January 1986

Report No. STAN-CS-86-1095

A Torture Test for METAFONT

by

Donald E. Knuth

Department of Computer Science

Stanford University
Stanford, CA 94305

A Torture Test for METAFONT

by Donald E. Knuth
Stanford University

(*Version 1.0, January 1986*)

Programs that claim to be implementations of METRFONT84 are supposed to be able to process the test routine contained in this report, producing the outputs contained in this report.

The preparation of this report was supported in part by the National Science Foundation under grants IST-8201926 and MCS-8300984, and by the System Development Foundation.

Introduction. People often think that their programs are "debugged" when large applications have been run successfully. But system programmers know that a typical large application tends to use at most about 50 per cent of the instructions in a typical compiler. Although the other half of the code—which tends to be the "harder half"—might be riddled with errors, the system seems to be working quite impressively until an unusual case shows up on the next day. And on the following day another error manifests itself, and so on; months or years go by before certain parts of the compiler are even activated, much less tested in combination with other portions of the system, if user applications provide the only tests.

How then shall we go about testing a compiler? Ideally we would like to have a formal proof of correctness, certified by a computer. This would give us a lot of confidence, although of course the formal verification program might itself be incorrect. A more serious drawback of automatic verification is that the formal specifications of the compiler are likely to be wrong, since they aren't much easier to write than the compiler itself. Alternatively, we can substitute an informal proof of correctness: The programmer writes his or her code in a structured manner and checks that appropriate relations remain invariant, etc. This helps greatly to reduce errors, but it cannot be expected to remove them completely; the task of checking a large system is sufficiently formidable that human beings cannot do it without making at least a few slips here and there.

Thus, we have seen that test programs are unsatisfactory if they are simply large user applications; yet some sort of test program is needed because proofs of correctness aren't adequate either. People have proposed schemes for constructing test data automatically from a program text, but such approaches run the risk of circularity, since they cannot assume that a given program has the right structure.

I have been having good luck with a somewhat different approach, first used in 1960 to debug an ALGOL compiler. The idea is to construct a test file that is about as different from a typical user application as could be imagined. Instead of testing things that people normally want to do, the file tests complicated things that people would never dare to think of, and it embeds these complexities in still more arcane constructions. Instead of trying to make the compiler do the right thing, the goal is to make it fail (until the bugs have all been found).

To write such a fiendish test routine, one simply gets into a nasty frame of mind and tries to do everything in the unexpected way. Parameters that are normally positive are set negative or zero; borderline cases are pushed to the limit; deliberate errors are made in hopes that the compiler will not be able to recover properly from them.

A user's application tends to exercise 50% of a compiler's logic, but my first fiendish tests tend to improve this to about 90%. As the next step I generally make use of frequency-counting software to identify the instructions that have still not been called upon. Then I add ever more fiendishness to the test routine, until more than 99% of the code has been used at least once. (The remaining bits are things that can occur only if the source program is really huge, or if certain fatal errors are detected; or they are cases so similar to other well-tested things that there can be little doubt of their validity.)

Of course, this is not guaranteed to work. But my experience in 1960 was that only two bugs were ever found in that ALGOL compiler after it correctly translated that original fiendish test. And one of those bugs was actually present in the results of the test; I simply had failed to notice that the output was incorrect. Similar experiences occurred later during the 60s and 70s, with respect to a few assemblers, compilers, and simulators that I wrote.

This method of debugging, combined with the methodology of structured programming and informal proofs (otherwise known as careful desk checking), leads to greater reliability of production software than any other method I know. Therefore I have used it in developing METAFONT84, and the main bulk of this report is simply a presentation of the test program that was used to get the bugs out of METAFONT.

Such a test file is useful also after a program has been debugged, since it can be used to give some assurance that subsequent modifications don't mess things up.

The test file is called TRAP.MF, because of my warped sense of humor: METAFONT's **companion system,** TEX, has a similar test file called TRIP, and I couldn't help thinking about Billy Goat Gruff and the story of "trip, trap, trip, trap."

The contents of this test file are so remote from what people actually do with METAFONT, I feel apologetic if I have to explain the correct translation of TRAP.MF; nobody really cares about most of the nitty-

gritty rules that are involved. Yet I believe TRAP exemplifies the sort of test program that has outstanding diagnostic ability, as explained above.

If somebody claims to have a correct implementation of METRFONT, I will not believe it until I see that TRAP.MF is translated properly. I propose, in fact, that a program must meet two criteria before it can justifiably be called METAFONT: (1) The person who wrote it must be happy with the way it works at his or her installation; and (2) the program must produce the correct results from TRAP.MF.

METRFONT is in the public domain, and its algorithms are published; I've done this since I do not want to discourage its use by placing proprietary restrictions on the software. However, I don't want faulty imitations to masquerade as METRFONT processors, since users want METAFONT to produce identical results on different machines. Hence I am planning to do whatever I can to suppress any systems that call themselves METAFONT without meeting conditions (1) and (2). I have copyrighted the programs so that I have some chance to forbid unauthorized copies; I explicitly authorize copying of correct METRFONT implementations, and not of incorrect ones!

The remainder of this report consists of appendices, whose contents ought to be described briefly here:

Appendix A explains in detail how to carry out a test of METRFONT, given a tape that contains copies of the other appendices.

Appendix B is TRAP.MF, the fiendish test file that has already been mentioned. People who think that they understand METRFONT are challenged to see if they know what METAFONT is supposed to do with this file. People who know only a little about METAFONT might still find it interesting to study Appendix B, just to get some insights into the methodology advocated here.

Appendix C is TRAPIN.LOG, a correct transcript file TRAP.LOG that results if INIMF is applied to TRAP.HF. (INIMF is the name of a version of METRFONT that does certain initializations; this run of INIMF also creates a binary base file called TRAP.BASE.)

Appendix D is a correct transcript file TRAP.LOG that results if INIMF or any other version of METRFONT is applied to TRAP, MF with format TRAP.FMT.

Appendix E is TRAP.TYP, the symbolic version of a correct output file TRAP.72270GF that was produced at the same time as the TRAP.LOG file of Appendix D.

Appendix F is TRAP.PL, the symbolic version of a correct output file TRAP.TFM that was produced at the same time as the TRAP.LOG file of Appendix D.

Appendix G is TRAP.FOT, an abbreviated version of Appendix D that appears on the user's terminal during the run that produces TRAP.LOG, TRAP.72270GF, and TRAP.TFM.

The debugging of METRFONT and the testing of the adequacy of TRAP.MF could not have been done nearly as well as reported here except for the magnificent software support provided by my colleague David R. Fuchs. In particular, he extended our local Pascal compiler so that frequency counting and a number of other important features were added to its online debugging abilities.

The method of testing advocated here has one chief difficulty that deserves comment: I had to verify by hand that METRFONT did the right things to TRAP.MF. This took many hours, and perhaps I have missed something (as I did in 1960); I must confess that I have not checked every single number in Appendices D, E, and F. However, I'm willing to pay $5.12 to the first finder of any remaining bug in METRFONT, and I will be surprised if that bug doesn't show up also in Appendix D.

Appendix A: How to test METRFONT.

0. Let's assume that you have a tape containing **TRAP.MF**, **TRAPIN.LOG**, **TRAP.LOG**, **TRAP.TYP**, **TRAP.PL**, and **TRAP.TYP**, as in Appendices B, C, D, E, F, and G. Furthermore, let's suppose that you have a working **WEB** system, and that you have working programs **TFtoPL** and **GFtype**, as described in the TEXware and METAFONTware reports.

1. Prepare a version of **INIMF**. (This means that your **WEB** change file should have **init** and **tini** defined to be null.) The **debug** and **gubed** macros should be null, in order to activate special printouts that occur when *tracingedges* > 1.0. The **stats** and **tats** macros should also be null, so that statistics are kept. Set *mem_top* and *mem_max* to 3000 (or to *mem-min* plus 3000, if *mem-min* isn't zero), for purposes of this test version. Also set error-line = *64, half-error-line = 32, max_print_line = 72, screen-width* = 100, and *screen-depth* = 200; these parameters affect many of the lines of the test output, so your job will be much easier if you use the same settings that were used to produce Appendix E. Also set *gf_buf_size* = *8,* since this tests more parts of the program. You probably should also use the "normal" settings of other parameters found in **MF.WEB** (e.g., *max_internal = 50, buf_size = 500,* etc.), since these show up in a few lines of the test output. Finally, change METAFONT's screen-display routines by putting the following simple lines in the change file:

```
@x Screen routines:
begin init_screen:=false;
@y
begin init_screen:=true; {screen instructions will be logged}
@z
```

None of the other screen routines *(update-screen, blank-rectangle, paint-row) should be* **changed** in any way; the effect will be to have METAFONT's actions recorded in the transcript files instead of on the screen, in a machine-independent way.

2. Run the **INIMF** prepared in step 1. In response to the first '**' prompt, type carriage return (thus getting another '**'). Then type '**\input trap**'. You should get an output that matches the file **TRAPIN**.LOG (Appendix C). Don't be alarmed by the error messages that you see, unless they are different from those in Appendix C.

3. Run **INIMF** again. This time type '␣**&trap**␣␣**trap**␣'. (The spaces in this input help to check certain parts of METRFONT that aren't otherwise used.) You should get outputs **TRAP.LOG**, **TRAP.72270GF**, and **TRAP.TFM**. Furthermore, your terminal should receive output that matches **TRAP.FOT** (Appendix G). During the middle part of this test, however, the terminal will not be getting output, because **batchmode** is being tested; don't worry if nothing seems to be happening for a while—nothing is supposed to.

4. Compare the TRAP.LOG file from step 3 with the "master" **TRAP.LOG** file of step 0. (Let's hope you put that master file in a safe place so that it wouldn't be clobbered.) There should be perfect agreement between these files except in the following respects:

   a) The dates and possibly the file names will naturally be different.

   b) If you had different values for *stack-size, buf_size*, etc., the corresponding capacity values will be different when they are printed out at the end.

   c) Help messages may be different; indeed, the author encourages non-English help messages in versions of METRFONT for people who don't understand English as well as some other language.

   d) The total number and length of strings at the end and/or "still untouched" may well be different.

   e) If your METAFONT uses a different memory allocation or packing scheme, the memory usage statistics may change.

   f) If you use a different storage allocation scheme, the capsule numbers will probably be different, but the order of variables should be unchaged when dependent variables are shown. METAFONT should also choose the same variables to be dependent.

   g) If your computer handles integer division of negative operands in a nonstandard way, you may get results that are rounded differently. Although TEX is careful to be machine-independent in this regard, METRFONT is not, because integer divisions are present in so many places.

5. Use **GFtype** to convert your file **TRAP.72270GF** to a file **TRAP.TYP**. (Both of **GFtype**'s options, i.e., mnemonic output and image output, should be enabled so that you get the maximum amount of output.) The resulting file should agree with the master **TRAP.TYP** file of step 0, assuming that your **GFtype** has the "normal" values of compile-time constants (*top-pixel* = *69*, etc.).
6. Use **TFtoPL** to convert your file **TRAP.TFM** to a file **TRAP.PL**. The resulting file should agree with the master **TRAP.PL** file of step 0.
7. You might also wish to test **TRAP** with other versions of METAFONT (i.e., **VIRMF** or a production version with another base file preloaded). It should work unless METAFONT's primitives have been redefined in the base file. However, this step isn't essential, since all the code of **VIRMF** appears in **INIMF**; you probably won't catch any more errors this way, unless they would already become obvious from normal use of the system.

Appendix B: The TRAP. **MF file.** The contents of **the** test routine are prefixed here with line numbers, for ease in comparing this file with the error messages printed later; the line numbers aren't actually present.

```
1 % This is a diabolical test file for MF84. Don't get stuck.
2 if not known inimf: .inimf.=.O. % next lines are skipped if format loaded
3   inimf; nonstopmode; tracingtitles:=day; showstopping:=year; hppp:=1000;
4   << == >> :: || '' JJ - !! ?? ## && @@ $$ [[ ]] {{ }} . (( 5.5.5 )) ++ ". .";
5   begingroup save =; let=,; save,; newinternal $=,; let ):=, endgroup;
6   let year=month; showvariable errorstopmode,readstring,2,"2",,,(,),<<,year;
7   tracingrestores :=tracingcommands:=.00000762939453125; % that's 2^(-17)
8   if tracingcommands>0:tracingcommands:=if not cycle "":1.1 forever;fi;
9   tracingcommands :=2.1 exitif tracingcommands>2 endfor;
10  tracingedges :=1/.00001; tracingequations :=$+I; p~=tracingedges+.00001;
11  interim tracingspecs:=tracingpens:=tracingchoices:=tracingstats:=
12  warningcheck:=tracingoutput:=tracingmacros:=1; $:=ASCII""; $:=x; p~:=p~;
13  delimiters (); delimiters begintext endtext;
14  vardef foo(text t)=t enddef;
15  foo begintext interim proofing:=(-.5; shipout nullpicture; special"3" endtext;
16  for n=tracingpens step 1 until proofing:fi endfor showstats;
17  path p~; p~=(0,0)..controls (15,4) and (-15,-12)..(4,0);
18  everyjob /*\;;
19  vardef /*\''@#=message @ & str#@ &jobname&char ASCII''&str@#!enddef;
20  let next=dump; vardef ''='' enddef; def ''="\*/" enddef;
21 elseif known"": ''pass[2.]; outer\; let next=\; delimiters ^~7! fi
22 next\; % the second pass will now compute silently; the first pass will halt
23 batchmode; ^~7,endgroup pausing:=I; exitif p exitif boolean pen pencircle endfor
24 scantokens begingroup message char0&"watch this"; "pair p[],';"&char-1endgroup
25 path p[][]p,w,qw; qw=(1,-2)..(2,-1)..(2.5,0.5)..(1,2)..(turningnumber',2.5);
26 numeric p[][];p[[[$]]]=lOOOOOOOOOOOOOOO; "this string constant is incomplete
27 string foo[]p,p~if true:[];
28 boolean p[]~,boolean fi.boolean; showvariable boolean; def\\= =end enddef;
29 picture e[]e[], e[], p~[]~[][];
30 pen p~[]~,q["a",qq; p~1~=q=pencircle scaled mexp(-3016.57654);
31 transform p,pp0; if p=p:qq=makepen((1,0)..cycle) xscaled hex "1000";fi
32 qq:=makepen((0,0)..(1,0)..(0,1)..(0,0)..(1,0)..(0,1)..cycle);
33 vardef p[][]p~ begintext suffix a,b endtext())suffix@=show #@; p.a.b() enddef;
34 expandafter let\endtext\\; outer endtext,\,;;qq:=pencircle scaled 4.5 yscaled 2;
35 (6,12)-p7=(0,1)transformed p=(2/(x-x),3/0)transformed p;
36 p1\2p=p007=begintext if string p~ [$]: p.1.2-p.1.199999,1 endtext transformed p;
37 showstopping:=0;showvariable p; p=p; let [[=begingroup; let ]]=endgroup;
38 (xxpart p+.002,yxpart p)=1[p1,p2]=(5,y+.00001)=(5,y)=(yypart p,xypart p);
39 reverse(p~..cycle) transformed p=p2. 3.p;
40 p[000000000001]2p~(,[2]3p~,-)=p~1~2[pausing];
41 vardef p~[]@# tertiary t:=if p@ @=@ @p fi; vardef p[][]p~[]=BAD; inner ;;
42 show p~[-2]~[3000,x]++4000>path p3; showvariable p,P;
43 numeric p[]~; p3~=2alpha; p[1/$]~=3beta;
44 begingroup save p; showvariable p; 3beta=1]]; showvariable p;
45 def//expr;=enddef;def!primary!false):!fi enddef;
46 def _aa__ secondary _a_=if(true enddef; qq:=makepen(qw..(qw scaled$)..cycle);
47 primarydef _**__=[[show _*__]] enddef;
48 secondarydef _***__=expandafter __ scantokens"**oct"_ enddef;
49 // //pencircle slanted length p~**makepen reverse subpath p7-p2 of
50  (p7{p2}..controls _aa__not odd.1(15) and known p or !p2and-p2..{1,1}(-p2
51 (curl- +1)..tension atleastl..cycle sqrt2++sqrt2***[[]];
```

```
52 [[interim proofing:=charcode:=-20.5;chardp:=-2048;shipout nullpicture]];
53 if charexists -275.50002>known p0 0p=known p~: randomseed charcode; fi
54 randomseed: ="goof"; a[($,18++1+-+18),(2,3)]=b[(3,2),(1,$);
55 show (^+1,~+2) slanted-l yscaled-2 zscaled-(3,4), p transformed(pp xscaled 9),
56 pp shifted (1,2) transformed(p transformed p), -_[0] [1,2]; show
57 floor sind mlog sqrt mexp200cosd angle(normaldeviate,uniformdeviate-chardp);
58 string s[] ; s1=s2=s4; s3=s5; s1=s2; i f s1<=s4<>(s1<>s3):show[[char34=s2:=s3]]fi;
59 substring penoffset point.1of.p~of[[pencircle]]rotated1080/2/1/3of decimal
60 directiontime postcontrol-1.5of(p~&cycle)-precontrol1/2of p~(p~)=s1:=s4:=s4;
61 path p~ []; p~1=p2{length" "}&cycle; p~1=p2=p~0; p2..controls-p2..cycle=p~2;(p7
62 ..tension1.2..p~[length p~2]..p~2&{0,1}p2..tension1and atleast1..cycle)..tension
63 x. . {curl1}-p7{curl hex "IsBad"}..tension.75and.74999..p2{0,1}&p2{_,'}..cycle:=p
64 ~4; subpath(-9,9)of subpath(3.5001,7.00001)of p~4=p~6;
65 show p-6, directiontime(1,2) of p~6, directiontime(l,-1.00001) of p-6;
66 p~3=(0,0)..controls (1,1) and (O,l)..(l,O); show p2..p2{p7}&{$,$}cycle,
67 (directiontime(1,1) of subpath(.314159,1) of p~3)[.314159,1];
68 p~5=(subpath(0,.25)of p~3&subpath(.25,1)of p~3)shifted begintext1,0;
69 p~3:=2/3'zscaled'{p~3}..controls(2,2/3(3))and penoffset(1/2x,y)of(0,1)(1,0);
70 show p~3 intersectiontimes reverse p-3, point.17227 of p-3, point1-.28339of p-3;
71 show point xpart(p~5 intersectiontimes p~5 shifted (.Ol,O))of p-5-
72 point ypart(p~5 intersectiontimes p~5 shifted (.01,0))of p~5;
73 [[interim tracingedges:=1; e C-l+-+ -1.000011 =nullpicture;
74 addto e0 doublepath p~5 scaled 3 withpen q; e0=e1=e2; cull el dropping (0,.1);
75 show el shifted(4089,-4095), e2 shifted(-4095,4092) shifted (-3,0),
76 e2 shifted(4089,-4095)]];addto el also e2 shifted(-2,$); e1:=e1 shifted(-4,$);
77 addto e0 also e1rotated89.999+e1scaled$; show e0 xscaled-10 yscaled2 xscaled82
78 yscaled683;addto el doublepath (0,9) withweight- withweight turningnumber p~6
79 withpen pencircle xscaled(oct"180"++1) rotated-angle(64,$) shifted (9,8)
ao withpen makepen(($,$)..(1,0)..(1,1)..($,0)..($,$)&cycle)xscaled4095.49999;
81 show el, totalweight el; chardp:=charcode:=5; xoffset:=-1.5; shipout el;
82 showstats; addto e2doublepath p~ yscaled1/60; e3:=e2 yscaled 0;
83 autorounding:=2; addto e3doublepath(.5,0)..(3.5,1.5)withweight2;
a4 tracingspecs:=0; q:=makepen((1,1)..cycle) yscaled 1.5;
85 p~8=(($,yy)rotated p{0,1}..{0,$}(1,0){0,$}..cycle)scaled2shifted(1000.49,9);
a6 turningcheck:=l;addto e2doublepath p~8 withpen q withweight p withpen cycle' p;
87 [[interim autorounding: =xx=.I; addto e2 contour p~8 withpen q withweight2]];
88 chardx:=chardp:=charext:=-14.5;shipout-(-e0-e2)+e2shifted(0,6turningnumber p~8);
a9 p~9=(0,0)..(1,.5)..(5,1.5)..(7,2.5)..(12,3.5)..(13,4);addto e3 doublepath p-9;
90 smoothing:=l; addto e3 doublepath p~9; addto e3 doublepath (-4095,0)..tension
91 3/4 and 999..(0,2); show e3 rotated-go, (e0+e0) rotated90$ rotated90;
92 if "a" if "ab">"b" elseif path reverse (3,4) : >="aa" : foo elseif f i "bar"
93 else if '-(1,yy)<': :fi else def dup text t=[[t;save enddef;def|suffix$=,$
94 enddef; def I |tertiary p=show substring p of ("a" enddef; ||(2,$)&"bc");
95 tertiarydef x++y=[[dup showtoken x;]];def quote x expr z of y=z enddef;
96 texts(quote x=(y+O)yty) ("xx",foo(x))=O]] enddef; def texts(text t,tt)expr?=
97 for n:=,for n"yy":n,length if false:endfor tt,t,:if string n:forsuffixes n=
98 fool, [foo(n)],':show t,tt|(n;exitif not('<='+(?,yy)) endfor for m=:+endfor
99 for m=alpha step-l. 1 3$: tm endfor fi endfor enddef; show (0,0){curl2}..
100 subpath(23.3,4.5)of p~9{curl3}. .($,$){curl4}..cycle;numspecial p~++2+3;[[
101 let?=if;save if,\;def if=endinput?enddef;def texts=input enddef;texts trap ]];
102 dup[[def texts secondary x=primarydef y++y=x@y enddef; showtoken++;x enddef]]\;
103 proof ing:=1; texts:="a"&"b";% strings "yy" and "ab" no longer appear in memory
104 texts-1.00001a1=-(a2+a3+a4+a5+a6);-(a3,a2)/.99999=-(a4+a5+[[showdependencies;
```

```
105 a6]],a3+a4+a5+a6)]]; 1.00001a4+1=a5+a6; alpha:=.9alpha+7; showdependencies;
106 a5=a6=2/3-a6; cull el dropping($,4a5)withweight1.5; charcode:=chardp:=27;
107 openwindow 3 from (0,0) to (0,0) at "whoops"; addto p; shipout p; cull p;
108 openwindow -.5 from ($,$) to ($,$) at (0,0); special p; numspecial "p";
109 openwindow 15.49999 from (0,32*1024) to (1,31*1057) at (0,0); shipout el;
110 openwindow 5 from (0,length((0,0){0,0})) to (4,2) at ($,1);
111 openwindow 6 from (length($,0),1) to (3,10) at (-5,2);
112 display nullpicture; display p inwindow 3; display p inwindow 6;
113 display el inwindow 6; cull e0 dropping (0.1,4095.999999) withweight 3.5
114 withweight-3.5; display e0 inwindow 5.5; addto e0 also p; addto e0 contour 0;
115 display e0 inwindow 5.49999; addto e0 contour p-9;
116 display el inwindow 3+3; display e0e0 inwindow[[vardef e[]e=enddef;6]];
117 addto e0 also el; display e0 inwindow 5;
118 display al inwindow 15;
119 display el inwindow 6;
120 b1:=c.a[ [[let c=++;vardef b=enddef;1]] ];
121 errhelp 0; errmessage "Be like Jane";
122 errhelp "He%%%lp%"; errmessage"";
123 errhelp ""; errmessage "Another";
124 headerbyte 0; headerbyte(48.5)substring(-9,9)of"long";
125 headerbyte 9:2a6,"q"; fontdimen 9:2a6,"q"; fontdimen 1:2048;
126 fontmaking:=1; extensible 5 5,"c"255.5,"d"; charlist 0:5:"a":"d";
127 ligtable "a"=:"b","c":"d" kern -2048,0:"c":99.5:"e"=:"f",0kern'; ligtable 5:O;
128 def clear(text x)=interim x:=$ enddef; clear(hppp); vppp=0;
129 [[clear(tracingmacros); clear(tracingcommands); clear(tracingoutput\;
130  clear(proofing);designsize:=.99999;charcode:=ASCII char-290.5;vppp:=designsize;
131 def dp expr d = charcode:=charcode+1; chardp:=d; shipout nullpicture enddef;
132 dp 13; dp 9; dp 0; dp 21; dp -2; dp 17; dp 11; dp 3; charic:=-1000; dp -1;
133 dp 25; dp 31; dp 19; dp 7; charwd:=256; chardy:=6; dp 23; dp 3011;
134 def f(suffix@@)(expr a,b)(text t)=numeric w; show a; % wipes out the old w
135 addto @@ contour (0,0)..(2,0)..(1,$)..(1,1)..cycle withpen qq; % strange path
136 addto @@ doublepath (0,0){1,1}..{2,1}(2,1) withpen qq; % carefully chosen
137 addto @@ doublepath(($,$){1,0}..(1,1){1,0})scaled.5 withpen nullpen;
138 cull @@ keeping (4,4) withweightl.5; enddef; def g(suffix$)=show $ enddef;
139 addto e0 contour (0,0){1,0}..{1,0}(1,0){0,$}..{0,1}cycle withpen qq;
140 f(e[3,w); g(e3,transformed p,penoffset-(1,1.3)of(pencircle scaled20 yscaled-.5),
141 directiontime (0,1) of ((0,0)..controls(1,1)and($,1)..(3,0)),
142 point 3.14169 of p~9 intersectiontimes subpath (3.14159,4) of p~9,
143 (($,1.1)..(1,$)) intersectiontimes precontrol$ of (0,0);
144 addto e3 doublepath(-4094.99998,0)..(4094.99998,-.00001) withpen penoffset 0 of
145 pencircle; addto e3 also e3 shifted (0,257); ,"flushed with pride"; numeric xx;
146 def f(expr x,y,z)=showdependencies;tracingcapsules:=1;showdependencies;show
147  1/3(3,6)*((x+y)+(y-x)), (1,1)/sqrt2 zscaled (x+1,x+2) - (x+1,x+2) rotated 45,
148 (0,1) zscaled (1,y+2)-(1,y+2) rotated 90 enddef; f((xx+1)/.3,(yy-1)/.5,(xx,0));
149 '=(1000o3-4000(o1-o2)+4000o2+9,-.01o3+3ooo+
150 [[oo=9/10(o2+o4+o5-20);o4=o5=8/9(o1+.5o2); o6=-.0001o2;showdependencies;
151   numeric o[];xpart(alfa,[[pair alfa;0]])]])=-2/3[[save p;(p$,1-p$)]];
152 xoffset:=yoffset:=4000[[oo=.5ooo=2*-1/2(ooo+[[numeric ooo;1]]);oo]];
153 for @=angle(sqrt$,mlog$):charext:=uniformdeviate$;charht:=2048;granularity:=-8;
154 addto e3 contour (0,-100)..tension 500..(100,-99)..tension 3000..cycle;
155 tracingoutput:=@; shipout e3; special "bye"; interim char 99 = "c"; true=false;
156 [[clear(tracingcommands); charcode:=ASCII char 141-13; shipouttnullpicture;
157 "careful" quote for for = @ step 200 until 2*2600: &" METAFONT" endfor;]];
```

```
158 scrollmode ; *hello again"&char31; save p; fillin:=-.04;
159 def f expr x=let )=] ; let [=(; show _ enddef; begingroup tracingspecs:=1;
160 show nullpen, makepath.qq, makepath(q rotated1), makepath pencircle rotated $;
161 addto e0 doublepath (0,2){0,$}..{0,$}(0,1)..{1,0}(3,0)..(4,0){1,0}..cycle
162  withpen makepen((0,0)..(5,2.9)..(4,3)..cycle); tracingonline:=1; f xx[1);
163 showdependencies; qq:=q; showstats; bye endtext
164 % things not tested:
165 % interaction (error insertion/deletion, interrupts, \pausing, files not there)
166 % date, time; initialization of random number generator uithout randomseed
167 % system-dependent parsing of file names, areas, extensions
168 % certain error messages, especially fatal ones
169 % things that can't happen in INIMF
170 % unusuai cases of fixed-point arithmetic
```

