

作りましょう 0.8

パラメタ方式フォントファミリ
校とプリティプリントのソース

Tsukurimashou 0.8

Parametric Font Family
Proofs and pretty-printed
source code

Matthew Skala

mskala@ansuz.sooke.bc.ca

2013年8月26日

August 26, 2013

Proofs and pretty-printed source code for Tsukurimashou
Copyright © 2011, 2012, 2013 Matthew Skala

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, version 3.

As a special exception, if you create a document which uses this font, and embed this font or unaltered portions of this font into the document, this font does not by itself cause the resulting document to be covered by the GNU General Public License. This exception does not however invalidate any other reasons why the document might be covered by the GNU General Public License. If you modify this font, you may extend this exception to your version of the font, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Contents

I Infrastructure	13
preintro.mp	15
Infrastructure	15
Font Parameter Defaults	16
tsuku-bk.mp	18
tsuku-kg.mp	19
tsuku-mg.mp	20
tsuku-mi.mp	21
tsuku-ps.mp	22
tsuku-bq.mp	23
tsuku-dq.mp	24
tsuku-el.mp	25
tsuku-eq.mp	26
tsuku-lw.mp	27
intro.mp	28
fntbase.mp	46
General Library Functions	46
Prefix And Suffix Handling	53
A Module That Finds An Envelope Of A Path Drawn With A Pen	55
Postscript Font Generation	67
obstack.mp	89
Object Stack Data	89
Object Stack Methods	90
frac-intro.mp	95
latin-intro.mp	97
accent.mp	101
bcircle.mp	112
bkencl.mp	115
buildkanji.mp	121
dakuten.mp	134
enclosed.mp	135
genjimon.mp	137
hiragana.mp	149
Hiragana Vowels	149
Hiragana Kakikukeko/Gagigugego	154
Hiragana Sashisuseso/Zajizuzezo	158

Hiragana Tachitsuteto/Dajizudedo	164
Hiragana Naninuneno	169
Hiragana Hahifuheho/Babibubebo/Papipupepo	174
Hiragana Mamimumemo	179
Hiragana Yayuyo	185
Hiragana Rarirurero	187
Hiragana Wawiwewo/N/Iteration	193
iching.mp	201
katakana.mp	203
Katakana Vowels	203
Katakana Kakikukeko/Gagigugego	207
Katakana Sashisuseso/Zajizuzezo	212
Katakana Tachitsuteto/Dajizudedo	217
Katakana Naninuneno	222
Katakana Hahifuheho/Babibubebo/Papipupepo	227
Katakana Mamimumemo	233
Katakana Yayuyo	237
Katakana Rarirurero	240
Katakana Wawiwewo/N/Iteration	246
latin.mp	252
numerals.mp	341
ogonek.mp	352
punct.mp	364
serif.mp	406
 II Shared kyouiku kanji	 413
gradeone.mp	415
gradetwo.mp	483
gradethree.mp	558
gradefour.mp	601
gradefive.mp	637
gradesix.mp	655
 III Other shared kanji	 669
bottomrad.mp	671
leftrad.mp	674
radical.mp	684
rightrad.mp	722
toprad.mp	727

gradeeight.mp	734
gradenine.mp	780
rare.mp	808

IV U+0000 to U+2FFF 859

tsuku-00.mp	861
Ascii	861
Latin-1 Extra Characters	955
Accented Latin	976
tsuku-01.mp	1034
Latin Extended A Uppercase	1034
Latin Extended A Lowercase	1089
Latin Extended A Other	1147
tsuku-02.mp	1149
Latin Extended B	1149
Spacing Modifier Letters	1150
tsuku-03.mp	1159
Combining Diacritical Marks	1159
tsuku-20.mp	1174
General Punctuation	1174
tsuku-21.mp	1191
Symbols Required By Mes-1	1191
tsuku-24.mp	1197
Circled Numerals	1197
Circled Latin And Zero	1217
Inverted Circled Numerals	1270
Doubly Circled Numerals	1280
One More Inverted Circled Numeral	1290
tsuku-25.mp	1292
Geometric Shapes	1292
tsuku-26.mp	1294
I Ching	1294
tsuku-27.mp	1311
Inverted Circled Numerals	1311
tsuku-2e.mp	1322
Cjk Radicals Supplement	1322
tsuku-2f.mp	1360
Kangxi Radicals	1360
Ideographic Description Characters	1480

V	U+3000 to U+4DFF	1495
	tsuku-30.mp	1497
	Ideographic Symbols And Punctuation	1497
	Hiragana	1517
	Katakana	1562
	tsuku-31.mp	1609
	Phonetic Extensions For Ainu	1609
	tsuku-32.mp	1626
	Circled Numerals	1626
	Circled Katakana	1656
	tsuku-34.mp	1704
	tsuku-4d.mp	1705
	I Ching	1705
VI	U+4E00 to U+61FF	1771
	tsuku-4e.mp	1773
	tsuku-4f.mp	1809
	tsuku-50.mp	1865
	tsuku-51.mp	1878
	tsuku-52.mp	1895
	tsuku-53.mp	1928
	tsuku-54.mp	1957
	tsuku-55.mp	1968
	tsuku-56.mp	1974
	tsuku-57.mp	1984
	tsuku-58.mp	1994
	tsuku-59.mp	2003
	tsuku-5a.mp	2017
	tsuku-5b.mp	2019
	tsuku-5c.mp	2054
	tsuku-5d.mp	2078
	tsuku-5e.mp	2088
	tsuku-5f.mp	2127
	tsuku-60.mp	2186
	tsuku-61.mp	2235
VII	U+6200 to U+75FF	2243
	tsuku-62.mp	2245
	tsuku-63.mp	2266

tsuku-65.mp	2274
tsuku-66.mp	2300
tsuku-67.mp	2317
tsuku-68.mp	2336
tsuku-69.mp	2345
tsuku-6a.mp	2352
tsuku-6b.mp	2358
tsuku-6c.mp	2373
tsuku-6d.mp	2424
tsuku-6e.mp	2439
tsuku-6f.mp	2453
tsuku-70.mp	2458
tsuku-71.mp	2464
tsuku-72.mp	2472
tsuku-73.mp	2485
tsuku-74.mp	2486
tsuku-75.mp	2490

VIII U+7600 to U+89FF	2503
tsuku-76.mp	2505
tsuku-77.mp	2512
tsuku-78.mp	2522
tsuku-79.mp	2524
tsuku-7a.mp	2540
tsuku-7b.mp	2550
tsuku-7c.mp	2565
tsuku-7d.mp	2569
tsuku-7f.mp	2587
tsuku-80.mp	2592
tsuku-81.mp	2601
tsuku-82.mp	2607
tsuku-83.mp	2640
tsuku-84.mp	2654
tsuku-85.mp	2659
tsuku-86.mp	2661
tsuku-87.mp	2662
tsuku-88.mp	2664
tsuku-89.mp	2691

IX	U+8A00 to U+9FFF	2707
	tsuku-8a.mp	2709
	tsuku-8b.mp	2766
	tsuku-8c.mp	2770
	tsuku-8d.mp	2781
	tsuku-8e.mp	2785
	tsuku-8f.mp	2790
	tsuku-90.mp	2804
	tsuku-91.mp	2824
	tsuku-92.mp	2836
	tsuku-93.mp	2840
	tsuku-95.mp	2844
	tsuku-96.mp	2862
	tsuku-97.mp	2885
	tsuku-98.mp	2889
	tsuku-99.mp	2903
	tsuku-9a.mp	2908
	tsuku-9b.mp	2913
	tsuku-9c.mp	2916
	tsuku-9e.mp	2918
	tsuku-9f.mp	2921
X	U+A000 to U+10FFFF	2923
	tsuku-f7.mp	2925
	Latin Small Caps	2925
	tsuku-f9.mp	2952
	tsuku-ff.mp	2954
	Full-Width Forms	2954
	Half-Width Punctuation	2980
	Half-Width Katakana	2983
	tsuku-1f1.mp	3047
	Squared Latin	3047
	Inverse Circled Latin	3073
	Inverse Squared Latin	3099
	tsuku-200.mp	3126
	tsuku-20a.mp	3127
	tsuku-21c.mp	3128
	tsuku-295.mp	3129
	tsuku-f17.mp	3130

Combining Dots For I Ching	3130
Miscellaneous	3139
Tomoe Ornaments	3145
Heavy Metal Umlaut	3153
Genjimon	3170
tsuku-ff0.mp	3224
Fraction Numerators	3224
tsuku-ff1.mp	3231
Fraction Denominators	3231
 XI Jieubsida core	 3233
hangul.mp	3235
Jamo Combining Operations	3237
jamo-basic.mp	3245
Filler Jamo	3245
Sios/Cieuc/Chieuch Family	3245
Kiyeok	3248
Nieun	3249
Tikeut	3251
Rieul	3253
Mieum	3254
Pieup	3255
Sios	3256
Ieung	3256
Cieuc	3257
Chieuch	3258
Khieukh	3259
Thieuth	3260
Phieuph	3261
Hieuh	3263
Mixed Tails	3264
Vowels	3265
jamo-extra.mp	3274
Pansios	3277
Yesieung	3277
Yeorinhieuh	3278
Chitueum And Ceongchieum Variants	3280
Kapyeoun Variants	3281
hglxtb.mp	3282

