y2 = a = 5[y1, y1]; w = y1, y1 = y1,
new v;
if fixwidth = 0: v = u;
else: v = 3u;
fi,
new w0, w0 = round 75[w0, w1]; rt0, x2 = round(x1 + 4v);
x0 = x1 = x1 - 7.5v;
draw (0 .) [w0] 1 . [w0] 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-(i-1.875sqrt(3)pu);
open; top_yh = lowast{h+b,round(a+5m)}; top_yh = bot_yh = m;
y0 = 5yh, y1; x0 = r-x0; x1 = x2 = x0; % left-right symmetry
x3 = r-x1; x4 = r-x0; x5 = x2;
y1 = y0; y2 = y0; y3 = y1 - y0; 5[y0,y1] = y0;
x1-x0 = (.5sqrt(3))ft, 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",

```

open;
if fixwidth = 0: if pa+8pu > ph:
    call charbegin('053,18,0,0,ph,ph--2pa,pa slant--5pu), top_yh = h,
    else call charbegin('053,18,0,0,pa+8pu,8pu--pa,pa slant--5pu),
    top_yh = a+3.5u;
fi;

```

```

else call charbegin('053,9,0,0,3.5pu+pa,3.5pu--pa,0); top_yh = a+3.5u;
fi;
.5[y0,y1] = a; x1 = x2 = 5r;
if fixwidth = round u; x1 = r-x1; y1 = y1 = a,
w0 draw 1. 2,
draw 3 4,

```

"Comma",

```

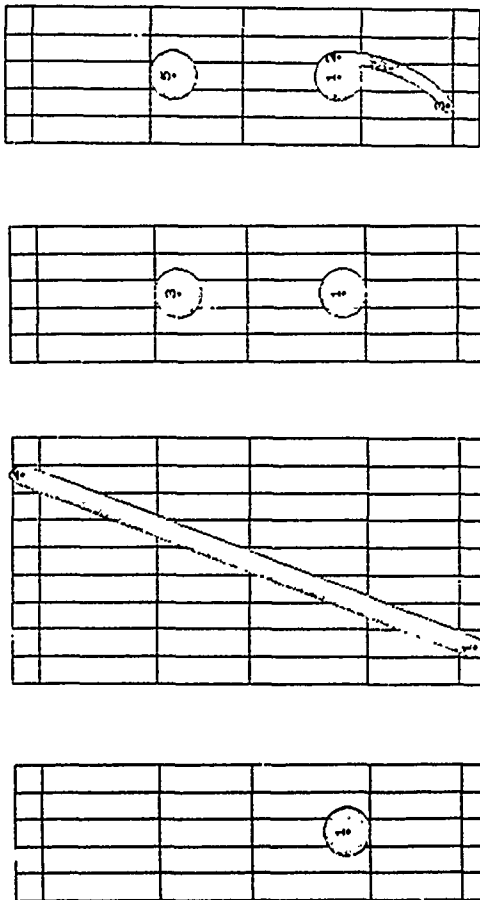
call charbegin('054,5,0,0,1.5pwin,pdd,0),
open, new w0;
if w1 < w0sqrt(2) w0 = round w0sqrt(2),
else w0 = w1,
fi;
x1 = goodw, 5r, bot_yh = 0; w0 draw 1,
y2 = y1, top_x1 = r0x2,
if fixwidth = 0: x1 = goodw, 1.5u,
else x2 = goodw, 1.5u;
fi;
open, bot_yh = -dd,
w0 draw 2{0,-1} .3{x1 x2,y1-y2}

```

% stem
% bar

% bulb

% point



```

"Period";
call charbegin(056, 5, 0, 0, 1.5pwt, 0, 0);
open, new w;
if w1 < w1sqrt 2; w1 = round w1sqrt 2;
else: w1 = w;
fi;
x1 = goodw1.5r; botw1y1 = 0; w1 draw 1.

"Virgule (slash)";
call charbegin(057, 9, 0, 0, ph + pb, ph + pb - 2pa, (ph + pb) slant - 5pu);
open; lft1x1 = round u; x2 = r - x1;
top1y2 = h + b; .5[y1, y2] = a;
w1 draw 1..2.

"Colon";
call charbegin(072, 5, 0, 0, px, 0, px slant + 5pwt sqrt 2 - 2pu);
open; new w;
if w1 < w1sqrt 2; w1 = round w1sqrt 2;
else: w1 = w;
fi;
x1 = goodw1.5r; botw1y1 = 0; w1 draw 1;
x1 = x1; topw1y1 = m; w1 draw 3.

% diagonal

```

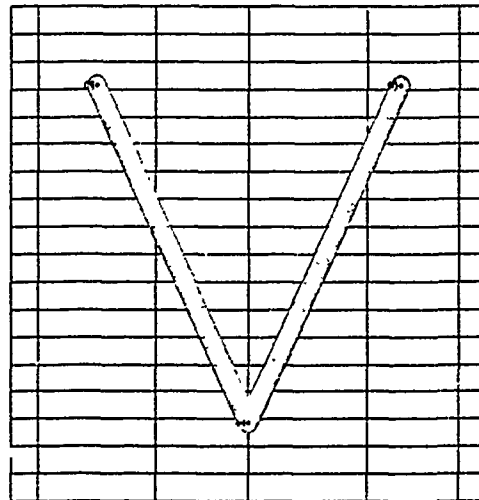
```

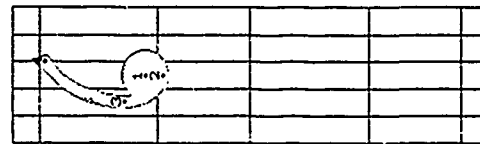
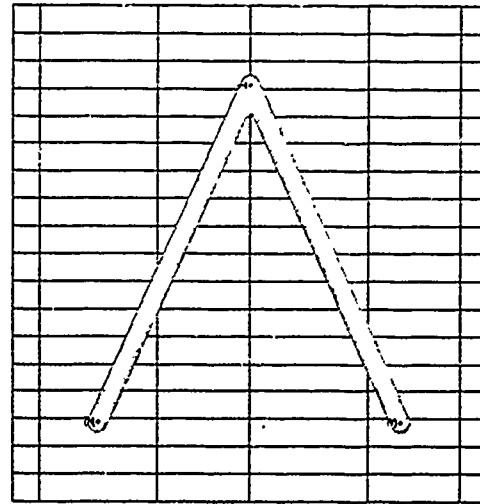
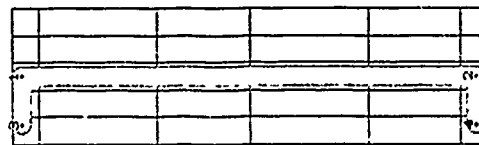
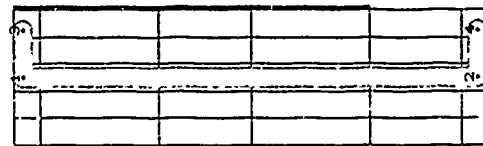
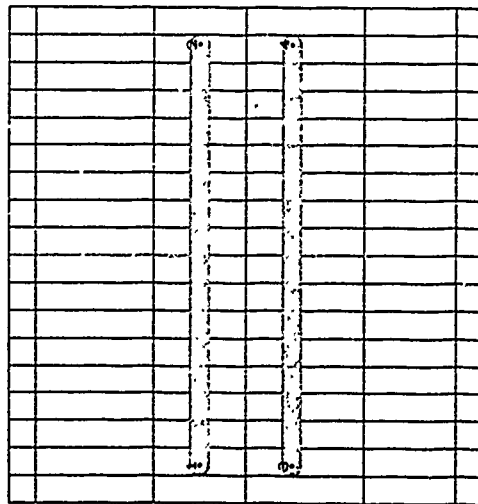
"Semicolon";
call charbegin(073, 5, 0, 0, px, pdd, px slant + 5pwt sqrt 2 - 2pu);
open; new w;
if w1 < w1sqrt 2; w1 = round w1sqrt 2;
else: w1 = w;
fi;
x1 = goodw1.5r; botw1y1 = 0; w1 draw 1;
x2 = x1; topw1y2 = m; w1 draw 5;
y2 = y1; rtw1x1 = rtw1x2;
if fixwidth = 0; x1 = goodw1.5u;
else: x1 = goodw1.5u;
fi;
hpen; botw1y1 = -dd;
w1 draw 2{0, -1} .3{3(x1 - x2), y1 - y2}

"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);
open; lft1x1 = round 2.5u; x2 = r - x1;
y2 = goodw1.5m/h; .5[y2, y1] = y1 = goodw1.5a;
w1 draw 2..1..1..3.

% built
% do

```





```
"Equal sign",
call charbegin(75, 0, 0, pa + 5(px - pc) + prt/2, 0, (pa + 5(px - pc)) slant -- 5pt),
open; if u0x1 == round u, x1 == x1, x2 == x1 + r -- x1;
y1 == y1, y2 == y1; y1 - y1 == round(m - e); 5[y1, y1] == a,
u0 draw 1..2;
draw 3..4;
```

% upper bar
% lower bar

```
"Greater than sign",
call charbegin(76, 0, 0, 5[px, ph] + prt/2,
5[px, ph] + prt/2 -- 2pa, pa slant -- 2pt);
open; if u0x2 == round 2.5u, x2 == x1 + r -- x1,
y2 == good 10 5[m, h], 5[y2, y1] == y1 == good 10a;
u0 draw 2..3
```

% diagonals

```
"Left bracket",
call charbegin(133, 5, 0, 0, ph + ph + ph + pb -- 2pa,
(ph + pb) slant + prt/2 -- 25pt),
open; x1 == x2 == good 10 5r, x1 == x1 + x1 + 1.75u + eps,
top10y1 == h + b, 5[y1, y1] == a, y1 == y1; y1 == y1,
u0 draw 3..4
```

% stem and points

```
"Right bracket",
call charbegin(135, 5, 0, 0, ph + ph + ph + pb -- 2pa,
(ph + pb) slant + prt/2 -- 2pt),
open; x1 == x2 == good 10 5r, x3 == x1 == x1 + 1.75u -- eps,
top10y1 == h + b, 5[y1, y1] == a, y1 == y1; y1 == y1,
u0 draw 3..4
```

% stem and points

```
"Reverse apostrophe",
call charbegin(140, 5, 0, 0, ph, 0, ph slant + 5pw - 1.5pt);
% There is rotational symmetry with respect to apostrophe
n w v, v == fixwidth[u, qd], % unit width adjusted to agree with opening quotes
open; bot y1 == bot u, top y1 == h, y1 == 2[y1, y1], y2 == y1 + m - h;
x1 == x2 == good 3.5r, x1 == good 1.5r - v -- eps, x1 == x1 + 5v + eps,
w1 draw 1;
hpen; u0 draw 2{--1.0} 5{0, 1} 4{3(x1, x1), y1 == y1}
```

% bulb
% tail

The file romita mf

% This file contains accents common to text roman and italic fonts.
 % It also contains the 'A', 'Q', and 'E', 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* ≠ 0, since other symbols are substituted
 % for those accents in non-ligature fonts)

"Grave accent";
 call charbegin('015, 9, 0, 0, ph, 0, 2[ph, px] slant + 5pw - pw/6 - 1/2 pu),
 open; lt1x1 = round 2u; x2 = 1/2{x1, r - x1},
 top y1 = h; y2 = 3/4{h, m};
 call ~a cdraw(1, 2, 1, 0);

% diagonal

"Acute accent";
 call charbegin('016, 9, 0, 0, ph, 0, ph slant - 1.5pu),
 open; rt1x1 = round(r - 2u); x2 = 1/2{x1, r - x1},
 top y1 = h; y2 = 3/4{h, m};
 call ~a cdraw(1, 2, 1, 0);

% diagonal

"Circumflex (hat) accent";
 call charbegin('017, 9, 0, 0, ph, 0, 5[px, ph] slant + 5pw - 2pu);
 x1 = good0 2 5u; x2 = x1 = r - x2; x3 = r - x1,
 vpen; y1 = y2 = 5[m, y0]; top y0 = top y2 = h; bot y0 = bot y2,
 u1 ddraw 2 .1, 4. .1;
 ddraw 2 .3, 4 .3.

% left-right symmetry

% left point
 % right point

"Hachek (hooklet) accent";
 call charbegin('020, 9, 0, 0, 75[px, ph] slant + .5pw - 2pu),
 x1 = good0 2 5u; x2 = x1 = r - x2; x3 = r - x1,
 vpen; top y0 = round .75[m, h]; y1 = y2;
 top y0 = h; y2 = .5[m, y0]; y1 - y0 = y20 - y21,
 top y0 = top y1; bot y0 = bot y2;
 u1 ddraw 2 .1, 4. .1;
 ddraw 2 .3, 4 .3

% left-right symmetry,

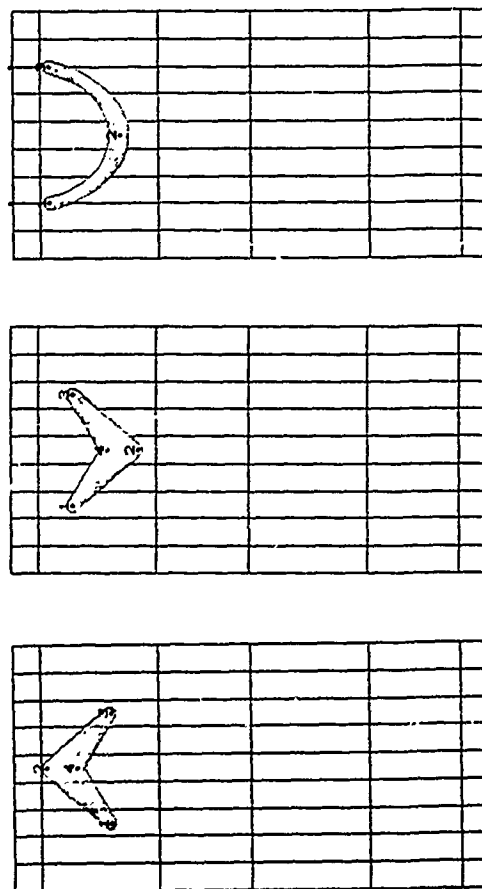
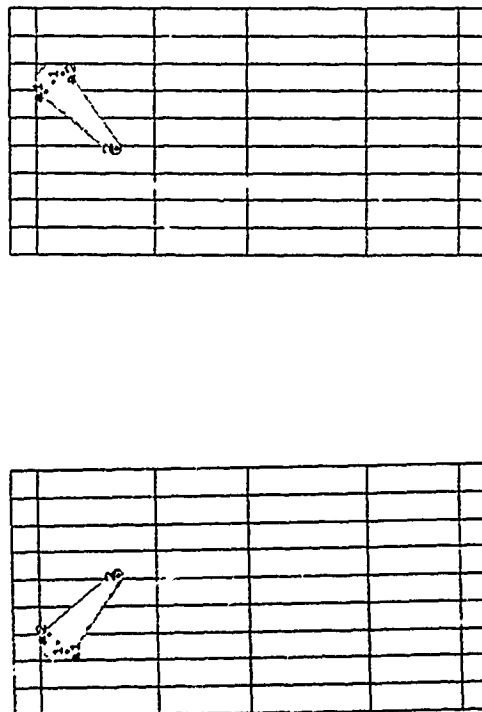
% inverted circumflex

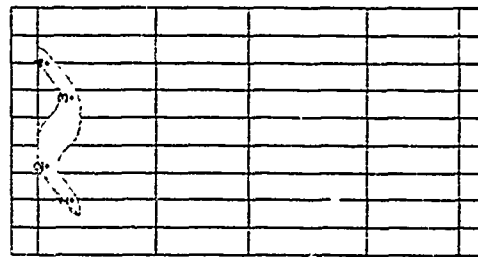
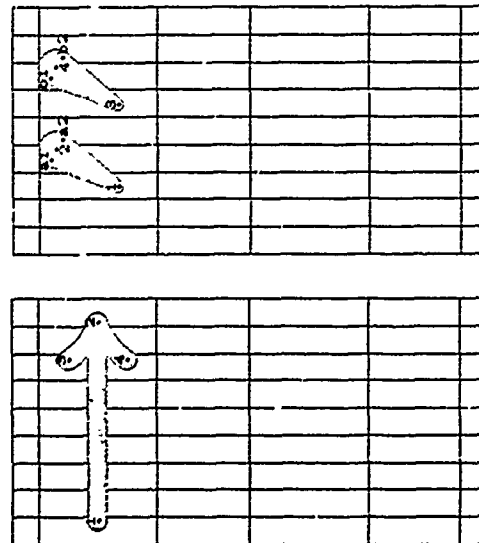
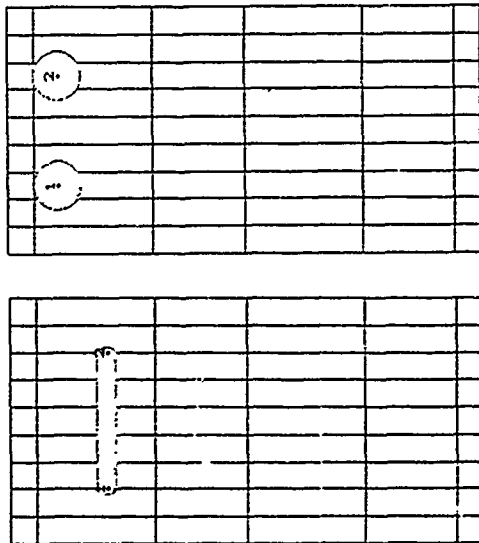
% left point
 % right point

"Breve accent";
 call charbegin('021, 9, 0, 0, ph, 0, ph slant + 5pw - 1.5pu),
 x1 = x2 = good0 2u; x3 = r - x2; x4 = x1 = r - x1,
 new w0, w1 = round 25[u0, w0],
 y0 = y1 = 1.25[m, h]; y2 = y1; y3 = 1/2{m, h},
 vpen; top y0 = h,
 draw [w0#12{-1, 0}...[w0#1]{.0};
 draw [w0#12{1, 0}...[w0#13]{...4}).

% left-right symmetry

% left point
 % right point





```

"Macron (bar) accent";
call charbegin('022,9,0,0,{px,ph}+pr aspect,0,{px,ph}slant+5pw-15pu);
x1=good,2u; x2=r-x1; y1=u;
vpen, new w0; w0=round 25{u,w}, bot,y1={m,h},
w0, draw 1..2
% bar

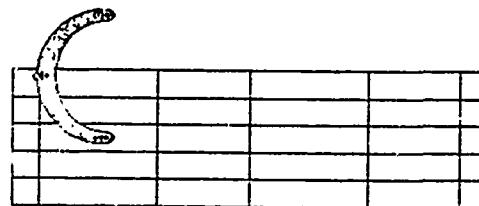
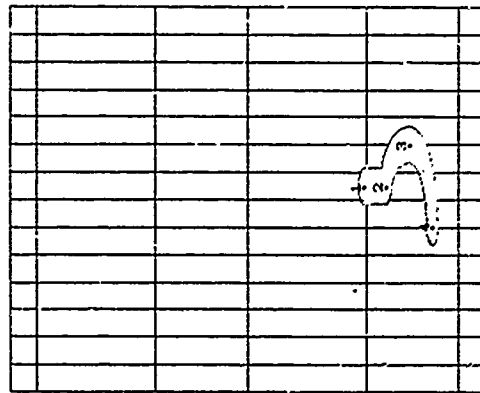
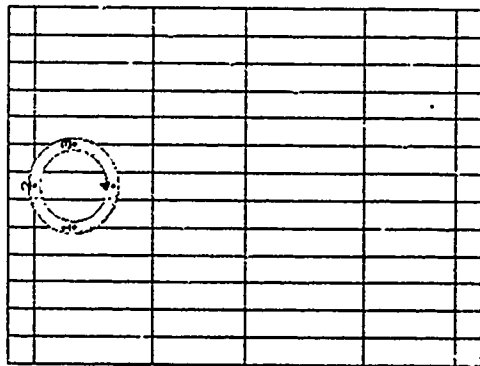
"Umlaut (double dot) accent";
call charbegin('023,9,0,0,ph,0,ph,slant+5pw-15pu);
x1=good,2.5u; x2=r-x1; y1=y2;
open, top,y1=h,
w, draw 1;
draw 2.
% left dot
% right dot

"Tilde (squiggle) accent";
call charbegin('024,9,0,0,ph,0,ph,slant+5pw-15pu);
x1=2u, x2=r-2u, x3=25{x1,x2}, x4=75{x1,x2},
new aa,bb,rr,cosh,sinh;
aa={x1-x2}, bb=2(h-m); rr=sqrt(aa-aa+bb-bb);
cosh=aa/rr; sinh=bb/rr;
spen(4{cosh cosh/w0/w0+sinh sinh/w0/w0},
8{cosh sinh(1/w0/w0-1/w0/w0)},
0,0,0);
4{cosh cosh/w0/w0+sinh sinh/w0/w0},
% oblique pen in direction {(x1-x2)/3,y2-y1}
top,y1=8{m,h}; top,y2=h; y1=y1; y2=y2;
w0, draw 1..2{1,0}..3{1,0}..4.
% points and stroke

if ligs<0 "Arrow (vector) accent",
call charbegin('025,9,0,0,75{px,ph}+pr aspect/2,0,5{px,ph}slant),
open, lt,x1=round 5u, x2=r-x1; y1=y2-5{m,h},
w0, draw 1..2;
rt,x1=round(x2-u); x2=x1;
y1=y2+(h-m)/4; y2=y2-(h-m)/4;
draw 3{5{x2-x1},y2-y1} 2{x2-x1,5{y2-y1}},
draw 4{5{x2-x1},y2-y1} 2{x2-x1,5{y2-y1}};
% bar
% upper point
% lower point

if ligs>0 "Long Hungarian umlaut accent",
call charbegin('026,9,0,0,ph,0,ph,slant+pu),
open, x1=good,2.5u, rt,x2=round 3r,
x1-x2=x1-x2; rt,x1=round(r-1.5u),
y1=y1-1{m,h}, top,y2=h, y2=y2;
call 'a cdraw(2,1,1,0),
call 'b cdraw(4,3,1,0),
% left diagonal
% right diagonal

```



```

"Scandinavian circle accent";
% This character is designed to be used as an ordinary accent on an 'a'
% or to be raised 3(ph - px) points and superimposed on an 'A'
call charbegin(027, 13, 0, 0, ph, 0, 0);
x1 = goodh, 5u; x2 = x1 = r - x2; x1 = r - x2;
lpen, topy2 = h + o; topy2 = round 1(m, h) + o; y1 = y1 = 5(y, y);
w1 draw 4(-1, 0) .. 1(0, 1) .. 2(1, 0);
draw 4(1, 0) .. 3(0, 1) .. 2(-1, 0);

"Cedilla accent";
% This character is designed to be used as an ordinary accent on an 'e'
% or to be superimposed on an 'C'.
call charbegin(030, 14, 0, 0, .75pd, 0);
x1 = x2 = 7.5u; x1 = x2 + 1.5u; x1 = x2 + 1.5u;
new wop, wop = round 5(w, w);
lpen, boty1 = -oo; boty1 = round(-25d - oo);
w1 draw 1..2;
boty1 = round(-75d - oo); y1 = 5(y, y);
draw 2(1, 0) .. 3(0, -1) .. 4(1, 0);

if lgs ≠ 0 "Cross for Polish l and L";
% This character is designed to be used as an ordinary accent on an 'l'
% or to be raised ph - px points and superimposed on an 'L'
call charbegin(031, 4 + pwv/pu, 2sc, 2sc, px - (ph - px), 0, 0);
lt 1.5(x1, x2) = 2u; x2 = x1 = 3u;
topy2 + (h - m) = m; y1 + (h - m) = e;
w1 draw 1..2;

fi.

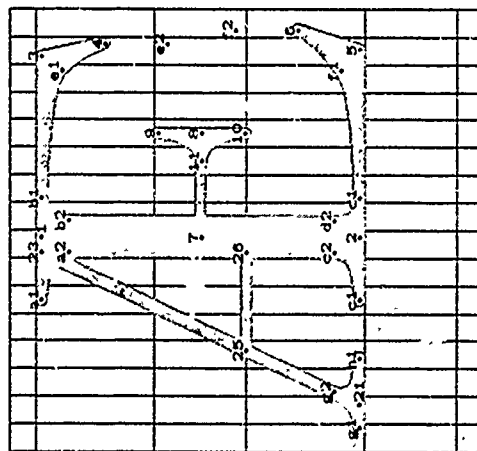
% diagonal

if lgs ≠ 0 "Tie accent";
% This character is tuned to work best with a dotless roman 1
call charbegin(032, 5, sc, sc, ph, 0, px slant + 5pw1 + (sc - 2)pu);
x1 = goodh, 5r; y1 = y1 = 1(m, h); x2 = 5(x1, 1); x1 = goodh(r + 2u);
vpen, topy2 = h + oo;
w1 draw 1(0, 1) .. 2(1, 0);
draw 3(0, 1) .. 2(-1, 0);

fi.

% left point
% right point

```

```

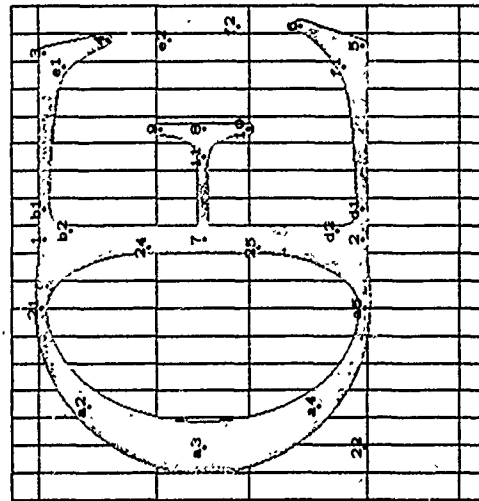
% upper case ligature AE";
%y charbegin(036,16,2sc,sc,ph,0,arunc);
hpen;
if (x1 == round 7u, x2 = 'i; top y1 = h, bot y2 = 0;
u1 draw 1..2;
if u3 != 0;
call 'a serif(1,4,2,-ucs);
call 'b serif(1,4,2,5ucs);
call 'c serif(2,4,1,-ucs);
call 'd serif(2,4,1,5ucs);

fi;
new ss; ss = 1.4*sqrt ucs-u + eps;
if ss + u6 > 25h; new ss; ss = .25h - u6 + eps;
fi;
rtotx1 = -round(r - 1.5u); x1 = x1 + 5u; y1 = y1; y1 = y1 - ss;
rtotx2 = -round(r - 1.25u); x2 = x1 + .5u; y2 = y2; y2 = y2 + ss;
call 'e arm(1,3,4);
call 'f arm(2,5,6);
u3 draw 7..8;
y3 = x1; y3 = y3 = .5[y1,y2]; x3 == good0 1.5u;
if ucs != 0; x1 = x1 = x3; y3 = y3 = 7ss, y10 = y3 - 7ss,
x2 = u1, u3 draw 9..10;
e c' x11 = x3 - u; y11 = y3;
minv 0; minv 0;
u3 ddraw 11; \ } 10{0,-1},8..10;
ddraw 11{1,0}..9{0,1},8..9;
minv .5; minv .5;

fi;

fi;
if (x1 == round 1.5u; bot y1 = 0; if (x2,1 == ft(x1,1,0,y1,y2) = h;
u3 draw 23..21;
y3 = y10 = c;
x3 = -1 = (y3 - y1)/(y1 - y2)[x1,x2], x10 = x1,
u3 draw 25..26,
if ucs != 0; call 'g serif(21,0,23,-.5ucs);
call 'h serif(21,0,23,ucs);
fi.

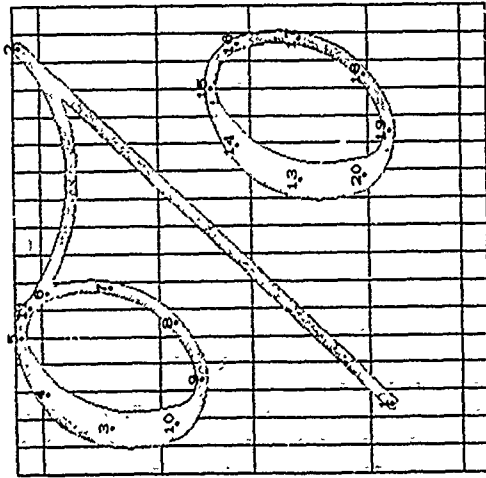
```



```

"Upper case ligature OE";
call charbegin(037, 18, 0, sc, ph, 0, armic);
hpen; new w0; w0y = round .5[w0, w0];
if0921 = round 9u; x2 = x1; top0y1 = h; bot0y2 = 0;
w0 draw 1..2;
if ucs ≠ 0;
  call "b serif(1.99, 2, .5ucs);
  call "d serif(2.99, 1, .5ucs);
fi;
new ss; ss = 1.4aspect-ucs-u + eps;
if ss + w0 > .25h; new ss; ss = .25h - w0 + eps;
fi;
r0x1 = round(r - 1.5u); x1 = x1 + .5u; y1 = y1 - ss;
r0x2 = round(r - 1.25u); x0 = x1 + .5u; y1 = y1 - ss;
call "o arm(1, 3, 4);
call "f arm(2, 5, 6);
x7 = x1; y7 = y8 = .5[y1, y2]; x8 = good0 13.5u;
w0 draw 7..8;
if ucs ≠ 0; x0 = x10 = x8; y0 = y8 + .7ss; y10 = y8 - .7ss;
  if u0 = w0; w0 draw 9..10;
  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;
    minvr 5; minvs .5;
  fi;
fi;
if fixwidth ≠ 0; new save, save = sqrttwo;
  new sqrttwo; sqrttwo = sqrt save;
  if0922 = round 1.5u;
  else; if0922 = round u;
fi;
x21 = 7u; top0y21 = h + oo; bot0y22 = -oo;
call "a darc(21, 22, w5);
x23 = x21; y23 = y22;
if0924 = if0921; x24 = x25; y24 = 3h; y25 = 1h;
w0 draw 21{1, 0}..24{0, -1};
draw 23{1, 0}..25{0, 1};
if ucs ≠ 0; if w0 ≠ w1;
  ddraw 21{1, 0} 24{0, -1}, 21{1, 0} 1{1, 0};
  ddraw 23{1, 0} 25{0, 1}, 23{1, 0} 2{1, 0};
fi;
fi;
if fixwidth ≠ 0; new sqrttwo; sqrttwo = save;
fi.

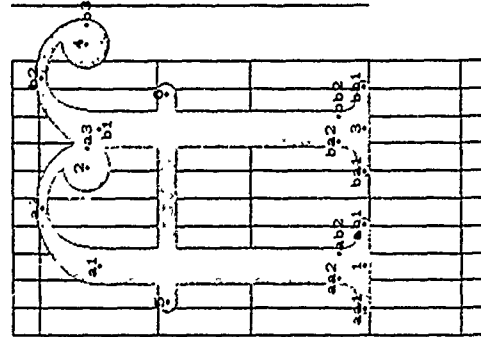
```



```

"Per cent sign";
call charbegin(0.45, 17, 0, 0, ph + pb, ph, (ph + pb).slant + 5pw - pw);
hpen; x1 = 2.5u; x2 = r - 1.5u; bot0y1 = -b, top0y2 = h + b;
u0 draw 1..2;
% diagonal
ll0x1 = round u; rt0x1 = round .5(r - 3u);
new ss, st; ss = .25(x2 - x1)/(y2 - y1);
% reciprocal slope for ellipses
y1 = y2 = .5(y1, y2); top0y5 = h + b; bot0y3 = round .5h;
st = 1/(sqrt 2); y1 = y2 = st[y1, y2]; y3 = y4 = st[y1, y2];
ss(y1 - y2) = x3 - x2; ss(y1 - y2) = x1 - x0 = x6 - x5;
x3 = .5(x1, x2) = .5(x2, x0); 5{x1, x0} = st{x1, x2}, 5{x2, x0} = st{x1, x2};
draw |w0#13{ss, 1}..12{w0, w0}|4{x1 - x2 + ss(y1 - y2), y1 - y2}..
|w0#15{1, 0}..6{x1 - x2 + ss(y1 - y2), y1 - y2}..7{-ss, -1}..
8{x0 - x1 + ss(y1 - y2), y1 - y2}..9{-1, 0}..
10{w0, w0}|10{x1 - x2 + ss(y1 - y2), y1 - y2}..11{13{ss, 1},
y1 = .9(y1, y2); x1 = (sqrt(1 - (.9)(.9)))(x1, x2) + ss(y1 - y2);
y2 = .9(y1, y2); x2 = (sqrt(1 - (.9)(.9)))(x2, x2) + ss(y1 - y2);
% upper bowl
% point on ellipse
% nearby point
% link
w0 draw (12..11..2{x2 - x1, y2 - y1});
x1 - x2 = x11 - x1 = x15 - x5 = x16 - x6 = x17 - x7 =
x18 - x8 = x19 - x9 = x20 - x10 = round .5(r + u);
y11 - y1 = y14 - y1 = y15 - y5 = y16 - y6 = y17 - y7 = y18 - y8 = y19 - y9 = y20 - y10;
bot0y9 = -b;
draw |w0#13{ss, 1}..12{w0, w0}|4{x1 - x2 + ss(y1 - y2), y1 - y2}..
|w0#15{1, 0}..6{x1 - x2 + ss(y1 - y2), y1 - y2}..7{-ss, -1}..
18{x0 - x1 + ss(y1 - y2), y1 - y2}..19{-1, 0}..
12{w0, w0}|20{x1 - x2 + ss(y1 - y2), y1 - y2}..13{ss, 1}
% lower bowl

```

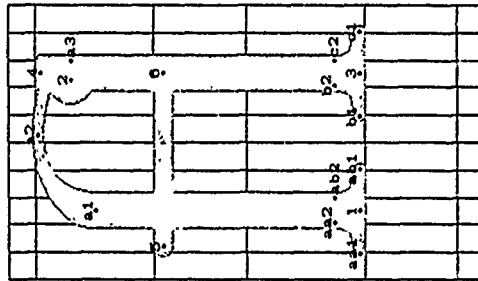
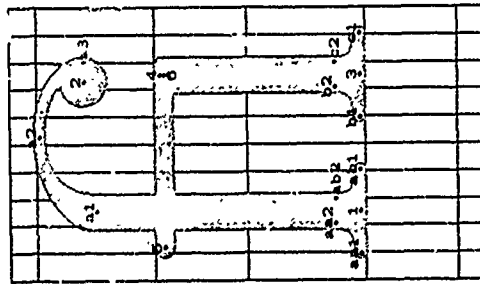


The file rom1.g.mf

```

"The ligature ff";
call charbegin(173, 10, 0, 0, ph, 0, ph, slant + 2pw);
hpen; x1 = good, 2.5u; x2 = good, (r - 2.5u);
if u0 = w1; rt1x2 = round 6.5u; x1 - x2 = x2 - x1;
else: rt1x2 = round(.5r + 2u); rt1x1 = round(.5r + 1.5u);
% left bulb, shoulder, stem, and serif
% right bulb, shoulder stem and serif
open; top0y2 = 8/m, h; y1 = y2;
call ~a fstroke(2, 1);
call ~b fstroke(4, 3);
ll0x2 = ll0x1 - u - eps, rt0x2 = rt1x1 + u + eps, top0y3 = m, h, ..y1;
w0 draw 5..6;
fig ~f: ~i = 174, ~f = 173, ~1 = 175;

```



```

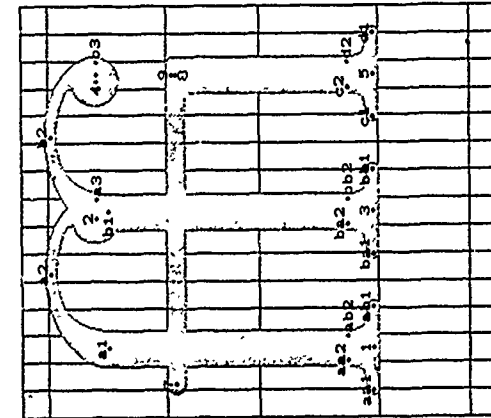
"The ligature ff";
call charbegin('174, 10, 0, 2sc, ph, 0, ph-slant + 5pwi + (2sc - 2)pu);
lpen; x1 = good1 2.5u; x2 = x1 = good1(r - 2.5u);
rt, x2 = rt, x1;
open; top, y2 = .8[m, h]; % bulb, shoulder, left stem, left serif
call 'a fstroke(2, 1); % right stem
lpen; top, y1 = m; bot, y1 = 0; w1 draw 3..4;
open; lft, x2 = lft, x1 - u - eps; x0 = x2; top, y2 = m; y0 = y2; % bar link
w1 draw 5..6;
if lcs ≠ 0: call 'b serif(3, 1, 4, -lcs); % right serif
call 'c serif(3, 1, 4, lcs); % appropriate end to "ff"
fi.

```

```

"The ligature ff";
call charbegin('175, 10, 0, 2sc, ph, 0, ph-slant + .5pwi + (2sc - 2)pu);
lpen; x1 = good1 2.5u; x2 = x1 = good1(r - 2.5u);
rt, x2 = rt, x1;
open; top, y2 = .9[m, h]; % bulb, shoulder, left stem, left serif
call 'a fstroke(2, 1); % right stem
lpen; top, y1 = h; bot, y1 = 0; w1 draw 3..4;
open; lft, x2 = lft, x1 - u - eps; x0 = x2; top, y2 = m; y0 = y2; % bar link
w1 draw 5..6;
if lcs ≠ 0: call 'b serif(3, 1, 4, -lcs); % right serif
call 'c serif(3, 1, 4, lcs);
fi.

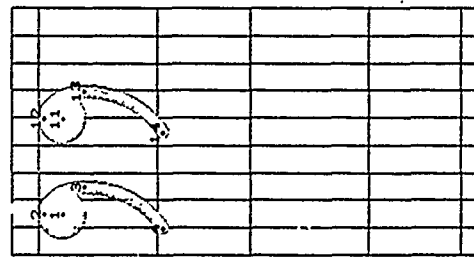
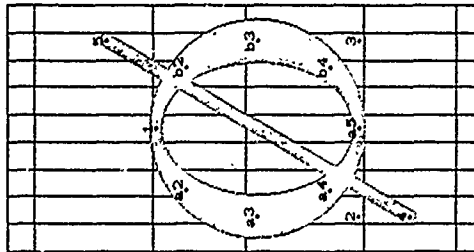
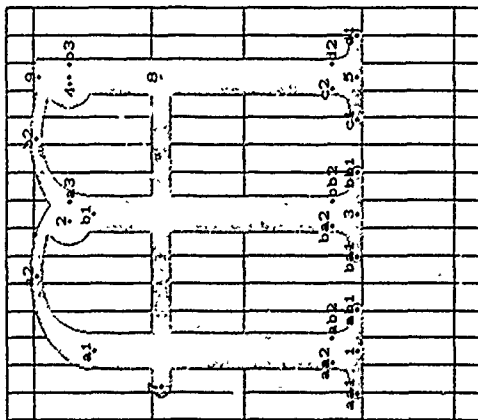
```



```

"The ligature ff";
call charbegin('176, 15, 0, 2sc, ph, 0, ph-slant + 5pwi + (2sc - 2)pu);
lpen; x1 = good1 2.5u; x2 = x1 = good1.5r;
x3 - x2 = x1 - x2 = x; x0 = x1; x0 = x2 = x3; y2 = y1 = y0;
open; top, y2 = .8[m, h]; rt, x1 = rt, x0; % left bulb, shoulder, stem, and serif
call 'a fstroke(2, 1); % right bulb and shoulder, middle stem and serif
open; top, y1 = m; bot, y1 = 0; w1 draw 5..9; % right stem
lpen; lft, x2 = lft, x1 - u - eps; x0 = x2; top, y2 = m; y0 = y2; % bar link
w1 draw 7..8;
if lcs ≠ 0: call 'c serif(5, 1, 6, -lcs); % right serif
call 'd serif(5, 1, 6, lcs);
fi.
lig '173: '1 = '176, '1 - '177,

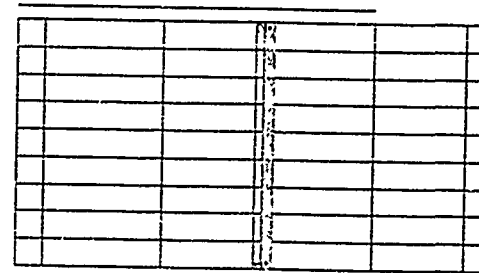
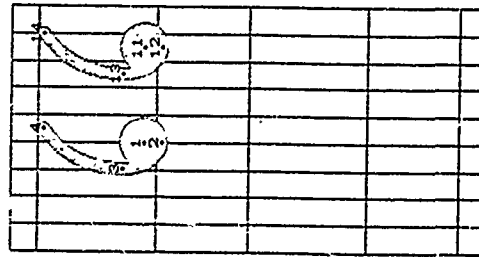
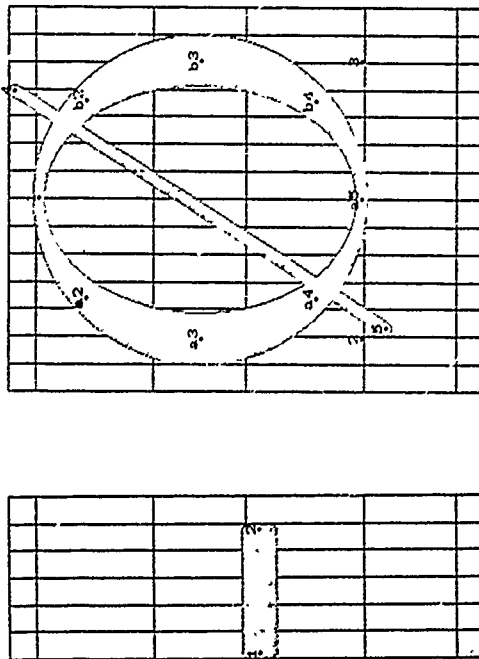
```



“The ligature III”;
 call charbegin(177, 15, 0, 2sc, ph, 0, ph slant + .5pu + (2sc - 2)pu),
 then; $x_1 = \text{good}, 2.5u, \quad x_1 = \text{good}, .5r$;
 $x_2 - x_1 = x_1 - x_2 = x_1 - x_1; \quad x_6 = x_7 = x_2, \quad y_2 = y_1 = y_4$;
 open; $\text{top}_1 y_5 = .9[m, h]; \quad \text{rt}_3 x_1 = \text{rt}_1 x_0$;
 call ‘a stroke(2, 1);
 call ‘b stroke(4, 3);
 open; $\text{top}_0 y_5 = h; \quad \text{bot}_0 y_5 = 0; \quad w_1 \text{ draw } 5..0$;
 open; $\text{lt}_{100} y_5 = \text{lt}_1 x_1 - u - \text{eps}; \quad x_3 = x_1; \quad \text{top}_{100} h = m, \quad y_8 = y_1$;
 $w_{10} \text{ draw } 7..8$;
 if $\text{ics} \neq 0$: call ‘c serif(5, 1, 6, -ics);
 call ‘d serif(5, 1, 6, lcs);
 fi.

The file romit1.mf

```
% This file defines characters common to roman and italic text fonts
% that do not appear in non-ligature fonts.
% Character codes '040, '042, '055, '100, '134, '136, '137 are used.
% The Scandinavian letter o-shash";
call charbegin('040, 9, 0, 0, px + .5pd, 5pd, (px + .5pd).slant + 5(pw - pwt));
when; x1 = r - x1;
else: ifxwidth = 0: lft2x2 = round .5u;
else: lft2x2 = round 1.5u;
fi;
x1, x2 = x1 - x1; top0y1 = m + oo; bot0y2 = -oo; y2 = y1,
call `a dare(1, 2, w2);
call `b dare(1, 3, w3);
x1 = x2; x3 = x1, y1 = -5d; y3 = m + 5d;
u3 draw 4..5.
"Closing quotes";
call charbegin('042, 9, 0, 0, ph, 0, {[px, ph].slant + 5pw - 2.5pn}),
when; top0y1 = top0y2 = h; top0y1 = m, y1 = {[y1, y1],
x1 = x2 = goody, 1.5u, x3 = goody(x1 + u + eps); x1 = x1 - 5u - eps;
x1 .. x1 = x12 - x2 = x11 - x1 = x11 - x1 - round 3.5u;
y1 = y1; y2 = y2; y14 = y1; y11 = y1;
w1 draw 1;
w1 draw 11;
draw 11;
when; u3 draw 2{1, 0}..3{0, -1}. 4{3(x1 - x3), y1 - y3};
draw 12{1, 0}..13{0, -1}..14{3(x11 - x13), y11 - y13}.
lig ... = '042;
```



```

"llyphen";
call charbegin('055, fixwidth{6, 9}, 0, 0, px, 0, 5px slant - .5pu);
vpen; y1 = y2 = .5m;
if fixwidth = 0 ift1x1 = 0; r1x2 = r - u;
else ift1x1 = 1.5u, x2 = r - x1;
fi;
w1 draw 1 .2.

"The Scandinavian letter O-slash";
call charbegin('100, 1/4, 0, 0, ph + pb, pb, 5pl slant - .5pu),
hpen;
if fixwidth ≠ 0 new save, save = sqrttwo;
new sqrttwo; sqrttwo = sqrt save,
ift1x2 = round 1.5u;
else ift1x2 = round u;
fi;
x1 = r - x1;
top0y1 = h + co,
bot0y2 = -oo; y1 = y2, x1 = r - x2,
call 'a dare(1, 2, w3),
call 'b dare(1, 3, w3),
if fixwidth ≠ u new sqrttwo, sqrttwo = save,
fi;
top0y1 = h + b, bot0y2 = -b; r10x1 = ift1x1, ift0x2 = r10x2,
w1 draw 4 .5;
fi.

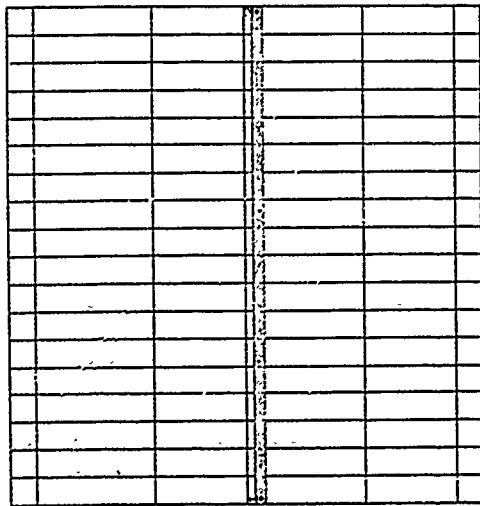
```

```

"Opening quotes";
call charbegin('134, 0, 0, 0, ph, 0, ph slant + 5pw - 5pu),
% There is rotational symmetry with respect to closing quotes
open; bot0y1 = bot0y2; top0y1 = h, y1 = {y1, y2}; y2 = y1 = m - h;
x1 = x2 = good, 4u, x1 = good, (x1 - u - eps), x1 = x1 + 5u + eps,
x11 = x1 = x12 - x2 = x11 - x1 = round 3.5u,
y11 = y1; y12 = y2, y13 = y1; y14 = y1;
w1 draw 1;
draw 11;
hpen, w1 draw 2{-1, 0} 3{0, 1} 4{3{x1 - x1}, y1 - y1},
draw 12{-1, 0} 13{0, 1} 14{3{x11 - x1}, y11 - y13}
lig '": ' = '134;

"En dash";
call charbegin('136, 0, 0, 0, 5px + 5pw aspect, 0, 5px slant + 5pu),
hpen; ift0x1 = 0; r10x2 = r; y1 = y2 = .5m;
w1 draw 1..2.
lig '~: '~ = '136;

```



```

%in dash";
call charbegin('137, 18, 0, 0, 5px + 5pw-aspect, 0, 5px-slant + 5pu);
lpen; lfto21 = 0; rto22 = r; y1 = y2 = .5m;
u0 draw 1..2.
fig '136: "- = '137;
% bar

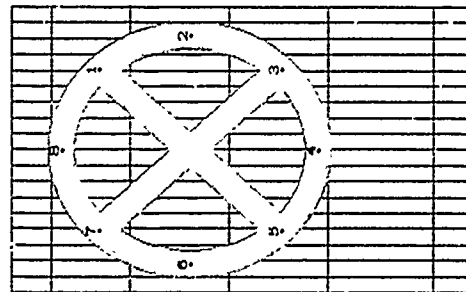
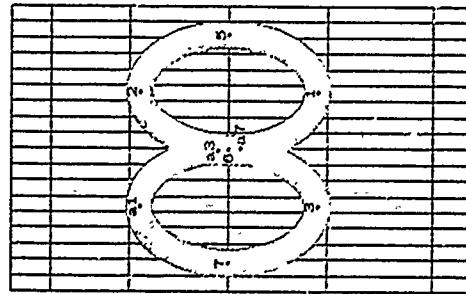
```

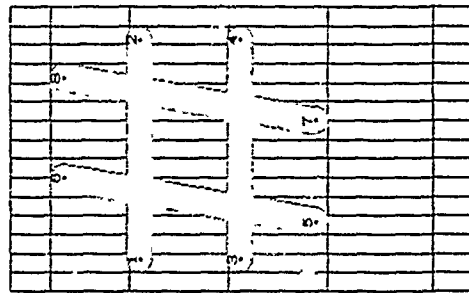
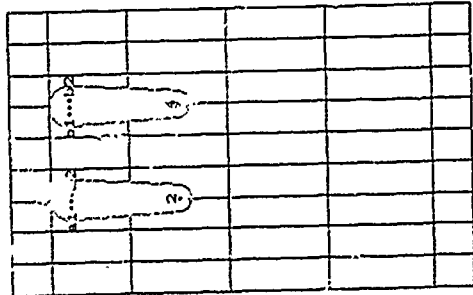
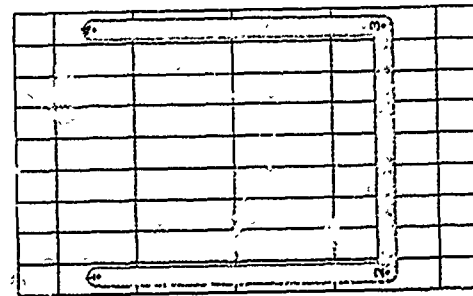
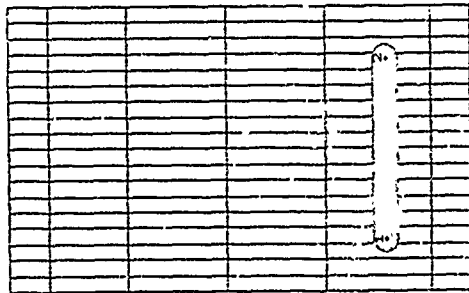
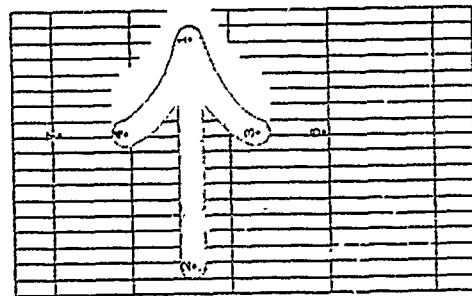
The file romits mf

```

% This file contains the characters substituted into a text font when lgs = 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, 5px-slant - 5pu),
new w0, w0; w0 = round 25{u0, w0}; w0 = 2{u0, w0};
vpen; top0y2 = m + oo; bot0y1 = -oo; y2 = y1; y1 = y1;
x1 = x1; x1 = x1; y1 = y1 = y1 = y1 = y1 = y1;
lft0x7 = round u; rto22 = round(r - u); x0 = 5{x7, x0};
new ss, mss; ss = -6u/m;
if u0 = w0; mss = ss;
else mss = .75ss;
fi;
call "a zdraw(5, 1, 6, 4, 7, w0, w0, ss);
w0 draw 5{0, 1}..2{-1, 0} 6{mss, -1}
3{-1, 0}..7{0, 1}.
"Circle-times operator";
call charbegin('026, 18, 0, 0, ph - 2pa, pa-slant - 5pu),
open; lft0x0 = round u; y0 = a; x0 = r - x0; top0y0 = h + oo;
call circle(1, 2, 3, 4, 5, 6, 7, 8, u0);
u0 draw 7..3;
draw 5..1.
% bowl
% upper left to lower right diagonal
% lower left to upper right diagonal

```





```

"Rightward arrow";
call charbegin(031, 18, 0, 0, 24pt + 5prt + pa, 24pt + 5prt - pa, pt slant - 5pu);
open; iftop2 == round u, rtw2 == x0 == round(r - u);
y1 == y2 == y5 == y8 == good10a;
w10 draw 1..2;
hpen; rt15 == x0;
x5 - x8 == x8 - x1 == fixwidth[3u, 6u] + eps; x1 == x1 == x0 == x1;
y3 - y6 == y1 - y3 == y1 - y1 == y1 - y1 == 2th + eps;
open; w10 + w1 draw (5..8) 3(.6);
hpen; draw (w1[5..8] 3(.6);
open; w10 + w1 draw (5..8) 4(.7);
hpen; draw (w1[5..8] 4(.7);

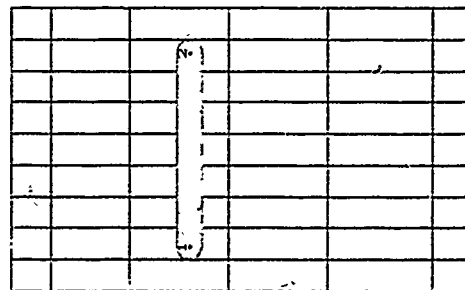
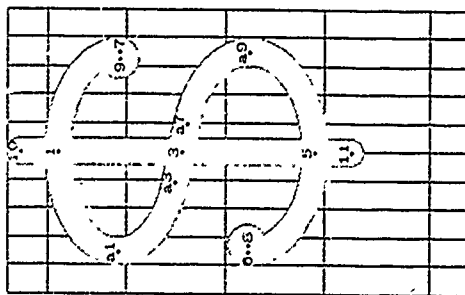
"Underbar suitable for < and >";
call charbegin(032, 18, 0, 0, 0, 5[px, ph], 5pd, 5[px, ph] slant + 25pu);
y2 == good10.5[m, h]; 5[y2, y2] == good10a; y1 == y2 == y21 - (m - c);
open; iftop2 == round 25u; x1 == x1 == round(r - 25u); x1 == x2, x1 == x1;
y1 == y1 == 5[m, h]; y2 == y1 == good10(-5u);
w10 draw 1..2..2..2..3..3..3..4;

"Blank-space sign";
call charbegin(040, 9, 0, 0, 0, 5[px, ph], 5pd, 5[px, ph] slant + 25pu);
open, new w1; w1 == round(75pxels pw + blacker);
iftop2 == round 25u; x1 == x1 == round(r - 25u); x1 == x2, x1 == x1;
y1 == y1 == 5[m, h]; y2 == y1 == good10(-5u);
w10 draw 1..2..2..2..3..3..3..4;

"Straight double quotes";
call charbegin(042, 9, 0, 0, 0, ph, 0, ph slant + 5pu..ii - 25pu);
new w1;
if w1 < w1sqrt2: w1 == round w1sqrt2;
else: w1 == w1;
if;
x1 == x2 == good10 3u, x1 == x1 == r - x1;
open; top2y1 == h; y2 == 5[c, m]; y1 == y1, y1 == y2;
call 'a cdraw(1, 2, 99, 0);
call 'b cdraw(3, 4, 99, 0)

"Sharp symbol (number sign or hash mark)";
call charbegin(043, 15, 0, 0, 0, ph, 0, ph - 2pu, 0);
open, iftop2 == round u; x1 == x1; x2 == x1 == r - x1;
y1 == y2, y1 == y1; y1 - y1 == round(m - c), 5[y1, y1] == a;
w10 draw 1..2;
draw 3..4;
x5 - 2u == x1; x4 + 2u == x2, x0 - x5 == x8 - x1; x0 - x1 == fixwidth[0, -3u];
y5 == y1; y6 == y8; top10a == h; 5[y1, y1] == a;
draw 5..6;
draw 7..8.

```

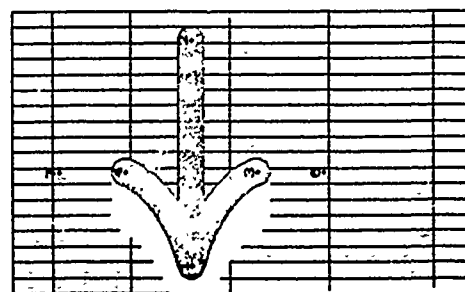
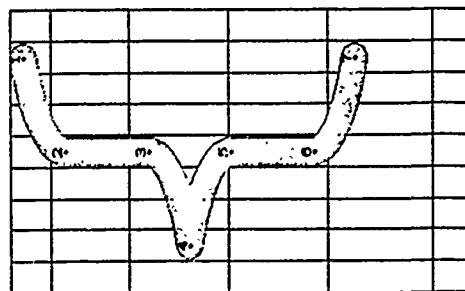
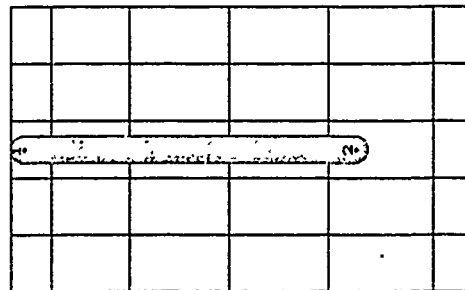
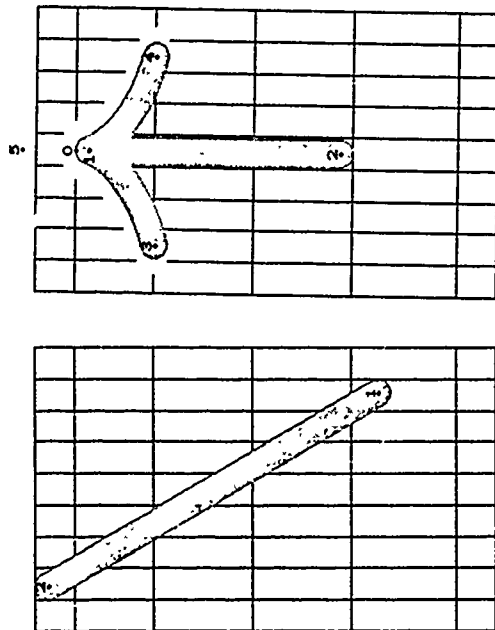
```

"Dollar sign";
call charbegin(0.04, 10, 0, 0, ph + pb, pb, ph-slant - .5pu);
hpen; top0y1 = h + oo; bot0y1 = -oo;
x1 = good10.5r; y1 = x2 = .52h; lft1x2 = round u, x1 = r - x2;
if ucs = 0: x1 = x2 = x1; y1 = 5[y1, y1]; y1 = .5[y1, y1];
else: if w0 = w1: x1 = x2 = x1; y1 = 5[y1, y1]; y1 = 5[y1, y1];
      else: x1 + .5u = x2 - .5u = x1; y1 = h/4 - 1, y1 = 8h + 1,
            fi;
fi;
y1 = y1; y1 = y1; cpen; lft1x2 = lft0x1; r1x2 = r0x1; x1 = x1; x1 = x1;
w1 draw 8;
draw 9;
hpen; w1 draw 6{0, --1} 5{1, 0};
draw 7{0, 1} 1{--1, 0};
call ~a sdraw(1, 2, 3, 4, 5, w1, w1, --h/(50u));
cpen; x10 = x1 = x2; top10y1 = h + b, bot10y1 = --b;
w10 draw 10..11.

"Minus sign";
cpen;
if fixwidth = 0: if pa + 8pu > pb;
  call charbegin(0.055, 18, 0, 0, ph, ph - 2pa, pa-slant - .5pu);
else: call charbegin(0.055, 18, 0, 0, 8pu + pa, pa-slant - 5pu);
fi;
else: call charbegin(0.055, 9, 0, 0, 3.5pu + pa, 3.5pu - pa, 0);
fi;
lft10x1 = round u; x2 = r - x1; y1 = y1 = a,
w10 draw 1..2.

"Alt sign";
call charbegin(100, 14, 0, 0, ph, 0, 5ph-slant + 5pw - .5pu);
hpen; x1 = x1 = x1 = .5r, x2 = good1.4u, x1 = x1 = x1 = r - x2,
lft0x2 = round u; x1 = x1 = x1 = r - .5r, x1 = x1 = x1 = x1,
top0y1 = h + oo; bot0y1 = -oo; y1 = y1;
y1 = good0.8[y1, y1]; y2 = y1 = y1 = good0.2[y1, y1];
y1 = .5[y1, y1]; y1 = good0.1[y1, y1]; y1 = 9[y1, y1];
y1 = 8[y1, y1]; y1 = 7[y1, y1];
call ~a darc(1, 2, w1);
call ~b darc(1, 3, w1);
draw lft14..lft15{0, --1} lft16{1, 0} 7{0, 1};
call ~c arc(8, 7, w1);
call ~d arc(8, 9, w1);
w1 draw 10{1, 0}..11{..12,

```



```

"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 = x3 = x8 = good10.5r;
w10 draw 1..2;
vpen; top10x8 = y0;
lpen; w10 draw 0..8; rpen; w10 draw 0..8;
y3 - y8 = y8 - y3 = .2dh + eps; y1 = y1 = y8 = y1;
x1 - x0 = x1 - x3 = x1 - x1 = x7 - x1 = 3u + eps;
lpen; w10 draw (5..8)..3(..6);
vpen; draw (w15..8)..[w13(..6);
rpen; w10 draw (5..8)..4(..7);
vpen; draw (w15..8)..[w14(..7).

"Leftward arrow";
call charbegin('137, 18, 0, 0, .24ph + r, : l pa, .24ph + .5prt - pa, pa slant - .5pu);
cpen; lft10x1 = x0 = round u; r10x2 = round(r - u);
y1 = y2 = y5 = y8 = good10a;
w10 draw 1..2;
hpen; lft10x8 = x0;
x5 - x8 = x8 - x3 = -fxwidth[3u, 6u] - eps; x1 = x1 = x0 = x1;
y1 - y8 = y1 - y3 = y1 - y1 = y7 - y1 = 2dh + eps;
lpen; w10 + w1 draw (5..8)..3(..6);
hpen; draw (w15..8)..[w13(..6);
lpen; w10 + w1 draw (5..8)..4(..7);
hpen; draw (w15..8)..[w14(..7).

% bar

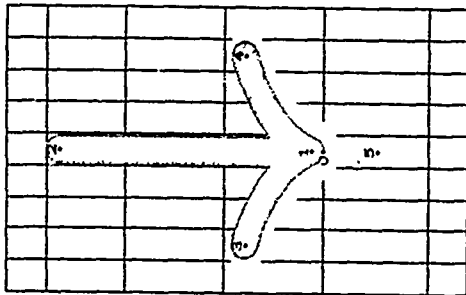
% erase excess at lower left
% lower point
% erase excess at upper left
% upper point

"Left brace";
call charbegin('173, 9, 0, 0, ph + pb, ph + pb - 2pa, (ph + pb)slant + .5pw - pu);
hpen; x2 = x1 = x3 = x0 = x8 = good.5r; x1 - x2 = x2 - x1 = 3u + eps; x1 = x1;
top10y1 = h + b; y1 = 5[y1, y1] = .5[y1, y1] = good10a;
y1 - y2 = y1 - y1 = (y1 - y1)/2;
draw [w15]1{3(x2 - x1), y2 - y1}..[w15]2{0, -1}..[w15]3{0, -1}..
[w15]4{3(x1 - x2), y1 - y1};
draw [w15]7{3(x0 - x7), y0 - y7}..[w15]6{0, 1}..[w15]5{0, 1}.
[w15]4{3(x1 - x2), y1 - y7}

"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);
open; top10y2 = h; .5[y1, y2] = a; y0 = bot10y1;
x0 = x1 = x2 = x3 = x4 = good10.5r;
w10 draw 1..2;
vpen; bot10y8 = y0;
lpen#; w10 draw 0..8; rpen#; w10 draw 0..8;
y1 - y8 = y8 - y1 = -2h - eps; y1 = y1 = y8 = y1;
x1 - x0 = x1 - x2 = x1 - x3 = x1 - x4 = 3u + eps;
lpen#; w10 draw (5..8)..3(.6);
vpen; draw {w15..8}..{w13(.6)};
rpen#; w10 draw (5..8)..4(.7);
vpen; draw {w15..8}..{w14(.7)};

"Right brace";
call charbegin('176, 9, 5, 0, ph + pb, ph + pb - 2pa, (ph + pb)slant + .5pwi - 4pu);
lpen; x2 = x1 = x3 = x4 = x0 = good1.5r; x1 - x2 = x2 - x1 = -3u - eps; x1 = x1;
top10y1 = h + b; y1 = .5[y1, y2] = .5[y1, y2] = good1.5r;
y1 - y2 = y1 - y1 = (y1 - y1)/4;
draw {w15}1{3(x2 - x1), y2 - y1}..{w15}2{0, -1}..{w15}3{0, -1}..
{w15}4{3(x1 - x1), y1 - y1};
draw {w15}7{3(x0 - x1), y0 - y1}..{w15}6{0, 1}..{w15}5{0, 1}..
{w15}4{3(x1 - x1), y1 - y1};

% stem
% clean the top
% erase excess at left
% left point
% erase excess at right
% right point

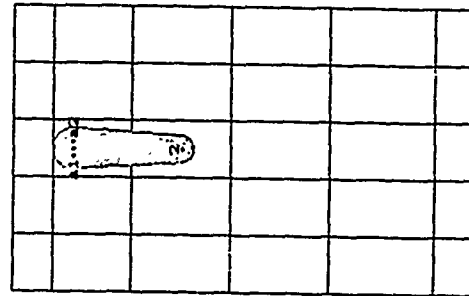
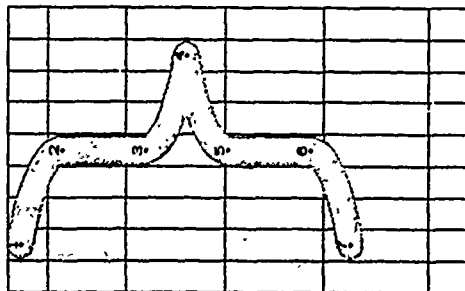
% upper stem
% lower stem

```

```

"Straight single quote";
call charbegin('177, 5, 0, 0, ph, 0, 0);
new w10;
if w1 < w1.sqr 2: w10 = round w1.sqr 2;
else: w10 = w1;
fi;
x1 = x2 = good10.5r;
open; top10y1 = h; y2 = .5[c, m];
call 'a cdraw(1, 2, 99, 0);

```



ITALIC CHARACTER DESIGNS

The file italic.mf

```
% The Computer Modern Italic family of fonts (by D. E. Knuth, 1979).
danger = mi/8;
input romitu;
input itail;
input itaid;
input romitp;
if mi ≠ 0:
  input greek1;
  input italms;
  input itmox;
  texinfo slant, 0pu, 3pu, 2pu, 18pu, px, 18pu, 2pu;
else:
  input romita;
  if lig ≠ 0: input itail;
  input itale;
  input romitl;
  input ittext;
  else: input romitex;
fi;
if fixwidth = 0: new k, kk, kkk;
  k = -1.5pu; kk = -1.5pu; kkk = -2.5pu; uk = +1pu;
  lig ~d: ~w: ~l: ~l kern uk;
  lig ~f: ~v:
    ~A kern kkk, ~o kern kk, ~e kern kk,
    ~a kern kk, ~u kern kk, ~r kern kk,
  ~K: ~X:
    ~O kern k, ~C kern k, ~G kern k, ~Q kern k;
  lig ~T: ~y kern kk,
  ~Y: ~o kern kk, ~e 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 ≠ 0: lig ~R: ;
fi;
lig ~A: ~t kern k, ~b kern k, ~h kern k, ~k kern k, ~l 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, ~O 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, 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 $tui = 0$.
 % Character codes '141'-'172 are generated.

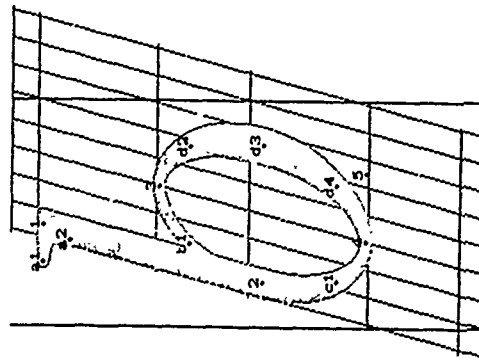
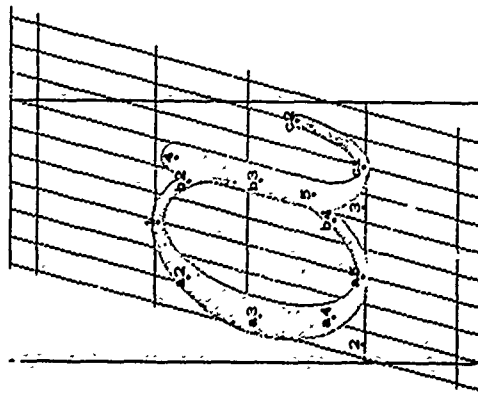
```

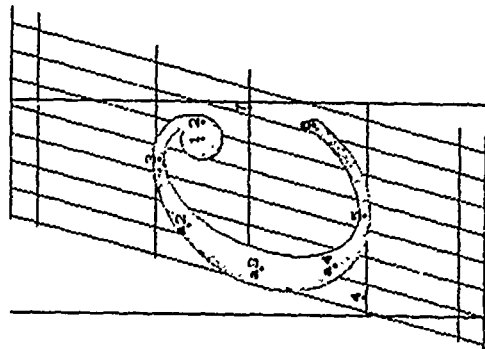
new mc, lbowl, lhook, rbowl, rhook, rstem;      % quantities used in spacing corrections
mc = mi/pu;
rhook = 1/3px-slant + .5pw + 5pu;
lbowl = 3px-slant - .5pwii + pu;
rbowl = 7px-slant + .5pwii - pu;
lhook = 1/3px-slant - .5pw - .5pu;
rstem = px slant + 5pvi - pu;

"italic letter a";
call charbegin("a, 9, mc-lbowl, -mc-rhook, px, 0, mi[rhook, 0]);
open; x1 = .5{x2, x1}; x2 = good, 1.5u; x1 = x1 = x2 = good(r - 2.5u);
top1y1 = m + oo; bot1y1 = -oo; y1 = y1; top1y1 = m - 0.2u;
call "a dare(i, 2, w2); call "b dare(1, 3, w3);
w1 draw 4;
call "c exit(5, r);
lpen; w1 draw 4..5.