Appendix C: The TRAPIN .LOG file. When INIMF makes the TRAP. BASE file, it also creates a file called TRAP .LOG that looks like this.

```
This is METAFONT, Version 1.0 (INIMF) 4 JAN 1986 09:32
**\input trap
(trap.mf
>> << == >> :: ||`` ''--!! ??## && @@ $$[[]]{{ }}((5.5 0.5))
>>"..."
! lot implemented: (unknown numeric)++(string).
<to be read again>

1.4 ...}} . (( 5.5.5 )) ++ "..";

I'm afraid I don't know how to apply that operation to that
combination of types. Continue, and I'll return the second
argument (see above) as the result of the operation.

..
! Missing `=' has been inserted.
<to be read again>

1.5   begingroup save =; let=,
                                ; save,; newinternal $=,; let )...
You should have said 'let symbol = something'.
But don't worry; I'll pretend that an equals sign
was present. The next token I read will be 'something'.

> errorstopmode=errorstopmode
> readstring=readstring
> 2
> "2"
> ,=,
> (=tag
> )=,
<< == >> :: ||`` ''--!! ??## && @@ $$[[]]{{ }}(([][]))=numeric
<< ==>>::||``J J - .!! ??## && @@ $$[[]]{{ }}((5.5 0.5))=<< = =  >> :: ||
`` JJ--!! ??## && 0 0  $$[[]]{{ }}((5.5 0.5))
> year=month
! OK.
1.6 ...ring,2,"2",,,(,),<<,year;

! Hissing `:' has been inserted.
<to be read again>
                    ;
1.8 ...not cycle "":1.1 forever;
                                fi;
The next thing in this loop should have been a `:'.
So I'll pretend that a colon was present;
everything from here to `endfor' will be iterated.

{fi}
{exitif}
([repeat the loop]}
```

```
{false}
{fi}
{exitif}
([repeat the loop]}
{(2.1)>(2)}
{true}
{tracingcommands:=2.1}
! Arithmetic overflow.
1.10    tracingedges:=1/.00001
                              ; tracingequations:=$+1; p~=trac...
Uh, oh. A little while ago one of the quantities that I was
computing got too large, so I'm afraid your answers will be
somewhat askew. You'll probably have to adopt different
tactics next time. But I shall try to carry on anyway.

{tracingedges:=32767.99998}
{(0)+(1)}
{tracingequations:=1}
{(32767.99998)+(0.00002)}
! Arithmetic overflow.
1.10 . . . p~=tracingedges+.00001;

Uh, oh. A little while ago one of the quantities that I was
computing got too large, so I'm afraid your answers will be
somewhat askew. You'll probably have to adopt different
tactics next time. But I shall try to carry on anyway.

((p-)=(32767.99998))
## p-=32767.99998
(interim3
{tracingmacros:=1}
{tracingoutput:=1}
{warningcheck:=1}
{tracingstats:=1}
{tracingchoices:=1}
{tracingpens:=1}
{tracingspecs:=1}
{ASCII("")}
{$:=-1}
{$:=x}
>> x
! Internal quantity '$' must receive a known value.
<to be read again>
                   ;
1.12 . ..os:=1; $:=ASCII""; $:=x;
                                    p~:=p~;
I can't set an internal quantity to anything but a known
numeric value, so I'll have to ignore this assignment.

{p~:=32767.99998}
## p-=32767.99998
! Value is too large (32767.99998).
1.12 . . . :=ASCII""; $:=x; p~:=p~;
```

```
The equation I just processed has given some variable
a value of 4096 or more. Continue and I'll try to cope
with that big value; but it might be dangerous.
(Set warningcheck:=0 to suppress this message.)

(delimiters)
(delimiters)
{vardef}

foo(TEXT2)->begingroup(TEXT2)endgroup
(SUFFIX0)<-
(SUFFIX1)<-foo
(TEXT2)<-interim.proofing:=(-0.5;shipout.nullpicture;special"3"
{begingroup}
(interim)
{-(0.5)}
! Kissing ')' has been inserted.
<to be read again>
                   ;
<argument> ...m.proofing:=(-0.5;
                                  shipout.nullpicture;special"3"
foo->begingroup(TEXT2)
                        endgroup
1.15 . ..ture; special"3" endtext

I found no right delimiter to match a left one. So I've
put one in, behind the scenes; this may fix the problem.

{proofing:=-0.5)
{shipout}
{nullpicture}
{special)
(restoring proofing=0}
{endgroup}
{for}
{showstats}
Memory usage 27385 (1950 still untouched)
String usage 25879 (1033&12100 still untouched)
! OK.
1.16 . . .ing:fi endfor showstats;


{path}
{-(15)}
{-(12)}
Path at line 17, before choices:
(0,0)..controls (15,4) and (-15,-12)
 ..(4,0)

Path at line 17, after choices:
(O,O)..controls (15,4) and (-15,-12)
 ..(4,0)
```

```
((unknown path p~)=(path)}
{everyjob}
{vardef}
{let}
{vardef}
{def}
{elseif}

Beginning to dump on file trap.base
 (preloaded base=trap 86.1.4)
975 strings of total length 19949
343 memory locations dumped; current usage is 299&35
253 symbolic tokens
```