Hangul Extension B	3282
Hangul Jungseong (Vowel) Jamo Extension B	3282
Hangul Jongseong (Tail) Jamo Extension B	3305
jieub-bt.mp	3355
jieub-do.mp	3356
jieub-sm.mp	3357
hglpage.mp	3358
jieub-l1.mp	3360
Hangul Choseong (Lead) Jamo	3360
Hangul Jungseong (Vowel) Jamo	3455
Hangul Jongseong (Tail) Jamo	3526
jieub-31.mp	3615
Hangul Compatibility Jamo	3615
jieub-a9.mp	3709
Hangul Choseong (Lead) Jamo Extended A	3709
jieub-ac.mp	3739
 XII Jieubsida alternates	 3793
jieub-ff2.mp	3795
Hangul Jungseong (Vowel) Jamo	3795
Hangul Jungseong (Vowel) Jamo Extension B	3866
jieub-ff3.mp	3890
Hangul Choseong (Lead) Jamo	3890
jieub-ff4.mp	4015
Hangul Choseong (Lead) Jamo	4015
jieub-ff5.mp	4140
Hangul Choseong (Lead) Jamo	4140
jieub-ff6.mp	4265
Hangul Choseong (Lead) Jamo	4265
jieub-ff7.mp	4390
Hangul Choseong (Lead) Jamo	4390
Hangul Choseong (Lead) Jamo Extended A	4485
 XIII TsuIta	 4515
tsuita-common.mp	4517
tsuita-at.mp	4528
tsuita-so.mp	4529
Additional Proofs	4531

XIV	Blackletter Lolita	4585
	bll.mp	4587
	bll-co.mp	4592
	pentacross.mp	4593
	Utilities For Pentagrams And Crosses	4593
	bll-f5c.mp	4595
	Pentagrams	4595

Volume XIV

Blackletter Lolita

bll.mp

BLL

bll-co.mp

CO

pentacross.mp

PENT

bll-f5c.mp

F5C

bll.mp

BLL

```
1 %
2 % Blackletter Lolita overrides for Tsukurimashou
3 % Copyright (C) 2011, 2012 Matthew Skala
4 %
5-29 [Standard copyright notice]
30
31 _____
32
33 familyname:="BLL";
34
35 is_blackletter:=true;
36
37 _____
38
39 pair bl_stroke_dir[];
40 numeric bl_stroke_width[];
41
42 bl_stroke_dir[0]:=dir 0;
43 bl_stroke_dir[1]:=dir 45;
44 bl_stroke_dir[2]:=dir 90;
45 bl_stroke_dir[3]:=dir 135;
46 bl_stroke_dir[4]:=dir 180;
47 bl_stroke_dir[5]:=dir 225;
48 bl_stroke_dir[6]:=dir 270;
49 bl_stroke_dir[7]:=dir 315;
50 num_bl_strokes:=8;
51
52 bl_stroke_width[0]:=0.4;
53 bl_stroke_width[1]:=0.6;
54 bl_stroke_width[2]:=1.6;
55 bl_stroke_width[3]:=0.8;
56 bl_stroke_width[4]:=0.4;
57 bl_stroke_width[5]:=0.6;
58 bl_stroke_width[6]:=1.6;
59 bl_stroke_width[7]:=0.8;
60
61 for state:=0 upto num_bl_strokes-1:
62   bl_stroke_dir[state]:=bl_stroke_dir[state]/abs(bl_stroke_dir[state]);
63 endfor;
64
65 _____
66
67 vardef tsu_render_segment(expr i,p,q) =
68   begingroup
69     save lp,glyph,pcorner,pdir,k,ta,tz,dl,dr,dp,sl,sr,goodness,best,dtl,dtr,
70     itl,itr;
```

```

71 path lp,glyph;
72 numeric pdir[],k,ta,tz,sl,sr,goodness,best,dtl,dtr,itl,itr;
73 pair pcorner[],dl,dr,dp;
74
75 % for debugging - dot the path to be approximated
76 if false:
77     for j:=0 step 0.25 until length p:
78         glstk[nxls]:=fullcircle scaled 25 shifted point j of p;
79         nxls:=nxls+1;
80     endfor;
81 fi;
82
83 pcorner[0]:=point 0 of p;
84 pdir[0]=-1;
85 k:=0;
86 ta:=0;
87 forever:
88     k:=k+1;
89
90     if ta=floor(ta):
91         dp:=(postcontrol ta of p)-(point ta of p);
92     else:
93         dp:=direction ta of p;
94     fi;
95     dp:=dp/abs(dp);
96
97     tz:=floor(ta+1);
98     forever:
99         exitif tz=length p;
100         dr:=(point tz of p)-(precontrol tz of p);
101         exitif (dr dotprod dp)<=0;
102         dl:=(postcontrol tz of p)-(point tz of p);
103         dl:=dl/abs(dl);
104         dr:=dr/abs(dr);
105         exitif (dl dotprod dr)<0.95;
106         tz:=tz+1;
107     endfor;
108
109     dr:=(point tz of p)-(point ta of p);
110     dr:=dr/abs(dr);
111     dp:=(dp+0.05*dr)/abs(dp+0.05*dr);
112
113     dl:=bl_stroke_dir[0];
114     dr:=bl_stroke_dir[num_bl_strokes-1];
115     sl:=0;
116     sr:=num_bl_strokes-1;
117     best:=(dl dotprod dp)+(dr dotprod dp);
118

```



```

119     for j:=0 upto num_bl_strokes-2:
120         goodness:=(bl_stroke_dir[j] dotprod dp)
121             +(bl_stroke_dir[j+1] dotprod dp);
122         if goodness>best:
123             best:=goodness;
124             dr:=bl_stroke_dir[j];
125             dl:=bl_stroke_dir[j+1];
126             sr:=j;
127             sl:=j+1;
128         fi;
129     endfor;
130 % message "ta "&(decimal ta)&
131 % " tz "&(decimal tz)&
132 % " sl "&(decimal sl)&
133 % " sr "&(decimal sr);
134
135     if (((point tz of p)-(point ta of p)) dotprod dl>=
136         abs((point tz of p)-(point ta of p))*0.999)
137         and (tz-ta<3):
138         pdir[k]:=sl;
139     elseif (((point tz of p)-(point ta of p)) dotprod dr>=
140         abs((point tz of p)-(point ta of p))*0.999)
141         and (tz-ta<3):
142         pdir[k]:=sr;
143     else:
144         itl:=xpart (p intersectiontimes
145             (((dl*10)+point ta of p)-((dl*1000)+point ta of p)));
146         itr:=xpart (p intersectiontimes
147             (((dr*10)+point ta of p)-((dr*1000)+point ta of p)));
148         if (itl<ta+0.01) or (itl>tz):
149             itl:=-1;
150         fi;
151         if (itr<ta+0.01) or (itr>tz):
152             itr:=-1;
153         fi;
154         dtl:=directiontime dl of p;
155         dtr:=directiontime dr of p;
156         if (dtl<ta+0.01) or (dtl>tz):
157             dtl:=-1;
158         fi;
159         if (dtr<ta+0.01) or (dtr>tz):
160             dtr:=-1;
161         fi;
162         if (itl>ta) and ((itl<=itr) or (itr<ta)):
163             tz:=itl;
164             pdir[k]:=sl;
165         elseif itr>ta:
166             tz:=itr;

```

```

167     pdir[k]:=sr;
168 elseif (dtl>ta) and ((dtl<dtr) or (dtr<ta)):
169     tz:=dtl;
170     pdir[k]:=sr;
171     pcorner[k]:=(whatever*dr)+point ta of p;
172     pcorner[k]=(whatever*dl)+point tz of p;
173     if k>1:
174         if pdir[k]=pdir[k-1]:
175             pcorner[k-1]:=pcorner[k];
176             k:=k-1;
177         fi;
178     fi;
179     k:=k+1;
180     pdir[k]:=sl;
181 elseif dtr>ta:
182     tz:=dtr;
183     pdir[k]:=sl;
184     pcorner[k]:=(whatever*dl)+point ta of p;
185     pcorner[k]=(whatever*dr)+point tz of p;
186     if k>1:
187         if pdir[k]=pdir[k-1]:
188             pcorner[k-1]:=pcorner[k];
189             k:=k-1;
190         fi;
191     fi;
192     k:=k+1;
193     pdir[k]:=sr;
194 elseif false and (pdir[k-1]=sr):
195     pcorner[k-1]:=whatever*dr+point ta of p;
196     pcorner[k-1]:=whatever*dl+point tz of p;
197     pdir[k]:=sl;
198 elseif false and (pdir[k-1]=sl):
199     pcorner[k-1]:=whatever*dl+point ta of p;
200     pcorner[k-1]:=whatever*dr+point tz of p;
201     pdir[k]:=sr;
202 elseif abs(ypart dl)>abs(ypart dr):
203     pdir[k]:=sr;
204     pcorner[k]:=whatever*dr+point ta of p;
205     pcorner[k]:=whatever*dl+point tz of p;
206     k:=k+1;
207     pdir[k]:=sl;
208 else:
209     pdir[k]:=sl;
210     pcorner[k]:=whatever*dl+point ta of p;
211     pcorner[k]:=whatever*dr+point tz of p;
212     k:=k+1;
213     pdir[k]:=sr;
214 fi;