"italic letter b";
call charbegin("b, 8, mc-lbowl, -mc-rhook, pl, 0, mi[rhook, 0]);
hpen; x1 = x2 = good, 1.5u; x1 = good(r - 1.5u); x1 = x1 = 5{x2, x3};
top1y1 = h; y1 = .5{y1, y1}; top1y1 = m + oo; bot1y1 = -oo; y1 = y1;
call "a serif(i, 4, -{cs);
w1 draw 1..2;
call "b arc(3, 2, w3); call "c arc(1, 2, w1);
call "d dare(3, 5, w2).

```



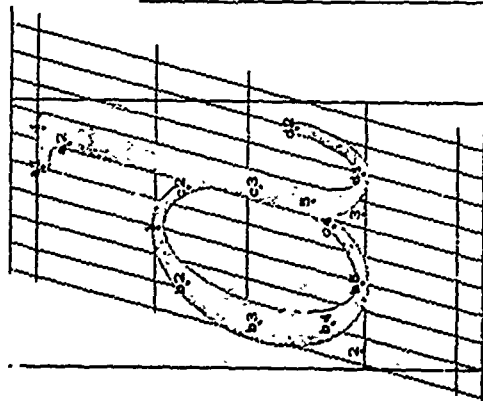


```
"italic letter c";
call charbegin("c, 8, mc-bowl, -1 mc-px-slant, px, 0, ml-px-slant - pu, 0));
hpen; rt2x1 = rt2x2 = round(r - 1.5u); x1 = x2 = .5(r + u); x1 = good2 1.5u;
rt2x2 = r - .5u; top2x2 = .5u; x2 = x0; y2 = c;
y1 = y2 = .5(c, m); top2y1 = m + oo; bot2y1 = -oo; y1 = y1;
w0 draw 2{0, 1}..3{-1, 0};
call "a dare(3, 4, w2);
w0 draw 5{1, 0}..6{.7};
open; w1 draw 1.
```

% shoulder
% bowl
% point
% bulb

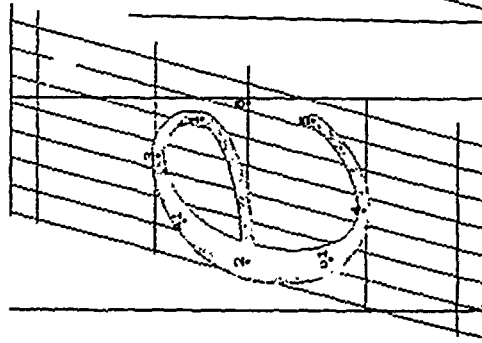
```
"italic letter d";
call max(ph-slant + .5pu - 2pu, rt2hook);
call charbegin("d, 9, mc-bowl, -1 mc-px-slant, px, 0, ml-px-slant - pu, 0));
hpen; x1 = .5(x2, x1); x2 = good2 1.5u; x1 = x1 = x2 = good2(r - 2.5u);
top2y1 = h; top2y2 = m + oo; bot2y2 = -oo; y1 = y2;
call "b dare(1, 2, w2); call "c dare(1, 3, w0);
call "d exit(5, r);
call "a serif(4, 1, 5, -1cs);
w1 draw 4..5.
```

% bowl
% closing hook
% serif
% stem



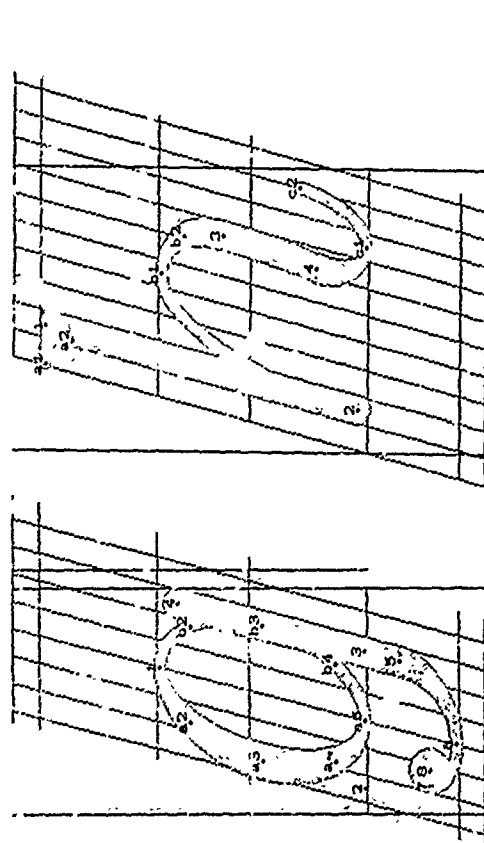
```
"italic letter e";
call charbegin("e, 8, mc-bowl, -1 mc-px-slant, px, 0, ml-px-slant - pu, 0));
hpen; rt2x1 = round(r - 1.5u); x2 = good2 1.5u;
x1 = x1 = 5(r + u); rt2x2 = r - .5u; x0 = x2;
y1 = .5(c, m); y2 = c; top2y1 = m + oo; bot2y1 = -oo; top2y2 = .5c, y2 = c;
w0 draw 2{1, 0}..3{-1, 0}..3{-1, 0}..3{-1, 0};
call "a arc(3, 2, w2); call "b arc(4, 2, w2);
draw 4{1, 0}..5{.6}.
```

% bar and shoulder
% bowl
% point



```
"italic letter f";
call charbegin("f, 7, 1 - fixwidth - mc(pd-slant + pu),
1 - fixwidth - mc(px-slant), ph, pd, ph-slant + 75pu - ml-px-slant);
open; lt, x1 = lt2x2 = round(.75u); rt, x1 = rt2x1 = round(r - 75u);
x2 = r - 2.25u; x0 = good2(.5r - .5u); x1 = good2(.5r + .5u); x2 = 2.25u;
bot2y1 = -.9d; y2 = y1; y1 = y1; y2 = y2 - y1;
top2y1 = h + oo; y1 = y1 = y2 - y1; bot2y1 = -.1d; bot2y2 = -d - oo;
x0 = x0 - 2.25u - eps; x10 = x0 + 1.75u + eps; top2y2 = m; y1 = y1;
w1 draw 1, draw 3;
w10 draw 9..10;
hpen; draw lw1{0, 1} lw1{5{-1, 0} lw1{6{x1 - x0, y1 - y0}}
lw1{7{x1 - x0, y1 - y0}} lw1{8{-1, 0}..2{0, 1}}.
```

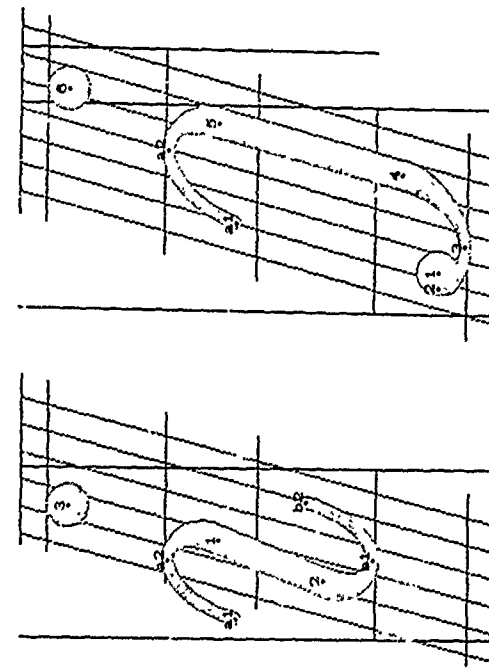
% bulbs
% bar
% stem



```

"italic letter g";
call max(-bowl, pd-slant - 1.5pt);
call charbegin{ g, 8, -mc-acc, -mc(r-tem - 1/2px slant), px, pd,
  mlfstem, 1/2px slant };
open; x1 = x0 = 5r; x2 = good 5u; x3 = x1 = x2 = good 1/2r - 1.5u;
mflx2x = mflx2y = round 2u;
top0y1 = m + 0r; bot0y2 = -0o; y1 = y2; top1y1 = m - 02h; y2 = -1d;
bot0y3 = -d - 0o; bot0y4 = -9d; y3 = y4;
w1 draw 8;
w1 draw 4;
call ~a darc(1, 2, w2);
call ~b darc(1, 3, w2);
hpen; draw |w1|4...|w1|5{0, -1} |w1|6{-1, 0} 7{0, 1}.
"italic letter h";
call charbegin{ h, 9, 0, -mc rhook, ph, 0, mlfhook, 0 };
open; x1 = x2 = good 1.5u; x3 = good 1/2r - 2.5u; x4 = x1 = 25u;
top0y1 = h; bot0y2 = -0o;
w1 draw 2;
call ~a scif(1, 1, 2, -les);
hpen; w1 draw 1..2;
call ~o italstroke(2, 3);
call ~c skewexit(4, r);
draw 3{3, -1}..4{-u, -m}.

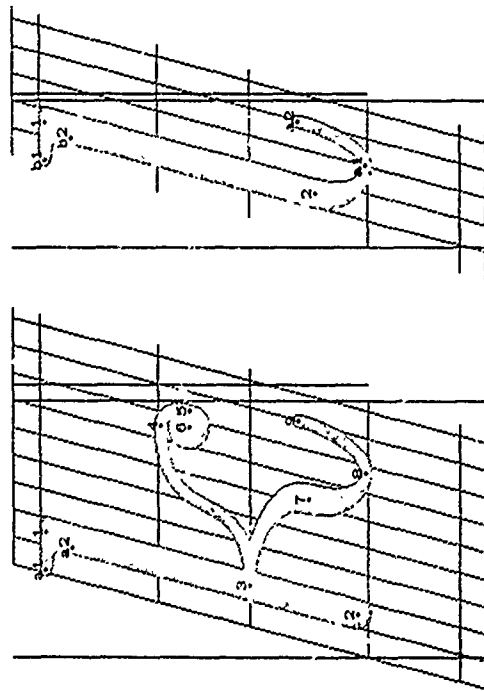
```



```

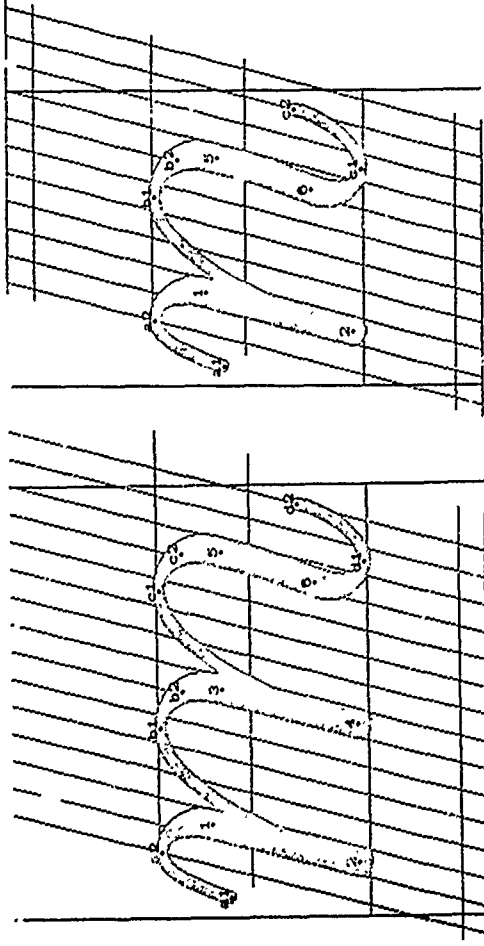
"italic letter i";
call max(rhook, ph-slant + 5pwi - 2pt);
call charbegin{ i, 1, 1, -fixwidth + mflhook, 1 - fixwidth - mc-acc, ph, 0, mlfacc, 0 };
x1 = 5r + 25u; x2 = 5r - 25u;
call ~a skewexit(u, 1);
call ~b skewexit(2, r - u);
hpen; w1 draw 1..2;
open; top0y1 = h; x1 = 5r; w1 draw 3
"italic letter j";
call charbegin{ j, 7.5, 1 - fixwidth - mc(ph + pd-slant),
  1 - fixwidth - mc(ph-slant + 5pwi - 1.5pt), ph, ph,
  mlfph slant + 1/2pwi - 1.5pt, (ph - pr) slant };
open; mflx2x = mflx2y = round 5u; x1 = 7.5u; x2 = good 1.5u; x3 = x1 = x2 = -1d;
top0y1 = h; bot0y2 = -9d; y1 = y2; bot0y3 = -d - 0o; bot0y4 = -1d;
w1 draw 1;
draw 6;
call ~a entry(u, 5);
hpen; draw |w1|5...|w1|4{0, -1} |w1|3{-1, 0} 2{0, 1}

```



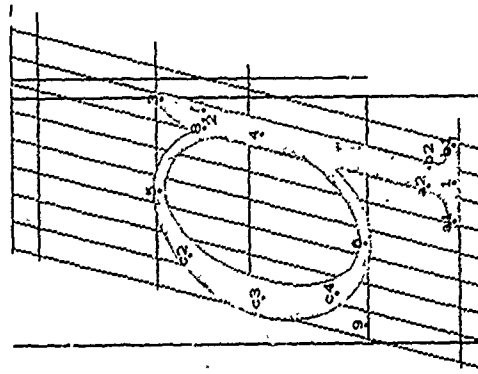
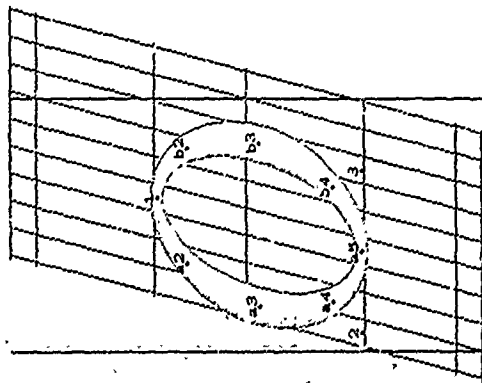
```
"italic letter k";
call charbegin("k, 8, 0, --mc-rhook, ph, 0, m[px-slant, px slant - rhook]);
open; x1 = x2 = x3 = good, 1.5u; x1 = r - 1.5u;
rt, x0 = rt, x3 = round(r - 5u); x7 = good, (r - 2.75u);
x8 = r - 1.25u; x9 = good, r;
top, y1 = h; bot, y2 = -oo; y1 = good, r; top, y1 = m + oo; y2 = y1 - 1/2(c, m);
y1 = .5c; bot, y2 = -oo; y1 = 1/2m;
w1 draw 2;
w2 draw 6;
call "a serif(1, 1, 2, -lcs);
hpen; w1 draw 1..2;
w2 draw 3(1, 0) .4(1, 0) 5(0, -1);
draw {w1#3(1, 0) .4(1, 0) .5(0, -1)};
{w1#8(1, 0) .9(0, 1)};

"italic letter l";
call m[x(rhook, ph, slant + .5nw - 2pu);
call charbegin("l, 6, 1 - fixwidth, 1 - fixwidth --mc-rhook, ph, 0, acc - m-rhook);
hpen; x1 = x2 = good, 2.5u; top, y1 = h;
call "a exit(2, r - u);
call "b serif(1, 1, 2, -lcs);
w1 draw 1..2
```



```
"italic letter m";
call charbegin("m, 15, mc-lhook, --mc-rhook, px, 0, m[rhook, 0]);
open; x1 = x2 = good, 2.5u; x3 = x1 = good, 5r; x5 = x1 = x2 - x1, x0 = x, 25r;
bot, y2 = -oo; y1 = y2; w1 draw 2; draw 4;
call "a entry(0, 1);
hpen; w1 draw 1..2;
call "b italicstroke(2, 3);
draw 3..4;
call "c italicstroke(4, 5);
call "d skewexit(6, r);
draw 5(0, -1) .6(-u, -m).

"italic letter n";
call charbegin("n, 10, mc-lhook, --mc-rhook, px, 0, m[rhook, 0]);
open; x1 = x2 = good, 2.5u; x3 = good, (r - 2.5u); x5 = -x, -25u;
bot, y2 = -oo; w1 draw 2;
call "a entry(0, 1);
hpen; w1 draw 1..2;
call "b italicstroke(2, 5);
call "c skewexit(6, r);
draw 5(0, -1) .6(-u, -m).
```

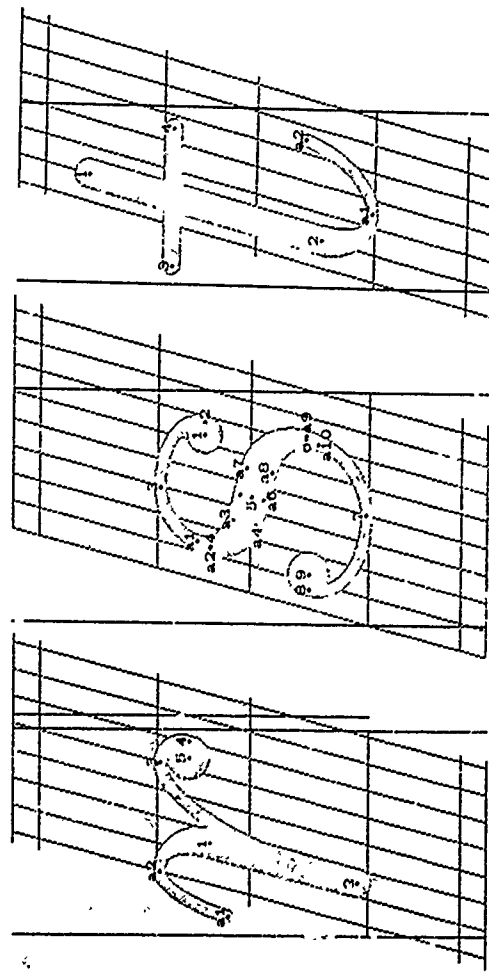
```

"italic letter o";
call charbegin("o, 9, mc-lbowl, -mc-rbowl, px, 0, m[rbowl, 0]);
% axis of left-right symmetry
hpen; x1 = r - x1;
x2 = good, 1.5u;
x1 - x2 = x3 - x1; topoy1 = m + oo; botoy2 = -oo; y2 = y1;
% left part of bowl
% right part of bowl
call "a dare(1, 2, w2);
call "b dare(1, 3, w2);

"italic letter p";
call max(-hook, pd-slant + 5pu - 2pu);
call charbegin("p, 9, -mc-acc, -mc-rbowl, px, pd, m[rbowl, 0]);
x1 = x2 = x1 = good, 2.5u; x1 = .5[x1, x2]; x1 = good, 2(r - 1.5u);
hpen; botoy1 = -d; topoy1 = m + oo; botoy2 = -oo; y2 = y1;
% opening hook
% stem
% serif
% left part of bowl
% right part of bowl
call "a entry(0, 1);
w1 draw 1..2;
call "b serif(2, 1, 1, -5les); call "c serif(2, 1, 1, les);
call "d dare(3, 4, w2);
call "e dare(3, 5, w2);

"The letter q";
call charbegin("q, 10, mc-lbowl, 1 - fixwidth - mc(rstem - 1/2 px-slant),
px, pd, m[rbowl, 1/2 px-slant]);
hpen; x1 = good, 1(r - 2.5u); x2 = 5(r - u); x2 = x1 = x1, botoy1 = -d,
rtotz = rtotz; x3 = x1; x4 = x2; if(x3 = m, x4);
topoy1 = m; y1 = 5[y1, y2]; topoy2 = m + oo, botoy2 = -oo;
new aa; if(x2 = aa[x1, x2]);
y2 = y1 = y2 = (sqrt(1 - aa[x1, x2]))[y1, y2];
if les > 0: call "a serif(1, 1, 2, -les);
call "b serif(1, 1, 2, les);
fi;
w1 draw 1..2;
w2 draw 7..8, 8(u, 1) 3{x1 - x2, 5(y1 - y2)};
w3 draw 6(1, 0), 4(0, 1), 5(-1, 0);
if w2 > 1.5u: if(x2 = round 75u;
else: x2 = good, 1.5u;
fi;
y1 = y2; call "c dare(5, 9, w2);

```



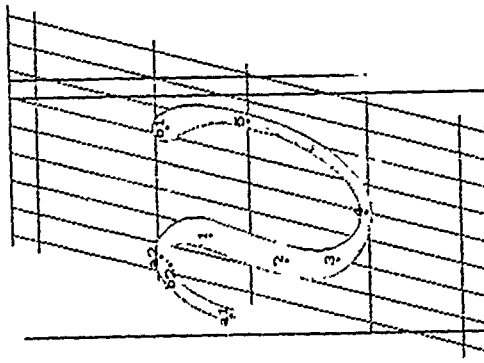
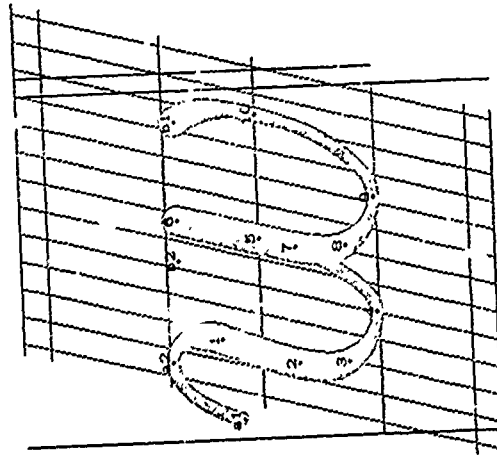
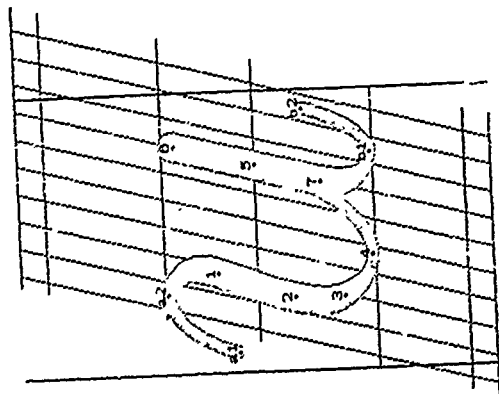
```

"italic letter r";
call charbegin("r, 7, mc-hook, -mc(px-slant - 75pu),
px, 0, ml{px slant - 25pu, 5pu});
cpn; x1 = x2 = good 1.25u; x3 = r - 2u; r1x3 = r1x1 = round(r - 75u);
bot1y2 = -oo; top1y3 = m + oo; y4 = y5 = 1/2{e, m};
w1 draw 2;
w1 draw 5;
call "a entry(0, 1);
hpen; w1 draw 1..2;
w0 draw 2{0, 1}..3{1, 0}..4{0, -1}.
% make end point round
% bulb
% opening hook
% stem
% stroke

"italic letter s";
call charbegin("s, 7, 0, -mc(px-slant - .5pu), px, 0, ml{px slant - 5pu, 0});
cpn; r1x1 = r1x2 = round(r - u); x1 = 3u; lf1x1 = round u;
x3 = 5r; r1x1x6 = round(r - 5u); x7 = r - 3u; lf1x3 = lf1x4 = round .5u;
y1 = y2 = 5/6{m}; top1y3 = m + oo; bot1y7 = -oo; y8 = y9 = 5e,
(y5 - w0 - .5w3)/(m + oo - 2w0 - w3) = e/m;
w2 draw 1; w1 draw 9;
hpen; w0 draw 2{0, 1}. 3{-1, 0}, draw 8{0, -1} .7{1, 0},
new aa;
if m - e > e. aa = m - e;
else. aa = e;
fi;
call "a sdraw(3, 4, 5, 6, 7, w1, w0, --aa/(18u)).
% stroke

"italic letter t";
cpn; 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 - pr, 0, ml{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, ml{acc, 0});
fi;
x1 = x2 = good 1.3u; x3 = x1 - 2.5u - eps; x4 = x1 + 2.5u + eps;
w1 draw 1;
top1y3 = m; y1 = y2 = y3; w0 draw 3 4,
call "a ext(2, r - u);
hpen, w1 draw 1 .2.
% make end point round
% bar
% closing hook
% stem

```



```

%italic letter "v",
call charbegin("v, 9.5, mc lhook, -mc rhook, px, 0, m[rhook, 0]),
open; x1 = .25u = x2 = good1 2.5u; x3 = x2 + .5u; x4 = 5[x2, x3];
x5 = x6 = x7 = good1(r - 2.5u);
y2 = .7c; y3 = 25[y1, c]; botdy1 = -oo; y5 = c; topdy = m + oo,
w1 draw 6;
call "a skewentry(0, 1);
lpen; draw [w1]{(-u, -m) | w1#12{0, -1} | 75[w1, w1]3...
| w1#14{1, 0} | 5{0, 1},
call "b endv(7, r);
w1 draw 6..7.

%stroke
%closing hook
%stem

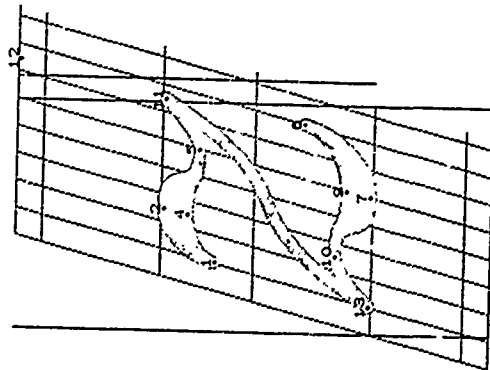
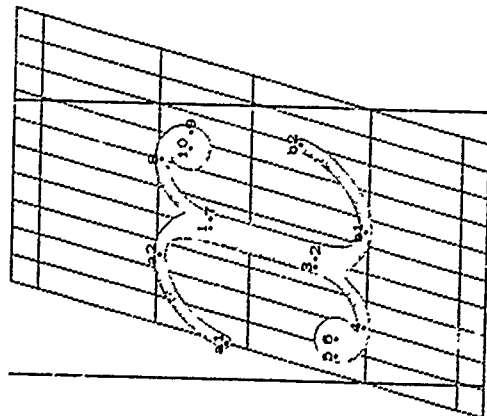
%italic letter "w",
call charbegin("w, 8, mc lhook, -mc px slant, px, 0, m[px slant, 1px slant]),
x1 = good1 2.5u; x2 = x1 + .25u; x3 = x2 + .5u;
call "a skewentry(0, 1);
lpen; y2 = .7[y1, c]; y3 = 25[y1, c]; botdy1 = -oo; x4 = .5u,
rtoz = round(r - 5u); y5 = c;
draw [w1]{(-u, -m) | w1#12{0, -1} | 75[w1, w1]3
| w1#14{1, 0} | 5{0, 1};
call "b endw(5);

%stroke
%closing bulb

%italic letter "w",
call charbegin("w, 12, mc lhook, -mc px slant, px, 0, m[px slant, 1px slant]),
open; x1 = .25u = x2 = good1 2.5u; x3 = x2 + .5u; x4 = x1 - 6[x2, x3];
x5 = x6 = x7 = good1 6.75u; x8 = x7 + .5u; x9 = r - 2.75u; rtoz = round(r - 5u),
y1 = .7c; y2 = 25[y1, c]; botdy1 = -oo; y3 = c; topdy = m + oo,
w1 draw 6;
y5 = y1; y6 = y1; y7 = y1; y8 = y1;
call "a skewentry(0, 1);
lpen; draw [w1]{(-u, -m) | w1#12{0, -1} | 75[w1, w1]3
| w1#14{1, 0} | 5{0, 1},
draw [w1]{6..|w1#17{0, -1} | 75[w1, w1]8 | w1#19{1, 0} | 10{0, 1},
call "b endw(10).

%opening hook
%left stroke
%right stroke
%closing hook

```



```

"italic letter v";
if px slant > rhooh: call charbegin( x, 8, 0, -mc-px-slant, px, 0, m[px-slant, 0]);
else: call charbegin( x, 8, mc lhook, -mc-rhook, px, 0, m[rhook, 0]);
fi;
open; x1 = x2 = good, 5r, lf1x2 = lf0x2; x3 = 2u, lf1x3 = lf1x2 = round 5u,
rt1x1 = rt0x1; x4 = r - 2u; rt0x2 = rt1x2 = round(r - 5u);
y1 = y2; bot0y1 = -oo, y3 = y0, top0y4 = m + oo; y5 = y0,
3x - y0 = y1 - y0; y6 = y1, top0y4 = m + oo; y7 = y0,
call "a entry(0, 1);
call "b exit(2, 7);
w3 draw 6; draw 10;
hpen; w0 draw 3{0, -1} 4{-1, 0} .. 5{0, 1};
draw 7{0, 1} 8{1, 0} .. 9{0, -1};
w1 draw 1..2.

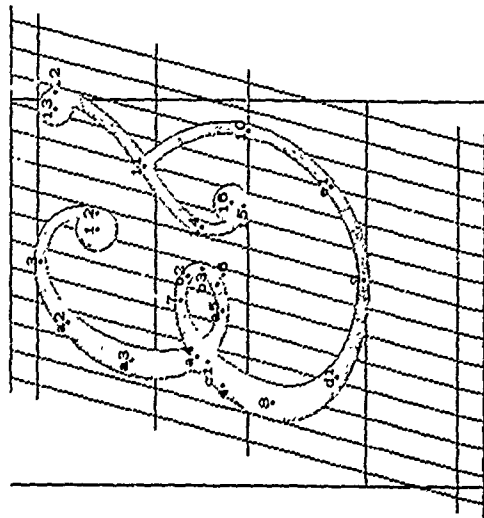
"italic letter y",
call charbegin( y, 8.5, mc lhook, -mc(rstem - 1/2 px slant),
px, pd, m[stem, 1/2 px-slant]),
open; x1 = 25u = x2 = good, 2.5u, x3 = x2 + .5u; x4 = 5[x2, x1],
x5 = x0 = x7 = good(r - 1.5u); x6 = 5r, lf0x0 = lf1x0 = round 2u,
y2 = 7e; y3 = 25[y0, e]; bot0y1 = -oo, y5 = e; top1y0 = m + oo,
w1 draw 6;
y7 = 0; bot0y4 = -d - oo; y8 = y0, bot0y0 = -75d, w1 draw 10,
call "a skewentry(0, 1);
hpen; draw w0[1]{-u, -m} .|w0#2{0, -1} | 75[y0, w0]3 .
|w0#4{1, 0} .. 5{0, 1};
w1 draw 6..7;
call "b arc(8, 7, w1);
w0 draw 8{-1, 0} 9{0, 1}.

```

```

"italic letter z";
call charbegin( z, 7, 0, -mc rhook, px, 0, px slant + 5pw - m rhook),
vpen; x1 = good0u, x2 = x1 = 2.5u, x3 = 5u;
x0 = good0r; x4 = x0 = 5u; x5 = 2.5u;
x11 = x12 = good0(r - .5u); x13 = x11 = x1,
y1 = 75m; top0y1 = top0y1 = m + oo; bot0y1 = bot0y1, y7 = 825m,
y0 = 1m, bot0y1 = bot0y1 = -oo, top0y4 = top0y4, y0 = 175m,
y11 = y1, y12 = y11 + (h + b - m); y13 = y1, y11 = y11 + (h + b - m),
w0 ddraw 1{0, 1} 2{1, 0} 5{1, 0} 1{0, 1} 4{1, 0} 5{1, 0},
ddraw 10{1, 0} 9{1, 0} 6{0, 1} 10{1, 0} 7{1, 0} 6{0, 1},
draw 5{1, 0} .11{ 12}, draw 11 13 10{1, 0},
draw 11{x13 - x10, 2(y11 - y11)} 13{x11 - x10, 2(y11 - y11)}

```



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; r1x1 = r10x2 = round .5(r + u); y1 = y2 = 5(e, h);

x1 = 4.5u; top0y = h + oo; ;

if1x1 = round 1.25u, top0y1 = round 1(e, m); x1 = x1;

r10x2 = round 6u; y6 = y1; bot0y = round 1(e, m);

if2x1 = round u; x1 = 7u; y8 = 5(y1, y6); bot0y = -oo;

r10x10 = round(r - 1.5u); y10 = e;

w1 draw 2(0, 1) .. 3(-1, 0);

call 'a darc(3, 4, w1);

call 'b darc(7, 6, w1);

call 'c arc(7, 8, w2); call 'd arc(9, 8, w2);

call 'e arc(9, 10, w3);

x11 = 9u; y11 = 5(e, h);

r1x13 = r10x12 = round(r - 2u); y12 = y13;

r10x11 = round(r - 5.5u); y11 = .5(r, m);

x13 = x10 = r - 4.5u;

open; top2y13 = h; bot2y13 = bot0y13 = round e;

w2 draw 1; w3 draw 13; draw 16;

hpen; w1 draw 10(0, 1) .. 11(2(x11 - x10), y11 - y10);

draw 12(0, -1) .. 13(2(x11 - x12), y11 - y12) .. 14(0, -1) .. 15(1, 0).

"italic numeral 0";

% (Same as in the roman font, except for spacing.)

call charbegin('0, 9, 0, ph, pd, ph-slant - 5pu);

if fixwidth = 0: new save; save = sqrttwo; new sqrttwo;

sqrttwo = sqrt(1/2314413save);

fi;

hpen; x2 = good2 1.5u;

x1 = r - x2;

x1 = r - x2; top0y1 = h + oo; bot0y2 = -oo; y1 = y1;

call 'a darc(1, 2, w1); call 'b darc(1, 3, w2);

if fixwidth = 0: new sqrttwo; sqrttwo = save;

fi;

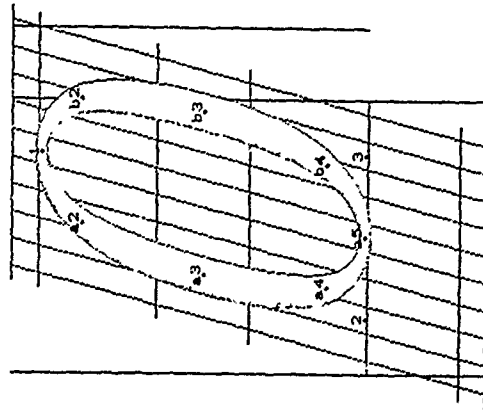
% shoulder
% upper bowl
% loop
% lower bowl
% link

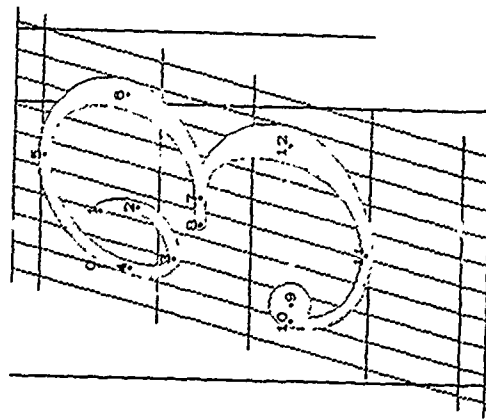
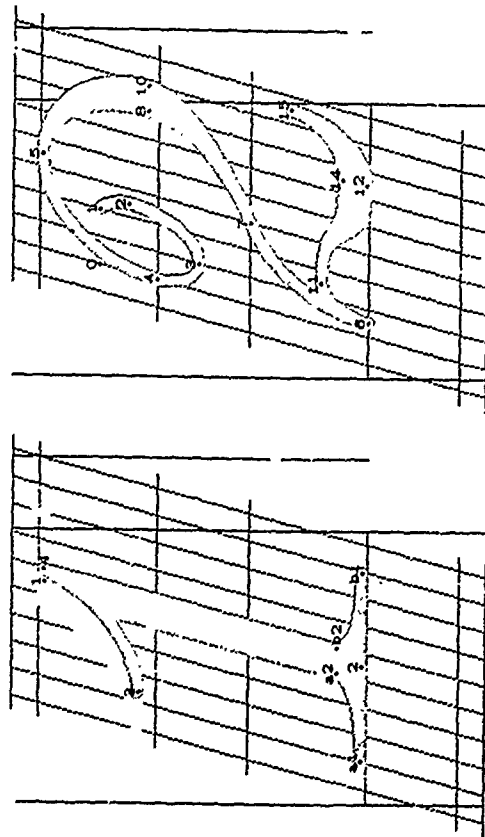
% bulbs
% stem
% arms

% the constant is 2/10

% axis of left-right symmetry

% bowl





"italic numeral 1":
call charbegin(-1,9,0,0,ph,pd,ph slant - .5pu);
lpen; x1 := good1.5r; top2y1 = h, bot2y1 = 0;
w2 draw l -.2;
call "a serif(2,2,1,-3);
top1y1 := 2[m,h]; x1 = lf2x1 - 2.5u - eps;
y1 = y1; rt2x1 = rf0x1; y1 = 1.5[pt,h]; x1 = x1;
lpen#; w2 draw (5...)4 .3{-1,0};
lpen; w0 draw (5...)4 3{-1,0}.

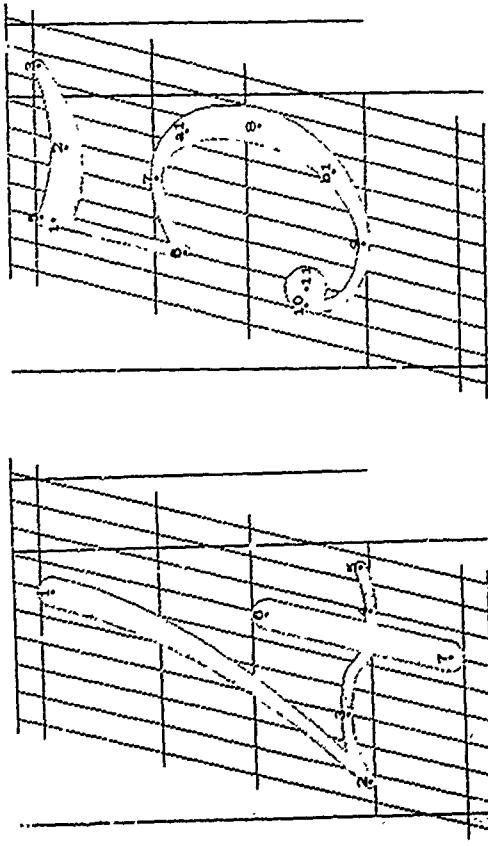
% stem
% serif
% erase excess
% point

"italic numeral 2":
call charbegin(-2,9,0,0,ph,pd,ph slant - .5pu);
lpen; x0 = u; w0 = .5[m,h], x1 = 3u; y1 = y0, x2 = good0.3.5u; y2 = 2.5[pt,h];
x3 = 2u; bot0y1 := round(.5[e,m],
x1 = good0.6; y1 = m; x3 = 5r, top0y1 = h + oo,
x0 = good{x - 1.5u}; lf0,x3 = lf0,e4, rt2,x3 = rf0,x1o,
x1 = 4u; y1 = e; x0 = good0.1.5u; bot0y1 = --o,
x1 = 2.5u; y1 = 4e;
vpen; bot0,y1 = --oo := bot0,y2, top,y1 = top0,y1;
x15 = good0(r - .5u); y15 = f.e,
lpen; w0 draw (0..1)2{0,-1} .3{-1,0}.4{0,1} 5{1,0};
draw w0 draw (0..1)2{0,-1} .7{2x1 - x1o},y1 = y1o,
5{1,0}.10{0,-1}..7{2x1 - x1o},y1 = y1o;
draw 7{2{x1 - x1o},y1 = y1o} 6{0,-1},
draw 6{0,1} 11{1,0};
draw 11{1,0} .. t4{1,0}..t5{0,1},11{1,0} .. t2{1,0} .t5{0,1}

% shoulder
% bowl
% link
% left part of bar
% bar

"italic numeral 3":
call charbegin(-3,9,0,0,ph,pd,ph slant - .5pu);
lpen; x0 = u; y0 = .5[m,h], x1 = 3u, y1 = y0; x2 = good0.3.5u, y2 = 5[y1,y1],
y1 = y4 = 52h; x1 = 2u; bot0y1 = round 1.25[pt,h];
lf0,x1 = round u, y1 = 1[u,y],
x3 = .5r; top0y1 = h + oo; rt,x0 = round{r - u}, y1 = .75h; x7 = 5r, x4 = x7 - u;
lf0,x0 = lf0,x1o = round.75u; y0 = y1o;
cpen; top3y1 = .3h, w1 draw g;
lpen; bot0y1 = --oo, x11 = 5[em,x1,d]; rt,x12 = round{r - u}, y12 = .25h;
w0 draw (0..1) 2{0,-1} 3{-1,0} 4{0,1} 5{1,0},
draw lw0#5{1,0} ..lw0#6{0,-1} . lw0#7{-1,0};
draw 7 .8;

% bulb
% shoulder
% upper bowl
% bar
% lower bowl
% link



```

"italic numeral 4";
call charbegin(4, 9, 0, 0, ph, pd, ph-slant - 5pu);
open; top_y1 = h + oo; x1 = good_5u; lift_x1 = round 5u; bot_y2 = 0;
x1 = 3u; y1 = .08h; x1 = r - 2u; y1 = 0; r1_x1 = round(r - .25u), y1 = .03h,
x1 = x1 = 3r; top_y2 = e; bot_y1 = -d,
w1 draw 1;
draw 6..7;
lipen; draw [w1]1(0, -1) .. [w2]2(x2 - x1, y1 - y1);
w2 draw 2(2(x1 - x2), y1 - y2) .. 3(1, 0) .. 4(1, 0) .. 5(x1 - x1, 2(y1 - y1))
% make end point round
% stem
% diagonal
% bar

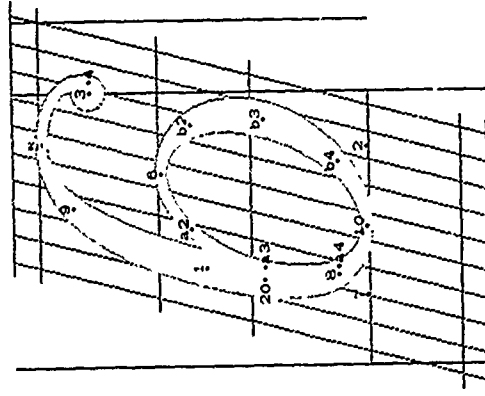
"italic numeral 5";
% (Same as in the roman font, except for spacing)
call charbegin(5, 9, 0, 0, ph, pd, ph-slant - 5pu);
open; x1 = good_2u; top_y1 = h; r1_x1 = round(r - 1.25u); top_y1 = h,
x2 = 5(x1, x1); new w1; w1 = round .75(w1, w1); top_y2 = round 95h;
x1 = -5u; x1 = r + 1.5u; y1 = y1 = 1.5h,
draw [w1]0..1) .. [w2]2(x2 - x1, y1 - y1);
lipen; x1 = x1 = x1; top_y1 = h; top_y2 = .75(e, m);
x1 = .5r; top_y1 = m + oo; x1 = x1 = x1, bot_y1 = -oo;
r1_x1 = round(r - u); y1 = 5(y1, y1);
w1 draw 5..6;
draw 9..10(7, 8, w2); call "b arc(9, 8, w2);
lift_x1 = lift_x1 = round u; y1 = y1 = 3u;
w1 draw 9(-1, 0) .. 10(0, 1)
open; w1 draw 11.

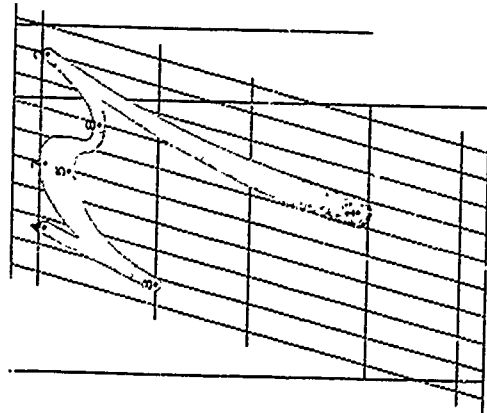
```

```

"italic numeral 6";
% (Same as in the roman font, except for spacing)
call charbegin(6, 9, 0, 0, ph, pd, ph-slant - 5pu);
x1 = good_1.5u; x2 = good_1(r - 1.5u);
new w1; w1 = round 75(w1, w2);
open; top_y1 = h - 25e,
if y1 < 5(p1, h); new y1, y1 = .5(m, h);
fi;
r1_x1 = r1_x1 = round(r - 1.5u); y1 = y1;
w1 draw 3;
lipen; x1 = good_1(x1 - 1.1u); x1 = x1 = x1 = 5(x1, x1); top_y1 = h + oo,
w1 draw 4(0, 1) .. 5(-1, 0);
bot_y1 = -oo; top_y1 = m + oo; y1 = .5(y1, y1);
y1 = y1 = y1; r1_x1 = r1_x1;
call "a darc(6, 7, w1); call "b darc(6, 2, w2);
new w1; w1 = 1(w1, w2);
x1 = x1; r1_x1 = r1_x1(1/sqrt(2)(x1, x1));
y1 = 1/sqrt(2)(y1, y1); y1 = y1 = y1 - y1; y1 = 5(y1, y1);
draw [w1]5(-1, 0) .. [w2]9(x1 - x1, y1 - y1) .. [w3]11(0, -1)
.. [w4]8(x1 - x1, y1 - y1) .. [w5]10(1, 0).

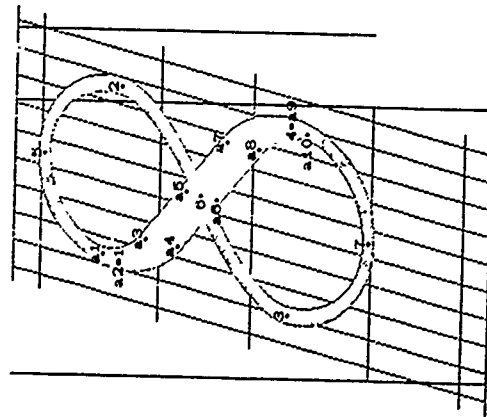
```



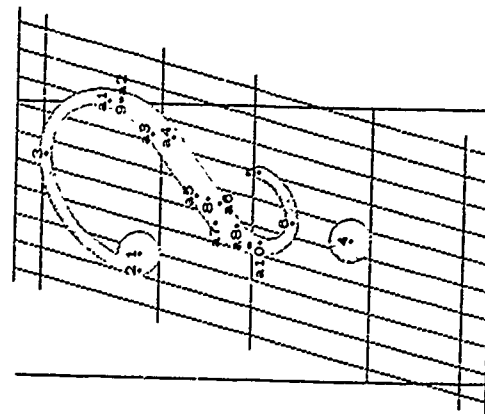
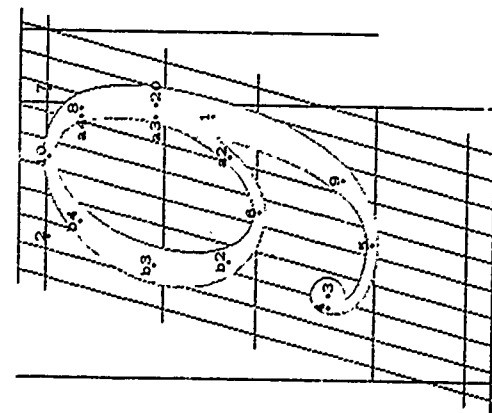


```
"Italic numeral 7";
call charbegin(7, 0, 0, ph, pd, ph.slant -.5pu);
open; x1 = good, 5.5u; bot1y1 = -oo; x2 = good0(r - u); top0y2 = h;
x3 = good0, 7.5u; y3 = m; y8 = good0, 5(m, h); y1 = y7 = y2;
(x1 - x3)/(y1 - y3) = (x2 - x1)/(y2 - y1);
top3y3 = top0y7; bot3y3 = bot0y5; x3 = x1 = 4u; x8 = 3r;
w1 draw 1;
hpen; draw 1..3, #2{2(x1 - x2), y1 - y2} |w1#1{0, -1};
n draw 4..3;
draw 8{1, 0}..2{2(x2 - x1), y2 - y1};
ddraw 3{x1 - x3, y1 - y3}..7{1, 0}..8{1, 0},
3{x1 - x3, y1 - y1}..5{1, 0}..8{1, 0}.
% make end point round
% stem
% serif
% link
% bar
```

```
"Italic numeral 8";
% (Same as in the roman font, except for spacing.)
call charbegin(8, 0, 0, ph, pd, ph.slant -.5pu);
new w38, w39, ss; w39 = round 1/3(w3, w1);
hpen; if0y1 = round u; x2 = r - x1; y1 = y2;
if0y2 = round, 7.5u; x4 = r - x; y1 = y1; x5 = r - x0 = x1;
top0y7 = h + oo; y8 = 52h; bot0y7 = -oo,
w38 = 2(w7, w8);
if w3 = w38; ss = 0;
else; ss = h/(18u);
fi;
% slope
% upper left and lower right strokes
% upper right and lower left strokes
```



```
call a sdraw(5, 1, 6, 4, 7, w39, w38, -ss);
w39 draw 5{1, 0}..2{0, -1}..6{-1, -.75ss}..
3{0, -1}..7{1, 0}.
```

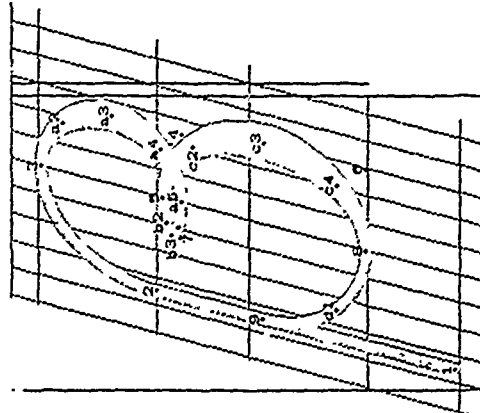
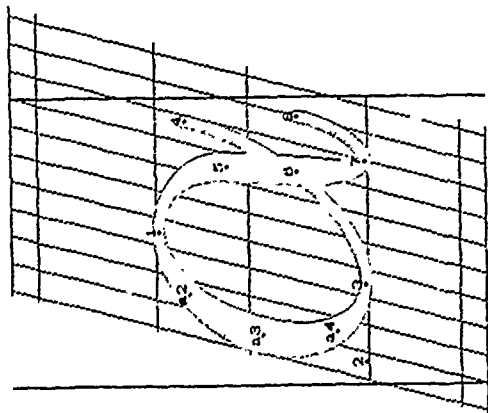



```

..italic numeral g",
% (Same as in the roman font, except for spacing.)
call charbegin(9, 9, 0, 0, ph, pd, ph slant - 5pt);
x1 = good(r - 1.5u); x2 = good(r - 1.5u);
new w0; w0 = round(.75(w0, w1),
open; bot.y1 = 25e;
if y1 > 5e: new y1; y1 = .5e;
fi;
lt.y1x1 = lt.y1x1 = round 1.5u; y1 = y1;
w0 draw 3;
open; x0 = good(x1 - .1u); x2 = x0 = x0 = 5{x0, x2}, bot.y1 = 5e;
w0 draw 4{0, -1}..5{1, 0};
top.y1 = h + .00; y0 = e - .00; y2 = 5{y2, y0},
y1 = y0 = y1; lt.y1 = lt.y1;
call "a dare(6, 7, w0); call "b dare(6, 2, w2);
new w0; w0 = {w0, w2};
x1 = x0; lt.y1x1 = lt.y1x1(sqrt(w0, x1));
y1 = 1/sqrt(w0, y1); y2 = y1 = y2 = y1; y1 = 5{y1, y0};
draw [w0, 1, 0]..[w0, 9]{x1, x0, y1 - y0}, [w0, 10]{0, 1}
..[w0, 10]{0, -1, 0} [w0, 10]{-1, 0}

"italic question mark";
call charbegin(77, 9, 0, 0, ph, 0, 0);
open; lt.y1x1 = lt.y1x1 = round u; y1 = y1; bot.y1 = m;
x2 = x1 = x0 = x2 = .5r; top.y1 = h + .00; y0 = 5e; y1 = good(r - 1.5u);
lt.y1x1 = round(r - 1.5u); lt.y1x1 = round 3u;
x2 = 3r; top.y1 = e; bot.y1 = 0;
w1 draw 4;
draw 1;
open; w0 draw 2{0, 1}..3{1, 0};
call "a sdraw(3, 9, 8, 7, 6, w0, w0, (y1 - y0)/(10u));
draw 6{1, 0}..5{0, 1};

```



T: c 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.

% quantities used in spacing corrections

new mc, lhowl, lhook, rhowl, rhook, rstem;

mc = 1/pt;

rhook = {px-slant + .5pw + 5pu;

lhook = .3px-slant - .5pw + pu;

rhowl = .7px-slant + .5pw - pu;

lhowl = {px-slant - .5pw - 5pu;

rstem = px-slant + .5pw - pu;

"Lower case Greek alpha";

call charbegin('013, 10, mc, lhowl, -mc, rhook, px, 0, 0);

lpen; x1 = x1 + .45w; lhook = round w;

topyl = m + oo; botyl = -oo; yl = yl;

call 'a darc(1, 2, w2);

rtol = round(r - u); topyl = 75[e, m];

uyl draw 3{1, 0}..4{0, 1};

x2 = r - 2.5u; x3 = r - 2u; y3 = {m; y3 = {m;

botyl = -oo; x1 = r - u; x3 = good(r; y3 = y3;

draw {uyl}{1, 0}..{uyl}{5{x0 - x1, y0 - y1}}..8{0, 1}

{uyl}{6{x0 - x1, y0 - y1}}..{uyl}{7{1, 0}}..8{0, 1}

"Lower case Greek beta";

call charbegin('014, 9, 5, mc, pu - pu-slant - .5pw);

-mc, rhowl, ph, pd, 5[px, ph-slant - pu - rhowl);

lpen; x1 = x1 = x1 = good, 1.5u; botyl = -d; yl = m;

x1 = x1 = x1 = 5u; rtol = round(r - 1.5u);

x0 = good(r - 1.5u); lhook = round 3.5u;

topyl = h + oo; botyl = m - w; topyl = m;

y3 = .5m;

uyl draw 1 2{0, 1}..3{1, 0};

call 'a darc(3, 4, w1);

call 'b darc(5, 7, w2);

call 'c darc(5, 6, w2);

call 'd arc(8, 9, w3).

% stem and shoulder

% upper bowl

% loop

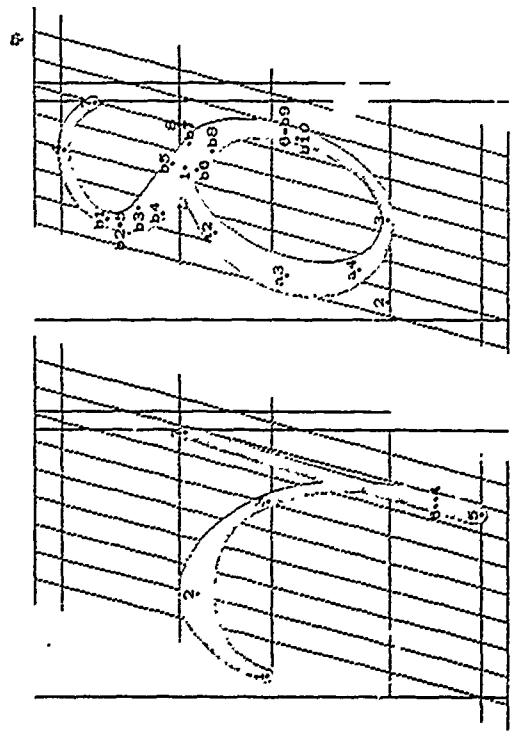
% lower bowl

% link

% left of bowl

% upper diagonal

% lower diagonal and tail



"Lower case Greek gamma",
 call charbegin(015, 10, mc {px slant - 5pu}, px, pd, 1/2 px slant);
 vpen; if(x1 == round 5u; boty1 = e, topy1 = m, x2 = 3u;
 x1 = 1/2 sqrt(2u, x1); y1 = 1/2 sqrt(2u, y1);
 r1(x1 = round(r - u); y1 = -5u, x2 = 5{x1, x0}, boty1 = -d - o,
 if(x1 = round(r - u); y1 = 1/2 sqrt(2u, y1); y1 = y1; topy1 = m;
 draw {u, 1/2 sqrt(2u, 1, 0)} {u, 1/2 sqrt(2u, 1, 0)} {x1 - x2, y1 - y2} 4{0, -1}
 5{-1, 0} 6{0, 1} 7{2(x1 - x0), y1 - m}

% stroke

"Lower case Greek delta",
 call charbegin(016, 8, mc {bowl, -mc(7px slant - 5pu), ph, 0},
 9ph slant + 5pu - 1.5pu - (7px slant - 5pu));
 hpen; x1 = 4.5u; topy1 = m, x2 = good 1.5u, y1 = y1;
 x1 = x1; boty1 = -oo; x1 = 4u, topy1 = h + oo,
 if(x1 = round(1.5u); r1(x1 = round(r - u);
 x1 = x1 = r - 2u, y1 = .3h; y1 = m,
 call "a dare(1.2 w);
 call "b sdraw(4, 5, 1, 6, 3, w11, w8, -h/(18u));
 w1 draw 4{1, 0} .. 7{.8).

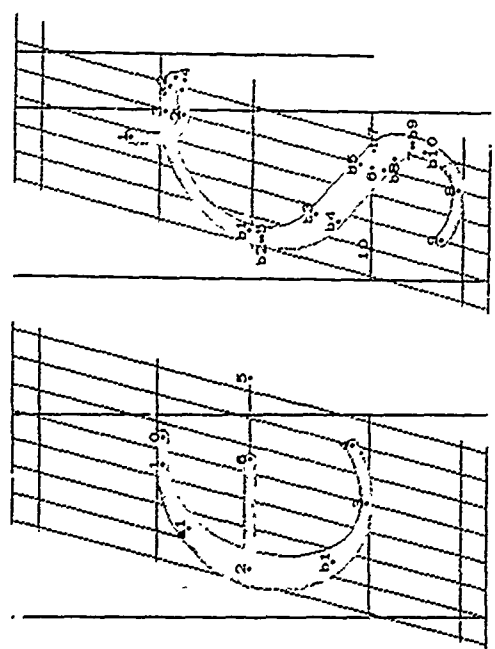
% left of bowl
 % shoulder and right of bowl
 % point

"Lower case Greek epsilon",
 call max{px slant + .5pu - 2pu, -5pu};
 call charbegin(017, 8, mc {bowl, -mc arc, px, 0, 0});
 hpen; x0 = 5u = x1 = r - 3u; x1 = 5(r + u), topy1 = m, y0 = y1, x0 = x0,
 x2 = good 1.5u; y2 = y1 = y1 = 5u; x1 = r - u; x2 = r + 5u; boty1 = -oo,
 new aa; x1 = aa{x0, x1}; y1 = (sqrt(1 - aa-aa))[m, y1];
 w1 draw 0..1;
 call "a arc(1, 2, w);
 call "b arc(3, 2, w);
 draw 3{1, 0} .. 4{5);
 draw 2..6.

% strengthen upper point
 % upper left of bowl
 % lower left of bowl
 % lower point
 % bar

"Lower case Greek zeta",
 call charbegin(020, 7, mc {bowl, 0, 25[px, ph], pd, px slant},
 hpen; if(x1 = round 3.5u; y1 = 25[m, h];
 x2 = x1 = x0 = x4 = 5u; boty1 = m - w1, y1 = y1, topy1 = m;
 r1(x1 = round(r - 5u); if(x1 = round u, y0 = 0; r1(x1 = round(r
 boty1 = -d; x1 = 3u; x0 = 2u, y1 = 75d, y0 = 0,
 w1 draw 1{0, -1} 2{1, 0};
 call "a dare(3, 4, w);
 call "b sdraw(3, 5, 6, 7, 8, w10, w8, -r/(6u));
 draw 8{-1, 0} 9{.10}

% flourish
 % loop
 % stroke
 % point



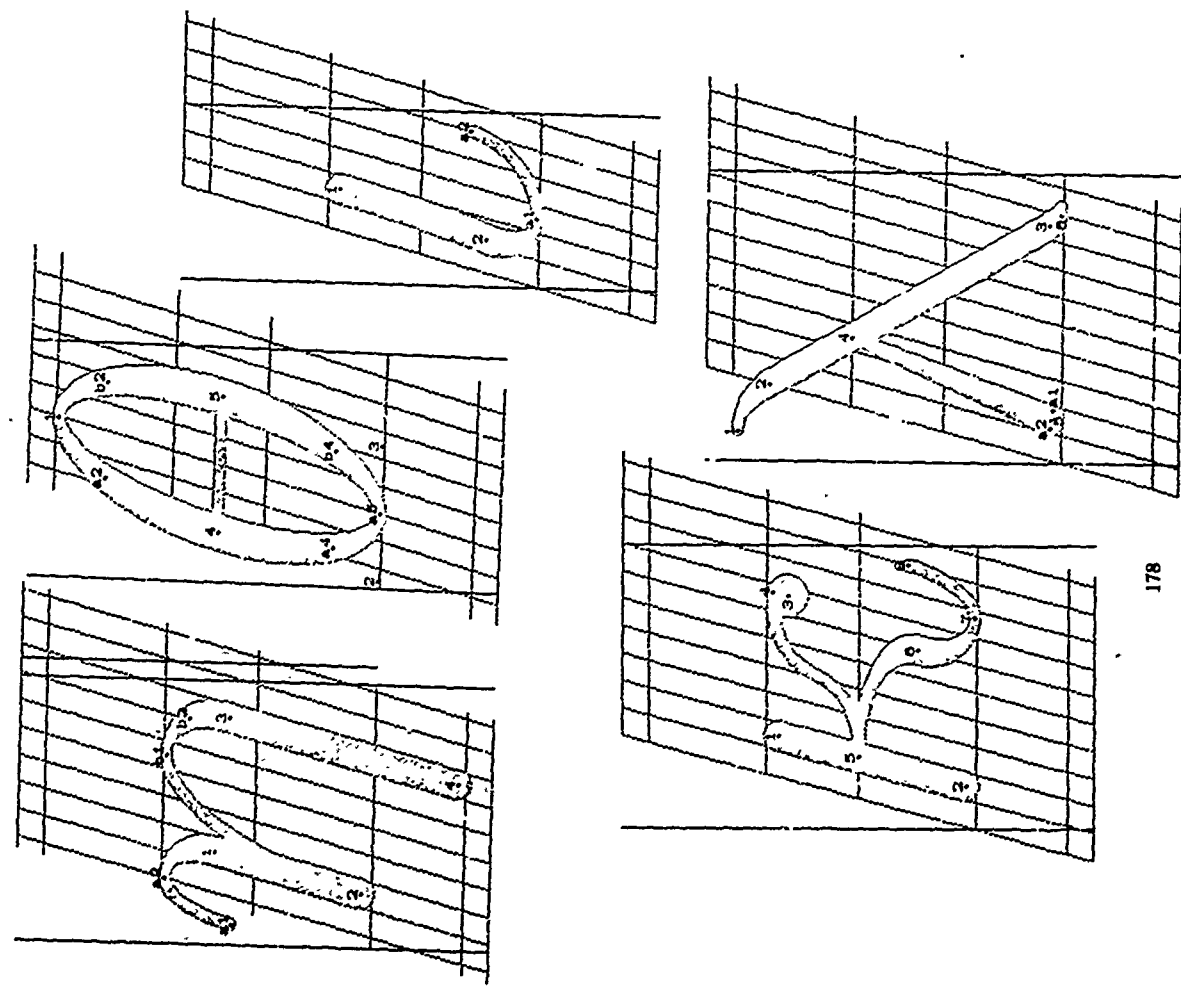
Lower case Greek eta";
 call charbegin(021,10,mc-hook,-mc(stem -- 1/3px slant),px,pd,1/3px slant);
 x1 = good, 2; call 'a entry(0,1);
 open; x2 = x1; bot1y1 = -oo; w1 draw 1..2;
 x1 = good,(r - 2.5u); call 'b itallstroke(2,3);
 x1 = x2; bot1y1 --d -o; draw 3..4.
 % opening hook
 % left stem
 % shoulder
 % right stem

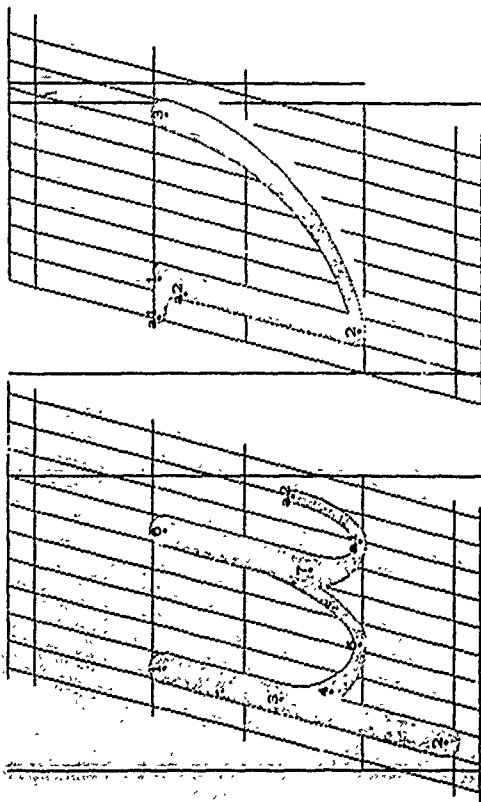
"Lower case Greek theta";
 call charbegin(022,8,mc(3ph-slant + pu -- 5pwi),
 --mc(7ph-slant -- pu + 5pwi),ph,0,0);
 x1 = r - x1;
 x2 = x1 = good, 1.5u; x1 = x2 = r - x1;
 top1y1 = h + oo; bot1y1 = -oo; y1 = y1; y1 = y1 = 5(y1,y2);
 call 'a darc(1,2,w2);
 call 'b darc(1,3,w3);
 w1 draw 4..5.
 % axis of left-right symmetry
 % left part of bowl
 % right part of bowl
 % bar

"Lower case Greek iota";
 call charbegin(023,5,0,-mc-rhook,px,0,0);
 open; x1 = good, 1.5u; x2 = x1 - 2.5u; top1y1 = m + oo;
 call 'a skewer(2,r);
 w1 draw 1{0,-1} .2{-u,-m}
 % closing hook
 % stem

"Lower case Greek kappa";
 call charbegin(024,9,0,-mc-rhook,px,0,0);
 open; x1 = x2 = x1 = good, 1.5u; top1y1 = m + oo; bot1y1 = -oo;
 top1y1 = top1y1 = m; x1 = x1; r1x1 = round(r - 1.5u);
 y1 = c, x1 = 6u, x1 = 7.75u; x1 = good,r;
 y1 = 5c; bot1y1 = -oo, y1 = 1m;
 w1 draw 1..2;
 w2 draw 3;
 lipen; w1 draw 5{1,0} .4{1,0};
 draw lip1{5{1,0} .lip1{6{0,-1} lip1{7{1,0} 8{0,1}}
 % stem
 % bulb
 % upper diagonal
 % lower diagonal

"Lower case Greek lambda";
 call charbegin(025,10,5,0,ph,0,0);
 open; top1y1 = h; x1 = -2u, x2 = 0, y1 = 7(y1,h), x1 = r 2u, y1 = 12c;
 new aa; x1 = aa{2,1}, y1 = aa{y1,y1}, y1 = m, x1 = 1.5u, bot1y1 = -oo;
 lipen; x1 = r - 1.5u, bot1y1 = -oo;
 draw lip1{1{1,0} lip1{2{x1 - x2,y1 - y1} 3{x1 - x2,y1 - y1}}
 6{2{4c - x1},u - y1},
 call 'a cdraw(5,4,1,0)
 % long diagonal
 % short diagonal