Appendix D: The TRAP. LOG file. Here is the major output of the TRAP test; it is generated by running **INIMF** and loading TRAP. BASE, then reading TRAP.WF.

```
This is METAFONT, Version 1.0 (preloaded base=trap 86.1.4)  4 JAI 1986 09:36
* * &trap trap
(trap.mf
{if}
{known(0)3
{not(true)}
ifalse}
{known("")}
{true}

/*\''@$->begingroup.message(SUFFIX1)&str(SUFFIX0)&jobname&char.ASCII''&s
tr(SUFFIX2)!endgroup
(SUFFIX0)<-/*\
(SUFFIX1)<-''
(SUFFIX2)<-pass2
{begingroup}
{message}

''->begingroup''endgroup
(SUFFIX0)<-
(SUFFIX1)<-''
{begingroup}

''->"\*/"
{endgroup}
{("\*/")&("/*\")}
{jobname}
{("\*//*\")&("trap")}

;''->"\*/"
{ASCII("\*/")3
{char(92)}
{("\*//*\trap")&("\")}
{("\*//*\trap\")&("pass2!")}

\*//*\trap\pass2!
{endgroup}
{outer}
{let}
{delimiters)

! Hissing symbolic token inserted.
(inserted text>
                INACCESSIBLE
1.21 . . .t next=\; delimiters ^^7
                                ! fi
Sorry: You can't redefine a number, string, or expr.
I've inserted an inaccessible symbol 80 that your
definition will be completed without mixing me up too badly.

! Extra tokens will be flushed.
<to be read again>
                   !
1.21 . . . next=\; delimiters ^^7!
                                  fi
I've just read as much of that statement a8 I could fathom,
80 a semicolon should have been next. It's very puzzling...
but I'll try to get myself back together, by ignoring
everything up to the next ';'.Please insert a semicolon
now in front of anything that you don't want me to delete.
(See Chapter 27 of The METAFONTbook for an example.)

! Forbidden token found while scanning to the end of the statement.
<inserted text>
                ;
<to be read again>

1.22 next\
```

```
          ; %. the second pars will now compute silently; the . . .
A previous error seems to hare propagated,
causing me to read past where you wanted me to stop.
I'll try to recover; but if the error is serious,
you'd better type 'E' or 'X' now and fix your file.

{\}
{batchmode}

! An expression can't begin with 'endgroup'.
<inserted text>
                0
<to be read again>
                   endgroup
1.23 batchmode; ^^7,endgroup
                              pausing:=1;exitif p exitif bool...
I'm afraid I need some sort of value in order to continue,
so I've tentatively inserted '0'. You may want to
delete this zero and insert something else;
see Chapter 27 of The METAFONTbook for au example.

! Hissing ' IBACCESSIBLEJ has been inserted.
Cto be read again>
                   endgroup
1.23 batchmode; ^^7,endgroup
                              pausing:=1;exitif p exitif bool...
I found no right delimiter to match a left one. So I've
put one in, behind the scenes; this may fix the problem.

! Extra 'endgroup'.
<recently read> endgroup

1.23 bat&mode; ^^7,endgroup
                              pausing:=1;exitif p exitif bool...
I'm not currently working on a 'begingroup',
so I had better not try to end anything.

{pausing:=13
{exit if }
{exitif}
{pencircle}
{endfor}
! Extra 'endfor'.
1.23 . . .ean pen pencircle endfor

I'm not currently working on a for loop,
so I had better not try to end anything.

{scantokens}
{begingroup}
(message3
{char(0)}
{("^^@")&("watch this"))
^^@watch this
{-(1)}
{char(-1)}
{("pair p[],';")&("^^?")}
{endgroup}
{pen(future pen)}
{boolean(true)}
{true}
! No loop is in progress.
<to be read again>
                   pair
<scantokens>pair
                  p[],';^^?
<to be read again>
                   path
1.25 path
          p[][]p,w,qw;qw=(1,-2)..(2,-1)..(2.5,0.5)..(1,2)..(...
Why say 'exitif' when there's nothing to exit from?
```

```
>> p
! Undefined condition will be treated as 'false'.
<to be read again>
                    pair
<scantokens>pair
                  p[],';^^?
<to be read again>
                    path
1.25 path
          p[][]p,w,qw; qw=(1,-2)..(2,-1)..(2.5,0.5)..(1,2)..(...
The expression shown above should hare had a definite
true-or-false value. I'm changing it to 'false'.

{false}
! Missing '; ' has been inserted.
<to be read again>
                    pair
<scantokens> pair
                  p[],';^^?
<to be read again>
                    path
1.25 path
          p[][]p,w,qw; qw=(1,-2)..(2,-1)..(2.5,0.5)..(1,2)..(...
After 'exitif (boolean exp>' I expect to see a semicolon.
I shall pretend that one was there.

{pair}
! Text line contain8 an invalid character.
<scantokens>pair p[],J;^^?

<to be read again>
                    path
1.25 path
          p[][]p,w,qw; qw=(1,-2)..(2,-1)..(2.5,0.5)..(1,2)..(...
A funny symbol that I can't read has just been input.
Continue,  and I'll forget that it ever happened.

{path}
{-(2)}
{-(1)}
{turningnumber((xpart ',ypart '))}
Path at line 25, before choices:
(1,-2){curl 13
 ..(2,-1)
 ..(2.5,0.5)
 ..(1,2)
 ..{curl 1}(0,2.5)

Path at line 25, after choices:
(1,-2)..controls (1.37755,-1.71404) and (1.71404,-1.37755)
 ..(2,-1)..controls (2.33353,-0.55965) and (2.59729,-0.04124)
 ..(2.5,0.5)..controls (2.36812,1.23369) and (1.6712,1.65662)
 ..(1,2)..controls (0.66821,2.16974) and (0.33485,2.33641)
 ..(0,2.5)

{(unknown path qw)=(path)}
(numeric)
! Enormous number has been reduced.
1.26 . . .[$]]]=10000000000000000
                                    ; "this string constant is in.. .
I can't handle numbers bigger than about 4096.99998;
so I've changed your constant to that maximum amount.

{(p[[ [-1] ]])=(4095.99998)}
## p[[ c-11 ]]=4095.99998
! Incomplete string token has been flushed.
1.26 . . .g constant is incomplete

Strings should finish on the same line as they began.
I've deleted the partial. string; you right want to
```

```
insert another by typing, e.g., 'I"new string"'.

{string}
! Declared variable conflicts with previous rardef.
<to be read again)
                  ,
1.27 string foo[]p,
                 p~if true: Cl;
You can't use, e.g., 'numeric foo[]' after 'rardef foo'.
Proceed, and I'll ignore the illegal redeclaration.

{if}
{true}
{true}
(boolean3
{fi}
{showvariable}
boolean.boolean=unknown boolean
! ox.
1.28 ...n; showvariable boolean;
                                  def\\= =end enddef;

{def}
(picture)
{pen}
! Illegal suffix of declared variable will be flushed.
<to be read again>
                  L
<to be read again>
                  "a"
1.30 pen p~[]~, q["a"
                   ,qq; p~1~=q=pencircle scaled mexp(-3016.5...
Variables in declarations must consist entirely of
names and collective subscripts, e.g., 'x[]a'.
Are you trying to use a reserved word in a variable name?
I'm going to discard the junk I found here,
up to the next comma or the end of the declaration.

{pencircle}
{-(3016.57654)}
{mexp( -3016.57654))
{(future pen)scaled(0)}
Pen polygon at line 30 (newly created):
(0.5,0)
 .. (0,0.5)
 .. (-0.5,0)
 .. (0,-0.5)
 .. cycle

{(unknown pen q)=(pen)}
{(unknown pen p~1~)=(pen)}
{transform)
! Illegal suffix of declared variable will be flushed.
<to be read again>
                  0
1.31 transform p,ppO
                     ; if p=p:qq=makepen((1,0)..cycle) xscaled...
Variables in declaration8 must consist entirely of
names and collective subscripts, e.g., 'x[]a'.
Explicit subscripts like 'x15a' aren't permitted.
I'm going to discard the junk I found here,
up to the next comma or the end of the declaration.

{if}
{((xpart p,ypart p,xxpart p,xypart p,yxpart p,yypart p))=((xpart p,ypart
 p, xxpart p, xypart p, yxpart p, yypart p))}
{true}
Path at line 31, before choices:
(1,0)
 ..cycle
```

```
Path at line 31, after choices:
(1,0)..controls (1,0) and (1,0)
 ..cycle

{makepen(path)}
{hex("1000")}
! lumber too large (4096).
<to be read again>

1.31 . . .cle) xacaled hex "1000";
                                fi
I have trouble with numbers greater than 4095; watch out.

{(future pen)xscaled(4096)}
! Pen too large.
<to be read again>

1.31 . . .cle) xacaled hex "1000";
                                fi
The cycle you specified has a coordinate of 4095.5 or more.
So I've replaced it by the trivial path '(0,0)..cycle'.

Pen polygon at line 31 (newly created):
(0,0)
 .. cycle

{(unknown pen qq)=(pen)}
{fi}
Path at line 32, before choices:
(0,0)
 ..(1,0)
 ..(0,1)
 ..(0,0)
 ..(1,0)
 ..(0,1)
 ..cycle

Path at line 32, after choices:
(0,0)..controls (0.29056,-0.29056) and (0.75859,-0.30772)
 ..(1,0)..controls (1.51964,0.66237) and (0.66237,1.51964)
 ..(0,1)..controls (-0.30772,0.75859) and (-0.29056,0.29056)
 ..(0,0)..controls (0.29056,-0.29056) and (0.75859,-0.30772)
 ..(1,0)..controls (1.51964,0.66237) and (0.66237,1.51964)
 ..(0,1)..controls (-0.30772,0.75859) and (-0.29056,0.29056)
 ..cycle

{makepen(path)}
! Pen cycle must be convex.
<to be read again>

1.32 . . .)..(1,0)..(0,1)..cycle);

The cycle you specified either has consecutive equal points
or turns right or turns through more than 360 degrees.
So I've replaced it by the trivial path '(0,0)..cycle'.

Pen polygon at line 32 (newly created):
(0,0)
 .. cycle

{qq:=pen}
{vardef}
! Missing parameter type; 'expr' will be assumed.
<to be read again>
                )
1.33 . . .ext suffix a,b endtext()
                                )suffix@=show #@; p.a.bO end...
You should've had 'expr' or 'suffix' or 'text' here.

{expandafter}
{\}
```

```
{let}

\\->-end
{outer}
{pencircle}
{(future pen)scaled(4.5)}
{(future pen)yscaled(2)}
Pen polygon at line 34 (newly created):
(0.5,-4.5)
 .. (1.5,-3.5)
 .. (2,-2)
 .. (2.5,0)
 .. (2,2)
 .. (1.5,3.5)
 .. (0.5,4.5)
 .. (-0.5,4.5)
 .. (-1.5,3.5)
 .. (-2,2)
 .. (-2.5,0)
 .. (-2,-2)
 .. (-1.5,-3.5)
 .. (-0.5,-4.5)
 .. cycle

{qq:=pen}
{((6,12))-((xpart p7,ypart p7))}
{((0,1))transformed((xpart p,ypart p,xxpart p,xypart p,yxpart p,yypart p
))}
{(x)-(x)}
{(2)/(0)}
>> 2
! Division by zero.
<to be read again>

1.35 . . .)transformed p=(2/(x-x),
                                  3/0)transformed p;
You're trying to divide the quantity shown above the error
message by zero. I'm going to divide it by one instead.

! Division by zero.
1.35 . . .ansformed p=(2/(x-x),3/0
                                  )transformed p;
I'll pretend that you meant to divide by 1.

{((2,3))transformed((xpart p,ypart p,xxpart p,xypart p,yxpart p,yypart p
))}
{((linearform,linearform))=((linearform,linearform))}
## yxpart p=-yypart p
## xxpart p=-xypart p
{((-xpart p7+6,-ypart p7+12))=((linearform,linearform))}
## ypart p7=-ypart p-yypart p+12
## xpart p7=-xpart p-xypart p+6
{\}
{if}
{string(unknown string p~[-1])}
{true}
{(p0.1 0.2)-(p0.1 0.2)}
! The token 'endtext' is no longer a right delimiter.
1.36 . . .1.2-p.1.199999,1 endtext
                                  transformed p;
Strange: This token has lost its former meaning!
I'll read it as a right delimiter this time;
but watch out, I'll probably miss it later.

{((0,1))transformed((xpart p,ypart p,-xypart p,xypart p,-yypart p,yypart
 p))}
{((linearform,linearform))=((linearform,linearform))}
## ypart p=-yypart p+6
#### y-part p7=6
## xpart p=-xypart p+3
#### xpart p7=3
```

```
{(unknown path p1 2p)=((3,6))}
{showstopping:=0}
{showvariable}
p=(-xypart p+3,-yypart p+6,-xypart p,xypart p,-yypart p,yypart p)
p Cl=pair
p[][]=numeric
p Cl Cl p=unknown path
p[][]p~=macro:(SUFFIX2)(SUFFIX3)(EXPR4)<suffix>->begingroup ETC.
p[]~=unknown boolean
p~=path
p~[]=unknown string
p~[]~=unknown pen
p~[]~[][]=unknown picture
p~[-1]=unknown string p~[-1]
p~1~=pen
p[[ Cl ]]=numeric
p[[ [-1] ]]=4095.99998
p0.1 0.2=p0.1 0.2
p1 2p=path
p7=(3,6)
{((-xypart p+3,-yypart p+6,-xypart p,xypart p,-yypart p,yypart p))=((-xy
part p+3,-yypart p+6,-xypart p,xypart p,-yypart p,yypart p))}
{let}
{let}
{xxpart((-xypart p+3,-yypart p+6,-xypart p,xypart p,-yypart p,yypart p))
3
{(-xypart p)+(0.002)}
{yxpart((-xypart p+3,-yypart p+6,-xypart p,xypart p,-yypart p,yypart p))
3
{((xpart p2,ypart p2))-((xpart p1,ypart p1))}
{(1)*((linearform,linearform))}
{((xpart p1,ypart p1))+((linearform,linearform))}
{(y)+(0.00002)}
{yypart((-xypart p+3,-yypart p+6,-xypart p,xypart p,-yypart p,yypart p))
3
{xypart((-xypart p+3,-yypart p+6,-xypart p,xypart p,-yypart p,yypart p))
,
{((5,y))=((yypart p,xypart p))}
## y=xypart p
## yypart p=5
#### ypart p=1
#### yxpart p=-5
{((5,xypart p+0.00002))=((5,xypart p))}
(((xpart p2,ypart p2))=((5,xypart p))}
## ypart p2=xypart p
## xpart p2=5
{((-xypart p+0.002,-5))=((5,xypart p))}
## xypart p=-5
#### ypart p2=-5
#### y=-5
#### xpart p=8
#### xxpart p=5
! Inconsistent equation (off by -0.002).
<to be read again>
                   ,
1.38...,y)=(yypart p,xypart p);

The equation I just read contradicts what was said before.
But don't worry; continue and I'll just ignore it.

Path at line 39, before choices:
(0,0)..controls (15,4) and (-15,-12)
 ..(4,0)
 ..cycle

Path at line 39, after choices:
(0,0)..controls (15,4) and (-15,-12)
 ..(4,0)..controls (17.52783,8.54388) and (-15.45978,-4.12262)
 ..cycle

{reverse(path)}
```

```
{(path)transformed((8,1,5,-5,-5,5))}
{(path)=(unknown path p2 3p)}

p1 2p~(SUFFIX2)(SUFFIX3)(EXPR4)<suffix>->begingroup.show(SUFFIX0);p(SUFF
IX2)(SUFFIX3)((EXPR4)endgroup
(SUFFIX0)<-p1 2p
(SUFFIX1)<-~
(SUFFIX2)<-
(SUFFIX3)<-2 3p~
! A primary expression can't begin with 'right delimiter that matches ('

<inserted text>
                0
<to be read again>
                   ,
1.40 . . .000000001]2p~(,[2]3p~,-)
                                  =p~1~2[pausing];
I'm afraid I need some sort of value in order to continue,
so I've tentatively inserted '0'. You may want to
delete this zero and insert something else;
see Chapter 27 of The METAFONTbook for an example.

{-(0)}
(EXPR4)<-0
(SUFFIX5)<-
{begingroup}
{show}
>> Path at line 40:
(3,6)


p2 3p~(SUFFIX2)(SUFFIX3)(EXPR4)<suffix>->begingroup.show(SUFFIX0);p(SUFF
IX2)(SUFFIX3)((EXPR4)endgroup
(SUFFIX0)<-p2 3p
(SUFFIX1)<-~
! Missing ',' has been inserted.
<to be read again>
                   (0)
p1 2p~->...IX2)(SUFFIX3)((EXPR4)
                                  endgroup
<to be read again>
                   =
1.40 . . .00000001]2p~(,[2]3p~,-)=
                                  p~1~2[pausing];
I've finished reading a macro argument and am about to
read another; the arguments weren't delimited correctly.
You might vant to delete some tokens before continuing.

(SUFFIX2)<-
! Hissing ',' has been inserted.
Cto be read again>
                   (0)
pl 2p~->...IX2)(SUFFIX3)((EXPR4)
                                  endgroup
Cto be read again>
                   =
1.40 . . .00000001]2p~(,[2]3p~,-)=
                                  p~1~2[pausing];
I've finished reading a macro argument and am about to
read another; the arguments weren't delimited correctly.
You might want to delete some tokens before continuing.

(SUFFIX3)<-
! Missing ')' has been inserted.
<to be read again>
                   endgroup
<to be read again>
                   =
1.40 . . .00000001]2p~(,[2]3p~,-)=
                                  p~1~2[pausing];
I've gotten to the end of the macro parameter list.
```

```
You might want to delete some tokens before continuing.

(EXPR4)<-0
(SUFFIX5)<-
{begingroup}
{show}
>> Path at line 40:
(8,1)..controls (-48.68579,57.68579) and (52.91974,-43.91974)
 ..(28,-19)..controls (-7,16) and (63,-54)
 ..cycle

>> (8,1,5,-5,-5,5)
! Isolated expression.
<to be read again>
                   `
p2 3p~->...;p(SUFFIX2)(SUFFIX3)(
                                  (EXPR4)endgroup
<to be read again>
                   endgroup
<to be read again>
                   =
l.40 ...00000001]2p~(,[2]3p~,-)=
                                  p~1~2[pausing];
I couldn't find an '=' or ':=' after the
expression that is shown above this error message,
so I guess I'll just ignore it and carry on.

! Extra tokens will be flushed.
<to be read again>
                   (
p2 3p~->...;p(SUFFIX2)(SUFFIX3)(
                                  (EXPR4)endgroup
<to be read again>
                    endgronp
Cto be read again>
                   =
1.40 . . .00000001]2p~(,[2]3p~,-)=
                                  p~1~2[pausing];
I've just read as much of that statement as I could fathom,
so a semicolon should have been next. It's very puzzling...
but I'll try to get myself back together, by ignoring
everything up to the next ';'. Please insert a semicolon
now in front of anything that you don't want me to delete.
(See Chapter 27 of The METAFONTbook for an example.)

{endgroup}
{endgroup}
{(vacuous)=(unknown picture p~1~2 1)}
>> vacuous
>> unknown picture p~1~2 1
! Equation cannot be performed (vacuous=unknown picture).
<to be read again>

1.40 . ..213p-. -)=p~1~2[pausing];

I'm sorry, but I don't know hoo to make such things equal.
(See the too expressions just above the error message.)

{vardef}
Runaway  definition?
if.p(SUFFIX1)(SUFFIX1)=(SUFFIX1)(SUFFIX1)p.fi
! Forbidden token found while scanning the definition of p~[].
<inserted  text>
                enddef
<to be read again>
                   ,
1.41 . ..iary t:=if p@ @=@ @p fi;
                                   vardef p[][]p~[]=BAD; inner ;;
I suspect you have forgotten an 'enddef',
causing me to read past where you wanted me to atop.
I'll try to recover; but if the error is serious,
```

```
you'd better type 'E' or 'X' now and fix your file.

{vardef}
! This variable already starts with a macro.
1.41 ...@p fi; vardef p[][]p~[]=
                                BAD; inner ;;
After 'vardef a' you can't say 'vardef a.b'.
So I'll have to discard this definition.

Runaway definition?
BAD
! Forbidden token found while scanning the definition of a bad variable.
<inserted text>
                enddef
<to be read again>

1.41 ...i; vardef p[][]p~[]=BAD;
                                 inner ;;
I suspect you have forgotten an 'enddef',
causing me to read past where you wanted me to atop.
I'll try to recover; but if the error is serious,
you'd better type 'E' or 'X' now and fix your file.

{inner}
{show}
{-(2)}

p~[-2]@@<tertiary>->begingroup.if.p(SUFFIX1)(SUFFIX1)=(SUFFIX1)(SUFFIX1)
p.fi.endgroup
(SUFFIX0)<-p~
(SUFFIX1)<-[-2]
(SUFFIX2)<-~
! A tertiary expression can't begin with '['.
(inserted text>
                0
<to be read again>
                   [
<to be read again>
                   (3000)
<to be read again>

1.42 show p~[-2]~[3000,
                      x]++4000>path p3; showvariable p,P;
I'm afraid I need some sort of value in order to continue,
so I've tentatively inserted '0'. You may want to
delete this zero and insert something else;
see Chapter 27 of The METAFONTbook for au example.

{(x)-(3000)}
{(0)*(x-3000)}
{(3000)+(0)}
{(3000)++(4000)}
(EXPR3)<-4999.99998
{begingroup}
{if}
{(p[-2][-2])=(-2)}
>> pc-21 [-2]+2
! Unknown relation will be considered false.
<to be read again>
                   c-21
p~[-2]->...1)=(SUFFIX1)(SUFFIX1)
                                p.fi.endgroup
<to be read again>
                   >
1.42 show p~[-2]~[3000,x]++4000>
                                path p3; ahoavariable p,P;
Oh dear. I can't decide if the expression above is positive,
negative, or zero. So this comparison teat won't be 'true'.

(false)
! Hissing ':' has been inserted
```

```
<to be read again>
                    c-21
p~[-2]->...1)=(SUFFIX1)(SUFFIX1)
                                p.fi.endgroup
<to be read again>
                   >
1.42 show p~[-2]~[3000,x]++4000>
                                path p3; ahoovariable p,P;
There should've been a colon after the condition.
I shall pretend that one was there.

{endgroup}
{path((xpart p3,ypart p3))}
{(vacuous)>(false)}
>> vacuous
>> false
! lot implemented: (vacuous)>(boolean).
<to be read again>
                   ,
1.42 ...~[3000,x]++4000>path p3;
                                 showvariable p,P;
I'm afraid I don't know how to apply that operation to that
combination of types. Continue, and I'll return the second
argument (see above) as the result of the operation.

>> false
{showvariable}
p-(8,1,5,-5,-5,5)
p Cl=pair
p[][]=numeric
p Cl Cl p=unknown path
p[][]p~=macro:(SUFFIX2)(SUFFIX3)(EXPR4)<suffix>->begingroup ETC.
p[]~=unknown boolean
p~=path
p~[]@#=macro:<tertiary>->begingroup.if.p(SUFFIX1)(SUFFIX1) ETC.
p[[ Cl ]]=numeric
p[[ c-11 ]]=4095.99998
p[-2] c-21 =p c-21 c-21
p0.1 0.2=p0.1 0.2
p1=(xpart p1,ypart p1)
p1 2p=path
p2=(5,-5)
p2 3p=path
p3=(xpart p3,ypart p3)
p7=(3,6)
> P=tag
{numeric}
{(2)*(alpha)}
{(p3~)=(2alpha)}
## alpha=0.5p3~
{(1)/(-1)}
{(3)*(beta)}
{(p[-1]~)=(3beta)}
## beta=0.33333p[-1]~
{begingroup}
{save}
{showvariable}
> p=tag
{(3)*(0.33333(SAVED)p[-1]~)}
{((SAVED)p[-1]~)=(1)}
## (SAVED)p[-1]~=1
#### beta=0.33333
{restoring p}
{endgroup}
{showvariable}
p=(8,1,5,-5,-5,5)
p Cl=pair
p[][]=numeric
p Cl Cl p=unknown path
p[][]p~=macro:(SUFFIX2)(SUFFIX3)(EXPR4)<suffix>->begingroup ETC.
p Cl~=numeric
```

```
p~=path
p~[]@$=macro:<tertiary>->begingroup.if.p(SUFFIX1)(SUFFIX1) ETC.
p[[ Cl ]]=numeric
p[[ C-l] ]]=4095.99998
p_c-21[-2]=pc-21 f-21
p[-1]~=1
p0.1 0.2=p0.1 0.2
p1=(xpart p1,ypart p1)
p1 2p=path
p2=(5,-5)
p2 3p=path
p3=(xpart p3,ypart p3)
p3~=p3~
p7=(3,6)
{def}
{def}
! Hissing '=' has been inserted.
<to be read again>
                   false
1.45 . . .enddef;def!primary!false
                                ):!fi enddef;
The next thing in this 'def' should have been '=',
because I've already looked at the definition heading.
But don't worry; I'll pretend that au equals sign
was present. Everything from here to 'enddef'
will be the replacement text of this macro.

{def}
{(path)scaled(-1)3
Path at line 46, before choices:
(1,-2)..controls (1.37755,-1.71404) and (1.71404,-1.37755)
 ..(2,-1)..controls (2.33353,-0.55965) and (2.59729,-0.04124)
 ..(2.5,0.5)..controls (2.36812,1.23369) and (1.6712,1.65662)
 ..(1,2)..controls (0.66821,2.16974) and (0.33485,2.33641)
 ..(0,2.5)
 ..(-1,2)..controls (-1.37755,1.71404) and (-1.71404,1.37755)
 ..(-2,1)..controls (-2.33353,0.55965) and (-2.59729,0.04124)
 ..(-2.5,-0.5)..controls (-2.36812,-1.23369) and (-1.6712,-1.65662)
 ..(-1,-2)..controls (-0.66821,-2.16974) and (-0.33485,-2.33641)
 ..(0,-2.5)
 ..cycle

Path at line 46, after choices:
(1,-2)..controls (1.37755,-1.71404) and (1.71404,-1.37755)
 ..(2,-1)..controls (2.33353,-0.55965) and (2.59729,-0.04124)
 ..(2.5,0.5)..controls (2.36812,1.23369) and (1.6712,1.65662)
 ..(1,2)..controls (0.66821,2.16974) and (0.33485,2.33641)
 ..(0,2.5)..controls (-0.37186,2.68167) and (-0.668,2.25146)
 ..(-1,2)..controls (-1.37755,1.71404) and (-1.71404,1.37755)
 ..(-2,1)..controls (-2.33353,0.55965) and (-2.59729,0.04124)
 ..(-2.5,-0.5)..controls (-2.36812,-1.23369) and (-1.6712,-1.65662)
 ..(-1,-2)..controls (-0.66821,-2.16974) and (-0.33485,-2.33641)
 ..(0,-2.5)..controls (0.37186,-2.68167) and (0.668,-2.25146)
 ..cycle

{makepen(path)}
Pen polygon at line 46 (newly created):
(1,-2)
 .. (2,-1)
 .. (2.5,0.5)
 .. (1,2)
 .. (0,2.5)
 .. (-1,2)
 .. (-2,1)
 .. (-2.5,-0.5)
 .. (-1,-2)
 .. (0,-2.5)
 .. cycle

{qq:=pen}
{primarydef}
```

```
{secondarydef}

//<expr>->

//<expr>->
{pencircle}
{length(path)}
{(future pen)slanted(1)}
{((3,6))-((5,-5))}

_aa__<secondary>->if(true
{(0.1)*(15)}
{odd(1.50009)}
{not(false)3
{known((8,1,5,-5,-5,5))}
{(true)and(true)}
(EXPR0)<-true
{if}
{true}

!<primary>->false):(EXPR0)fi
(EXPR0)<-(5,-5)
{false}
{(true)or(false)}
{true}
{fi}
{-((5,-5))}
{-((5,-5))}
{+(1)}
{-(1)}
>> -1
! Improper curl has been replaced by 1.
<to be read again>
                   )
1.51 {curl- +1)
                ..tension atleaatl..cycle sqrt2++sqrt2***[[]];
A curl must be a known, nonnegative number.

! Hissing '}' has been inserted.
Cto be read again>
                   )
1.51 (curl- +1)
                ..tension atleaatl..cycle sqrt2++sqrt2***[[]];
I've scanned a direction spec for part of a path,
so a right brace should have come next.
I shall pretend that one was there.

Path at line 51, before choices:
(-5,5)

Path at line 51, after choices:
(-5,5)

Path at line 51, before choices:
(3,6)..controls (5,-5) and (-5,5)
 ..(-5,5)..tension atleaatl
 ..{2896.30943,-2896.30934}cycle

Path at line 51, after choices:
(3,6)..controls (5,-5) and (-5,5)
 ..(-5,5)..controls (-3.29726,7.86205) and (0.64516,8.35484)
 ..cycle

! Hissing ')' has been inserted.
<to be read again>
                   sqrt
1.51 . . . ion atleast1..cycle sqrt
                              2++sqrt2***[[]];
I found no right delimiter to match a left one. So I've
put one in, behind the scenes; this may fix the problem.
```

```
{((-2,11))subpath(path)}
{reverse(path)}
{makepen(path)}

**->[[show(EXPR0)*(EXPR1)]]
(EXPR0)<-future pen
(EXPR1)<-future pen
{begingroup}
{show}
{(future pen)*(future pen)}
>> future pen
>> future pen
! Rot implemented: (future pen)*(future pen).
<to be read again>
                   ]]
<to be read again>
                   aqrt
1.51 . . . ion atleast1..cycle aqrt
                                  2++sqrt2***[[]];
I'm afraid I don't know how to apply that operation to that
combination of types. Continue, and I'll return the second
argument (see above) as the result of the operation.

! Pen path mat be a cycle.
<to be read again>
                   ]]
<to be read again>
                   sqrt
1.51 . . . ion atleast1..cycle aqrt
                                  2++sqrt2***[[]];
I can't make a pen from the given path.
So I've replaced it by the trivial path '(0,0)..cycle'.

>> Pen polygon at line 51:
(0,0)
 .. cycle

{endgroup}
(EXPR0)<-vacuous
{sqrt(2)}
{sqrt(2)}
{(1.41422)++(1.41422)}
{begingroup}
{endgroup}

**->expandafter(EXPR1)scantokens"**oct"(EXPR0)
(EXPR0)<-2
(EXPR1)<-vacuous
{expandafter}
{scantokens}
{oct(2)}
>> 2
! lot implemented: oct(known numeric).
<to be read again>
                   ;
1.51 . . .cle sqrt2++sqrt2***[[]];

I'm afraid I don't know how to apply that operation to that
particular type. Continue, and I'll simply return the
argument (shown above) as the result of the operation.


**->[[show(EXPR0)*(EXPR1)]]
(EXPR0)<-vacuous
(EXPR1)<-2
{begingroup}
{show}
{(vacuous)*(2)}
>> vacuous
>> 2
! lot implemented: (vacuous)*(known numeric).
```

```
<to be read again>
                   ]]
<to be read again>

1.51 . . .cle sqrt2++sqrt2***[[]];

I'm afraid I don't know how to apply that operation to that
combination of types. Continue, and I'll return the second
argument (see above) as the result of the operation.

>> 2
{endgroup}
(EXPR0)<-vacuous
{begingroup}
{interim)
{-(20.5)}
{charcode:=-20.5}
{proofing:=-20.5}
{-(2048)}
{chardp :=-2048)
{shipout}
{nullpicture}
! Enormous cherdp has been reduced.
<to be read again>
                   ]]
1.52 . . .48;shipout nullpicture]]
                              ;
Font metric dimensions mat be leas then 2048pt.

{restoring proof ing=0}
{endgroup}
{if}
{-(275.50002)}
{charexists(-275.50002)}
{known(unknown path p0 0p)}
{(true)>(false)}
{known(path)}
{(true)=(true)}
{true}
{randomseed}
! Hissing ':=' has been inserted.
<to be read again>
                   charcode
1.53 . . . p-: randomseed charcode
                                 ; fi
Always say 'randomseed:=<numeric expression>'.

{randomseed:=-20.5}
{fi}
{randomseed}
>> "goof"
! Unknown value will be ignored.
<to be read again>
                   ;
1.54 randomseed:="goof";
                          a[($,18++1+-+18),(2,3)]=b[(3,2),(1,$);
Your expression was too random for me to handle,
so I won't change the random seed just now.

{(18)++(1)}
{(18.02776)+-+(18)}
{((2,3))-((-1,1))}
{(a)*((3,2))}
{((-1,1))+((3a,2a))}
! Hissing ']' has been inserted.
<to be read again>
                   ,
1.54 . . .8),(2,3)]=b[(3,2),(1,$);

I've scanned an expression of the form 'a[b,c',
80 a right bracket should have come next.
```

```
I shall pretend that one was there.

{((1,-1))-((3,2))}
{(b)*((-2,-3))}
{((3,2))+((-2b,-3b))}
{((3a-1,2a+1))=((-2b+3,-3b+2))}
## b=-0.66667a+0.33333
## a=2
#### b--l
{show}
{(^)+(1)}
{(~)+(2)}
{-(1)}
{((^+1,~+2))slanted(-1)}
{-(2)}
{((linearform,~+2))yscaled(-2)}
{-((3,4))}
{((linearform,-2~-4))zscaled((-3,-4))}
>> (-5~-3^-13,10~-4^+16)
{((xpart pp,ypart pp,xxpart pp,xypart pp,yxpart pp,yypart pp))xscaled(9)
3
{((8,1,5,-5,-5,5))transformed((9xpart pp,ypart pp,9xxpart pp,9xypart pp.
yxpart pp,yypart pp)) 3
>> (9xpart pp+72xxpart pp+9xypart pp,ypart pp+8yxpart pp+yypart pp,45xxp
art pp-45xypart pp,-45xxpart pp+45xypart pp,5yxpart pp-5yypart pp,-5yxpa
rt pp+5yypart pp)
{((xpart pp,ypart pp,xxpart pp,xypart pp,yxpart pp,yypart pp))shifted((1
,2))}
{((8,1,5,-5,-5,5))transformed((8,1,5,-5,-5,5))}
{((xpart pp+1,ypart pp+2,xxpart pp,xypart pp,yxpart pp,yypart pp))transf
ormed((43,-34,50,-50,-50,50))}
>> (50xpart pp-50ypart pp-7,-50xpart pp+50ypart pp+16,50xxpart pp-50yxpa
rt pp,50xypart pp-50yypart pp,-50xxpart pp+50yxpart pp,-50xypart pp+50yy
part pp)
{(2)-(1)}
{(_0)*(1)}
{(1)+(_0)}
{-(_0+1)}
>> -_0-1
{show}
{normaldeviate}
1-C-204813
{uniformdeviate(2048)}
{angle((-6.41438,761.67789))}
{cosd(90.03117)}
{(200)*(-0.00055)}
{mexp(-0.10986)}
Csqrt(0.99957) 3
{mlog(0.99979)}
{sind(-0.05469)}
{floor(-0.00096)}
>> -1
{string}
{(unknown string s2)=(unknown string s4)}
{(unknown string s1)=(unknown string s2)}
{(unknown string s3)=(unknown string s5)}
{(unknown string s2)=(unknown string s4)}
! Redundant equation.
<to be read again>

1.58 . . . s1=s2=s4; s3=s5; s1=s2;
                                  if s1<=s4<>(s1<>s3):show[[ch...
I already knew that this equation was true.
But perhaps no harm has been done; let's continue.

{if}
{(unknown string s2)<=(unknown string s1)}
{(unknown string s2)<>(unknown string s5)}
>> unknown string a2
>> unknown string a5
! Unknown relation will be considered false.
```

```
<to be read again>
                   )
1.58 ...=s2; if s1<=s4<>(s1<>s3)
                                   :show[[char34=s2:=s3]]fi;
The quantities shown above have not been equated.

{(true)<>(false)}
{true}
{show}
{begingroup}
{char(34)}
{s2:=unknown string s5}
{("""")=(unknown string s2)}
{endgroup}
{fi}
>> vacuous
{(0.1)point(path)}
{begingroup}
{pencircle}
Pen polygon at line 59 (newly created):
(0.5,0)
 .. (0,0.5)
 .. (-0.5,0)
 .. (0,-0.5)
 .. cycle

{endgroup}
{((3.24413,0.64801))penoffset(pen)}
{((0,-0.5))rotated(540)}
{((0,0.5))/(0.33333)}
{-(1.5)}
! Paths don't touch; `&' will be changed to `..'.
<to be read again>

1.60 . . .tcontrol-1.5of(p&cycle)
                                   -precontrol1/2of p(p)=s1:=s...
When you join paths `p&q', the ending point of p
must be exactly equal to the starting point of q.
So I'm going to pretend that you said `p..q' instead.

Path at line 60, before choices:
(0,0)..controls (15.4) and (-15,-12)
 ..(4,0)
 ..cycle

Path at line 60, after choices:
(0,0)..controls (15,4) and (-15,-12)
 ..(4,0)..controls (17.52783,8.54388) and (-15.45978,-4.12262)
 ..cycle

{(-1.5)postcontrol(path)}
{(0.5)precontrol(path)}
{((-2.75,-5))-((3.75,-1))}
! Hissing `of' has been inserted for directiontime.
<to be read again>
                   (
1.60 . . .cle)-precontrol1/2of p(
                                    p)=s1:=s4:=s4;
I've got the first argument; will look now for the other.

{((-6.5,-4))directiontime(path)}
{decimal(0.5)}
{((0,1.50003))substring("0.5")}
{s4:=unknown string s1}
{s1:=unknown string s4}
{("0.")=(unknown string s1)}
{path}
{length("  ")}
! Hissing `,' has been inserted.
<to be read again>
                   }
```

```
1.61 . ..p~[]; p~1=p2{length" "}
                                &cycle; p~1=p2=p~0; p2..contr...
I've got the x coordinate of a path direction;
will look for the y coordinate next.

! An ● zpreaaion can't begin with '}'.
<inserted text>
                0
<to be read again>
                }
1.61 . ..p~[]; p~1=p2{length" "}
                                &cycle; p~1=p2=p~0; p2..contr...
I'm afraid I need aoae sort of value in order to continue,
• o I've tentatively inserted '0'. You may want to
delete this zero end insert aoaething else;
see Chapter 27 of The METAFONTbook for an example.

Path at line 61, before choicer:
(5,-5){4096,0}
 ..{4096,0}cycle

Path at line 61, after choices:
(5,-5)..controls (5,-5) and (5,-5)
 ..cycle

{(unknown path p~1)=(path)}
{((5,-5))=(unknown path p~0)}
{(path)=(path)}
! Redundant or inconsistent equation.
<to be read again>

1.61 . .."  ")&cycle; p~1=p2=p~0;
                                 p2..controls-p2..cycle=p~2;(p7
An equation between already-known quantities can't help.
But don't worry; continue end I'll just ignore it.

{-((5,-5))}
Path at line 61, before choices:
(5,-5)..controls (-5,5) end (-5,5)
 ..cycle

Path at line 61, after choices:
(5,-5)..controls (-5,5) and (-5,5)
 ..cycle

{(path)=(unknown path p~2)}
{length(path)}
Path at line 62, before choices:
(3,6)..tension 1.2
 ..(5,-5)..controls (5,-5) and (5,-5)
 ..(5,-5)
 ..(5,-5)..controls (-5,5) and (-5,5)
 ..(5,-5){0,4096}..tension 1 and atleastl
 ..cycle

Path at line 62, after choices:
(3,6)..controls (-3.01212,4.82085) end (-1.0424,-6.01257)
 ..(5,-5)..controls (5,-5) and (5,-5)
 ..(5,-5)..controls (5,-5) and (5,-5)
 ..(5,-5)..controls (-5,5) and (-5,5)
 ..(5,-5)..controls (5,0.02791) and (4.99951,6.39217)
 ..cycle

>> x
! Improper tension has been set to 1.
<to be read again>

1.63 x..
         {curl1}-p7{curl hex "IsBad"}..tension.75and.74999..p...
The expression above should have been a number > 3/4.
```

```
{-((3,6))}
{hex("IsBad")}
>> "IsBad"
! String contains illegal digits.
<to be read again>
                    3
1.63 . . .l1}-p7{curl hex "IsBad"}
                                  ..tension.75and.74999..p2{0,1...
I zeroed out characters that weren't hex digits.

>> 0.74998
! Improper tension has been set to 1.
<to be read again>

1.63 . . .}..tension.75and.74999..
                                  p2{0,1}&p2{_,'}..cycle:=p
The expression above should have been a number > 3/4.

>> _
! Undefined x coordinate has been replaced by 0.
<to be read again>
                    ,
1.63 . . .and.74999..p2{0,1}&p2{_,
                                  '3..cycle:=p
I need a 'known' x value for this part of the path.
The value I found (see above) was no good;
so I'll try to keep going by using zero instead.
(Chapter 27 of The METAFONTbook explains that.
you might want to type 'I ???' now.)

>> (xpart ',ypart ')
! Undefined y coordinate has been replaced by 0.
<to be read again>
                    3
1.63 . . .d.74999..p2{0,1}&p2{_,'}
                                  ..cycle:=p
I need a 'known' y value for this part of the path.
The value I found (see above) was no good;
so I'll try to keep going by using zero instead.
(Chapter 27 of The METAFONTbook explains that
you might want to type 'I ???' now.)

Path at line 63, before choices:
(3,6)..controls (-3.01212,4.82085) and (-1.0424,-6.01257)
 ..(5,-5)..controls (5,-5) and (5,-5)
 ..(5,-5)..controls (5,-5) and (5,-5)
 ..(5,-5)..controls (-5,5) and (-5,5)
 ..(5,-5)..controls (5,0.02791) and (4.99951,6.39217)
 ..(3,6)
 ..{curl 1}(-3,-6){curl 2989}..tension 0.75 and 1
 ..{0,4096}(5,-5){curl 13
 ..cycle

Path at line 63, after choices:
(3,6)..controls (-3.01212,4.82085) and (-1.0424,-6.01257)
 ..(5,-5)..controls (5,-5) and (5,-5)
 ..(5,-5)..controls (5,-5) and (5,-5)
 ..(5,-5)..controls (-5,5) and (-5,5)
 ..(5,-5)..controls (5,0.02791) and (4.99951,6.39217)
 ..(3,6)..controls (-2.44821,4.93144) and (-5.41408,-1.00029)
 ..(-3,-6)..controls (-3.70045,-12.19485) and (5,-11.61679)
 ..(5,-5)..controls (12.45406,-3.75087) and (10.41669,7.45464)
 ..cycle

>> path
! Improper ':=' will be changed to '='.
1.63 . . .p2{0,1}&p2{_,'}..cycle:=
                              P
I didn't find a variable name at the left of the ':=',
so I'm going to pretend that you said '=' instead.
```

```
{(path)=(unknown path p~4)}
{-(9)}
{((3.5001,7.00002))subpath(path)}
{((-9,9))subpath(path)}
{(path)=(unknown path p~6)}
{show}
>> Path at line 86:
(-2.5,2.5)..controls (-2.49893,2.49893) and (0.00107,-0.00107)
 ..(5,-5)..controls (5,0.02791) and (4.99951,6.39217)
 ..(3,6)..controls (-2.44821,4.93144) and (-5.41408,-1.00029)
 ..(-3,-6)..controls (-3.70045,-12.19485) and (5,-11.61679)
 ..(5,-5)..controls (5.0001,-4.99998) and (5.00021,-4.99997)
 ..(5.00032,-4.99995)

{((1,2))directiontime(path)}
>> 1
{-(1.00002)}
{((1,-1.00002))directiontime(path)}
>> 3.24937
Path at line 88, before choices:
(0,0)..controls (1,1) and (0,1)
 ..(1,0)

Path at line 88, after choices:
(0,0)..controls (1,1) and (0,1)
 ..(1,0)

{(unknown path p~3)=(path)}
{show}
Path at line 88, before choices:
(5,-5){-2896.30943,-2896.30934}
 ..{1831.78674,3663.57385}cycle

Path at line 88, after choices:
(5,-5)..controls (5,-5) and (5,-5)
 ..cycle

>> Path at line 86:
(5,-5)..controls (5,-5) and (5,-5)
 ..cycle

{((0.31416,1))subpath(path)}
{((1,1))directiontime(path)}
{(1)-(0.31416)}
{(0.27097)*(0.68584)}
{(0.31416)+(0.18584)}
>> 0.5
{((0,0.25))subpath(path)}
{((0.25,1))subpath(path)}
Path at line 68, before choices:
(0,0)..controls (0.25,0.25) and (0.375,0.4375)
 ..(0.4375,0.5625)..controls (0.625,0.9375) and (0.25,0.75)
 ..(1,0)

Path at line 88, after choices:
(0,0)..controls (0.25,0.25) and (0.375,0.4375)
 ..(0.4375,0.5625)..controls (0.625,0.9375) and (0.25,0.75)
 ..(1,0)

! Hissing `endtext' has been inserted.
<to be read again>

1.88  ..~3)shifted begintext1,0;

I found no right delimiter to match a left one. So I've
put one in, behind the scenes; this may fix the problem.

{(path)shifted((1,0))}
{(unknown path p~5)=(path)}
{(2/3)*((xpart ',ypart '))}
{((0.66667xpart ',0.66667ypart '))zscaled((xpart ',ypart '))}
```

```
>> (0,0,xpart ',-ypart ',ypart ',xpart ')
! Transform components aren't all known.
<to be read again>
                   {
1.89 p~3:=2/3'zscaled'{
                        p~3}..controls(2,2/3(3))and penoffset(...
I'm unable to apply a partially specified transformation
except to a fully known pair or transform.
Proceed, and I'll omit the transformation.

>> 0.66667xpart '
! Undefined x coordinate has been replaced by 0.
<to be read again>
                   {
1.89 p~3:=2/3'zscaled'{
                        p~3}..controls(2,2/3(3))and penoffset(...
I need a 'known' x value for this part of the path
The value I found (see above) was no good;
so I'll try to keep going by using zero instead.
(Chapter 27 of The METAFONTbook explains that
you might want to type 'I ???' now.)

>> 0.66667ypart '
! Undefined y coordinate has been replaced by 0.
<to be read again>
                   {
1.89 p~3:=2/3'zscaled'{
                        p~3}..controls(2,2/3(3))and penoffset(...
I need a 'known' y value for this part of the path.
The value I found (see above) was no good;
so I'll try to keep going by using zero instead.
(Chapter 27 of The METAFONTbook explains that
you might want to type 'I ???' now.)

>> path
! Undefined coordinates have been replaced by (0,0).
<to be read again>
                   3
1.89 p~3:=2/3'zscaled'{p~3}
                             ..controls(2,2/3(3))and penoffset(...
I need x and y numbers for this part of the path.
The value I found (see above) was no good;
so I'll try to keep going by using zero instead.
(Chapter 27 of The METAFONTbook explains that
you might want to type 'I ???' now.)

{(2/3)*(3)}
{(1/2)*(x)}
{((0.5x,-5))penoffset((0,1))}
>> (0.5x,-5)
>> (0,1)
! lot implemented: penoffset(unknown pair)of(pair).
<to be read again>

1.69 . . .enoffset(1/2x,y)of(0,1)(
                                  1,0);
I'm afraid I don't know how to apply that operation to that