```

```

215         fi;
216         pcorner[k]:=point tz of p;
217
218         if k>1:
219             if pdir[k]=pdir[k-1]:
220                 pcorner[k-1]:=pcorner[k];
221                 k:=k-1;
222             fi;
223         fi;
224
225         exitif tz>=length p;
226         ta:=tz;
227     endfor;
228
229     for j:=1 upto k:
230         if (abs(pcorner[j-1]-pcorner[j])>10) and (pdir[j]>=0):
231             lp:=subpath (0.01,0.99) of (pcorner[j-1]-pcorner[j]);
232             default_nib:=fix_nib(obstackna.bosize[i]*tsu_brush_max
233                 *bl_stroke_width[pdir[j]],
234                 obstackna.bosize[i]*tsu_brush_max*tsu_brush_shape
235                 *bl_stroke_width[pdir[j]],
236                 tsu_brush_angle);
237             pen_stroke()(lp)(glyph);
238             glstk[nxls]:=regenerate(glyph);
239             nxls:=nxls+1;
240         fi;
241     endfor;
242 endgroup;
243 enddef;

```

bll-co.mp

```
1 %
2 % Blackletter Lolita Cosette
3 % Copyright (C) 2011 Matthew Skala
4 %
5-29 [Standard copyright notice]
30
31 % BLL COSETTE
32
33 input preintro.mp;
34
35 stylename:="Cosette";
36
37 (0,4) transformed tsu_brush_xf = (4,0.75);
38 (1,1) transformed tsu_brush_xf = (1,0.62);
39 (4,0) transformed tsu_brush_xf = (0,0.75);
40
41 tsu_brush_min:=0.62;
42 tsu_brush_max:=0.75;
43
44 def tsu_brush_opt(expr n,l) = cut(n,rel 90)(l) enddef;
45 sharp_corners:=true;
46
47 input intro.mp;
48 input bll.mp;
```

pentacross.mp

```
1 %
2 % Pentagrams and crosses for Blackletter Lolita
3 % Copyright (C) 2011 Matthew Skala
4 %
```

```
5-29 [Standard copyright notice]
```

```
30
31 inclusion_lock(pentacross);
32
33
34
```

Utilities For Pentagrams And Crosses

```
35 % UTILITIES FOR PENTAGRAMS AND CROSSES
36
37 path my_nib,my_path;
38
39 % Golden Ratio
40 phi:=(1+sqrt(5))/2;
41
42 % lw - line width
43 % ct - corner type, use -1 for rounded, 0 for bevelled, 1 for mitred
44 % dp - path to draw
45 vardef draw_stroked(expr lw,ct)(expr dp) =
46   default_nib:=fix_nib(lw,lw,0);
47   if ct<0:
48     draw_stroked_opts()(dp);
49   else:
50     draw_stroked_opts(tip(ct))(0 for i:=1 upto length dp: ,i endfor))(dp);
51   fi;
52 enddef;
53
54 vardef draw_stroked_opts(text myopts)(expr dp) =
55   begingroup
56     save glyph;
57     path glyph;
58     pen_stroke(myopts)(dp)(glyph);
59     if cycle dp:
60       glyph.r:=regenerate(glyph.r);
61       glyph.l:=regenerate(glyph.l);
62       dangerousFill glyph.r;
63       dangerousFill glyph.l;
64     else:
65       glyph:=regenerate(glyph);
66       dangerousFill glyph;
67     fi;
68   endgroup;
```

PENT

PENT

```
69 enddef;
70
71 path pentagram;
72 pentagram:=
73 ((dir 0)-(dir 144)-(dir 288)-(dir 72)-(dir 216)-cycle)
74 rotated 90 scaled 0.5;
75
76 % "tip isolated" pentagram, used to de-emphasize corners
77 path tipentagram;
78 tipentagram:=insert_nodes(pentagram)
79 (0.15,0.85,1.15,1.85,2.15,2.85,3.15,3.85,4.15,4.85);
80
81 vardef cross_path(expr hwid) =
82 begingroup
83   save x,y;
84   numeric x,y;
85   z1=(0,0.5); % spectacles
86   z2=(0,-0.5); % testicles
87   z3=(y3-y1,(phi-1)[y2,y1]); % wallet
88   z4=(y1-y3,y3); % watch
89   x5=x12=x13=x16=x1-hwid;
90   x6=x7=x10=x11=x1+hwid;
91   x8=x9=x4;
92   x14=x15=x3;
93   y5=y6=y1;
94   y7=y8=y15=y16=y3+hwid;
95   y9=y10=y13=y14=y3-hwid;
96   y11=y12=y2;
97   z5-z6-z7-z8-z9-z10-z11-z12-z13-z14-z15-z16-cycle
98 endgroup
99 enddef;
```

bll-f5c.mp

```
1 %
2 % Unicode page F5C (pentagrams and crosses) for Blackletter Lolita
3 % Copyright (C) 2011, 2012 Matthew Skala
4 %
5-29 [Standard copyright notice]
30
31
32
33 beginfont
34
35 % AUTODEPS
36 input pentacross.mp;
37
38 do_late_includes;
39
40
41
```