```

"Lower case Greek mu";
call charbegin('026,9,mc(pu - pd-slant -.5pw),-mc-rhook,px,pd,0);
cpn; x1 = x2 = x3 = good, 1.5u;
x1 = x1 + .75u; x2 = .55[x1,x0]; x0 = x1 = good, 6.5u;
top,y1 = m + oo; bot,y2 = -d - oo; y1 = .7c; y2 = .25[y1,d]; bot,y3 = -oo;
y0 = y1;
% closing loop
% stems
% stroke
w1 draw 1..2; draw 6..7;
w1 draw [w1#3{0,-1}..1.75[ub,w0]]4 [w1#5{1,0}..6{0,1}];
hpen; draw [w1#3{0,-1}..1.75[ub,w0]]4 [w1#5{1,0}..6{0,1}];

"Lower case Greek nu";
call charbegin('027,9,0,-mc(rstem -- {px-slant},px,0,{px-slant});
cpn; x1 = x2 = good, 1.5u; x3 = r - 1.5u;
top,y1 = m; bot,y2 = 0; top,y3 = m + oo;
hpen; w1 draw 1..2;
call 'a serif(1,1,2,-les);
rpen#; w1 draw 2{36u m} .3{0,1};
hpen; w1 draw 3;
hpen; draw [w1#2{36u,m}..[w1#3{0,1}];

```

```

"Lower case Greek xi";
call charbegin('030,9,0,0,ph,pd,.75[px,ph]-slant -- 2pu);
hpen; top,y1 = h; lt,x1 = round 3u;
x2 = x3 = 5u; bot,y2 = top,y3 = w; top,y4 = round(y1 - 25(h - m)), y1 = y4;
rt,x1 = round x2 + 1.5u; lt,x2 = round u; y2 = .5[y1,w];
x0 = x1 = 5u; bot,y3 = top,y4 = w; top,y5 = round 5u; y4 = y5;
rt,x3 = round x0 + 1.5u;
lt,x1 = round u; x2 = r - 3u; y2 = -{d; rt,x1 = round(r - 5u);
x3 = r - 2u; bot,y4 = -d - oo; x10 = x2 - 2u; y10 = -75d, x11 = x10 - 2u;
y11 = 0;
w1 draw 1{0,-1}..2{1,0};
call 'a dare(3,4,w);
call 'b arc(3,5,w);
call 'c arc(6,5,w);
call 'd dare(7,8,w);
call 'e sdraw(7,11,12,13,9,w0,w0,-c/(12u));
draw 9{-1,0}..10{...14};

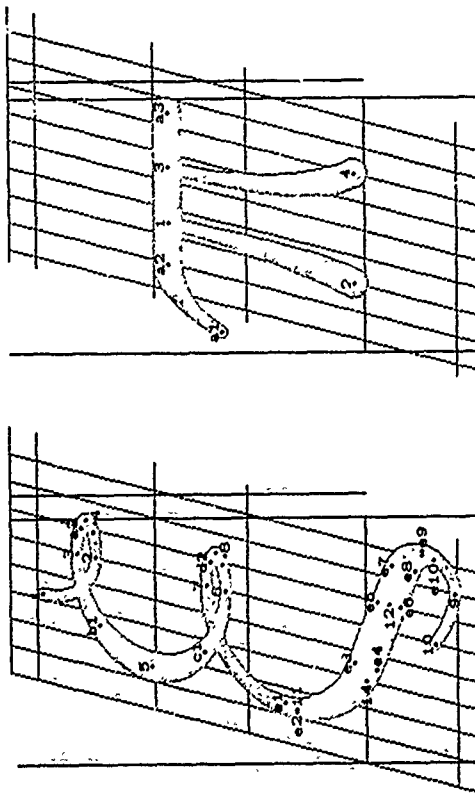
% flourish
% upper loop
% top of bowl
% bottom of bowl
% lower loop
% stroke
% point

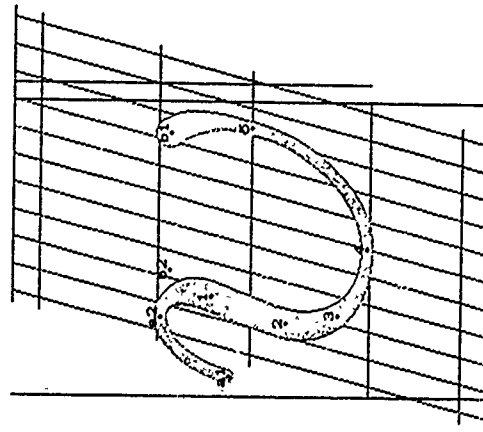
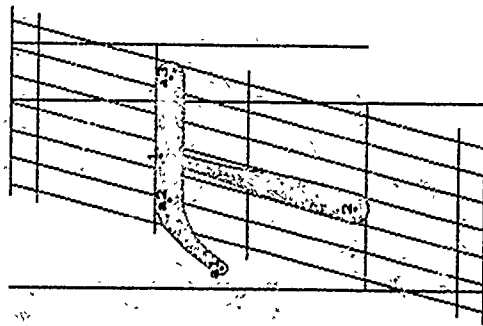
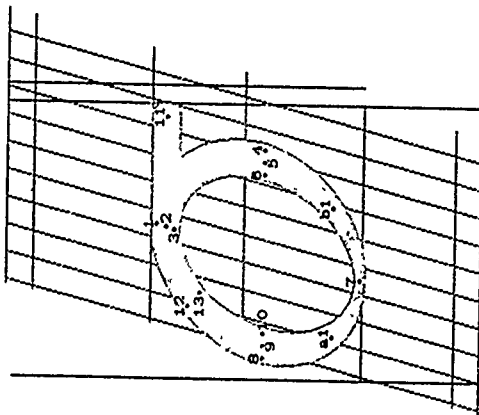
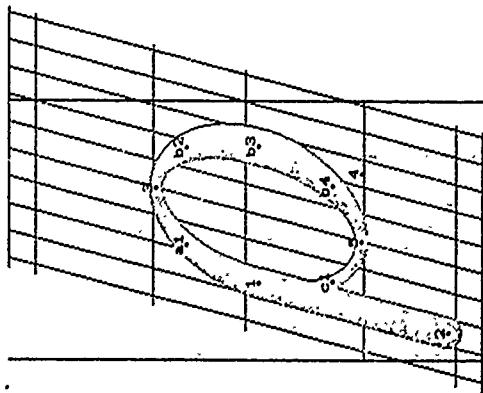
```

```

"Lower case Greek pi";
call charbegin('031,9,mc(7px-slant -- 5pw -.5pu),
-mc({px-slant -- pu + .5pw-aspect},px,0,{px-slant});
call 'a pstroke;
cpn; bot,y2 = -oo; top,y1 = m; y3 = y1; y1 = y2;
x1 = good, 3.5u; x2 = good, 3u; x3 = good, 5.5u; x4 = good, 7u;
hpen; draw [w1#1{0,-1}..[w1#2{3.14159(x2 - x1),y1 - y1};
draw [w1#3{0,-1}..[w1#4{3.14159(x1 - x2),y1 - y2};
cpn; w1 draw 2; draw 4.
% left stem
% right stem
% make the end points round

```





```

"Lower case Greek rho";
call charbegin(0.32, 8, mc(pu - pd-slant - .5pw), -mc-rhawl, px, pd, 0);
open; x1 = x2 = good, 1.5u; x1 = good, (r - 1.5u); x1 = x2 = 5[x1, x1];
bot1y2 = -d - o; y1 = .5[y1, y1]; top3y2 = m + oo; bot3y2 = -oo; y1 = y2;
w1 draw 2..1;
call 'a arc(3, 1, w1);
call 'b darc(3, 4, w2);
call 'c arc(5, 1, w3);

% stem
% upper left part of bowl
% right part of bowl
% lower left part of bowl

"Lower case Greek sigma";
call charbegin(0.33, 10, mc-rhawl, -mc({px-slant - 5pu}, px, 0, {px-slant});
open; top1y2 = top3y2 = m; bot1y2 = bot3y2; y11 = y2;
y4 = y1 = y10 = y2 = y3 = y4 = y1 = .5[y2, y2]; bot4y2 = -oo;
x1 = x2 = x3 = x4 = x5 = .5(r - u); r1x11 = r - u;
lt0x2 = lt2x2 = round 5u; r1x20 = r1x22;
lt0x2 = lt2x2; r1x21 = r1x23 = round(r - 1.5u);
x12 = 1/sqrttwo[x1, x2]; y2 = 1/sqrttwo[y1, y2];
x13 = 1/sqrttwo[x1, x3]; y13 = 1/sqrttwo[y10, y1];
w1 draw 2..11;
hpen; w1 ddraw 8{0, 1}..12{x1 - x2, y1 - y2}..1{1, 0}..4{0, -1},
10{0, 1}..13{x1 - x2, y2 - y10}..3{1, 0}..6{0, -1};
call 'a arc(7, 9, w2);
call 'b arc(7, 5, w2);

% bar
% upper part of bowl
% lower left part of bowl
% lower right part of bowl

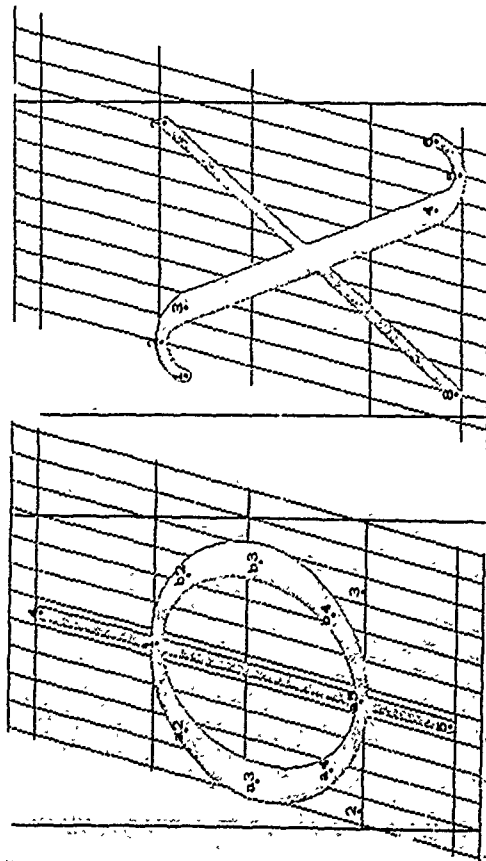
"Lower case Greek tau";
call charbegin(0.34, 8, mc(.7px-slant - 5pw - .5pu), 5,
px, 0, px-slant - .5pu + .5pw-aspect);
call 'a pstroke;
open; x1 = x2 = good, 3.5u; top4y1 = m; bot1y2 = -oo;
w1 draw 2;
hpen; draw [w1#1]..[w1#12];

% bar
% make lower end point round
% stem

"Lower case Greek epsilon";
call charbegin(0.35, 10, mc-lhook, -3mc-px-slant, px, 0, {px-slant});
x2 = good, 2.5u; x1 = x2 + .25u; x1 = x2 + .8u;
call 'a skewentry(0, 1);
hpen; y2 = .7[y1, 0]; y1 = .25[y1, 0]; bot0y1 = -oo; x1 = 6u;
r1x2 = round(-.5u); y1 = 0;
draw [w1#1](-u, -m)..[w1#12](0, -1)..1.75[w1, w1]3..
[w1#14](1, 0)..5{0, 1};
call 'b endv(5);

% opening hook
% stroke
% closing bulb

```



“Lower case Greek phi”;
 call charbegin(‘036, 11, mc-bowl, -mc-bowl, ph, pd, 0);
 $x_1 = r - x_1$;
 $x_2 = \text{good}_2 1.5u$; $x_1 = r - x_2$;
 hpen; $\text{top}_0 y_1 = m + oo$; $\text{bot}_0 y_2 = -oo$; $y_1 = y_2$;
 $\text{top}_0 y_1 = h$; $\text{bot}_0 y_2 = -d$; $x_1 = x_2 = x_1$;
 call ‘a darc(1, 2, w);
 call ‘b darc(1, 3, w);
 w draw 4..5.

% axis of left-right symmetry

% left half of bowl
 % right half of bowl
 % stem

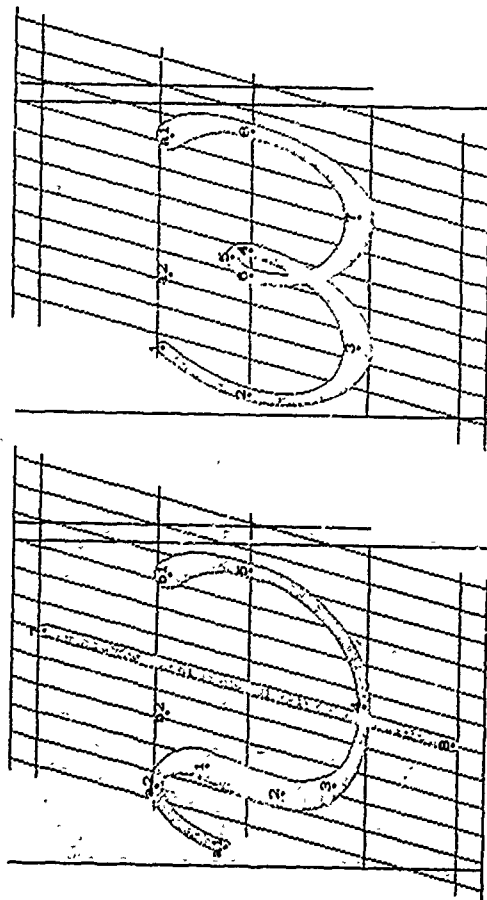
“Lower case Greek chi”;
 if $(px + pd) \cdot \text{slant} > 2pu$;
 call charbegin(‘037, 11, mc(1.5pu - pd-slant - .5pw),
 -mc(px slant + .5pw - 1.5pu), px, pd, 0),
 else. call charbegin(‘037, 11, mc(px-slant - .5pw - 5pu),
 -mc(.5pu - pd slant + .5pw), px, pd, 0);
 fi;
 hpen; $x_1 = \text{good}_0 0$; $x_2 = u$; $x_1 = 2.5u$; $x_1 = r - x_1$; $x_2 = r - x_2$; $x_1 = r - x_1$;
 $x_1 = \text{good}_0(r - 2u)$; $x_2 = r - x_1$;
 $y_1 = y_2 = .375m$; $\text{top}_0 y_2 = m + oo$; $\text{bot}_0 y_2 = -d - oo$; $y_2 - y_1 = y_1 - y_2$; $y_1 = y_2$;
 $\text{top}_0 y_1 = m$; $\text{bot}_0 y_2 = -d$;
 draw $\{w_1\} \{0, 1\} \dots \{w_2\} \{2, 1, 0\}$ $\{w_1\} \{3\} \{x_1 - x_2, y_1 - y_2\}$ $\{w_1\} \{1\} \{x_1 - x_2, y_1 - y_2\}$
 $\{w_2\} \{5\} \{1, 0\} \dots \{0, 1\}$;
 w draw 7..8.

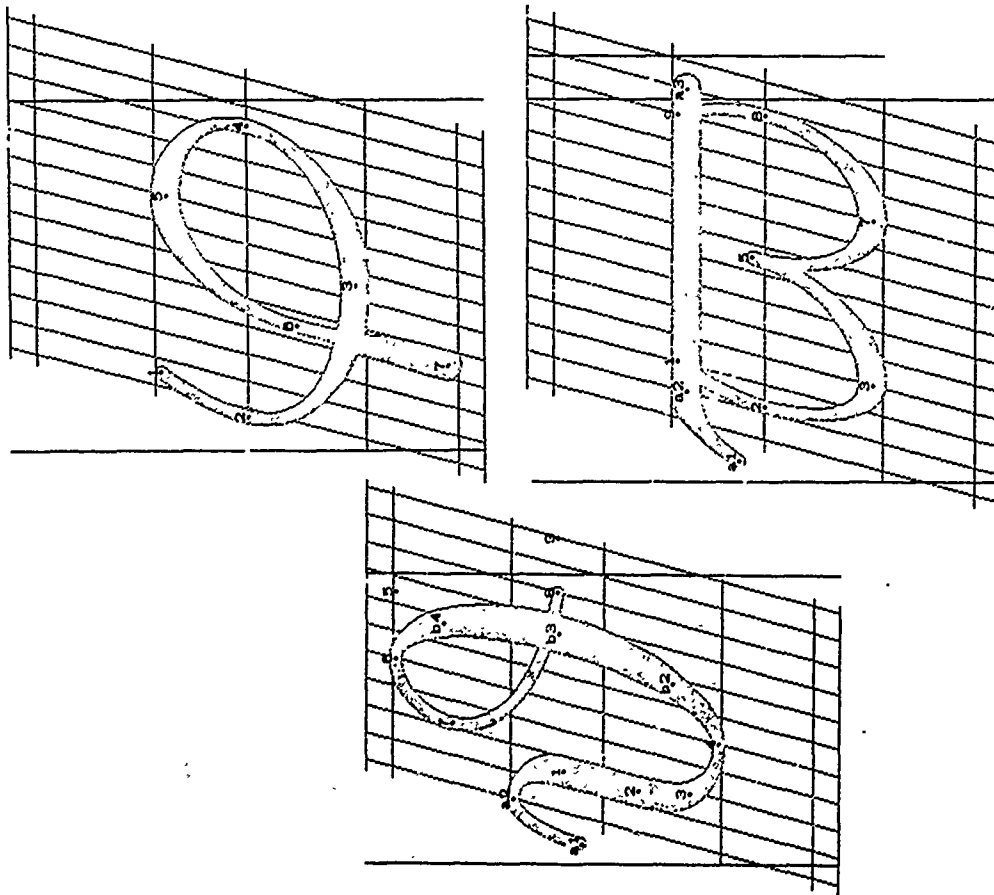
“Lower case Greek psi”;
 call charbegin(‘173, 11, mc-hook, -mc-px slant, ph, pd, {px-slant});
 $x_2 = \text{good}_1 2.5u$; $x_1 = x_2 + .25u$; $x_1 = x_2 + 8u$;
 call ‘a skewwidy(0, 1),
 hpen; $y_2 = .7[y_1, d]$; $y_1 = 25[y_1, d]$; $\text{bot}_0 y_1 = -oo$; $x_1 = 6.5u$;
 $\text{rt}_0 x_2 = \text{round}(r - .5u)$; $y_1 = e$;
 draw $\{w_1\} \{1\} \{x_1 - x_2, y_1 - y_2\}$ $\{w_1\} \{2\} \{0, -1\}$ $\{.75\{w_1, w_1\} \{3\} \dots$
 $\{w_2\} \{4\} \{1, 0\} \dots \{5\} \{0, 1\}$;
 call ‘b endv(5);
 $x_1 = x_2 = .5(r + u)$; $\text{top}_0 y_1 = h$; $\text{bot}_0 y_2 = -d$; w draw 7..8.

% opening hook
 % stroke
 % closing bulb
 % stem

“Lower case Greek omega”;
 call charbegin(‘174, 11, mc-pe slant, -mc-px slant, px, 0, {px-slant});
 hpen; $x_1 = \text{good}_0 1.5u$; $\text{rt}_0 x_2 = \text{round } 5u$; $x_2 = 3.5u$; $x_3 = \text{good}_2 5r$;
 $\text{rt}_0 x_1 = \text{rt}_1 x_2$; $\text{rt}_0 x_3 = \text{rt}_1 x_2$; $x_1 = r - 2.75u$; $\text{rt}_0 x_3 = \text{round}(r - 5u)$;
 vpen; $\text{top}_0 y_1 = m$; $y_2 = .5$; $y_3 = y_2$; $y_3 = y_2$; $\text{bot}_0 y_1 = -oo$; $y_3 = \text{good}_6 2[r, m]$;
 $y_1 = y_2$;
 draw $\{w_1\} \{1\} \{x_2 - x_1, y_2 - y_1\}$ $\{w_2\} \{2\} \{0, -1\}$ $\{w_3\} \{3\} \{1, 0\}$ $\{w_4\} \{4\} \{0, 1\} \dots$
 $\{5\} \{-1, 0\} \dots \{w_5\} \{6\} \{0, -1\}$ $\{w_6\} \{7\} \{1, 0\} \dots \{w_7\} \{8\} \{0, 1\}$;
 call ‘a endv(8).

% strokes
 % closing bulb





```

"Variant lower case Greek phi";
call charbegin(175, 12, 3mc-px-slant, -7mc-px-slant, ph, pd, 0);
hpen; x1 = good0 1.5u; lft0x2 = round .5u; x3 = .5(r + u);
rt0x1 = round(r - .5u); x7 = 8u; x0 = x1 = good0 1.5u;
vpen; top0y1 = m; y2 = y1 = e; bot0y1 = -oo; top0y2 = m + oo; y0 = .5m;
bot0y2 = -d - oo;
draw lft0x1{2(x2 - x1), y2 - y1}..lft0x2{0, -1}..lft0x3{3(1, 0)}..lft0x4{0, 1}.
lft0x5{5(-1, 0)}..lft0x6{0, -1};
hpen; draw lft0x6{6}..lft0x7;
open; w1 draw 7.
% stroke
% stem
% 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);
hpen; x2 = x1; y2 = .7(y1, e); x3 = x2 + .4u; y3 = .25(y1, e);
x4 = .5(x1, x3); bot0y1 = -oo; x5 = good2(r - 1.5u); top0y2 = h + oo;
x0 = x1; y0 = y1; lft0x2 = round 3u; y7 = .5(m, h);
x8 = r; y8 = y1 = good0 5(e, m); x9 = r + 2u;
draw lft0x1{1}..lft0x2{2(0, -1)}..lft0x3{3}..lft0x4{4(1, 0)};
call "b darc(4, 5, w2);
w1 draw 6{-1, 0}..7{0, -1}..8{.9}.

"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 pstroke;
hpen; x1 = good0 3u; lft0x2 = round 2u; x3 = 4u; x4 = good 7.5u;
x5 = r - 4.5u; rft0x6 = round(r - 1.5u); x7 = r - 2.5u;
vpen; top0y1 = m; y2 = y8 = e; bot0y2 = -oo; y7 = y1; y0 = y1;
draw lft0x1{2(x2 - x1), y2 - y1}..lft0x2{2(0, -1)}..lft0x3{3(1, 0)}..
lft0x4{5(0, 1)};
draw lft0x5{5(0, -1)}..lft0x6{7(1, 0)}..lft0x7{8(0, 1)}..
9{2(x1 - x8), y1 - y8}.
% bar
% left bowl
% right bowl

```


The file ita1ms.mf

% This file contains miscellaneous symbols of main italic fonts
 % Math spacing is assumed.

% The character codes are '040, '042-'045, '055, and '100.

new mc, lbowi, lhook, rbowi, rhook, rstem; % quantities used in spacing corrections

mc = 1/pu;

lhook = 1/px slant + 5pw + .5pu;

lbowi = .3px slant - .5pwii + pu;

rbowi = .7px slant + .5pwii - pu;

rhook = 1/px slant - .5pw - .5pu;

rstem = px slant + .5pwi - pu;

"Dolless italic letter l";

call charbegin('040, 7, 1 + mc-lhook, 1 - mc-rhook, px, 0, 0);

x1 = .5r + .2u; x2 = .5r - .25u;

call 'a skewentry(u, 1);

call 'b skewexit(2, r - u);

hpen; w1 draw 1..2.

% opening hook
 % closing hook
 % stem

"Straight double quotes";

call charbegin('042, 9, 0, 0, ph, 0, ph slant + .5pwii - 2.5pu);

new w0;

if w3 < w0 sqrt 2: w0 = round w0 sqrt 2;

else: w0 = w3;

fi;

x1 = x2 = good0, 3u; x3 = x1 = r - x1;

open; top0y1 = h; y1 = .5[e, m], y1 = y1; y1 = y1;

call 'a cdraw(1, 2, 99, 0);

call 'b cdraw(3, 4, 99, 0).

% left stem
 % right stem

"Lower case italic script l";

call charbegin('043, 6, 0, -mc(.8ph slant - pu), ph, 0, 0);

hpen; x1 = good0, 0; r0x2 = round(r - 1.5u); x1 = 5r; if x1 = round .5u;

x3 = r - 2.25u; x0 = good0(r + 2); % {two in a row will connect}

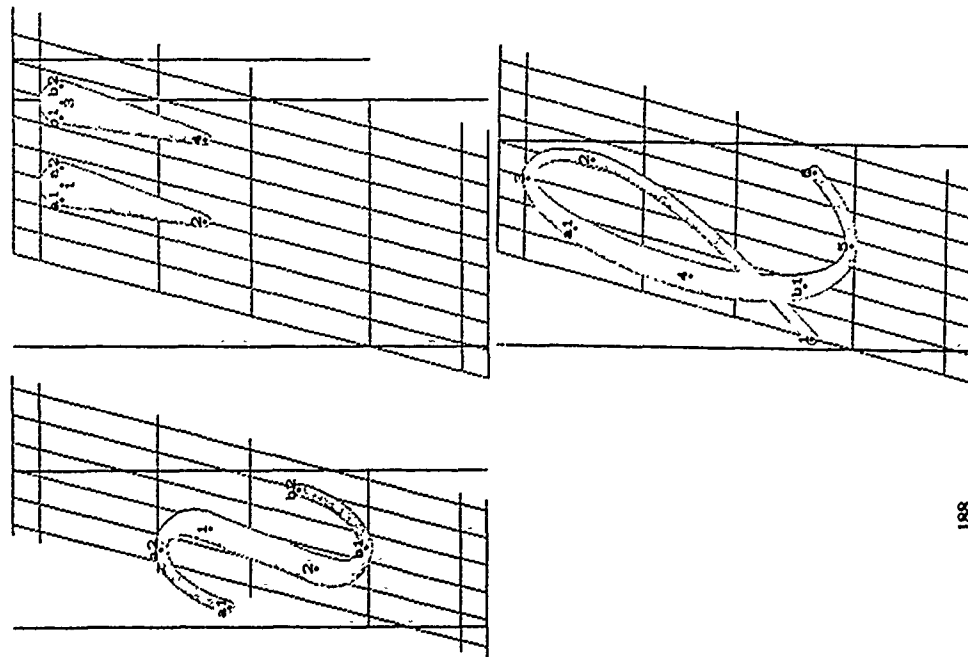
y1 = y0 = .125h; y2 = 8h; top0y1 = h + oo; y1 = .5[y0, y1]; bot0y1 = -oo;

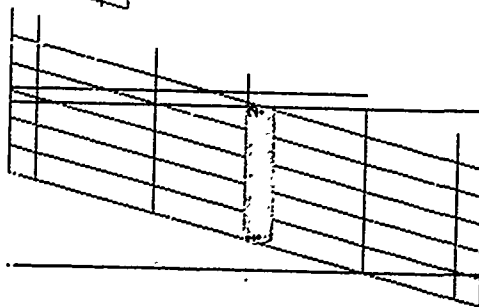
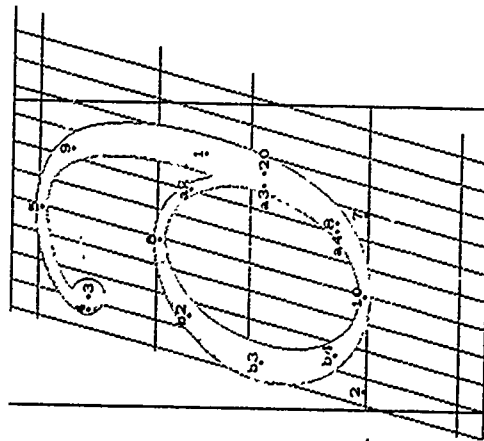
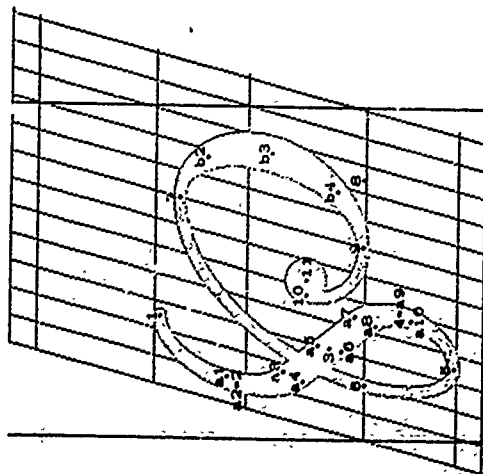
w1 draw 1{10u, h}. 2{0, 1} 3{-1, 0);

call 'a arc(3, 4, w1); call 'b arc(5, 4, w1);

draw 5{1, 0}..6{10u, h}.

% right of bowl
 % left of bowl
 % link





```

"Wienerstrass p";
call charbegin("044, 11, 0, -mc-rbowl, px, pd, 0);
lpen; x1 = 2.5u; top0y1 = m; lft0x2 = round .5u;
x2 = 3u; y1 = .5e, -.5d; rft0x1 = round 5.25u;
x3 = 3.5u; bot0y2 = -d - .oo;
call "a sdraw(1, 2, 3, 4, 5, w10, w8, -m/(8u));
x0 = good0.2u; w0 = 0; x7 = 7u; top0y7 = round 8[e, m];
w0 draw 5{-1, 0}..6(0, 1)..7{1, 0};
x8 = good2(r - 1.5u); bot0y8 = -oo; x1 = x7; y0 = y8;
call "b darc(7, 8, w2);
lft1x1 = lft0x10 = round 4.5u; y10 = y11 = .5e;
draw 9{-1, 0}..10{0, 1};
open, w1 draw 11.

% flourish
% stem and shoulder
% bowl
% link
% bulb

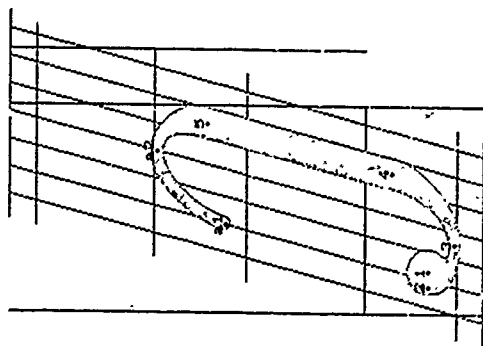
"Partial differential sign";
call charbegin("045, 10, mc-bowl, -mc(.7 ph-slant + .5 pwii - pu), ph, 0, 0);
% (This is a mirror-reflected number 6, one unit wider)
x1 = good1(r - 1.5u); x2 = good1.5u;
new w0; w0 = round .75[w0, w1];
open; top1y1 = h - .25e;
if y1 < 5[m, h]; new y1; y1 = .5[m, h];
fi;
lft0x1 = lft0x1 = round 1.5u; y1 = y1;
w0 draw 3;
lpen; x20 = good2(x1 - 1u); x3 = x0 = x10 = .5[x20, x2]; top0y7 = h + oo;
w0 draw 4{0, 1}..5{1, 0};
bot0y2 = -oo; top0y8 = m + oo; y20 = .5[y2, y6];
y1 = y2 = y10; lft0x7 = lft0x0;
call "a darc(6, 7, w0); call "b darc(6, 2, w2);
new w0; w0 = 1/2[w0, w2];
x8 = x0; lft0x8 = lft0(1/sqrt(2)[x0, x1]);
y8 = 1/sqrt(2)[y0, y1]; y5 = y8 = y8 - y7; y1 = 5[y7, y10];
draw [w0#5{1, 0}..[w0]9{x7 - x0, y7 - y10}..[w0]11{0, -1}
..[w0]8{x0 - x7, y7 - y10}..[w0]10{-1, 0}.

% intermediate width used in darc routine
% stroke

"Hyphen";
call charbegin("055, 6, 0, 0, px, 0, 5px-slant - .5pu);
open; y1 = y2 = .5m;
if fixwidth = 0; lft1x1 = 0; rft1x2 = r - u;
else; lft1x1 = 1.5u; x0 = r - x1;
fi;
w7 draw 1..2.

% bar

```

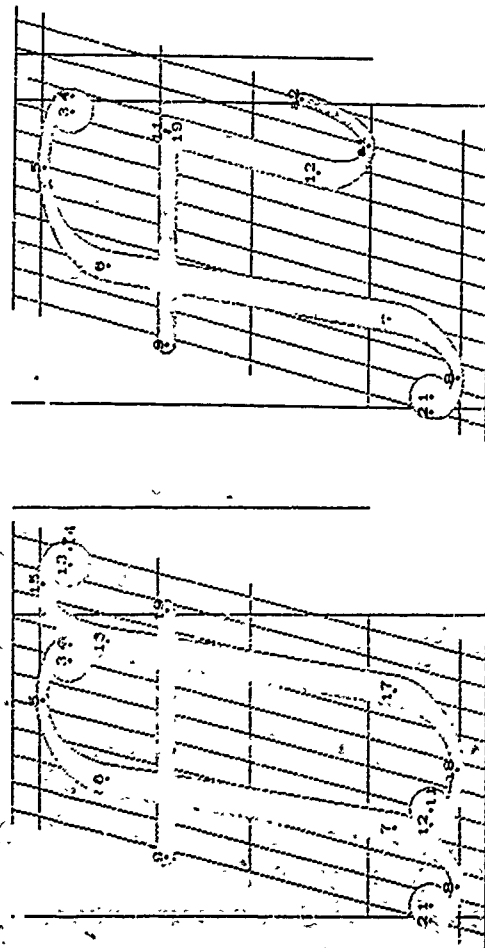


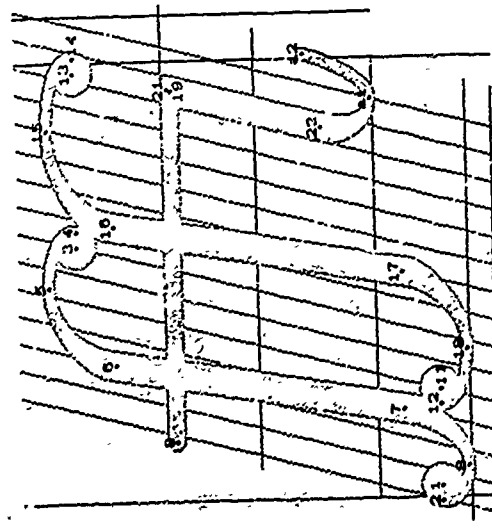
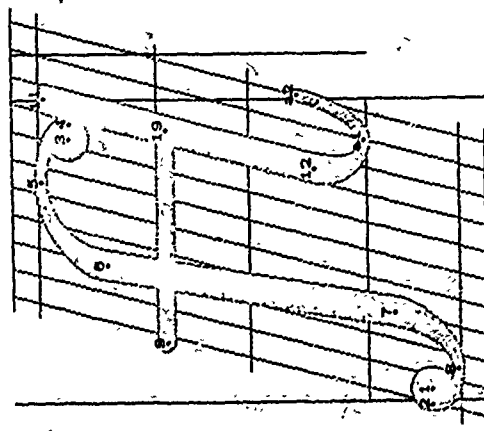
```
"Dotless italic letter j";
call charbegin(100, 7.5, 1 -- mc(pu + pd-slant, 1 -- mc(pr-slant + 5pwi -- 1.5pu),
    ph, pd, (ph -- pc)-slant);
open; if(x1 == lf0x2 == round .5u; x1 == 2.5u; x2 == x3 == good, 4.5u;
    bot,y1 == -.9d; y2 == y1; bot,y3 == -d - oo; bot,y4 == -.1d;
    w1 draw 1;
    call ~a entry(u, 5);
    hpen; draw [w1]5...[w1]4(0, -1) . [w1]3(-1, 0) . 2(0, 1);
    % bulb,
    % opening hook
    % stem and tail
```

The file ita!ig.mf

```
"Italic ligature ff";
call charbegin(173, 10, 0, 0, ph, pd, ph-slant + .75pu);
open; if(x1 == lf0x2 == round(-.25u); rt,x3 == rt,x1 == round(.5r + 1.75u);
    x2 == .5[x0, x1]; x0 == good, (.25r -- .5u); x7 == good, (.25r + .5u); x8 == .5[x2, x1];
    bot,y1 == -.3d; y2 == y1; y3 == y1; y4 == y2 == y3 -- y5;
    top,y6 == h - 1 - oo; y5 == y4 == y3; bot,y7 == -.1d; bot,y8 == -d - oo;
    lf,x11 == lf,x12 == round(.5r -- 1.75u); rt,x13 == rt,x11 == round(.r + 2.5u);
    x15 == .5[x0, x1]; x16 == good, (.75r -- .5u); x17 == good, (.75r + .5u); x18 == 5[x12, x1];
    y11 == y1; y12 == y2; y13 == y1; y14 == y1; y15 == y5; y16 == y6; y17 == y7; y18 == y8;
    x9 == x0 -- 2.25u -- eps; x10 == x16 + 1.75u + eps; top,y9 == m; y9 == y10;
    w1 draw 1; draw 3; draw 11; draw 13;
    w10 draw 9..19;
    hpen; draw [w1]4(0, 1) . [w1]5(-1, 0) . [w1]6(x7 -- x0, y7 -- y6) .
    [w1]7(x7 -- x0, y7 -- y6) . [w1]8(-1, 0) . 2(0, 1);
    draw [w1]4(0, 1) . [w1]5(-1, 0) . [w1]6(x7 -- x0, y7 -- y6) .
    [w1]7(x7 -- x0, y7 -- y6) . [w1]8(-1, 0) . 12(0, 1);
    lig ~f: ~1 = '174, ~r = '173, ~1 = '175;
```

```
"Italic ligature ff";
call max(rhook, ph-slant + .5pwi -- 2pu);
call charbegin(174, 10, 0, 0, ph, pd, acc);
open; if(x1 == lf0x2 == round(-.25u); rt,x3 == rt,x1 == round(.5r + 1.75u);
    x2 == .5[x0, x1]; x0 == good, (.25r -- .5u); x7 == good, (.25r + .5u); x8 == 5[x2, x1];
    bot,y1 == -.3d; y2 == y1; y3 == y1; y4 == y2 == y3 -- y5;
    top,y6 == h - 1 - oo; y5 == y4 == y3; bot,y7 == -.1d; bot,y8 == -d - oo;
    x11 == x12 == good, .75r; top,y9 == m;
    x9 == x0 -- 2.25u -- eps; x10 == x16 == m; x19 == x11; y19 == y9;
    w1 draw 1; draw 3;
    w10 draw 9..19;
    hpen; draw [w1]4(0, 1) . [w1]5(-1, 0) . [w1]6(x7 -- x0, y7 -- y6) .
    [w1]7(x7 -- x0, y7 -- y6) . [w1]8(-1, 0) . 2(0, 1);
    call ~a exit(12, r);
    w1 draw 11..12.
```





```

"italic ligature ff";
call max{rhook, ph-slant + .5pwi - 2pu};
call charbegin('175, 10, 0, 0, ph, pd, acc);
open; lift{x1 = lift{x2 = round(-.25u);
x2 = .5{x0, x1}; x0 = good1(.25r + .5u), x1 = .5{x2, x1};
bot3y1 = -1.3d; y2 = y1; y3 = y1; y4 = y1 - y8;
top3y1 = h + oo; y5 = y1 - y6 = y1 - y8;
x11 = x12 = good1(.75r); top3y1 = h;
x0 = x0 - 2.25u - eps; top3y1 = m; y10 = y1;
w3 draw 1; draw 3;
w10 draw 9..19;
hpen; draw {w1#4{0, 1}..w1#15{-1, 0}..w1#16{x1 - x0, y1 - y6}..
w1#17{x1 - x0, y1 - y6}..w1#18{-1, 0}..2{0, 1},
call 'a exit(12, r);
w1 draw 11..12.

% bulbs
% bar

% left stem
% closing hook
% right stem

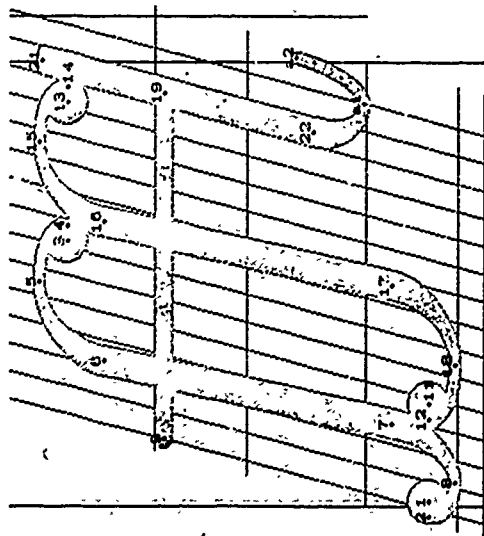
"italic ligature ff";
call max{rhook, ph-slant + .5pwi - 2pu};
call charbegin('176, 15, 0, 0, ph, pd, acc);
open; lift{x1 = lift{x2 = round(-.25u);
x2 = .5{x0, x1}; x0 = good1(.25r + .5u), x1 = .5{x2, x1};
bot3y1 = -1.3d; y2 = y1; y3 = y1; y4 = y1 - y8;
top3y1 = h + oo; y5 = y1 - y6 = y1 - y8;
x11 = x12 = good1(.75r); top3y1 = h;
x0 = x0 - 2.25u - eps; top3y1 = m; y10 = y1;
w3 draw 1; draw 3;
w10 draw 9..19;
hpen; draw {w1#4{0, 1}..w1#15{-1, 0}..w1#16{x1 - x0, y1 - y6}..
w1#17{x1 - x0, y1 - y6}..w1#18{-1, 0}..2{0, 1},
draw {w1#14{0, 1}..w1#15{-1, 0}..w1#16{x1 - x0, y1 - y6}..
w1#17{x1 - x0, y1 - y6}..w1#18{-1, 0}..12{0, 1},
call 'a exit(22, r);
w1 draw 21..22.

% bulbs
% bar

% left stem
% right stem
% closing hook
% right stem

fig '173: '1 = '176, '2 = '177;

```



```
"italic ligature ff",
call max(rhook, pl, slant + .5pt - 2pt);
call charbegin('177, 15, 0, 0, pl, pd, acc);
open; if(x1 == lf0x2 == round(-.5u), rt0x1 == round(.5r + 2u),
x1 == .5[x0, x1]; x0 == good, 2.25u; x1 == good, 2.75u; x2 == .5[x2, x1];
bot0y1 == h + .oo; y1 == y0; y2 == y1; y3 == y2 - .5d; bot0y2 == d - .oo;
top0y1 == h + .oo; y1 == y0; y2 == y1; y3 == y2 - .5d; bot0y3 == d - .oo;
lf0x11 == lf0x12 == round(.5r - 2u); rt0x11 == rt0x12 == round(.5r + 2u);
x12 == .5[x10, x11]; x10 == good, 7.25u; x17 == good, 7.75u; x18 == .5[x1, x17];
y11 == y1; y12 == y2; y13 == y3; y14 == y1; y15 == y1; y16 == y1; y17 == y1; y18 == y1;
x21 == x22 == good, (r - 2.5u); top0y1 == h;
x0 == x0 - 2.25u - eps; x10 == x21; top0y1 == m; y0 == y0;
w0 draw 1; draw 3; draw 11; draw 13;
w0 draw 9..19;
hpen; draw [w0]4{0, 1}..[w0]5{-1, 0}; [w0]6{x1 - x0, y1 - y0}..
[w0]7{x1 - x0, y1 - y0}..[w0]8{-1, 0}..2{0, 1};
draw [w0]14{0, 1}..[w0]15{-1, 0}; [w0]16{x17 - x10, y17 - y10}..
[w0]17{x17 - x10, y17 - y10}; [w0]18{-1, 0}..12{0, 1};
call "a exit(22, r);
w1 draw 21..22.
```

% bulbs
% bar
% left stem
% right stem
% closing hook
% right stem

The file itals.mf

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

"Dotless italic letter i";

call charbegin('013, 7, 1 - fixwidth, 1 - fixwidth, px, 0, 0),

x1 == .5r + .25u; x2 == .5r - .25u;

call "a skewentry(u, 1);

call "b skewexit(2, r - u);

hpen; w1 draw 1..2.

% opening hook
% closing hook
% stem

"Dotless italic letter j";

call charbegin('014, 8, 1 - fixwidth, 1 - fixwidth, px, pd, 0);

open; lf0x1 == lf0x2 == round .5u; x1 == 2.5u; x2 == good, .5(r + u),

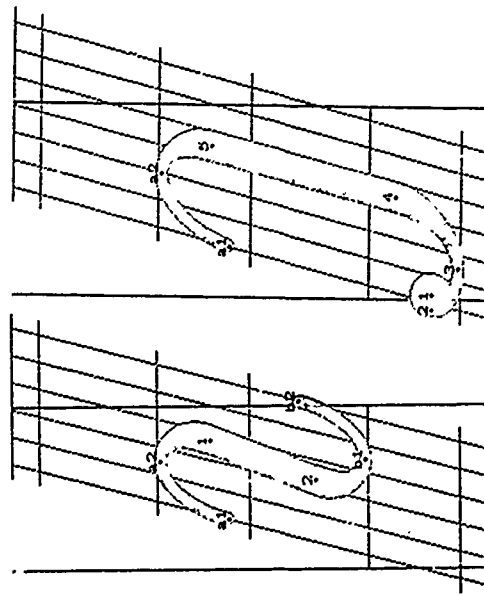
bot0y1 == - .5d; y1 == y0; y2 == y1; bot0y2 == d - .5d;

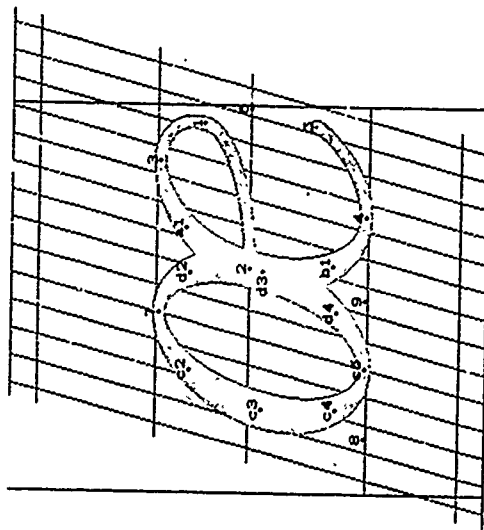
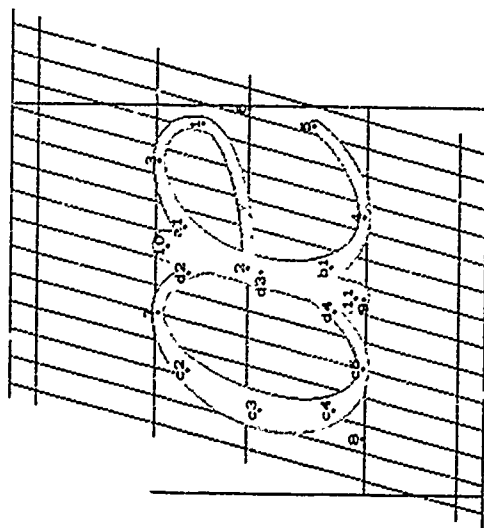
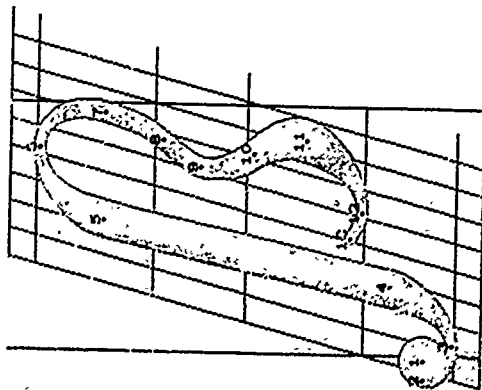
w1 draw 1;

call "a entry(u, 5);

hpen; draw [w0]5..[w0]4{0, -1}; [w0]3{-1, 0}..2{0, 1};

% bulb
% opening hook
% stem and tail





"Italic German letter ss";

call charbegin('033, 9, 0, 0, ph, pd, 0);

open; lt₀x₁ := lt₀x₂ := round(-.5u); x₁ := .5[x₂, x₁];

x₁ := good; 2.75u; x₁ := good; 2.25u;

x₀ := .5[x₃, x₇]; r₁w₅ := round(r - 2.5u); x₃ := .6[x₉, x₇]; r₁x₀ := round(r - 3.5u);

x₀ := .5[x₂, x₁]; r₁x₁ := round(r - u); x₁₂ := x₁₀ - u; lt₀x₁₃ := r₁x₁ + 5u,

bot₁y₁ := -.9d; y₁ := y₂; bot₀y₁ := -d - oo; bot₀y₁ := -.3d; y₄ - y₇ := y₁ - y₁;

top₀w₆ := h + oo; y₁ := y₁; y₄ := .6[y₇, y₁]; y₁ := .5[e, m]; y₁₀ := 5[y₁, y₁₁].

y₁₁ := .5e; bot₀y₁₂ := -oo; y₁₃ := .125e;

w₁ draw 1;

lpen; draw l₀{2(0, -1) .. l₀w₁{3{1, 0) .. l₀x₁{4{x₂ - x₁, y₅ - y₁}.
l₀w₁{5{x₅ - x₁, y₅ - y₁} .. l₀w₁{6{1, 0) .. l₀w₁{7{0, -1} .. 8{1.5{x₀ - x₃, y₉ - .5} ..
l₀w₁{9{0, -1} .. 1.5{l₀w₁₀, w₁}10{2{x₁₀ - x₀, y₁₀ - y₁} .. l₀w₁{11{0, -1} ..
l₀w₁{12{-1, 0) .. 13{ .. 9).

% bulh

% stroke

"Italic ligature ae";

call charbegin('034, 13, 0, 0, px, 0, 0);

lpen; r₁x₂ := round(r - 1.5u); x₂ := good; .5r;

x₁ := .5(r + 6u); r₁x₀ := r - .5u; x₀ := x₁;

y₁ := .5[e, m]; y₂ := e; top₀y₁ := m + oo; bot₀y₁ := -oo, top₀y₂ := .5e; y₀ := e;

w₀ draw 2{1, 0) .. 1{0, 1} 3{-1, 0} .. 3{-1, 0);

call 'a arc(3, 2, w₁); call 'b arc(4, 2, w₁);

draw 4{1, 0) .. 5{ .. 6);

x₁ := .5[x₉, x₁]; x₈ := good; 1.5u; x₁ := y₁; y₄ := y₁ := y₁;

call 'c dare(7, 8, w₂); call 'd dare(7, 9, w₁);

open; x₁₀ := x₁₁ := x₂; top₁y₁₀ := m + oo; bot₁y₁₁ := -oo;

w₁ draw 10..11.

% left bowl

% point

% stem

"Italic ligature oe";

call charbegin('035, 13, 0, 0, px, 0, 0);

lpen; r₁x₂ := round(r - 1.5u); x₂ := good; .5r;

x₁ := .5(r + 6u); r₁x₀ := r - .5u; x₀ := x₁;

y₁ := .5[e, m]; y₂ := e; top₀y₁ := m + oo, bot₀y₁ := -oo, top₀y₂ := .5e; y₀ := e;

w₀ draw 2{1, 0) .. 1{0, 1} 3{-1, 0} .. 3{-1, 0);

call 'a arc(3, 2, w₂); call 'b arc(4, 2, w₂);

draw 4{1, 0) .. 5{ .. 6);

x₁ := .5[x₉, x₁]; x₈ := good; 1.5u; x₁ := x₂, y₁ := y₁ := y₁;

call 'c dare(7, 8, w₂); call 'd dare(7, 9, w₂).

% bar and shoulder

% right bowl

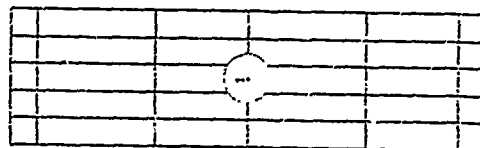
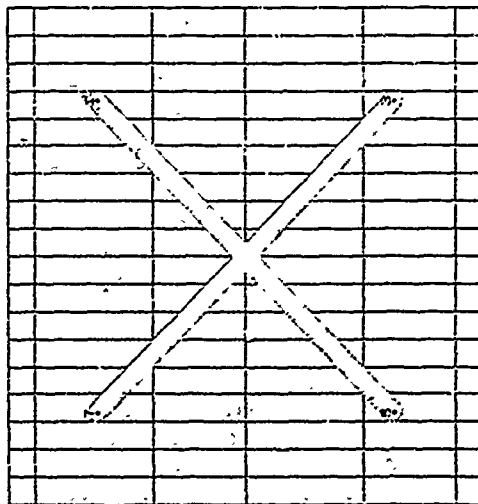
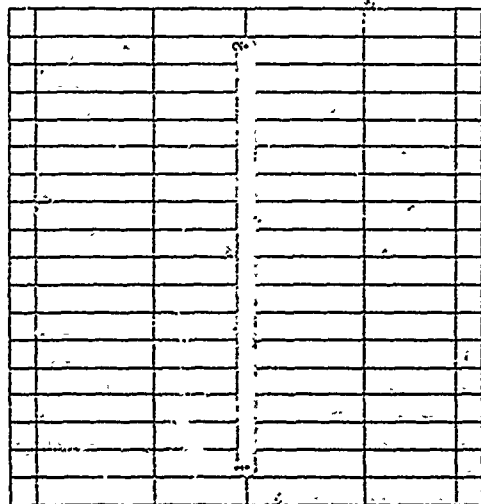
% point

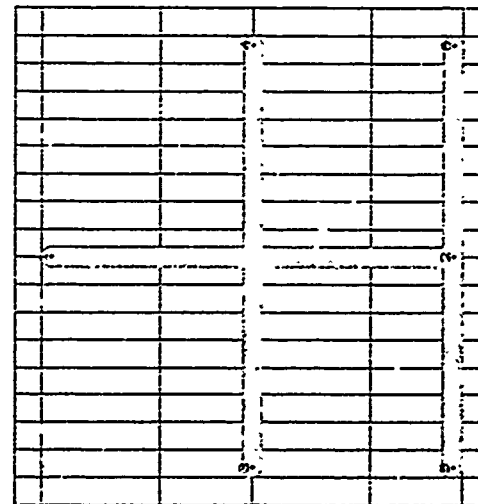
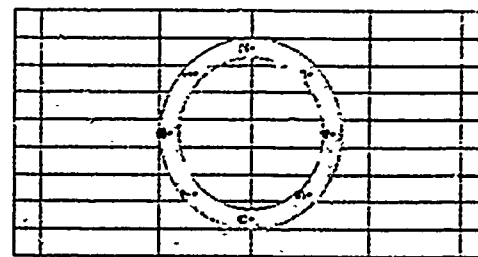
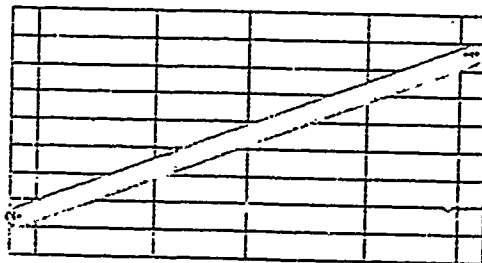
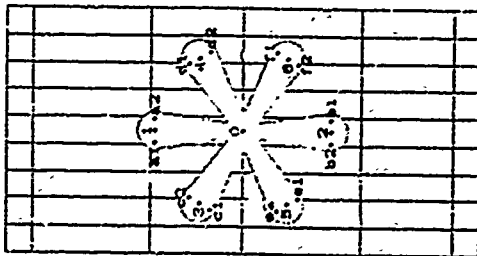
% left bowl

SYMBOL CHARACTER DESIGNS

The file symbol.mf

```
% The Computer Modern Symbols family of fonts (by D. E. Knuth, 1979).
danger = 0; % upper case script alphabet
mi = 1; input script;
texinfo slant, 6pt, 3pt, 2pt, px, 18pt;
% (The calling file should give the rest of the texinfo.)
new slant; slant = 0; trxy 0; % the non-script characters are unslanted
"Minus sign";
open;
if fiwidth = 0: if pa + 8pt > ph;
    call charbegin('000, 18, 0, 0, 0, ph, ph - 2pa, 0);
else: call charbegin('000, 18, 0, 0, 8pt - 1 pa, 8pt - pa, 0);
fi;
else: call charbegin('000, 9, 0, 0, 3.5pt + pa, 3.5pt - pa, 0);
fi;
if u1 = round u; x2 = r - x1; y1 = y2 = a;
w10 draw 1..2; % bar
% dot
"Times operator";
call charbegin('002, 18, 0, 0, 1/sqrt(2)(pa, ph), 1/sqrt(2)(pa, ph - 2pa), 0);
open; x7 = 1/sqrt(2)(5r, u); y7 = 1/sqrt(2)(a, h);
x5 = x7; x4 = x1 = r - x7; y1 = y7; y1 = y7; .5[y1, y1] = a;
w10 draw 7..3; % upper left to lower right diagonal
draw 5..1; % lower left to upper right diagonal
```





```

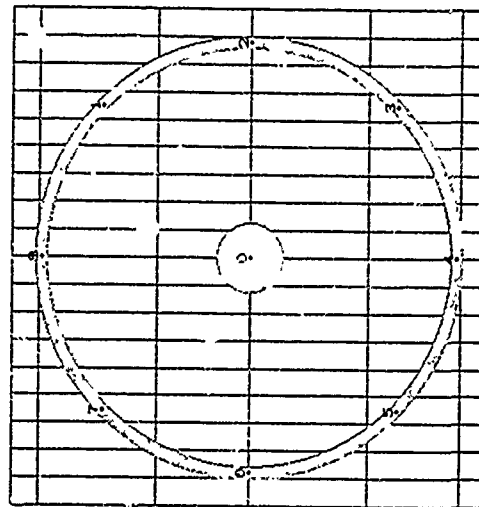
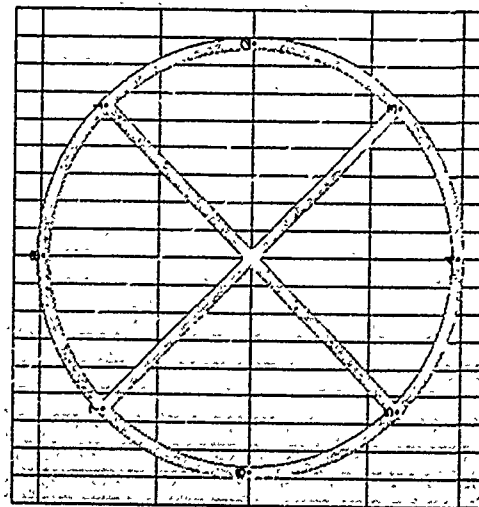
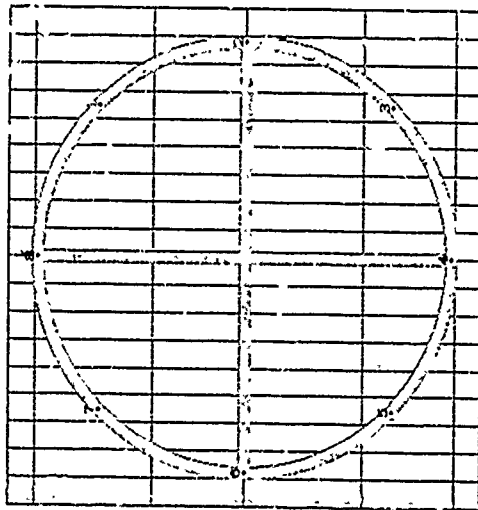
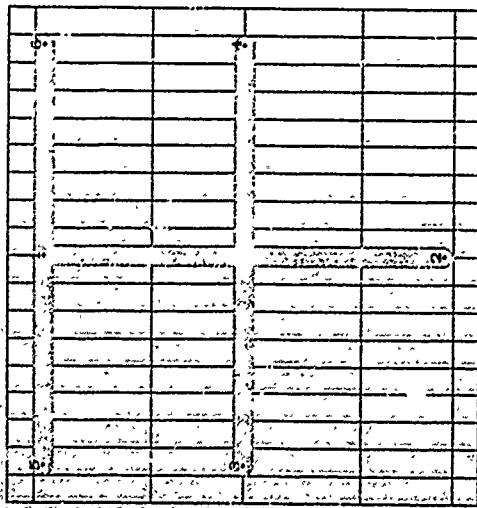
Asterisk at the axis";
call charbegin('003,9,0,0,5px+pa,5px--pa,0);
open; top,y1=round(a+5m); top,y2=m;
y0=.5[y1,y2]; x0=r--x0; x1=r--x0; x2=r--x0; x3=r--x0; x4=r--x0;
% left-right symmetry
y1=y2; y3=y0; y4=y0; y5=y1-100; .5[y1,y2]=y0;
x1-x0=.5sqrt(3)[y1-3.75u;
call 'a edraw(1,0,1,0);
call 'b edraw(2,0,1,0);
call 'c edraw(3,0,1,0);
call 'd edraw(4,0,1,0);
call 'e edraw(5,0,1,0);
call 'f edraw(6,0,1,0);