combination of types. Continue, and I'll return'the second
argument (see above) as the result of the operation.

! Hissing '..' has been inserted.
<to be read again>
                   \
1.89 . . .enoffset(1/2x,y)of(0,1)(
                                  1,0);
A path join command should end with two dots.

Path at line 89, before choices:
(0,0)..controls (2,2) and (0,1)
 ..(1,0)
```

```
Path at line 69, after choices:
(0,0)..controls (2,2) and (0,1)
 ..(1,0)

{p~3:=path}
{show}
{reverse(path)}
{(path)intersectiontimes(path)}
>> (0.17227,0.28339)
{(0.17227)point(path)}
>> (0.71329,0.78188)
{(1)-(0.28339)}
{(0.71661)point(path)}
>> (0.7133,0.78189)
{show}
{(path)shifted((0.01,0))}
{(path)intersectiontimes(path)}
{xpart((1.47693,1.18973))}
{(1.47693)point(path)}
{(path)shifted((0.01,0))}
{(path)intersectiontimes(path)}
{ypart((1.47693,1.18973))}
{(1.18973)point(path)}
{((1.50499,0.71521))-((1.495,0.71521))}
>> (0.01,0)
{begingroup}
{interim}
{tracingedges:=1}
{-(1)}
{-(1.00002)}
{(-1)+-+(-1.00002)}
! Pythagorean subtraction 1+-+1.00002 has been replaced by 0.
1.73  ...es:=1;e[-1+-+ -1.000011
                                  =nullpicture;
Since I don't take square roots of negative numbers,
I'm zeroing this one. Proceed, with fingers crossed.

{nullpicture}
{(unknown picture e0)=(picture)}
{addto}
{(path)scaled(3)}
Path at line 74, before subdivision into octants:
(3,0)..controls (3.75,0.75) and (4.125,1.3125)
 ..(4.3125,1.6875)..controls (4.875,2.8125) and (3.75,2.25)
 ..(6,0)..controls (3.75,2.25) and (4.875,2.8125)
 ..(4.3125,1.6875)..controls (4.125,1.3125) and (3.75,0.75)
 ..cycle

Cycle spec at line 74, aftersubdivision:
(3,0) % beginning in octant 'NNE'
   ..controls (3.75,0.75) and (4.125,1.3125)
 ..(4.3125,1.6875) % segment 0
   ..controls (4.5,2.0625) and (4.5,2.25)
 ..(4.5,2.25) % segment 1
% entering octant 'NNW'
% entering octant 'WNW'
% entering octaut 'WSW'
% entering octant 'SSW'
% entering octsnt 'SSE'
   ..controls (4.5,2.25) and (4.5,1.5)
 ..(6,0) % segment 1
% entering octant 'ESE'
% entering octant 'ENE'
% entering octsnt 'NNE'
% entering octsnt 'NNW'
   ..controls (4.5,1.5) and (4.5,2.25)
 ..(4.5,2.25) % segment 2
% entering octant 'WNW'
% entering octaut 'WSW'
% entering octsnt 'SSW'
    ..controls (4.5,2.25) and (4.5,2.0625)
```

```
 ..(4.3125,1.6875) % segment 2
   ..controls (4.125,1.3125) and (3.75,0.75)
 ..(3,0) % segment 3
% entering octant 'SSE'
% entering octant 'ESE'
% entering octant 'ENE'
 & cycle

Tracing edges at line 74: (weight 1)
(4,0)(4,1)(5,1)(5,3)(4,3)(4,1)(5,1)(5,0)(6,0)(6,1)(5,1)(5,3)
(4,3)(4,1)(3,1)(3,0).

{(unknown picture e1)=(unknown picture e2)}
{(picture)=(unknown picture e1)}
{cull}
{show}
{-(4095)}
{(picture)shifted((4089,-4095))}
>> Edge structure at line 75:
row -4093: | 4093+ 4094-
row -4094: | 4093+ 4094-
row -4095: | 4092+ 4093- 4094+ 4095-

{-(4095)3
{(picture)shifted((-4095,4092))}
{-(3)}
{(picture)shifted((-3,0))}
>> Edge structure at line 75:
row 4094: -4094+ -4093- -4094+ -4093- |
row 4093: -4094+ -4093- -4094+ -4093- |
row 4092: -4095+ -4092- -4093+ -4094- |

{-(4095)}
{(picture)shifted((4089,-4095))}
! Too far to shift.
<to be read again>
                   ]]
1.76 e2 shifted(4089,-4095)]]
                                ;addto e1 also e2 shifted(-2,$)...
I can't shift the picture as requested---it would
make some coordinates too large or too small.
Proceed, and I'll omit the transfonsation.

>> Edge structure at line 76:
row 2: 4+ 5- 4+ 5- |
row 1: 4+ 5- 4+ 5- I
row 0: 3+ 6- 5+ 4- |

{restoring  tracingedgesm32787.99998)
{endgroup}
{addto}
{-(2)}
{(picture)shifted((-2,-1))}
{-(4)}
{(picture)shifted((-4,-1))}
{e1:=picture}
{addto}
{(picture)rotated(89.999)}
! That transformation is too hard.
<to be read again>
                   +
1.77 . . .e0 also e1rotated89.999+
                                    e1scaled$; show e0 xscaled-10...
I can apply complicated transformations to paths,
but I can only do integer operations on pictures.
Proceed, and I'll omit the transformation.

{(picture)scaled(-1)}
{(picture)+(picture)}
{show}
{-(10)}
```

```
{(picture)xscaled(-10)}
{(picture)yscaled(2)}
{(picture)xscaled(82)}
! Scaled picture would be too big.
<to be read again>
                   yscaled
1.78 yscaled
           883;addto el doublepath (0,9) sithaeight-3 withwe...
I can't xscale the picture as requested---it would
make some coordinates too large or too small.
Proceed, and I'll omit the transformation.

{(picture)yscaled(683)}
! Scaled picture would be too big.
<to be read again>

1.78 yscaled683;
               addto el doublepath (0,9) oithoeight-3 withue...
I can't yscale the picture as requested---it would
make some coordinates too large or too small.
Proceed, and I'll omit the transformation.

>> Edge structure at line 78:
row 5: -40- -50+ -40- -50+ I
row 4: -40- -50+ -40- -50+ I
row 3: -30+ o- -10+ -20- -40- -50+ -40- -50+ | -10+ 0-
row 2: -30+ o- -10+ -20- -40- -50+ -40- -50+ | -10+ 0-
row 1: 20- 10+ 20- 10+ -20+ -10- -20+ -10- -30- -60+ -50- -40+ |
 -10+ -10+ o- o- 10+ 20-
row 0: 20- 10+ 20- 10+ -20+ -10- -20+ -10- -30- -60+ -50- -40+ |
 -10+ -10+ o- o- 10+ 20-
row -1: 20- 10+ 20- 10+ -20+ -10- -20+ -10- I -20+ -10- o+ 0+ 10-
 10-
row -2: 20- 10+ 20- 10+ -20+ -10- -20+ -10- I -20+ -10- o+ o+ 10-
 10-
row -3: 30- 0+ 10- 20+ I 0+ 10-
row -4: 30- 0+ 10- 20+ I 0+ 10-

{addto}
{-(3)}
{turningnumber(path)}
! Weight must be -3, -2, -1, +1, +2, or +3.
<to be read again>
                   withpen
1.79 oithpen
             pencircle xscaled(oct"180"++1) rotated-angle(64...
I'll ignore the bad 'with' clause and look for another.

{pencircle}
{oct("180")}
>> "180"
! String contains illegal digits.
<to be read again>
                      ++
1.79 . . . ircle xscaled(oct"180"++
                                    1) rotated-angle(64,$) shifte...
I zeroed out characters that weren't in the range 0..7.

{(64)++(1)}
{(future pen)xscaled(64.00781)}
{angle((64,-1))}
{-(-0.89517)}
{(future pen)rotated(0.89517)}
((future pen)shifted((9,8))}
Pen polygon at line 80 (newly created):
(9,7.5)
 .. (37.5,8)
 .. (41,8.5)
 .. (9,8.5)
 .. (-19.5,8)
 .. (-23,7.5)
```

```
 .. cycle

Path at line 80, before choices:
(-1,-1){curl 13
 ..(1,0)
 ..(1,1)
 ..(-1,0)
 ..{curl 1}cycle

Path at line 80, after choices:
(-1,-1)..controls (-0.18568,-1.14464) and (0.62712,-0.73824)
 ..(1,0)..controls (1.16457,0.32584) and (1.21637,0.71506)
 ..(1,1)..controls (0.46898,1.69931) and (-0.7587,1.2167)
 ..(-1,0)..controls (-1.06546,-0.33012) and (-1.06546,-0.66988)
 ..cycle

{makepen(path)}
{(future pen)xscaled(4095.49998)}
Pen polygon at line 80 (newly created):
(4095.49998,0)
 .. (4095.49998,1)
 .. (-4095.49998,0)
 .. (-4095.49998,-1)
 .. cycle

Path at line 80, before subdivision into octants:
(0,9)..controls (0,9) and (0,9)
 ..cycle

! Curve out of range.
<to be read again>

1.80 . . .cycle)xscaled4095.49999;

At least one of the coordinates in the path I'm about to
digitize was really huge (potentially bigger than 4095).
So I've cut it back to the maximum size.
The results will probably be pretty wild.

Cycle spec at line 80, after subdivision:
(0,0) % beginning in octant 'ENE'
% entering octant 'NNE'
% entering octant 'NNW'
% entering octant 'WNW'
% entering octant 'WSW'
% entering octant 'SSW'
% entering octant 'SSE'
% entering octant 'ESE'
 & cycle

Tracing edges at line 80: (weight -3)
@ Octant ENE (2 offsets), from (-4095.49998,-1) to (4095.49998,0)
@ transition line 0, from (-4095.49998,-1) to (-4095.49998,-1)
@ transition line 1, from (-4095.49998,-1) to (4095.49998,0)
@ transition line 2, from (4095.49998,0) to (4095.49998,0)
(0,-1)
@ Octant NNE (1 offset), from (4095.49998,0) to (4095.49998,1)
@ transition line 2, from (4095.49998,0) to (4095.49998,0)
@ transition line 1, from (4095.49998,0) to (4095.49998,1)
(0,0)(4095,0)
@ Octant NNW(0 offsets), from (4095.49998,1) to (4095.49998,1)
@ transition line 0, from (4095.49998,1) to (4095.49998,1)
@ Octant WNW(0 offsets), from (4095.49998,1) to (4095.49998,1)
@ transition line 1, from (4095.49998,1) to (4095.49998,1)
@ Octant WSW(2 offsets), from (4095.49998,1) to (-4095.49998,0)
@ transition line 0, from (4095.49998,1) to (4095.49998,1)
@ transition line 1, from (4095.49998,1) to (-4095.49998,0)
@ transition line 2, from (-4095.49998,0) to (-4095.49998,0)
(4095,1)(0,1)
@ Octant SSW(1 offset), from (-4095.49998,0) to (-4095.49998,-1)
@ transition line 2, from (-4095.49998,0) to (-4095.49998,0)
```

```
@ transition line 1, from (-4095.49998,0) to (-4095.49998,-1)
(0,0)(-4095,0)
@ Octsnt SSE (0 offsets), from (-4095.49998,-1) to (-4095.49998,-1)
@ transition line 0, from (-4095.49998,-1) to (-4095.49998,-1)
@ Octsnt ESE (0 offsets), from (-4095.49998,-1) to (-4095.49998,-1)
@ transition line 1, from (-4095.49998,-1) to(-4095.49998,-1)
(-4095,-1).

{show}
>> Edge structure at line 81:
row 1: I 0+ 1-
row 0: 0--- 4095+++ -2+ -1- -2+ -1- I 0+ 1-
row -1: -4095--- 0+++ -2+ -1- -2+ -1- | -1+ 0- 1+ 2-
row -2: -3+ 0- -1+ -2- I

{totalweight(picture)}
>> -0.37478
{charcode:=5}
{chardp:=5}
{-(1.5)}
{xoffset:=-1.5}
{shipout}
[5]
Edge structure at line 81 (just shipped oat):
row 1: I -1+ 0-
row 0: I-3+ -3+ -2- -2- -1--- -1+ 0- 4094+++
row -1: I -4096--- -3+ -3+ -2- -2- -2+ -1- -l+++ 0+ 1-
row -2: I -4+ -3- -2+ -1-

{showstats}
Memory usage 1066&170 (706 still untouched)
String usage 18&67 (1004&11979 still untouched)
{addto}
{(path)yscaled(0.01666)}
Path at line 82, before subdivision into octants:
(0,0)..controls (15,0.06665) and (-15,-0.19995)
 ..(4,0)..controls (-15,-0.19995) and (15,0.06665)
 ..cycle

Cycle spec at line 82, after subdivision:
(0,0) % beginning in octant 'ENE'
   ..controls (1.74272,0.00775) and (2.87802,0.01099)
 ..(3.55333,0.01099) % segment 0
% entering octant 'ESE'
   ..controls (4.12251,0.01099) and (4.36491,0.00868)
 ..(4.36877,0.00482) % segment 0
% entering octant 'SSE'
   ..controls (4.3688,0.00479) and (4.36882,0.00476)
 ..(4.36882,0.00473) % segment 0
% entering octant 'SSW'
   ..controls (4.36882,0.0047) and (4.3688,0.00467)
 ..(4.36877,0.00464) % segment 0
%entering octant 'WSW'
   ..controls (4.34875,-0.01538) and (-1.88759,-0.07579)
 ..(-2.48894,-0.07579) % segment 0
%entering octant 'WNW'
   ..controls (-2.5191,-0.07579) and (-2.5351,-0.07564)
 ..(-2.53542,-0.07532) % segment 0
% entering octant 'NNW'
% entering octant 'NNE'
% entering octant 'ENE'
   ..controls (-2.53209,-0.07199) and (-0.88672,-0.05144)
 ..(4,0) % segment 0
% entering octant 'NNE'
%entering octant 'NNW'
%entering octant 'WNW'
% entering octant 'WSW'
   ..controls (-0.88672,-0.05144) and (-2.5321,-0.072)
 ..(-2.53542,-0.07532) % segment 1
% entering octant 'SSW'
% entering octant 'SSE'
```

```
% entering octant 'ESE'
   ..controls (-2.53511,-0.07562) and (-2.5191,-0.07579)
 ..(-2.48894,-0.07579) % segment 1
% entering octant 'ENE'
   ..controls (-1.88759,-0.07579) and (4.34877,-0.01537)
 ..(4.36877,0.00464) % segment 1
% entering octant 'NNE'
   ..controls (4.3688,0.00467) and (4.36882,0.0047)
 ..(4.36882,0.00473)% segment 1
% entering octant 'NNW'
   ..controls (4.36882,0.00476) and (4.3688,0.00479)
 ..(4.36877,0.00482) % segment 1
% entering octant 'WNW'
   ..controls (4.36491,0.00868) and (4.12251,0.01099)
 ..(3.55333,0.01099) % segment 1
% entering octant 'WSW'
   ..controls (2.87802,0.01099) and (1.74272,0.00775)
 ..(0,0) % segment 1
% entering octant 'SSW'
% entering octant 'SSE'
% entering octant 'ESE'
 & cycle

Tracing edges at line 82: (weight 1)
(No new edges added.)

{(picture)yscaled(0)}
{e3:=picture}
{autorounding:=2}
{addto}
Path at line 83, before choices:
(0.5,0){curl 13
 ..{curl 1}(3.5,1.5)

Path at line 83, after choices:
(0.5,0)..controls (1.5,0.5) and (2.5,1)
 ..(3.5,1.5)

Path at line 83, before subdivision into octants:
(0.5,0)..controls (1.5,0.5) and (2.5,1)
 ..(3.5,1.5)..controls (2.5,1) and (1.5,0.5)
 ..cycle

Cycle spec at line 83, after subdivision and double autorounding:
(0.5,0) % beginning in octant 'ENE'
   ..controls (1.5,0.5) and (2.5,1)
 ..(3.5,1.5) % segment 0
% entering octant 'NNE'
% entering octant 'NNW'
% entering octant 'WNW'
% entering octant 'WSW'
   ..controls (2.5,1) and (1.5,0.5)
 ..(0.5,0)% segment 1
% entering octant 'SSW'
% entering octant 'SSE'
% entering octant 'ESE'
 & cycle

Tracing edges at 1ine 83: (weight 2)
(2,0)(2 1)(4,1)(4,2)(4,1)(2,1)(2,0).

{tracingspecs:=0}
Path at line 84, before choices:
(1,1)
 ..cycle

Path at line 84, after choices:
(1,1)..controls (1,1) and (1,1)
 ..cycle

{makepen(path)}
```

```
{(future pen)yscaled(1.5)}
Pen polygon at line 84 (newly created):
(1,1.5)
 .. cycle

{q:=pen}
{((-1,yy))rotated((8,1,5,-5,-5,5))}
>> (8,1,5,-5,-5,5)
! Improper transformation argument.
<to be read again>
                   {
1.85 p~8=(($,yy)rotated pi
                              0,1}..{0,$}(1,0){0,$}..cycle)scaled...
The expression shown above has the wrong type,
so I can't transform anything using it.
Proceed, and I'll omit the transformation.

>> yy
! Undefined y coordinate has been replaced by 0.
<to be read again>
                   {
1.86 p~8=(($,yy)rotated p{
                              0,1}..{0,$}(1,0){0,$}..cycle)scaled...
I need a 'known' y value for this part of the path.
The value I found (see above) was no good;
so I'll try to keep going by using zero instead.
(Chapter 27 of The METAFONTbook explains that
you might want to type 'I ???' now.)

Path at line 85, before choices:
(-1,0){0,4096}
 ..{0,-4096}(1,0){0,-4096}
 ..{0,4096}cycle

Path at line 85, after choices:
(-1,0)..controls (-1,1.33333) and (1,1.33333)
 ..(1,0)..controls (1,-1.33333) and (-1,-1.33333)
 ..cycle

{(path)scaled(2)}
{(path)shifted((1000.49,9))}
((unknown path p~8)=(path)}
{turningcheck:=1}
{addto}
>> (8,1,5,-5,-5,5)
! Improper type.
<to be read again>
                   sithpen
1.86 . . .n q aithoeight p oithpen
                                  cycle p;
Next time say 'withweight <known numeric expression>';
I'll ignore the bad 'with' clause and look for another.

{cycle((8,1,5,-5,-5,5))}
>> false
! Improper type.
<to be read again>

1.88 . . .eight p aithpen cycle p;

Bert time say 'withpen <known pen expression>'
I'll ignore the bad 'with' clause and look for another.

Tracing edges at line 88: (weight 1)
@ Octsnt ENE (0 offsets), from (999,10.5) to (999.15451,11.65451)
@ transition line 1, from (999.15451,11.65451) to (999.15451,11.65451)
(999,111
@ Octant ENE (0 offsets), from (999.15451,11.65451) to (1001,12)
@ transition line 0, from (999.15451,11.65451) to (999.15451,11.65451)
@ Octant ESE (0 offsets), from (1001,12) to (1002.8455,11.6545)
@ transition line 1, from (1002.8455,11.6545) to (1002.8455,11.6545)
```

```
@ Octant SSE (0 offsets), from (1002.8455,11.6545) to (1003,10.5)
@ transition line 0, from (1002.8455,11.6545) to (1002.8455,11.6545)
(999,12)(1003,12)
@ Octant SSW(0 offsets), from (1003,10.5) to (1002.84549,9.34549)
@ transition line 1, from (1002.84549,9.34549) to (1002.84549,9.34549)
@ Octant WSW(0 offsets), from (1002.84549,9.34549) to (1001,9)
@ transition line 0, from (1002.84549,9.34549) to (1002.84549,9.34549)
@ Octant WNW (0 offsets), from (1001,9) to (999.1545,9.3455)
@ transition line 1, from (999.1545,9.3455) to (999.1545,9.3455)
@Octant NNW(0 offsets), from (999.1545,9.3455) to (999,10.5)
@ transition line 0, from (999.1545,9.3455) to (999.1545,9.3455)
(1003,9)(999,9)(999,11).

Tracing edges at line 88: (weight 1)
@ Octant SSE (0 offsets), from (999,10.5) to (999.15451,9.34549)
@ transition line 0, from (999,10.5) to (999,10.5)
(999,111
@ Octant ESE (0 offsets), from (999.15451,9.34549) to (1001,9)
@ transition line 1, from (1001,9) to (1001,9)
@ Octsnt EIE (0 offsets), from (1001,9) to (1002.8455,9.3455)
@ transition line 0, from (1001,9) to (1001,9)
@ Octant NNE (0 offsets), from (1002.8455,9.3455) to (1003,10.5)
@ transition line 1, from (1003,10.5) to (1003,10.5)
(999,9)(1003,9)
@Octant NNW(0 offsets), from (1003,10.5) to (1002.84549,11.65451)
@ transition line 0, from (1003,10.5) to (1003,10.5)
@Octsnt WNW(0 offsets), from (1002.84549,11.65451) to (1001,121
@ transition line 1, from (1001,121 to (1001,121
@ Octant WSW(0 offsets), from (1001,12) to (999.1545,11.6545)
@ transition line 0, from (1001,121 to (1001,121
@ Octant SSW(0 offsets), from (999.1545,11.6545) to (999,10.5)
@ transition line 1, from (999,10.5) to (999,10.5)
(1003,12)(999,12)(999,11).

{begingroup}
(interim)
{(xx)=(0.1)}
%% xx=0.1
{autorounding:=0.1}
{addto}
Tracing edges at line 87: (weight -2)
@ Octant NNE (0 offsets), from (999,10.5) to (999.38197,11.42706)
@ transition line 1, from (999.38197,11.42706) to (999.38197,11.42706)
@ Octsnt EIE (0 offsets), from (999.38197,11.42706) to (1001,121
@ transition line 0, from (999.38197,11.42706) to (999.38197,11.42706)
(999,111
@ Octant ESE (0 offsets), from (1001,12) to (1002.61804,11.42705)
@ transition line 1, from (1002.61804,11.42705) to (1002.61804,11.42705)
(999,12)(1003,12)
@ Octant SSE (0 offsets), from (1002.61804,11.42705) to (1003,10.5)
@ transition line 0, from (1002.61804,11.42705) to (1002.61804,11.42705)
@ Octant SSW(0 offsets), from (1003,10.5) to (1002.61803,9.57294)
@ transition line 1, from (1002.61803,9.57294) to (1002.61803,9.57294)
@ Octsnt WSW(0 offsets), from (1002.61803,9.57294) to (1001,9)
@ transition line 0, from (1002.61803,9.57294) to (1002.61803,9.57294)
@Octsnt WNW(0 offsets), from (1001,9) to (999.38196,9.57295)
@ transition line 1, from (999.38196,9.57295) to (999.38196,9.57295)
(1003,9)(999,9)
@ Octant NNW (0 offsets), from (999.38196,9.57295) to (999,10.5)
@ transition line 0, from (999.38196,9.57295) to (999.38196,9.57295)
(999,111.

{restoring autorounding=2}
{endgroup}
{-(14.5)}
{charext:=-14.5}
{chardp:=-14.5}
{chardx:=-14.5}
{shipout}
(-(picture))
{(picture)-(picture))
```

```
{- (picture) 3
{turningnumber(path)}
{(6)*(-1)}
{(picture)shifted((0,-6))}
{(picture)+(picture)3
[5.-14]
Edge structure at line 88 (just shipped out):
row 11: I 998- 998+ 998++ 1002-- 1002- 1002+
row 10: I 998- 998+ 998++ 1002-- 1002- 1002+
row 9: I 998- 998+ 998++ 1002-- 1002- 1002+
row 5: | 998- 998+ 998++ 1002-- 1002- 1002+
row 4: | 998- 998+ 998++ 1002-- 1002- 1002+
row 3: I 998- 998+ 998++ 1002-- 1002- 1002+
roo2: | 3+ 3+ 3+ 3+ 4- 4- 4- 4-
row 1: I -1+ -1+ o- o- 1+ 2- 3+ 3+ 3+ 3+ 4- 4- 4- 4-
row 0: I -3+ -3+ -3+ -2- -2- -2- -1+ -1+ o- o- 0+ o+ 1- 1- 2+ 2+
 3- 3- 4+ 4+ 5- 5-
row -1: | -3+ -3+ -2- -2- -2+ -2+ -1- -1- 0+ 0+ 0+ 1- 1- 1-
row -2: | -4+ -3- -2+ -2+ -1- -1-
row-4: | 3+ 3+ 4- 4-
row -5: | 3+ 3+ 4- 4-
row-8: | 2+ 3- 4+ 5-

Path at line 89, before choices:
(0,0){curl 13
 ..(1,0.5)
 ..(5,1.5)
 ..(7,2.5)
 ..(12,3.5)
 ..{curl 1}(13,4)

Path at line 89, after choices:
(0,0)..controls (0.3153,0.20053) and (0.6504,0.36807)
 ..(1,0.5)..controls (2.29114,0.98723) and (3.72412,0.97192)
 ..(5,1.5)..controls (5.68967,1.78545) and (6.30771,2.22081)
 ..(7,2.5)..controls (8.58867,3.1407) and (10.3946,2.90645)
 ..(12,3.5)..controls (12.3506,3.62962) and (12.68594,3.7973)
 ..(13,4)

{(unknown path p~9)=(path)}
{addto}
Tracing edges at line 89: (weight 1)
(1,0)(1,1)(5,1)(5,2)(7,2)(7,3)(12,3)(12,4)(12,3)(7,3)(7,2)(5,2)
(5,1)(1,1)(1,0).

{smoothing:=1}
{addto}
Tracing edges at line 90: (weight 1)
(1,0)(1,1)(4,1)(4,2)(8,2)(8,3)(12,3)(12,4)(12,3)(8,3)(8,2)(4,2)
(4,1)(1,1)(1,0).

{addto}
{-(4095)}
Path at line 91, before choices:
(-4095,0){curl 13..tension 0.76 and 999
 ..{curl 1}(0,2)

Path at line 91, after choices:
(-4095,0)..controls (-2275,0.88889) and (-1.36636,1.99933)
 ..(0,2)

Tracing edges at line 91: (weight 1)
(-3071,0)(-3071,1)(-1024,1)(-1024,2)(-1024,1)(-3071,1)(-3071,0).

{show}
{-(90)}
{(picture)rotated(-90)}
>> Edge structure at line 91:

{(picture)+(picture)}
{(90)*(-1)}
```

```
{(picture)rotated(-90)}
{(picture)rotated(90)}
>> Edge structure at line 91:
row 2: |  4+++ 4+ 5--- 5-
row 1: |  0+++ 0+ 1--- 1- 2++ 3-- 4+++ 4+ 5--- 5-
row 0: |  -2+++ -2+++ -1--- -1--- 0+++ 0+ 2--- 2- 3++ 4-- 5++ 6--
row -1: |  -2+++ -2+ 0--- 0- 1+++ 1+++ 2--- 2---
row -2: |  -3++ -2-- -1+++ -1+ 0--- 0-

{if}
{if}
{elseif}
! Missing ':' has been inserted.
<inserted text>

<to be read again>
                   elseif
1.92 if "a" if "ab">"b" elseif
                                path reverse (3,4): >="aa":foo

{("ab")>("b")}
Ifalse}
{reverse((3,4))}
{path(path)}
{true}
{("a")>=("aa")}
{false)
{if}
{((xpart ',ypart '))-((1,yy))}
{((xpart '-1,linearform))<((xpart ',ypart '))}
{true}
{fi}
{else}
! Extra else.
1.93 . . . if'-(1,yy)<'::fi else
                                    def dup text t=[[t;save endd
I'm ignoring this; it doesn't match any if.

{def}
{def}
{def}

I|<tertiary>->show.substring(EXPR0)of("a"
(EXPR0)<-(2,-1)
{show}
{("a")&("bc")}
{((2,-1))substring("abc")}
>> "ba"
{tertiarydef}
{def}
{show}
{((23.3,4.5))subpath(path)}
Path at line 100, before choices:
(0,0){curl 23
 ..(13,4)..controls (12.84297,3.89865) and (12.68063,3.80606)
 ..(12.51372,3.72261){curl 33
 ..{curl 4}(-1,-1){curl 43
 ..{curl 2}cycle

Path at line 100, after choices:
(0,0)..controls (-40.15552,-36.7088) and (58.71173,33.50317)
 ..(13,4)..controls (12.84297,3.89865) and (12.68063,3.80606)
 ..(12.51372,3.72261)..controls (8.00914,2.1484) and (3.50458,0.5742)
 ..(-1,-1)..controls (-0.66667,-0.66667) and (-0.33333,-0.33333)
 ..cycle

>> Path at line 100:
(0,0)..controls (-40.15552,-36.7088) and (58.71173,33.50317)
 ..(13,4)..controls (12.84297,3.89865) and (12.68063,3.80606)
 ..(12.51372,3.72261)..controls (8.00914,2.1484) and (3.50458,0.5742)
 ..(-1,-1)..controls (-0.66667,-0.66667) and (-0.33333,-0.33333)
```

```
..cycle

{numspecial}
{(2)+(3)}

++->[[dup.showtoken(EXPR0);]];def.x.expr.z.of(EXPR1)=z.enddef;texts(x=((
EXPR1)+0)(EXPR1)+(EXPR1))("xx",foo((EXPR0)))=0]]
(EXPR0)<-path
(EXPR1)<-5
{begingroup}

dup<text>->[[(TEXT0);save
(TEXT0)<-showtoken(path)
{begingroup}
{showtoken}
> (path)
{save}
{restoring ; }
{endgroup}
{def}
! Hissing symbolic token inserted.
<inserted text>
                 INACCESSIBLE
++-> . . .]];def.x.expr.z.of(EXPR1)
                                  =z.enddef;texts(x=((EXPR1)+0)...
<to be read again>

1.100 . . .cle;numspecial p~++2+3;
                                    cc
Sorry: You can't redefine a number, string, or expr.
I've inserted an inaccessible symbol so that your
definition will be completed without mixing me up too badly.


texts(TEXT0)(TEXT1)<expr>->for.n:=,for.n"yy":n,length.if.false:endfor(TE
XT1),(TEXT0),:if.string.n:forsuffixes.n=foo1,[foo(n)],':show(TEXT0),(TEX
T1)|(n;exitif.not('<='+((EXPR2),yy))endfor.for.m= :+endfor.for.m=alpha.s
top-l.1 3$:+m.endfor.fi.endfor
(TEXT0)<-x=((5)+0)(5)+(5)
(TEXT1)<-"xx",foo((path))
(EXPR2)<-0
{for3
{for}
! Hissing '=' has been inserted.
<to be read again>
                  "yy"
texts->for.n:=,for.n"yy"
                       :n,length.if.false:endfor(TEXT1),(TEX...
<to be read again>
                  ]]
<to be read again>

1.100 . . .cle;numspecial p~++2+3;
                                     cc
The next thing in this loop should hare been '=' or ':='
But don't worry; I'll pretend that an equals sign
was present, and I'll look for the values next.

{loop value="yy"}
{if}
{false}
{false}
! Incomplete if; all text was ignored after line 100.
<inserted text>
                fi
<to be read again>
                   EIDFOR
<for("yy")> . . . .if.false:EIDFOR

texts->. . .length.if.false:endfor
                                  (TEXT1),(TEXT0),:if.string.n:...
```

```
<to be read again>
                   11
<to be read again>

1.100 . . .cle;numspecial p~++2+3;
                                  [[
A forbidden 'outer' token occurred in skipped text.
This kind of error happens when you say 'if...' and forget
the matching 'fi'. I've inserted a 'fi'; this might work.

{[repeat the loop]3
{length("xx")}

foo(TEXT2)->begingroup(TEXT2)endgroup
(SUFFIX0)<-
(SUFFIX1)<-foo
(TEXT2)<-(path)
{begingroup}
{endgroup}

x<expr>of<primary>->(EXPR0)
{(5)+(0)}
(EXPR0)<-5
! Hissing 'of' has been inserted for x.
<to be read again>
                  (5)
<argument> x=((5)+0)(5)
                       +(5)
texts-). . .:endfor(TEXT1),(TEXT0)
                                  ,:if.string.n:forsuffixes.n=f...
<to be read again>
                   11
<to be read again>

1.100 . . .cle;numspecial p~++2+3;
                                  CC
I've got the first argument; will look now for the other.

(EXPR1)<-5
{(5)+(5)}
{loop value="yy"}
{if}
{string("yy")}
{true}
{forsuffixes}
! Hissing symbolic token inserted.
<inserted text>
                INACCESSIBLE
<for("yy")>. . . orsuffixes(EXPR0)
                                  =foo1,[foo((EXPR0))],':show.x...
texts->. . .3$:+m.endfor.fi.endfor

<to be read again>
                   11
<to be read again>

1.100 . . . cle;numspecial p~++2+3;
                                  [[
Sorry: You can't redefine a number, string, or expr.
I've inserted an inaccessible symbol so that your
definition will be completed without mixing me up too badly.


foo(TEXT2)->begingroup(TEXT2)endgroup
(SUFFIX0)<-
(SUFFIX1)<-foo
(TEXT2)<-("yy")
{begingroup}
{endgroup}
>> "yy"
 ! Improper subscript has been replaced by zero.
```

```
<for("yy")>...o1,[foo((EXPR0))]
                                  ,':show.x=((5)+0)(5)+(5),"xx"...
texts->..  .3$:+m.endfor.fi.endfor

<to be read again>
                    13
<to be read again>

1.100 . . .cle;numspecial p~++2+3;
                                  CC
A bracketed subscript must have a known numeric value;
unfortunately, what I found was the value that appears just
above this error message. So I'll try a zero subscript.

{loop value=foo1}
{show}

x<expr>of<primary>->(EXPR0)
{(5)+(0)}
(EXPR0)<-5
! Hissing 'of' has been inserted for x.
<to be read again>
                    (5)
<for(foo1)> show.x=((5)+0)(5)
                               +(5),"xx",foo((path))|(("yy");ex...
<for("yy")>. . . '+((0),yy))endfor
                                  .for.m= :+endfor.for.m=alpha....
texts-).  . .3$:+m.endfor.fi.endfor

<to be read again>
                    11
<to be read again>
                    ;
1.100 . . .cle;numspecial p~++2+3;
                                  [[
I've got the first argument; will look now for the other.

(EXPR1)<-5
{(5)+(5)}
>> 10
>> "xx"
foo(TEXT2)->begingroup(TEXT2)endgroup
(SUFFIX0)<-
(SUFFIX1)<-foo
(TEXT2)<-(path)
{begingroup}
{endgroup}

|<suffix>->,(SUFFIX0)
! Hissing ')' has been inserted.
<to be read again>
                    ("yy")
<for(foo1)>. . .o((path))|(("yy")
                                  ;exitif.not('<='+((0),yy))El...
<for("yy")>. . . '+((0),yy))endfor
                                  .for.m= :+endfor.for.m=alpha....
texts->.  . .3$:+m.endfor.fi.endfor

<to be read again>
                    11
<to be read again>
                    ,
1.100 . . .cle;numspecial p~++2+3;
                                  [[
I've gotten to the end of the macro parameter list.
You might want to delete some tokens before continuing.

(SUFFIX0)<-
>> Path at line 100:
(0,0)..controls (15,4) and (-15,-12)
 ..(4,0)
```

```
>> "yy"
{exitif}
{((xpart ',ypart '))+((0,yy))}
{((xpart ',ypart '))<=((xpart ',linearform))}
>> -yy
! Unknown relation will be considered false.
<to be read again>
                    )
<for(foo1)>. . .ot('<='+((0),yy))
                                    EIDFOB
<for("yy")>. . . '+((0),yy))endfor
                                   .for.m= :+endfor.for.m=alpha....
texts-). . .3$:+m.endfor.fi.endfor

<to be read again>
                   11
Cto be read again>

1.100 . . .cle;numspecial p~++2+3;
                                   cc
Oh dear. I can't decide if the expression above is positive,
negative, or zero. So this comparison test won't be 'true'.

{[repeat the loop])
{loop value=0}
{not(false)}
{true}
{for}
{for}
>> 0.5p3~
! Improper initial value has been replaced by 0.
<to be read again>
                   step
<for("yy")>. . . .for.m=alpha.step
                                  -1.1 3$:+m.endfor.fi EIDFOR
texts->.. .3$:+m.endfor.fi.endfor

<to be  read again>
                   11
<to be read again>

1.100 . . .cle;numspecial p~++2+3;
                                   [[
When you say 'for x=a step b until c',
the initial value 'a' and the step size 'b'
and the final value 'c' must have known numeric values.
I'm zeroing this one. Proceed, with fingers crossed.

{-(1.1)}
! Hissing 'until' has been inserted.
<to be read again>
                   3
<for("yy")>. . .=alpha.step-1.1 3
                                   $:+m.endfor.fi EEDFOR
texts-). . .3$:+m.endfor.fi.endfor

<to be read again>
                   11
<to be read again>
                   ;
1.100 . . .cle;numspecial p~++2+3;
                                   [[
I assume you meant to say 'until' after 'step'.
So I'll look for the final value and colon next.

{(3)*(-1)}
(loop value=0}
{[repeat the loop])
{loop value=-1.1}
{+(0)3
```

```
{[repeat the loop]}
{loop  value=-2.200013
{(0)+(-1.1)}
{[repeat the loop]}
{fi}
{[repeat the loop]}
{loop value-23
{if}
{string(2)}
{false}
{[repeat the loop]}
{loop value=path}
{if}
{string(path)}
{false}
{[repeat the loop]}
{loop value=10}
{if}
{string(10)}
{false}
([repeat the loop13
{(-1.1)+(-2.20001)}
{endgroup}
{begingroup}
{let}
{save}
{def}
{def}

texts->input
{input}
(trap.mf
if->endinput?
{endinput}
{if}
{known(0)}
{not(true)}
{false}
)
! Incomplete if; all text was ignored after line 2.
(inserted text>
                fi
1.101 . . . input enddef;texts trap
                                  11;
The file ended while I was skipping conditional text.
This kind of error happens when you say 'if...' and forget
the matching 'fi'. I've inserted a 'fi'; this might work.

{restoring \}
{restoring if3
{endgroup}

dup<text>->[[(TEXT0);save
Runaway text?
[[def.texts.secondary.x=primarydef.y++y=x@y.enddef;showtokenETC.
! Forbidden token found while scanning a text argument.
(inserted text>
                endgroup
<to be read again>

1.102  ..showtoken++;x enddef]]\
                                 ;
It seems that a right delimiter was left out,
causing me to read past where you wanted me to atop
I'll try to recover; but if the error is serious,
you'd better type 'E' or'I' now and fix your file.

(TEXT0)<-[[def.texts.secondary.x=primarydef.y++y=x@y.enddef;showtoken++;
x.enddef]]
{begingroup}
{begingroup}
```

```
{def}
{endgroup}
{save}
! Missing symbolic token inserted.
<inserted text>
                INACCESSIBLE
<to be read again>

1.102 . . .showtoken++;x enddef]]\

Sorry: You can't redefine my error-recovery tokens.
I've inserted an inaccessible symbol so that your
definition will be completed without mixing me up too badly.

{\}
(proofing :=1}

texts<secondary>->primarydef.y++y=(EXPR0)@y.enddef;showtoken++;(EXPR0)
(EXPR0)<-"a"
{primarydef}
{showtoken}
> ++=primarydef'd macro:
("a")@(EXPR1)
{("a")&("b")}
ab
texts<secondary>->primarydef.y++y=(EXPR0)@y.enddef;showtoken++;(EXPR0)
{(1.00002)*(a1)}
{-(1.00002a1)}
(EXPR0)<--1.00002a1
{primarydef}
{showtoken}
> ++=primarydef'd macro:
(-1.00002a1)@(EXPR1)
{(a2)+(a3)}
{(linearform)+(a4)}
{(linearform)+(a5)}
{(linearform)+(a6)}
{-(linearform)}
{(-1.00002a1)=(linearform)}
## a1=0.99998a6+0.99998a5+0.99998a4+0.99998a3+0.99998a2
{-((a3,a2))}
{((-a3,-a2))/(0.99998)}
{(a4)+(a5)}
{begingroup}
{showdependencies}
a1=0.99998a6+0.99998a5+0.99998a4+0.99998a3+0.99998a2
alpha=0.5p3~
{endgroup}
{(linearform)+(a6)}
{(a3)+(a4)}
{(linearform)+(a5)}
{(linearform)+(a6)}
{-((linearform,linearform))}
{((-1.00002a3,-1.00002a2))=((linearform,linearform))}
## a2=0.99998a6+0.99998a5+0.99998a4+0.99998a3
## a3=0.99998a6+0.99998a5+0.99998a4
(restoring INACCESSIBLE}
{endgroup}
{(1.00002)*(0.25a4*4)}
{(0.25a4*4)+(1)}
{(0.25a5*4)+(0.25a6*4)}
{(0.25a4*4+1)=(linearform)}
## a4*4=0.99998a6*4+0.99998a5*4-3.99994
{(0.9)*(0.5p3~)}
{(0.45p3~)+(7)}
{alpha:=0.45p3~+7}
## alpha=0.45p3~+7
{showdependencies}
alpha=0.45p3~+7
a4=0.25a6*4+0.25a5*4-0.99998
a3=0.49998a6*4+0.49998a5*4-0.99997
```

```
a2=0.99997a6*4+0.99997a5*4-1.99992
a1=1.99992a6*4+1.99992a5*4-3.99982
{(0.66667)-(0.25a6*4)}
{(0.25a6*4)=(-0.25a6*4+0.66667)}
## a6*4=1.33334
{(0.25a5*4)=(0.33333)}
## a5*4=1.33331
#### a4=-0.33333
#### a3=0.33333
#### a2=0.66664
#### a1=1.33328
{cull}
{(4)*(0.33333)}
{chardp:=27}
{charcode:=27}
{openwindow}
! Improper 'openwindow'.
<to be read again>
                   ,
1.107 . . .) to (0,0) at "whoops";
                                  addto p; ahipout p; cull p;
Say 'openwindow k from (r0,c0) to (r1,c1) at (x,y)',
where all quantities are known and k is between 0 and 15.

{addto}
>> (8,1,5,-5,-5,5)
! lot a suitable variable.
<to be read again>

1.107 . . .) at "whoops"; addto p;
                                  ahipout p; cull p;
At this point I needed to see the name of a picture variable.
(Or perhaps you have indeed presented me with one; I might
hare missed it, if it wasn't followed by the proper token.)
So I'll not change anything just now.

{shipout}
! Variable p is the wrong type (transform).
<to be read again>

1.107 . ..a". addto p; ahipout p;
                                  cull p;
I was looking for a "known" picture variable
So I'll not change anything just now.

{cull}
>> (8,1,5,-5,-5,5)
! lot a suitable variable.
<to be read again>
                   ;
1.107 . ..o p; ahipout p; cull p;

At this point I needed to see the name of a picture variable.
(Or perhaps you have indeed presented me with one; I might
hare missed it, if it wasn't followed by the proper token.)
So I'll not change anything just now.

{openwindow}
i-to.513

Calling BLANKRECTANGLE(0,0,0,0)
Calling UPDATESCREEN
{special}
>> (8,1,5,-5,-5,5)
! Unsuitable expression.
<to be read again>

1.108 . . .$) at (0,0); special p;
                                  numspecial "p";
The expression shown above has the wrong type to be output.
```

```
{numspecial}
>> "p"
! Unsuitable expression.
<to be read again>

1.108 ...cial p; numspecial "p";

The expression shown above has the wrong type to be output.

{openwindow}
{(32)*(1024)}
! Arithmetic overflow.
1.109 . ...49999 from (0,32*1024)
                                  to (1,31*1057) at (0,0); shi...
Uh, oh. A little while ago one of the quantities that I was
computing got too large, so I'm afraid your answers will be
somewhat askew. You'll probably have to adopt different
tactics next time. But I shall try to carry on anyway.

{(31)*(1057)}

Calling BLANKRECTANGLE(100,100,0,1)
Calling  UPDATESCREER
{shipout}
[27.-14]
Edge structure at line 109 (just shipped out):
row 0: | -3++ -2-- -1++ 4094--
row -1: | -4096++ -3-- -2++ -1--

{openwindow}
Path at line 110, before choices:
(0,0)

Path at line 110, after choices:
(0,0)

{length(path)}

Calling BLANKRECTANGLE(0,2,0,4)
Calling  UPDATESCREER
{openwindow}
{length((-1,0))}
{-(5)}

Calling BLANKRECTANGLE(1,10,1,3)
Calling  UPDATESCREER
(display)
{nullpicture}
>> picture
! lot a suitable variable.
<to be read again>

1.112 display nullpicture;
                          display p inwindow 3; display p in...
At this point I needed to see the name of a picture variable.
(Or perhaps you have indeed presented me with one; I might
have missed it, if it wasn't followed by the proper token.)
So I'll not change anything just now.

{display)
>> 3
! Bad window number.
Cto be read again>
                  ;
1.112 . . . . display p inwindow 3;
                                  display p inwindow 6;
It should be the number of an open window.

{display)
! Variable p is the wrong type (transform).
<to be read again>
```

```
                    ,
1.112 . . . . display p inwindow 6;

I was looking for a "known" picture variable.
So I'll not change anything just now.

{display}

Calling BLANKRECTANGLE(1,10,1,3)
Calling PAINTROW(2,1;4,5,6,10)
Calling  UPDATESCREER
{cull}
! Enormous number has been reduced.
1.113 . . .opping (0.1,4095.999999
                                  ) withweight 3.5
I can't handle numbers bigger than about 4095.99998;
so I've changed your constant to that maximum amount.

! Weight must be -3, -2, -1, +1, +2, or +3.
<to be read again>
                   withweight
1.114 withweight
                -3.5; display e0 inwindow 5.5; addto e0 also . . .
I'll ignore the bad 'with' clause and look for another.

{-(3.5)}
! Bad culling amounts.
<to be read again>
                    ,
1.114 withweight-3.5;
                       display e0 inwindow 5.5; addto e0 also . . .
Always cull by known amounts that exclude 0.

{display}

Calling BLANKRECTANGLE(1,10,1,3)
Calling PAINTROW(2,1;4,5,6,8,9,10)
Calling PAINTROW(1,1;6,7,8,9)
Calling UPDATESCREEN
{addto}
>> (8,1,5,-5,-5,5)
! Improper 'addto'.
<to be read again>

1.114 . ..B 5.5; addto e0 also p;
                                    addto e0 contour 0;
This expression should have specified a known picture.
So I'll not change anything just now.

{addto}
>> 0
! Improper 'addto'.
<to be read again>

1.114 . . . p; addto e0 contour 0;

This expression should have been a known path.
So I'll not change anything just now.

{display)

Calling BLANKRECTANGLE(0,2,0,4)
Calling PAINTROW(2,1;0,1)
Calling PAINTROW(1,1;0,1)
Calling PAINTROW(0,1;1,2)
Calling  UPDATESCREEI
{addto}
! lot a cycle.
<to be read again>
                   ;
1.115 . . . . addto e0 contour p-9;
```

```
That contour should have ended with   .cycle' o r '&cycle ' .
So I'll not change anything just now

(display)
{(3)+(3)}

Calling BLANKRECTANGLE(1,10,1,3)
Calling PAINTROW(2,1;4,5,6,10)
Calling  UPDATESCREEI
(display)
{begingroup}
{vardef}
{endgroup}
! Variable e0e0 has been obliterated.
<to be read again>

1.116 . . .vardef e[]e=enddef;6]];

It seems you did a nasty thing---probably by accident,
but nevertheless you nearly hornaaoggled me...
While I was evaluating the right-hand aide of this
command, something happened, and the left-hand aide
is no longer a variable! So I won't change anything.

{addto}
{display}
Calling PAINTROW(1,1;0,1,2)
Calling PAINTROW(0,0;0,1,2)
Calling  UPDATESCREER
{display}
{display)
Calling  UPDATESCREER
{begingroup}
{let}
{vardef}
{endgroup}
! Variable c.al has been obliterated.
<to be read again>
                    ,
1.120 . . , ; vardef b=enddef ; 1]]]; 

It seems you did a nasty thing---probably by accident,
but nevertheless you nearly hornswoggled me...
Mile I was evaluating the suffix of this variable,
something was redefined, and it's no longer a variable!
In order to get back on my feet, I've inserted '0' instead.

{b1:=0}
! Variable bl has been obliterated.
Cto be read again>

1.120 . . .;vardef b=enddef;1]] I;

It seems you did a nasty thing---probably by accident,
but nevertheless you nearly hornswoggled me...
While I was evaluating the right-hand aide of this
command, something happened, and the left-hand side
is no longer a variable! So I won't change anything.

{errhelp}
>> 0
! Rot a string.
<to be read again>

1.121 errhelp 0;
                  err-message "Be like Jane";
A message should be a known string expression.

{errmessage}
! Be like Jane.
```

```
<to be read again>

1.121  ...message "Be like Jane";

This error message was generated by an 'errmessage'
command, so I can't give any explicit help.
Pretend that you're Hiss Marple, examine all clues,