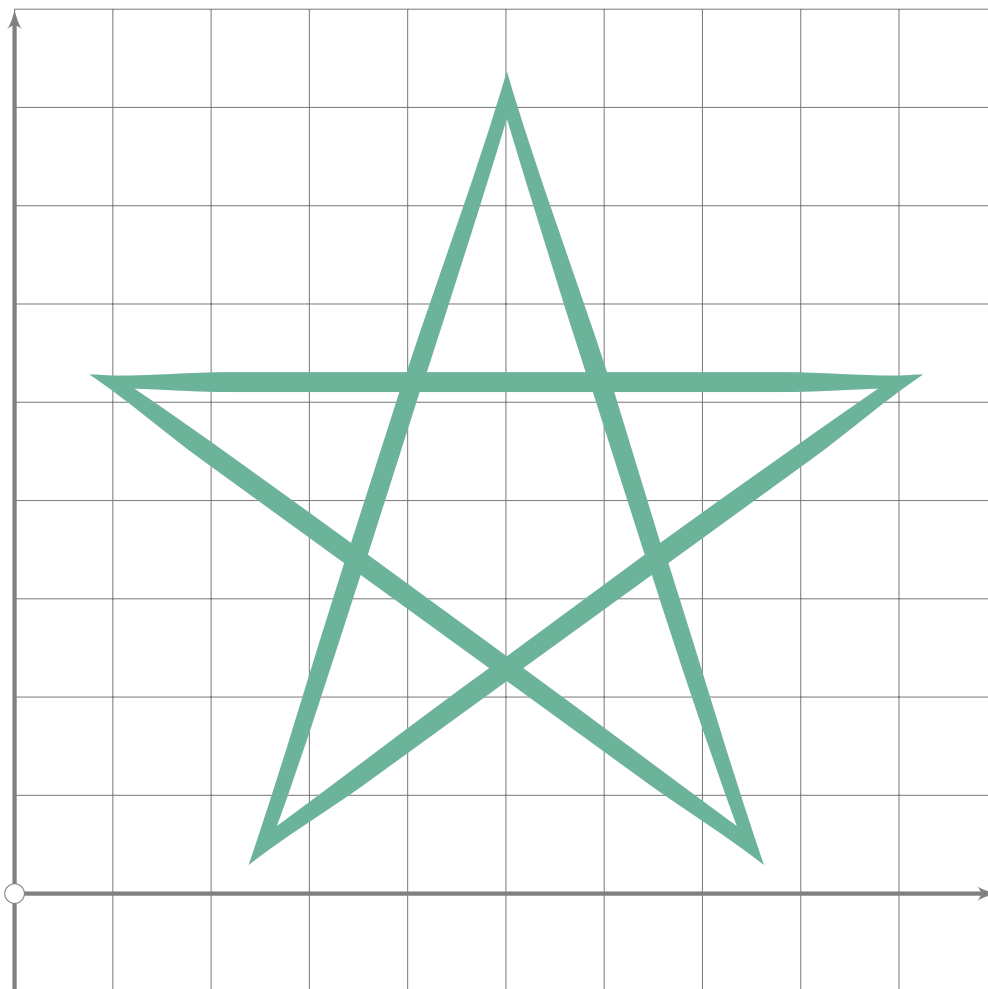
F5C

Pentagrams

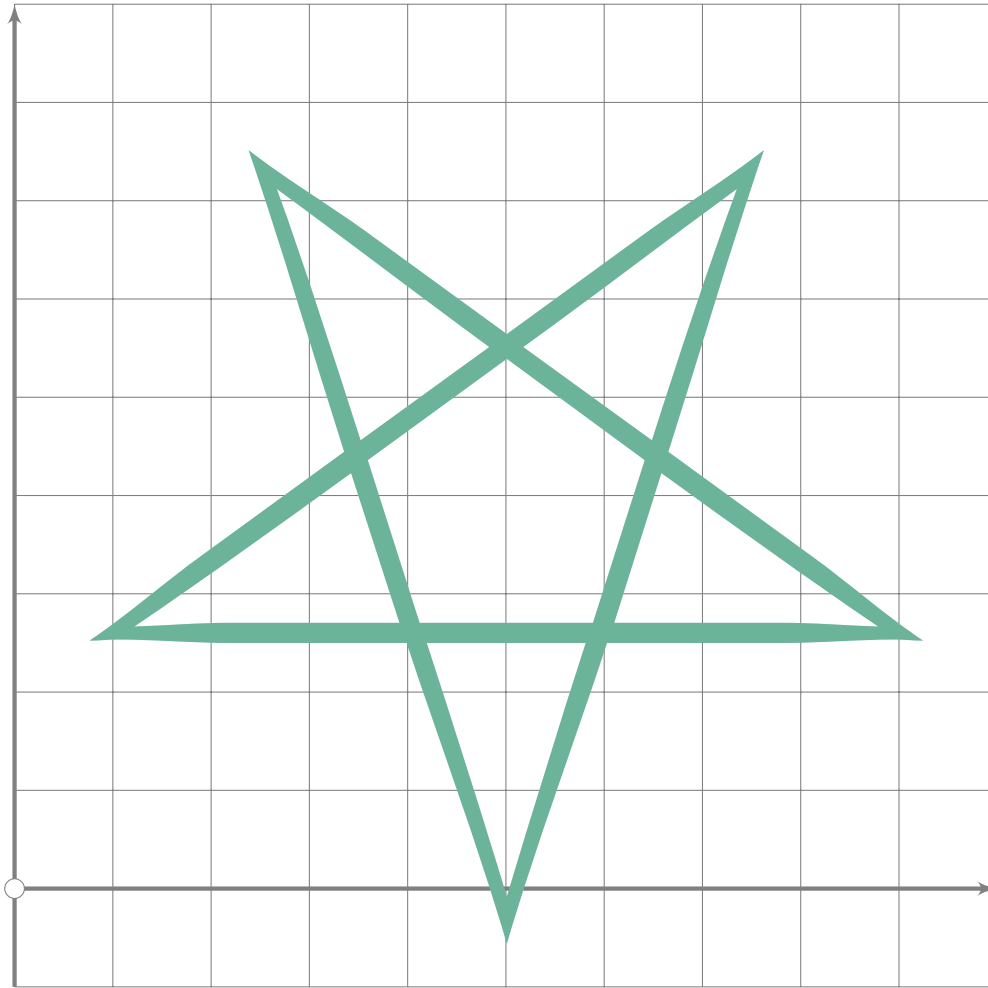
42 % PENTAGRAMS

U+F5C01
bll.pentagram01

F5C



```
43
44 begintsuglyph("pentagram01";1);
45   default_nib:=fix_nib(20,20,0);
46   my_nib:=fix_nib(14,14,0);
47   draw_stroked_opts(tip(my_nib,1,1)(0,3,6,9,12,15))
48     (tipentagram scaled 844 shifted centre_pt);
49 endsuglyph;
```

F5C

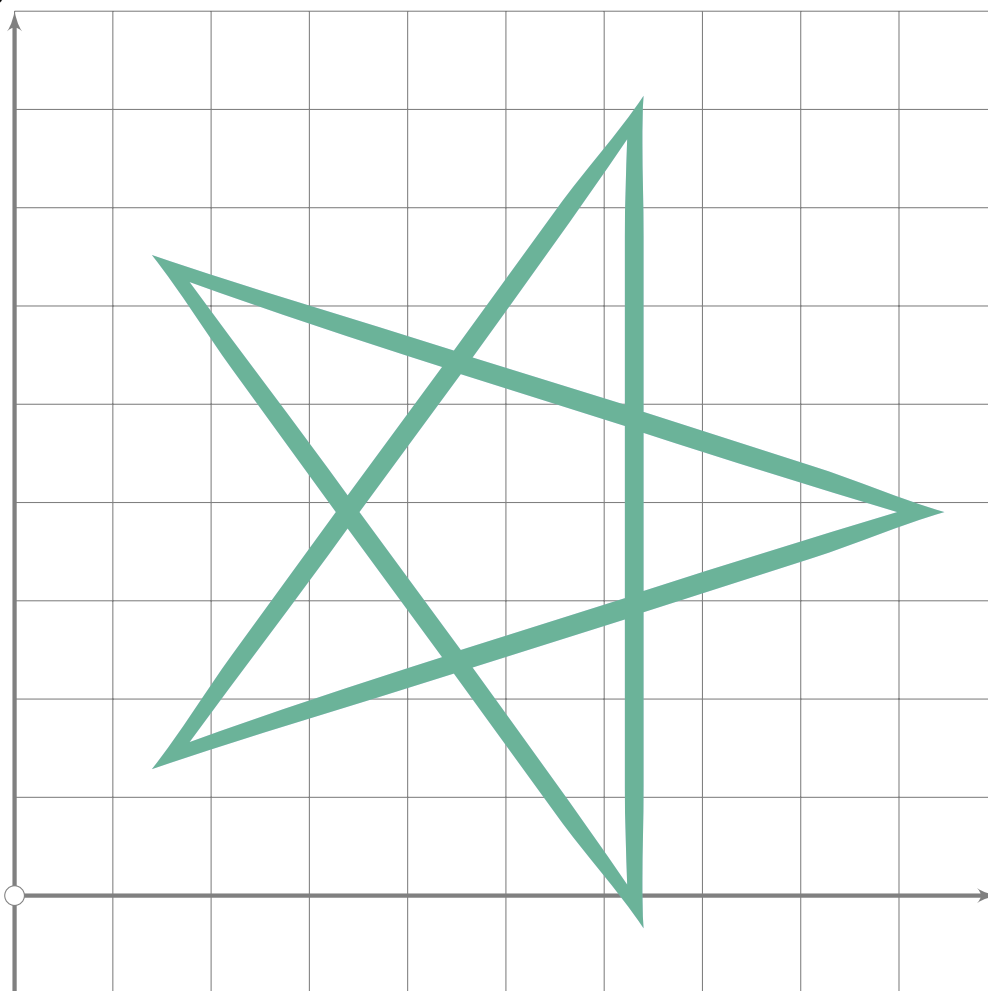
```

50
51 begintsuglyph("pentagram02";2);
52  default_nib:=fix_nib(20,20,0);
53  my_nib:=fix_nib(14,14,0);
54  draw_stroked_opts(tip(my_nib,1,1)(0,3,6,9,12,15))
55    (tipentagram rotated 180 scaled 844 shifted centre_pt);
56 endsuglyph;

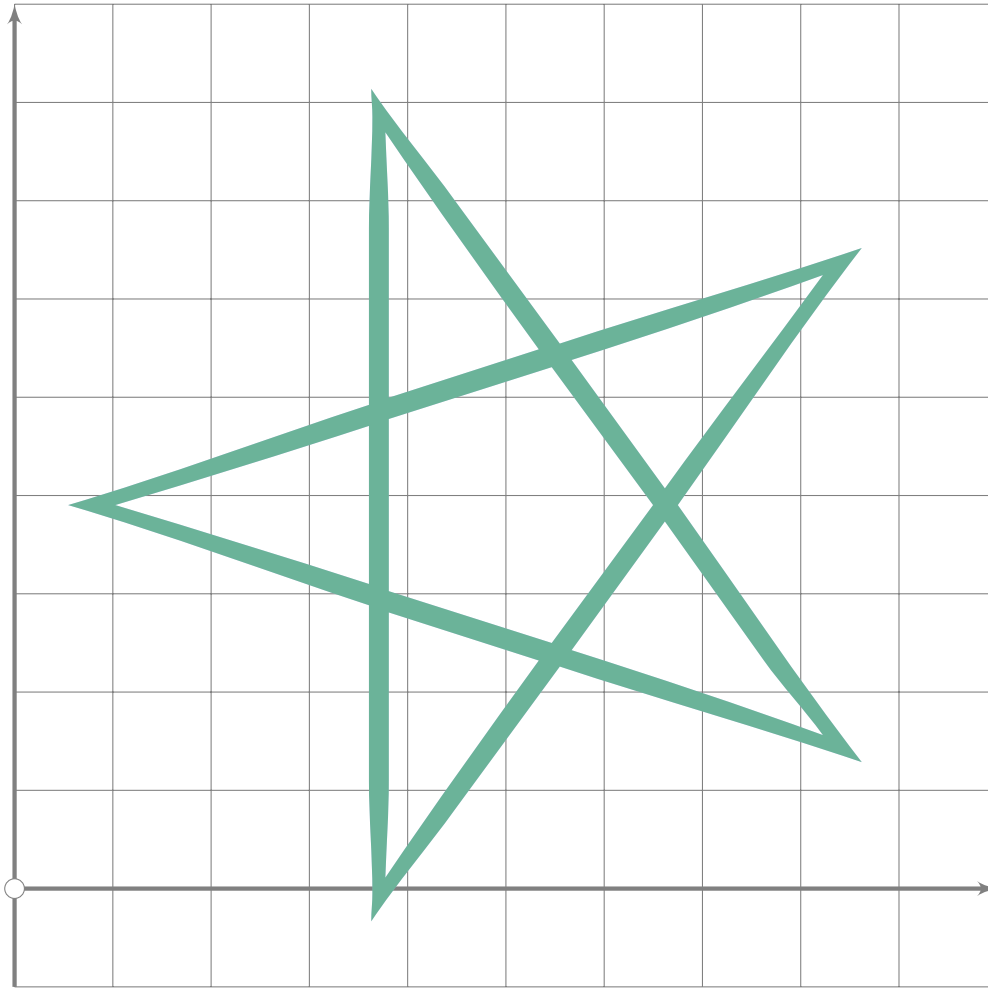
```