"Reverse slash";
call charbegin('004,9,0,0,ph+pb,ph+pb-2[pa,0);
open; left,y1=round(u); x2=r--x1;
top,y2=h+b; .5[y1,y2]=a;
w0 draw 1..2;

"Circle operator";
call charbegin('005,9,0,0,3.5pu+pa,3.5pu--pa,0);
open; x4=5r; left,y1=round(u); top,y2=round(a+3.5u); y3=a;
call circle(1,2,3,4,5,6,7,8,w0);

"Plus or minus sign";
open;
if fixwidth=0: if pa+8pu>ph;
call charbegin('006,18,0,0,ph,ph-2[pa,0); top,y2=h;
else: call charbegin('006,18,0,0,8pu+pa,8pu--pa,0); top,y2=a+8u;
fi;
else: call charbegin('006,9,0,0,3.5pu+pa,3.5pu--pa,0); top,y2=a+3.5u;
fi;
.5[y1,y2]=a; x1=x2=5r;
left,y1=round(u); x1=r--x1; y1=y1=a;
w0 draw 1..2;
draw 3..4;
x3=x1; x4=x1; y3=y1; y4=y1; draw 5..6;
% stem
% plus bar
% minus bar

```

```

"Minus or plus sign";
open;
if fixwidth = 0: if pa + 8pu > ph;
    call charbegin('007, 18, 0, 0, ph, ph - 2pa, 0); topud; = h;
else: call charbegin('007, 18, 0, 0, 8pu + pa, 8pu - pa, 0); topud; = a + 8u;
fi;
else: call charbegin('007, 9, 0, 0, 3.5pu + pa, 3.5pu - pa, 0); topud; = a + 3.5u;
fi;
.5[u, y] = a; x1 = x2 = .5r;
if fixwidth = round u; x1 = r - x; u1 = y1 = a;
w0 draw 1..2;
draw 3..4;
x1 = x2; x0 = x1; y1 = y0 = y1; draw 5..6;
% stem
% plus bar
% minus bar

"Circle-plus operator";
call charbegin('010, 18, 0, 0, ph, ph - 2pa, pa-slant - 5pu);
open; if fixwidth = round u; u0 = a; x0 = r - x; topud; = h + oo;
call circle(1, 2, 3, 4, 5, 6, 7, 8, u0);
w0 draw 6..2;
draw 4..8;
% bowl
% bar
% stem

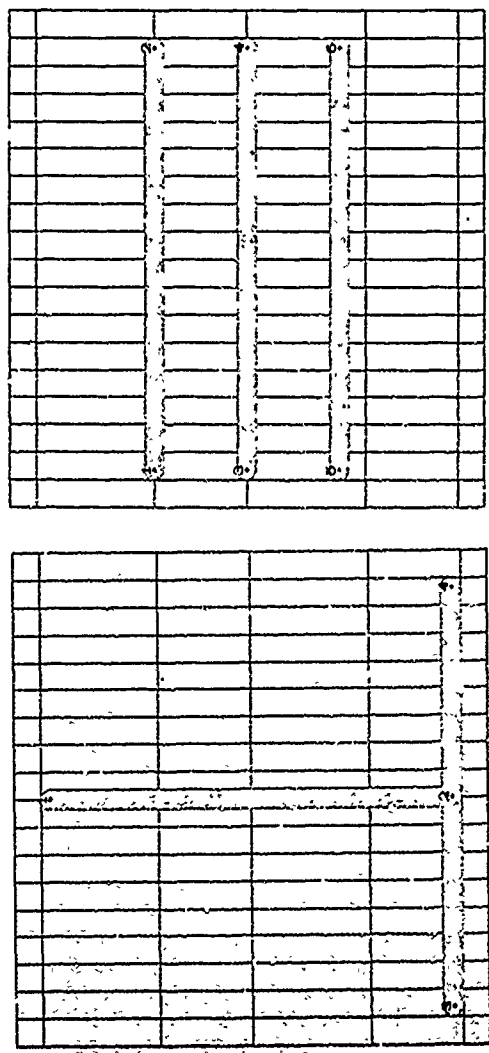
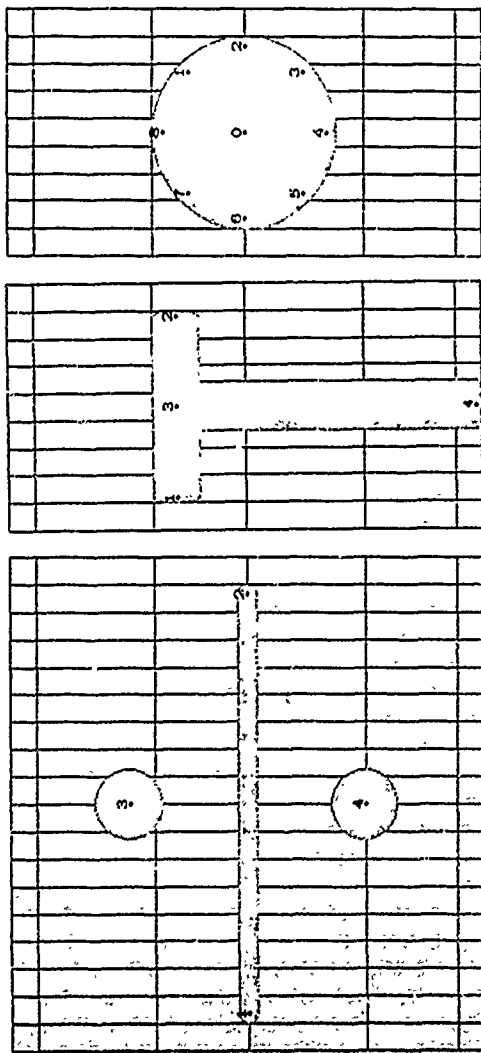
"Circle-minus operator";
call charbegin('011, 18, 0, 0, ph, ph - 2pa, pa-slant - 5pu);
open; if fixwidth = round u; u0 = a; x0 = r - x; topud; = h + oo;
call circle(1, 2, 3, 4, 5, 6, 7, 8, u0);
w0 draw 6..2;
draw 4..8;
% bowl
% bar
% stem

"Circle-times operator";
call charbegin('012, 18, 0, 0, ph, ph - 2pa, pa-slant - 5pu);
open; if fixwidth = round u; u0 = a; x0 = r - x; topud; = h + oo;
call circle(1, 2, 3, 4, 5, 6, 7, 8, u0);
w0 draw 7..3;
draw 5..1;
% upper left to lower right diagonal
% lower left to upper right diagonal

"Circle-divide operator";
call charbegin('013, 18, 0, 0, ph, ph - 2pa, pa-slant - 5pu);
open; if fixwidth = round u; u0 = a; x0 = r - x; topud; = h + oo;
call circle(1, 2, 3, 4, 5, 6, 7, 8, u0);
w0 draw 5..1;
% bowl
% lower left to upper right diagonal

"Circle dot operator";
call charbegin('014, 18, 0, 0, ph, ph - 2pa, pa-slant - 5pu);
open; if fixwidth = round u; u0 = a; x0 = r - x; topud; = h + oo;
call circle(1, 2, 3, 4, 5, 6, 7, 8, u0);
x0 = x; u0 = y;
open; new w0; w0 = w0 spt 2; w0 draw 0;
% bowl
% dot

```

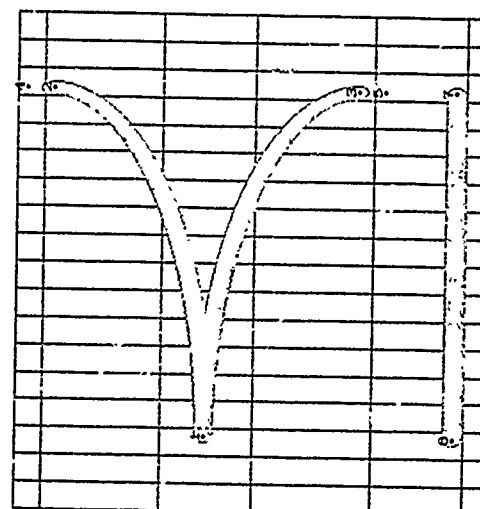
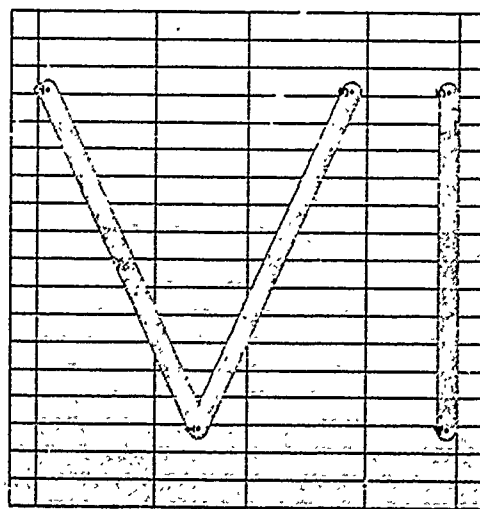
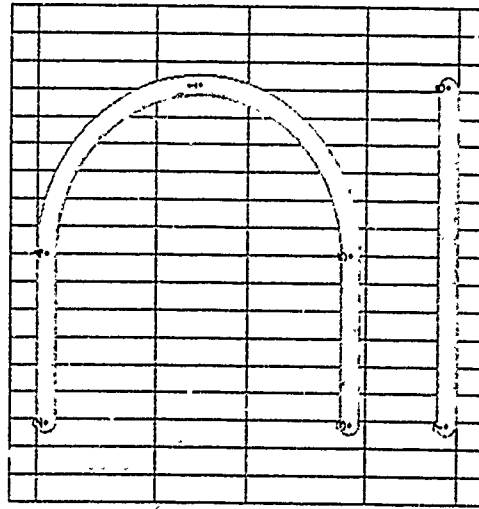
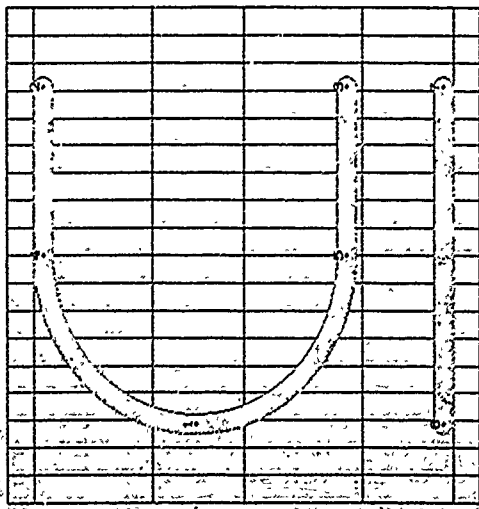
[illegible]

```

"Perpendicular sign or lattice bottom";
open;
if fixwidth = 0: if pa + 8pu > ph:
    call charbegin('020, 0, 0, ph, ph - 2pa, 0); top10h1 = h;
else: call charbegin('020, 18, 0, 0, 8pu + pa, 8pu - pa, 0); top10h1 = a + 8u;
fi;
else: call charbegin('020, 9, 0, 0, 3.5pu + pa, 3.5pu - pa, 0); top10h1 = a + 3.5u;
fi;
5[y1, y2] = a; x1 = x2 = .5r,
lf10x2 = round u; x1 = r - x2; y3 = y1 = u;
w10 draw 1..2;
draw 3..4.

"Equivalence or congruence sign";
call charbegin('021, 18, 0, 0, px = pr + prt/2 + pa, px - pr + prt/2 - pa, 0),
open; lf10x2 = round u; x1 = x2 = x; x2 = x1 = x0 = r - x;
y1 = y; y1 = y1 = a; y2 = y3; y1 = y2 = y3 = y4 = round(m c);
w10 draw 1..2;
draw 3..4;
draw 5..6.

```



"Reflexive subset sign";
call charbegin('022, 18, 0, 0, ph, ph - 2pa, 0);
open; if₁₀x₁ = round 2.5a; x₂ = x₁ = r - x₁;
top₁₀y₂ = h; .5[y₂, y₁] = y₁; y₂ - y₁ = (good₁₀ 5[m, h]) - (good₁₀a);
x₁ = x₂ = .5r; y₁ = y₂; y₂ = y₁;
w₁₀ draw 2...4{-1, 0} . 1{0, -1} . 5{1, 0} . 3;
x₀ = x₁; x₁ = x₂; y₀ = y₁; bot₁₀y₀ = 2a - h;
draw 6 . 7.

% stroke
% bar

"Reflexive superset sign";
call charbegin('023, 18, 0, 0, ph, ph - 2pa, 0);
open; if₁₀x₁ = round 2.5a; x₂ = x₁ = r - x₁;
top₁₀y₂ = h; .5[y₂, y₁] = y₁; y₂ - y₁ = (good₁₀ 5[m, h]) - (good₁₀a);
x₁ = x₂ = .5r; y₁ = y₂; y₂ = y₁;
w₁₀ draw 2...4{1, 0} . 1{0, -1} . 5{-1, 0} . 3;
x₀ = x₁; x₁ = x₂; y₀ = y₁; bot₁₀y₀ = 2a - h;
draw 6...7.

% stroke
% bar

"Less than or equal to sign";
call charbegin('024, 18, 0, 0, ph, ph - 2pa, 0);
open; if₁₀x₁ = round 2.5a; x₂ = x₁ = r - x₁;
top₁₀y₂ = h; .5[y₂, y₁] = y₁; y₂ - y₁ = (good₁₀ 5[m, h]) - (good₁₀a);
w₁₀ draw 2...1...1 . 3;
x₁ = x₁; x₂ = x₂; y₁ = y₂; bot₁₀y₁ = 2a - h;
draw 4...5.

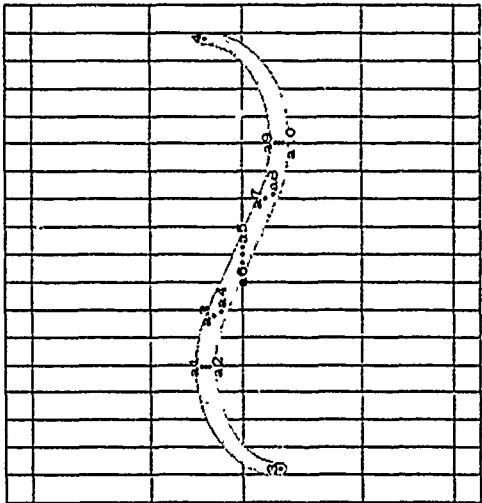
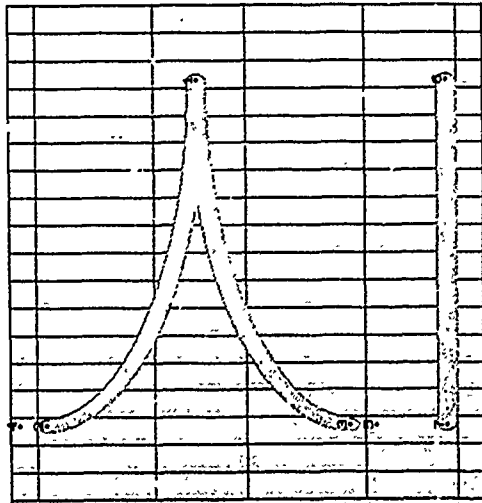
% diagonals
% bar

"Greater than or equal to sign";
call charbegin('025, 18, 0, 0, ph, ph - 2pa, 0);
open; if₁₀x₁ = round 2.5a; x₂ = x₁ = r - x₁;
top₁₀y₂ = h; .5[y₂, y₁] = y₁; y₂ - y₁ = (good₁₀ 5[m, h]) - (good₁₀a);
w₁₀ draw 2...1...1 . 3;
x₁ = x₁; x₂ = x₂; y₁ = y₂; bot₁₀y₁ = 2a - h;
draw 4...5.

% diagonals
% bar

"Precedes or equals sign";
call charbegin('026, 18, 0, 0, ph, ph - 2pa, 0);
open; if₁₀x₁ = round 2.5a; x₂ = x₁ = r - x₁;
top₁₀y₂ = h; .5[y₂, y₁] = y₁; y₂ - y₁ = (good₁₀ 5[m, h]) - (good₁₀a);
x₁ = x₂ = x₂; y₁ = y₂ + b; y₂ = y₁ - b;
w₁₀ draw (4) 2 1{-1, 0} 1{1, 0} 3(5);
x₀ = x₁; x₁ = x₂; y₀ = y₁; bot₁₀y₀ = 2a - h;
draw 6 . 7.

% diagonals
% bar



```

"Follows or equals sign";
call charbegin('027, 18, 0, 0, ph, ph - 2pa, 0);
vpen; lft10z2 = round 2.5u; x2 = x3 = r - x1;
top10z2 = h; .5[y2, y1] = y1; y2 - y1 = (good10.5[m, h]) - (good10.6);
x1 = x2 = x3; y1 = y2 + b; y5 = y3 - b;
w10 draw (4..2) .. 1{1, 0} .. 1{-1, 0} .. 3{...5};
x3 = x1; x7 = x2; y3 = y1; bot10z3 = 2a - h;
draw 6...7.

% diagonals
% bar

"Similarity sign";
call charbegin('030, 18, 0, 0, .5(px - pc) + pa, 0, 0);
vpen; top10y1 = round(a + .5(m - c) + cps); top10y1 - bot10y2 = round(m - c);
lft10z3 = round u; y5 = .5[y1, y2]; y3 = y2; y1 = y1; x1 = r - x1; x3 = r - x3;
call 'a zdraw(3, 1, 5, 2, 4, w10, w10 + deltaw, 7.5u/(c - m));

% stroke

"Approximate equality sign";
call charbegin('031, 18, 0, 0, 1.1(px - pc) + pa, 1.1(px - pc) - pa, 0);
vpen; top10y1 = round(a + 1.1(m - c) + cps); top10y1 - bot10y2 = round(m - c);
lft10z3 = round u; y5 = .5[y1, y2]; y3 = y2; y1 = y1; x1 = r - x1; x3 = r - x3;
x3 = x3; x9 = x1; x10 = x5;
y1 - y3 = y2 - y1 = y3 - y5 = y4 - y3 = y5 - y4 = round 1.2(m - c);
call 'a zdraw(3, 1, 5, 2, 4, w10, w10 + deltaw, 7.5u/(c - m));
call 'b zdraw(8, 6, 10, 7, 9, w10, w10 + deltaw, 7.5u/(c - m));

% upper stroke
% lower stroke

```

```

"Proper subset sign";
call charbegin('032, 18, 0, 0, .5(px, ph) + prt/2, .5(px, ph) + prt/2 - 2pa, 0);
cpen; lft10z1 = round 2.5u; x2 = x3 = r - x1;
y2 = good10.5[m, h]; .5[y2, y1] = y1 = good10.6;
x1 = x2 = .5r; y1 = y2; y5 = y3;
w10 draw 2..4{-1, 0} .. 1{0, -1} .. 5{1, 0} .. 3.

% stroke

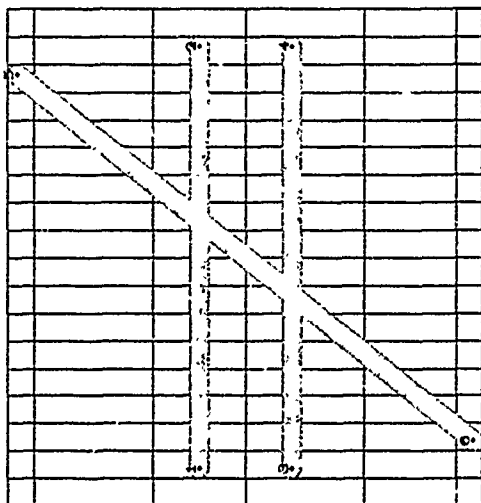
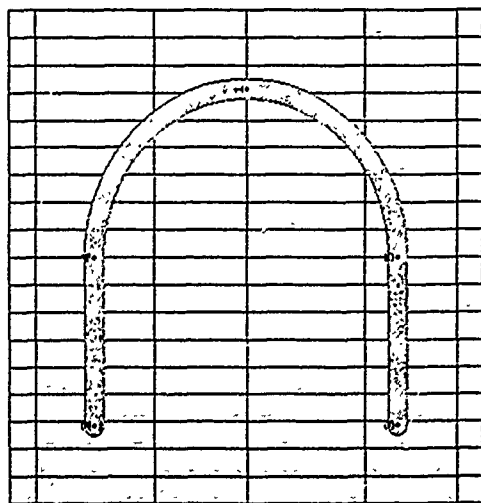
"Proper superset sign";
call charbegin('033, 18, 0, 0, .5(px, ph) + prt/2, .5(px, ph) + prt/2 - 2pa, 0);
cpen; lft10z2 = round 2.5u; x2 = x3 = r - x1;
y2 = good10.5[m, h]; .5[y2, y1] = y1 = good10.6;
x1 = x2 = .5r; y1 = y2; y5 = y3;
w10 draw 2..4{1, 0} .. 1{0, -1} .. 5{-1, 0} .. 3.

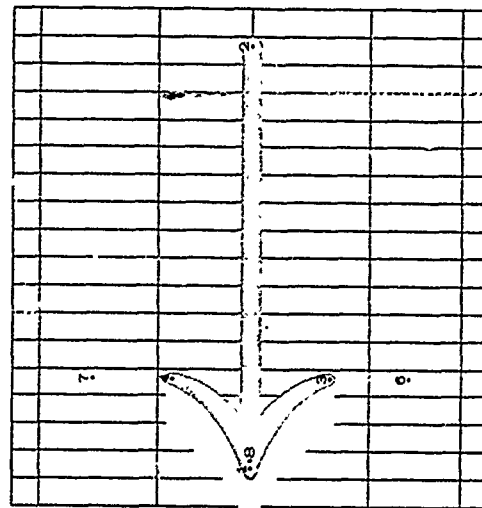
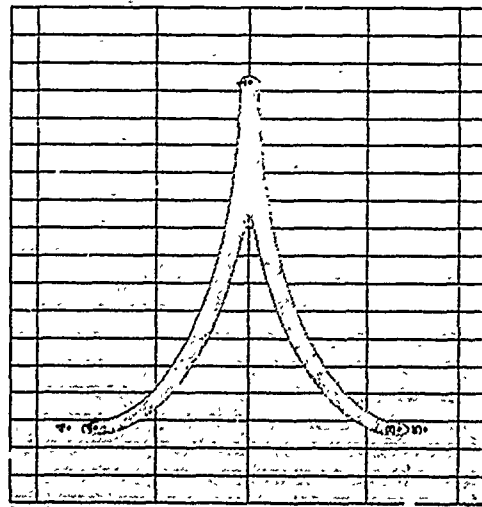
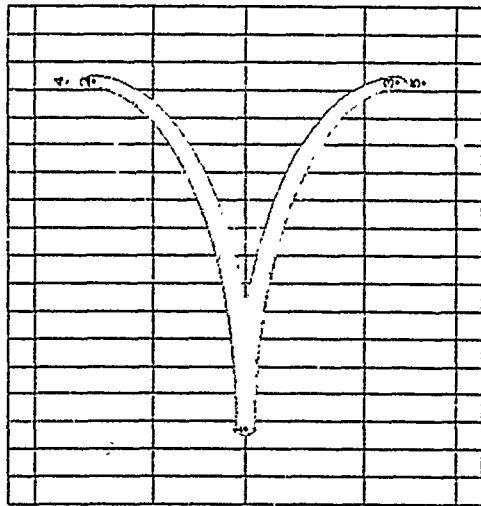
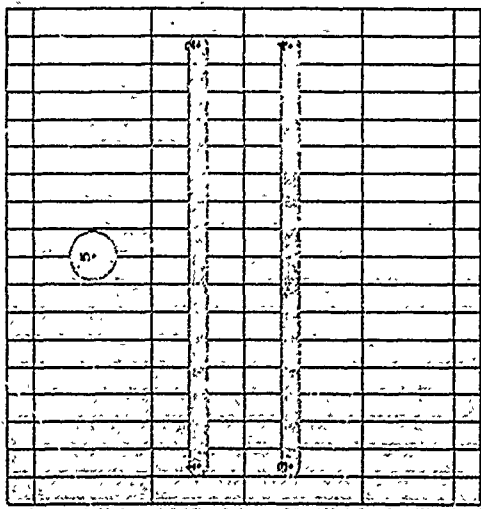
% stroke

"Unequal sign";
call charbegin('034, 18, 0, 0, ph + pb, ph + pb - 2pa, 0);
cpen; lft10z1 = round u; x3 = x1; x2 = x1 = r - x1;
y1 = y2; y3 = y1; y1 - y3 = round(m - c); .5[y1, y1] = a;
w10 draw 1..2;
draw 3..4;
rt10z3 = round(r - 2u); lft10z3 = round 2u;
top10z5 = h + b; bot10z5 = -d - b;
draw 5..8.

% upper bar
% lower bar
% diagonal

```





```

"Dot over equal sign";
call charbegin('035, 18, 0, 0, ph, 0, 0);
cpn; lft10x1 = round u; x2 = x1; x3 = x1;
y1 = y2; y3 = y1; y1 - y3 = round(m - e); .5[y1, y3] = a;
w10 draw l . 2;
draw 3..4;
new w10;
if w3 < w10sqrt 2; w10 = round w10sqrt 2;
else; w10 = w3;
fi;
x2 = .5r; y3 = 5[m, h]; w10 draw 5

% upper bar
% lower bar

% dot

"Precedes sign";
call charbegin('036, 18, 0, 0, 5[px, ph] + prt/2, 5[px, ph] + prt/2 - 2pa, 0);
vpen; lft10x1 = round 2.5u; x2 = x1; x3 = r - x1;
y2 = good10 5[m, h]; .5[y2, y1] = y1 = good10a;
x1 = x2; y1 = y2 + b; y3 = y1 - b;
w10 draw (4..2)..1{-1, 0}..1{1, 0}..3(.5).

% diagonals

"Follows sign";
call charbegin('037, 18, 0, 0, 5[px, ph] + prt/2, 5[px, ph] + prt/2 - 2pa, 0);
vpen; lft10x2 = round 2.5u; x2 = x1; x3 = r - x1;
y2 = good10 5[m, h]; .5[y2, y1] = y1 = good10a;
x1 = x2; y1 = y2 + b; y3 = y1 - b;
w10 draw (4..2)..1{1, 0}..1{-1, 0}..3(.5).

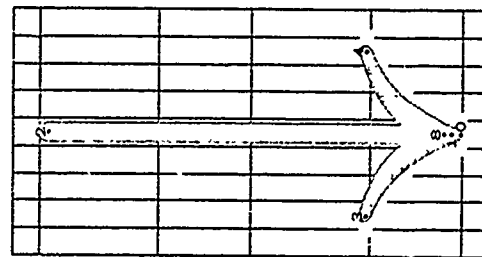
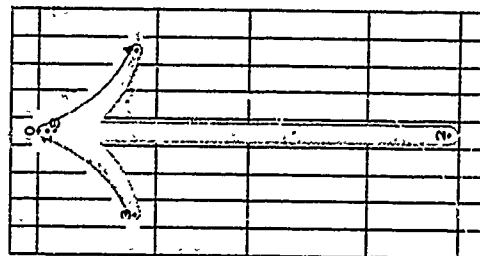
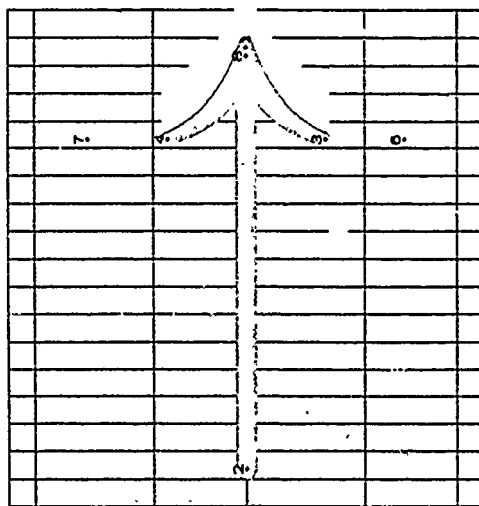
% diagonals

"Leftward arrow";
call charbegin('040, 18, 0, 0, 24ph + 7prt + pa, 24ph + 5prt - pa, 0);
cpn; lft10x1 = x0 = round u; r10x2 = round(r - u);
y1 = y2 = y3 = y8 = good10a;
w10 draw l..2;
lpen; lft10x3 = x0;
x3 - x8 = x8 - x3 = ..fixwidthh[3u, 6u] - eps; x1 = x1 = x6 = x7;
y3 - y8 = y1 - y1 = y1 - y1 = y1 - y1 = 24h + eps;
lpen#; w10 + w1 draw (5..8) 3(.6);
lpen; draw (w1[5..8]..w10[3] 6);
lpen#; w10 + w1 draw (5..8) 3(.6);
lpen; draw (w1[5..8]..w10[4] 7)

% bar

% erase excess at lower left
% lower point
% erase excess at upper left
% upper point

```



```
"Rightward arrow";
call charbegin(041,13,0,0,.24ph + pa,.24ph + 5prt - pa,0);
cpen; ft10x2 = round u; rt10x1 = x0 = round(r - u);
u1 = y2 = y3 = y8 = good10a;
w10 draw l..2;
lpen; rt10x3 = x0;
x3 - x8 = x8 - x1 = fixwidth[3u,6u] + eps; x7 = x4 = x0 = x7;
y1 - y8 = y1 - y1 = y1 - y1 = y1 = .24h + eps;
rpen#; w10 + w1 draw (5..)8..3(..6);
hpen; draw {w1|5..)8..|w1|3(..6);
rpen#; w10 + w1 draw (5..)8..4(..7);
lpen; draw {w1|5..)8..|w1|4(..7);
```

% bar

% erase excess at lower right
% lower point
% erase excess at upper right
% upper point

```
"Upward arrow";
call charbegin(042,9,0,0,ph,ph - 2pa,0);
cpen; top10y1 = y0 = h; .5[y1,y2] = a;
x0 = x1 = x2 = x3 = x8 = good10.5r;
w10 draw l..2;
vpen; top7y8 = y0;
lpen#; w10 draw 0..8; rpen#; w10 draw 0..8;
y1 - y8 = y8 - y1 = .24h + eps; y2 = y1 = y0 = y1;
x1 - x0 = x1 - x1 = x1 - x1 = x1 = x1 = 3u + eps;
lpen#; w10 draw (5..)8..3(..6);
vpen; draw {w1|5..)8..|w1|3(..6);
rpen#; w10 draw (5..)8..4(..7);
vpen; draw {w1|5..)8..|w1|4(..7);
```

% stem

% clean the top

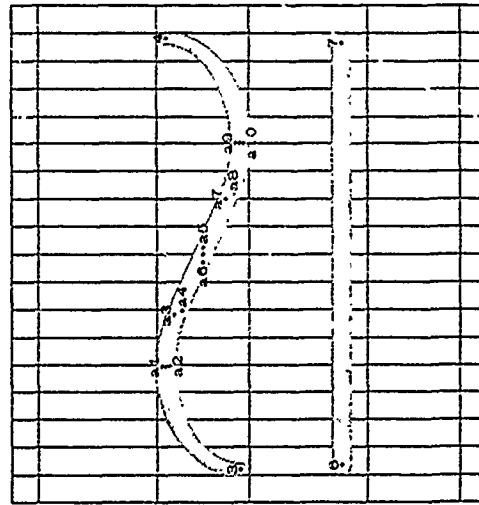
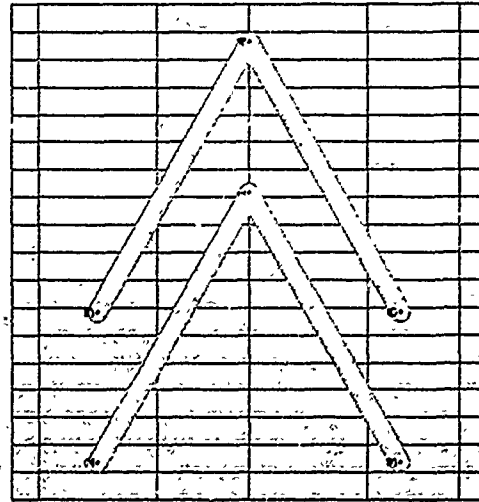
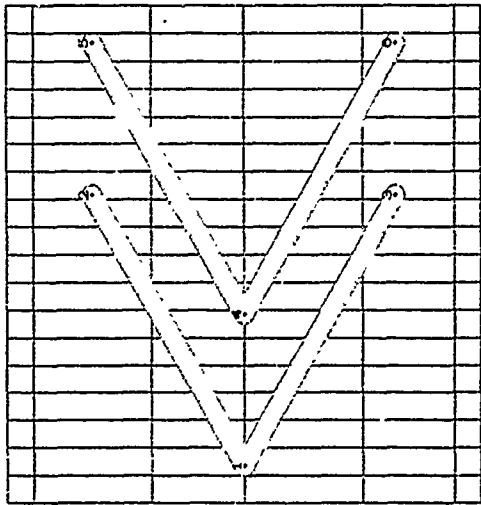
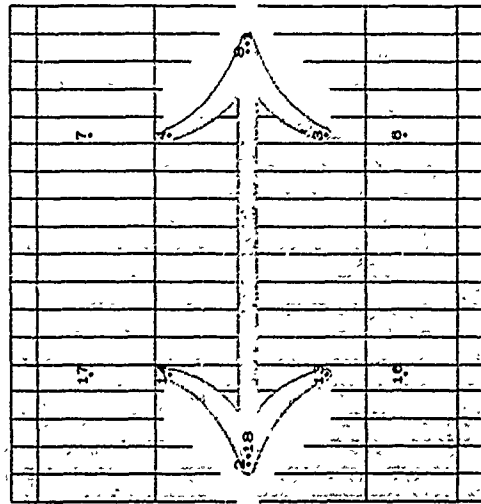
% erase excess at left
% left point
% erase excess at right
% right point

```
"Downward arrow";
call charbegin(043,9,0,0,ph,ph - 2pa,0);
cpen; top10y2 = h; .5[y1,y2] = a; y0 = bot10y1;
x0 = x1 = x2 = x3 = x8 = good10.5r;
w10 draw l..2;
vpen; bot7y8 = y0;
lpen#; w10 draw 0..8; rpen#; w10 draw 0..8;
y1 - y8 = y8 - y1 = .24h - eps; y2 = y1 = y0 = y1;
x1 - x0 = x1 - x1 = x1 - x1 = x1 = x1 = 3u + eps;
lpen#; w10 draw (5..)8..3(..6);
vpen; draw {w1|5..)8..|w1|3(..6);
rpen#; w10 draw (5..)8..4(..7);
vpen; draw {w1|5..)8..|w1|4(..7);
```

% stem

% clean the top

% erase excess at left
% left point
% erase excess at right
% right point



```

"Left-and-right arrow";
call charbegin("044, 18, 0, 0, 24pt + pa, 24pt + 5pt - pa, 0);
eps; ll10x1 = x10 = round u; r10x1 = x0 = round(r - u);
y1 = y2 = y5 = y8 = good10;
w10 draw l..2;
hpen; r1x8 = x0; ll1x18 = x10;
x3 - x8 = x3 - x5 = x13 - x18 - x15 = fixwidth[3u, 6u] + eps;
x2 = x1 = x0 = x7; x13 = x11 = x16 = x17;
y3 - y8 = y1 - y1 = y1 - y1 = y1 - y1 = 24t + eps,
y13 = y1; y1 - y1 = y6; y15 = y5; y16 = y6; y17 = y1, y18 = y8;
rpen; w10 + w1 draw (5..8..3(.6);
hpen; draw (w1[5..]8..l10[3(.6);
rpen; w10 + w1 draw (5..8..4(.7);
hpen; draw (w1[5..]8..l10[4(.7);
lpen; w10 + w1 draw (15..)18..13(.16);
hpen; draw (w1[15..]8..l10[13(.16);
lpen; w10 + w1 draw (15..)18..14(.17);
hpen; draw (w1[15..]18..l10[14(.17);

```

```

"Much less sign";
call charbegin(.045, 18, 0, 0, .5[px, ph] + prt/2, 5[px, ph] + prt/2 - 2, n, 0);
open; 16[02] == round u; 14[02] == round 11.5u; x1 == x2;
y2 == good10.5[m, h]; .5[y2, y1] == y2 == good10u;
w10 draw 2.1.1.1.3;
16[02] == round(r - u); x1 == x2; y1 == y2;
draw 5.4.4.4.6.
% left diagonals
% right diagonals

```

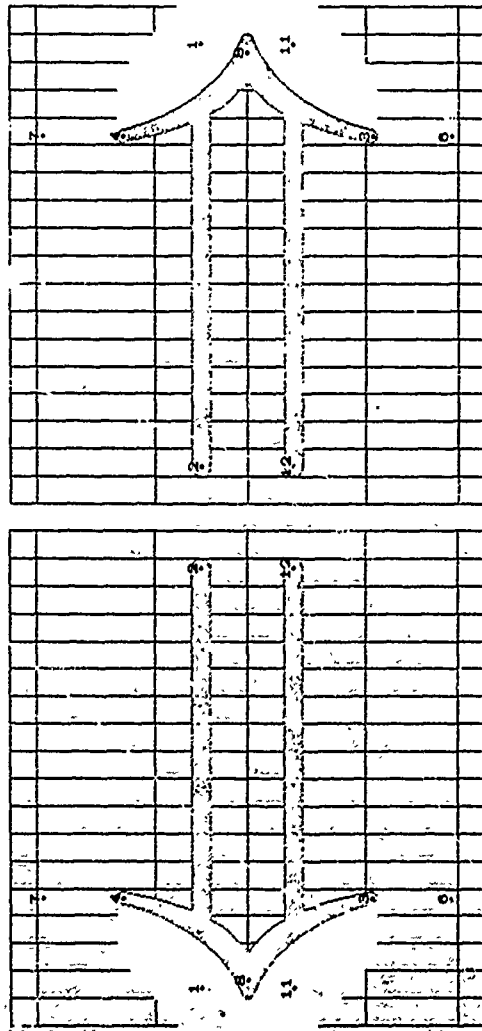
```

"Much greater sign";
call charbegin(0.46, 18, 0, 0, 5[px, ph] + prt/2, 5[px, ph] + prt/2 -- 2pa, 0);
open; ll0px2 == round u; rt0px1 == round 11.5u; x3 == x2;
y4 = good10, 5[rt0, h]; .5[y2, y1] == y1 == good106;
w10 draw 2...1...1...3;
rt0px1 == round (u - u); x2 == x3; x1 - x1 == x5 - x2; y1 == y1; y2 == y1;
draw 5...4...4...6;
draw 5...4...4...6;

```

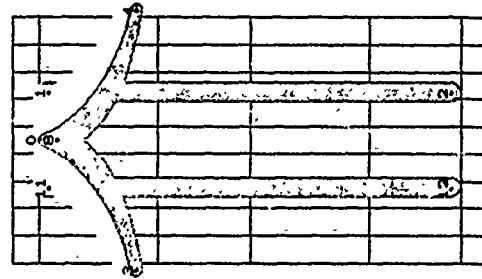
```

"Similar or equal sign";
call charbegin(0.47, 18, 0, px -- pc + prt/2 -- pa, prt/2 -- pa, 0);
ifn; top1u1h == round(a + (m - r) + cpx); top1u1h -- bot1u1h == round(m - e);
let1u3 == round u; y5 == .5(y1, y4); y5 == y1; x1 == r - x1; x5 == r -- x5;
call "a zdraw(3, 1.5, 2, 4, w0, w10 + deltax, 7.5u(e - m));
draw; y5 == y1; a - y5 == round(m - e); lft1u5 == round u; x7 == r -- x0;
w0 draw 6...7;
%
```



```
"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; r10x2=round(r-u); x11=x1; x12=x2;
y5=y8=good106; y1=y2; y11=y12; .5[y1,y11]=y5; y1-y11=round(m--e);
w10 draw 1..2; draw 11..12;
hpen; lft1x8=x0;
x5-x8=x3-x2=-fixwidth[3u,6u]--eps; x1=x11=x8=x7;
y5-y8=y11-y1=y4-y1=y1-y1=.24h+.eps;
hpen; w10+w1 ddraw (5..8..3(.6),11..11;
hpen; draw (w1[5..8]..w1[3](.6);
hpen; w10+w1 ddraw (5..8..4(.7),1..1;
hpen; draw (w1[5..8]..w1[4](.7).

"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; r10x1=x0=round(r-u); x11=x1; x12=x2;
y5=y8=good106; y1=y2; y11=y12; .5[y1,y11]=y5; y1-y11=round(m--e);
w10 draw 1..2; draw 11..12;
hpen; r11x8=x0;
x5-x8=x3-x2=fixwidth[3u,6u]+eps; x1=x11=x8=x7;
y5-y8=y11-y1=y1-y1=.24h+.eps;
rpen; w10+w1 ddraw (5..8..3(.6),11..11;
hpen; draw (w1[5..8]..w1[3](.6);
rpen; w10+w1 ddraw (5..8..4(.7),1..1;
hpen; draw (w1[5..8]..w1[4](.7).
```

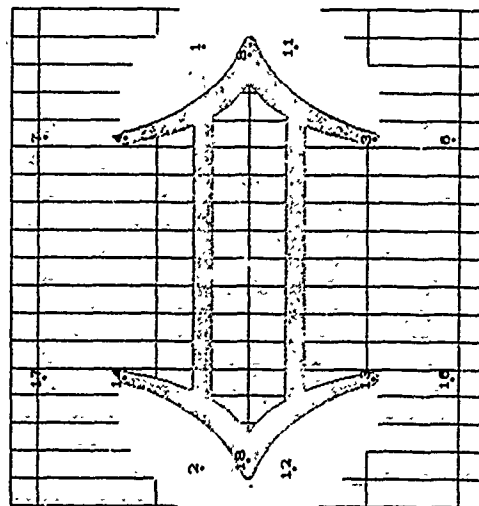


```
"Double upward arrow";
call charbegin('052,9,0,0,ph,ph-2pa,0);
cpen; top10y1=y0=h; .5[y1,y1]=a; y11=y1; y12=y2;
x0=x3=x8=good10.5r; x1=x2; x11=x12;
.5[x1,x11]=x3; x1-x11=round 3.5u;
w10 draw 1..2; draw 11..12;
vpen; top1x8=y0;
hpen; 2u draw 0..8; rpen; 2u draw 0..8;
y5-y8=y11-y1=.24h+.eps; y1=y11=y8=y7;
x1-x8=x12-x2=x1-x1=x1-x1=.3u+.eps;
hpen; 2u draw (5..8..3(.6);
vpen; draw (w1[5..8]..w1[3](.6);
rpen; 2u draw (5..8..4(.7);
vpen; draw (w1[5..8]..w1[4](.7).
```




7

8



15

5

```

"Double downward arrow";
call charbegin("053;9,0,0,ph,plh-2pa,0);
open; topu[2] = h; .5[y1,y2] = a; y0 = bot[0]h; y1 = y1; y12 = y2;
x0 = x2 = x3 = good10.5r; x1 = x2; x1 = x12;
.5[x1,x1] = x2; x1 - x11 = round 3.5r;
w10 draw 1..2; draw 11..12;
vpen; bot[2]x = w;
lpen#; 2u draw 0..8; rpen#; 2u draw 0..8;
y1 - y8 = y1 - y1 = -2.4h - eps; y1 = y1 = w = w;
x3 - x0 = x2 - x3 = x1 - x1 = x1 - x1 = 3u + eps;
lpen#; 2u draw (5..8..3(.6);
vpen; draw (w1[5..8..]w1[3(.6);
rpen#; 2u draw (5..8..]w1[3(.7);
vpen; draw (w1[5..8..]w1[3(.7);

"Double left-and-right arrow";
call charbegin("054,18,0,0,24ph+.5prt+.5(px--pc)+pa,
.24ph+.5prt+.5(px--pc)-pa,0);
open; lft[0]x2 = x10 = round u; rft[0]x1 = x0 = round(r--u); x11 = x1; x12 = x2;
y3 = y4 = good10r; y1 = y2; y11 = y12; .5[y1,y11] = y1; y1 - y11 = round(m--e);
w10 draw 1..2; draw 11..12;
lpen; rft[2]x = x0; lft[2]x18 = x10;
x3 - x8 = x3 - x1 = x11 - x18 = x18 - x15 = fixwidth[3u,6u] + eps;
x3 = x1 = x4 = x2; x13 = x11 = x10 = x17;
y1 - y6 = y11 - y1 = y1 - y1 = y1 - y1 = .24h + eps;
y13 = y3; y11 = y1; y15 = y1; y16 = y1; y17 = y1;
rpen#; w10 + w1 ddraw (5..8..3(.6),11..11;
lpen; draw (w1[5..8..]w1[3(.6);
rpen#; w10 + w1 ddraw (5..8..]w1[3(.7);
lpen; draw (w1[5..8..]w1[3(.7);
rpen#; w10 + w1 ddraw (15..18..13(.16),12..12;
lpen; draw (w1[15..18..]w1[13(.16);
rpen#; w10 + w1 ddraw (15..18..14(.17),2..2;
lpen; draw (w1[15..18..]w1[14(.17);

```

% stems

% clean the top

% erase excess at left

% left point

% erase excess at right

% right point

% bars

% erase excess at lower right

% lower right point

% erase excess at upper right

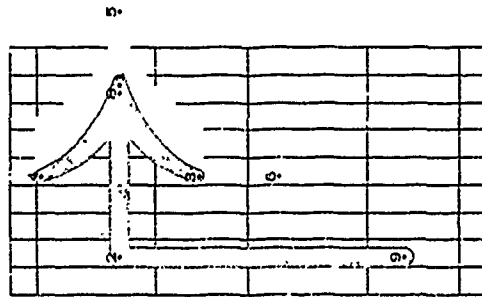
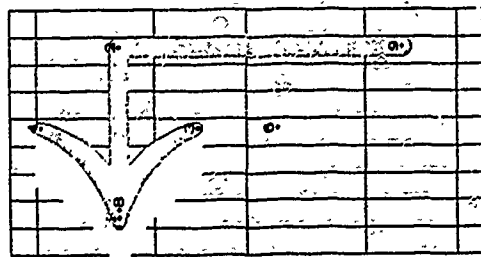
% upper right point

% erase excess at lower left

% lower left point

% erase excess at upper left

% upper left point



"Left shift sign";
 call charbegin(055, 9, 0, 0, ph, 5pd, 0);
 cpen; lft10x1 = x0 = round u; r10x2 = round(r - u);
 y1 = y2 = y3 = y4 = good10.75h;
 w10 draw 1..2;
 x3 = x2; bot10y = -.5d; draw 2..9;
 lpen; lft1x3 = x0;
 x3 - x4 = x3 - x2 = -3u - eps; x1 = x2 = x0 = x1;
 y3 - y4 = y1 - y2 = y1 - y1 = y1 - y1 = .24h + eps;
 lpen; w10 + w1 draw (5..8) .3(.6);
 lpen; draw (w15..8) .3(.6);
 lpen; w10 + w1 draw (5..8) .4(.7);
 lpen; draw (w15..8) .4(.7);

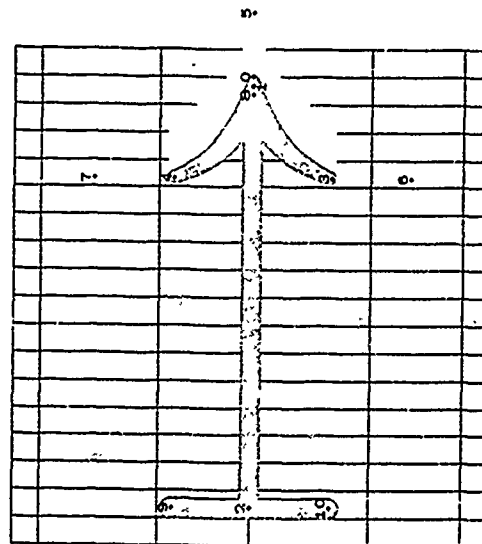
% bar
 % stem
 % erase excess at lower left
 % erase excess at upper left
 % upper point

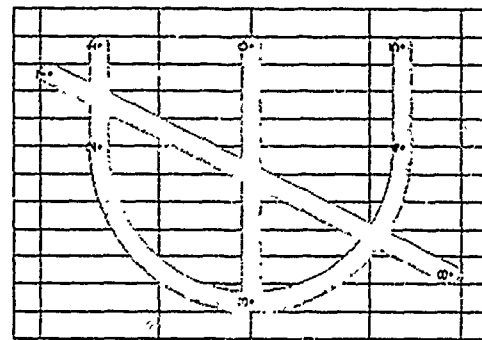
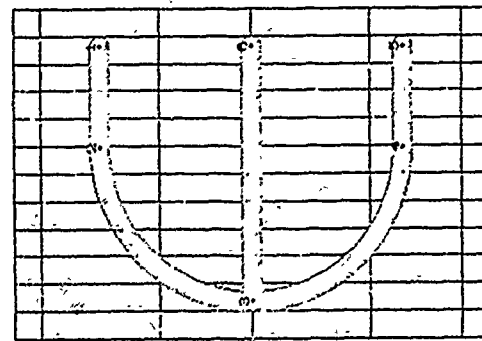
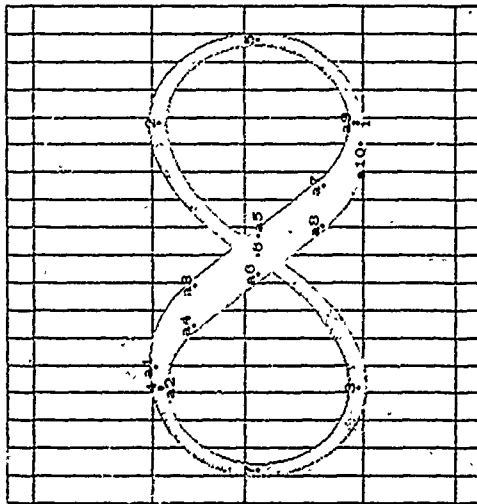
"Right shift sign";
 call charbegin(056, 9, 0, 0, ph, 5pd, 0);
 cpen; lft10x2 = round u; r10x1 = x0 = round(r - u);
 y1 = y2 = y3 = y4 = good10.75h;
 w10 draw 1..2;
 x1 = x2; bot10y = -.5d; draw 2..9;
 lpen; r1x3 = x0;
 x3 - x4 = x3 - x2 = 3u + eps; x1 = x2 = x0 = x1;
 y1 - y4 = y1 - y2 = y1 - y1 = y1 - y1 = .24h + eps;
 rpen; w10 + w1 draw (5..8) .3(.6);
 rpen; draw (w15..8) .3(.6);
 rpen; w10 + w1 draw (5..8) .4(.7);
 rpen; draw (w15..8) .4(.7);

% bar
 % stem
 % erase excess at lower right
 % lower point
 % erase excess at upper right
 % upper point

"Maps-to relation";
 call charbegin(057, 18, 0, 0, 24ph + 5prt + p2, 2lph + 5prt - p2, 0);
 cpen; lft10x2 = round u; r10x1 = x0 = round(r - u);
 y1 = y2 = y3 = y4 = y5 = y6 = good10.6;
 w10 draw 1..2;
 lpen; r1x3 = x0;
 x3 - x4 = x3 - x2 = fixwidth(3u, 6u) + eps; x1 = x2 = x0 = x1;
 y1 - y4 = y1 - y2 = y1 - y3 = y1 - y4 = y1 - y1 = .24h + eps;
 rpen; w10 + w1 draw (5..8) .3(.6);
 lpen; draw (w15..8) .3(.6);
 rpen; w10 + w1 draw (5..8) .4(.7);
 lpen; draw (w15..8) .4(.7);
 x3 = x10 = x2; y3 = y1; y10 = y1;
 cpen; w10 draw 9..10.

% bar
 % erase excess at lower right
 % lower point
 % erase excess at upper right
 % upper point
 % stem





```

"Prime symbol (intended as superscript only)";
call charbegin('060,4.5,0,0,8ph,0,0);
new w8,w9; w8 = round(51w0,w1); w9 = round(bold + 2deltaw);
open; top9y1 = round(8h); r10x1 = r; bot8y1 = 0; if9x2 = 0;
call cdraw(1,2,99,98).

"infinity";
call charbegin('081,18,0,0,px,0,.5pxslant-.5pa);
new w8,w9; w9 = round(251w0,w1); w8 = 2{w1,w2};
open; top9y1 = m + oo; bot9y1 = -oo; y1 = y1; y1 = y1;
x1 = x2; x1 = x1; y1 = y2 = y1 = 5{y1,y2};
if10x1 = round u; r10x1 = round(r - u); x0 = 5{x1,x2};
new ss,mss;
if w0 = w8 mss = ss; ss = -u/m;
else: mss = .75ss; ss = -6u/m;
fi.
call 'a zdraw(7,4,6,1,5,w8,w8,ss);
w1 draw 5{0,1} 2{-1,0}..6{mss,-1}..3{-1,0}..
7{0,1}.

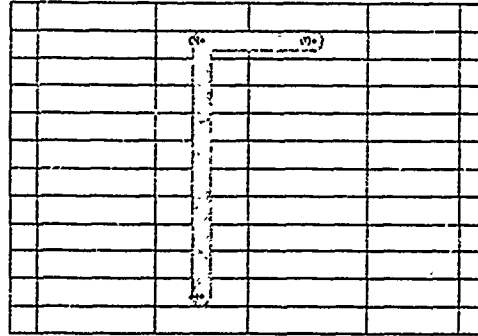
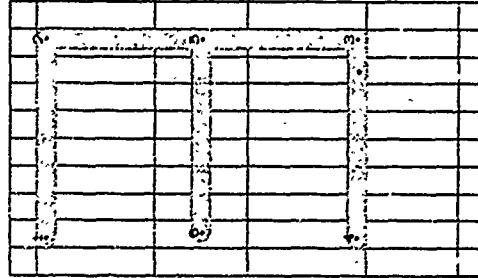
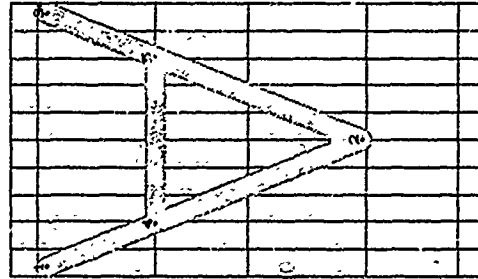
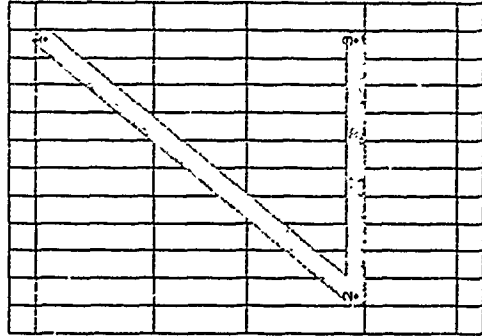
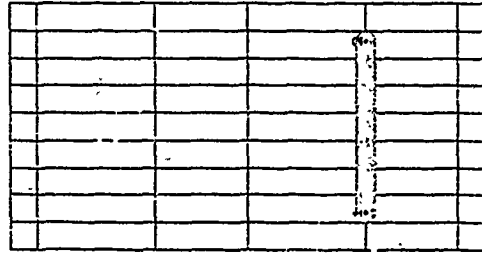
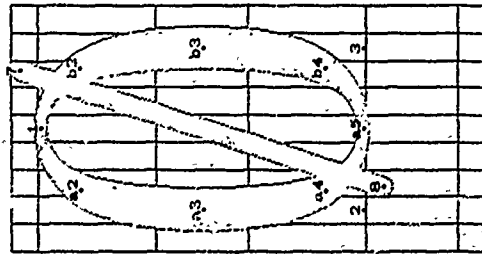
"Element sign";
call charbegin('082,12,0,0,5{px,ph} + prt/2,.5{px,ph} + prt/2 - 2pa,0);
open; r10x1 = round(r - u); if10x1 = round u;
x1 = x0 = x1; x2 = x1 = .5(r + 2u);
y1 = y1 = good10.5{m,h}; y1 = y0 = a; y1 = y1; 5{y1,y1} = y1;
w10 draw 1..2{-1,0}..3{0,-1}..4{1,0}..5;
draw 3..6.

% howl
% bar

"Nonlement sign";
call charbegin('083,12,0,0,ph,ph - 2pa,0);
open; r10x1 = round(r - u); if10x1 = round u;
x1 = x0 = x1; x2 = x1 = .5(r + 2u);
y1 = y1 = good10.5{m,h}; y1 = y0 = a; y1 = y1; 5{y1,y1} = y1;
w10 draw 1..2{-1,0}..3{0,-1}..4{1,0}..5;
draw 3..6;
r10x1 = round(r - 2u); if10x1 = round 2u; top10y1 = h; 5{y1,y1} = a;
draw 7..8.

% bowl
% bar
% diagonal

```



```

"Empty set symbol";
call charbegin('064, 9, 0, 0, ph + pb, 0);
if fixwidth = 0: new save; save = sqrttwo; new sqrttwo;
    sqrttwo = sqrt(1.23114113save);
fi;
hpen;
if w2 > 1.5u: lft2x2 = round.75u;
else: x2 = good2 1.5u;
fi;
x1 = r - x1;
top10y1 = h + oo; bot10y1 = -oo; y1 = y1;
call "a dir. (1, 2, w2); call "b dir. (1, 3, w2);
if fixwidth = 0: new sqrttwo; sqrttwo = save;
fi;
cpen; r10x2 = round(r - 2u); lft10x4 = round2u; top10y1 = h + b; bot10y1 = -b;
w10 draw 7..8.
% axis of left-right symmetry
% bowl

"Underline";
call charbegin('065, 9, 0, 0, 0, 0, 0);
cpen; lft10x1 = round u; x2 = r - x1; y1 = y2 = 0;
w10 draw 1..2.
% bar

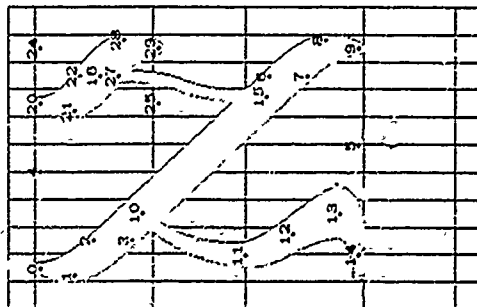
"Angle sign";
call charbegin('066, 12, 0, 0, 0, ph, 0, 0);
cpen; lft10x2 = round u; x1 = x1 = r - x2; top10y1 = h; bot10y1 = 0; y1 = y2;
w10 draw 1..2..2..3.
% diagonal and bar

"Universal quantifier";
call charbegin('070, 10, 0, 0, 0, ph, 0, 0);
cpen; lft10x1 = 0; x2 = good10.5r = 5[x1, x1];
top10y1 = h; y1 = y1; bot10y2 = -o; y1 = y1 = good10m;
new aa, bb; x1 = aa[x1, x2]; y1 = aa[y1, y1]; x3 = bb[x1, x2]; y3 = bb[y1, y2];
w10 draw 1..2 2..3;
draw 4..5.
% diagonals
% bar

"Existential quantifier";
call charbegin('071, 10, 0, 0, 0, ph, 0, 0);
cpen; lft10x1 = round u; r10x2 = round(r - u); x1 = x1 = x2; x1 = x1 = 25u = x1;
top10y1 = h; bot10y1 = 0; y1 = y1; y3 = y3 = 5[y1, y1]; y1 = y1;
w10 draw 1..2..2..3..3..4;
draw 5..6.
% upper bar, stem, lower bar
% middle bar

"Logical NOT";
call charbegin('072, 12, 0, 0, 0, px, 0, 0);
cpen; lft10x1 = round u; x2 = x1 = r - x1;
y1 = y2 = good10.5[e, m]; y1 = y1 = 1.2(m - e);
w10 draw 1..2..2..3.
% bar and stem

```



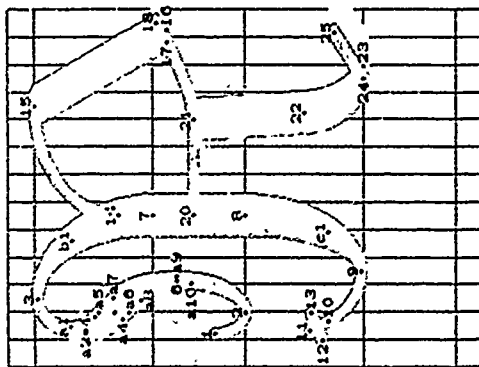
```

"Jehew letter aleph",
call charbegin(073, 11, 0, 0, ph, 0, 0);
vpen; if(x1 = round u; y1 = {m, h},
x1 + z3 = x1 + x0 = x2 + x1 = x0 + x1 = x1 + x8 = r; x2 = x1 = 2.5u;
new aa. x2 = aa[x1, x8]; y9 = aa[y1, y8];
top:y9 = top:y9; bot:y9 = bot:y9;
x0 = 1.5u; top:y9 = h; y1 = y9; x1 = 5u;
y1 + y2 = y1 + y0 = y2 + y1 = y1 + y2 = y1 + y2 = h;
u, ddraw 0..0{0, -1}..2{x8 - x1, y8 - y1}..6{x8 - x1, y8 - y1} 8{0, -1} 9{ 5},
(1..10..1{0, -1}..3{x8 - x1, y8 - y1}..7{x8 - x1, y8 - y1}..
9{0, -1}..9;
lpen; x0 = 3.5u; new aa; x0 = aa[x1, x8]; y0 = aa[y1, y8];
new w9; w9 = round .5{w9, w1};
x11 = good; 2u; y11 = e;
x12 = .5{x11, x13}; y12 = .5{y11, y13};
x13 = good; 3.5u; y13 = .2e;
x14 = good; 2u; bot:y14 = 0;
draw {w9}10{x8 - y1, x1 - x8}..{w9}11{0, -1}..
|5{w9, w1}12{1.5{x13 - x11}, y11 - y11}..
|w9}13{0, -1}..14{-1, 0};
x15 = 7.75u; new aa; x15 = aa[x1, x8]; y15 = aa[y1, y8];
x16 = .5{x21, x23}; y16 = .5{y21, y23};
vpen; if(x21 = round r; y1 = y1,
x20 + x21 = x21 + x28 = x22 + x21 + x25; x28 = x24,
y20 + y21 = y21 + y28 = y22 + y21 = y2; y2 = 25{m, h},
top:y20 = top:y21; x22 = x27;
x20 = 7.5u; x21 = 9.5u; y20 = y21 = y6;
u, ddraw 20..20{0, -1}..22{x28 - x21, y28 - y21}..28{0, -1} 27{ 25},
(21..20..21{0, -1}..27{x28 - x21, y28 - y21}..
29{0, -1}..29;
draw 15{0, 1}..16{0, 1}.

```

% lower left stroke

% short diagonal
% link



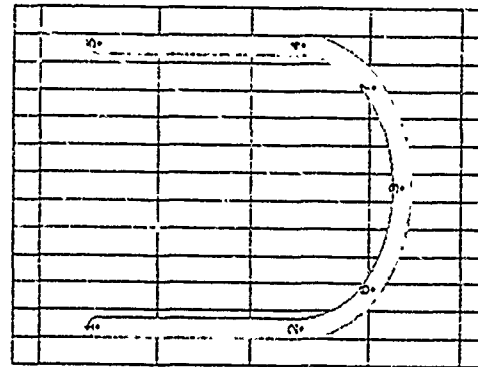
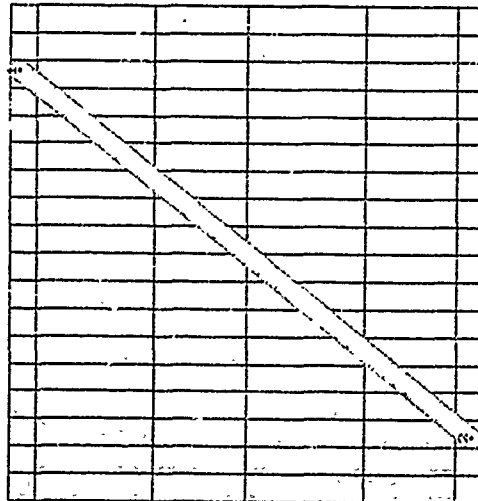
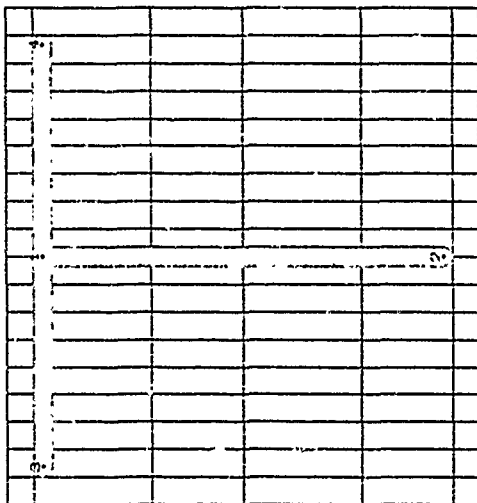
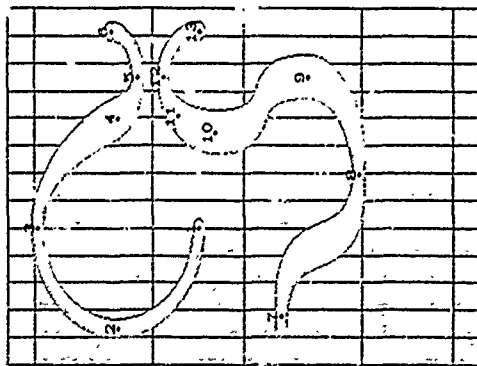
a3
5

```

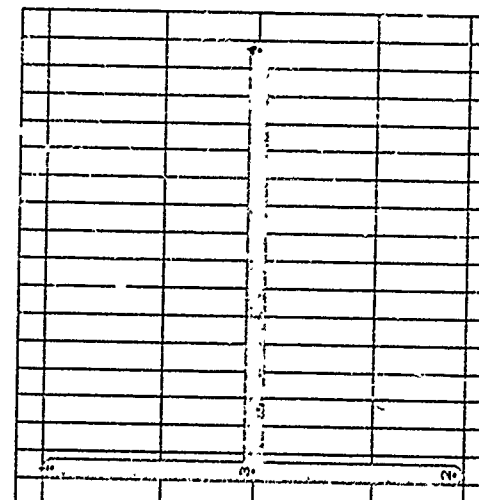
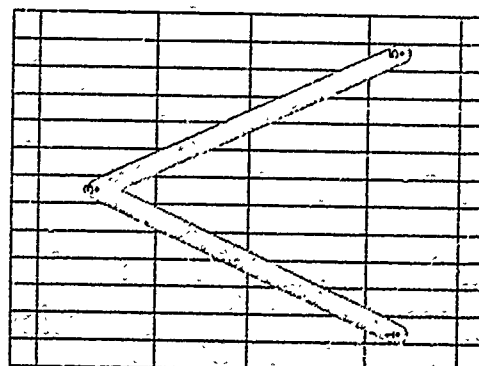
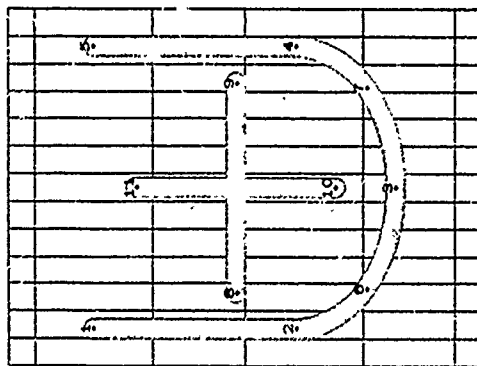
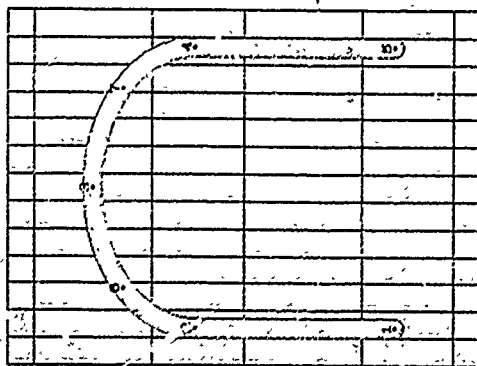
"Upper case Fraktur R";
call charbegin( 074, 13, 0, 0, ph, 0, 0);
new w0; w0y = round .25{w0, w0};
hpen; lt0x = round u; y1 = 1/3{e, m}; x2 = 2u; y2 = c;
w0 draw 1{0, -1}..2{1, 0};
x3 = 2.5u; topy = h + oo; lt0x4 = round u;
x5 = 2c; y5 = 1/3{m, h}; rt0x4 = round 3.5u;
call "a sdraw(3, 4, 5, 6, 2, w0y, w0, -(h - c)/(8u));
x7 = x8 = good(5.5u; y7 = m; y8 = c;
call "b arc(3, 7, w1);
w1 draw 7..8;
x9 = 3.5u; bot0y = -oo; call "c arc(9, 8, w1);
x10 = ju; y10 = .3c; x11 = ju;
x12 = u; y12 = .35c; x13 = 2u; y13 = .45c;
new aa; x14 = x10 + aa(y12 - y13);
y11 = y10 + aa(x13 - x12);
w0 draw 9{-1, 0}..10{x11 - x10, y11 - y10}..11;
draw 12..13;
x14 = x7; y14 = .75h; x15 = 9.5u; topy15 = h + o;
draw 14{0, 1}..15{1, 0};
lt1x16 = 11.5u; y16 = .8h;
w1 draw 15..16;
lt0x17 = 11.5u; y17 = y16;
new aa, bb; vtopx18 = aa{rt1x15, rt1x16}; y18 = aa{y15, y16};
x18 = x17 + bb(y15 - y16); y18 = y17 + bb(x16 - x15);
rpen8; w1 draw 17..18;
x20 = x7; y20 = y17 = 52h; x21 = 9u;
hpen; w0 draw 20..21{1, 0}..17{x14 - x17, y18 - y17}..18;
x22 = 9.25u; y22 = .5c; lt0x23 = lt1x16; bot1y24 = -o;
x23 = 10.5u; y24 = y21; rt0x25 = 12.5u; y25 = c/4;
w1 draw 21..22{x22 - x21, y22 - y21}..23{1, 0};
rpen8; w1 draw 24..25;
hpen; w0 draw 24..25;

```

% point of upper left flourish
% flourish
% upper ink
% stem
% lower link
% perpendicular to 12. 13
% tail
% cross
% top of bowl
% diagonal of bowl
% perpendicular to 15. 16
% erase excess
% bottom of bowl
% lower diagonal
% erase excess at lower right
% point of lower right diagonal



Upper case Fraktur 'r';
 call charbegin('075, 13, 0, 0, ph, 0, 0);
 % (this letter extended to be same width as the l)
 new w0; w0 = round 25{w0, w1};
 lpen; x1 = 5u; y1 = goodh, 5h, lft0p0 = round u, y1 = 75h;
 x1 = x1; top0p0 = h + oo;
 w0 draw l(-1, 0), 2{0, 1}, 3{1, 0};
 x1 = 9u; y1 = 75h, x2 = 10.5u, bot0p0 = 7h + 1;
 r0p0 = 12.5u; l0p0 = 75h + 1;
 draw l0p0{3{1, 0}, l0p0{4{6, -h} .. l0p0{5{1, 0}, 6{0, 1}};
 lft0p0 = round u; y1 = goodh, 25h;
 x1 = 7u; bot0p0 = -oo;
 x2 = good, 10.5u; y1 = h/6;
 x0 = good, 8.5u; y0 = 45h;
 x1 = 1/sqrt(2){x1, x0}; y1 = 1/sqrt(2){y0, y1};
 x2 = x0; top0p0 = 625h;
 r0p0 = round 12.5u; y1 = 5h;
 draw l0p0{7{1, 0}, 8{1, 0}, 9{0, 1}, l0p0{10{0, 1}};
 l0p0{11{1, 0}, l0p0{12{1, 0}, l0p0{13{0, -1}};
 % shoulder
 % upper stroke
 % lower stroke
 "Lattice top",
 open;
 if fixwidth = 0: if pa + 8pu > ph;
 call charbegin('076, 18, 0, 0, ph, ph, 2pa, 0), top0p0 = h,
 else: call charbegin('076, 18, 0, 0, 8pu + pa, 8pu + pa, 0), top0p0 = a + 8u,
 fi;
 else: call charbegin('076, 9, 0, 0, 3.5pu + pa, 3.5pu + pa, 0), top0p0 = a + 3.5u;
 fi;
 5{y0, y1} = a; x1 = x2 = 5r;
 lft0p0 = round u; x1 = r = x1, y1 = y1 = y1,
 w0 draw 1..2;
 draw 3..4.
 "Zero-width slash to negate a relation";
 call charbegin('100, 18, 0, 0, ph + ph, ph + ph + 2pa, 0);
 cl arwd 0; charwd 0;
 open; r1, x1 = round(r - 2u), lft0p0 = round 2u;
 top0p0 = h + b, 5{y0, y1} = a,
 w0 draw 1..2.
 "Set union sign",
 call charbegin('133, 13, 0, 0, ph, ph, 2pa, 0),
 open; lft0p0 = round u; x1 = r = x1, x2 = r = x1, x1 = r = x1,
 y1 = goodh, 5{y0, y1} = a, y2 = y1 = 4{y0, y1}, y1 = y1,
 call qcirc(3, 6, 2, w0); call qcirc(3, 7, 4, w0);
 w0 draw 1..2; draw 4..5.
 % stem
 % bar
 % diagonal
 % cup
 % stems



```

"Set intersection sign";
call charbegin('134, 13, 0, 0, ph, ph - 2pa, 0);
open; lift10x1 = round u; x2 = x1; x1 = r - x1; x1 = x2 = r - x1;
y1 = good10(5[m, h]); .5[y1, y1] = a; y2 = y1 = {y1, y1} y5 = y1;
call qcirc(3, 6, 2, w10); call qcirc(3, 7, 4, w10);
w10 draw 1..2; draw 4..5.