and deduce the truth by inspired guesses.

{errhelp}
{errmessage}

<to be read again>

                    ,
1.122 . . .e%%%lp%"; errmessage"";

He%
lp


{errhelp}
{errmessage}
! Another.
<to be read again>
                  ;
1.123 . . . . errmessage "Another";

(That was another 'errmessage'.)

{headerbyte}
>> 0
! Improper location.
<to be read again>

1.124 headerbyte 0;
                    headerbyte(48.5)substring(-9,9)of"long";
I was looking for a known, positive number.
For safety's sake I'll ignore the present command.

{headerbyte}
! Missing ':' has been inserted.
<to be read again>
                   substring
1.124 . . .aderbyte(48.5)substring
                                  (-9,9)of"long";
A colon should follow a headerbyte or fontinfo location.

{-(9)}
{((-9,9))substring("long")}
>> "long"
! Invalid code has been replaced by 0.
<to be read again>

1.124 . . .ubstring(-9,9)of"long";

I was looking for a number between 0 and 255, or for a
string of length 1.  Didn't find it; will use 0 instead.

{headerbyte}
{(2)*(0.33333)}
{fontdimen}
{(2)*(0.33333)}
>> "q"
! Improper font parameter.
<to be read again>

1.125 ..."; fontdimen 9:2a6,"q";
                                  fontdimen 1:2048;
I'm zeroing this one.  Proceed, with fingers crossed.

{fontdimen}
```

```
{fontmaking:=1}
(extensible)
! Hissing ':' has been inserted.
<to be read again>
                   5
1.126 . . .king:=1;extensible 5 5
                                  ,"c"255.5,"d"; charlist 0:5:"...
I'm processing 'extensible c: t,m,b,r'.

! Hissing ',' has been inserted.
<to be read again>
                   255.5
1.126 . . .extensible 5 5,"c"255.5
                                  ,"d"; charlist 0:5:"a":"d";
I'm processing 'extensible c: t,m,b,r'.

>> 255.5
! Invalid code has been replaced by 0.
<to be read again>

1.126 . . .xtensible 5 5,"c"255.5,
                                  "d"; charlist 0:5:"a":"d";
I was looking for a number between 0 and 255, or for a
string of length 1. Didn't find it; will use 0 instead.

{charlist}
! Character code 5 is already extensible.
<to be read again>

1.126 . . .,"d"; charlist 0:5:"a":
                                  "d"
It's not legal to label a character more than once.
So I'll not change anything just now.

{ligtable}
{-(2048)}
! Character code 0 is already in a charlist.
<to be read again>

1.127 . . ."c":"d" kern -2048,0:
                                  "c":99.5:"e"=:"f",0kern';lig...
It's not legal to label a character more than once.
So I'll not change anything just now.

! Character c is already in a ligtable.
<to be read again>

1.127 . . .":"d" kern -2048,0:"c":
                                  99.5:"e"=:"f" 0kern' ; ligtabl...
It's not legal to label a character more than once.
So I'll not change anything just now.

>> (xpart ',ypart ')
! Improper kern.
<to be read again>

1.127 . . .":99.5:"e"=:"f",0kern';
                                   ligtable 5:0;
The amount of kern should be a known numeric value.
I'm zeroing this one. Proceed, with fingers crossed.

{ligtable}
! Character code 5 is already extensible.
<to be read again>

1.127 . . ."f",0kern'; ligtable 5:
                                  0;
It's not legal to label a character more than once.
So I'll not change anything just now.

! Illegal ligtable step.
```

```
<to be read again>

1.127 ...",0kern';ligtable 5:0;

I was looking for '=:' or 'kern' here.

{def}

clear(TEXT0)->interim(TEXT0):=$
(TEXT0)<-hppp
{interim)
{hppp:=-1}
{(0)=(0)}
! Redundant equation.
<to be read again>

1.128 . . .f; clear(hppp); vppp=0;

I already knew that this equation was true.
But perhaps no harm has been done; let's continue.

{begingroup}

clear(TEXT0)->interim(TEXT0):=$
(TEXT0)<-tracingmacros
{interim)
{tracingmacros:=-1}
(interim)
{tracingcommands:=-1}
Runaway text?
tracingoutput
! Forbidden token found while scanning a text argument.
<inserted text>

<to be read again>

1.129 . ..>. clear(tracingoutput\

It seems that a right delimiter was left out,
causing me to read past where you wanted me to atop
I'll try to recover; but if the error is serious,
you'd better type 'E' or 'X' now and fix your file.

{restoring proofing=1}
{restoringtracingoutput=1}
{restoringtracingcommands=2.1}
(restoringtracingmacros=1}
{endgroup}
{def}
{def}
{addto}
Path at line 139, before choices:
(0,0){4096,0}
 ..{4096,0}(1,0){0,-4096}
 ..{0,4096}cycle

Path at line 139, after choices:
(0,0)..controls (0.33333,0) and (0.66667,0)
 ..(1,0)..controls (1,-0.66667) and (0,-0.66667)
 ..cycle

> 0 EIE 1 (ESE SSE) SSV WSW WNW NNW 0 (NNE)
! Backwards path (turning number is negative).
<to be read again>
                   ;
1.139 ....{0,1}cycle withpen qq;

The path doesn't have a counterclockwise orientation,
so I'll probably have trouble drawing it.
(See Chapter 27 of The METAFONTbook for more help.)
```

```
Tracing edges at line 139: (weight 1)
@ Octant EIE (2 offsets), from (2.5,-0.5) to (1.5,-2)
@ retrograde line from (2.5,-0.5) to (0.5,-2)
(3,0)(3,-1)(1,-1)
@ transition line 0, from (0.5,-2) to (0.5,-2)
@ transition line 1, from (1.5,-2) to (2.5,-1.5)
@ transition line 2, from (2.5,-1.5) to (3.5,-0.5)
(1,-2)(3,-2)(3,-1)(4,-1)
@ retrograde line from (3.5,-0.5) to (1.5,-2)
(4,0)(4,-1)(2,-1)
@ Octant ESE (2 offsets), from (1.5,-2) to (-1,0)
@ retrograde line from (1.5,-2) to (-1,0)
(2,-2)(1,-2)(1,-1)(0,-1)
@ transition line 3, from (-1,0) to (0.5,-1.5)
@ transition line 2, from (0.5,-1.5) to (1.5,-2)
@ transition line 1, from (1.5,-2) to (1.5,-2)
(0,0)(0,-1)(1,-1)
@ retrograde line from (1.5,-2) to (-1,0)
(1,-2)(1,-1)(0,-1)
@ Octant SSE (0 offsets), from (-1,0) to (-1,0)
@ transition line 0, from (-1,0) to (-1,0)
@ Octant SSW (2 offsets), from (-1,0) to (-0.82407,0.67593)
@ retrograde line from (-1,0) to (-0.5,1.5)
(0,0)(-1,0)(-1,1)(0,1)
a transition line 3, from (-0.5,1.5) to (-0.5,1.5)
a transition line 2, from (-0.5,1.5) to (-1,0)
a transition line 1, from (-1.06265,-0.39499) to (-0.56265,1.10501)
@ transition line 2, from (-0.82407,0.67593) to (-1.32407,-0.82407)
a transition line 1, from (-1.32407,-0.82407) to (-1.32407,-0.82407)
(0,2)(0,1)(-1,1)
@ retrograde line from (-1.32407,-0.82407) to (-0.82407,0.67593)
(-1,-1)
@ Octant WSW (2 offsets), from (-0.82407,0.67593) to (1,2)
@ retrograde line from (-0.82407,0.67593) to (1.17593,2.17593)
(-1,1)(0,1)
@ transition line 0, from (1.17593,2.17593) to (1.17593,2.17593)
@ transition line 1, from (1.17593,2.17593) to (0.17593,1.67593)
a transition line 2, from (0,1.5) to (-1,0.5)
(0,2)
a retrograde line from (-1,0.5) to (1,2)
(0,1)
a Octant III (2 offsets), from (1,2) to (3.32407,0.17593)
@ retrograde line from (1,2) to (3.5,0)
(0,2)(2,2)(2,1)(3,1)
@ transition line 3, from (3.5,0) to (2,1.5)
@ transition line 2, from (1.82407,1.67593) to (0.82407,2.17593)
@ transition line 1, from (0.82407,2.17593) to (0.82407,2.17593)
(3,0)(3,1)(2,1)
@ retrograde line from (0.82407,2.17593) to (3.32407,0.17593)
(2,2)(2,1)(3,1)
a Octant NNW (0 offsets), from (3.32407,0.17593) to (3,1)
a transition line 0, from (3.32407,0.17593) to (3.32407,0.17593)
(3,0)
@ Octant NNE (2 offsets), from (3,1) to (2.5,-0.5)
@ retrograde line from (3,1) to (2.5,-0.5)
(3,1)
@ transition line 3, from (2.5,-0.5) to (2.5,-0.5)
@ transition line 2, from (2.5,-0.5) to (3,1)
a transition line 1, from (3,1) to (3,1)
(3,0)
@ retrograde line from (3,1) to (2.5,-0.5)
(3,1)(3,0).


f(SUFFIX0)(EXPR1)(EXPR2)(TEXT3)->numeric.w;show(EXPR1);addto(SUFFIX0)con
tour(0,0)..(2,0)..(1,$)..(1,1)..cycle.withpen.qq;addto(SUFFIX0)doublepat
h(0,0){1,1}..{2,1}(2,1)withpen.qq;addto(SUFFIX0)doublepath(($,$){1,0}..(
1,1){1,0})scaled0.5withpen.nullpen;cull(SUFFIX0)keeping(4,4)withweight1.
5;
! Missing `]' has been inserted.
<to be read again>
```

```
1.140 f(e[3,
          w); g(e3,transformed p,penoffset-(1,1.3)of(pencir...
I've seen a '[' and a subscript value, in a suffix,
so a right bracket should have come next.
I shall pretend that one was there.

(SUFFIX0)<-e3
(EXPR1)<-unknown path ʙ
! Hissing argument to f.
<to be read again>

1.140 f(e[3,w);
                g(e3,transformed p,penoffset-(1,1.3)of(pencir...
That macro has more parameters than you thought.
I'll continue by pretending that each missing argument
is either zero or null.

(EXPR2)<-0
! Hissing argument to f.
<to be read again>
                  ;
1.140 f(e[3,w);
                g(e3,transformed p,penoffset-(1,1.3)of(pencir...
That macro has more parameters than you thought.
I'll continue by pretending that each missing argument
is either zero or null.

(TEXT3)<-
{numeric}
{show}
>> unknown path %CAPSULE1579
{addto}
Path at line 140, before choices:
(0,0)
 ..(2,0)
 ..(1,-1)
 ..(1,1)
  .cycle

Path at line 140, after choices:
(0,0)..controls (0.56189,-0.5286) and (1.43811,0.5286)
 ..(2,0)..controls (2.70831,-0.66634) and (1.66634,-1.70831)
 ..(1,-1)..controls (0.4714,-0.43811) and (1.5286,0.43811)
 ..(1,1)..controls (0.33366,1.70831) and (-0.70831,0.66634)
 ..cycle

> 0 ESE ENE ESE 1 SSE SSV VSV VIV NNW 2 NNE NNW 3 VIV VSV SSV SSE ESE 0
! Strange path (turning number is zero).
<to be read again>
                  ;
f->.....(1,1)..cycle.withpen.qq;
                                addto(SUFFIX0)doublepath(0,0)...
<to be read again>
                  ;
1.140 f(e[3,w);
                g(e3,transformed p,penoffset-(1,1.3)of(pencir...
The path doesn't have a counterclockwise orientation,
so I'll probably have trouble drawing it.
(See Chapter 27 of The METAFONTbook for more help.)

Tracing edges at line 140: (weight 1)
@ Octaut EIE (2 offsets), from (0.26321,-2.6526) to (1.73679,-2.34741)
@ transition line 0, from (0.26321,-2.6526) to (0.26321,-2.6526)
@ transition line 1, from (1.73679,-2.34741) to (2.73679,-1.84741)
@ transition line 2, from (2.73679,-1.84741) to (3.73679,-0.84741)
(1,-3)(1,-2)(3,-2)
@ retrograde line from (3.73679,-0.84741) to (1.73679,-2.34741)
(3,-1)
@ Octant ESE (2 offsets), from (1.73679,-2.34741) to (-0.25465,-0.56029)
a retrograde line from (1.73679,-2.34741) to (-0.76321,-0.34741)
```

```
(3,-2)(1,-2)(1,-1)(-1,-1)
@ transition line 3, from (-0.76321,-0.34741) to (0.73679,-1.84741)
@ transition line 2, from (0.73679,-1.84741) to (1.73679,-2.34741)
@ transition line 1, from (2.05333,-2.41772) to (1.05333,-1.91772)
@ transition line 2, from (1.24535,-2.06029) to (2.24535,-2.56029)
@ transition line 1, from (2.24535,-2.56029) to (2.24535,-2.56029)
(-1,0)(-1,-1)(0,-1)(0,-2)(2,-2)
@ retrograde line from (2.24535, -2.56029) to (-0.25465,-0.56029)
(2,-3)(2,-2)(1,-2)
@ Octant SSE (0 offsets), from (-0.25465,-0.56029) to (0,-1.13196)
@ transition line 0, from (-0.25465,-0.56029) to (-0.25465,-0.56029)
@ Octant SSV(2 offsets), from (0,-1.13196) to (0.25,-0.25)
@ retrograde line from (0,-1.13196) to (0.5,0.36804)
(1,-1)(0,-1)
a transition line 3, from (0.5,0.36804) to (0.5,0.36804)
@ transition line 2, from (0.5,0.36804) to (0,-1.13196)
@ transition line 1, from (-0.04594,-1.41693) to (0.45406,0.08307)
@ transition line 2, from (0.25,-0.25) to (-0.25,-1.75)
@ transition line 1, from (-0.25,-1.75) to (-0.25,-1.75)
(0,0)
@ retrograde line from (-0.25,-1.75) to (0.25,-0.25)
(0,-2)
@ Octant VSV(2 offsets), from (0.25,-0.25) to (1.63197,1)
@ retrograde line from (0.25,-0.25) to (2.25,1.25)
(0,0)(1,0)
@ transition line 0, from (2.25,1.25) to (2.25,1.25)
@ transition line 1, from (2.25,1.25) to (1.25,0.75)
@ transition line 2, from (1.03186,0.59352) to (2.03186,1.09352)
@ transition line 1, from (1.63197,1) to (0.63197,0.5)
@ transition line 2, from (0.63197,0.5) to (-0.36803,-0.5)
(1,1)
@ retrograde line from (-0.36803,-0.5) to (1.63197,1)
(1,0)
@ Octant VIV (2 offsets), from (1.63197,1) to (3.56024,-0.74529)
@ retrograde line from (1.63197,1) to (4.13197,-1)
(1,1)(2,1)(2,0)(4,0)
@ transition line 3, from (4.13197,-1) to (2.63197,0.5)
@ transition line 2, from (2.63197,0.5) to (1.63197,1)
@ transition line 1, from (1.29863,1.07983) to (2.29863,0.57983)
@ transition line 2, from (2.06024,0.75471) to (1.06024,1.25471)
@ transition line 1, from (1.06024,1.25471) to (1.06024,1.25471)
(4,-1)(4,0)(3,0)
@ retrograde line from (1.06024,1.25471) to (3.56024,-0.74529)
(3,1)(2,1)(2,0)(3,0)
@ Octant IIV(0 offsets), from (3.56024,-0.74529) to (3.3474,-0.23679)
@ transition line 0, from (3.56024,-0.74529) to(3.56024,-0.74529)
(3,-1)
@ Octant IIE (2 offsets), from (3.3474,-0.23679) to (3.65259,1.23679)
@ retrograde line from (3.3474,-0.23679) to (2.8474,-1.73679)
(3,0)
@ transition line 3, from (2.8474,-1.73679) to (2.8474,-1.73679)
@ transition line 2, from (2.8474,-1.73679) to (3.3474,-0.23679)
@ transition line 1, from (3.65259,1.23679) to (3.65259,1.23679)
(3,-2)
@ Octant [illegible](0 offsets), from (3.65259,1.23679) to (3.43976,1.74529)
@ transition line 0, from (3.65259,1.23679) to (3.65259,1.23679)
(3,1)(4,1)
@ Octant UN (2 offsets), from (3.43976,1.74529) to (0.36803,4)
@ transition line 3, from (3.43976,1.74529) to (1.93976,3.24529)
@ transition line 2, from (1.70137,3.42017) to (0.70137,3.92017)
@ transition line 1, from (0.36803,4) to (0.36803,4)
(4,2)(3,2)(3,3)(2,3)
@ Octant VSV(2 offsets), from (0.36803,4) to (-2.25,2.25)
@ transition line 0, from (0.36803,4) to (0.36803,4)
a transition line 1, from (-0.03186,3.90648) to (-1.03186,3.40648)
@ transition line 2, from c-1.25.3.25) to (-2.25,2.25)
(2,4)(-1,4)(-1,3)(-2,3)
@ Octant SSW(2 offsets), from (-2.25,2.25) to (-3,0.13196)
@ transition line 3, from (-2.25,2.25) to (-2.25,2.25)
@ transition line 2, from (-2.45406,1.91695) to (-2.95406,0.41695)
a transition line 1, from (-3,0.13196) to (-3,0.13196)
```

```
(-2,2)(-3,2)
@ Octant SSE(0 offsets), from (-3,0.13196) to (-2.74533,-0.43973)
@ transition line 0, from(-3,0.13196) to (-3,0.13196)
@ Octant ESE (2 offsets), from (-2.74533,-0.43973) to (0.26321,-2.6526)
a transition line 3, from (-2.74533,-0.43973) to (-1.24533,-1.93973)
@ transition line 2, from (-1.05331,-2.08229) to (-0.05331,-2.58229)
@ transition line 1, from (0.26321,-2.6526) to (0.26321,-2.6526)
(-3,-1)(-2,-1)(-2,-2)(0,-2)(0,-3).

{addto}
Path at line 140, before choices:
(0,0){2896.30943,2896.30934}
 ..{3663.57385,1831.78674}(2,1)

Path at line 140, after choices:
(0,0)..controls (0.53566,0.53566) and (1.32689,0.66344)
 ..(2,1)

Tracing edges at line 140: (weight 1)
@ Octant EIE (2 offsets), from (0,-2.5) to (4,0)
@ transition line 0, from (0,-2.5) to (0,-2.5)
@ transition line 1. from (0,-2.5) to (1,-2)
@ transition line 2, from (1.607,-1.57747) to (0.607,-2.07747)
@ transition line 1, from (2,-1.5) to (3,-1)
a transition line 2, from (3,-1) to (4,0)
(2,-2)(2,-1)(4,-1)
@ Octaut NNE (2 offsets), from (4,0) to (4.5,1.5)
a transition line 3, from (4,0) to (4,0)
a transition line 2, from (4,0) to (4.5,1.5)
a transition line 1, from (4.5,1.5) to (4.5,1.5)
(4,1)(5,1)
a Octant NNW(0 offsets), from (4.5,1.5) to (4.5,1.5)
a transition line 0, from (4.5,1.5) to (4.5,1.5)
@ Octant VIV (2 offsets), from (4.5,1.5) to (2,3.5)
a transition line 3, from (4.5,1.5) to (3,3)
@ transition line 2, from (3,3) to (2,3.5)
@ transition line 1, from (2,3.5) to (2,3.5)
(5,2)(4,2)(4,3)(2,3)
@ Octant VSV(2 offsets), from (2,3.5) to (-2,1)
@ transition line 0, from (2,3.5) to (2,3.5)
@ transition line 1, from (0.607,2.92253) to (-0.393,2.42253)
@ transition line 2, from (-1,2) to (-2,1)
(2,4)(2,3)(0,3)(0,2)(-1,2)
@ Octant SSV(2 offsets), from (-2,1) to (-2.5,-0.5)
@ transition line 3, from (-2,1) to (-2,1)
@ transition line 2, from (-2,1) to (-2.5,-0.5)
@ transition line 1, from (-2.5,-0.5) to (-2.5,-0.5)
(-1,1)(-2,1)
@ Octaut SSE (0 offsets), from (-2.5,-0.5) to (-2.5,-0.5)
@ transition line 0, fron (-2.5,-0.5) to (-2.5,-0.5)
@ Octant ESE (2 offsets), from (-2.5,-0.5) to (0,-2.5)
@ transition line 3, from (-2.5,-0.5) to (-1,-2)
@ transition line 2, from (-1,-2) to (0,-2.5)
@ transition line 1, from (0,-2.5) to (0,-2.5)
(-2,-1)(-1,-1)(-1,-2).

{addto}
Path at line 140, before choices:
(-1,-1){4096,0}
 ..{4096,0}(1,1)

Path at line 140, after choices:
(-1,-1)..controls (0.10457,-1) and (-0.10457,1)
 ..(1,1)

{(path)scaled(0.5)}
{nullpen}
Tracing edges at line 140: (weight 1)
(0,0)(0,1)(0,0).

{cull}
```

```
g(SUFFIX0)->show(SUFFIX0)
(SUFFIX0)<-e3
! Too many arguments to g;
  Missing ')' has been inserted.
1.140 f(e[3,w); g(e3,
                    transformedp,penoffset-(1,1.3)of(pencir...
I'm going to assume that the comma I just read was a
right delimiter, and then I'll begin expanding the macro.
You might want to delete some tokens before continuing.

{show}
{(picture)transformed((8,1,5,-5,-5,5))}
>> Edge structure at line 140:

{-((1,1.3))}
{pencircle}
{(future pen)scaled(20)}
{-(0.5)}
{(future pen)yscaled(-0.5)}
Pen polygon at line 140 (newly created):
(1.5,-5)
 .. (4.5,-4.5)
 .. (6,-4)
 .. (8,-3)
 .. (9.5,-1.5)
 .. (10,-0.5)
 .. (10,0.5)
 .. (9.5,1.5)
 .. (8,3)
 .. (6,4)
 .. (4.5,4.5)
 .. (1.5,5)
 .. (-1.5,5)
 .. (-4.5,4.5)
 .. (-6,4)
 .. (-8,3)
 .. (-9.5,1.5)
 .. (-10,0.5)
 .. (-10,-0.5)
 .. (-9.5,-1.5)
 .. (-8,-3)
 .. (-6,-4)
 .. (-4.5,-4.5)
 .. (-1.5,-5)
 .. cycle

{((-1,-1.3))penoffset(pen)}
>> (-9.5,1.5)
Path at line 141, before choices:
(0,0)..controls (1,1) and (-1,1)
 ..(3,0)

Path at line 141, after choices:
(0,0)..controls (1,1) and (-1,1)
 ..(3,0)

{((0,1))directiontime(path)}
>> 0.33333
{(3.14159)point(path)}
{((3.14159,4))subpath(path)}
{((7.68668,2.72437))intersectiontimes(path)}
>> (0,0)
Path at line 143, before choices:
(-1,1.1){curl 13
 ..{curl 1}(1,-1)

Path at line 143, after choices:
(-1,1.1)..controls (-0.33333,0.40001) and (0.33333,-0.3)
 ..(1,-1)
```

```
{(-1)precontrol((0,0))}
{(path)intersectiontimes((0,0))}
>> (-1,-1)
{addto}
{-(4094.99998)}
{-(0.00002)}
Path at line 144, before choices:
(-4094.99998,0){curl 1}
 ..{curl 1}(4094.99998,-0.00002)

Path at line 144, after choices:
(-4094.99998,0)..controls (-1365,0) and (1365,-0.00002)
 ..(4094.99998,-0.00002)

{pencircle}
{(0)penoffset(future pen))
Pen polygon at line 145 (newly created):
(0.5,0)
 .. (0,0.5)
 .. (-0.5,0)
 .. (0,-0.5)
 .. cycle

>> 0
>> pen
! lot implemented: penoffset(known numeric)of(pen).
<to be read again>
                   ;
1.145 pencircle;
                addto e3 also e3 shifted (0,257);,"flushed . . .
I'm afraid I don't know how to apply that operation to that
combination of types. Continue, and I'll return the second
argument (see above) as the result of the operation.

Tracing edges at line 145: (weight 1)
@ Octant ESE(1 offset), from (-4095.49998,0) to (4094.99998,-0.50002)
@ transition line 2, from (-4095.49998,0) to (-4094.99998,-0.5)
@ transition line 1, from (4094.99998,-0.50002) to (4094.99998,-0.50002)
(-4095,0)
@ Octant ENE (1 offset), from (4094.99998,-0.50002) to (4095.49998,-0.00
002)
@ transition line 0, from (4094.99998,-0.50002) to (4094.99998,-0.50002)
@ transition line 1, from (4094.99998,-0.50002) to (4095.49998,-0.00002)
(-4095,-1)(4095,-1)
@ Octant NNE (0 offsets), from (4095.49998,-0.00002) to (4095.49998,-0.0
0002)
@ transition line 1, from (4095.49998,-0.00002) to (4095.49998,-0.00002)
@ Octant NNW (0 offsets), from (4095.49998,-0.00002) to (4095.49998,-0.0
0002)
@ transition line 0, from (4095.49998,-0.00002) to (4095.49998,-0.00002)
@ Octant WNW (1 offset), from (4095.49998,-0.00002) to (-4094.99998,0.5)
@ transition line 2, from (4095.49998,-0.00002) to (4094.99998,0.49998)
@ transition line 1, from (-4094.99998,0.5) to (-4094.99998,0.5)
(4095,0)(-4095,0)
@ Octant WSW (1 offset), from (-4094.99998,0.5) to (-4095.49998,0)
@ transition line 0, from (-4094.99998,0.5) to (-4094.99998,0.5)
@ transition line 1, from (-4094.99998,0.5) to (-4095.49998,0)
(-4095,1)
@ Octant SSW (0 offsets), from (-4095.49998,0) to (-4095.49998,0)
@ transition line 1, from (-4095.49998,0) to (-4095.49998,0)
@ Octant SSE (0 offsets), from (-4095.49998,0) to (-4095.49998,0)
@ transition line 0, from (-4095.49998,0) to (-4095.49998,0)
(-4095,0).

{addto}
{(picture)shifted((0,257))}
! A statement can't begin with ','.
<to be read again>
                   ,
1.145 ...o e3 shifted (0,257); ,
                                "flashed with pride"; numeric...
```

```
I was looking for the beginning of a new statement.
If you just proceed without changing anything, I'll ignore
everything up to the next ';'. Please insert a semicolon
now in front of anything that you don't want me to delete.
(See Chapter 27 of The METAFONTbook for an example.)

! Extra tokens will be flushed.
<to be read again>

1.145 . . .o e3 shifted (0,257); ,
                                  "flushed with pride"; numeric...
I've just read as much of that statement as I could fathom,
so a semicolon should have been next. It's very puzzling...
but I'll try to get myself back together, by ignoring
everything up to the next ';'. Please insert a semicolon
now in front of anything that you don't want me to delete.
(See Chapter 27 of The METAFONTbook for an example.)

{numeric)
{def}

f(EXPR0)(EXPR1)(EXPR2)->showdependencies;tracingcapsules:=1;showdependen
cies;show1/3(3,6)*(((EXPR0)+(EXPR1))+((EXPR1)-(EXPR0))),(1,1)/sqrt2zscal
ed((EXPR0)+1,(EXPR0)+2)-((EXPR0)+1,(EXPR0)+2)rotated45,(0,1)zscaled(1,(E
XPR1)+2)-(1,(EXPR1)+2)rotated90
{(xx)+(1)}
{(xx+1)/(0.3)}
(EXPR0)<-3.3333xx+3.3333
{(yy)-(1)}
{(yy-1)/(0.5)}
(EXPR1)<-2yy-2
(EXPR2)<-(xx,0)
{showdependencies}
alpha=0.45p3~+7
{tracingcapsules:=1}
{showdependencies}
xpart %CAPSULE1515=xx
%CAPSULE1523=2yy-2
%CAPSULE1531 = 3.3333xx+3.3333
alpha=0.45p3~+7
{show}
{(1/3)*((3,6))}
{(3.3333xx+3.3333)+(2yy-2)}
{(2yy-2)-(3.3333xx+3.3333)}
{(linearform)+(linearform)}
{((1,2))*(4yy-4)}
>> (4yy-4,8yy-8)
{sqrt(2)}
{((1,1))/(1.41422)}
{(3.3333xx+3.3333)+(1)}
{(3.3333xx+3.3333)+(2)}
{((0.7071,0.7071))zscaled((3.3333xx+4.3333,3.3333xx+5.3333))}
{(3.3333xx+3.3333)+(1)}
{(3.3333xx+3.3333)+(2)}
{((3.3333xx+4.3333,3.3333xx+5.3333))rotated(45)}
{((-0.7071,4.71399xx+6.83531))-((-0.7071,4.71399xx+6.83531))}
>> (0,0)
{(2yy-2)+(2)}
{((0,1))zscaled((1,2yy))}
{(2yy-2)+(2)}
{((1,2yy))rotated(90)}
{((-2yy,1))-((-2yy,1))}
>> (0,0)
{(1000)*(o3)}
{(o1)-(o2)}
{(4000)*(linearform)}
{(1000o3)-(linearform)}
{(4000)*(o2)}
{(linearform)+(4000o2)}
{(linearform)+(9)}
{(0.01)*(o3)}
```

```
{-(0.01o3)}
{(3)*(ooo)}
{(-0.01o3)+(3ooo)}
{begingroup}
{(o2)+(o4)}
{(linearform)+(o5)}
{(linearform)-(20)}
{(9/10)*(linearform)}
{(oo)=(linearform)}
## oo=0.9o5+0.9o4+0.9o2-18
{(0.5)*(o2)}
{(o1)+(0.5o2)}
{(8/9)*(linearform)}
{(o5)=(linearform)}
## o5=0.44444o2+0.88889o1
{(o4)=(linearform)}
## o4=0.44444o2+0.88889o1
{(0.0001)*(o2)}
{-(0.0001o2)}
{(o6)=(-0.0001o2)}
## o6=-0.0001o2
{showdependencies}
o6=-0.0001o2
o4=0.44444o2+0.88889o1
o5=0.44444o2+0.88889o1
oo=1.7o2+1.6o1-18
%CAPSULE1437 = 3000-0.0103
xpart %CAPSULE1461 = 8000o2-4000o1+1000o3+9
xpart %CAPSULE1539=xpart '
ypart %CAPSULE1539=ypart '
alpha=0.45p3^+7
{numeric}
### -1.6o1=-oo+1.7o2-18
! Arithmetic overflow.
1.151   numeric o[];
                    xpart(alfa,[[pair alfa;0]])]])=-2/3[[save...
Uh, oh. A little while ago one of the quantities that I was
computing got too large, so I'm afraid your answers will be
somewhat askew. You'll probably have to adopt different
tactics next time. But I shall try to carry on anyway.

### -3062.5o2*4 = -xpart %CAPSULE1461-2500.00002oo+1000o3-32758.99998
#### o6=0
### -0.0457803 = -o4+0.44112oo+8.50041
### -04 = -0s
### 0.2183405 = -%CAPSULE1437+0.09631oo+3ooo+1.85596
{begingroup}
{pair}
### -alfa=-xpart %CAPSULE1103
{endgroup}
{xpart((xpart %CAPSULE1103,0))}
### -xpart %CAPSULE17=-%CAPSULE680
{endgroup}
{(%CAPSULE1437)+(%CAPSULE680)}
### -%CAPSULE1437=-%CAPSULE434+%CAPSULE680
### -%CAPSULE434=-ypart %CAPSULE1461
{begingroup}
{save}
{(1)-(p$)}
{restoring p}
### p$=-ypart %CAPSULE1092+1
{endgroup}
{(2/3)*((-ypart %CAPSULE1092+1,ypart %CAPSULE1092))}
### ypart %CAPSULE1092=-xpart %CAPSULE1092+1
### -0.66667xpart %CAPSULE1092=-xpart %CAPSULE1669
{-((xpart %CAPSULE1669,-xpart %CAPSULE1669+0.66667))}
### xpart %CAPSULE1669=-xpart %CAPSULE1736
{((xpart %CAPSULE1461,ypart %CAPSULE1461))=((xpart %CAPSULE1736,-xpart %
CAPSULE1736-0.66667))}
## xpart %CAPSULE1736=-ypart %CAPSULE1461-0.66667
## ypart %CAPSULE1461=-xpart %CAPSULE1461-0.66667
```

```
### -xpart %CAPSULE1461=-xpart %CAPSULE1736
{((xpart ',ypart '))=((xpart %CAPSULE1736,-xpart %CAPSULE1736-0.66667))}
## xpart %CAPSULE1736=-ypart j-0.66667
## xpart '=-ypart '-0.66667
{begingroup}
{(0.5)*(ooo)}
{begingroup}
{numeric}
### -ooo=-%CAPSULE1496
{endgroup}
{(%CAPSULE1496)+(1)}
### -%CAPSULE1496=-%CAPSULE1484+1
{(1/2)*(%CAPSULE1484)}
### -0.5%CAPSULE1484=-%CAPSULE1476
{-(%CAPSULE1476)}
### %CAPSULE1476=-%CAPSULE1468
{(2)*(%CAPSULE1468)}
### -2%CAPSULE1468=-%CAPSULE1435
{(-0.5%CAPSULE1435-0.5)=(%CAPSULE1435)}
## %CAPSULE1435=-0.33333
{(oo)=(-0.33333)}
## oo=-0.33333
{endgroup}
{(4000)*(-0.33333)}
{yoffset:=-1333.31299}
{xoffset:=-1333.31299)
{for}
{sqrt(-1)}
! Square root of -1 has been replaced by 0.
1.153 for @=angle(sqrt$,
                        mlog$):charext:=uniformdeviate$;charh...
Since I don't take square roots of negative numbers,
I'm zeroing this one. Proceed, with fingers crossed.

{mlog(-1)}
! Logarithm of -1 has been replaced by 0.
1.153 for @=angle(sqrt$,mlog$)
                                :charext:=uniformdeviate$;charh...
Since I don't take logs of non-positive numbers,
I'm zeroing this one. Proceed, with fingers crossed.

{angle((0,0))}
! angle(0,0) is taken as zero.
1.153 for @=angle(sqrt$,mlog$):
                                charext:=uniformdeviate$;charh...
The 'angle' between too identical points is undefined.
I'm zeroing this one. Proceed, vith fingers crossed.

Runaway loop?
charext:=uniformdeviate$;charht:=2048;granularity:=-8;ETC.
! Forbidden token found while scanning the text of a for loop.
<inserted text>
                endfor
<to be read again>
                    endtext
1.163 . . . shoastats; bye endtext

I suspect you have forgotten an 'endfor',
causing me to read past where you wanted me to stop.
I'll try to recover; but if the error is serious,
you'd better type 'E' or 'X' now and fix your file.

{loop value=0}
{uniformdeviate(-1)}
{charext:=-0.00507}
{charht:=2048}
{-(8)}
{granularity:=-8}
{addto}
{-(100)}
{-(99)}
```

```
Path at line 183, before choices:
(0,-100)..tension 500
 ..(100,-99)..tension 3000
 ..cycle

Path at line 163, after choices:
(0,-100)..controls (-36.92659,-103.51175) and (136.98943,-101.7726)
 ..(100,-99)..controls (99.9889,-98.99916) and (0.01108,-99.99895)
 ..cycle

Tracing edges at line 163: (weight 1)
(-3,-101)(-3,-102)(71,-102)(71,-101)(99,-101)(99,-100)(103,-100)
(103,-99)(48,-99)(48,-100)(-7,-100)(-7,-101).

{tracingoutput:=0}
{shipout}
! Enormous charht has been reduced.
<to be read again>
                   ;
<for(0)> . . .=(EXPR0);shipout.e3;
                                   special"bye";interim.char99="...
<inserted text> endfor

<to be read again>
                   endtext
1.163 . . . shoostats; bye endtext

Font metric dimensions must be less than 2048pt.

[109]
{special}
{interim}
! The token 'char' isn't an internal. quantity.
<to be read again>
                   char
<for(0)> . . .al"bye";interim.char
                                   99="c";true=false;[[clear(tra...
<inserted text> endfor

<to be read again>
                   endtext
1.163 . . . showstats; bye endtext

Something like `tracingonline' should follow 'interim'.

{char(99)}
{("c")=("c")}
! Redundant equation.
<to be read again>
                   ,
<for(0)>. . . . interim.char99="c";
                                   true=false;[[clear(tracingcom...
<inserted text> endfor

<to be read again>
                   endtext
1.163 . . . shoostats; bye endtext

I already knew that this equation was true.
But perhaps no harm has been done; let's continue.

{true}
(false3
{(true)=(false)}
! Inconsistent equation
<to be read again>
                   ;
<for(0)> . . .ar99="c";true=false;
                                   [[clear(tracingcommands);char...
<inserted text> endfor
```

```
<to be read again>
                    endtext
1.163 . . . shoostats; bye endtext

The equation I just read contradicts what was said before.
But don't worry; continue and I'll just ignore it.

{begingroup}

clear(TEXT0)->interim(TEXT0):=$
(TEXT0)<-tracingcommands
{interim}
{tracingcommands:=-1}
! Enormous charht has been reduced.
<to be read again>

<for(0)>...shipout+nullpicture;
                                  "careful"for.for=(EXPR0)step2...
<inserted text> endfor

<to be read again>
                    endtext
1.163 . . . showstats; bye endtext

Font metric dimensions must be less than 2048pt.

[0]
careful HETAFOIT HETAFORT HETAFOIT HETAFORT HETAFOIT HETAFOIT HETAFOIT M
ETAFOIT METAFONT HETAFORT HETAFOIT HETAFOIT HETAFOIT HETAFOIT HETAFOIT M
ETAFOIT HETAFOIT METAFONT HETAFOIT HETAFOIT HETAFOIT HETAFOIT HETAFOIT M
ETAFOIT HETAFOIT METAFONT HETAFOIT
{restoring tracingcommands=2.1}
{endgroup}
{scrollmode}

{char(31)}
{("hello again")&("^^_")}
hello again^^_
{save}
### -0.45p3'=-alpha+7
{-(0.04)}
{fillin:=-0.043
{def}
{begingroup}
{tracingspecs:=1}
{show}
{nullpen}

>> Pen polygon at line 163:
(0,0)
 .. cycle

{makepath(pen)}

>> Path at line 163:
(1,-2)..controls (1,-2) and (2,-1)
 ..(2,-1)..controls (2,-1) and (2.5,0.5)
 ..(2.5,0.5)..controls (2.5,0.5) and (1,2)
 ..(1,2)..controls (1,2) and (0,2.5)
 ..(0,2.5)..controls (0,2.5) and (-1,2)
 ..(-1,2)..controls (-1,2) and (-2,1)
 ..(-2,1)..controls (-2,1) and (-2.5,-0.5)
 ..(-2.5,-0.5)..controls (-2.5,-0.5) and (-1,-2)
 ..(-1,-2)..controls (-1,-2) and (0,-2.5)
 ..(0,-2.5)..controls (0,-2.5) and (1,-2)
 ..cycle

{(pen)rotated(1)}
Pen polygon at line 163 (newly created):
(0.97366,1.51723)
 .. cycle
```

```
{makepath(pen)}

>> Path at line 163:
(0.97366,1.51723)..controls (0.97366,1.51723) and (0.97366,1.51723)
 ..cycle

{pencircle}
{makepath(future pen)}
Pen polygon at line 163 (newly created):
(0.5,-0.5)
 .. (0.5,0.5)
 .. (-0.5,0.5)
 .. (-0.5,-0.5)
 .. cycle

{(path)rotated(-1)}

>> Path at line 163:
(0.4912,-0.50865)..controls (0.4912,-0.50865) and (0.50865,0.4912)
 ..(0.50865,0.4912)..controls (0.50865,0.4912) and (-0.4912,0.50865)
 ..(-0.4912,0.50865)..controls (-0.4912,0.50865) and (-0.50865,-0.4912)
 ..(-0.50865,-0.4912)..controls (-0.50865,-0.4912) and (0.4912,-0.50865)
 ..cycle

{addto}
Path at line 163, before choices:
(0,2){0,-4096}
 ..{0,-4096}(0,1){0,-4096}
 ..{4096,0}(3,0){4096,0}
 ..{4096,0}(4,0){4096,0}
 ..{0,-4096}cycle

Path at line 163, after choices:
(0,2)..controls (0,1.66667) and (0,1.33333)
 ..(0,1)..controls (0,-0.20886) and (1.63324,0)
 ..(3,0)..controls (3.33333,0) and (3.66667,0)
 ..(4,0)..controls (13.63031,0) and (0,10.64238)
 ..cycle

Path at line 163, before choices:
(0,0)
 ..(5,2.9)
 ..(4,3)
 ..cycle

Path at line 163, after choices:
(0,0)..controls (0.9363,-8.0181) and (11.81119,0.07109)
 ..(5,2.9)..controls (4.68596,3.03043) and (4.33879,3.02028)
 ..(4,3)..controls (1.86353,2.87216) and (-0.21431,1.83528)
 ..cycle

{makepen(path)}
Pen polygon at line 163 (newly created):
(5,2.9)
 .. (4,3)
 .. (0,0)
 .. cycle

Path at line 163, before subdivision into octants:
(0,2)..controls (0,1.66667) and (0,1.33333)
 ..(0,1)..controls (0,-0.20886) and (1.63324,0)
 ..(3,0)..controls (3.33333,0) and (3.66667,0)
 ..(4,0)..controls (13.63031,0) and (0,10.64238)
 ..cycle

Cycle spec at line 163, after subdivision and double autorounding:
(1.5,0.00272) % beginning in octant 'SSE'
   ..controls (1.5,-0.49254) and (1.63411,-0.82727)
 ..(1.8592,-1.05237) % segment 1
% entering octant 'ESE'
```

```
   ..controls (2.23936,-1.43253) and (2.87909,-1.5)
  ..(3.57066,-1.5) % segment 1
% entering octant 'ENE'
   ..controls (3.95114,-1.5) and (4.3473,-1.47957)
  ..(4.72458,-1.47957) % segment 1
   ..controls (5.08287,-1.47957) and (5.44116,-1.47957)
  ..(5.79944,-1.47957) % segment 2
   ..controls (7.35368,-1.47957) and (8.34424,-1.12393)
  ..(8.90538,-0.56279) % segment 3
% entering octant 'NNE'
   ..controls (9.31982,-0.14835) and (9.5,0.37819)
  ..(9.5,0.95631) % segment 3
% entering octant 'NNW'
   ..controls (9.5,2.00116) and (8.61821,3.4318)
  ..(7.41951,4.6305) % segment 3
% entering octant 'WNW'
   ..controls (6.32909,5.72092) and (5.11603,6.5)
  ..(4.2104,6.5) % segment 3
% entering octant 'WSW'
   ..controls (3.64668,6.5) and (3.12122,6.31134)
  ..(2.68588,5.876) % segment 3
% entering octant 'SSW'
   ..controls (1.97163,5.16176) and (1.5,3.78351)
  ..(1.5,1.485) % segment 3
   ..controls (1.5,0.99092) and (1.5,0.49681)
  ..(1.5,0.00272) % segment 0
 & cycle

Tracing edges at line 163: (weight 1)
@ Octant SSE (0 offsets), from (1.5,0.00272) to (1.8592,-1.05237)
@ transition line 0, from (1.5,0.00272) to (1.5,0.00272)
(2,0)
@ Octant ESE (0 offsets), from (1.8592,-1.05237) to (3.57066,-1.5)
@ transition line 1, from (3.57066,-1.5) to (3.57066,-1.5)
@ Octant EIE (2 offsets), from (3.57066,-1.5) to (13.90538,2.3372)
@ transition line 0, from (3.57066,-1.5) to (3.57066,-1.5)
@ transition line 1, from (8.41534,-0.93503) to (13.41534,1.96497)
@ transition line 2, from (13.90538,2.3372) to (13.90538,2.3372)
(2,-1)(9,-1)(9,0)(11,0)(11,1)(13,1)
@ Octant NNE (0 offsets), from (13.90538,2.3372) to (14.5,3.85631)
@ transition line 1, from (14.5,3.85631) to (14.5,3.85631)
(13,2)(14,2)
@ Octant NNW(0 offsets), from (14.5,3.85631) to (12.41951,7.53049)
@ transition line 0, from (14.5,3.85631) to (14.5,3.85631)
(14,6)(13,6)(13,7)(12,7)     .
@ Octant WNW (2 offsets), from (12.41951,7.53049) to (8.2104,9.5)
@ transition line 3, from (12.41951,7.53049) to (12.41951,7.53049)
@'transition line 2, from (9.38113,9.39124) to (8.38113,9.49124)
@ transition line 1, from (8.2104,9.5) to (8.2104,9.5)
(12,8)(11,8)(11,9)(8,9)
@ Octant WSW (2 offsets), from (8.2104,9.5) to (2.68588,5.876)
@ transition line 0, from (8.2104,9.5) to (8.2104,9.5)
@ transition line I, from (6.92668,9.08539) to (2.92668,6.08539)
@ transition line 2, from (2.68588,5.876) to (2.68588,5.876)
(8,10)(8,9)(6,9)(6,8)(5,8)(5,7)(3,7)
@ Octaut SSW (0 offsets), from (2.68588,5.876) to (1.5,0.00272)
@ transition line I, from (1.5,0.00272) to (1.5,0.00272)
(3,6)(2,6)(2,0).

Path at line 163, before subdivision into octants:
(0,2)..controls (0,10.64238) and (13.63031,0)
  ..(4,0)..controls (3.66667,0) and (3.33333,0)
  ..(3,0)..controls (1.63324,0) and (0,-0.20886)
  ..(0,1)..controls (0,1.33333) and (0,1.66667)
  ..cycle

Cycle spec at line 163, after subdivision and double autorounding:
(2.68588,5.876) % beginning in octant 'ENE'
   ..controls (3.12122,6.31134) and (3.64668,6.5)
  ..(4.2104,6.5) % segment 0
% entering octant 'ESE'
```

```
   ..controls (5.11603,6.5) and (6.32909,5.72092)
 ..(7.41953,4.63048) % segment 0
% entering octant 'SSE'
   ..controls (8.61824,3.43176) and (9.5,2.00116)
 ..(9.5,0.95631) % segment 0
% entering octant 'SSW'
   ..controls (9.5,0.37819) and (9.31982,-0.14835)
 ..(8.90538,-0.56279) % segment 0
% entering octant 'WSW'
   ..controls (8.34425,-1.12392) and (7.35368,-1.47957)
 ..(5.79944,-1.47957) % segment 0
   ..controls (5.44116,-1.47957) and (5.08287,-1.47957)
 ..(4.72458,-1.47957) % segment 1
   ..controls (4.3473,-1.47957) and (3.95114,-1.5)
 ..(3.57066,-1.5) % segment 2
% entering octant 'WNW'
   ..controls (2.87909,-1.5) and (2.23936,-1.43253)
 ..(1.85919,-1.05235) % segment 2
% entering octant 'NNW'
   ..controls (1.6341,-0.82726) and (1.5,-0.49254)
 ..(1.5,0.00272) % segment 2
% entering octant 'NNE'
   ..controls (1.5,0.49681) and (1.5,0.99092)
 ..(1.5,1.485) % segment 3
   ..controls (1.5,3.78351) and (1.97165,5.16177)
 ..(2.68588,5.876) % segment 0
 & cycle

Tracing edges at line 163: (weight 1)
@ Octant ENE (2 offsets), from (7.68588,8.776) to (4.2104,6.5)
@ retrograde line from (7.68588,8.776) to (2.68588,5.876)
(7,9)(7,8)(5,8)(5,7)(4,7)
@ transition line 0, from (2.68588,5.876) to (2.68588,5.876)
@ transition line 1, from (2.68588,5.876) to (7.68588,8.776)
@ transition line 2, from (8.13123,9.12088) to (3.13123,6.22089)
@ transition line 1, from (4.2104,6.5) to (9.2104,9.4)
@ transition line 2, from (9.2104,9.4) to (9.2104,9.4)
(4,6)(4,7)(6,7)(6,8)(8,8)
@ retrograde line from (9.2104,9.4) to (4.2104,6.5)
(8,9)(8,8)(6,8)
@ Octant ESE (0 offsets), from (4.2104,6.5) to (7.41953,4.63048)
@ transition line 1, from (7.41953,4.63048) to (7.41953,4.63048)
(6,7)(4,7)(4,6)(6,6)
@ Octant SSE (0 offsets), from (7.41953,4.63048) to (9.5,0.95631)
@ transition line 0, from (7.41953,4.63048) to (7.41953,4.63048)
(6,5)(8,5)(8,3)(9,3)
@ Octant SSU(0 offsets), from (9.5,0.95631) to (8.90538,-0.56279)
@ transition line 1, from (8.90538,-0.56279) to (8.90538,-0.56279)
@ Octant WSW(2 offsets), from (8.90538,-0.56279) to (7.57066,1.5)
@ retrograde line from (8.90538,-0.56279) to (12.90538,2.43721)
(9,-1)(9,0)(10,0)(10,1)(12,1)
@ transition line 0, from (12.90538,2.43721) to (12.90538,2.43721)
@ transition line 1, from (12.90538,2.43721) to (8.90538,-0.56279)
@ transition line 2, from (8.6665,-0.76936) to (12.6665,2.23064)
@ transition line I, from (7.57066,1.5) to (3.57066,-1.5)
@ transition line 2, from (3.57066,-1.5) to (3.57066,-1.5)
(12,2)(8,2)(8,1)(6,1)(6,0)(5,0)
@ retrograde line from (3.57066,-1.5) to (7.57066,1.5)
(5,-1)(5,0)(6,0)(6,1)(8,1)
@ Octant WNW(2 offsets), from (7.57066,1.5) to (6.85919,1.84764)
@ retrograde line from (7.57066,1.5) to (8.57066,1.4)
(8,2)
@ transition line 3, from (8.57066,1.4) to (8.57066,1.4)
@ transition line 2, from (8.57066,1.4) to (7.57066,1.5)
@ transition line 1, from (6.95764,1.52585) to (7.95764,1.42584)
@ transition line 2, from (6.85919,1.84764) to (5.85919,1.94765)
@ transition line 1, from (5.85919,1.94765) to (5.85919,1.94765)
(8,1)
@ retrograde line from (5.85919,1.94765) to (6.85919,1.84764)
@ Octant NNW (0 offsets), from (6.85919,1.84764) to (6.5,2.90271)
@ transition line 0, from (6.85919,1.84764) to (6.85919,1.84764)
```

```
(8,2)(7,2)
@ Octant NNE (0 offsets), from (6.5,2.90271) to (7.68588,8.776)
@ transition line 1, from (7.68588,8.776) to (7.68588,8.776)
(7,9).

{tracingonline:=1}

f<expr>->let)=];let[=(;show._
(EXPRO)<-xx
{let}
{let}
{show}
>> _1
{showdependencies}
xpart '=-ypart '-0.66667
{qq:=pen}
{showstats}
Memory usage 1458&556 (94 still untouched)
String usage 35&136 (960&8305 still untouched)
([repeat the loop])
! A group begun on line 163 never ended.
<to be read again>
                   endtext
1.163 . . . shosstats; bye endtext

I saw a 'begingroup' back there that hasn't been matched
by 'endgroup'. So I've inserted 'endgroup' now.

{endgroup}
(end occurred when else on line 93 was incomplete)
(end occurred when if on line 36 was incomplete)
(end occurred when elseif on line 21 was incomplete)
Here is how much of METAFONT's memory you used:
 62 strings out of 1022
 3727 string characters out of 12032
 2907 words of memory out of 3001
 277 symbolic tokens out of 2100
 8i,42n,14r,8p,167b stack positions out of 30i,100n,300r,150p,500b

(illegal design size has been changed to 128pt)
(some chardp values had to be adjusted by as much as 0.5pt)
(4 font metric dimensions had to be decreased)
(You used 3w,2h,16d,2i,4l,2k,1e,10p metric file positions
  out of 256w,16h,16d,64i,300l,256k,256e,50p)

Font metrics written on trap.tfm
Output written on trap.72270gf (5 characters, 1028 bytes).
```