U+F5C03
bll.pentagram03

F5C



```
57
58 begintsuglyph("pentagram03";3);
59 default_nib:=fix_nib(20,20,0);
60 my_nib:=fix_nib(14,14,0);
61 draw_stroked_opts(tip(my_nib,1,1)(0,3,6,9,12,15))
62   (tipentagram rotated 270 scaled 844 shifted centre_pt);
63 endsuglyph;
```



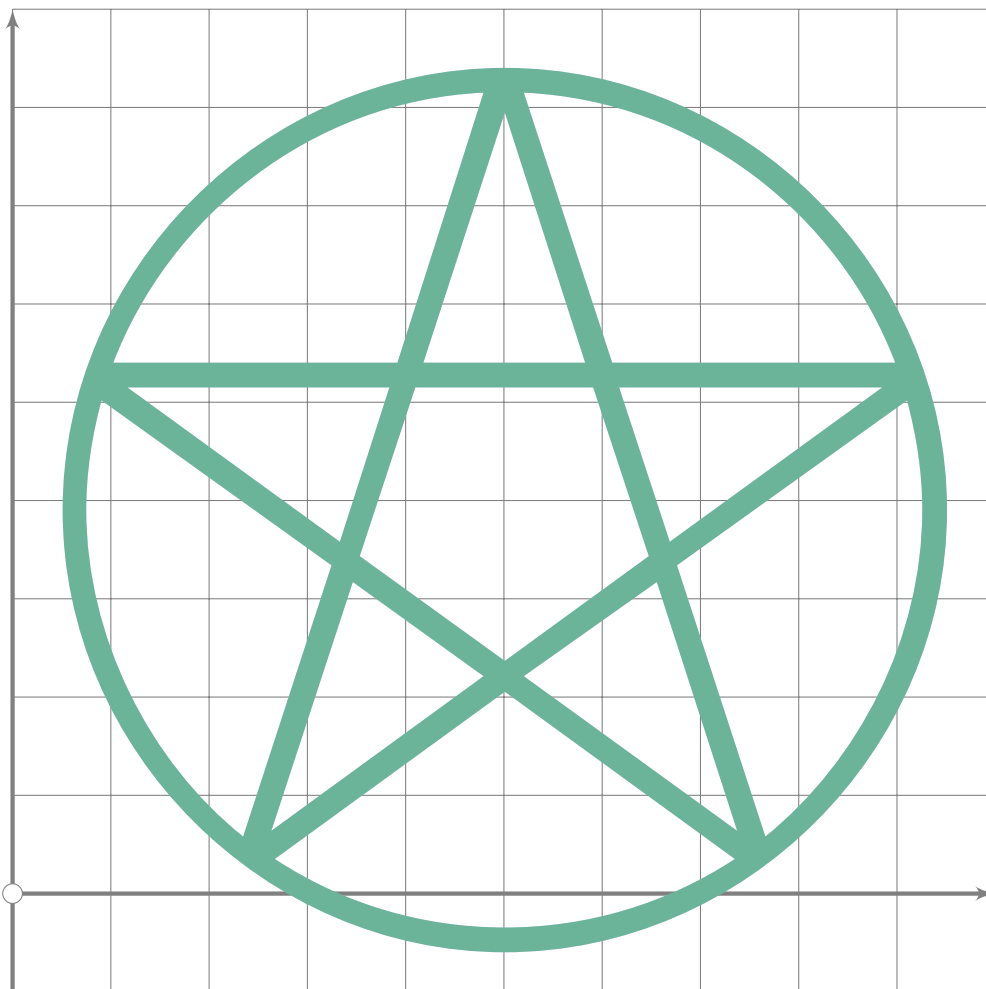
F5C

```

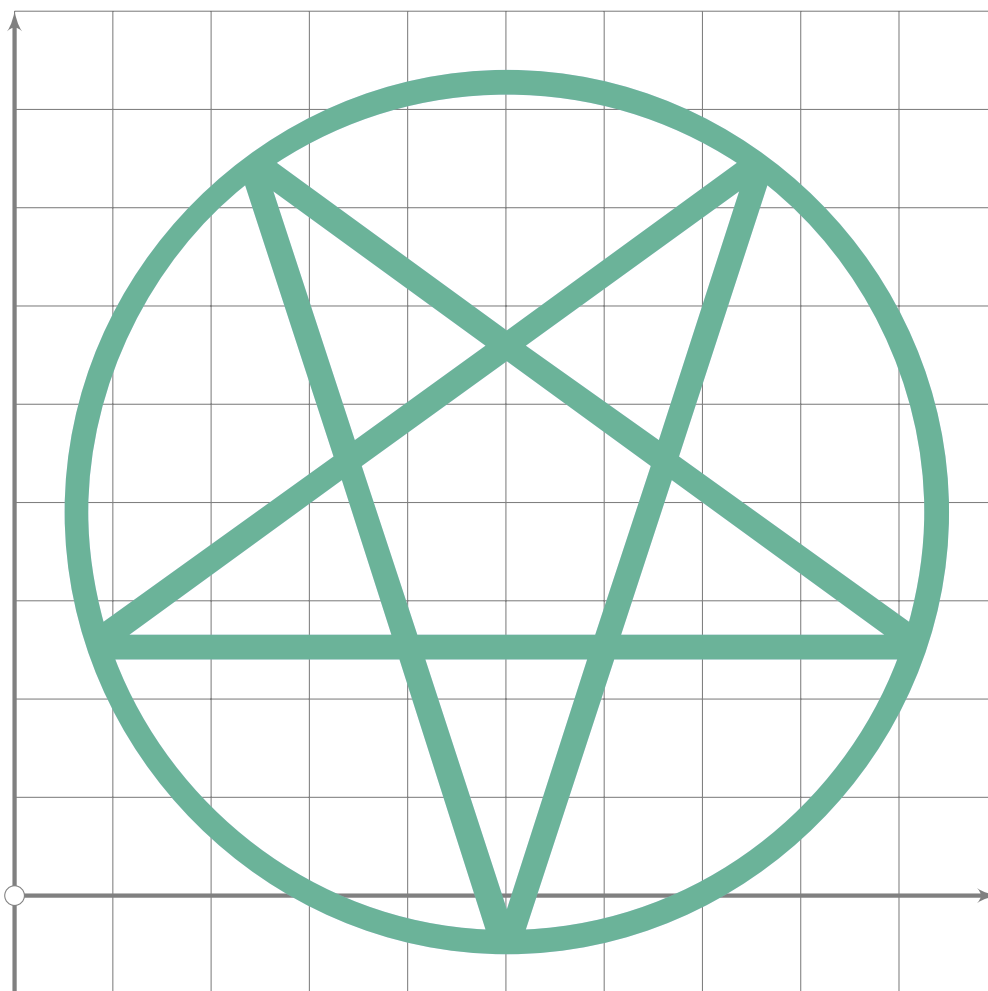
64
65 beginsuglyph("pentagram04";4);
66 default_nib:=fix_nib(20,20,0);
67 my_nib:=fix_nib(14,14,0);
68 draw_stroked_opts(tip(my_nib,1,1)(0,3,6,9,12,15))
69   (tipentagram rotated 90 scaled 844 shifted centre_pt);
70 endsuglyph;
```