% cap
% stems

"Multiset union sign";
call charbegin('135, 13, 0, 0, ph, ph - 2pa, 0);
open; lift10x1 = round u; x2 = x1; x1 = r - x1; x1 = x2 = r - x1;
y1 = good10(5[m, h]); .5[y1, y1] = a; y2 = y1 = {y1, y1} y5 = y1;
call qcirc(3, 6, 2, w10); call qcirc(3, 7, 4, w10);
w10 draw 1..2; draw 4..5;
y8 = y1 = .47[y1, y1]; x8 = r - x2 = x1 + 1.75w10 - e10;
x10 = x1 = x2; .5[y10, y1] = y8; y11 = y10 = x2 - x8;
draw 8..9; draw 10..11.

% enclosed plus sign

"Lattice infimum (logical AND) sign";
call charbegin('136, 13, 0, 0, ph, ph - 2pa, 0);
open; lift10x1 = round u; x2 = r - x1; x3 = r - x2; x5 = r - x1;
y1 = good10(5[m, h]); .5[y1, y1] = a; y5 = y1;
w10 draw 1..3 .3..5.

% diagonals

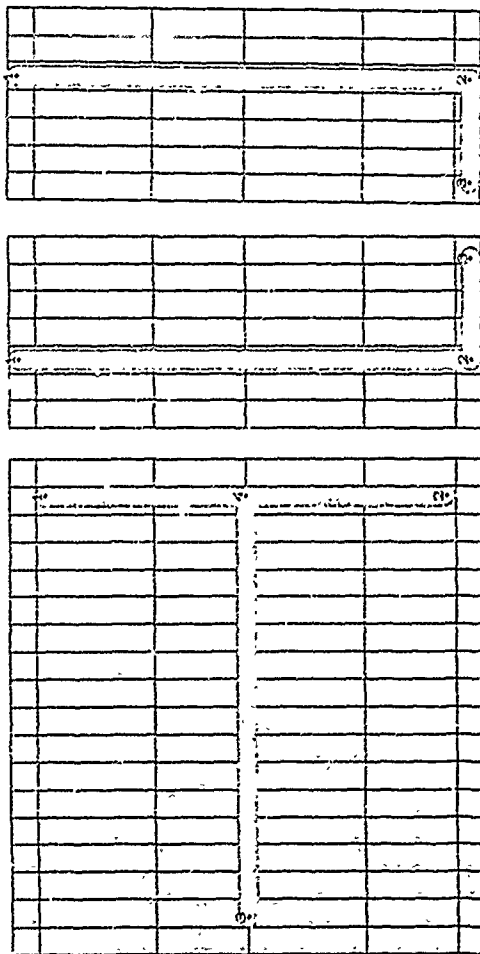
"Lattice supremum (logical OR) sign";
call charbegin('137, 13, 0, 0, ph, ph - 2pa, 0);
open; lift10x1 = round u; x2 = r - x1; x3 = r - x2; x5 = r - x1;
y1 = good10(5[m, h]); .5[y1, y1] = a; y5 = y1;
w10 draw 1..3 .3..5.

% diagonals

"Left turnstile";
open;
if fixwidth = 0; if pa + 8pu > ph.
    call charbegin('140, 18, 0, 0, ph, ph - 2pa, 0); top10y1 = h;
else: call charbegin('140, 18, 0, 0, 8pu + pa, 8pu - pa, 0); top10y1 = a + 8u;
fi;
else: call charbegin('140, 9, 0, 0, 3.5pu + pa, 3.5pu - pa, 0); top10y1 = a + 3.5u;
fi;
.5[y1, y1] = a; x1 = x2 = x1;
lift10x1 = round u; x1 = r - x2; y1 = y1 = a;
w11 draw 1..2;
draw 3..4.

% stem
% bar

```

"Right ternstile";
 open;
 if fixwidth = 0: if $pa + 8pu > ph$:
 call charbegin('141, 18, 0, 0, ph, ph - 2pa, 0); top₁₀h = h;
 else: call charbegin('141, 18, 0, 0, 8pu + pa, 8pu - pa, 0); top₁₀h = a + 8u;
 fi;
 else: call charbegin('141, 9, 0, 0, 3.5pu + pa, 3.5pu - pa, 0), top₁₀h = a + 3.5u,
 fi;
 .5[y₁, y₂] = a; x₁ = x₂ = x₃;
 w₁₀ draw 1 . 2;
 draw 3 . 4.

% stem
 % bar

"Left floor bracket";
 call charbegin('142, 7, 0, 0, ph + pb, ph + pb - 2pa, 0);
 open; x₁ = x₂ = good₁₀(2.5u); x₃ = x₁ + 3.75u + eps;
 top₁₀h = h + b; .5[y₁, y₂] = a; y₁ = y₂;
 w₁₀ draw 1 . 2...2...3.

% stem and bar

"Right floor bracket";
 call charbegin('143, 7, 0, 0, ph + pb, ph + pb - 2pa, 0),
 open; x₁ = x₂ = good₁₀(r - 2.5u); x₃ = x₁ - 3.75u - eps,
 top₁₀h = h + b; .5[y₁, y₂] = a; y₁ = y₂;
 w₁₀ draw 1...2...2...3.

% stem and bar

"Left ceiling bracket";
 call charbegin('144, 7, 0, 0, ph + pb, ph + pb - 2pa, 0);
 open; x₁ = x₂ = good₁₀(2.5u); x₃ = x₁ + 3.75u + eps;
 top₁₀h = h + b; .5[y₁, y₂] = a; y₁ = y₂;
 w₁₀ draw 3 . 1 . 1...2.

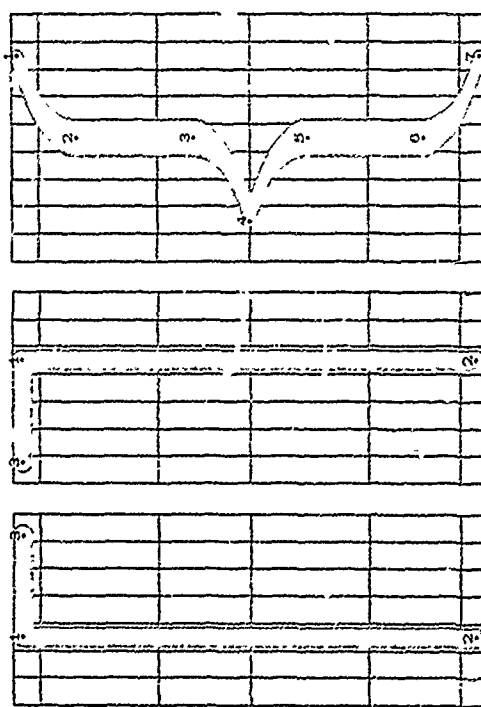
% bar and stem

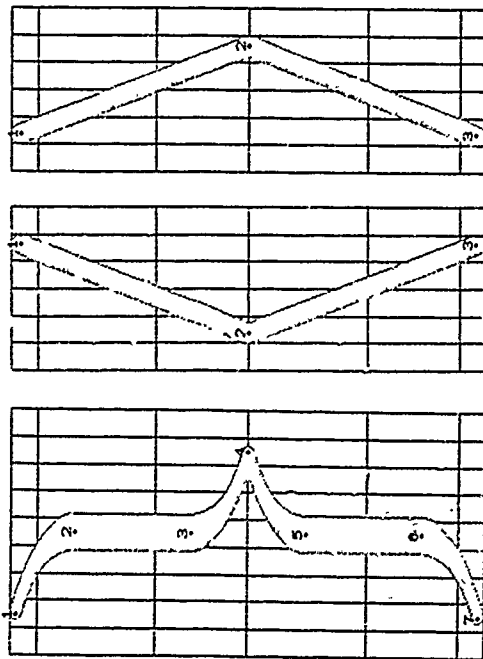
"Right ceiling bracket";
 call charbegin('145, 7, 0, 0, ph + pb, ph + pb - 2pa, 0),
 open; x₁ = x₂ = good₁₀(r - 2.5u); x₃ = x₁ - 3.75u - eps;
 top₁₀h = h + b; .5[y₁, y₂] = a; y₁ = y₂;
 w₁₀ draw 3...1...1...2.

% bar and stem

"Left brace";
 call charbegin('146, 9, 0, 0, ph + pb, ph + pb - 2pa, (ph + pb) slant + 5pw - pw),
 open; x₂ = x₃ = x₁ = x₀ = good, 5r; x₁ = x₂ = x₃ = x₁ = 3u + eps, x₁ = x₂,
 top₁₀h = h + b, y₁ = .5[y₁, y₂] = .5[y₁, y₂] = good, a,
 y₁ = y₂ = y₃ = y₁ = (y₁ - y₂)/4,
 draw {w₁₀#11(3(x₂ - x₁), y₂ - y₁) {w₁₀#12(0, -1) {w₁₀#13(0, -1)
 {w₁₀#14(3(x₁ - x₂), y₁ - y₂),
 draw {w₁₀#17(3(x₀ - x₂), y₂ - y₁) {w₁₀#16(0, 1) {w₁₀#15(0, 1)..
 {w₁₀#14(3(x₁ - x₂), y₁ - y₂).

% upper stem
 % lower stem





"Right brace";
 call charbegin('147, 9, 0, 0, ph + pb, ph + pb - 2pa, {ph + pb} slant + 5pwi - 4pu);
 lpen; $x_1 = x_2 = x_0 = \text{good}_1 5r$; $x_1 - x_2 = x_2 - x_1 = -3u - \text{eps}$, $x_1 = x_2$,
 $\text{top}_0 y_1 = h + b$; $y_1 = .5[y_1, y_2] = .5[y_1, y_2] = \text{good}_0 a$;
 $y_1 - y_2 = y_2 - y_1 = (y_1 - y_2)/4$;
 draw $\{w_0\} \{3(x_2 - x_1), y_2 - y_1\} \cdot \{w_1\} \{2(0, -1) \dots [w_1] \{3(0, -1) \dots$
 $\{w_2\} \{3(x_1 - x_2), y_1 - y_2\}$;
 draw $\{w_3\} \{7\{3(x_1 - x_2), y_1 - y_2\} \cdot \{w_4\} \{5\{0, 1\} \cdot$
 $\{w_5\} \{3(x_1 - x_2), y_1 - y_2\}$;
 % upper stem
 % lower stem

"Left angle bracket";
 call charbegin('150, 6, 0, 0, ph + pb, ph + pb - 2pa, 0);
 open; $r_{10} x_1 = \text{round}(r - u)$; $x_1 = x_2$, $\text{lt}_{10} x_2 = \text{round fixwidth}[u, \{u\}$;
 $\text{top}_0 y_1 = h + b$; $.5[y_1, y_2] = y_2 = \text{good}_0 a$;
 $w_0 \text{ draw } 1..2..2..3$.

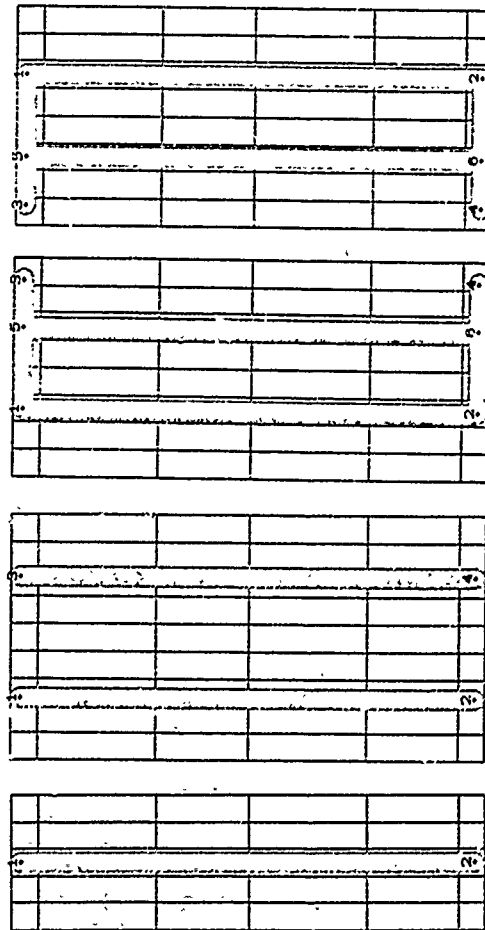
"Right angle bracket";
 call charbegin('151, 6, 0, 0, ph + pb, ph + pb - 2pa, 0);
 open; $r_{10}(r - x_1) = \text{round}(r - u)$; $x_1 = x_2$, $\text{lt}_{10}(r - x_2) = \text{round fixwidth}[u, \{u\}$;
 $\text{top}_0 y_1 = h + b$; $.5[y_1, y_2] = y_2 = \text{good}_0 a$;
 $w_0 \text{ draw } 1..2..2..3$.

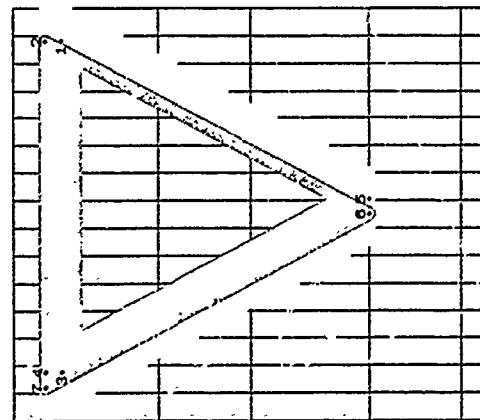
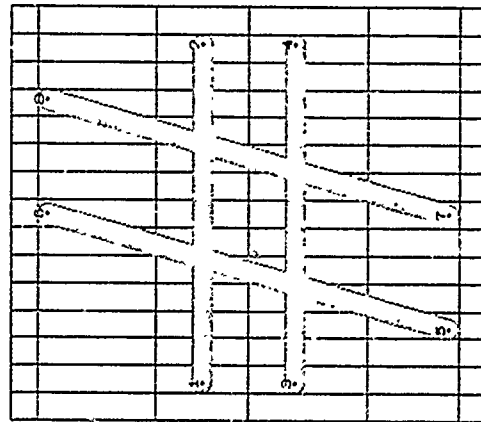
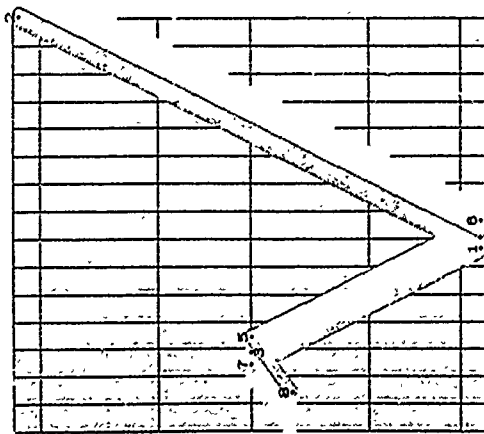
"Vertical line (absolute value or length)";
 call charbegin('152, 5, 0, 0, ph + pb, ph + pb - 2pa, 0);
 open; $x_1 = x_2 = \text{good}_{10} 5r$, $\text{top}_0 y_1 = h + b$, $.5[y_1, y_2] = a$,
 $w_0 \text{ draw } 1..2$.

"Double vertical line (norm or cardinality)";
 call charbegin('153, 9, 0, 0, ph + pb, ph + pb - 2pa, 0);
 open; $x_1 = x_2 = \text{good}_{10} 25r$; $\text{top}_0 y_1 = h + b$; $.5[y_1, y_2] = a$;
 $x_3 = x_1 = r - x_1$; $y_1 = y_2$; $y_1 = y_2$;
 $w_0 \text{ draw } 1..2$; draw 3..4

"Double left bracket";
 call charbegin('154, 8, 0, 0, ph + pb, ph + pb - 2pa, 0);
 open; $x_1 = x_2 = \text{good}_{10} 25r$; $x_1 = x_2 = x_1 + 475u + \text{eps}$, $x_3 = x_0 = x_1 + \text{round } 3u$,
 $\text{top}_0 y_1 = h + b$; $.5[y_1, y_2] = a$; $y_1 = y_2 = y_1$; $y_1 = y_2 = y_2$,
 $w_0 \text{ draw } 3 \cdot 1..2..2..4$;
 draw 5..6.

"Double right bracket";
 call charbegin('155, 8, 0, 0, ph + pb, ph + pb - 2pa, 0);
 open; $x_1 = x_2 = \text{good}_{10}(r - 25u)$;
 $x_3 = x_1 = x_1 - 475u - \text{eps}$; $x_3 = x_0 = x_1 - \text{round } 3u$;
 $\text{top}_0 y_1 = h + b$; $.5[y_1, y_2] = a$; $y_1 = y_2 = y_1$; $y_1 = y_2 = y_2$;
 $w_0 \text{ draw } 3..1..1..2..2..4$;
 draw 5..6.





```

"Radical sign";
call charbegin('160,15,0,0,ph+pb,ph+pb-2pa,0);
hpen; x1 = goodu0(2u); x2 = r + 1, topu2 = h + b,
y1 = y2 = y; y1 = goodu0(2u); y2 = y1 + b,
x1 = 1.5[x2,x1]; lt10x1 = lt10x1; rt10x1 = rt10x1;
lt10x1 = lt10x1; rt10x1 = rt10x1;
x1 = x2 - u; new x1; x2 = aa[x1,x2]; y1 = aa[y1,y2];
hpen; w2 draw 3..4;
w10 draw 7..4; w0 draw 5..6;
hpen; w2 draw 8..5;
hpen; w0 draw 8..5;
rpen; w2 draw 1..2;
hpen; w10 draw 1..2.

"Sharp symbol (number sign or hash mark)";
call charbegin('161,15,0,0,ph,ph-2pa,0);
cpen; lt10x1 = round u; x1 = x1; x2 = x1 + r - x1;
y1 = y2; y1 = y1; y2 = y1 - y1 = round(m - c), .5[y1,y1] = a,
w10 draw 1..2;
draw 3..4;
x1 = 2u = x1; x2 = 2u = x2; x3 = x1 - x1; x4 = x1 - x1 = fixwidth[0,-3u],
y1 = y1; y2 = y2; topu2 = h; .5[y1,y1] = a,
draw 5..6;
draw 7..8.

"Nabla or backwards-difference operator";
call charbegin('162,15,0,0,ph,0,0);
hpen; rt0x2 = round(r - u); lt10x1 = round u; topu2 = h, top y1 = h,
bot y2 = -c; y1 = y1;
x1 = x2 = x1 - x1; lt10x1 = lt10x1; x1 = x2, x1 = x1;
vpen; top y1 = h; y1 = y1;
w1 draw 1..3;
hpen; w1 draw 5..4;
rpen; w1 draw 5..4;
hpen; w1 draw 6..2; w1 draw 6..2;
y1 = y1; lt10x1 = lt10x1;
w1 draw 2..7..7..6.

% left diagonal
% sharpen the corners
% erase excess at upper left
% serif
% erase excess at lower right
% right diagonal

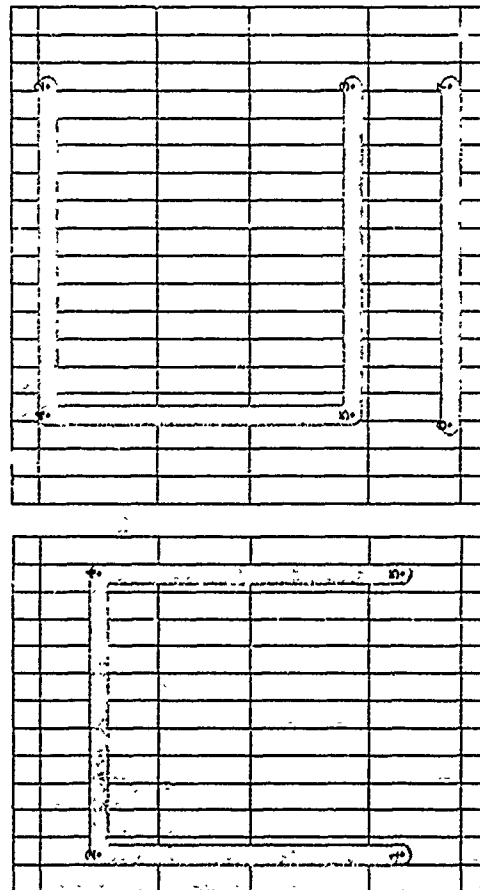
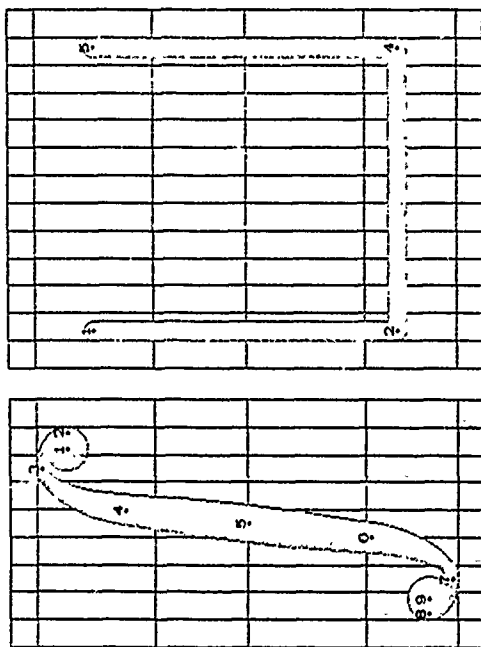
% upper bar
% lower bar

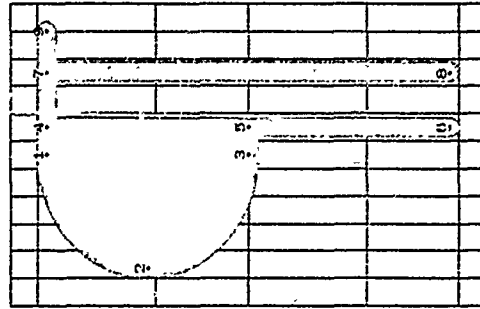
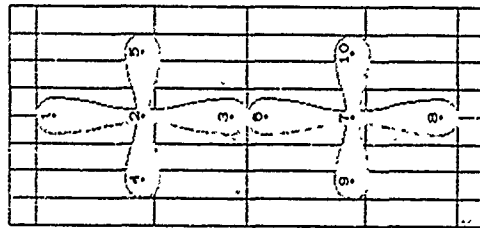
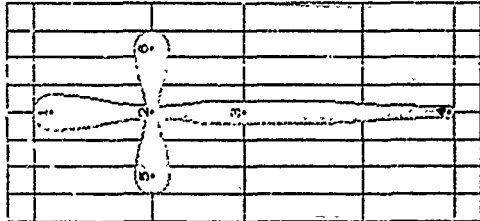
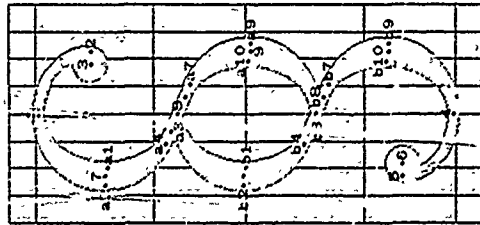
% left diagonal
% right diagonal

% bar line
% erase excess at upper left
% left diagonal
% erase excess at right
% right diagonal

% sharpen upper left corner

```

[illegible]



```

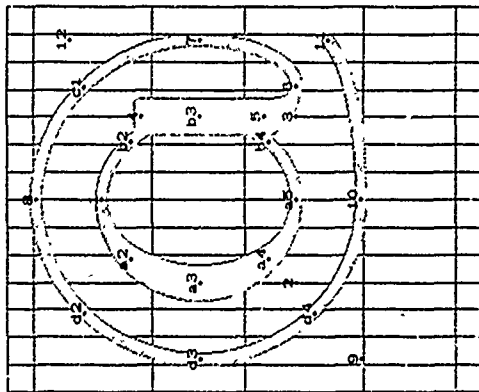
"Section sign";
call charbegin(170, 8, 0, 0, ph, pd, 0);
hpen; x1 = x1 + .5r; rt2 = rt1 + round(1.5u); lt0x = lt0x + round(1.5u);
top0y = h + .oo; bot0y = -d - .oo;
y2 = y1 + .125[y0, y1]; y3 = y2 + .125[y0, y1];
u0 draw 1{1, 0}..2{0, -1};
draw 4{-1, 0}..5{0, 1};
cpen; w1 draw 3; draw 6;
x1 = x1 + x2; x3 = x1 + x2; x5 = x10 = x1;
y2 = 1/2[y0, y1]; y3 = 1/2[y0, y1];
call "a sdraw(1, 7, 8, 9, 10, w1 - delta, w0, -(h + d)/(36u));
call "b sdraw(8, 11, 10, 12, w1 - delta, w0, -(h + d)/(36u))

"Dagger mark";
call charbegin(171, 8, 0, 0, ph, pd, 0);
cpen; x1 = x2 = x3 = x1 = good1(.5r); lt1x = round u; x0 - x2 = x2 - x5;
top1y = h; bot1y = -d; y2 = y3 = y4 = good1m; y1 = c;
w1 draw 1; draw 5; draw 6;
hpen; draw w11..w12;
draw w12 (round 5[w0, w1] - cps#13..w14;
draw w12 (round 5[w0, w1] - cps#13..w14;
vpen; draw w15..w16; draw w16..w17;

"Double dagger mark";
call charbegin(172, 8, 0, 0, ph, pd, 0);
cpen; x1 = x2 = x3 = x4 = x5 = x6 = good1(.5r); lt1x = round u; x1 - x2 = x2 - x5;
x3 = x4; x10 = x5;
top1y = h; bot1y = .5[y0, y1]; top1y = -d;
y2 = y1 = y3 = good1 5[y0, y1]; y3 = y4 = good1 5[y0, y1];
w1 draw 1; draw 3; draw 5; draw 6; draw 8; draw 9; draw 10; % bulbs
hpen; draw w11..w12; draw w13..w14;
draw w16..w17; draw w18..w19;
vpen; draw w14..w15; draw w15..w16;
draw w19..w17; draw w110..w17;

"Paragraph mark";
call charbegin(173, 11, 0, 0, oh, pd, 0);
cpen; top1y = h; bot1y = -d; y1 = y2 = y3 = y4;
y5 = y6; y1 = y2 = good1 5[y0, y1];
lt1x2 = round u; y2 = .5 y0, y1; x1 = x2 = .5r; x3 = x4 = x5 = good1 5(r + 2u);
x7 = x8 = x9 + 2u; x10 = x11 = u;
w0 ddraw 4..1{-1, 0}..2{0, -1} 3{1, 0}..5..4..4..5..5;
draw 6..4..1..3;
draw 7..8;

```



```

"at sign";
call charbegin(174, 14, 0, 0, ph, 0, 0);
hpen; x1 = x3 = x10 = .5r; x2 = good1 4u; x3 = x1 = x1 = r - x2;
lft x2 = round u; x7 = x1 = x12 = r - x2; x6 = .5 2 [x2, x7];
top y8 = h + oo; bot y10 = -oo; y1 = y10;
y1 = good1 8[y10, y8]; y2 = y1 = y8 = good1 2[y10, y8];
y7 = .5[y10, y8]; y11 = good1 1[y10, y8]; y12 = .9[y10, y8];
y1 = 8[y2, y1]; y5 = .3[y7, y1];
call "a dare(1, 2, w1);
call "b dare(1, 3, w1);
draw [w1]4..[w1]5{0, -1}; [w1]6{1, 0}..7{0, 1};
call "c arc(8, 7, w1);
call "d dare(8, 9, w1);
w1 draw 10{1, 0}..11{. 12).

```

% left part of inner bowl
 % right part of inner bowl
 % stem and link
 % right part of outer bowl
 % left part of outer bowl
 % point

```

"Copyright symbol";
call charbegin(175, 18, 0, 0, ph, 0, 0);
new up; up = .5{(h - m) - d);
cpen; r1, x1 = round(r - 6u);
if top1(top1top0e + 2) > .9[c, m]; top1 y1 = 9[c, m] + up;
else: y1 = top1top0e + 2 + up;
fi;
w1 draw 1,

```

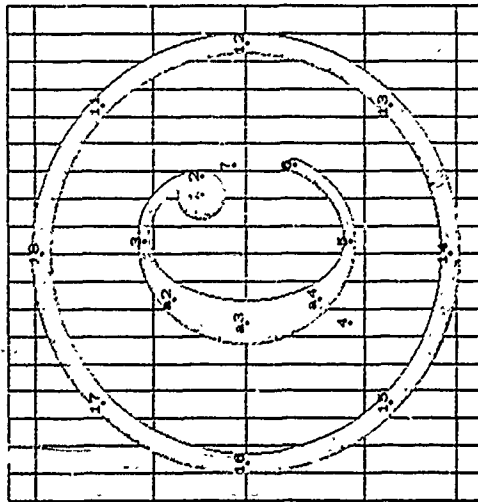
% bulb
 % shoulder

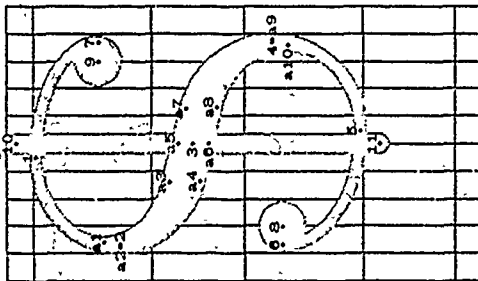
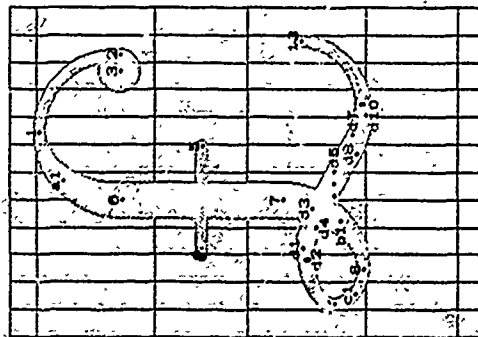
```

hpen; r1, x2 = r1, x1; y2 = y1; x1 = x3 = r1 + u; top0 y1 = m + oo + up;
w1 draw 2{0, 1}..3{-1, 0};
if w2 > 1.5u; lft x1 = round(5.75u);
else: x1 = good1 6.5u;
fi;
y1 = y7; bot y7 = -oo + up;
call "a dare(3, 4, w1);
if w1 = w1; x0 = x2; x7 = x1 = x3 = x1; y1 = .5[y2, y7];
new aa; x0 = aa[x2, x7]; y0 = (sqrt(1 - aa aa)) [y7, y7];
else lft x0 = r1, x2; y0 = .5e - 1 + up; x1 = x0; y1 = e + up;
fi;
w1 draw 5{1, 0}..6{. 7);
cpen; top1 y1 = h + oo; bot1 y1 = -d - oo;
lft x1 = round u; r1, x12 = round(r - u);
call circle(11, 12, 13, 14, 15, 16, 17, 18, w10).

```

% stroke
 % point
 % enclosing circle





```

"sterling sign";
call charbegin(176, 12, 0, 0, ph, 0, 0);
hpen; x1 = 7.5u; topy1 = h + oo; r1x2 = r1x1 = round(r - 1.5u);
y2 = y1 = .75h; y1 = y1 = .5h; x1 = 3u - cps; x2 = 7u + cps;
x3 = x1 = good, 5u; y3 = .75[y1, y1]; y1 = .25[y1, y1]; boty3 = -oo;
x3 = 2.5u; lftx3 = round u; y1 = .1h;
w1 draw 1{1, 0}..2{0, -1};
cpen; w2 draw 3;
call "a arc(1, 0, w1);
hpen; w1 draw 6..7;
w1 draw 4..5;
call "b arc(8, 7, w1); call "c arc(8, 9, w1);
topy1y0 = round 2h; boty1y2 = -oo;
x1 = .5[x1, x1] - u; y1 = y1;
r1x2 = round(r - u); y1 = 2h;
call "d zdraw(9, 11, 12, 13, w1, w1, -(x1 - x3 - 4u)/(2h)).

"Dollar sign";
call charbegin(177, 10, 0, 0, ph + ph, ph, ph, slant - 5pu);
hpen; topy1h = h + oo; boty25 = -oo;
x1 = good, 10.5r; y1 = .52h; lftx2 = round u; x1 = r - x1;
if ucs = 0; x1 = x1 = x1; y1 = .5[y1, y1]; y1 = .5[y1, y1];
else: if w3 = w3; x1 = x1 = x1; y1 = .5[y1, y1]; y1 = 5[y1, y1];
else: x1 = .5u = x1 - .5u = x1; y1 = h/4 - 1; y1 = 8u + 1;
fi;
y1 = y1; y1 = y1; cpen; lftx3 = lftx2; r1x1 = r1x2; x1 = x1; x1 = x1;
w1 draw 8;
draw 9;
hpen; w1 draw 6{0, -1}..5{1, 0};
draw "0, 1}..1{-1, 0};
call "a sdraw(1, 2, 3, 4, 5, w1, w1, -h/(50u));
cpen; x10 = x1 = x1; topy1y0 = h + 0; boty1y1 = -b;
w10 draw 10..11.

```

% link
 % bulb
 % shoulder
 % stem
 % bar
 % loop

 % arm

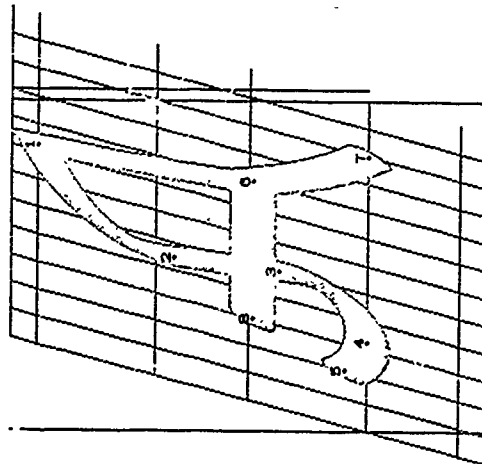
 % lower bulb
 % upper bulb
 % lower left stroke
 % upper right stroke
 % middle stroke
 % stem

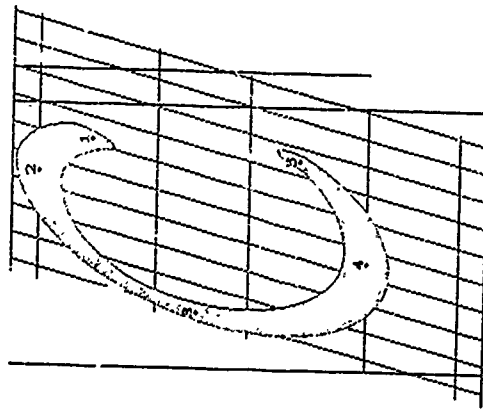
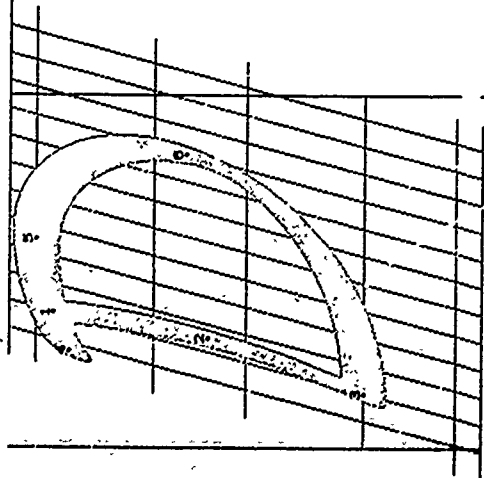
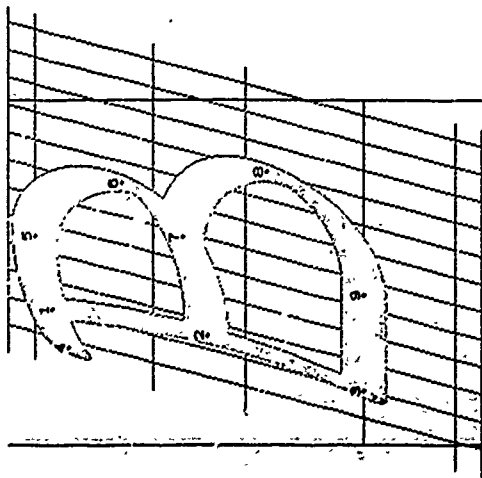
```
i:put symnext; % possible characters '067, '077, '156, '157, '167
```

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.
% variables to specify the spen
new aa, bb, aa, bbb, spa, spb, spc;
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);
no penreset;
new cor, corr;
cor = ph-slant/pw;
if cor > 1: corr = 1;
else: corr = cor;
fi;

"Script A"; spen;
call charbegin("A, 12, 0, 0, ph, 0, ph-slant - (3 75 - m)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{x4 - x3, y4 - y3}..
4{-1, 0}..5{x5 - x4, y5 - y4}..6{x6 - x5, y6 - y5}..7{1.5(x7 - x6), y7 - y6};
draw 1{0, 1}..6{x7 - x6, 1.5(y7 - y6)}..7{1.5(x7 - x6), y7 - y6};
draw 8..6. % left diagonal
% right diagonal
% bar
```





```

"Script B"; spen;
call charbegin("B, 12, mi-corr, -.5mi-corr, 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, x0; y7 = (sqrt.5)|y2, y0;
x8 = 10u; y8 = .3h;
x9 = 6.5u; y9 = 0;
draw 1{2(x2 - x1), y2 - y1}..2{0, -1}..3{2(x1 - x2), y1 - y2};
draw 4{x2 - x1, 3(y2 - y1)}..5{1, 0}..6{0, -1}..
7{x2 - x0, y2 - y0}..8{0, -1}..9{-1, 0}..3.
draw 1{1, 0}..8{0, -1}..9{-1, 0}..3.

"Script C"; spen;
call charbegin("C, 10 5, 5mi-corr, -.2mi-corr, ph, 0,
mi|.5ph 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{x3 - x1, 4(y3 - y1)}.

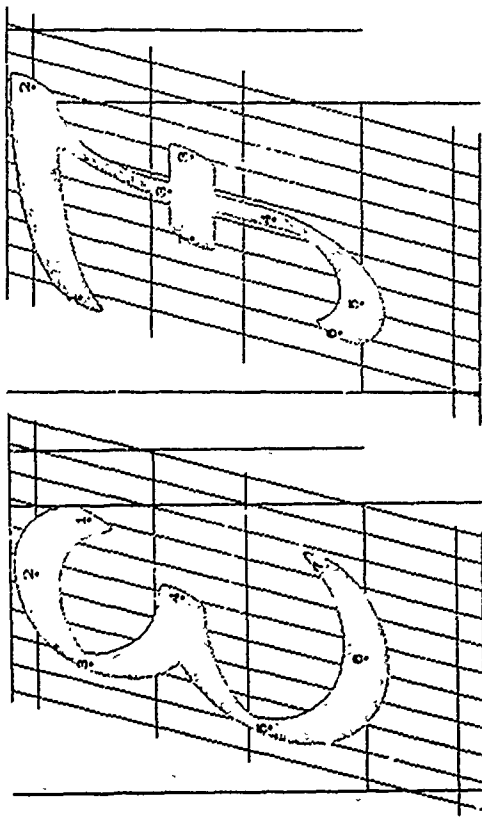
"Script D"; spen;
call charbegin("D, 12, mi-corr, -.6mi-corr, ph, 0, mi|.6ph slant - pu, 0));
x1 = 3u; y1 = .95h;
x2 = 3.5u; y2 = .48h;
x3 = 3u; y3 = 0;
x4 = 1.75u; y4 = .9h;
x5 = 5.5u; y5 = h;
x6 = 10u; y6 = .55h;
draw 1{2(x2 - x1), y2 - y1}..2{0, -1}..3{2(x1 - x2), y1 - y2};
draw 4{x2 - x1, 3(y2 -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

```

% stem
% upper bowl
% lower bowl

% bowl

% stem
% bowl

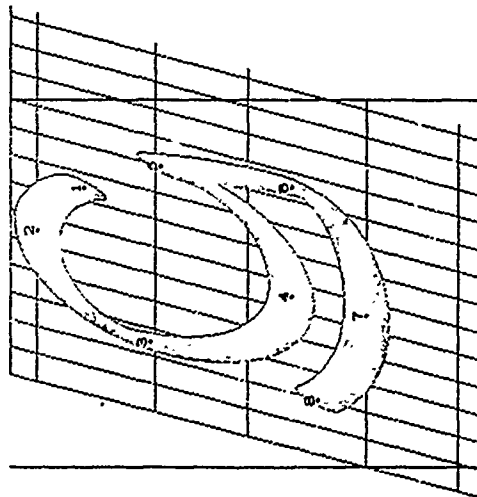


```

"Script E"; spen;
call charbegin("E, 10.5, .2mi-corr, -.2mi-corr, 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};
draw 4{-1, 0}..5{0, -1}..6{1, 0}..7{x7 - x6, 3{y7 - y6}};
% upper bowl
% 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};
draw 2{-1, 0}..3{0, -1}..4{0, -1}..5{-1, 0}..6{x6 - x5, 3{y6 - y5}};
draw 7..8;
% shoulder
% stem
% bar

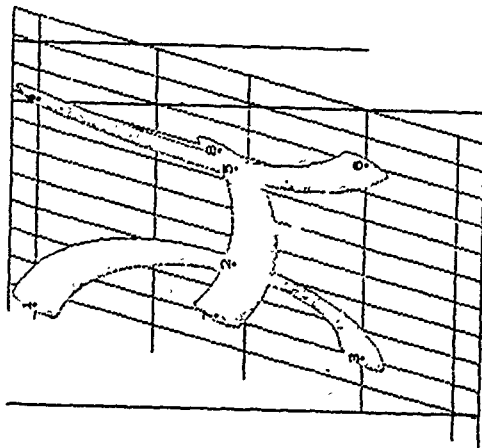
```



```

"Script G"; spen;
call charbegin("G, 11.5, 0, mi(-.75cor + .5cor), 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 - y1}}..2{-1, 0}..3{0, -1}..4{1, 0}..5{0, 1};
draw 5{2{x6 - x5, y6 - y5}}..6{0, -1}..7{-1, 0}..8{x8 - x7, 3{y8 - y7}};
% upper bowl
% 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{(x2 - x1), y2 - y1}..2{(0, -1)}..3{-1, 0};
draw 4{(x3 - x1), y3 - y1}..5{(0, -1)}..6{(x6 - x1), y6 - y1};
draw 7{(x4 - x1), y4 - y1}..8{(x8 - x1), y8 - .15h};
```

% left stem
% right stem
% bar

```
"Script I"; spen;
call charbegin("I, 9, 0, 0, pt, 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{(x2 - x1), y2 - y1}..2{(0, -1)}..3{(x3 - x1), y3 - y1};
draw 4{(x1 - x1), y1 - y1}..5{(x5 - x1), y5 - y1};
draw 6{(x3 - x1), y3 - y1}..7{(x7 - x1), y7 - y1};
```

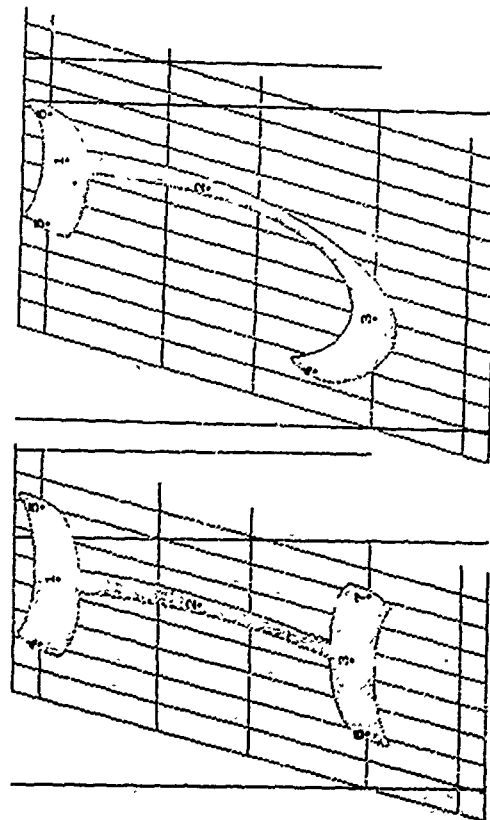
% stem
% upper serif
% lower serif

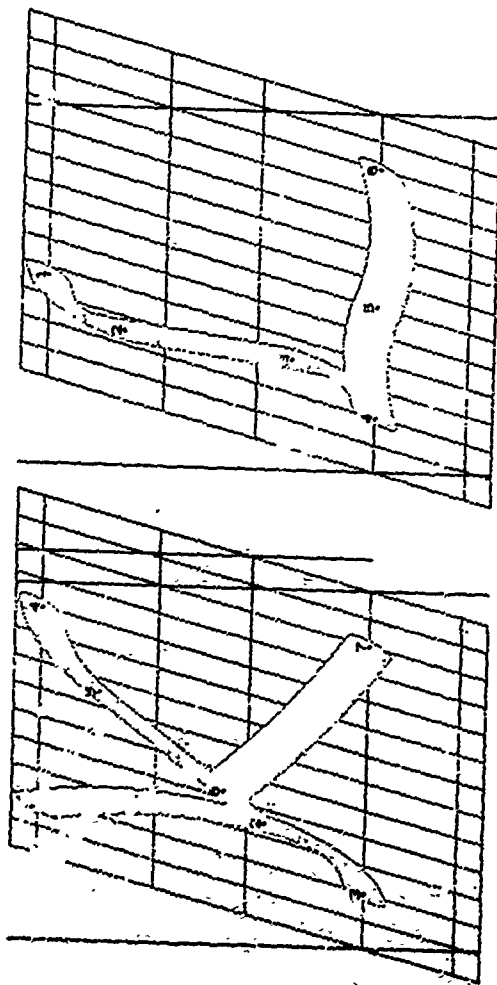
```
"Script J"; spen;
call charbegin("J, 10, 0, -5 mi-corr, ph, 0, mi|ph-slant - pu, 5ph-slant));
```

```
x1 = 6.5u; y1 = .95h;
x2 = 7u; y2 = .52h;
x3 = 4u; y3 = 0;
x4 = 1.5u; y4 = .18h;
x5 = 4u; y5 = h;
x6 = 8u; y6 = h;
```

```
draw 1{(x2 - x1), y2 - y1}..2{(0, -1)}..
3{-1, 0}..4{(x1 - x1), y1 - y1};
draw 5{(x1 - x1), y1 - y1}..6{(x6 - x1), y6 - y1};
```

% stem and tail
% serif





```
"Script K"; spen;
call charbegin("K", 13, 0, 0, ph, 0, ph, shant - (3 - ml)pu);
x1 = 1.75u; y1 = h;
x2 = 3.5u; y2 = .31h;
x3 = 2u; y3 = 0;
x4 = 9u; y4 = h;
x5 = 5.5u; y5 = .82h;
x6 = 4.25u, new aa; x0 = aa[x2, x1]; y0 = aa[y4, y5];
x7 = 11u, y7 = 0;
draw 1{(x2 - x1), (y2 - y1) .. 2{0, -1} .. 3{-1, 0};
draw 4{-1, 0} .. 5{(x2 - x0), (y2 - y0) .. 2;
draw 6{(x1 - x0), 1.25(y7 - y0)} .. 7{1.25(x7 - x0), y7 - y0}.
```

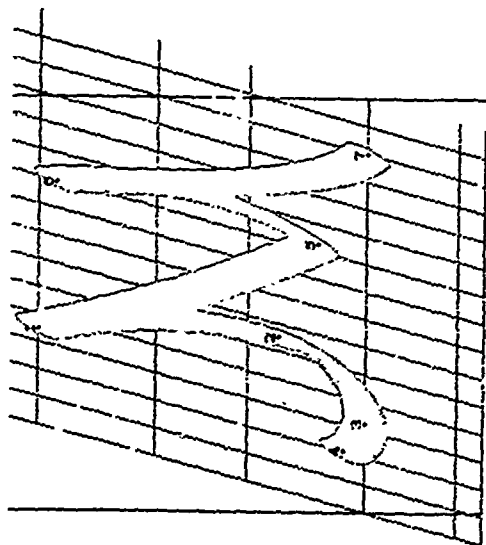
% stem
% upper diagonal
% lower diagonal

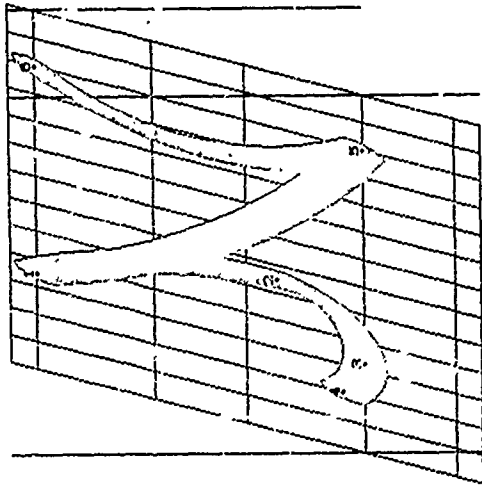
```
"Script L"; spen;
call charbegin("L", 13, 0, 0, ph, 0, 0);
x1 = 3.75u; y1 = h;
x2 = 2.5u; y2 = .77h;
x3 = 3.25u; y3 = .26h;
x4 = 2u; y4 = 0;
x5 = 6u; y5 = 0;
x6 = 1u; y6 = 0;
draw 1{-1, 0} .. 2{(x3 - x2), (y3 - y2) .. 3{(x1 - x2), (y1 - y2) .. 4{-60u, -h};
draw 4{30u, h} .. 5{30u, -h} .. 6{30u, h}.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

% stem
% 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{(x2 - x1), (y2 - y1) .. 2{0, -1} ..
3{-1, 0} .. 4{(x3 - x2), (y3 - y2) .. 5{(x1 - x2), (y1 - y2)};
draw 1{(x5 - x1), 1.25(y7 - y1) .. 5{1.25(x5 - x1), y7 - y1};
draw 5{4u, h} .. 6{-4u, h};
draw 6{(x1 - x0), 1.5(y7 - y0) .. 7{3(x7 - x0), y7 - y0}.
```

% first diagonal
% second diagonal
% third diagonal
% fourth diagonal





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

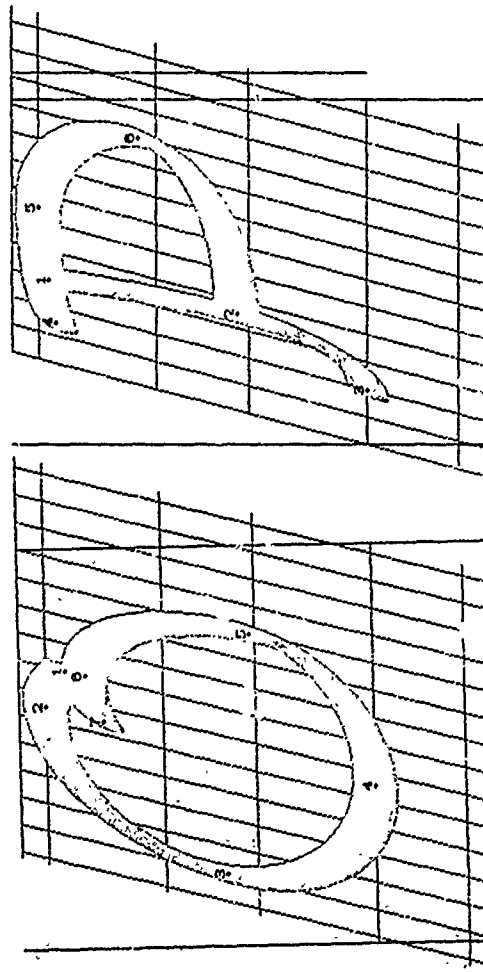
% left stem
% diagonal
% right stem

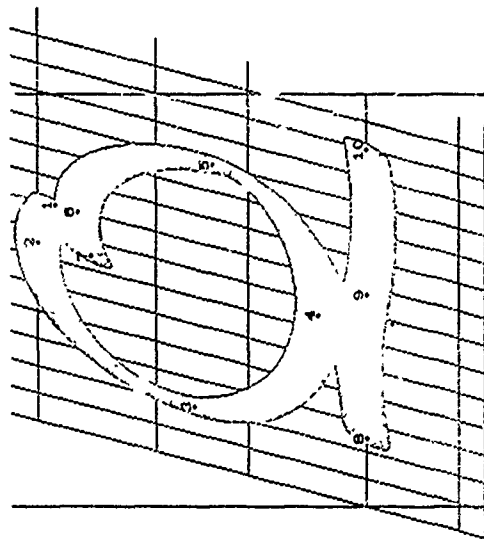
```
"Script O"; spen;
call charbegin("O", 13, .25mi-cor, -.75mi-cor, ph, 0, mi| 75ph slant -- pu, 0));
x1 = 8u; y1 = .94h;
x2 = 6.5u; y2 = h;
x3 = 2u; y3 = .46h;
x4 = 6.5u; y4 = 0;
x5 = 11u; y5 = .38h;
x6 = 8u; y6 = .88h;
x7 = 6.5u; y7 = .83h;
draw 1{x2 - x1, 2(y2 - y1)} .. 2{-1, 0} .. 3{0, -1} .. 4{1, 0} .. 5{0, 1} ..
6{-1, 0} .. 7{x1 - x6, 4(y1 - 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, -1} .. 3{-1, 0};
draw 4{x3 - x4, 4(y3 - y4)} .. 5{1, 0} .. 6{0, -1} .. 7{-1, 0}.
```

% stem
% bowl





```

Script Q"; spen;
call charbegin("Q,15,0,0,ph,0,0);
x1 = 8u; y1 = .95h;
x2 = 6.5u; y2 = h;
x3 = 2u; y3 = .53h;
x4 = 6.5u; y4 = .15h;
x5 = 11u; y5 = .47h;
x6 = 8u; y6 = .88h;
x7 = 6.5u; y7 = .84h;
x8 = 2.5u; y8 = 0;
x9 = 7.75u; y9 = 0;
x10 = 13u; y10 = 0;
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}};
draw 5{0,-1}..8{-120u,-h};
draw 8{60u,h}..9{60u,-h}..10{60u,h}.

```

% low
% link
% bar

```

Script R"; spen;
call charbegin("R,14,0,0,ph,0,0);
x1 = 3u; y1 = .97h;
x2 = 3.5u; y2 = .4h;
x3 = 2u; y3 = 0;
x4 = 1.5u; y4 = .95h;
x5 = 5.5u; y5 = h;
x6 = 5u; y6 = .7h;
x7 = 4.25u; y7 = .4h;
x8 = .06h;
x9 = 10.5u; y9 = 0;
(x1--x8)/(y3--y8) = 2(x6--x7)/(y4--y7);
x10 = 12u; y10 = .08h;
draw 1{2{x2--x1},y2--y1}..2{0,-1}..3{-1,0};
draw 4{x3--x1,4{y3--y1}}..5{1,0}..6{0,-1}..7{-1,0};
draw 7..8{x8--x7,y8--y7}..9{1,0}..10{x10--x9,3{y10--y9}}.

```

% x3 w1. be defined shortly

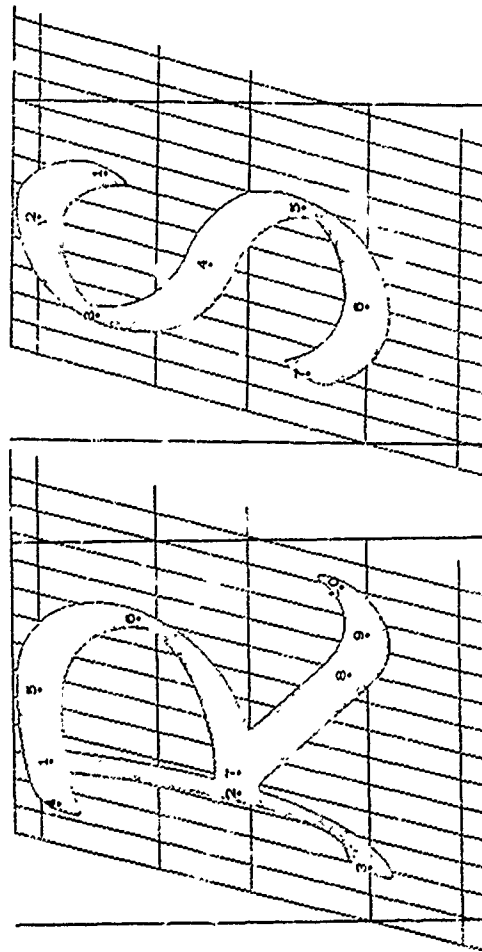
% stem
% bowl
% diagonal

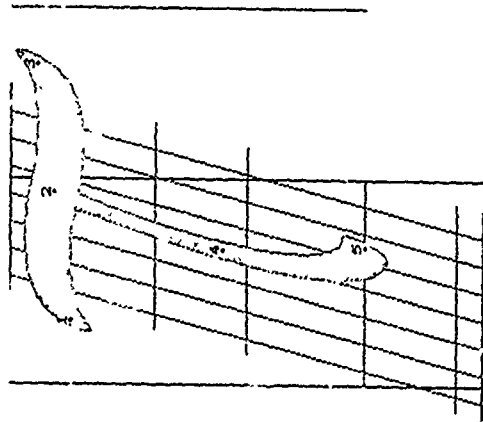
```

Script S"; spen;
call charbegin("S,10,0,-m(cor - 75corr),ph,0,m[ph slant - 175pu,0]);
x1 = 7.25u; y1 = .8h;
x2 = 5u; y2 = h;
x3 = 2u; y3 = .82h;
x4 = 5u; y4 = .48h;
x5 = 8u; y5 = .2h;
x6 = 5u; y6 = 0;
x7 = 2u; y7 = .18h;
draw 1{x2--x1,4{y2--y1}}..2{-1,0}..3{0,-1}..4{32u,h}..5{0,-1}..
6{-1,0}..7{x7--x6,3{y7--y6}}.