Appendix E: The TRAP. TYP file. Here is another major component of the test. It shows the output of **GFtype** applied to the file **TRAP.72270GF** that is created at the same time Appendix D was produced.

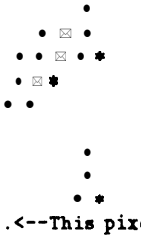
```
This is GFtype,Version 2.2
Options selected: Mnemonic output = true; pixel output = true.
' HETAFOIT output 1986.01.04:0936'

35: beginning of char 5: -4096<=m<=4094 -2<=n<=1
(initially n=1) paint (4095)1
64: skip0 0 (n=0) paint (4093)1
69: skip0 0 (n=-1) paint (4096)1
74: skip0 0 (n=-2) paint (4092)1(1)1
81: eoc
(The character is too large to be displayed in full.)
.<--This pixel's lower left comer is at (-4096,2) in HETAFOIT coordinates



.<--This pixel's upper left comer is at (-4096,-2) in HETAFOIT coordinates

82: beginning of char 5 with extension -14: -4<=m<=1002 -6<=n<=11
(previous character with the same code started at byte 35)
(initially n=11) paint (1002)4
111: skip0 0 (n=10) paint (1002)4
116: skip0 0 (n-9) paint (1002)4
121: skip1 3 (n=5) paint (1002)4
127: skip0 0 (n-4) paint (1002)4
132: skip0 0 (n-3) paint (1002)4
137: newrow 7 (n-2) paint 1
139: newrow 3 (n=1) paint 1(1)1(1)1
145: newrow 1 (n=0) paint 1(1)2(1)1(1)1
153: newrow 1 (n=-1) paint 2(1)1
157: newrow 0 (n=-2) paint 1(1)1
161: skip1 1 (n=-4) paint (7)1
165: newrow 7 (n=-5) paint 1
167: newrow 6 (n=-6) paint 1(1)1
171: eoc
(The character is too large to be displayed in full.)
.<--This pixel's lower left comer is at (-4,12) in HETAFOIT coordinates
```



```
.<--This pixel's upper left corner is at (-4,-6) in HETAFOIT coordinates

172: yyy -216270 (-3.30002)
177: xxx 'title ab'
187: xxx 'xoffset'
196: yyy -65536 (-1)
201: beginning of char 27 with extension -14: -4096<=m<=4094 -1<=n<=0
(initially n=0) paint (4093)1(1)4095
234: newrow 0 (n=-1) paint 4093(1)1
240: eoc
(The character is too large to be displayed in full.)
.<--This pixel's lower left corner is at (-4096,1) in HETAFOIT coordinates
```

```
**********************************************************************************
.<--This pixel's upper left corner is at (-4096,-1) in HETAFOIT coordinates

241: xxx 'xoffset'
250: yyy -87359488 (-1333)
255: xxx 'yoffset'
264: yyy -87359488 (-1333)
269: beginning of char 109: -5428<=m<=2762 -1435<=n<=-1077
(initially n=-1077) paint (0)8190
298: skip2 256 (n=-1334) paint (0)8190
305: skip1 98 (w-1433) paint (4143)55
311: skip0 0 (m-1434) paint (4088)106
317: skip0 0 (w-1435) paint (4092)74
323: eoc
(The character is too large to be displayed in full.)
.C--This pixel's lower left corner is at (-5428,-1076) in HETAFOIT coordinates
**********************************************************************************
```

```
.<--This pixel's upper left corner is at (-5428,-1156) in HETAFOIT coordinates

324: xxx 'bye'
329: xxx 'xoffset'
338: yyy -87359488 (-1333)
343: xxx 'yoffset'
352: yyy -87359488 (-1333)
357: beginning of char 0: 0<=m<=0 0<=n<=0
(initially n=0)
363: eoc
(The character is entirely blank.)

364: xxx 'title careful HETAFOIT HETAFOIT HETAFOIT HETAFOIT HETAFOIT METAF
OIT HETAFOIT HETAFOIT HETAFOIT METAFONT HETAFOIT HETAFOIT HETAFOIT METAFONT HE
TAFONT HETAFOIT HETAFOIT HETAFOIT HETAFOIT HETAFOIT HETAFOIT HETAFOIT HETAFOIT
 METAFONT HETAFOIT HETAFOIT HETAFOIT'
624: xxx 'title hello again?'
624: ! non-ASCII character in xxx command!

Postamble starts at byte 644, after special info at byte 364.
design size = 134217728 (128pt)
check sum = 765094058
hppp = -65536 (-1)
vppp = 65535 (0.99998)
min m = -5428, max m = 4094
min n = -1435, max n = 11
Character 0: dx -950272 (-14.5), dy 393216 (6), width 2097152 (-256), loc 324
Character 5: dx -950272 (-14.5), width 0 (0), loc 82
Character 27: dx -950272 (-14.5), width 0 (0), loc 172
Character 95: dx -950272 (-14.5), width 0 (0), loc -1
Character 96: dx -950272 (-14.5), width 0 (0), loc -1
Character 97: dx -950272 (-14.51, width 0 (0), loc -1
Character 98: dx -950272 (-14.5), width 0 (0), loc -1
Character 99: dx -950272 (-14.5), width 0 (0), loc -1
Character 100: dx -950272 (-14.5), width 0 (0), loc -1
Character 101: dx -950272 (-14.5), width 0 (0), loc -1
Character 102: dx -950272 (-14.5), width 0 (0), loc -1
Character 103: dx -950272 (-14.5), width 0 (0), loc -1
Character 104: dx -950272 (-14.5), width 0 (0), loc -1
Character 105: dx -950272 (-14.5), width 0 (0), loc -1
Character 106: dx -950272 (-14.5), width 0 (0), loc -1
Character 107: dx -950272 (-14.51, width 0 (0), loc -1
Character 108: dx -950272 (-14.5), dy 393216 (6), width 2097152 (-256), loc -1
Character 109: dx -950272 (-14.5), dy 393216 (6), width 2097152 (-256), loc 241
Character 236: dx 0 (0), width 0 (0), loc -1
The file had 5 characters altogether.
```