U+F5C05
bll.pentagram05

F5C



```
71  
72 begintsuglyph("pentagram05";5);  
73 draw_stroked(25,0)(pentagram scaled 888 shifted centre_pt);  
74 draw_stroked(25,-1)(fullcircle scaled 875 shifted centre_pt);  
75 endsuglyph;
```



F5C

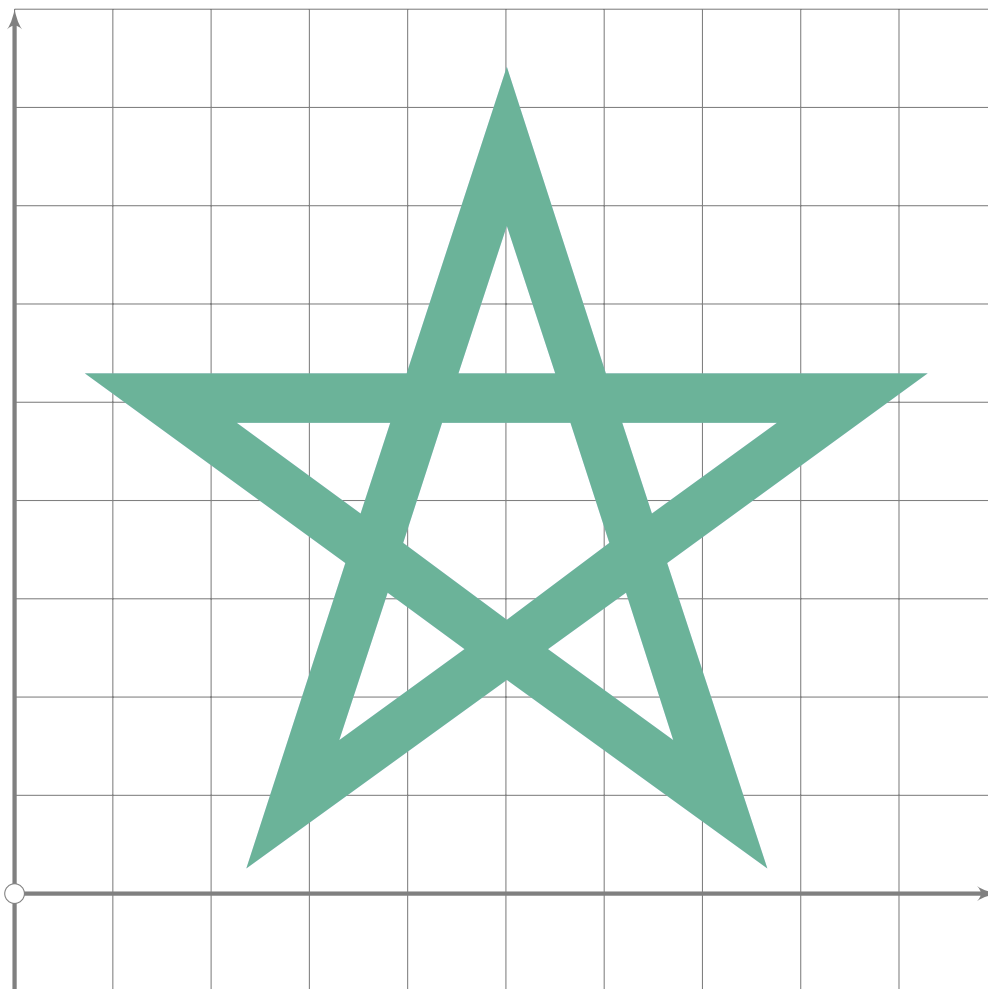
```

76
77 begintsuglyph("pentagram06",6);
78   draw_stroked(25,0)(pentagram rotated 180 scaled 888 shifted centre_pt);
79   draw_stroked(25,-1)(fullcircle scaled 875 shifted centre_pt);
80 endsuglyph;

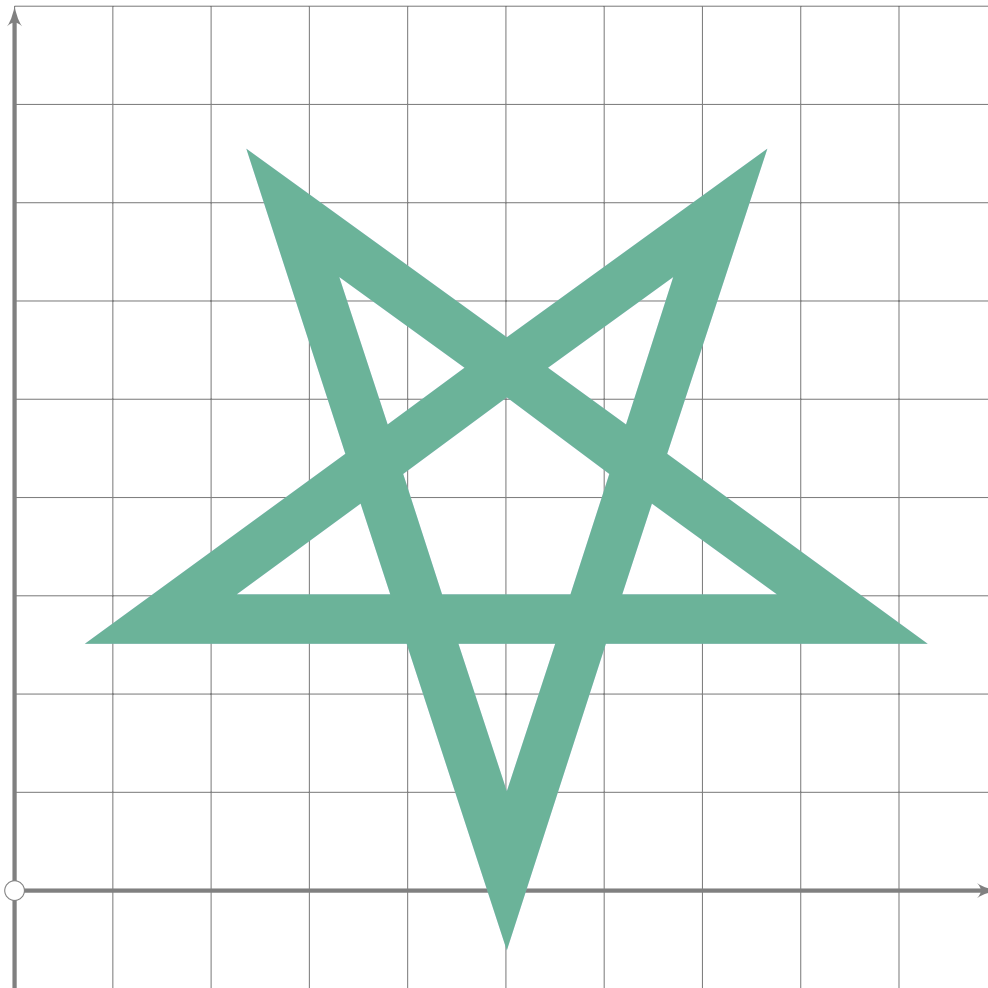
```

U+F5C07
bll.pentagram07

F5C



```
81  
82 begintsuglyph("pentagram07";7);  
83 draw_stroked(50,1)(pentagram scaled 740 shifted centre_pt);  
84 endsuglyph;
```