```

% stroke



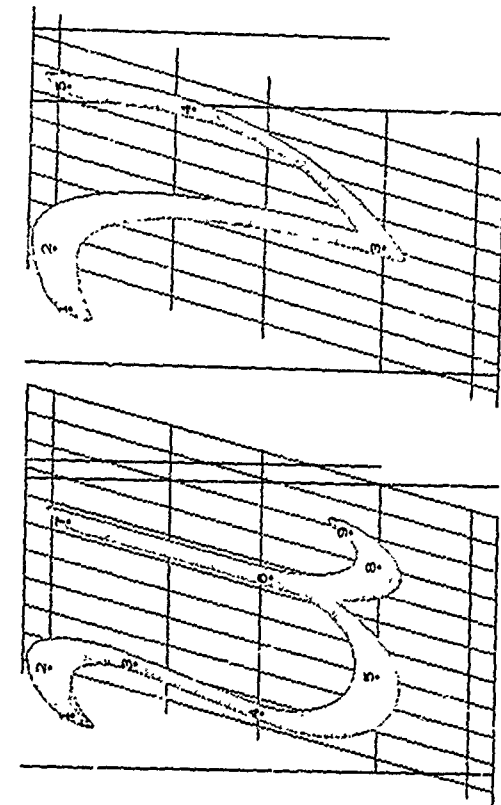


```
"Script T"; spen;
call charbegin("T", 13, 8mi-cor, 3mi, ph, 0, ph slant + (4mi - 1)pu);
x1 = 2u; y1 = .9h;
x2 = 6.5u; y2 = .95h;
x3 = 11u; y3 = h;
x4 = 6u; y4 = .44h;
x5 = 7.5u; y5 = 0;
draw 1{(6u, h) .. 2{(60u, -h) .. 3{(6u, h)};
draw 2{(x1 - x2), y1 - y2} .. 4{(0, -1)} .. 5{(1, 0)}.
```

% bar
% stem

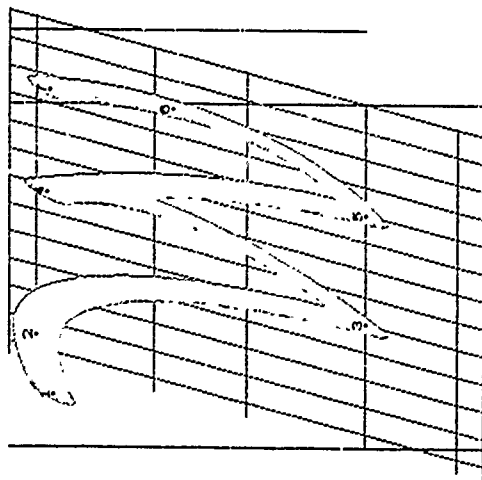
```
"Script U"; spen;
call charbegin("U", 13, 8mi-cor, 0, ph, 0, ph slant - (3.5 - mi)pu);
x1 = 1.5u; y1 = .93h;
x2 = 3u; y2 = h;
x3 = 4u; y3 = .74h;
x4 = 3.5u; y4 = .35h;
x5 = 6u; y5 = 0;
x6 = 8.5u; y6 = .33h;
x7 = 8.5u; y7 = .95h;
x8 = 10u; y8 = 0;
x9 = 11u; y9 = .09h;
draw 1{(x2 - x1, 2(y2 - y1)) .. 2{(1, 0)} .. 3{(x1 - x1, y1 - y1)} .. 4{(x1 - x1, y1 - y1)} .. 5{(1, 0)} .. 6{(0, 1)} .. 7;
draw 6{(0, -1)} .. 8{(1, 0)} .. 9{(x9 - x8, 4(y9 - y8))}.
```

% stroke and stem
% point



```
"Script V"; spen;
call charbegin("V", 12, 8mi-cor, 0, ph, 0, ph slant - (1.5 - m)pu);
x1 = 1.5u; y1 = .95h;
x2 = 3.5u; 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)};
draw 3{(x1 - x1, y1 - y1)} .. 4{(0, 1)} .. 5{(2(x2 - x1), y5 - y1)}.
```

% left diagonal
% right diagonal

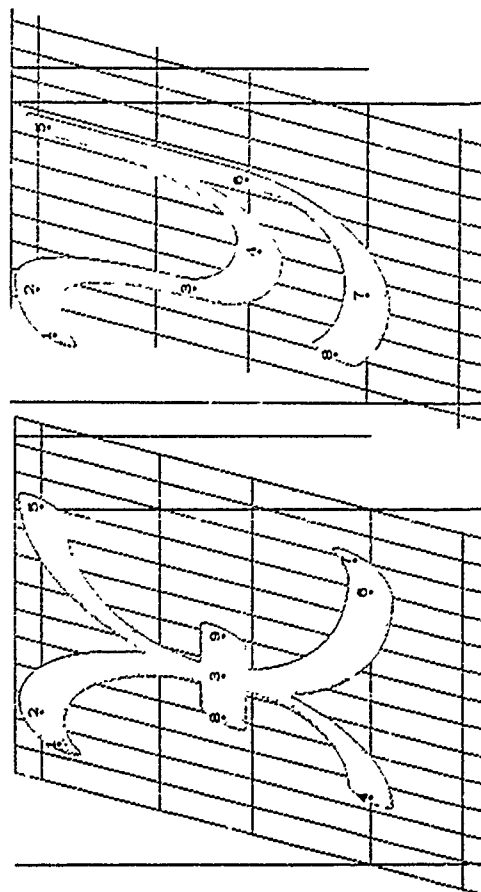


```
"Script W"; spen;
call charbegin("W", 15, 8mi-cor, 0, ph, 0, oh slant -- (1.5 - mu)pu);
x1 = 1.5u; y1 = .95h;
x2 = 3.5u; y2 = h;
x3 = 7u; y3 = 0;
x4 = 8.75u; y4 = .97h;
x5 = 11u; y5 = 0;
x6 = 13u; y6 = .59h;
x7 = 12.5u; y7 = .96h;
draw 1{x2 - x1, 2{y2 - y1}} 2{1, 0} .. 3{0, -1};
draw 3{x7 - x1, y7 - y1} 4{2{x1 - x2}, y1 - y2};
draw 4{2{x3 - x1}, y3 - y1} 5{0, -1};
draw 5{3{x6 - x2}, y6 - y2} 6{0, 1} .. 7{2{x7 - x6}, y7 - y6};
```

% first diagonal
% second diagonal
% third diagonal
% fourth diagonal

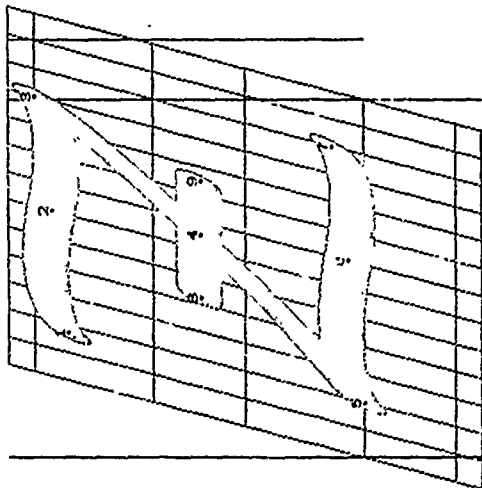
```
"Script X"; spen;
call charbegin("X", 13, 0, 0, ph, 0, ph slant -- (2 - 1.5mu)pu);
x1 = 1.5u; y1 = .95h;
x2 = 2.5u; y2 = h;
x3 = 5.5u; y3 = .45h;
x4 = 2.5u; y4 = 0;
x5 = 10u; y5 = h;
x6 = 10u; y6 = 0;
x7 = 11u; y7 = .05h;
x8 = 4u; y8 = .45h;
x9 = 7u; y9 = .45h;
draw 1{x2 - x1, 3{y2 - y1}} .. 2{1, 0} .. 3{0, -1} .. 4{-1, 0};
draw 5{-1, 0} .. 3{0, -1} .. 6{1, 0} 7{x7 - x6, 3{y7 - y6}};
draw 8 .. 9.
```

% left stroke
% right stroke
% 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 = 7u; 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};
draw 5 .. 6{0, -1} .. 7{-1, 0} .. 8{x8 - x7, 3{y8 - y7}}.
```

% stroke
% stem and tail



```

Script Z"; spen;
call charbegin(~Z, 13, 0, 0, ph, 0, ph.slant - (2 - mt)pt);
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 = 4.25u; y8 = .5h;
x9 = 8.5u; y9 = .5h;
draw 1 {6u, h}..2{60u, -h}..3{6u, h},
draw 3{x1 - x2, 2{y1 - y2}}..4{x3 - x4, y3 - y4}..5{x5 - x6, 2{y5 - y6}};
draw 5{6u, h}..6{60u, -h}..7{6u, h},
draw 8..9.
% upper bar
% diagonal
% lower bar
% 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;
pt = typesize/10;
% one virtual point
% assumes 10 point specifications

% The following subroutines break up the large characters on an Alphanumeric CRIS,
% assuming that 10pt equals 10 points.
subroutine eighteen:
if mode = 2: crsbreak -9pt cf;
fi.

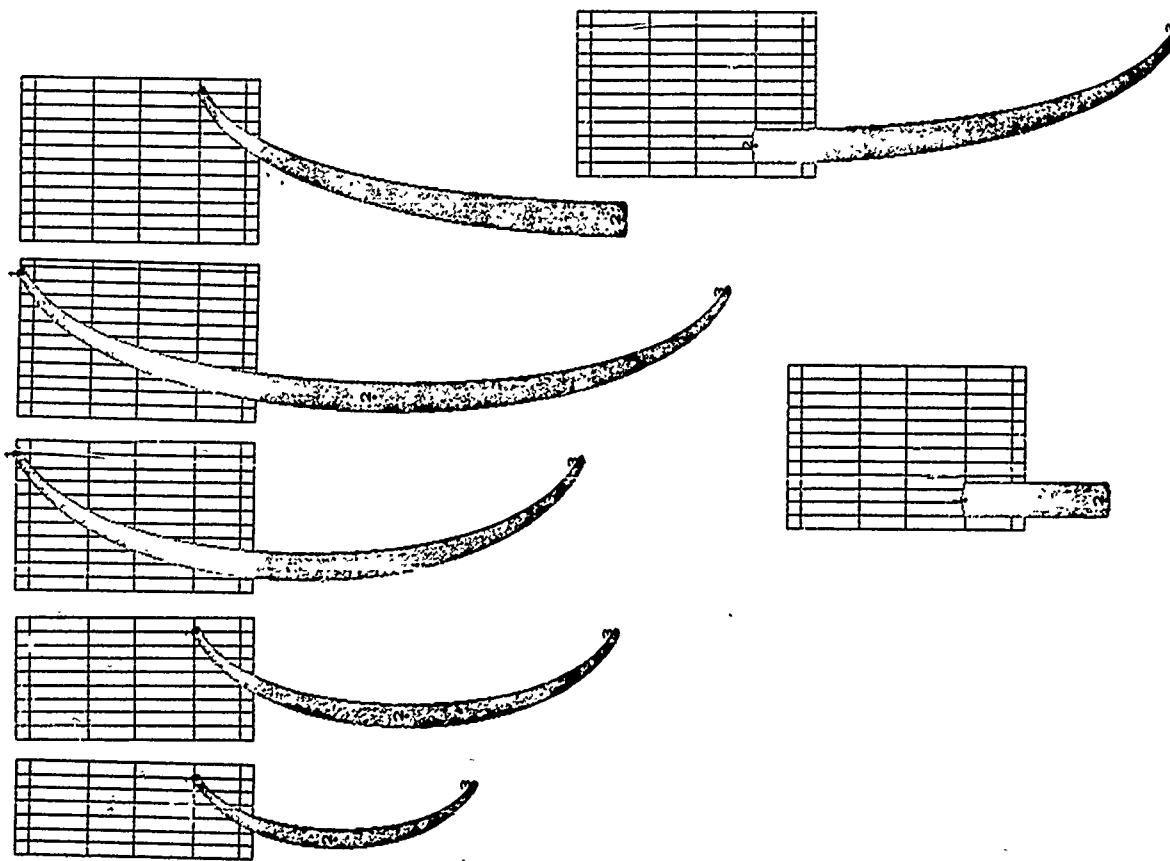
subroutine twentyfour:
if mode = 2: crsbreak (ph + pb - 12pt) cf;
fi.

subroutine thirty:
if mode = 2: crsbreak (ph + pb - 12pt) cf; crsbreak (ph + pb - 13pt) cf;
fi.

input mathd1;
input mathop;
input mexext;
% the large delimiters
% the large operators
% nonstandard characters (any of the codes
% '017, '037, '055, '057, '077, '104, '105, '140, '157, '167, '176, '177)
texinfo slant, 6pu, 3pu, 2pu, px, 18pu, 2pu, prt;
% (The calling file should supply the remaining texinfo.)
```

The file `mathd1.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, height, depth, 0);
new w18, w19; w18 = round minps; w19 = round maxps;
hpen; x1 = x3 = good (x - u);
top18y1 = round pixels.height, bot18y3 = 1 - round pixels.depth; y2 = 5[y1, w3];
lft18x2 = round u;
y0 = y1; y1 = y6; x = x1 + 1.875(units - 2)u;
draw (0..)[w18]1..[w18]2(0, -1) ..[w18]3( . 4).
% stroke
```



```

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

"18 point left parenthesis";
call biglp('020, 9, w0, bold, 0, 18pt); call eighteen;

"24 point left parenthesis";
call biglp('022, 11, w0 + .2deltaw, bold + deltaw, ph + pb, 24pt - ph - pb);
call twentyfour;

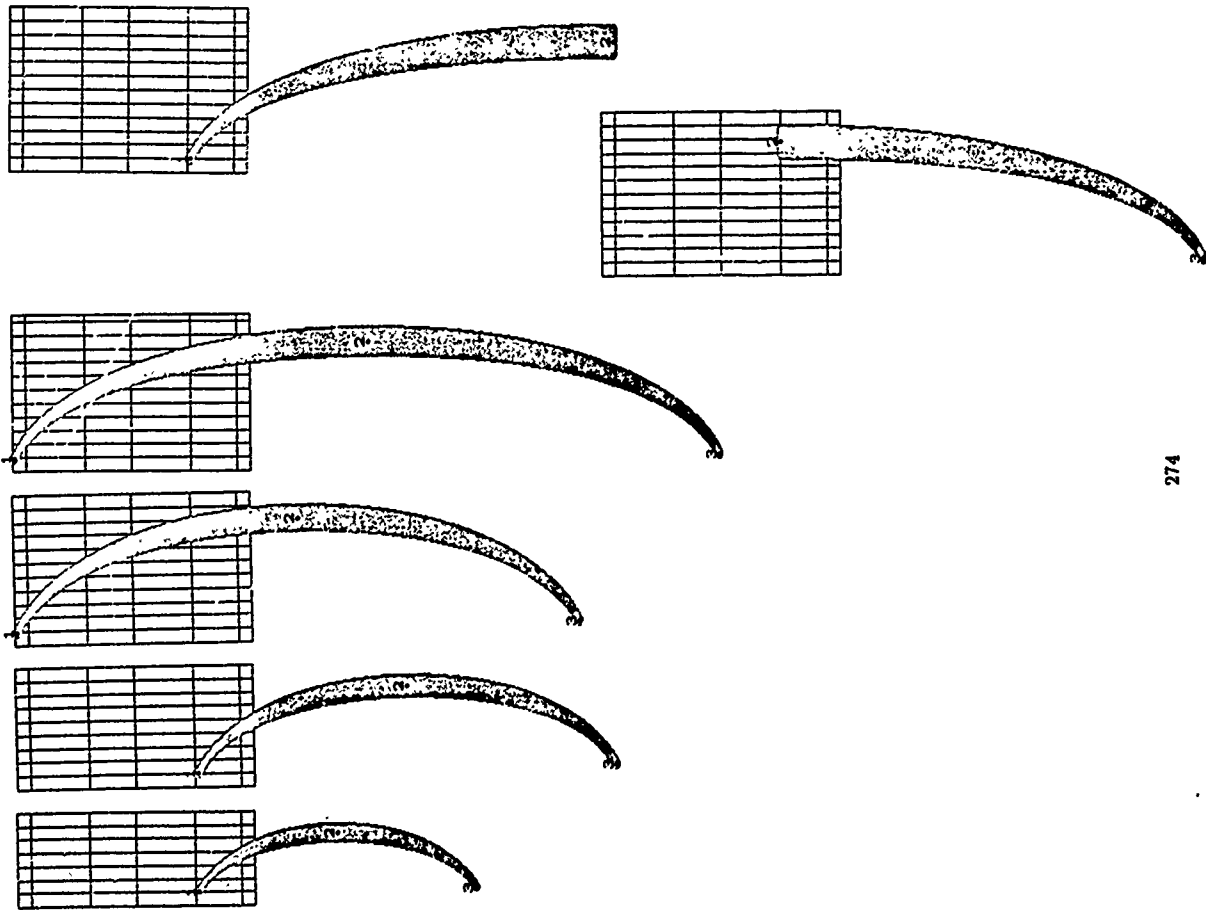
"30 point left parenthesis";
call biglp('040, 11.5, w0 + .4deltaw, bold + 2deltaw, ph + pb, 30pt - ph - pb);
call thirty;

subroutine biglp: % free up METAFONT's memory
new wys, wyg; wys = round(w0 + .5deltaw); wyg = 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 = goodys(r - u); topysy1 = 0,
lftysx2 = round u; y2 = round(.5 - 15pt-pixels);
x0 = x1 + 1.875(10u); y0 = y1;
draw (0..)[wys]1..[wys]2(0, -1);
% upper part of stroke

"Extensible left parenthesis-bottom";
call charbegin('100, 12, 0, 0, 0, 18pt, 0);
hpen; x1 = goodys(r - u); botysy1 = round(.5 - 18pt-pixels);
lftysx2 = round u; y2 = 0;
x1 = x1 + 1.875(10u); y1 = y2;
draw [wys]2(0, -1)..[wys]3(.4);
% lower part of stroke

"Extensible left parenthesis extension module";
call charbegin('102, 12, 0, 0, 0, 6pt, 0);
varchar 0, 0, '102;
% left parenthesis extension modules only
hpen; lftysx1 = round u; y1 = 0;
x2 = x1; y2 = round(.5 - 6pt-pixels);
wys draw 1..2;
% middle part of stroke

```



```

% Right parentheses (left-right symmetric with left ones)
charlist '001, '021, '023, '041, '061, 0;
subroutine bigrp(var code, var units, var niups, var maxps, var height, var depth);
call charbegin(code, units, 0, 0, height, depth, 0);
new wps, wps; wps = round niups; wps = round maxps;
hpen; x1 = x1, r - x1 = good, (r - u);
top, isy1 = round pixels-height; bot, isy3 = 1 - round pixels-depth; y2 = .5[y1, y3];
lft, isy2(r - x2) = round u;
ys = y1; y1 = y3; x0 = x1 - 1.875(units - 2)u;
draw (0..)wps[1.. wps]2{0, -1}..wps[3](.4).
% stroke

"12 point right parenthesis";
call bigrp('001, 7, w, w, 0, 12pt).

"18 point right parenthesis";
call bigrp('021, 9, w, w, bold, 0, 18pt); call eighteen.

"24 point right parenthesis";
call bigrp('023, 11, w, w + 2deltaw, bold + deltaw, ph + pb, 24pt - ph - pb);
call twentyfour.

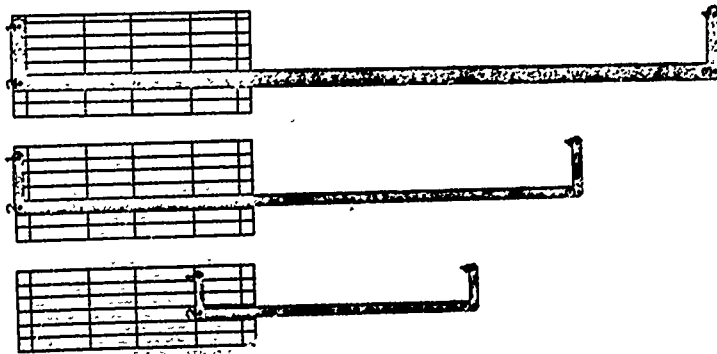
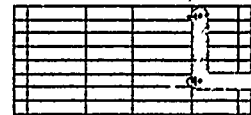
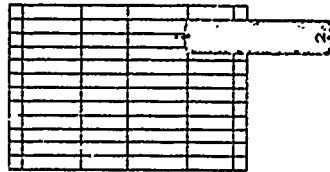
"30 point right parenthesis";
call bigrp('041, 11.5, w, w + 4deltaw, bold + 2deltaw, ph + pb, 30pt - ph - pb);
call thirty.

subroutine bigrp;
% free up METAFONT's memory

new wps, wps; wps = round(w, w + 3deltaw); wps = round(bold + 3deltaw);
"Extensible right parenthesis-top";
call charbegin('061, 12, 0, 0, 0, 18pt, 0);
varchar '061, 0, '101, '103;
hpen; r - x1 = good, (r - u); top, isy1 = 0;
lft, isy2(r - x2) = round u; y2 = round(.5 - 18pt pixels);
x0 = x1 - 1.875(10u); y0 = y1;
draw (0..)wps[1.. wps]2{0, -1}.
% upper part of stroke

"Extensible right parenthesis-bottom";
call charbegin('101, 12, 0, 0, 0, 18pt, 0);
hpen; r - x1 = good, (r - u); bot, isy3 = round(.5 - 18pt pixels);
lft, isy2(r - x2) = round u; y2 = 0;
x1 = x3 - 1.875(10u); y1 = y3;
draw wps[2]{0, -1}..wps[3](.4).
% lower part of stroke

```



```

"Extensible right parenthesis-extension module";
call charbegin('103,12,0,0,0,6pt,0);
varchar 0, 0, 0, '133;
% right parenthesis extension modules only
hpen; ltop(r - x1) = round u; y1 = 0;
x2 = x1; y2 = round(.5 - 6pt-pixels);
% middle part of stroke
w9 draw 1..2.

% Left brackets (left-right symmetric with right ones)
charlist '002, '024, '042, '062, 0;
subroutine biglb(var code, var units, var psze, var height, var depth);
call charbegin(code, units, 0, height, depth, 0);
new w9; w9 = round psze;
open; x1 = x1 = good(r - .75u); x2 = x1 = good w(2.5u);
top w9 = round pixels-height; bot w9 = 1 - round pixels depth; y1 = y2; y1 = y1;
w9 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 w9; w9 = round(w1 - .5deltaw);
"Extensible left bracket top";
call charbegin('062, 8, 0, 0, 0, 18pt, 0);
varchar '062, 0, '064, '066;
open; x1 = good(r - .75u); top w9 = 0;
x2 = good w(2.5u); y2 = y1;
x1 = x2; y1 = round(.5 - 18pt-pixels);
w9 draw 1..2..2..3.

"Extensible left bracket-bottom";
call charbegin('064, 8, 0, 0, 0, 18pt, 0);
varchar 0, 0, '064, '066;
open; x1 = good(r - .75u); bot w9 = round(.5 - 18pt-pixels);
x2 = good w(2.5u); y2 = y1;
x1 = x1; y2 = 0;
w9 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,'086;
open; x1 = goodwp 2.5u; y1 = 0;
x2 = x1; y2 = round(.5 - 6pt-pixels);
wwp draw 1..2.
```

% extensible left ceiling bracket

% 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 wwp; wwp = round psize;
open; x1 = x1; r - x1 = goodwp(r - .75u); x2 = x1; r - x2 = goodwp(2.5u);
top wwp = round pixels-height; bot wwp = 1 - round pixels depth; y1 = y1; y2 = y1;
wwp draw 1..2..2..3..3..3..4.
```

"12 point right bracket";

call bigrb('003,6,w_{wp},0,12pt)

"24 point right bracket";

call bigrb('025,7,w_{wp}-.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 w_{wp}; w_{wp} = round(w1 - .6deltaw);

"Extensible right bracket-top";

call charbegin('063,8,0,0,0,18pt,0);

varchar '063,0,'065,'087;

open; r - x1 = good_{wp}(r - .75u); top w_{wp} = 0;

r - x2 = good_{wp} 2.5u; y2 = y1;

x3 = x2; y3 = round(.5 - 18pt-pixels);

w_{wp} draw 1..2..2..3.

% extensible right bracket

% upper part of stroke

"Extensible right bracket-bottom";

call charbegin('065,8,0,0,0,18pt,0);

varchar 0,0,'065,'087;

open; r - x1 = good_{wp}(r - .75u); bot w_{wp} = round(.5 - 18pt-pixels);

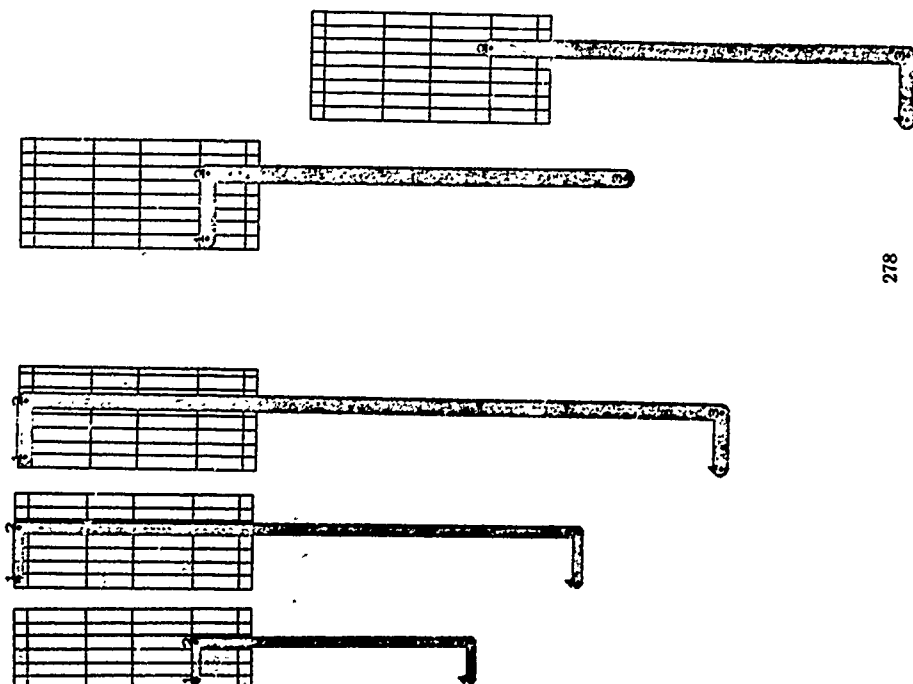
r - x2 = good_{wp} 2.5u; y3 = y1;

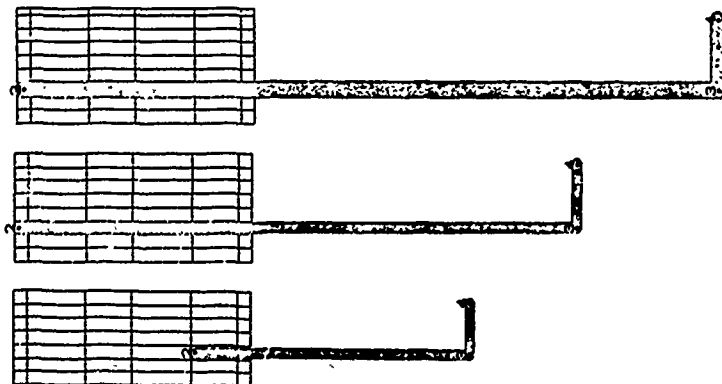
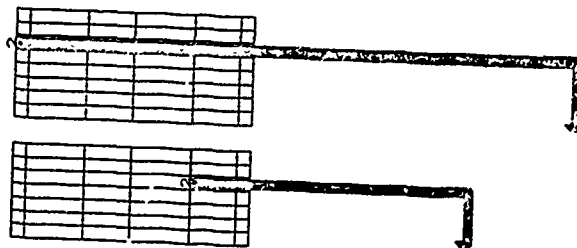
x2 = x2; y2 = 0;

w_{wp} draw 2..3..3..4.

% extensible right floor bracket

% lower part of stroke





```

Extensible right bracket-extension module";
call charbegin('067, 8, 0, 6, 0, 6pt, 0);
varchar '063, 0, 0, '067;
open; r - x1 = good, 2.5u; y1 = 0;
x2 = x1; y2 = round(.5 -- 6pt pixels);
w10 draw 1..2

% middle part of stroke

% Left floor brackets (left-right symmetric with right ones)
charlist '004, '026, '044, '064, 0;
subroutine bigflb(var code, var units, var psiz, var height, var depth);
call charbegin(code, units, 0, 0, height, depth, 0);
new w10; w10 = round psiz;
open; x1 = good, .75u; x2 = x1 = good, 2.5u;
top y2 = round pixels height; bot y3 = 1 - round pixels depth; y1 = y1;
w10 draw 2..3..3..4.

"12 point left floor bracket";
call bigflb('004, 7, w10, 0, 12pt);

"24 point left floor bracket";
call bigflb('026, 8, w10, + 2deltaw, ph + pb, 24pt - ph - pb);
call twentyfour

"30 point left floor bracket";
call bigflb('044, 8.5, w1 - 8deltaw, ph + pb, 30pt - ph - pb);
call thirty.

subroutine bigflb; % free up METAFONT's memory

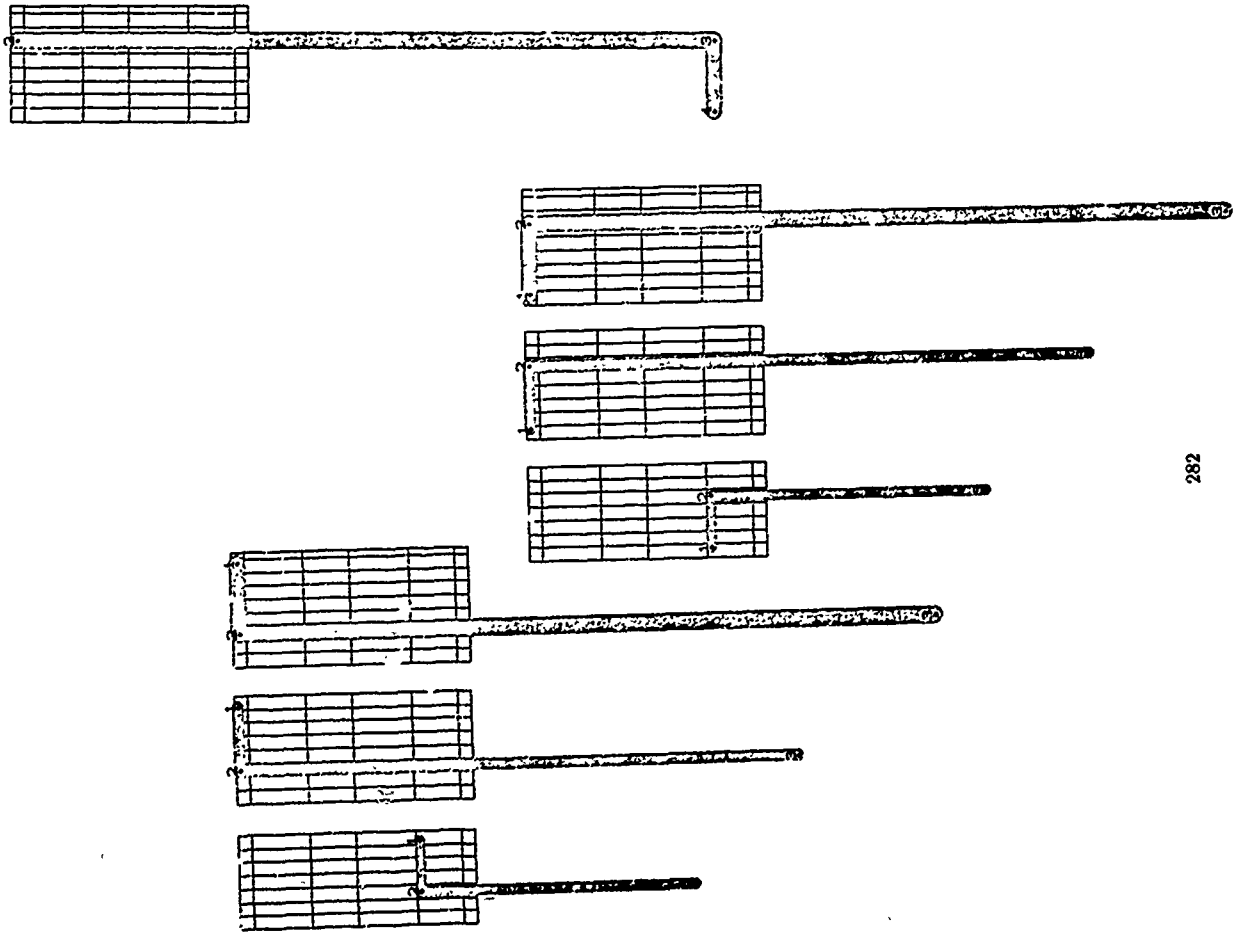
% Right floor brackets (left-right symmetric with left ones)
charlist '005, '027, '045, '065, 0;
subroutine bigflb(var code, var units, var psiz, var height, var depth);
call charbegin(code, units, 0, 0, height, depth, 0);
new w10; w10 = round psiz;
open; r - x1 = good, .75u; x2 = x1; r - x2 = good, 2.5u;
top y2 = round pixels height; bot y3 = 1 - round pixels depth; y1 = y1;
w10 draw 2..3..3..4.

% stroke

"12 point right floor bracket";
call bigflb('005, 7, w10, 0, 12pt);

"24 point right floor bracket";
call bigflb('027, 8, w10 + 2deltaw, ph + pb, 24pt - ph - pb);
call twentyfour.

```



```

"30 point right floor bracket";
call bigrfl('045, 8.5, w1 - .8deltaw, ph + pb, 30pt - ph - pb);
call thirty.

subroutine bigrfl:
% free up METAFONT's memory

% Left ceiling brackets (left-right symmetric with right ones)
charlist '006, '030, '046, '066, 0;
subroutine bigrclb(var code, var psiz, var height, var depth);
call charbegin(code, units, 0, 0, height, depth, 0);
new w0; w0 = round psiz;
open; x1 = good w(r - 75u); x2 = x1 = good w(2.5u);
top ph = round pixels-height; bot w0 = 1 - round pixels-depth; y1 = y2;
w0 draw 1..2 2..3.
% stroke

"12 point left ceiling bracket";
call biglcl('006, 7, w0, 0, 12pt)

"24 point left ceiling bracket";
call biglcl('030, 8, w0 + .2deltaw, ph + pb, 24pt - ph - pb);
call twentyfour.

"30 point left ceiling bracket";
call biglcl('046, 8.5, w1 - .8deltaw, ph + pb, 30pt - ph - pb);
call thirty.

subroutine bigrclb:
% free up METAFONT's memory

% Right ceiling brackets (left-right symmetric with left ones)
charlist '007, '031, '047, '067, 0;
subroutine bigrclb(var code, var units, var psiz, var height, var depth);
call charbegin(code, units, 0, 0, height, depth, 0);
new w0; w0 = round psiz;
open; x1 = x1 = good w(r - 75u); x2 = x1; r - x1 = good w(2.5u);
top ph = round pixels-height; bot w0 = 1 - round pixels-depth; y1 = y2;
w0 draw 1..2 2..3.
% stroke

"12 point right ceiling bracket";
call bigrcl('007, 7, w0, 0, 12pt)

"24 point right ceiling bracket";
call bigrcl('031, 8, w0 + .2deltaw, ph + pb, 24pt - ph - pb);
call twentyfour.

"30 point right ceiling bracket";
call bigrcl('047, 8.5, w1 - .8deltaw, ph + pb, 30pt - ph - pb);
call thirty.

```


% free up METAFONT's memory

subroutine bigreb:

```
% Left braces (left-right symmetric with right ones)
charlist '010, '032, '050, '070, 0,
subroutine biglbr(var code, var units, var minps, var maxps, var height, var depth);
call charbegin(code, units, 0, height, depth, 0);
new wts, wps; wts = round minps; wps = round maxps;
hpen; x2 = x1 = x3 = x6 = x0 = good w(.5r);
x1 - x2 = x2 - x1 = 5(units -- 3)u + eps; x1 = x7;
top y0 = round height pixels; bot y0 = 1 -- round pixels depth;
y1 = 5[y0, y1] = good w(.5r);
y1 - y2 = y2 - y1 = y0 - y1 = (y1 - y0)/4;
draw |wts#1|{3(x2 - x1), y2 - y1} |wps#2|{0, -1} |wps#3|{0, -1};
|wts#4|{3(x1 - x1), y1 - y1};
draw |wts#1|{3(x0 - x1), y0 - y1} |wps#5|{0, 1} |wps#6|{0, 1} .
|wts#7|{3(x1 - x1), y1 - y1};
% upper stem
% lower stem
```

"12 point left brace";

call biglbr('010, 9, w0, w1, 0, 12pt);

"21 point left brace";

call biglbr('032, 11, w0 + 2deltaw, bold + 4deltaw, ph + ph, 24pt -- ph -- pb);

call twentyfour;

"30 point left brace";

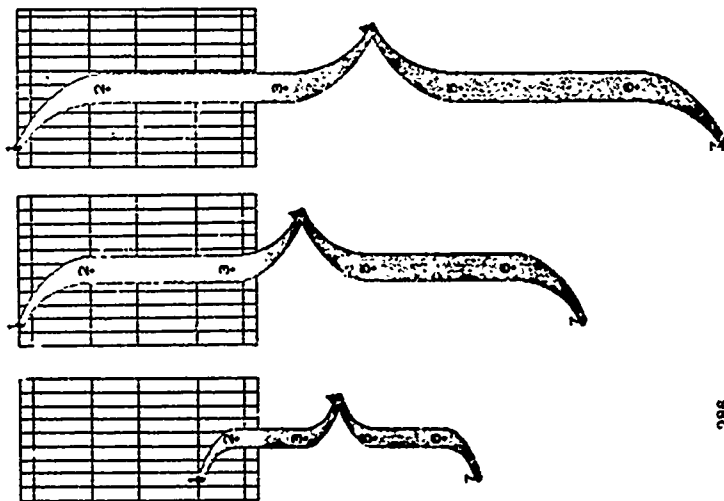
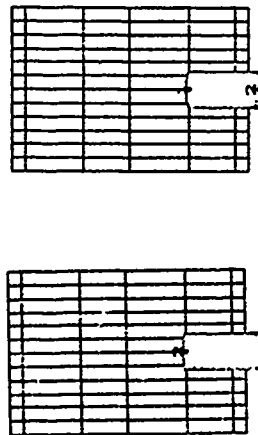
call biglbr('050, 11.5, w0 + 4deltaw, bold + 2deltaw, ph + ph, 30pt -- ph -- pb);

call thirty;

% free up METAFONT's memory

subroutine biglbr:

```
new wts, wps; wts = round(w0 + 6deltaw); wps = round(bold + 4deltaw);
"Extensible left brace top";
call charbegin('070, 12, 0, 0, 9pt, 0);
varchar '070, '074, '072, '076,
hpen; x2 = x1 = good w(.5r); x1 - x2 = 4.5u + eps;
top y0 = 0; y2 = 5[y0, y1]; y1 = round(.5 -- 9pt pixels);
draw |wts#1|{3(x2 - x1), y2 - y1} |wps#2|{0, -1} .
5{0, -1};
"Extensible left brace bottom";
call charbegin('072, 12, 0, 0, 9pt, 0);
varchar '070, 0, '073, '076;
hpen; x0 = x1 = good w(.5r); x1 - x0 = 4.5u + eps;
y0 = 0; y0 = 5[y0, y1]; bot y0 = round(.5 -- 9pt pixels);
draw |wts#1|{3(x0 - x1), y0 - y1} |wps#6|{0, 1} .
5{0, 1};
% top left, bottom right combination
% extensible left brace
% top of upper stem
% bottom of lower stem
```



```

%Extensible left brace middle";
call charbegin('074,12,0,0,18pt,0);
varchar 0,0,0,'066;
% left bracket extension modules only
hpen; x2 = x0 = x1 = x3 = goodu(5); x3 = x1 + 45u + eps;
y2 = goodu(25); y1 = goodu(5 - 18pt-pixels); y1 = goodu(5[y2,y1]);
y1 = 5[y2,y1]; y1 = y1 - y2;
draw [w0]#2..[w0]#3{0,-1}..
[w0]#4{3(x1-x2),y1-y2};
draw [w0]#6..[w0]#5{0,1}..
[w0]#1{3(x1-x2),y1-y2};

%Extensible braces extension module";
call charbegin('076,12,0,0,0,3pt,0);
varchar 0,0,0,'076;
hpen; x1 = x2 = goodu(5r); y1 = 0; y2 = round(5 - 3pt-pixels);
w0 draw 1..2.

% Right braces (left-right symmetric with left ones)
charlist '011','033','051','071,0;
subroutine bigbr(var units, var n-ops, var w-xps, var height, var depth);
call charbegin(code,units,0,0,height,depth,0);
new w0,w1; w0 = round minps; w1 = round maxps;
hpen; x2 = x1 = x3 = x0 = goodu(5r);
x1 = x2 = x3 = x4 = -5(units - .3)u + eps; x1 = x7;
top,y1 = round height-pixels; bot,y2 = 1 - round pixels-depth;
y1 = 5[y1,y2] = goodu(5[y1,y2]);
draw [w0]#1{3(x1-x2),y2-y1}..[w0]#2{0,-1}..[w0]#3{0,-1}..
[w0]#4{3(x1-x2),y1-y2};
draw [w0]#7{3(x0-x2),y0-y2}..[w0]#6{0,1}..[w0]#5{0,1}..
[w0]#1{3(x1-x2),y1-y2};

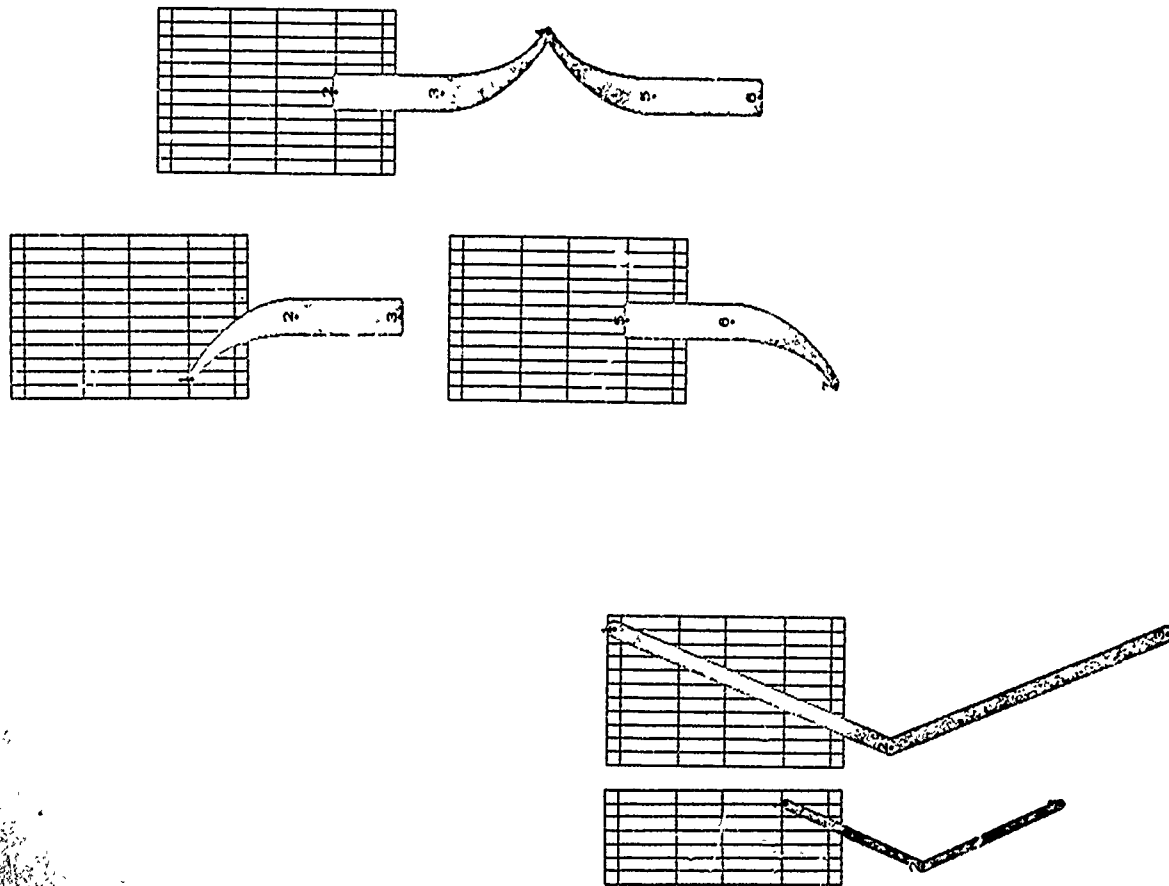
"12 point right brace";
call big-br('011,9,w0,w1,0,12pt);

"24 point right brace";
call big-br('053,11,w0 + 2deltaw,bold + deltaw,ph + pb,24pt - ph - pb);
call twen,tout;

"30 point right brace";
call big-br('051,11.5,w0 + 4deltaw,bold + 2deltaw,ph + pb,30pt - ph - pb);
call thirty;

subroutine bigbr
% free up METAFONT's memory

```



```

new w9, w9; w9 = round(w9 + 0.5*delta), w9 = round(w9 + 0.5*delta);
"Extensible right brace top";
call charbegin('071, 12, 0, 0, 0, 9pt, 0);
varchar '071, 0, '072, '076,
hpen; x2 = x1 = good_w(5r), x1 = x2 = -1.5u + eps;
top_y1 = 0; y2 = 5[y1, y1]; y1 = round(5 - 9pt pixels);
draw [w9, #1]{3(x2 - x1), y2 - y1} [w9, #2]{0, -1}
3{0, -1};
% top of upper stem

"Extensible right brace bottom";
call charbegin('073, 12, 0, 0, 0, 9pt, 0)
varchar '071, 0, '072, '076,
hpen; x2 = x1 = good_w(5r), x1 = x2 = -1.5u + eps;
y2 = 0; y1 = 5[y1, y1]; bot_w1 = round(5 - 9pt pixels);
draw [w9, #1]{3(x2 - x1), y2 - y1} [w9, #2]{0, 1}
5{0, 1};
% bottom of lower stem

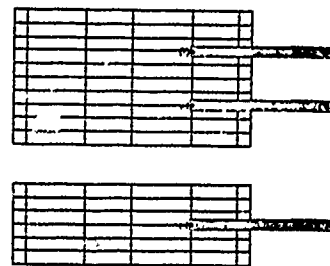
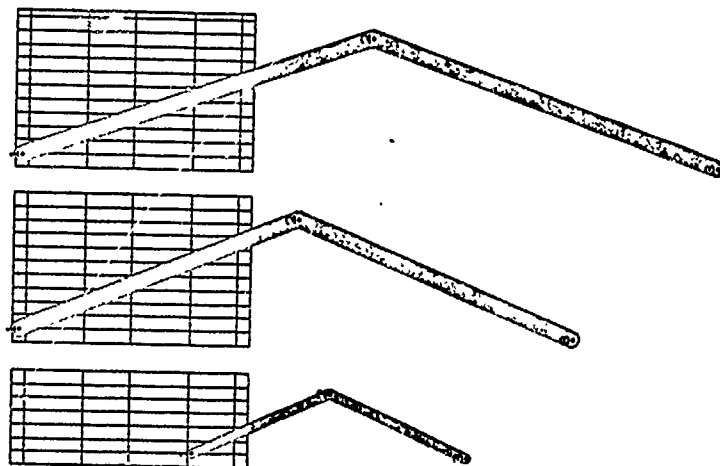
"Extensible right brace middle";
call charbegin('075, 12, 0, 0, 0, 18pt, 0);
varchar 0, 0, '067;
hpen; x2 = x1 = x1 = good_w(5r), x1 = x2 = -1.5u + eps;
y2 = good_w(25); y1 = good_w(5 - 18pt pixels), y1 = good_w(5[y1, y1]);
draw [w9, #1]{3(x2 - x1), y2 - y1} [w9, #2]{0, -1}...
draw [w9, #3]{3(x2 - x1), y2 - y1} [w9, #4]{0, 1}...
draw [w9, #5]{3(x2 - x1), y2 - y1} [w9, #6]{0, 1}...
draw [w9, #7]{3(x2 - x1), y2 - y1} [w9, #8]{0, 1}...
% top right, bottom left combination
% top of upper stem
% bottom 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 w9, w9; w9 = round(psize);
hpen; x1 = x1 = good_w(r - u), y1 = y1 = round(u);
top_y1 = round(pixels-height), bot_y1 = 1 - round(pixels depth);
y1 = good_w(5[y1, y1]);
w9 draw 1. 2. 2. 3.
% stroke

"12 point left angle bracket";
call biglab('012, 7, w9, 0, 12pt);

"24 point left angle bracket";
call biglab('034, 11, w1 - delta, ph + pb, 24pt - ph - pb);
call twentyfour;

```



```

"30 point left angle bracket";
call biglab('052, 11.5, 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 biglab(var code, var units, var psiz, var height, var depth);
call charbegin(code, units, 0, height, depth, 0);
new w1; w1 = round psiz;
open; x1 = x1; r = x1 - good; y1(r - u); if y1(r - x1) - round u,
top y1 = round pixels height; bot y1 = 1 - round pixels depth;
y2 = good; y1 = y1;
w1 draw 1 2..2 .3;
% stroke

"12 point right angle bracket";
call biglab('013, 7, w1, 0, 12pt);

"24 point right angle bracket";
call biglab('035, 11, w1 - deltaw, ph + pb, 24pt - ph - pb);
call twentyfour;

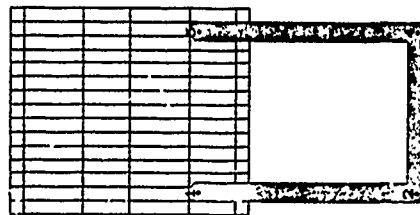
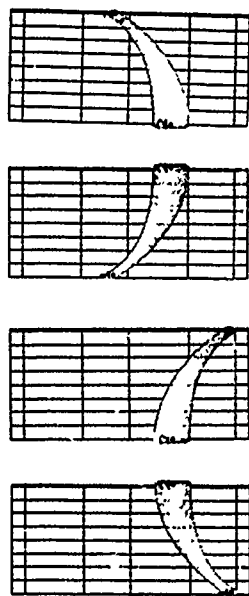
"30 point right angle bracket";
call biglab('053, 11.5, w1 - 8deltaw, ph + pb, 30pt - ph - pb);
call thirty;

subroutine biglab;
% free up METAFONT's memory

% Vertical lines
new w1; w1 = round(w1 + 2deltaw);
"Extensible vertical line extension module";
charlist '014, 0;
call charbegin('014, 6, 0, 0, 6pt, 0);
varchar 0, 0, 0, '014;
open; x1 = x1 = good; y1 = 0; y2 = round(5 - 6pt pixels);
w1 draw 1..2;
% stem

"Extensible double vertical line extension module";
charlist '015, 0;
call charbegin('015, 10, 0, 0, 6pt, 0);
varchar 0, 0, 0, '015;
open; x1 = x2 = good; y1 = 0; y2 = round(5 - 6pt pixels);
y1 = y1 = 0; y2 = y1 = round(5 - 6pt pixels);
w1 draw 1..2; draw 3..4;
% stems

```

```
% Parts for extendable horizontal braces to match vertical ones
new ruleht, wss, wss;
ruleht = 5[pwii, pwiii] + 4[pwii - pwii];
wss = round(wss + .5delaw);
wss = round(wss + 4delaw);
"Extendable downwards brace-left";
call charbegin('172, 4.5pt/pu, 0, 0, ruleht, 0, 0);
vpen; lftssx1 = 0; x2 = r + 1;
botssy2 = 0; y1 = y2 - 4.5u - eps;
draw [wss#1]{x2 - x1, 3(y2 - y1)} .. [wss#2]{1, 0}.

"Extendable downwards brace-right";
call charbegin('173, 4.5pt/pu, 0, 0, ruleht, 0, 0);
vpen; rhtssx1 = r + 1; x2 = 0;
botssy2 = 0; y1 = y2 - 4.5u - eps;
draw [wss#1]{x2 - x1, 3(y2 - y1)} .. [wss#2]{-1, 0}.

"Extendable upwards brace-left";
call charbegin('174, 4.5pt/pu, 0, 0, ruleht, 0, 0);
vpen; lftssx1 = 0; x2 = r + 1;
botssy2 = 0; y1 = y2 + 4.5u + eps;
draw [wss#1]{x2 - x1, 3(y2 - y1)} .. [wss#2]{1, 0}.

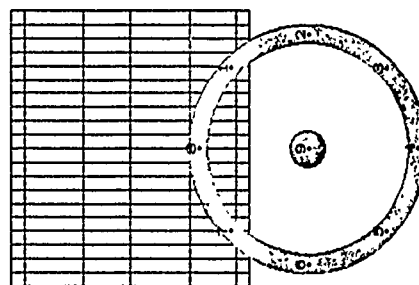
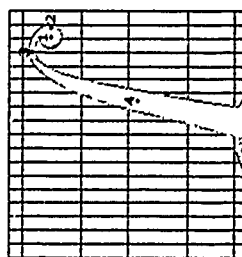
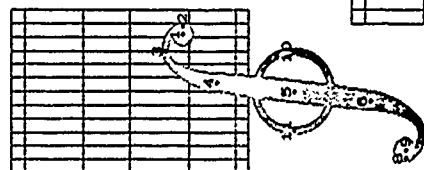
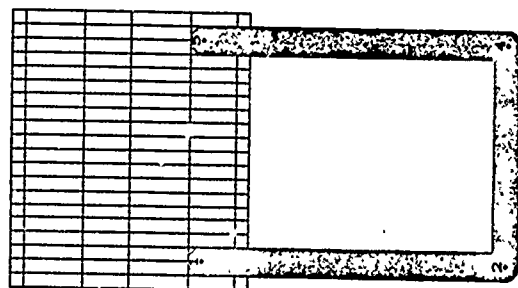
"Extendable upwards brace-right";
call charbegin('175, 4.5pt/pu, 0, 0, ruleht, 0, 0);
vpen; rhtssx1 = r + 1; x2 = 0;
botssy2 = 0; y1 = y2 + 4.5u + eps;
draw [wss#1]{x2 - x1, 3(y2 - y1)} .. [wss#2]{-1, 0}

% point
% point
% point
% point
% point
% point
% bar
% stems

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, depth, 0);
new wss; wss = round size;
vpen; lftssx1 = round u; x2 = x1; x1 = x1 - x1;
topssy1 = 0; botssy2 = 1 - round depth-pixels;
y2 = y1; y1 =
size draw 2.
draw 1 .2; u 5

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



"18 point square union sign";
call *bigsqun*(107, 20, *w*₃ + *deltaw*, 14pt),
call *eighteen*.

subroutine *bigsqun*: % free up METAFONT's memory

% Contour integral signs

charlist '110, '111,

subroutine *bigoint*(var *code*, var *units*, var *keyps*, var *maxps*,
var *dotps*, var *height*, var *depth*, var *kerncorr*):

new *w*₉, *w*₁₀; *w*₉ = round *maxps*; *w*₁₀ = round *dotps*;

hpen; *r*_{1,2} = *r*₁₀*x*₂ = round(*r* - *u*); *y*₁ = *y*₂ = *y*₃ = 75*w*₁₀;

lft *px*₀ = if *px*₅ = round *u*, *ys* = *y*₁ = *y*₂ + 75*w*₁₀,

*x*₃ = *r* - 3*u*, *x*₁ = 3*u*;

*top*₀*y*₁ = round pixels *height*, *bot*₀*y*₁ = 1 - round pixels *depth*,

*x*₅ = good is 5*r*, *y*₅ = 5*y*₁ 3*h*;

*x*₁ = *x*₃ + 3(*r* - 10*u*); *y*₁ = *y*₅ + 3(*y*₁ - *u*);

*x*₀ = *x*₃ - 3(*r* - 10*u*); *y*₀ = *y*₅ - 3(*y*₁ - *u*);

open; *w*₁₀ draw 1;

draw 9;

hpen; draw *lu*₁2(0, 1) - *lu*₀3(-1, 0) | *keyps*4(*x*₀ - *x*₁, *y*₀ - *y*₁) | *w*₁₀#|5. % upper bulb

% lower bulb

% stem

% bowl

"12 point contour integral sign",

call *bigoint*('110, 12, *w*₁, *w*₂, bold, 10pt/2, 10pt, 3.5pt).

"24 point contour integral sign",

call *bigoint*('111, 18, *w*₁, bold + 2*deltaw*, *w*₁, *ph* + *pb* - 3pt,

24pt - *ph* - *pb* - 3pt, 8pt);

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, *depth*, 0);

new *w*₉, *w*₁₀ = round *size*;

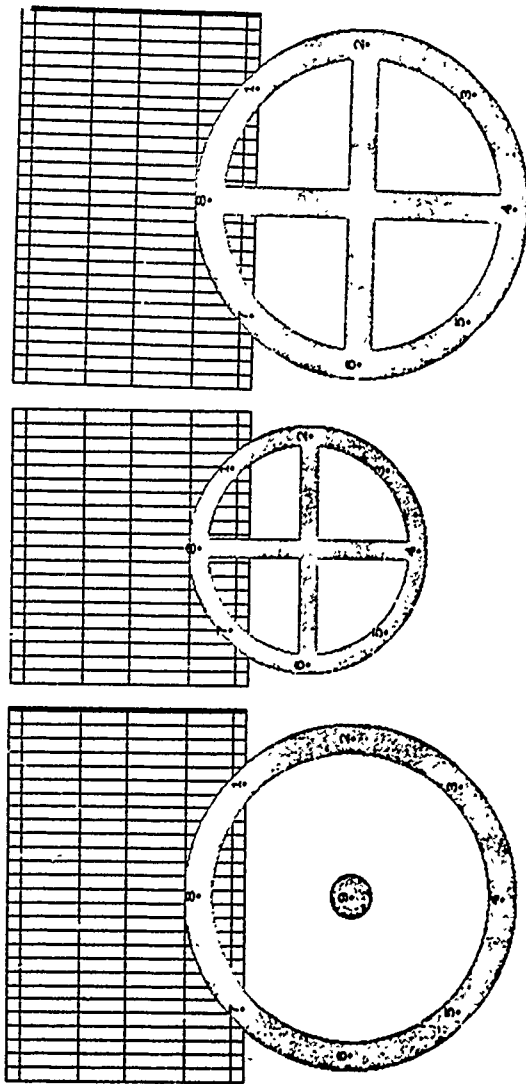
open; lft *px*₀ = round *u*; *top* *px*₀ = 0; *x*₂ = *r* - *x*₀, *bot* *px*₀ = 1 - round *depth* pixels,

call *circle*(1, 2, 3, 4, 5, 6, 7, 8, *size*); % circle

*x*₀ = .5(*x*₀, *x*₂); *y*₀ = .5(*y*₀, *y*₁); *dotsize* draw 9. % dot

"12 point circle-dot operator",

call *bigodot*('112, 20, *w*₁, bold + 4*deltaw*, 10pt).



```
"18 point circle-dot operator";
call bigodot('113,27.2,w1+deltaw,bold+6deltaw,14pt);
call eighteen.

subroutine bigodot:
% 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,depth,0);
new w0; w0 = round size;
open, ift w0 = round u; top w0 = 0; x2 = r - x0; bot w0 = 1 - round depth pixels;
% circle
% plus
w0 draw 2..6; draw 4..8.

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

"18 point circle-plus operator";
call bigoplus('115,27.2,w1+deltaw,14pt);
call eighteen.
```

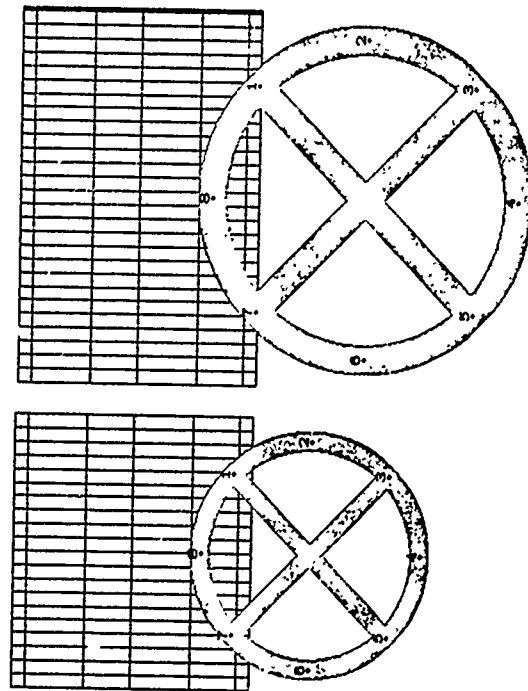
```
subroutine bigoplus:
% free up METAFONT's memory

% Circle-times operators
charlist '116,'117;
subroutine bigotimes(var code,var units,var size,var depth);
call charbegin(code,units,0,0,depth,0);
new w0; w0 = round size;
open, ift w0 = round u; top w0 = 0; x2 = r - x0; bot w0 = 1 - round depth pixels;
% circle
% times
w0 draw 1..5; draw 3..7.

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

"18 point circle-times operator";
call bigotimes('117,27.2,w1+deltaw,14pt);
call eighteen.
```

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




```

% Summation signs
charlist '120, '130;
subroutine bigsum(var code, var units, var minps, var maxps, var serif, var depth);
call charbegin(code, units, 0, 0, depth, 0);
new w18, w19; w18 = round minps aspect; w19 = round maxps aspect;
hpen; lft0x26 = round u; rft0x28 = r - u; x27 = good(x28 - (r - 2u)/11);
new ss, ss = 1.4aspect serif u + eps;
if ss + w1 > 25depth pixels new ss, ss = 25depth pixels - u6 + eps;
fi;

vpen, top121 = top0y1 = 0, y17 = y16, bot0y26 = bot121;
y27 = y26, y28 = y27 - ss;
bot0y1 = bot0y6 = 1 - round(depth pixels); y1 = y16;
top0y6 = top0y16; y17 = y16, y28 = y17 + ss;
x21 = x26 = x1 = x0, x25 = x27 = x1, x3 = x28;
call 'a arm(26, 27, 28);
w18 draw 24 .25; w19 draw 4 .5;
if ucs = 0, draw 5 8; w18 draw 25 28;
else if w1 < w1 draw w15 5 ju18,
draw w18 25 ju128;
else draw 5 8, w18 draw 25 28;
fi;

fi;
new w18, w19; w18 = round minps, w19 = round maxps;
hpen; lft0x21 = lft0x21; x1 = x1; x10 = 5[x0, x1];
x16 = round(u + 1/11(r - 2u)); lft0x12 = lft0x15;
lft0x29 = x16; rft0x29 = rft0x11; rft0x11 = rft0x15;
y10 = 5[y16, y11]; y11 = y16 = 5[y16, y1]; y12 = y15, y13 = y11 - y1;
new aa, bb; lft0x15 = aa[lft0x1, x16]; y15 = aa[y16, y1];
lft0x15 = bb[x0, x1], y15 = bb[y16, y1];
w19 draw 13. 12;
hpen; w18 draw 1..15;
hpen, w18 draw 1 15;
rpen; w19 draw 10 11;
hpen; w19 draw 9. 10;
hpen; w18 draw 9..11;
draw 14..11.

% upper diagonal
% erase excess at upper left corner
% sharpen upper left corner
% erase excess at right
% erase excess at left
% lower diagonal
% sharpen middle corner

"12 point summation sign";
call bigsum('120, 19, w18, w19, ucs, 10pt)

"18 point summation sign";
call bigsum('130, 26, w18 + deltax, bold + 4deltaw, 18 * 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, depth, 0);
new w18, w19; w18 = round minps; w19 = round maxps;
open; if w21 = round u; x1 - x1 = round(serif.u + cps);
if w23 = if w25; r1, w25 = r1, w25;
top w18 = 0; y3 = y1 - ss; y10 = y1;
x2 = {x1, {x1, x2}}; y2 = {y1, {y1, y2}};
bot w18 = 1 - round(depth, pixels);
w1 + y1 = y2 + y2 = y1 + y1; x3 = x1; x5 = x2; x1 = x1;
x0 = x10; x1 + x2 = x3 + x3 = x0 + x1; y1 = y1; y5 = y5;
y11 = y1; y12 = y2; y13 = y1; y14 = y1; y15 = y1;
y16 = y1; y17 = y1; y18 = y1; y19 = y1; y20 = y1;
r = x1 + x1 = x12 + x2 = x13 + x3 = x14 + x1 = x15 + x1 =
x16 + x1 = x17 + x1 = x18 + x3 = x19 + x1 = x20 + x1;
w18 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},
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};
draw 10..20.

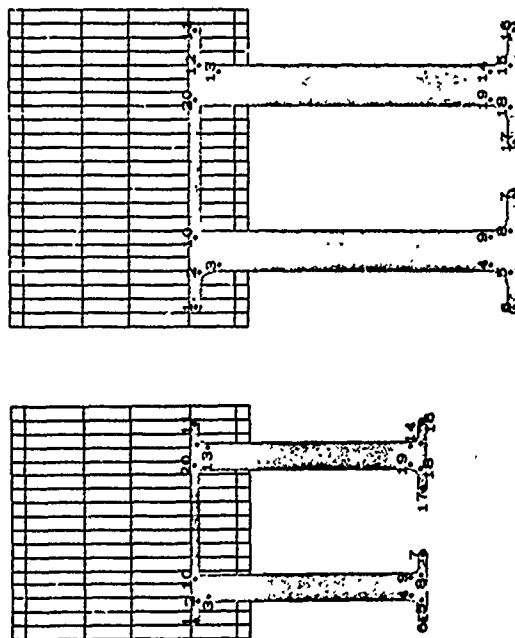
% left stem and serifs
% right stem and serifs
% bar

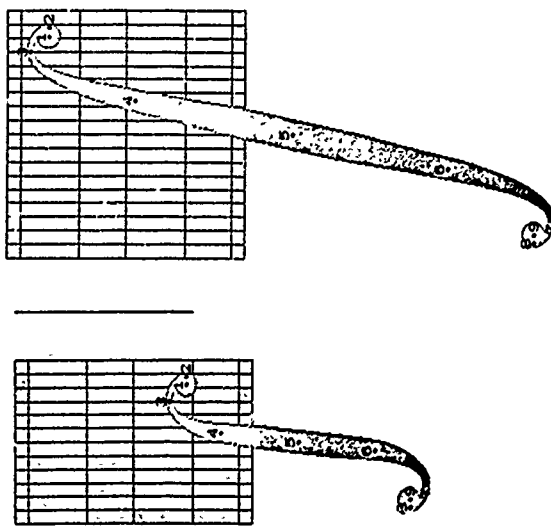
"12 point product sign";
call bigprod("121, 17, w10, bold + deltaw, ucs, s, 10pt).

"18 point product sign";
call bigprod("131, 23, w10 + deltaw, bold + 5deltaw, 1.8 * ucs, 1.8s, 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 dots, var height, var depth, var kerncorr)
call charbegin(code, units, 0, 0, height, depth, kerncorr);
new w1, w2; w1 = round maxps; w2 = round dots;
hpen; rt1 = r10x2 = round(r - u); y1 = y2 = y3 = .75w1;
lt1 = r10x3 = round u; y3 = y4 = y5 = y7 + .75w1;
x1 = r - 3u, x2 = 3u;
top1y = round pixels height, bot1y = 1 - round pixels depth;
x1 = good u, y1 = 5[y1, y7];
x1 = x2 + 3(r - 10u); y1 = y2 + 3(y1 - y7);
x0 = x1 - 3(r - 10u); y1 = y2 - 3(y1 - y7);
open; w1 draw 1;
draw 9;
hpen; draw [u1]2{0, 1} . [u2]3{-1, 0} . [keyps]4{x1 - x2, y1 - y2} [u3]5{...
    % upper bulb
    % lower bulb
    % 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, w1, bold + 2deltaw, w1, ph + pb - 8pt,
    24pt - ph - pb - 8pt, 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, depth, 0);
new w1; w1 = round size;
open; lt1 = round u; x2 = x1; x3 = r - x1, x1 = x2 = r - x1;
top1y = 0; bot1y = 1 - round depth pixels;
y2 = y1 = {y1, y4}; y5 = y1;
call qcirc(3, 6, 2, size); call qcirc(3, 7, 4, size);
size draw 1..2; draw 4..5.

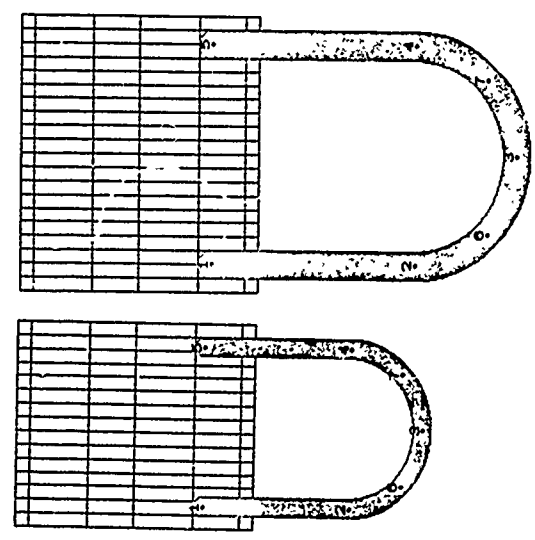
"12 point set union sign";
call bigun('123, 15, w1, 10pt).

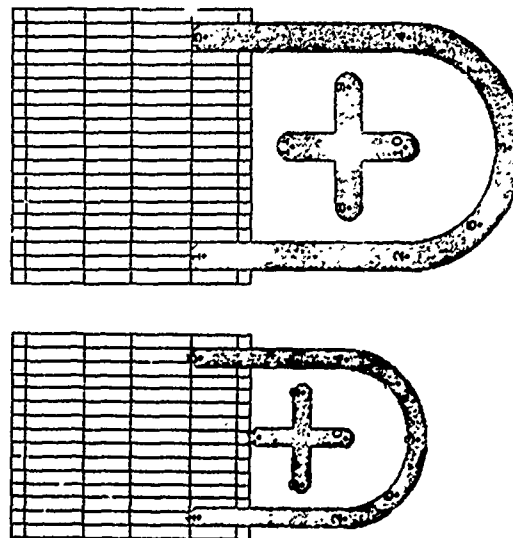
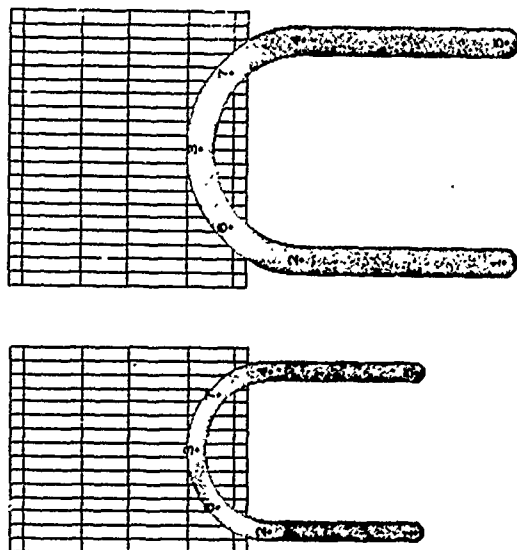
"18 point set union sign";
call bigun('133, 20, w1 + deltaw, 14pt);
call eighteen.