Appendix F: **The** TRAP.PL file. In this case we have the output of **TFtoPL** applied to the file TRAP **.TFM** that is created at the same time Appendix D was produced.

```
(CODINGSCHEME Q)
(DESIGNSIZE R 128.0)
(COMMENT DESIGISIZE IS II POINTS)
(COMMENT OTHER SIZES ARE HULTIPLES OF DESIGISIZE)
(CHECKSUM 0 5546464252)
(FONTDIMEN
   (SLANT R 2047.999999)
   (SPACE R 0.0)
   (STRETCH R 0.0)
   (SHRINK R 0.0)
   (XHEIGHT R 0.0)
   (QUAD R 0.0)
   (EXTRASPACE R 0.0)
   (PARAMETER D 8 R 0.0)
   (PARAMETER D 9 R 0.005208)
   (PARAMETER D 10 R 0.0)
   )
(LIGTABLE
   (COMMENT THIS PART OF THE PROGRAM IS NEVER USED!
      (LIG C a C b)
      )
   (LABEL C c)
   (KRN C d R -15.999999)
   (LABEL C d)
   (LIG C e C f)
   (KRN 0 0 R 0.0)
   (STOP)
   )
(CHARACTER 0 0
   (CHARWD R 2.0)
   (CHARHT R 15.999999)
   (CHARDP R 0.238281)
   (CHARIC R -7.8125)
   (NEXTLARGER 0 5)
   )
(CHARACTER 0 5
   (CHARWD R 0.0)
   (CHARDP R -0.113281)
   (VARCHAR
      (TOP 0 5)
      (HID C c)
      (REP C d)
      )
   )
(CHARACTER 0 33
   (CHARWD R 0.0)
   (CHARDP R 0.210937s)
   )
(CHARACTER 0 137
   (CHARWD R 0.0)
   (CHARDP R 0.1015625)
   )
(CHARACTER 0 140
   (CHARWD R 0.0)
   (CHARDP R 0.0703125)
   )
(CHARACTER C a
   (CHARWD R 0.0)
   (NEXTLARGER C d)
   )
(CHARACTER C b
   (CHARWD R 0.0)
   (CHARDP R 0.1640625)

(CHARACTER C c
   (CHARWD R 0.0)
   (CHARDP R -0.011719)
   (COMMENT
```

```
      (KRN C d R -15.999999)
      (LIG C e C f)
      (KRN 0 0 R 0.0)

   ,
(CHARACTER c d
   (CHARWD R 0.0)
   (CHARDP R 0.1328125)
   (COMMENT
      (LIG C e C f)
      (KRN 0 0 R 0.0)


(CHARACTER C e
   (CHARWD R 0.0)
   (CHARDP R 0.0859375)

(CHARACTER c f
   (CHARWD R 0.0)
   (CHARDP R 0.0234375)

(CHARACTER c g
   (CHARWD R 0.0)
   (CHARDP R -0.011719)
   (CHARIC R -7.8125)

(CHARACTER c h
   (CHARWD R 0.0)
   (CHARDP R 0.1953125)
   (CHARIC R -7.8125)
   ,
(CHARACTER C i
   (CHARWD R 0.0)
   (CHARDP R 0.238281)
   (CHARIC R -7.8125)

(CHARACTER C j
   (CHARWD R 0.0)
   (CHARDP R 0.1484375)
   (CHARIC R -7.8125)

(CHARACTER c k
   (CHARWD R 0.0)
   (CHARDP R 0.0546875)
   (CHARIC R -7.8125)

(CHARACTER C 1
   (CHARWD R 2.0)
   (CHARDP R 0.1796875)
   (CHARIC R -7.8125)

(CHARACTER C m
   (CHARWD R 2.0)
   (CHARHT R 15.999999)
   (CHARDP R 0.238281)
   (CHARIC R -7.8125)

(CHARACTER 0 354
   (CHARWD R 0.0)
   (CHARDP R -15.999999)
```