F5C

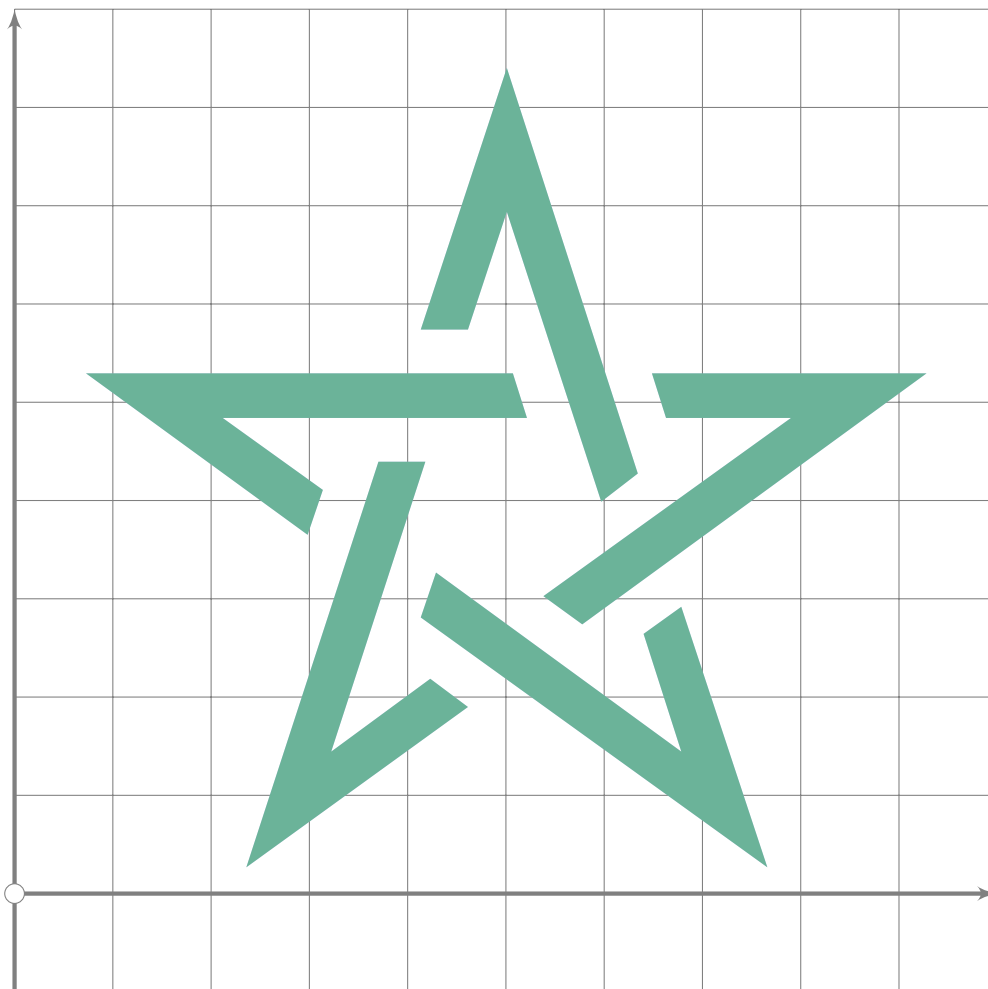
```

85
86 begintsuglyph("pentagram08";8);
87 draw_stroked(50,1)(pentagram rotated 180 scaled 740 shifted centre_pt);
88 endsuglyph;
89
90 vardef penta_ell(expr lw,loff) =
91   begingroup
92     save myl;
93     path myl[];
94     myl1:=(dir 90)–(dir 234);
95     myl2:=((dir 162)–(dir 18)) shifted (loff*dir 90);
96     myl3:=myl1 shifted (lw*dir 342);
97     myl6:=(dir 90)–(dir 306);
98     myl4:=myl6 shifted (lw*dir 198);
99     myl5:=((dir 18)–(dir 234)) shifted ((loff+lw)*dir 126);
100    (dir 90)–(myl6 intersectionpoint myl5)–
101      (myl5 intersectionpoint myl4)–(myl4 intersectionpoint myl3)–
102      (myl3 intersectionpoint myl2)–(myl2 intersectionpoint myl1)–cycle
103   endgroup
104 enddef;

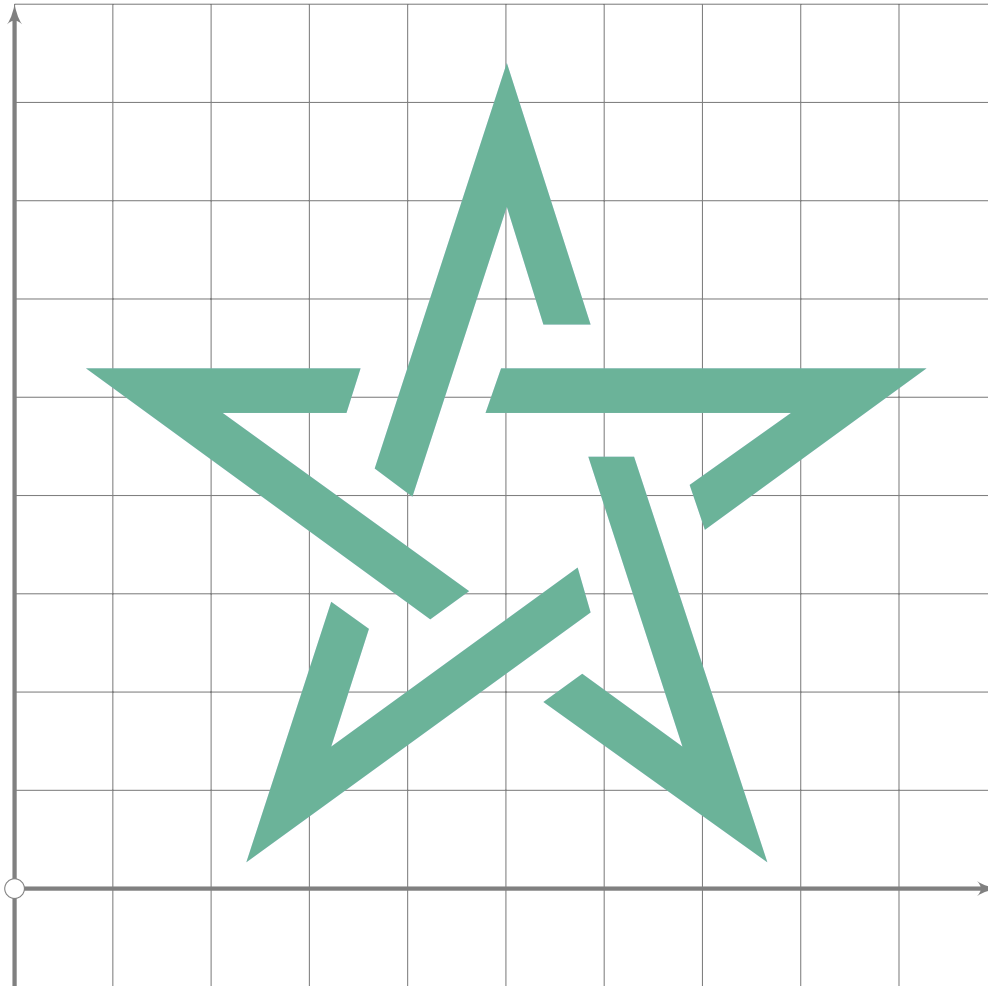
```

U+F5C09
bll.pentagram09

F5C



```
105
106 begint Suglyph("pentagram09";9);
107   my_path:=penta_ell(0.1,0.1);
108   for i:=0 upto 4:
109     dangerousFill my_path rotated (i*72) scaled 450 shifted centre_pt;
110   endfor;
111 endtsuglyph;
```

F5C

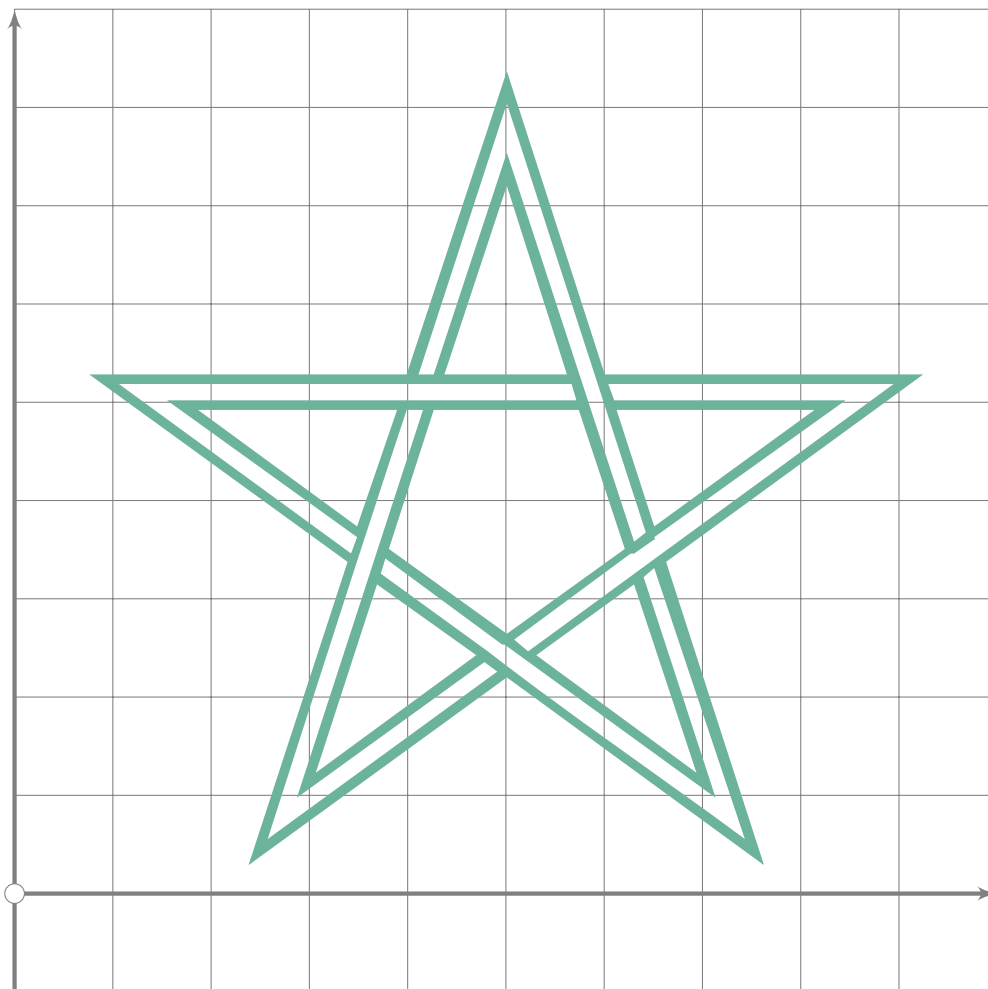
```

112
113 begintsuglyph("pentagram10",10);
114   my_path:=penta_ell(0.1,0.1);
115   for i:=0 upto 4:
116     dangerousFill my_path
117       reflectedabout (down,up) rotated (i*72) scaled 450 shifted centre_pt;
118   endfor;
119 endtsuglyph;