```

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

```





```

% Set intersection signs
charlist '124, '134;
subroutine bigun(var code, var units, var size, var depth);
call charbegin(code, units, 0, 0, depth, 0);
new w10; w10 = round size;
open; lt10x1 = round u; x2 = x1; x1 = r - x1; x1 = x2 = r - x1;
top10y1 = 0; bot10y1 = 1 - round depth-pixels;
y2 = y1 = 1/2(y1, y2); y5 = y1;
call qcirc(3, 6, 2, size); call qcirc(3, 7, 4, size);
size draw 1..2; draw 4..5.
% cap
% stems

"12 point set intersection sign";
call bigun('124, 15, w1, 10pt).

"18 point set intersection sign";
call bigun('134, 20, w3 + deltaw, 14pt);
call eighteen.

subroutine bigun:
% free up METAFONT's memory

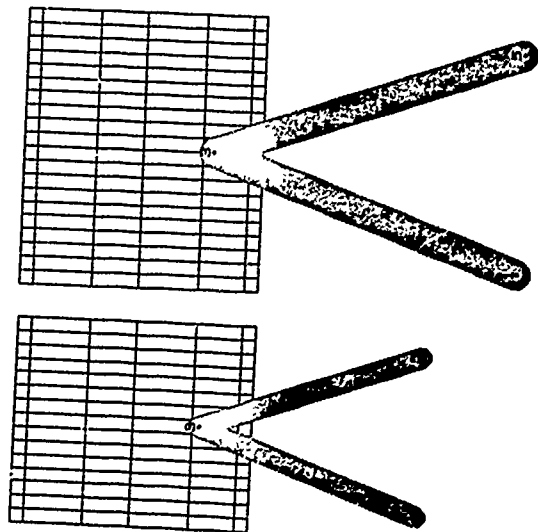
% Multiset union signs
charlist '125, '135;
subroutine bigmun(var code, var units, var size, var depth);
call charbegin(code, units, 0, 0, depth, 0);
new w10; w10 = round size;
open; lt10x1 = round u; x2 = x1; x3 = r - x3; x1 = x2 = r - x1;
top10y1 = 0; bot10y1 = 1 - round depth-pixels;
y2 = y1 = 1/2(y1, y2); y5 = y1;
call qcirc(3, 6, 2, size); call qcirc(3, 7, 4, size);
size draw 1..2; draw 4..5;
y8 = y9 = .47(y1, y2); x8 = r - x9 = x1 + 1.75size - eps;
x10 = x11 = x1; 5[y10, y11] = y8; y11 - y10 = x1 - x8;
draw 8..9; draw 10..11.
% cup
% stems
% enclosed plus sign

"12 point multiset union sign";
call bigmun('125, 15, w1, 10pt).

"18 point multiset union sign";
call bigmun('135, 20, w3 + deltaw, 14pt);
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 w19; w19 = round size;
open; ll1921 = round u; x3 = r - x1; x5 = r - x1;
top1921 = 0; bot1921 = 1 - round depth/pixels - o;
y5 = y1;
size draw 1..3; draw 3..5.
```

% diagonals

```
"12 point lattice meet sign";
call bigmeet('126, 15, w1, 10pt).
```

```
"18 point lattice meet sign";
call bigmeet('136, 20, w3 + deltav, 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 w19; w19 = round size;
open; ll1921 = round u; x3 = r - x1; x5 = r - x1;
top1921 = o; bot1921 = 1 - round depth/pixels;
y5 = y1;
size draw 1..3; draw 3..5
```

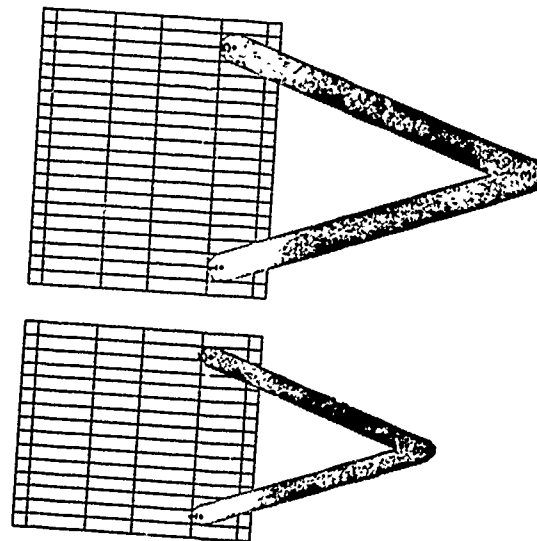
% diagonals

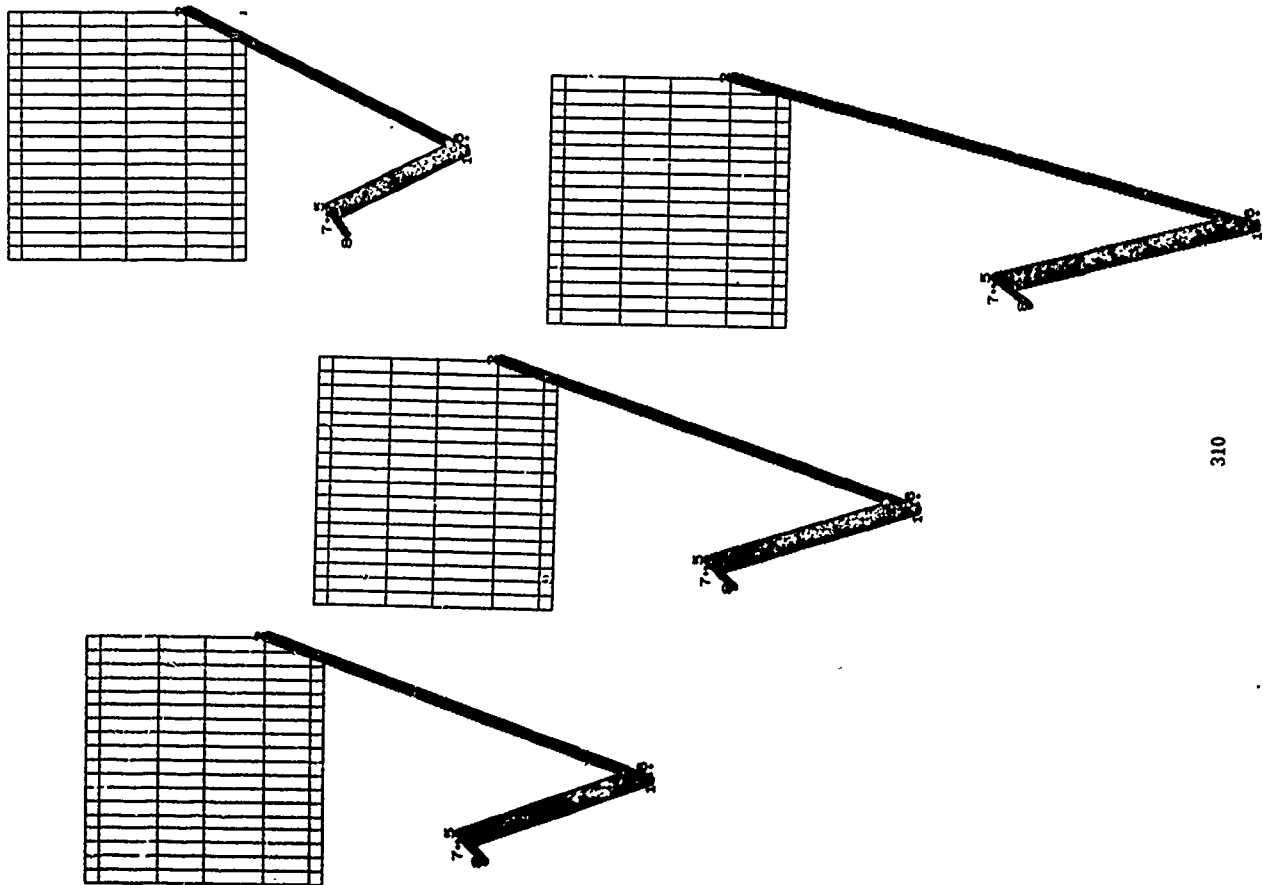
```
"12 point lattice join sign";
call bigjoin('127, 15, w1, 10pt).
```

```
"18 point lattice join sign";
call bigjoin('137, 20, w3 + 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; x1 = good10(1/3 r); x2 = r + 1; bot10y1 = 1 - round depth-pixels;
top10y2 = 0;
y1 = y1 = y1 = good10(.5[y1, y2]), y1 = y1 = y1;
x1 = 1.5[x1, x1]; lt10x1 = lt10x1; rt10x1 = rt10x1;
lt10x1 = lt10x1; rt10x1 = rt10x1;
x8 = x1 - u; new x1; x8 = aa[x1, x2]; y8 = aa[y1, y2];
hpen; w1 draw 3..4;
w10 draw 7..4; w10 draw 5..6;
hpen; w2 draw 8..5;
hpen; w3 draw 8..5;
hpen; w4 draw 1..2;
hpen; w10 draw 1..2.

"12 point radical sign";
call bigroot('160, 18, 0, 12pt).

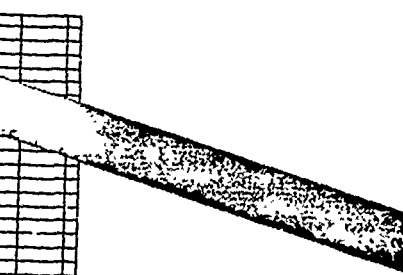
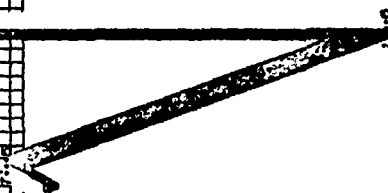
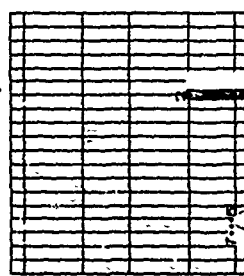
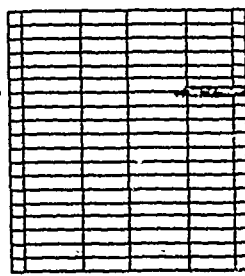
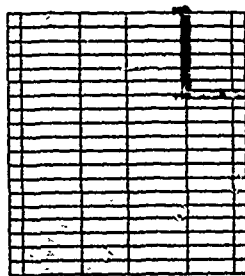
"18 point radical sign";
call bigroot('161, 18, 0, 18pt);
call eighteen.

"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, 18pt, 0);
varchar '166, 0, '164, '166;
hpen; x1 = x2 = good10(r - 6u); y2 = 0; bot10y1 = round(.5 - 18pt.pixels);
y3 = y2 = y7 = good10(1/2y, y1); y4 = y3 = y6 = y1;
x7 = 3u; ft10x7 = ft2x3; rt2x3 = rt10x3;
ft10x1 = ft2x1; rt2x1 = rt10x1;
x8 = x7 - u; new aa; x8 = aa[x0, x2]; y4 = aa[y0, h + b];
hpen; w2 draw 3 4;
w10 draw 7..4; w3 draw 5..6;
lpen#; w2 draw 8..5;
hpen; w3 draw 8..5;
rpen#; w2 draw 1..2;
hpen; w10 draw 1..2.

"Extensible radical sign-top";
call charbegin('166, 19, 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.

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

% left diagonal
% sharpen the corners
% erase excess at upper left
% serif
% erase excess at lower right
% right diagonal

% link end stem
% stem

```

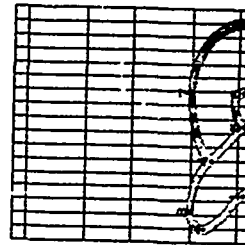
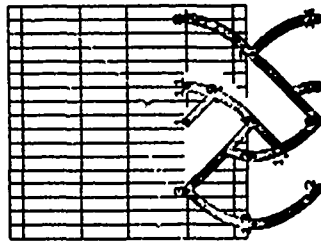
The file mexext mf

```

% 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 + ph, 24pt - ph - ph);
call twentyfour.

```



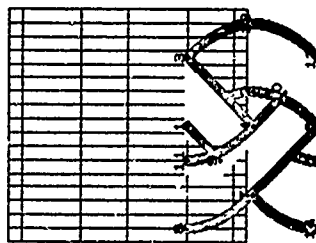
```

% Left pretzels (left-right symmetric with respect to right ones)
"Extensible left pretzel-extension module";
call charbegin(056,17,0,0,0,0,0,0,0,0);
varchar '144,0,'146,'056;
hpen; x13 = good10u; z3 = x12 = 3.5u; z2 = good10 6u; z1 = x1 = x0 = 8.5u; % the left pretzel
z1 = x2 = good10 11u; z7 = 13.5u; z8 = x11 = good10 16u;
y1 = y7 = y8 = y11 = 0;
y1 = -3/8 pt-pixels; y7 = y8 + 3/8 pt-pixels; y2 = y13 = .5[y7,y12];
y5 = y6 = y12 = y14 = round(.5 - 3/8 pt-pixels);
z0 = 11u; y0 = -7/8 pt-pixels;
z10 = 6u; y10 = y12 + 7/8 pt-pixels;
w10 draw 1{x0 - x7,y0 - y7}..2{0,-1};
rpen#; u ddraw 3..4,1..4; % right of upper left strand
hpen; w10 draw 4{x1 - x7,y1 - y7}..5{0,-1}; % erase part covered by upper middle strand
hpen#; u ddraw 6..7,6..4; % left of lower right strand
hpen; w10 draw 3..4; % erase part covered by lower middle strand
draw 6..7{x7 - x0,y7 - y0}..8{0,1}; % right of lower middle/upper right strand
draw 1..9; % right of upper middle strand
rpen#; u ddraw 4{x1 - x10,y1 - y10}..11{0,1}; % erase part covered by upper right strand
hpen; w10 draw 10;
4{x1 - x10,y1 - y10}..11{0,1}; % left of upper right/lower middle strand
hpen#; u ddraw 2{0,-1}..6{x0 - x1,y0 - y1}, % erase part covered by lower left strand
2..10;
hpen; w10 draw 12{x2 - x1,y2 - y1}..
13{0,1}..3{x10 - x12,y10 - y12}; % left of lower left/upper left strand
draw 2{0,-1}..6{x0 - x1,y0 - y1}; % right of lower left strand
draw 7{x7 - x0,y7 - y0}..14{0,-1}; % right of upper middle/lower right strand

"Extensible left pretzel-top";
call charbegin(144,17,0,0,0,0,0,0,0,0);
hpen; z1 = 3.5u; y1 = y7 = y8 = y0 = round(5 - 3/8 pt-pixels);
z2 = good10u; y2 = 1/8 y1;
z1 = 5[x1,x2]; y0 = y7 = good0 0;
z1 = 6u; y1 = y5 + 7/8 pt-pixels;
z4 = 8.5u; x0 = x7 = 11u; x8 = good0 16u;
w10 draw 1{x1 - x5,y1 - y5}..2{0,1}..3{1,0}..
4{7,-x1,y5 - y1}..5;
draw 5{x5 - x1,y1 - y5}..6{0,-1};
draw 4{x5 - x1,y1 - y5}..7{1,0}..8{0,-1};

```

% end piece
 % bottom of twist
 % top of twist

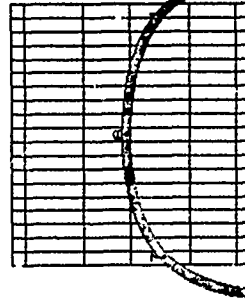
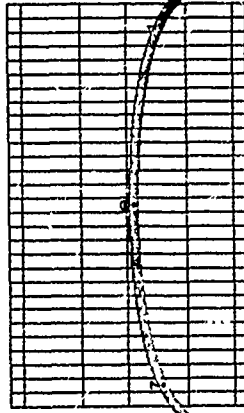
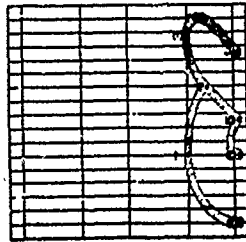


```

% Extensible left pretzel-bottom";
call charbegin(146, 17, 0, 0, 0, 33pt, 0);
hpen; x1 = 3.5u; y1 = y5 = y6 = y8 = 0;
x2 = good10u; y2 = 1/2 y1;
x3 = 5x1, x4; y3 = y7 = good10(5 - 33pt-pixels);
x1 = 6u; y1 = y5 = y7 = 23pt-pixels;
x5 = 3.5u, x6 = x7 = 11u; x8 = good10 16u;
w10 draw 1{x1 - x5, x5 - y1}..5;
4{x5 - x6, y5 - y1}..5;
draw 5{x5 - x6, y1 - y5}..6{0, 1};
draw 4{x5 - x6, y1 - y5}..7{1, 0}..8{0, 1}.

% Right pretzels (left-right symmetric with respect to left ones)
"Extensible right pretzel-extension module";
call charbegin(145, 0, 147, 057;
varchar 145, 0, 147, 057;
hpen; r - x13 = good10u; r - x3 = r - x12 = 3.5u, r - x2 = good10 6u, x1 = x1 =
x0 = r - 8.5u;
x11 = x5 = r - good10 11u; r - x7 = 13.5u, x4 = x14 = r - good10 16u;
y1 = y1 = y8 = y11 = 0;
y1 = -23pt-pixels; y7 = y6 + 23pt-pixels; y2 = y11 = .5[y1, y2];
y5 = y6 = y12 = y14 = round(.5 - 33pt-pixels);
r - x3 = 11u; y3 = -23pt-pixels;
r - x10 = 6u; y10 = y12 + 23pt-pixels;
w10 draw 1{x6 - x7, y6 - y7}..2{0, -1};
lpen 4; u ddraw 3..4, 1..4; % erase part covered by upper middle strand
hpen; w10 draw 4{x1 - x3, y1 - y1}..5{0, -1}; % right of lower left strand
lpen 4; u ddraw 6..7, 6..4; % erase part covered by lower middle strand
hpen; w10 draw 4; % right of upper middle strand
draw 6..7{x7 - x11, y7 - y6}..8{0, 1}; % left of lower middle/upper left strand
draw 1..9; % left of upper middle strand
lpen 4; u ddraw 4{x1 - x10, y1 - y10}..11{0, 1};
9, 11; % erase part covered by upper left strand
hpen; w10 draw 10..
4{x1 - x10, y1 - y10}..11{0, 1}; % right of upper left/lower middle strand
rpen 4; u ddraw 2{0, -1}..6{x9 - x1, y9 - y1},
2..10; % erase part covered by lower right strand
hpen; w10 draw 12{x1 - x1, y1 - y1}
13{0, 1}..13{x10 - x1, y10 - y12};
draw 2{0, -1}..6{x9 - x1, y9 - y1}; % right of lower right strand
draw 7{x7 - x9, y7 - y9}..14{0, -1}; % left of upper middle/lower left strand

```



```

"Extensible right pretzel-top";
call charbegin(145, 17, 0, 0, 0, 23 pt, 0);
hpen; r - x1 = 3.5u; y1 = y5 = y6 = y8 = round(.5 - 23 pt pixels);
r - x2 = good10u; y2 = y1;
x3 = .5[x1, x2]; y3 = y1 = good0;
r - x1 = 6u; y1 = y5 + 23 pt pixels;
r - x2 = 8.5u; r - x0 = r - x1 = 11u; r - x8 = good10; 16u;
w10 draw 1{x1 - x2, y1 - y2}..2{0, 1}..3{-1, 0}..
4{x2 - x1, y2 - y1}..5;
draw 5{x2 - x1, y1 - y2}..6{0, -1};
draw 4{x2 - x1, y1 - y2}..7{-1, 0}..8{0, -1}.

"Extensible right pretzel-bottom";
call charbegin(147, 17, 0, 0, 0, 23 pt, 0);
hpen; r - x1 = 3.5u; y1 = y5 = y6 = y8 = 0;
r - x2 = good10u; y2 = y1;
x3 = .5[x1, x2]; y3 = y1 = good0(5 - 23 pt pixels);
r - x1 = 6u; y1 = y5 - 23 pt pixels;
r - x2 = 8.5u; r - x0 = r - x1 = 11u; r - x3 = good10; 16u;
w10 draw 1{x1 - x2, y1 - y2}..2{0, -1}..3{-1, 0}..
4{x2 - x1, y2 - y1}..5;
draw 5{x2 - x1, y1 - y2}..6{0, 1};
draw 4{x2 - x1, y1 - y2}..7{-1, 0}..8{0, 1}.

% Circumscribed circles
subroutine bigcircle(var code, var units, var depth, var asp);
call charbegin(code, units, 0, 0, depth, 0);
new aa; aa = .5 sqrt(r + depth-depth pixels/asp/asp);
x8 = .5r; y8 = -.5 round depth pixels;
x2 = good10(x8 + aa); y4 = good10(y2 + asp aa);
call circle(1, 2, 3, 4, 5, 6, 7, 8, w10).

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

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

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

```

% end piece
% bottom of twist
% top of twist

% end piece
% top of twist
% bottom of twist

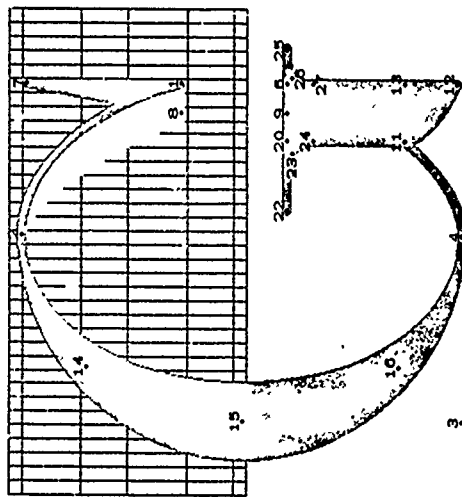
% how

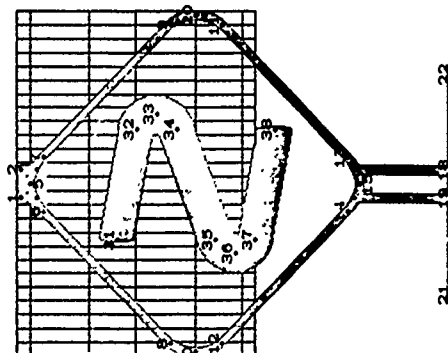
63

```

"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);
open; r197x1 = rt98x0 = round(1/4r); if99x1 = round(r/14;
x2 = x1 = 1/4r; top97y2 = h + 2o; bot97y1 = 1 - round(12pt-pixels - 2o);
y3 = y4; y5 = .8[y2, y4]; y1 = good0.25[y2, y4];
x6 = x1; y6 = good0.75[y2, y4];
x7 = x1; top97y7 = h; if97x8 = if98x1; y8 = y1;
hpcn; w97 ddraw 1..7, 8..7,
lpn4; w97 ddraw (6..1)..2{-1, 0},
(6..8)..2{-1, 0};
open; w97 draw (6..1)..2{-1, 0};
x15 = x1, y15 = 5[y2, y4];
call hcirc(2, 14, 15, 16, 17, w97);
call a darc(2, 3, w99);
crsbreak .5[y2, y4];
y11 = y10 = y13 = 1/2[y1, y4];
if97x11 = if98x9; r197x12 = r198x9; y12 = y1 + 2o;
w97 draw 4{1, 0}..11..9{0, 1};
x20 = x11, y20 = y9; x21 = x12; y21 = y1;
ddraw 20..11, 21..13;
ddraw 13..12, 11{0, -1}..12{2(x12 - x11), y12 - y11};
y22 = y15 = y20; y24 = y27 = y20 - 2s;
y23 = y28 = 1/2[y20, 1/2[y22, y24]];
x22 = x20 - 1/4r; x23 = x20; x24 = 1/2[x20, 1/2[x22, x24]];
x25 = x21 + r/14; x27 = x21; x28 = 1/2[x21, 1/2[x23, x25]];
minvr 0; minvs 0;
ddraw 22{1, 0}..23..24{0, -1}, 25{-1, 0}..26 27{0, -1},
minvr 0.5; minvs 0.5.

```





```

'Dangerous bend sign';
call charbegin(177,25,0,0,ph+pi/2,0,0);
% 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;
top10y1 = h + b; y2 = y1;
x3 = x1; x4 = x2; x5 = r - x3; x6 = r - x7; x8 = r - x9; x0 = r - x20;
x0 = x5 - 2u; x4 = u; left0x0 = 0;
y5 = good10(.33(h + b)); y6 = y7 = .33(h + b); y8 = y9 = .33(h + b); y0 = y20 = 0;
y11 + y0 = y12 + y8 = y13 + y7 = y14 + y6 =
y15 + y5 = y16 + y1 = y17 + y3 = 0;
x11 - x0 = x12 - x4 = x13 - x7 = x14 - x6 = x15 - x5 =
x10 - x1 = x17 - x3 = x18 - x2 = x19 - x4 = 0;
y18 = y19 = y21 = y22; bot10y21 = 1 - round(11pt.pixels);
x21 = r - x22 = good10(.5r - 8u);
y3 = y1 = .5[y7, y6];
w10 draw 1..2;
draw 21..22;
draw 1..3; draw 15..17;
draw 2..4; draw 18..16;
rpen#; draw 115{-1,0}..2u16{x0 - x6, y4 - y6};
draw 1115{-1,0}..2u14{x12 - x11, y12 - y11};
ipen#; draw 115{1,0}..2u17{x0 - x7, y0 - y7};
draw 1115{1,0}..2u13{x11 - x10, y11 - y10};
cpen, w10 draw 5{-1,0}..6{x0 - x6, y0 - y6}..
8{x0 - x6, y4 - y6}..0{0,-1};
9{x0 - x7, y0 - y7}..20{0,-1};
draw 15{-1,0}..14{x12 - x11, y12 - y11}..
12{x12 - x10, y12 - y11}..0{0,1};
draw 15{1,0}..13{x11 - x10, y11 - y10}..
11{x11 - x13, y11 - y11}..20{0,1};
new w99; w99 = round(bold + 4deltaw);
x31 = r - x30 = x3 - 4u, x32 = x11 = x38;
x35 = x37 = x31; x33 = r - x30 = good99(x3 + 5u);
y31 = .33(h + b); y32 = .33(h + b); y33 = .33(h + b);
y31 + y35 = y32 + y37 = y33 + y36 = y31 + y35 = 0;
ypen; w99 draw 31..32;
draw 38..37;
cpen; w99 draw 32{x32 - x31, y32 - y31}..33{0,-1}..
34{x35 - x34, y35 - y34}..35{x35 - x31, y35 - y31}..36{0,-1}..
37{x38 - x37, y38 - y37};
crsbreak 0;
minvr .5; minvs .5.

```

```

% left-right symmetry;
% top-bottom symmetry
% top of signpost
% ground level
% left edge of signpost
% right edge of signpost
% erase hidden left edge
% erase hidden right edge
% upper left edge of sign
% upper right edge of sign
% lower left edge of sign
% lower right edge of sign
% upper bar
% lower bar
% the dangerous bend

```

	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Α	Β	Η	Σ	Υ
'010	Φ	Ψ	Ω	Ι	Ј	·	·	·
'020	·	·	·	·	·	·	·	·
'030	·	·	·	Β	Ξ	ϰ	Ε	ϰ
'040	ϰ	Ι	"	'	∞	%	&	·
'050	(	)	*	+	,	·	·	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	;	<	=	>	?
'100	Ø	Α	Β	С	Д	Е	Ғ	Г
'110	Н	Ι	Ј	К	Л	М	Ν	О
'120	Р	Q	᠕	᠐	᠔	᠕	᠖	᠗
'130	Х	Y	Z	[	"	]	-	—
'140	·	а	б	с	д	е	ғ	г
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	ff	ff	ff	ff	ff

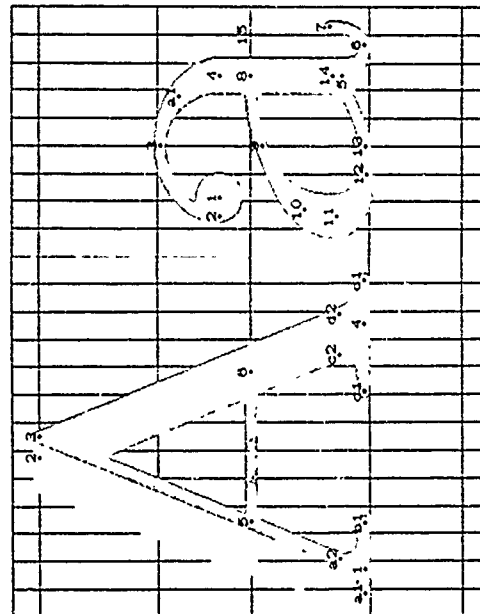
Mathematics books and journals do not look as beautiful as they used to. It is not that their mathematical content is unsatisfactory, rather that the old and well developed traditions of typesetting have become too expensive. Fortunately, it now appears that mathematics itself can be used to solve this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

STANDARD FONTS

The file cmr10.mf

"Computer Modern Roman 10 point";
 ph = 28; p: = 18; pe = 20; pd = 38;
 pb = 28; po = 16; ps = 38; pa = .5(ph - pd);
 pw = 38; pvi = 38; pwv = 38; pwid = 38;
 pwiv = 38; pwi = 38; aspect = 1.0;
 pu = 20; lcs = 1.075; ucs = 1.7; sc = 0; ls = 0;
 slant = 0; sqrtwo = 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	Α	Ε	Η	Σ	Τ
'010	Φ	Ψ	Ω	Ι	Ј	ˆ	˙	˚
'020	˚	˚	˚	˚	˚	˚	˚	˚
'030	˚	˚	˚	Β	Ξ	ϰ	Ξ	ϰ
'040	ϰ	Ι	˚	˚	∞	%	Ξ	˙
'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	[	ˆ	]	-	-
'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	Œ	ß	À	Ä	Å

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 *Esop's Envr* is naive 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".

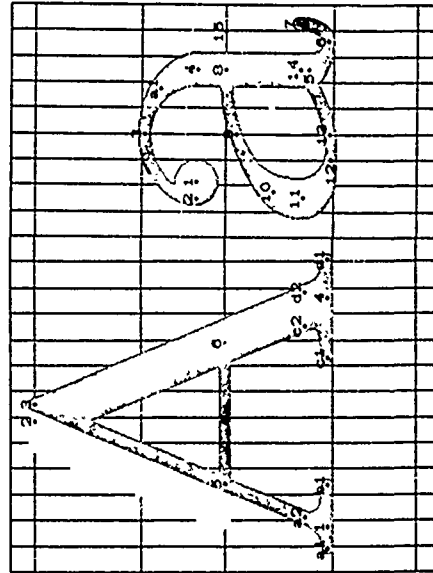
The file cmr9.mf

```

"Computer Modern Roman 9 point",
ph = 225; px = 144; pe = 38; pd = 98;
pb = 144; po = 14; ps = 18; pa = .5(ph - pd);
pw = 38; pwi = 38; pwii = 38; pwiii = 38;
pwiv = 38; pwv = 38; aspect = 1.0;
pu = 144; lcs = 1.05; ucs = 1.65; sc = 0; ls = 0;
slant = 0; sqrttwo = sqrt 2; fixwidth = 0;
halfld = 0; varg = 0; lowast = 0; ligs = 1.

input cmbase; call fontbegin;
input roman;
end

```

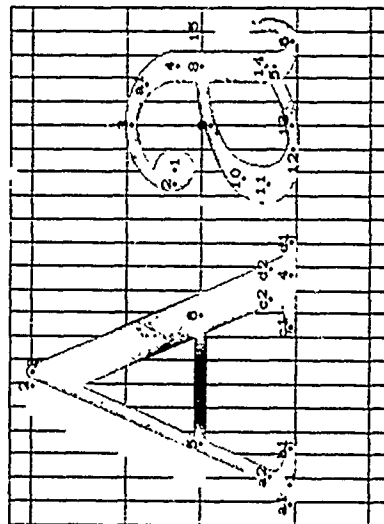


```

"Computer Modern Roman 8 point";
ph = 200; px = 128; pe = 72; pd = 38;
pb = 18; po = 36; ps = 38; pa = .5(ph - pd);
pw = 9; pwi = 36; pwi1 = 36; pwi2 = 36;
pwiv = 36; pvv = 36; aspect = 1.0;
pu = 12; lcs = 1.03; ucs = 1.56; sc = 0; ls = 0;
slant = 0; squitwo = 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	Α	Β	Π	Σ	Υ
'010	Φ	Ψ	Ω	ι	ϋ	·	·	·
'020	·	·	·	·	·	·	·	·
'030	·	·	·	Β	·	·	·	·
'040	·	·	·	·	·	·	·	·
'050	(	)	·	+	·	·	·	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	;	<	=	>	?
'100	Θ	Α	Β	Γ	Δ	Ε	Ζ	Η
'110	Θ	Α	Β	Γ	Δ	Ε	Ζ	Η
'120	Θ	Α	Β	Γ	Δ	Ε	Ζ	Η
'130	Θ	Α	Β	Γ	Δ	Ε	Ζ	Η
'140	Θ	Α	Β	Γ	Δ	Ε	Ζ	Η
'150	Θ	Α	Β	Γ	Δ	Ε	Ζ	Η
'160	Θ	Α	Β	Γ	Δ	Ε	Ζ	Η
'170	Θ	Α	Β	Γ	Δ	Ε	Ζ	Η

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 *Knop's Cuvres* is naïve about the efficient preparation of *Knop's Cuvres*. This is a sample of the font when the resolution is 200 dots per inch and 3.6 dots per "point".

the file cmr7.mf

```

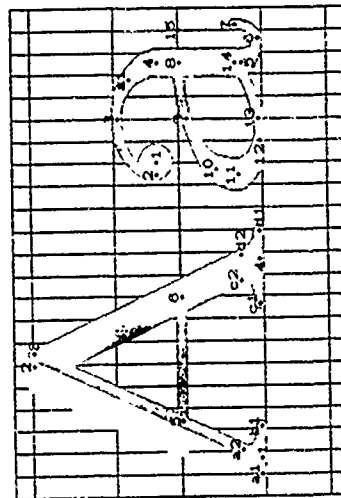
"Computer Modern Roman 7 point";
ph = 1/2; px = 1/2; pe = 3/8; pd = 3/8;
pb = 1/2; po = 3/8; ps = 1/2; pa = 5(ph - pd);
pw = 3/8; pwi = 3/8; pwi = 3/8; pwiii = 3/8;
pwiv = 3/8; pwv = 3/8; aspect = 1.0;
pu = 1/2; lcs = .97; uc = 1.44; sc = 0; ls = 0;
slant = 0; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 0; varg = 0; lowast = 0; ligs = 1.

input cmbae; call fontbegin;
input roman;
end

```

	0	1	2	3	4	5	6	7
'000	Γ	Δ	⊕	Λ	Σ	Π	Σ	T
'010	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'020	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'030	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'040	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'050	(	)	⊕	+	⊕	⊕	⊕	/
'060	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'070	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'100	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'110	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'120	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'130	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'140	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'150	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'160	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
'170	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕

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 *Amp's Ceuves* is naive about the efficient preparation of *flavus soufés*. This is a sample of the font when the resolution is 200 dots per inch and 3.6 dots per "point".



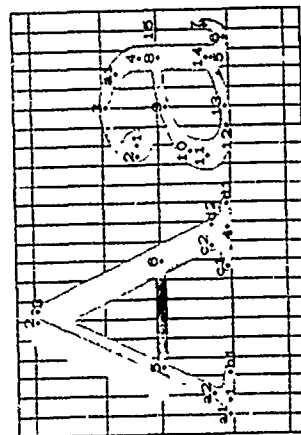
The file cmr6.mf

```
"Computer Modern Roman 6 point";
ph = 1.5; px = 1.5; pe = 1.5; pd = 1.5;
pb = 1.5; po = 1.5; ps = 1.5; pa = .5(ph - pd);
pw = 1.5; pwi = 1.5; pwil = 1.5; pwil = 1.5;
pwiv = 1.5; pws = 1.5; aspect = 1.0;
pu = 1.5; lcs = 89; ucs = 1.43; sc = 0; ls = 0;
slnt = 0; sqrttwo = sqrt 2; fixwidth = 0;
halfid = 0; varg = 0; lowast = 0; ligs = 1.

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

	0	1	2	3	4	5	6	7
'000	T	A	●	▲	■	■	Σ	Υ
'010	●	●	●	●	●	●	●	●
'020	●	●	●	●	●	●	●	●
'030	●	●	●	●	●	●	●	●
'040	●	●	●	●	●	●	●	●
'050	(	)	●	+	●	●	●	/
'060	●	1	2	3	4	5	6	7
'070	●	●	●	●	●	●	●	●
'100	●	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	●	●	●	●	●	●	●	●
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	●	●	●	●	●

Mathematical 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 *Comp's* *TeX* was in 1982 about the efficient preparation of American scientific papers. This is a sample of the font when the resolution is 300 dots per inch and 3.6 dots per "x" unit.



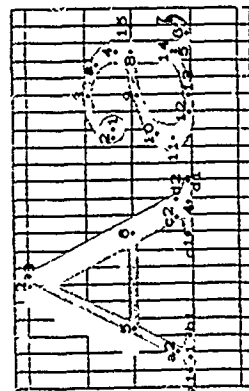
The file cmr5.mf

```
"Computer Modern Roman 5 point";
ph = 1/3; px = 3/8; pe = 3/8; pd = 3/8;
pb = 3/8; po = 3/8; ps = 3/8; pa = .5(ph - pd);
pw = 3/8; pwi = 3/8; pwil = 3/8; pwil = 3/8;
pwiv = 3/8; pwi = 3/8; aspect = 1.0;
pu = 1/3; 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	r	A	o	A	z	n	z	T
'010	o	o	n	o	z	.	.	.
'020	.	.	.	.	.	.	.	.
'030	.	.	.	.	.	.	.	.
'040	.	.	.	.	.	.	.	.
'050	(	)	.	+	.	.	.	/
'060	.	1	2	3	4	5	6	7
'070	.	.	.	.	.	.	.	.
'100	.	A	B	C	D	E	F	G
'110	.	I	J	K	L	M	N	O
'120	.	P	Q	R	S	T	U	V
'130	.	X	Y	Z	.	.	.	.
'140	.	.	.	.	.	.	.	.
'150	.	.	.	.	.	.	.	.
'160	.	.	.	.	.	.	.	.
'170	.	.	.	.	.	.	.	.

Mathematical books and journals do not look as beautiful as they used to. It is not that their mathematical content is worthless; rather that the old and cultured setting of the text is gone. The present volume is a collection of mathematical symbols that mathematicians (and you) can use to solve this problem. It is a collection of symbols that the first edition of *Computer Modern* had about the efficient preparation of scientific documents. This is a sample of the font when the resolution is 300 dots per inch and 3.5 dots per "point".



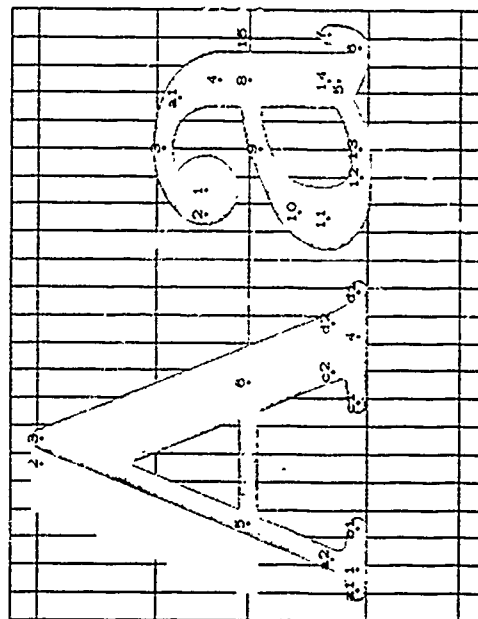
	0	1	2	3	4	5	6	7
'000	T	A	Q	A	E	II	'	T
'010	Q	Q	Q	Q	J	.	.	.
'020	.	.	.	.	.	.	.	.
'030	.	.	.	B	m	m	E	CE
'040	Q	I	.	.	oo	%	E	.
'050	(	)	.	+	,	.	.	/
'060	Q	1	2	3	4	5	6	7
'070	8	9	:	:	<	=	>	?
'100	Q	A	B	C	D	E	F	G
'110	H	I	J	K	L	M	N	O
'120	P	Q	R	S	T	U	V	W
'130	X	Y	Z	[	^	]	-	-
'140	.	a	b	c	d	e	f	g
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	{		~	~	~

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 *Lesep's Euvres* is naive about the efficient preparation of flawless soufflés. This is a sample of the font when the resolution is 300 dots per inch and 3.6 dots per "point".

T:\file emb10.mf

```
"Computer Modern Bold Roman 10 point";
ph = 48; px = 48; pe = 38; pd = 38;
pb = 38; po = 38; ps = 38; pa = 5(ph - pd);
pw = 38; pwi = 38; pwii = 38; pwiii = 38;
pwiv = 38; pwv = 38; aspect = 1.0;
pu = 38; lcs = 85; ucs = 1.5; sc = 0; ls = 0;
slant = 0; sqrtwo = 1.375; fixwidth = 0;
halfd = 0; varg = 0; lowast = 0; ligs = 1.
```

```
input embase; call fontbegin;
input roman;
end
```



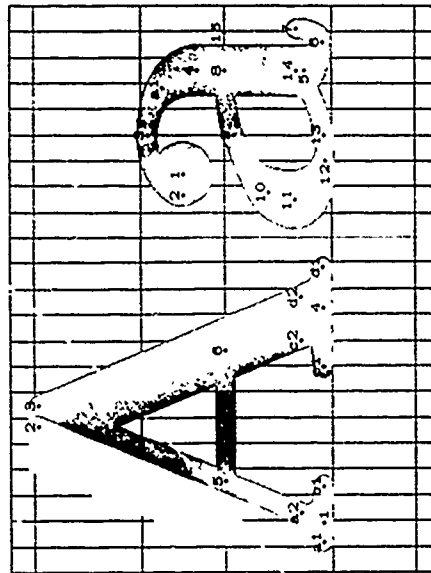
The file cmb9 mf

```
"Computer Modern Bold Roman 9 point";
ph = 225; px = 34; pe = 38; pd = 38;
pb = 18; po = 38; ps = 38; pa = .5(ph - pd);
pw = 43; pwi = 38; pwii = 38; pwiii = 38;
pwiv = 38; pwv = 38; aspect = 1.0;
pu = 18.5; l = .84; ucs = 1.46; 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	Δ	Θ	Α	Η	Π	Σ	Τ
'010	Φ	Ψ	Ω	ι	ϋ	.	.	.
'020	.	.	.	.	.	.	.	.
'030	.	.	.	Β	α	ο	Ε	Ϛ
'040	ρ	ι	ν	ι	οο	π	Δ	.
'050	(	)	•	+	,	.	.	/
'060	θ	ι	ζ	3	4	5	6	7
'070	8	9	:	:	<	=	>	?
'100	Θ	Α	Β	Γ	Δ	Ε	Ζ	Η
'110	Θ	Ι	Ζ	Κ	Λ	Μ	Ν	Ο
'120	Ρ	Q	Ρ	Σ	Τ	Υ	Φ	Ψ
'130	Χ	Υ	Ζ		.		-	-
'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	fl	fr	ss	m

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 Eep's *Euclid* is naive about the efficient preparation of flawless outlines. This is a sample of the font when the resolution is 200 dots per inch and 2.6 dots per "point".



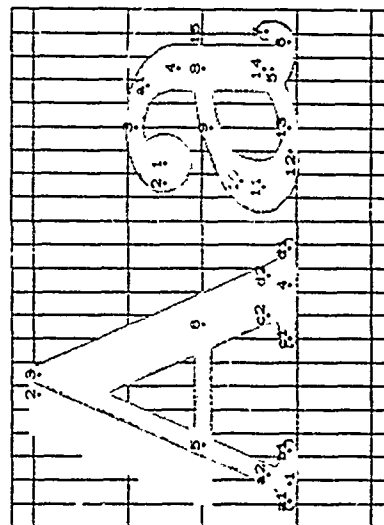
The file cmb8.mf

```

"Computer Modern Bold Roman 8 point";
ph = 30; px = 10; pc = 74; pd = 30;
pb = 30; po = 30; ps = 10; pa = 5(ph - pd);
pw = 30; pwi = 30; pwii = 30; pwiii = 30;
pwiv = 30; pwv = 30; aspect = 1.0;
pu = 30; lcs = 82; ucs = 1.41; sc = 0; ls = 0;
slant = 0; sqrttwo = 1.375; fixwidth = 0;
halfd = 0; varg = 0; lowast = 0; ligss = 1.

input cmbase; call fontbegin;
input roman;
end

```



	0	1	2	3	4	5	6	7
'000	T	A	Q	A	B	H	S	T
'010	Q	Q	Q	Q	Q	Q	Q	Q
'020	Q	Q	Q	Q	Q	Q	Q	Q
'030	Q	Q	Q	Q	Q	Q	Q	Q
'040	Q	Q	Q	Q	Q	Q	Q	Q
'050	(	)	Q	+	Q	Q	Q	/
'060	Q	Q	Q	Q	Q	Q	Q	Q
'070	Q	Q	Q	Q	Q	Q	Q	Q
'100	Q	A	B	Q	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	[	Q	Q	Q	Q
'140	Q	Q	Q	Q	Q	Q	Q	Q
'150	Q	Q	Q	Q	Q	Q	Q	Q
'160	Q	Q	Q	Q	Q	Q	Q	Q
'170	Q	Q	Q	Q	Q	Q	Q	Q

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 *Knop's Cnives* is alive about the efficient preparation of flawless combs. This is a sample of the font when the resolution is 300 dots per inch and 3.0 dots per "point".

```

"Computer Modern Slanted Roman 10 point",
ph = 238; px = 198; pe = 38; pd = 38;
pb = 38; po = 14; ps = 38; pa = 5(ph - pd);
pw = 38; pwi = 38; pwil = 38; pwil = 38;
pwiv = 38; pvv = 38; aspect = 1.0;
pu = 38; les = 1.075; ucs = 1.7; sc = 0; ls = 0;
slant = .15; sqrttwo = sqrt 2; fixwidth = 0;
halfid = 0; varg = 0; lowast = 0; ligs = 1.

input embase, call fontbegin;
input roman;
end

```

	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Α	Β	Π	Σ	Τ
'010	Φ	Ψ	Ω	Ι	Ζ	·	·	·
'020	·	·	·	·	·	·	·	·
'030	·	·	·	Β	Ξ	α	Æ	Œ
'040	θ	ι	·	·	∞	%	&	·
'050	(	)	*	+	,	-	·	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	;	<	=	>	?
'100	Ø	Α	Β	Γ	Δ	Ε	Ζ	Η
'110	Θ	Ι	Ζ	Κ	Λ	Μ	Ν	Ξ
'120	Π	Ρ	Σ	Τ	Υ	Φ	Ψ	Ω
'130	Χ	Υ	Ζ	Ι	·	Ι	-	-
'140	·	α	β	γ	δ	ε	ζ	η
'150	θ	ι	κ	λ	μ	ν	ξ	ο
'160	π	ρ	σ	τ	υ	φ	ψ	ω
'170	α	β	γ	δ	ε	ζ	η	θ

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 cmsa mf

```
"Computer Modern Slanted Roman 9 font";
px = 23; pe = 13; pd = 9;
pb = 3; po = 3; ps = 15; pa = 5(ph - pd);
pw = 3; pwi = 3; pwi = 3; pwii = 3;
pwiv = 3; pwv = 3; ispec = 1.0;
pu = 19; lcs = 1.05; ias = 1.65; sc = 0; ls = 6;
slant = .15; spttwo = - rt 2, fixwidth = 0;
halfd = 0; varg = 0; lowast = 0; lig = 1.
```

```
input , mbase; call fontbegin;
input roman;
end
```

	0	1	2	3	4	5	6	7
'000	T	A	Q	4	B	H	Σ	T
'010	Φ	W	Ω	1	J	.	.	.
'020	.	.	.	.	.	.	.	.
'030	.	.	.	B	æ	æ	Æ	Æ
'040	μ	!	*	!	∞	%	£	.
'050	{	)	+	+	,	.	.	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	:	<	=	>	!
'100	6	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	[	"	l	-	-
'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	£	£	£	£	£

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 Esop's *Œuvres* is naive 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 cma8 mf

```

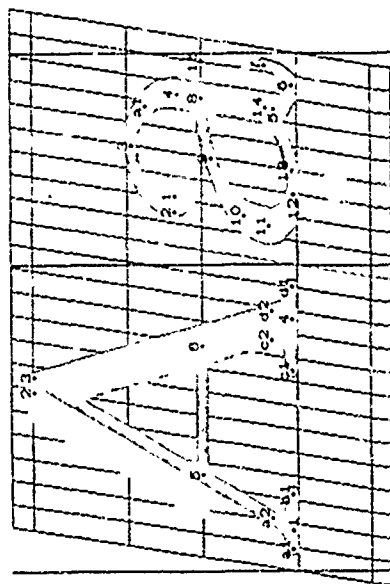
"Computer Modern Slanted Roman 8 point";
ph = 200; px = 123; pc = 123; pd = 38;
pb = 110; po = 38; ps = 38; pa = 5(ph -- pd);
pw = 38; pwi = 38; pwh = 38; pwin = 38;
pwiv = 38; pwv = 38; aspect = 10;
pu = 110; lcs = 1.03; ucs = 1.56; ac = 0; ls = 0;
slant = .15; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 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	Β	Ε	Σ	Υ
'010	Φ	Ψ	Ω	ι	ι	.	.	.
'020	.	.	.	.	.	.	.	.
'030	.	.	.	Β	Β	Β	Β	Β
'040	δ	ι	π	ι	∞	%	Δ	.
'050	(	)	°	+	,	.	.	/
'060	Θ	Ι	Ζ	3	4	5	6	7
'070	8	9	:	:	<	=	>	?
'100	Φ	Α	Β	Γ	Δ	Ε	Ζ	Η
'110	Θ	Ι	Ζ	Κ	Λ	Μ	Ν	Ξ
'120	Ρ	Q	Ρ	Σ	Τ	Υ	Φ	Ψ
'130	Χ	Υ	Ζ	Ι	Β	Γ	Δ	Ε
'140	.	α	β	γ	δ	ε	ζ	η
'150	θ	ι	κ	λ	μ	ν	ξ	ο
'160	π	ρ	σ	τ	υ	φ	χ	ψ
'170	ω	γ	ζ	η	θ	ι	κ	λ

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 *TeX*'s *Computer Modern* is naive 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".



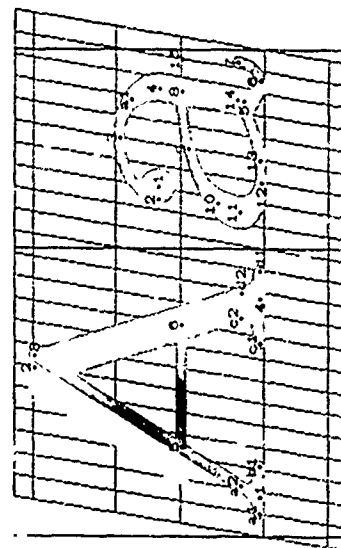
The file cms7.mf

```
"Computer Modern Slanted Roman 7 point";
ph = 1/2; px = 1/2; pe = 1/2; pd = 1/2;
pb = 1/2; po = 1/2; ps = 1/2; pa = .5(ph - pd);
pw = 1/2; pwi = 1/2; pwt = 1/2; pwii = 1/2;
pwiv = 1/2; pwv = 1/2; aspect = 1.0;
pu = 1/2; lcs = 97; ucs = 1.44; sc = 0; ls = 0;
slant = .15; sqrttwo = sqrt 2; flxwidth = 0;
halfd = 0; varg = 0; lowast = 0; ligs = 1.
```

```
input cmbase; call fontbegin;
input rman;
end
```

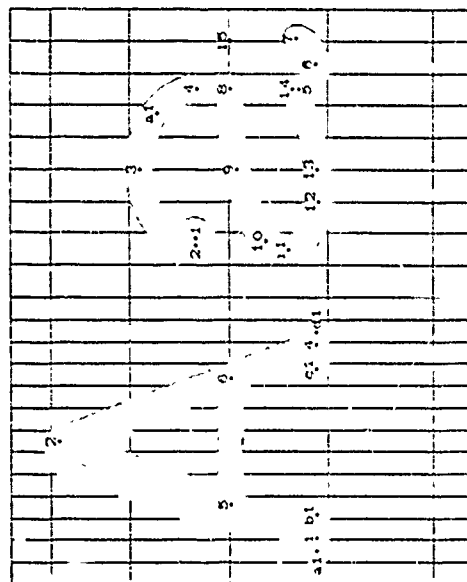
	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Τ
'010	ϕ	ϕ	η	ι	ϰ	·	·	·
'020	·	·	·	·	·	·	·	·
'030	·	·	·	Β	Ξ	·	Æ	Æ
'040	·	ι	·	ι	∞	·	·	·
'050	(	)	·	+	·	·	·	/
'060	θ	ι	ζ	ζ	4	5	6	7
'070	θ	υ	·	·	·	·	·	·
'100	θ	Α	Β	Γ	Δ	Ε	Ζ	Η
'110	θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
'120	Ρ	Q	Ρ	Σ	Τ	Υ	Φ	Ψ
'130	Χ	Υ	Ζ	Ι	·	·	·	·
'140	·	·	·	·	·	·	·	·
'150	·	·	·	·	·	·	·	·
'160	·	·	·	·	·	·	·	·
'170	·	·	·	·	·	·	·	·

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 *TeX*'s *TeX* is naive about the efficient preparation of fluted columns. This is a sample of the font when the resolution is 200 dots per inch and 3.6 dots per "point"



	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Α	Ε	Π	Σ	Τ
'010	Φ	Ψ	Ω	Ι	Ζ	-	-	-
'020	-	-	-	-	-	•	•	•
'030	•	→	-	Ο	•	•	Ε	Ε
'040	Π	!	"	#	\$	%	Δ	•
'050	(	)	*	+	,	-	•	/
'060	Ο	1	2	3	4	5	6	7
'070	8	9	:	:	<	=	>	?
'100	•	Α	Β	Γ	Δ	Ε	Ε	Γ
'110	Η	Ι	Ζ	Κ	Λ	Μ	Ν	Ο
'120	Ρ	Q	Ρ	Σ	Τ	Υ	Υ	Ψ
'130	Χ	Υ	Ζ	[	\	]	↑	+
'140	•	α	β	γ	δ	•	ε	ζ
'150	η	ι	ζ	κ	ι	μ	ν	ο
'160	ρ	q	ρ	σ	τ	υ	φ	ψ
'170	χ	υ	z	{		↑	}	,

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 Knop's *Knives* is naive 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".

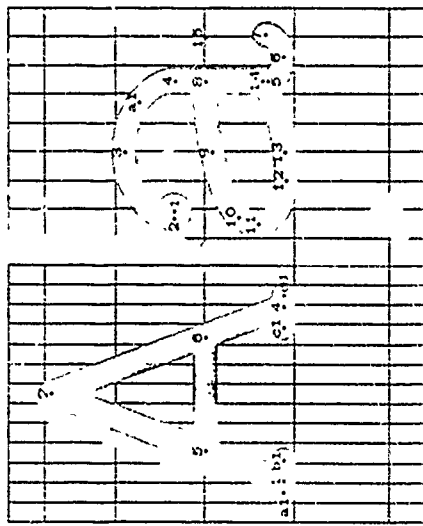


	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Λ	Β	Π	Σ	Υ
'010	Φ	Ψ	Ω	Ι	Ζ	·	·	·
'020	·	·	·	·	·	·	·	·
'030	·	·	·	·	·	·	·	·
'040	Λ	Ι	·	·	·	·	·	·
'050	(	)	·	·	·	·	·	·
'060	Ο	Ι	2	3	4	5	6	7
'070	8	9	:	:	<	=	>	?
'100	Θ	Α	Β	Γ	Δ	Ε	Ζ	Η
'110	Θ	Ι	Ζ	Κ	Λ	Μ	Ν	Ξ
'120	Ρ	Q	Ρ	Σ	Τ	Υ	Φ	Ψ
'130	Χ	Υ	Ζ	[	\	]	†	+
'140	·	·	·	·	·	·	·	·
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	{		~	}	,

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 Asop's Duves is naive about the efficient preparation of flawless souffles. This is a sample of the font when the resolution is 200 dots per inch and 3.0 dots per "point".

The file emtt9 mf

"Computer Modern Typewriter Type for use with 9 point",
 ph = 10, px = 10, pe = 10, pd = 10,
 pb = 10, po = 10, ps = 10, pa = 10, ph = 10,
 pw = 10, pwi = 10, pwu = 10, pwi = 10,
 pwv = 10, pwv = 10, aspect = 10,
 pu = 10, lcs = 11, ucs = 11, sc = 0, ls = 0,
 sl = 0, sqrt2 = sqrt 2, fixwidth = 1,
 halld = 1, varg = 0; lowast = 1, ligs = 0.
 input embase; call fontbegin;
 input roman;
 end

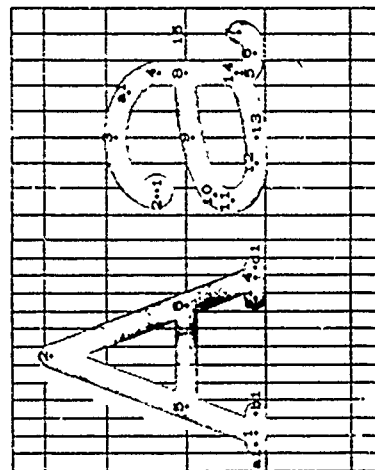


	0	1	2	3	4	5	6	7
'000	r	A	B	A	E	n	E	r
'010	o	o	o	1	j	-	-	-
'020	-	-	-	-	-	-	o	-
'030	.	+	-	B	o	o	A	u
'040	u	i	o	o	o	g	A	.
'050	(	)	o	+	.	-	.	/
'060	o	1	2	3	4	5	o	7
'070	o	o	:	:	<	=	>	?
'100	o	A	B	C	D	E	F	o
'110	H	I	J	K	L	M	N	o
'120	P	Q	R	S	T	U	V	W
'130	X	Y	Z	[	\	]	†	+
'140	.	a	b	c	d	e	f	E
'150	a	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	{		↓	}	.

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 Asap's curves is naive about the efficient preparation of flawless souffles. This is a sample of the font when the resolution is 300 dots per inch and 3.0 dots per .point...

The file cmtt8.mf

"Computer Modern Typewriter Type for use with 8 point";
ph = 108; px = 120; pe = 90; pd = 94;
pb = 108; po = 108; ps = 108; pa = 108;
pw = 108; pwii = 108; pwiii = 108;
pwiv = 108; pwv = 108; aspect = 1.0;
pu = 108; lcs = 108; ucs = 108; sc = 0; ls = 0;
slant = 0; sqrttwo = sqrt 2; fixwidth = 1;
halfd = 1; varg = 0; low-st = 1; ligs = 0.
input cmbase; call font-egin;
input roman;
end

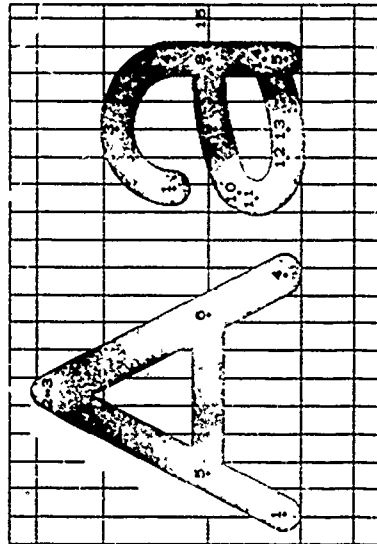


```
"Computer Modern Sans Serif Quotation 8 point";
ph = 28; px = 18; pe = 38; pd = 38;
pb = 38; po = 38; ps = 38; pa = .5(pt - pd);
pw = pwi = pwii = pwiii = 38;
pwiv = pwv = 38; aspect = 1.0;
pu = 38; fcs = 0; ucs = 0; sc = 1; ls = 0;
slant = 0; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 0; varg = 1; lowast = 0; ligs = 1.

input embase; call fontbegin;
input roman;
end
```

	0	1	2	3	4	5	6	7
'000	Γ	Δ	●	Λ	≡	Π	Σ	Τ
'010	●	ψ	Ω	ι	ι	·	·	·
'020	·	·	·	·	·	·	·	·
'030	·	·	·	·	·	·	·	·
'040	·	·	·	·	·	·	·	·
'050	(	)	·	+	·	·	·	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	:	<	=	>	?
'100	⊙	Α	Β	Γ	Δ	Ε	Ζ	Η
'110	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
'120	Ρ	Σ	Τ	Υ	Φ	Ψ	Ω	·
'130	×	Υ	Ζ	Ι	·	·	·	·
'140	·	·	·	·	·	·	·	·
'150	·	·	·	·	·	·	·	·
'160	·	·	·	·	·	·	·	·
'170	·	·	·	·	·	·	·	·

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 *AESOP's CUVRES* is naive about the efficient preparation of flawless sources. This is a sample of the font when the resolution is 200 dots per inch and 3.6 dots per "point".