Appendix G: The TRAP .FOT file. This shows what appeared on the terminal while Appendix D was being produced.

```
This is METAFONT, Version 1.0(INIMF)
** &trap trap
(trap.mf
\*//*\trap\pass2!
! Missing symbolic token inserted.
<inserted text>
                INACCESSIBLE
1.21 . . .t next=\; delimiters ^^7
                                 ! fi
! Extra tokens will be flushed.
<to be read again>

1.21 . . . next=\; delimiters ^^7!
                                 fi
! Forbidden token found while scanning to the end of the statement.
<inserted text>
                ,
<to be read again>
                  \
1.22 next\
          ; % the second pass will now compute silently; the . . .

hello again-^-
>> pen (see the transcript file)
>> path (see the transcript file)
>> path (see the transcript file)
>> path (see the transcript file)
f<expr>->let)=];let[=(;show._
(EXPRO)<-xx
{let}
{let}
{show}
>> _1
{showdependencies}
xpart '=-ypart'-0.66667
{qq:=pen}
{showstats}
Memory usage 1458&556 (94 still untouched)
String usage 35&136 (960&8305 still untouched)
([repeat the loop])
! A group begun on line 163 never ended.
<to be read again>
                  endtext
1.163 . . . shoastats; bye endtext

{endgroup}
(end occurred when else on line 93 was incomplete)
(end occurred when if on line 36 was incomplete)
(end occurred when elseif on line 21 was incomplete)
(see the transcript file for additional information)
(illegal design size has been changed to 128pt)
```

```
(some chardp values had to be adjusted by as much as 0.5pt)
(4 font metric dimensions had to be decreased)
Font metrics written on trap.tfm
Output written on trap.72270gf (5 characters, 1028 bytes).
Transcript written on trap.log.
```