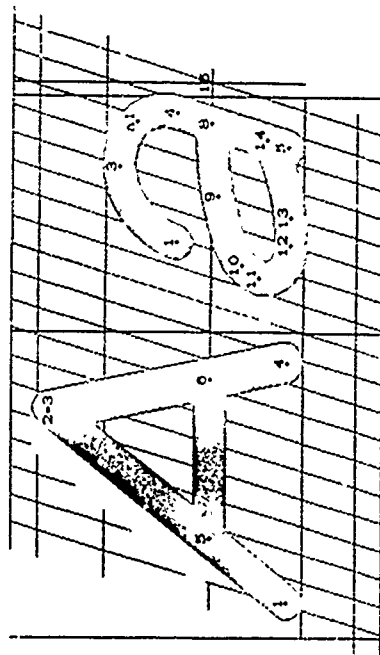
	0	1	2	3	4	5	6	7
'000	r	Δ	●	A	Ξ	Π	Σ	Υ
'010	Φ	ψ	Ω	ι	ι	·	·	·
'020	·	·	·	·	·	·	·	·
'030	·	·	·	B	∞	∞	∞	∞
'040	·	·	·	·	∞	∞	∞	·
'050	(	)	+	+	·	·	·	/
'060	0	1	2	3	4	5	6	7
'070	·	·	·	·	·	·	·	·
'100	·	A	B	C	D	E	F	G
'110	H	I	J	K	L	M	N	O
'120	P	Q	R	S	T	U	V	W
'130	X	Y	Z	I	·	I	·	·
'140	·	·	·	·	·	·	·	·
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	π	π	π	π	π

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 *Euclid's Elements* is naive about the efficient preparation of new typeset fonts. This is a sample of the font when the resolution is 200 dots per inch and 3.8 dots per "point".

The file cm3:s8 m2

"Computer Modern Slanted Sans Serif Quotation 8 point";
 ph = 48; px = 48; pe = 48; nd = 48;
 pb = 48; po = 48; ps = 48; pa = 5(ph - pd);
 pw = 48; pwi = 48; pwil = 48; pwil = 48;
 pwiv = 48; pvv = 48; aspect = 1 0;
 pu = 48; lcs = 0; ucs = 0; sc = 1; ls = 0;
 slant = 1; sqrttwo = sqrt 2; fixwidth = 0;
 halfd = 0; varg = 1; lowast = 0; ligs = 1.

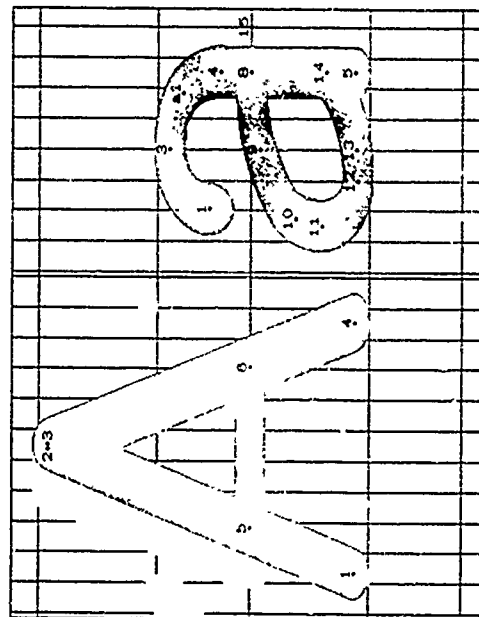
input embase; call fontbegin;
 input roman;
 end



The file cmsb.mf

```
"Computer Modern Sans Serif 10 point Bold Extended";
ph = 1/10; px = 1/10; pe = 1/10; pd = 1/10;
pb = 1/10; po = 1/10; ps = 1/10; pa = .5(ph - pd);
pw = pwi = pwil = pwil = 1/10;
pwiv = pwv = 1/10; aspect = 1/10;
pu = 1/10; lcs = 0; ucs = 0; sc = 1/2; ls = 0;
slant = 0; sqrtwo = 1.35; fixwidth = 0;
halfd = 0; varg = 0; lowast = 0; ligs = 1.

input embase, call fontbegin,
input roman;
end
```



	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Τ
'010	Φ	Ψ	Ω	ι	ϋ	·	·	·
'020	·	·	·	·	·	·	·	·
'030	·	·	·	·	·	·	·	·
'040	ϐ	ι	·	·	·	·	·	·
'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	·	·	·	·	·	·	·	·
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	{		~	~	~

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 *Esop's Œuvres* is naive 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".

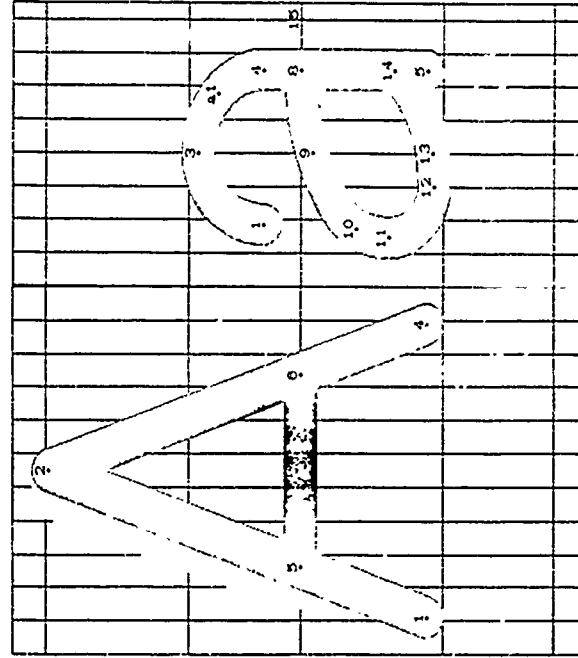
	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ
'010	Φ	Ψ	Ω	ι	ϋ	Ϙ	ϙ	Ϛ
'020	ϛ	Ϝ	ϝ	Ϟ	ϟ	Ϡ	ϡ	Ϣ
'030	ϣ	Ϥ	ϥ	Ϧ	ϧ	Ϩ	ϩ	Ϫ
'040	ϫ	Ϭ	ϭ	Ϯ	ϯ	ϰ	ϱ	ϲ
'050	(	)	*	+	,	-	.	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	;	<	=	>	?
'100	∅	A	B	C	D	E	F	G
'110	H	I	J	K	L	M	N	O
'120	P	Q	R	S	T	U	V	W
'130	X	Y	Z	[	"	]	-	—
'140	'	a	b	c	d	e	f	g
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	ff	fi	fl	ff	ff

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 cmss12 mf

```
"Computer Modern Sans Serif 12 point";
ph = 40, px = 10, pe = 108, pd = 30;
pb = 30, po = 10, ps = 30, pa = .5(ph - pd);
pw = pw1 = pw2 = pw3 = 10;
pwiv = pwv = 10, aspect = 8;
uc = 30, lcs = 0, ucs = 0, sc = 5, ls = 0;
shat = 0, sqrtwo = sqrt 2, fixwidth = 0,
halfd = 0, varg = 0, lowast = 0, ligs = 1.

input embase, call fontbegin,
input roman;
end
```



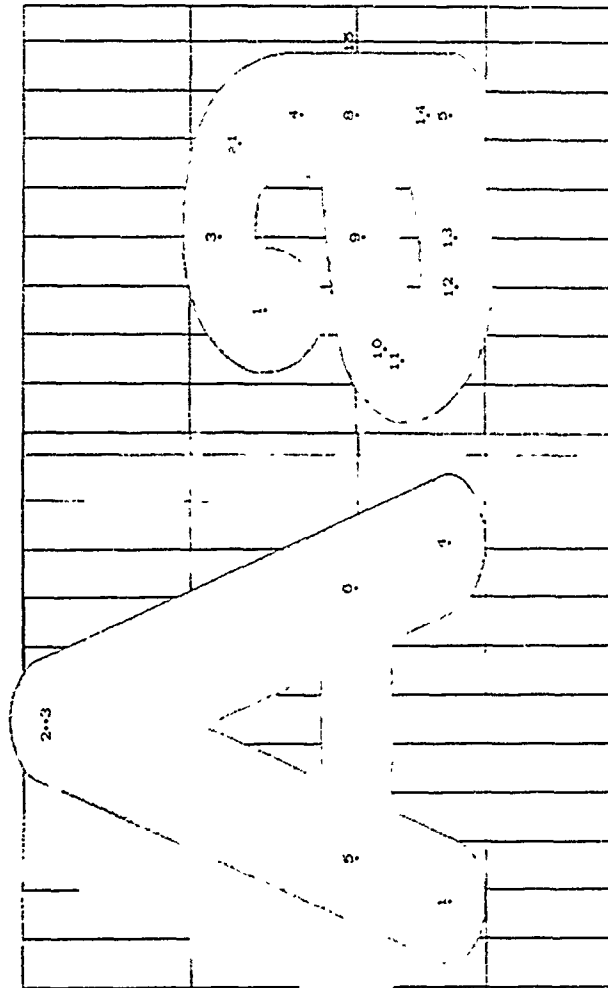
	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Α	≡	Π	Σ	Υ
'010	●	ψ	Ω	ι	Ј	•	•	•
'020	•	•	•	•	•	•	•	•
'030	•	•	•	•	•	•	•	•
'040	•	•	•	•	•	•	•	•
'050	(	)	•	+	,	•	•	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	;	<	=	>	?
'100	Q	A	B	C	D	E	F	G
'110	H	I	J	K	L	M	N	O
'120	P	Q	R	S	T	U	V	W
'130	X	Y	Z	[	α	Ј	-	—
'140	•	•	•	•	•	•	•	•
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	ff	ff	ff	ff	ff

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 *Aesop's Œuvres* is naïve about the efficient preparation of flawless *œuvres*. This is a sample of the font when the resolution is 200 dots per inch and 3.6 dots per "point".

'T:: file cmittl mf

"Computer Modern Sans Serif Extrabold 14 point";
 ph = 330; px = 424; pe = 38; pd = 38;
 pb = 0; po = 18; ps = 38; pa = 5(ph - pd);
 pw = 38; pwi = 38; pwil = 38; pwiii = 38;
 pwiv = 38; pwv = 38; aspect = 83;
 pu = 38; lcs = 0; ucs = 0; sc = 12; ls = 0;
 slant = 0; sqrtwo = 1.3; fixwidth = 0;
 halfd = 0; varg = 1; lowast = 0; ligs = 1.

input cmbase; call fontbegin;
 input roman;
 end



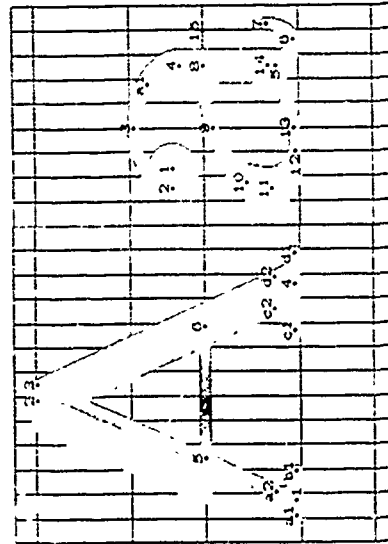
	0	1	2	3	4	5	6	7
'000	F	A	Θ	A	B	Π	Σ	T
'010	Φ	Ψ	Ω	1	J	.	.	.
'020	.	.	.	.	.	.	.	.
'030	.	.	.	B	∞	∞	Æ	OE
'040	θ	1	.	.	∞	%	£	.
'050	(	)	•	+	.	.	.	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	:	<	=	>	?
'100	Ø	A	B	C	D	E	F	G
'110	H	I	J	K	L	M	N	O
'120	P	Q	R	S	T	U	V	W
'130	X	Y	Z	I	.	I	-	-
'140	.	a	b	c	d	e	f	g
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	z	z	z	z	z

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 typesetters 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 *Æscop's* *Œuvres* is naive about the efficient preparation of flawless soufflés. This is a sample of the font when the resolution is 300 dots per inch and 2.6 dots per "point".

The file emsc10.mf

```
"Computer Modern Roman Small Caps for 10 point";
ph = 20; px = 18; pc = 18; pd = 18;
pb = 18; po = 20; ps = 18; pa = 5(pl - pd);
pw = 30; pwi = 30; pwv = 30; pwin = 30;
pwiv = 30; pwi = 30; aspect = 1.0;
pu = 18; lcs = 1; ucs = 1.23; sc = 0; ls = 0;
s'int = 0; sqrttwo = 1.35; fixwidth = 0;
halfid = 0; varg = 0; lowast = 0; ligs = 1.
```

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



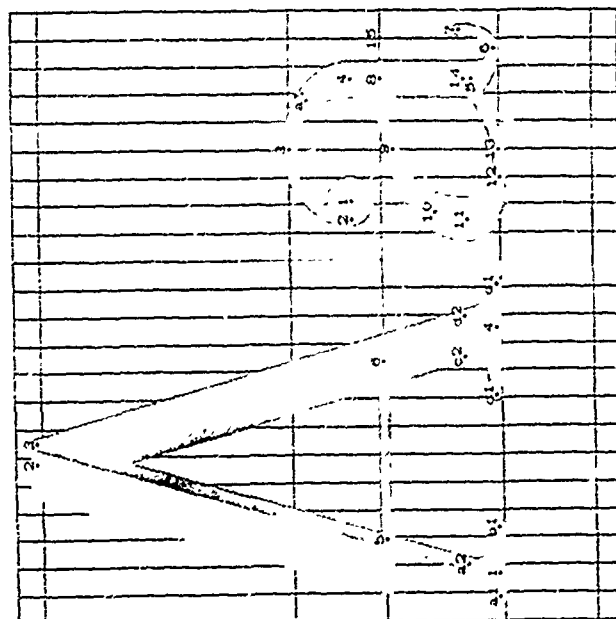
	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ
'010	Φ	Ψ	Ω	ι	ϋ	Ϙ	ϙ	Ϛ
'020	ϛ	Ϝ	ϝ	Ϟ	ϟ	Ϡ	ϡ	Ϣ
'030	ϣ	Ϥ	ϥ	Ϧ	ϧ	Ϩ	ϩ	Ϫ
'040	ϫ	Ϭ	ϭ	Ϯ	ϯ	ϰ	ϱ	ϲ
'050	(	)	*	+	,	-	.	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	;	<	=	>	?
'100	Ø	A	B	C	D	E	F	G
'110	H	I	J	K	L	M	N	O
'120	P	Q	R	S	T	U	V	W
'130	X	Y	Z	[	^	]	-	_
'140	'	a	b	c	d	e	f	g
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	x	y	z	ff	fi	fl	ff	ff

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 Euclides* 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 emdunh mf

"Computer Modern Dunhill 10 point";
 ph = 100; px = 100; pe = 100; pd = 100;
 pb = 100; po = 100; ps = 100; pa = 5(200 - pd);
 pw = 100; pwi = 100; pwil = 100; pwilw = 100;
 pwiv = 100; pwv = 100; aspect = 1;
 pu = 100; lcs = 1.075; ucs = 1.7; sc = 0; ls = 0;
 slant = 0; sqrttwo = sqrt 2; fixwidth = 0;
 halfid = 0; varg = 0; lowast = 0; ligs = 1.

input cmbase, call fontbegin;
 input roman;
 end



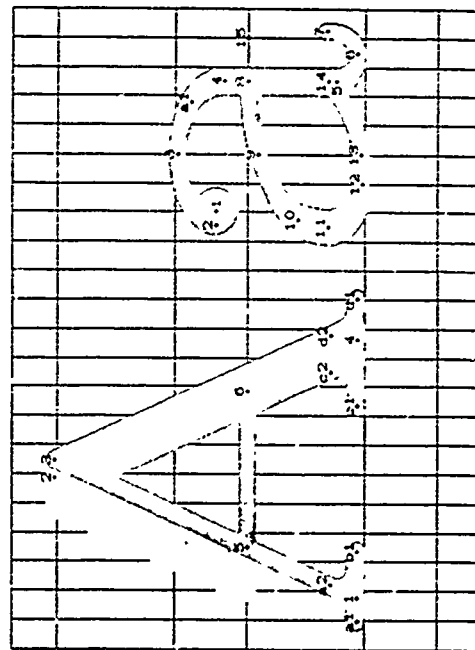
	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Α	Η	Σ	Τ	
'010	⊙	Ψ	Ω	Ι	Ζ	·	·	·
'020	·	·	·	·	·	·	·	·
'030	·	·	·	Β	Ξ	Ε	Θ	·
'040	⊙	Ι	·	·	∞	%	Δ	·
'050	(	)	·	+	·	·	·	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	:	<	=	>	?
'100	φ	Α	Β	Ο	Δ	Ε	Ζ	Γ
'110	Η	Ι	Ζ	Κ	Λ	Μ	Ν	Ο
'120	Ρ	Q	Ρ	Σ	Τ	Υ	Ψ	Ω
'130	Χ	Υ	Ζ		·	·	·	·
'140	·	·	·	·	·	·	·	·
'150	·	·	·	·	·	·	·	·
'160	·	·	·	·	·	·	·	·
'170	·	·	·	·	·	·	·	·

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 *Æoep's* *Œuvres* is naive about the efficient preparation of flawless *œuvres*. This is a sample of the font when the resolution is 200 dots per inch and 3.6 dots per "point".

The file cmf.1.b mf

"Computer Modern Fibonacci 10 point";
 ph = 30; px = 30; pe = 30; pd = 30;
 pb = 30; po = 30; ps = 30; pa = 5(ph - pd);
 pw = 30; pwi = 1; pwii = 30; pwiii = 30;
 pwiv = 30; pwv = 30; aspect = 1;
 pu = 30; lcs = 1; ucs = 161803; sc = 0; ls = 0;
 slant = 0; sqrtwo = 38197; flxwidth = 0;
 halfd = 0; varg = 1; towast = 0; lgs = 1.

input cmbase call fontbegin;
 input roman;
 end



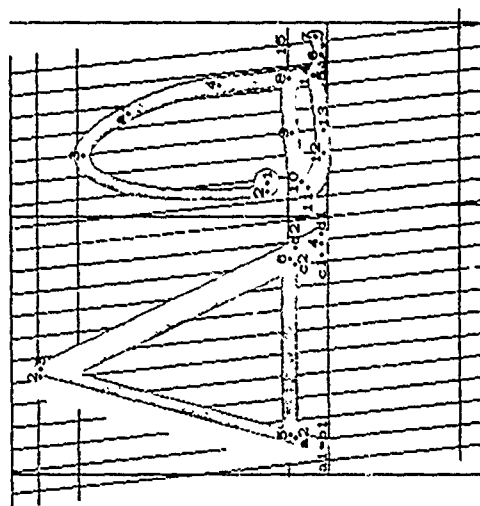
	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Τ
'010	Φ	Ψ	Ω	Ι	Ј	-	-	-
'020	-	-	-	-	-	-	-	-
'030	ρ	ι	-	β	α	α	Æ	Œ
'040	φ	ι	-	ι	∞	%	ℤ	·
'050	(	)	*	+	,	-	-	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	;	<	=	>	?
'100	Ø	Α	Β	Γ	Δ	Ε	Ζ	Η
'110	Θ	Ι	Ј	Κ	Λ	Μ	Ν	Ο
'120	Ρ	Σ	Τ	Υ	Φ	Ψ	Ω	Α
'130	Χ	Υ	Ζ	Ι	-	Ј	-	-
'140	-	Α	Β	Γ	Δ	Ε	Ζ	Η
'150	Θ	Ι	Ј	Κ	Λ	Μ	Ν	Ο
'160	Ρ	Σ	Τ	Υ	Φ	Ψ	Ω	Α
'170	Χ	Υ	Ζ	Ι	Π	Ο	Μ	Ν

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 *Æscop's* *Œuvre* is naive about the efficient preparation of flawless outlines. This is a sample of the font when the resolution is 200 dots per inch and 3.6 dots per point.

The file *cmff.mf*

```
"Computer Modern Funny Font";
ph = 28; px = 18; pc = 38; pd = 48;
pb = 38; po = 38; ps = 38; pa = 5(ph - pd);
pw = 38; pwi = 38; pwii = 48; pwiii = 38;
pwiv = 38; pwv = 38; aspect = 1.5;
pu = 18; lcs = 18; ucs = 18; sc = 2; ls = 0;
slant = -.1; sqrtwo = 1.5; fixwidth = 0;
halfd = 0; varg = 0; lowast = 0; ligs = 1.
```

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



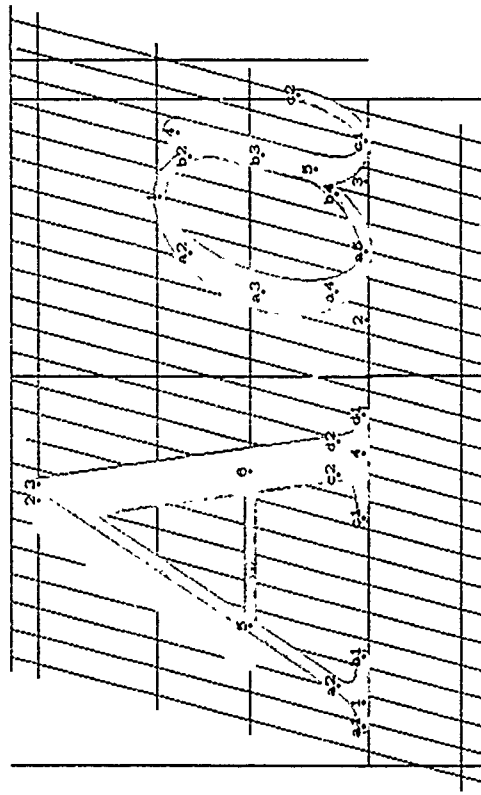
	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Α	Ε	Η	Σ	Τ
'010	Φ	Ψ	Ω	ι	ζ	·	·	·
'020	·	·	·	·	·	·	·	·
'030	·	·	·	β	α	α	Ε	Ε
'040	·	ι	·	·	·	%	θ	·
'050	(	)	*	+	·	·	·	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	;	<	=	>	?
'100	Ø	A	B	C	D	E	F	G
'110	H	I	J	K	L	M	N	O
'120	P	Q	R	S	T	U	V	W
'130	X	Y	Z	I	"	I	-	-
'140	'	a	b	c	d	e	f	g
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	z	y	x	ff	ff	ff	ff	ff

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 *Esop'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 cmbt10.mf

```
"Computer Modern Text Italic 10 point";
ph = 230; px = 180; pe = 20; pd = 20;
pb = 20; po = 10; ps = 20; pa = 5(ph - pd);
pw = 20; pwi = 20; pwil = 20; pwil = 20;
pwiv = 20; pvv = 20; aspect = 1.0;
pr = 20; lcs = 1.075; scs = 1.68; sc = 0; ls = .5;
slant = .25; sqrttwo = sqrt 2; fixwidth = 0;
halfld = 0; lowast = 0; ligas = 1; mi = 0.
```

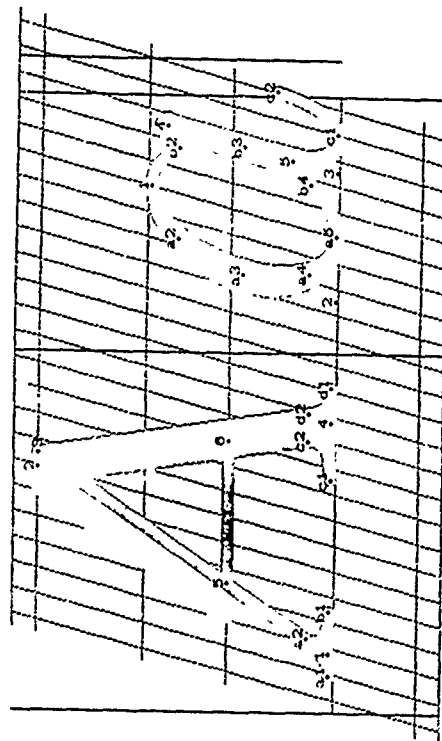
```
input cmbase; call fontbegin;
input italic;
end
```



"Computer Modern Text Italic 9 point";
 ph = 12; px = 1/10; pe = 1/10; pd = 1/10;
 pb = 1/10; po = 1/10; ps = 1/10; pa = 5(ph - pd);
 pw = 1/10; pwi = 1/10; pwii = 1/10; pwiii = 1/10;
 pwiv = 1/10; pwv = 1/10; aspect = 1.0;
 pu = 1/10; les = 1.054; ura = 1.65; sc = 0; ls = 9/18.5;
 slant = .25; sqrttwo = sqrt 2; fixwidth = 0;
 halfd = 0; lowast = 0; ligs = 1; mi = 0.
 input embase; call fontbegin;
 input italic;
 end

	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Α	Β	Π	Σ	Υ
'010	Φ	Ψ	Ω	ι	ϰ	·	·	·
'020	·	·	·	·	·	·	·	·
'030	·	·	·	·	·	·	·	·
'040	·	·	·	·	·	·	·	·
'050	(	)	*	+	·	·	·	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	;	<	=	>	?
'100	Ø	Α	Β	Γ	Δ	Ε	Ζ	Η
'110	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
'120	Ρ	Σ	Τ	Υ	Φ	Ψ	Ω	·
'130	Χ	Υ	Ζ	Ι	·	·	·	·
'140	·	α	β	γ	δ	ε	ζ	η
'150	θ	ι	κ	λ	μ	ν	ξ	ο
'160	π	ρ	σ	τ	υ	φ	ψ	ω
'170	z	y	x	w	v	u	t	s

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 *Esop's Quares* is naive about the efficient preparation of *flowless souffles*. This is a sample of the font when the resolution is 200 dots per inch and 3.5 dots per "point".



	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Α	Ε	Η	Σ	Υ
'010	Φ	Ψ	Ω	ΰ	ϰ	·	·	·
'020	·	·	·	·	·	·	·	·
'030	·	·	·	·	·	·	·	·
'040	·	·	·	·	·	·	·	·
'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	·	·	·	·	·	·	·	·
'150	h	i	j	k	l	m	n	o
'160	p	q	r	s	t	u	v	w
'170	z	y	z	ff	fl	fl	fl	fl

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 *Euclid's Elements* is naive about the efficient preparation of flawless scrolls. This is a sample of the font when the resolution is 200 dots per inch and 3.6 dots per "point".

The file cmu10.mf

```

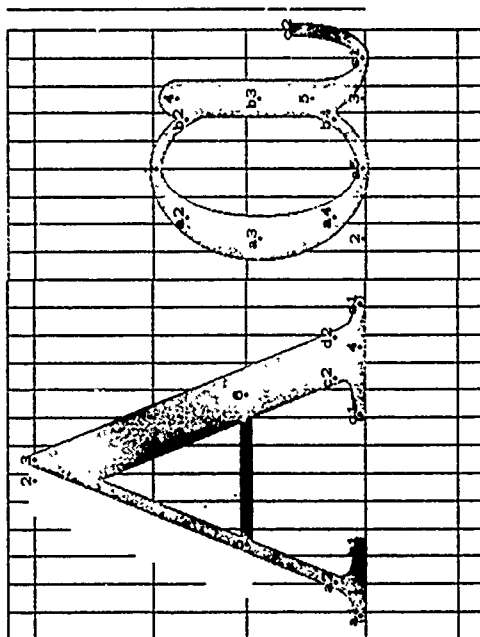
"Computer Modern Unslanted Italic 10 point";
ph = 20; px = 10; pe = 30; pd = 30;
pb = 30; po = 30; ps = 30; pa = .5(ph - pd);
pw = 30; pwi = 30; pwii = 30; pwiii = 30;
pwiv = 30; pwv = 30; aspect = 1.0;
pu = 30; lcs = 1.075; ucs = 1.7; sc = 0; ls = 0;
slant = 0; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 0; lowast = 0; ligs = 1; mi = 0.

```

```

input embase; call fontbegin;
input italic;
end

```



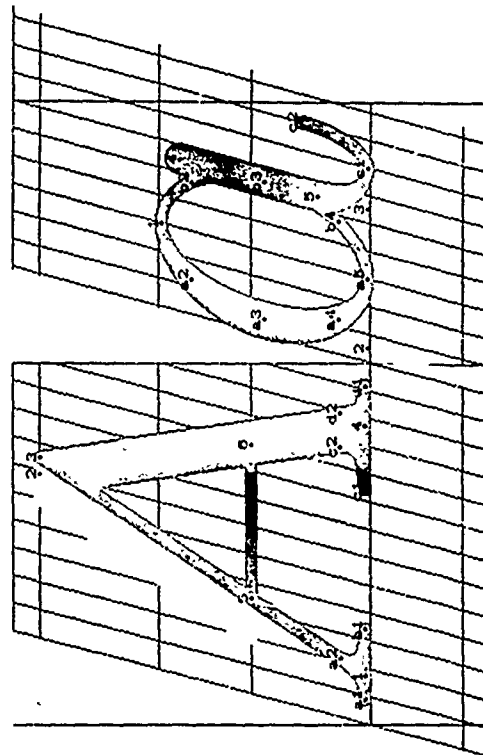
The file cmi10.mf

```
"Computer Modern Math Italic 10 point";
ph = 38; px = 38; pe = 38; pd = 38;
pb = 38; po = 38; ps = 38; pa = 38;
pw = 38; pwi = 38; pwii = 38; pwiii = 38;
pwiv = 38; pwv = 38; aspect = 1.0;
pu = 38; lcs = 1.075; ucs = 1.68; sc = 0; ls = 0;
slant = .25; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 0; lowast = 0; ligs = 0; mi = 1.
```

```
input cmbase; call fontbegin;
input italic;
end
```

	0	1	2	3	4	5	6	7
'000	Γ	Δ	Θ	Α	Β	Π	Σ	Τ
'010	Φ	Ψ	Ω	α	β	γ	δ	ε
'020	ζ	η	θ	ι	κ	λ	μ	ν
'030	ξ	π	ρ	σ	τ	υ	φ	χ
'040	ς	ι	ρ	ι	ρ	θ	θ	ι
'050	(	)	*	+	,	-	.	/
'060	0	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	O
'120	P	Q	R	S	T	U	V	W
'130	X	Y	Z	I		I		
'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	φ	ω	φ	θ	ω

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 *φσop's Xvures* is naive 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".

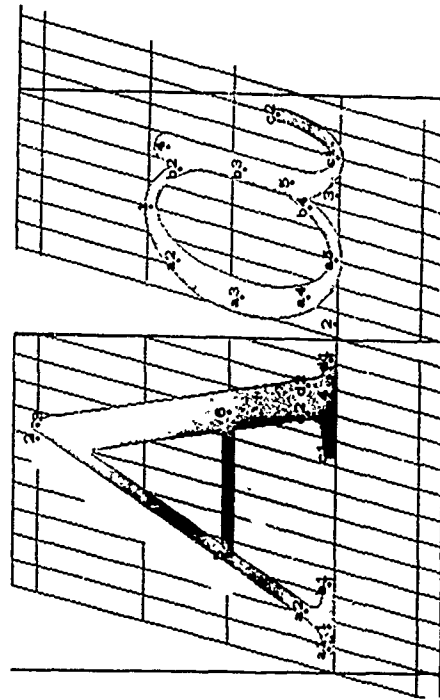


The file cm19.mf

```
"Computer Modern Math Italic 9 point";
ph = 22; px = 14; pe = 31; pd = 33;
pb = 38; po = 30; ps = 48; pa = .5(ph - pd);
pw = 36; pwi = 38; pwii = 36; pwiii = 36;
pwiv = 36; pwv = 38; aspect = 1.0;
pu = 18; lcs = 1.054; ucs = 1.65; sc = 0; ls = 0;
slant = .25; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 0; lowast = 0; ligs = 0; mi = 1.
input embase; call fontbegin;
input italic;
end
```

	0	1	2	3	4	5	6	7
'000	T	A	G	A	B	H	E	T
'010	Φ	Ψ	Ω	α	β	γ	δ	ε
'020	ζ	η	θ	ι	κ	λ	μ	ν
'030	ξ	π	ρ	σ	τ	υ	φ	χ
'040	ς	ι	ο	ι	ρ	θ	θ	,
'050	(	)	*	+	,	-	.	/
'060	0	1	2	3	4	5	6	7
'070	8	9	:	;	<	=	>	?
'100	J	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	I		I		
'140	a	b	c	d	e	f	g	h
'150	i	j	k	l	m	n	o	p
'160	q	r	s	t	u	v	w	x
'170	y	z	z	φ	ω	φ	φ	ω

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 *TeX*'s X_YTeX is named about the efficient preparation of flawless soufflé. This is a sample of the font when the resolution is 800 dots per inch and 3.6 dots per "point".



The file cm18.mf

```

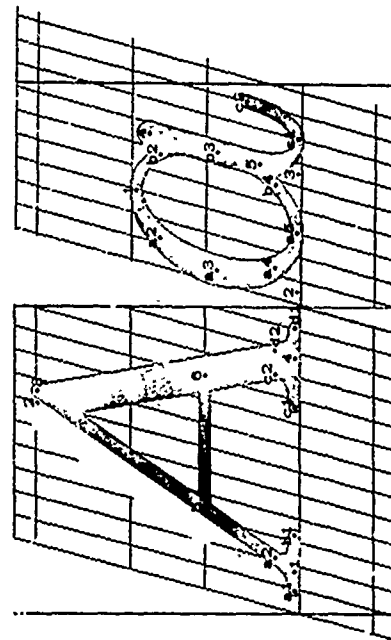
"Computer Modern Math Italic 8 point";
ph = 20; px = 138; pe = 38; pd = 38;
pb = 38; po = 38; ps = 38; pa = .5(ph - pd);
pw = 38; pwj = 38; pwii = 38; pwiii = 38;
pwiv = 38; pwv = 38; aspect = 1.0;
pu = 17; lcs = 1.028; ucs = 1.56; sc = 0; ls = 0;
slant = .25; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 0; lowast = 0; ligs = 0; mi = 1.
input cmbase; call fontbegin;
input italic;
end

```

0 1 2 3 4 5 6 7

000	Γ	Δ	Θ	Α	Β	Π	Σ	Τ
010	Φ	Ψ	Ω	α	β	γ	δ	ε
020	ζ	η	θ	ι	κ	λ	μ	ν
030	ξ	π	ρ	σ	τ	υ	φ	χ
040	ς	ι	ρ	ε	φ	θ	θ	ι
050	(	)	+	+	.	.	.	/
060	ν	ι	ς	ς	ι	ς	ς	γ
070	ς	θ	ι	ι	<	=	>	?
100	ς	Α	Β	Γ	Δ	Ε	Ζ	Η
110	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
120	Ρ	Σ	Τ	Υ	Φ	Ψ	Ω	Α
130	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι
140	Κ	Λ	Μ	Ν	Ξ	Ο	Π	Ρ
150	Σ	Τ	Υ	Φ	Ψ	Ω	Α	Β
160	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ
170	Λ	Μ	Ν	Ξ	Ο	Π	Ρ	Σ

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 *Computer X* was in more about the efficient preparation of flawless scientific documents. This is a sample of the font when the resolution is 400 dots per inch and 2.6 dots per "point".



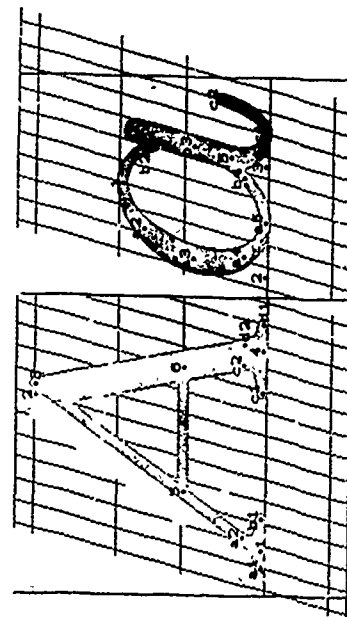
	0	1	2	3	4	5	6	7
'000	A	B	C	D	E	F	G	H
'010	I	J	K	L	M	N	O	P
'020	Q	R	S	T	U	V	W	X
'030	Y	Z	[	\	^	_	`	a
'040	b	c	d	e	f	g	h	i
'050	j	k	l	m	n	o	p	q
'060	r	s	t	u	v	w	x	y
'070	z	{	}	~	°	±	×	÷
'100	¼	½	¾	⅓	⅔	⅕	⅖	⅗
'110	⅙	⅚	⅛	⅜	⅝	⅞	⅔	⅕
'120	⅖	⅗	⅔	⅕	⅙	⅚	⅛	⅜
'130	⅝	⅞	⅔	⅕	⅙	⅚	⅛	⅜
'140	⅔	⅕	⅙	⅚	⅛	⅜	⅝	⅞
'150	⅕	⅙	⅚	⅛	⅜	⅝	⅞	⅔
'160	⅙	⅚	⅛	⅜	⅝	⅞	⅔	⅕
'170	⅚	⅛	⅜	⅝	⅞	⅔	⅕	⅙

Mathematics books and journals do not look as beautiful as they used to. It is not that their mathematical content is unattractive, 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 *drop's Xures* to measure about the efficient preparation of flabber wuffles. This is a sample of the font when the resolution is 200 dots per inch and 3.6 dots per 'point'.

The file cm17.mf

```
"Computer Modern M5th Italic 7 point";
ph = 1/3; px = 1/3; pc = 1/3; pd = 1/3;
pb = 1/3; po = 1/3; ps = 1/3; pa = .5(ph - pd);
pw = 1/3; pwi = 1/3; pwii = 1/3; pwiii = 1/3;
pwiv = 1/3; pwv = 1/3; aspect = 1.0;
pu = 1/3; lcs = 969; ucs = 1.44; sc = 0; ls = .25;
slant = .25; sqrttwo = sqrt 2; fixwidth = 0;
halfid = 0; lowast = 0; ligs = 0; mi = 1.
```

```
input cmbase; call fontbegin;
input italic;
end
```



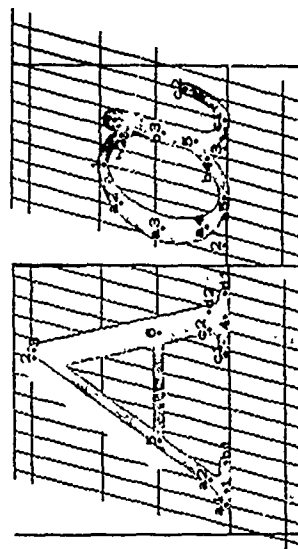
The file cmi6.mf

```
"Computer Modern Math Italic: 6 point";
ph = 1/3; px = 1/3; pe = 1/3; pd = 1/3;
pb = 1/3; po = 1/3; ps = 1/3; pa = .5(ph - pd);
pw = 1/3; pwi = 1/3; pwii = 1/3; pwiii = 1/3;
pwiv = 1/3; pwv = 1/3; aspect = 1.0;
pu = 1/3; lcs = .59; ucs = 1.43; sc = 0; ls = 1/3;
slant = .25; sqrttwo = sqrt 2; fixwidth = 0;
halfid = 0; lowast = 0; ligs = 0; mi = 1.
```

```
input cmbase; call fontbegin;
input italic;
end
```

	0	1	2	3	4	5	6	7
'000	r	Δ	⊙	Δ	Δ	π	Δ	τ
'010	⊙	⊙	⊙	⊙	⊙	τ	⊙	⊙
'020	⊙	⊙	⊙	⊙	⊙	λ	⊙	⊙
'030	⊙	⊙	⊙	⊙	⊙	⊙	⊙	x
'040	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
'050	(	)	⊙	+	⊙	⊙	⊙	/
'060	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
'070	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
'100	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
'110	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
'120	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
'130	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
'140	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
'150	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
'160	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
'170	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙

Mathematicians 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 superficial. Fortunately, it now appears that mathematics itself can be used to solve this problem, in spite of the fact that the first edition of *TeX* is now about the efficient preparation of *TeX* source code. This is a sample of the font when the resolution is 300 dots per inch and 2.0 dots per "point".

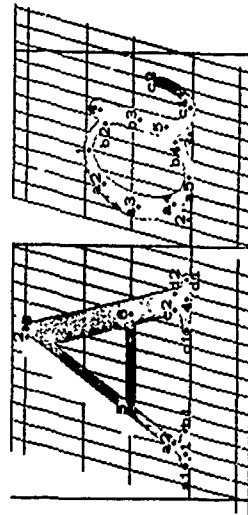


The file cmi5.mf

```
"Computer Modern Math Italic 5 point";
ph = 48; px = 38; pe = 48; pd = 38;
pb = 38; po = 38; ps = 38; pa = .5(ph - pd);
pw = 36; pwi = 36; pwil = 38; pwil = 38;
pwiv = 38; pvv = 38; aspect = 1.0;
pu = 48; lcs = .84; ucs = 1.32; sc = 0; ls = 10/12.5;
slant = .25; sqrtwo = sqrt 2; fixwidth = 0;
halfd = 0; lowast = 0; lige = 0; mi = 1.
input cmbase; call fontbegin;
input italic;
end
```

	0	1	2	3	4	5	6	7
'000	.	.	.	.	.	.	.	.
'010	.	.	.	.	.	.	.	.
'020	.	.	.	.	.	.	.	.
'030	.	.	.	.	.	.	.	.
'040	.	.	.	.	.	.	.	.
'050	.	.	.	.	.	.	.	.
'060	.	.	.	.	.	.	.	.
'070	.	.	.	.	.	.	.	.
'100	.	.	.	.	.	.	.	.
'110	.	.	.	.	.	.	.	.
'120	.	.	.	.	.	.	.	.
'130	.	.	.	.	.	.	.	.
'140	.	.	.	.	.	.	.	.
'150	.	.	.	.	.	.	.	.
'160	.	.	.	.	.	.	.	.
'170	.	.	.	.	.	.	.	.

Mathematical books and journals do not look as beautiful as they used to. It is not that their mathematical content is poorer, but that the old and well-developed traditions of typesetting have become too costly to maintain. Now, when the Computer Modern Math fonts can be used to solve this problem, the type of the future will be different. It will be a type that is different from the old, but it will be a type that is different from the old, and it will be a type that is different from the old.



	0	1	2	3	4	5	6	7
'000	-	.	x	*	/	°	±	≠
'010	⊕	⊖	⊗	⊙	⊘	+	×	•
'020	⊥	≡	≤	≥	≤	≥	≠	≠
'030	≈	≈	≈	≈	≈	≈	≈	≈
'040	↑	↑	↑	↑	↑	↑	↑	↑
'050	↓	↓	↓	↓	↓	↓	↓	↓
'060	∞	∞	∞	∞	∞	∞	∞	∞
'070	∞	∞	∞	∞	∞	∞	∞	∞
'100	/	/	/	/	/	/	/	/
'110	∞	∞	∞	∞	∞	∞	∞	∞
'120	∞	∞	∞	∞	∞	∞	∞	∞
'130	∞	∞	∞	∞	∞	∞	∞	∞
'140	∞	∞	∞	∞	∞	∞	∞	∞
'150	∞	∞	∞	∞	∞	∞	∞	∞
'160	∞	∞	∞	∞	∞	∞	∞	∞
'170	∞	∞	∞	∞	∞	∞	∞	∞

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064 065 066 067 068 069 070 071 072 073 074 075 076 077 078 079 080 081 082 083 084 085 086 087 088 089 090 091 092 093 094 095 096 097 098 099 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000

The file cmsy10.mf

```

"Computer Modern Symbols 10 point";
ph = 28; px = 48; pe = 38; pd = 38;
pb = 38; po = 38; ps = 38; pa = 5(ph - pd);
pw = 38; pwi = 38; pwii = 38;
pwiv = 38; pwv = 38; aspect = 1.0;
pu = 38; lcs = 1.075; ucs = 1.7; sc = 0; ls = 0;
slant = .25; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 0.

```

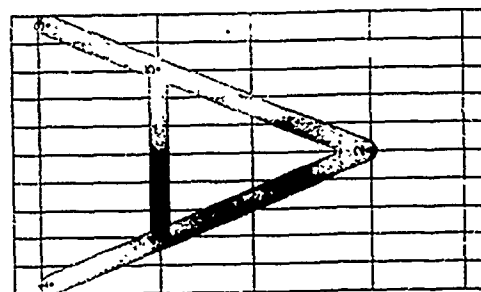
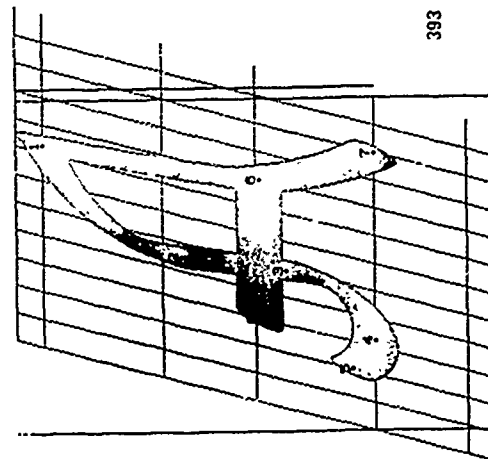
input embase; call fontbegin;

```

input symbol;
texinfo 0,
228/36,
113/36,
162/36,
244/36,
108/36,
141/36,
130.9/36,
104/36,
54/36,
99/36,
142/36,
18/36,
23.9,
10.1,
pa;
end

```

% math spacing should be variable
 % numerator baseline in displays
 % numerator baseline in nondisplay atops
 % denominator baseline in displays
 % superscript baseline in unmodified displays
 % superscript baseline in modified styles
 % subscript baseline when superscript present
 % baseline offset for superscripted large boxes
 % size of \comb delimiters in displays
 % axis height



0	1	2	3	4	5	6	7
000	-	.	x	/	.	±	±
010	⊕	⊗	⊙	⊙	+	+	•
020	⊥	≡	⊆	⊆	⊆	⊆	⊆
030	~	~	⊂	⊂	⊂	⊂	⊂
040	→	→	→	→	→	→	→
050	←	←	←	←	←	←	←
060	∞	∞	∞	∞	∞	∞	∞
070	∅	∅	∅	∅	∅	∅	∅
100	/	/	/	/	/	/	/
110	∧	∧	∧	∧	∧	∧	∧
120	∨	∨	∨	∨	∨	∨	∨
130	∩	∩	∩	∩	∩	∩	∩
140	⊂	⊂	⊂	⊂	⊂	⊂	⊂
150	⊆	⊆	⊆	⊆	⊆	⊆	⊆
160	⊇	⊇	⊇	⊇	⊇	⊇	⊇
170	⊈	⊈	⊈	⊈	⊈	⊈	⊈

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063 064 065 066 067 068 069 070 071 072 073 074 075 076 077 078 079 080 081 082 083 084 085 086 087 088 089 090 091 092 093 094 095 096 097 098 099 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999

The file cmay9.mf

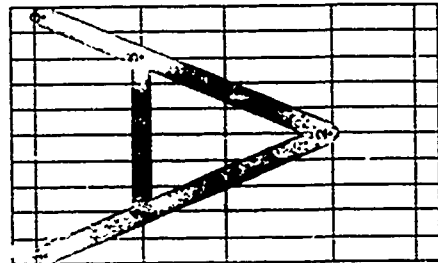
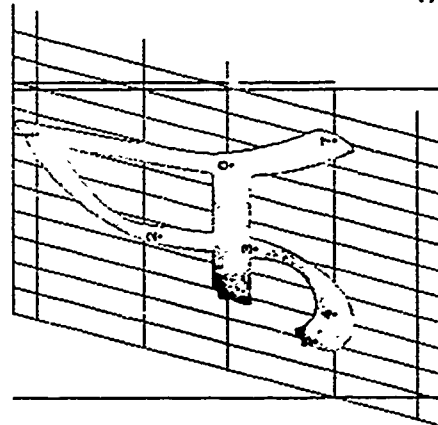
```

"Computer Modern Symbols 9 point";
ph = 33; px = 33; pe = 33; pd = 33;
pb = 33; po = 33; pa = 33; pa = .5(ph - pd);
pw = 33; pwi = 33; pwii = 33; pwiii = 33;
pwiv = 33; pwv = 33; aspect = 1.0;
pu = 33; lcs = 1.05; ucs = 1.65; sc = 0; ls = 0;
slant = .25; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 0.

input cmbase; call fontbegin;
input symbol;
texinfo 0,
221/36,
104/36,
146/36,
237/36,
92/36,
129/36,
110.9/36,
93/36,
36/36,
81/36,
126/36,
18/36,
23.9,
9.1,
pa;
end

% math spacing should be variable
% numerator baseline in displays
% numerator baseline in nondisplays, nonatops
% numerator baseline in nondisplay atops
% denominator baseline in displays
% denominator baseline in nondisplays
% superscript baseline in unmodified displays
% superscript baseline in modified styles
% subscript baseline when superscript absent
% subscript baseline when superscript present
% baseline offset for superscripted large boxes
% baseline offset for subscripted large boxes
% size of \comb delimiters in displays
% size of \comb delimiters in nondisplays
% axis height

```



	0	1	2	3	4	5	6	7
.000	-	.	x	.	/	.	±	±
.010	⊕	⊖	⊗	⊙	⊙	+	±	•
.020	⊥	≡	⊂	⊃	⊆	⊇	⊈	⊉
.030	~	≅	⊄	⊅	⊆	⊇	⊈	⊉
.040	⊊	⊋	⊌	⊍	⊎	⊏	⊐	⊑
.050	⊒	⊓	⊔	⊕	⊖	⊗	⊘	⊙
.060	⊚	⊛	⊜	⊝	⊞	⊟	⊠	⊡
.070	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪
.100	/	∕	∕	∕	∕	∕	∕	∕
.110	∕	∕	∕	∕	∕	∕	∕	∕
.120	∕	∕	∕	∕	∕	∕	∕	∕
.130	∕	∕	∕	∕	∕	∕	∕	∕
.140	∕	∕	∕	∕	∕	∕	∕	∕
.150	∕	∕	∕	∕	∕	∕	∕	∕
.160	∕	∕	∕	∕	∕	∕	∕	∕
.170	∕	∕	∕	∕	∕	∕	∕	∕

[illegible]

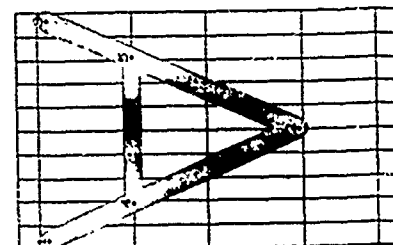
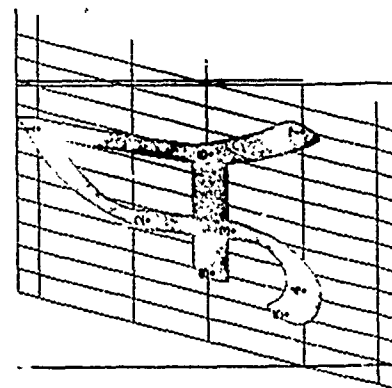
The file cmay8.nf

```

"Computer Modern Symbols 8 font";
ph = 200; px = 238; pe = 26; pd = 38;
pb = 14; ps = 36; pa = 5(ph - pd);
pw = 36; pwil = 36; rwi = 36; rwiil = 38;
pwiw = 36; pww = 38; aspect = 1.3;
cpu = 16; lcs = 1.03; ucs = 1.56; sc = 0; ls = 0;
slant = .25; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 0.

input cmbase; call fontbegin;
input symbol;
texinfo 0.
189/36,
95/36,
137/36,
203/36,
76/36,
113/36,
104/36,
82/36,
36/36,
72/36,
110/36,
18/36,
11.9,
9.1,
pa;
end

```



	0	1	2	3	4	5	6	7
'000	-	.	x	.	\	.	±	±
'010	⊖	⊖	⊖	⊖	⊖	+	+	•
'020	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
'030	~	~	~	~	~	~	~	~
'040	+	+	+	+	+	+	+	+
'050	+	+	+	+	+	+	+	+
'060	+	+	+	+	+	+	+	+
'070	+	+	+	+	+	+	+	+
'100	/	/	/	/	/	/	/	/
'110	x	x	x	x	x	x	x	x
'120	p	p	p	p	p	p	p	p
'130	x	x	x	x	x	x	x	x
'140	+	+	+	+	+	+	+	+
'150	<	<	<	<	<	<	<	<
'160	✓	✓	✓	✓	✓	✓	✓	✓
'170	!	!	!	!	!	!	!	!

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000

The file cmay7.mf

```

"Computer Modern Symbols 7 point";
ph = 1/2; px = 1/2; pe = 1/2; pd = 1/2;
pb = 1/2; po = 1/2; ps = 1/2; pa = 1/2;
pw = 1/2; pwi = 1/2; pwii = 1/2; pwiii = 1/2;
pwiv = 1/2; pwv = 1/2; aspect = 1.0;
pu = 1/2; lcs = .97; ucs = 1.44; sc = 0; ls = 0;
slant = .25; sqrtwo = sqrt 2; fixwidth = 0;
halfd = 0.

```

input embase; call fontbegin;

input symbol;

texinfo 0;

169/36,

86/36,

121/36,

182/36,

85/36,

93/36,

84/36,

72/36,

36/36,

72/36,

94/36,

18/36,

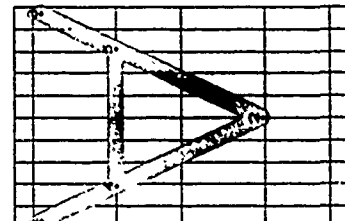
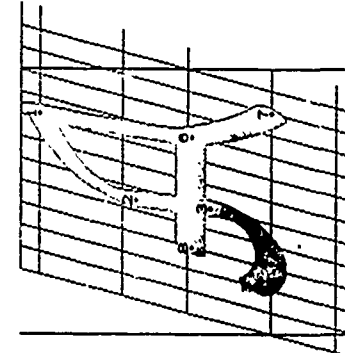
11.9,

8.1,

pe;

end

% math spacing should be variable
 % numerator baseline in displays
 % numerator baseline in nondisplays, nonatops
 % numerator baseline in nondisplay atops
 % denominator baseline in displays
 % denominator baseline in nondisplays
 % superscript baseline in unmodified dirplays
 % superscript baseline in unmodified nondisplays
 % superscript baseline when superscript absent
 % subscript baseline when superscript present
 % baseline offset for superscripted large boxes
 % size of \comb delimiters in displays
 % size of \comb delimiters in nondisplays
 % axis height



0	1	2	3	4	5	6	7
000	001	002	003	004	005	006	007
008	009	010	011	012	013	014	015
016	017	018	019	020	021	022	023
024	025	026	027	028	029	030	031
032	033	034	035	036	037	038	039
040	041	042	043	044	045	046	047
048	049	050	051	052	053	054	055
056	057	058	059	060	061	062	063
064	065	066	067	068	069	070	071
072	073	074	075	076	077	078	079
080	081	082	083	084	085	086	087
088	089	090	091	092	093	094	095
096	097	098	099	100	101	102	103
104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119
120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151
152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167
168	169	170	171	172	173	174	175

000 001 002 003 004 005 006 007
 008 009 010 011 012 013 014 015
 016 017 018 019 020 021 022 023
 024 025 026 027 028 029 030 031
 032 033 034 035 036 037 038 039
 040 041 042 043 044 045 046 047
 048 049 050 051 052 053 054 055
 056 057 058 059 060 061 062 063
 064 065 066 067 068 069 070 071
 072 073 074 075 076 077 078 079
 080 081 082 083 084 085 086 087
 088 089 090 091 092 093 094 095
 096 097 098 099 100 101 102 103
 104 105 106 107 108 109 110 111
 112 113 114 115 116 117 118 119
 120 121 122 123 124 125 126 127
 128 129 130 131 132 133 134 135
 136 137 138 139 140 141 142 143
 144 145 146 147 148 149 150 151
 152 153 154 155 156 157 158 159
 160 161 162 163 164 165 166 167
 168 169 170 171 172 173 174 175

The file cmas5.mf

```

"Computer Modern Symbols 6 point";
ph = 10; px = 30; pe = 10; pd = 10;
pb = 10; po = 30; ps = 10; pa = 5(ph - pd);
pw = 30; pwi = 30; pwii = 30; pwiii = 30;
pwiv = 30; pwv = 30; aspect = 1.0;
pu = 10; lcs = 89; ucs = 1.43; sc = 0; ls = 0;
slant = .25; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 0.

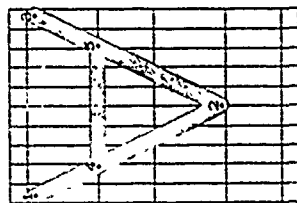
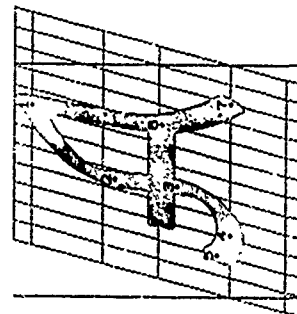
input cbase; call fontbegin;
input symbol;
texin'o 0.

150/36,
77/36,
112/36,
162/36,
94/36,
81/36,
72/36,
62/36,
36/36,
72/36,
78/36,
17/36,
11.9,
8.1,
pa;

% math spacing should be variable
% numerator baseline in displays
% numerator baseline in nondisplays, nonatops
% numerator baseline in nondisplay atops
% denominator baseline in displays
% denominator baseline in nondisplays
% superscript baseline in unmodified displays
% superscript baseline in unmodified nondisplays
% superscript baseline in modified styles
% subscript baseline when superscript absent
% subscript baseline when superscript present
% baseline offset for superscripted large boxes
% baseline offset for subscripted large boxes
% size of \comb delimiters in displays
% size of \comb delimiters in nondisplays
% axis height

end

```



The file cmsy5.mf

```
"Computer Modern Symbols 5 point";
ph = 12; px = 38; pc = 38; pd = 38;
pb = 12; po = 38; ps = 38; pa = .5(ph - pd);
pw = 38; pwi = 38; pwii = 38; pwiii = 38;
pwiv = 38; pwv = 38; aspect = 1.0;
pu = 12; lcs = .84; ucs = 1.32; sc = 0; ls = 0;
slant = .25; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 0.
```

```
input cmbase; call fontbegin;
```

```
input symbol;
```

```
texinfo 0,
```

```
130/36,
```

```
68/36,
```

```
103/36,
```

```
142/36,
```

```
103/36,
```

```
69/36,
```

```
60/36,
```

```
53/36,
```

```
36/36,
```

```
72/36,
```

```
62/36,
```

```
16/36,
```

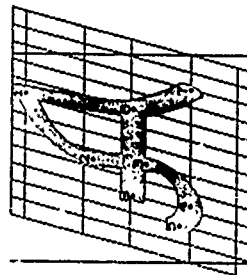
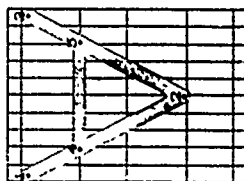
```
9.9,
```

```
7.1,
```

```
ps;
```

```
end
```

% math spacing should be variable
 % numerator baseline in displays
 % numerator baseline in nondisplays, nonatops
 % numerator baseline in nondisplay atops
 % denominator baseline in displays
 % denominator baseline in nondisplays
 % superscript baseline in unmodified nondisplays
 % superscript baseline in unmodified nondisplays
 % superscript baseline in modified styles
 % subscript baseline when superscript absent
 % baseline offset for superscripted large boxes
 % baseline offset for subscripted large boxes
 % size of \comb delimiters in displays
 % size of \comb delimiters in nondisplays
 % axis height



	0	1	2	3	4	5	6	7
'000	(	)	[	]	[	]	[	]
'010	{	}	<	>	[	]	/	/
'020	(	)	(	)	[	]	[	]
'030	[	]	{	}	<	>	/	/
'040	(	)	[	]	[	]	[	]
'050	{	}	<	>	/		ek	no
'060	/	\	[	]	[	]	'	'
'070	'	'	'	'	{	}	.	.
'100							u	u
'110	f	f	o	o	o	o	o	o
'120	Σ	Π	$\int$	$\int$	$\cap$	$\cup$	$\vee$	$\wedge$
'130	Σ	Π	$\int$	$\int$	$\cap$	$\cup$	$\vee$	$\wedge$
'140	O	O	o					
'150								
'160	$\vee$	$\vee$	$\vee$	$\vee$	$\vee$	$\vee$	$\vee$	$\vee$
'170	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ

The file cmathx.mf

```

"Computer Modern Math Extension Font";
ph = 230; px = 190; pe = 33; pd = 33;
pb = 33; po = 43; ps = 33; pa = 5(ph -- pd);
pw = 33; pwi = 33; pwii = 33; pwiii = 33;
pwiv = 33; pvv = 33; aspect = 1.0;
pu = 33; lcs = 1.075; ucs = 1.7; sc = 0; ls = 0;
slant = 0; sqrttwo = sqrt 2; fixwidth = 0;
halfd = 0.

input cmbase; call fontbegin;
input mathex;
texinfo 40/36,
60/36,
108/36,
252/36,
108/36.

% minimum glue space above large displayed operator
% minimum glue space below large displayed operator
% minimum distance to baseline of upper limit
% minimum distance to baseline of lower limit
% extra padding above and below displayed limits

```

end

INDEX

For each character code number, this index lists all pages that define a character having that code. An entry like "romext" means that the code is not defined for the roman font in this report, but it could be defined as a nonstandard character in the romext extension font.

'000: 27,201,273.	'100: 135,141,193,233,273.
'001: 27,201,275.	'101: 39,251,275.
'002: 29,201,277.	'102: 39,253,273.
'003: 29,203,279.	'103: 41,253,277.
'004: 31,203,281.	'104: 41,253,mexext.
'005: 31,203,281.	'105: 43,255,mexext.
'006: 33,203,283.	'106: 45,255,295.
'007: 35,205,283.	'107: 47,255,297.
'010: 35,205,285.	'110: 49,257,297.
'011: 37,205,287.	'111: 49,257,297.
'012: 37,205,289.	'112: 49,257,297.
'013: 105,175,197,205,291.	'113: 51,259,299.
'014: 105,175,197,205,291.	'114: 51,259,299.
'015: 119,177,207,291.	'115: 53,259,299.
'016: 119,177,207,293.	'116: 53,261,299.
'017: 119,177,207,mexext(313).	'117: 55,261,299.
'020: 119,177,207,273.	'120: 55,261,301.
'021: 119,179,207,275.	'121: 57,263,303.
'022: 121,179,209,273.	'122: 59,263,305.
'023: 121,179,209,275.	'123: 61,263,305.
'024: 121,179,209,277.	'124: 63,265,307.
'025: 121,137,179,209,279.	'125: 63,265,307.
'026: 121,137,181,209,281.	'126: 65,265,309.
'027: 123,181,211,281.	'127: 65,267,309.
'030: 123,181,211,283.	'130: 67,267,301.
'031: 123,139,183,211,283.	'131: 67,267,303.
'032: 123,139,183,211,285.	'132: 69,269,305.
'033: 105,183,199,211,287.	'133: 117,233,305.
'034: 107,183,199,211,289.	'134: 135,143,itmext,235,307.
'035: 109,183,199,213,291.	'135: 117,235,307.
'036: 125,185,213,293.	'136: 135,143,itmext,235,309.
'037: 127,185,213,mexext(313).	'137: 137,143,itmext,235,309.
'040: 133,139,189,213,273.	'140: 117,235,mexext(319).
'041: 109,215,275.	'141: 69,149,237,mexext(319).
'042: 133,139,189,215,277.	'142: 71,149,237,mexext(319).
'043: romext,itttext,139,189,215,279.	'143: 71,151,237,mexext.
'044: romext,itttext,141,191,217,281.	'144: 73,151,237,mexext(315).
'045: 129,191,217,283.	'145: 73,151,237,mexext(319).
'046: 93,165,217,283.	'146: 73,151,237,mexext(317).
'047: 111,217,283.	'147: 75,153,239,mexext(319).
'050: 111,219,285.	'150: 77,153,239,mexext.
'051: 111,219,287.	'151: 77,153,239,mexext.
'052: 113,219,291.	'152: 77,153,239,mexext.
'053: 113,221,291.	'153: 79,155,239,mexext.
'054: 113,221,293.	'154: 79,155,239,mexext.
'055: 135,141,191,223,mexext.	'155: 79,155,239,mexext.
'056: 115,223,mexext(315).	'156: 81,155,symext,mexext.
'057: 115,223,mexext(317).	'157: 81,157,symext,mexext.
'060: 93,165,225,273.	'160: 81,157,241,311.
'061: 95,167,225,275.	'161: 83,157,241,311.
'062: 95,167,225,277.	'162: 85,159,241,311.
'063: 97,167,225,279.	'163: 85,159,243,311.
'064: 97,169,227,277.	'164: 87,159,243,313.
'065: 99,169,227,279.	'165: 87,161,243,313.
'066: 99,169,227,279.	'166: 89,161,243,313.
'067: 101,171,symext,281.	'167: 89,161,symext,mexext.
'070: 101,171,227,285.	'170: 91,163,245,293.
'071: 103,173,227,289.	'171: 91,163,245,293.
'072: 115,227,285.	'172: 91,163,245,295.
'073: 115,229,289.	'173: 129,143,185,193,245,295.
'074: 115,231,287.	'174: 131,143,185,193,247,295.
'075: 117,233,289.	'175: 131,145,187,195,247,295.
'076: 117,233,287.	'176: 131,145,187,195,249,mexext(321).
'077: 103,173,symext,mexext.	'177: 133,145,187,197,249,mexext(323).