```

U+F5C0B
bll.pentagram11

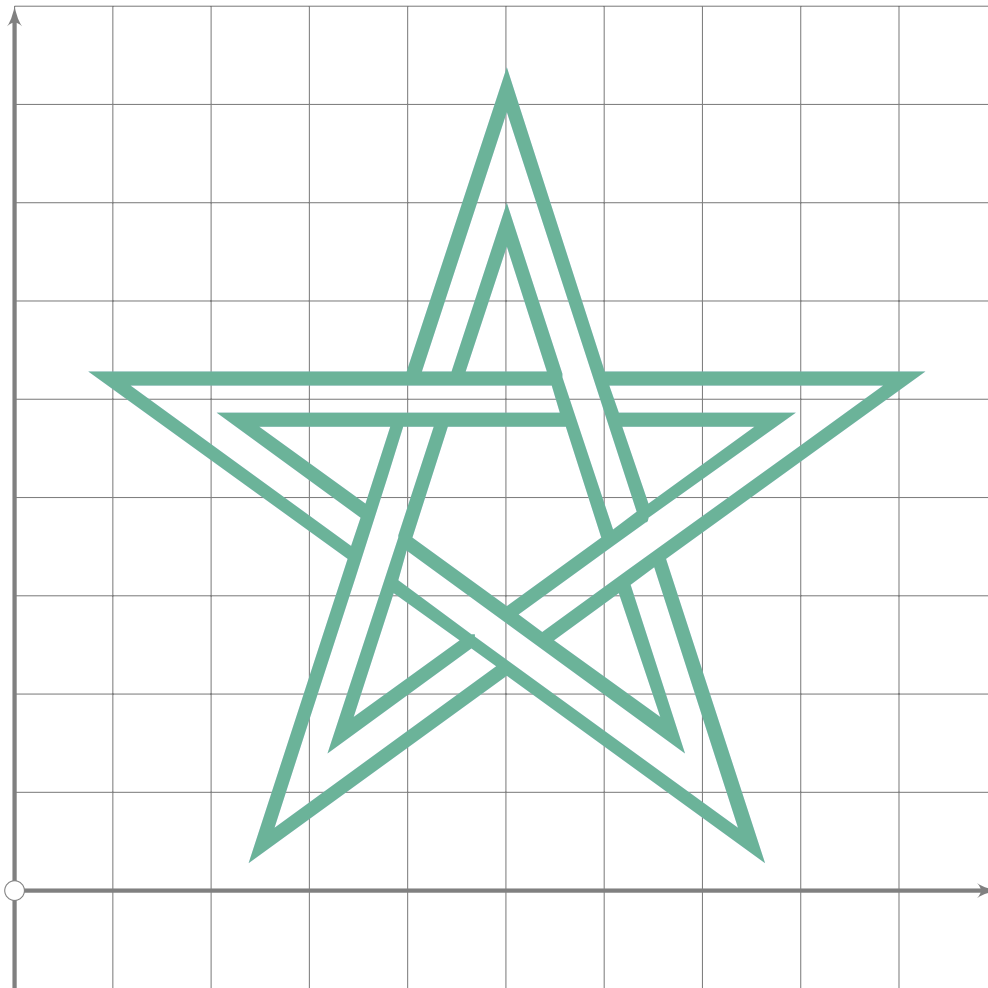
F5C



```

120
121 begintsuglyph("pentagram11",11);
122   begingroup
123     save lp;
124     path lp[];
125     my_path:=penta_ell(0.06,0);
126     lp1:=subpath (5,7) of my_path;
127     lp2:=subpath (2,4) of my_path;
128     default_nib:=fix_nib(10,10,0);
129     pen_stroke(tip(1)(1))(lp1 scaled 430 shifted centre_pt)(lp3);
130     pen_stroke(tip(1)(1))(lp2 scaled 430 shifted centre_pt)(lp4);
131     lp3:=regenerate(lp3);
132     lp4:=regenerate(lp4);
133     for i:=0 upto 4:
134       dangerousFill lp3 rotatedaround (centre_pt,i*72);
135       dangerousFill lp4 rotatedaround (centre_pt,i*72);
136     endfor;
137   endgroup;
138 endtsuglyph;

```



F5C

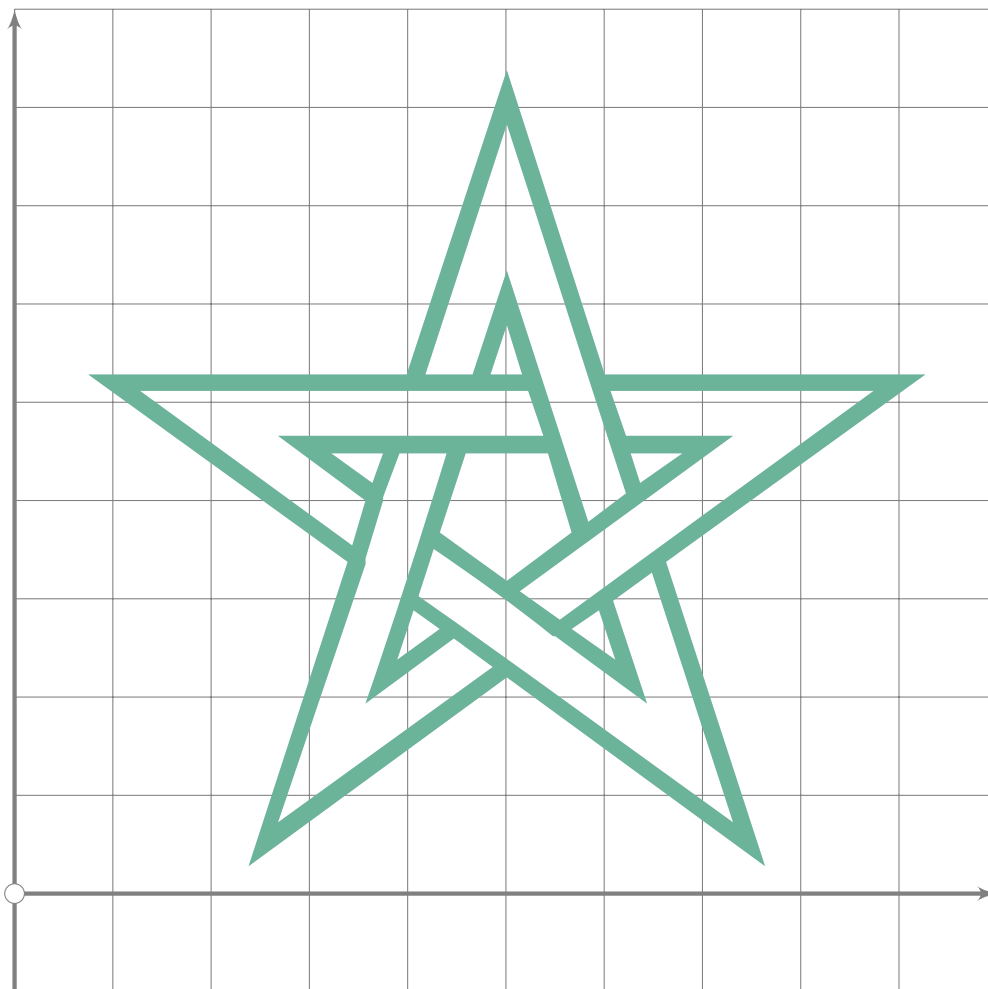
```

139
140 begintsuglyph("pentagram12",12);
141   begingroup
142     save lp;
143     path lp[];
144     my_path:=penta_ell(0.10,0);
145     lp1:=subpath (5,7) of my_path;
146     lp2:=subpath (2,4) of my_path;
147     default_nib:=fix_nib(14,14,0);
148     pen_stroke(tip(1)(1))(lp1 scaled 425 shifted centre_pt)(lp3);
149     pen_stroke(tip(1)(1))(lp2 scaled 425 shifted centre_pt)(lp4);
150     lp3:=regenerate(lp3);
151     lp4:=regenerate(lp4);
152     for i:=0 upto 4:
153       dangerousFill lp3 rotatedaround (centre_pt,i*72);
154       dangerousFill lp4 rotatedaround (centre_pt,i*72);
155     endfor;
156   endgroup;
157 endtsuglyph;

```

U+F5C0D
bll.pentagram13

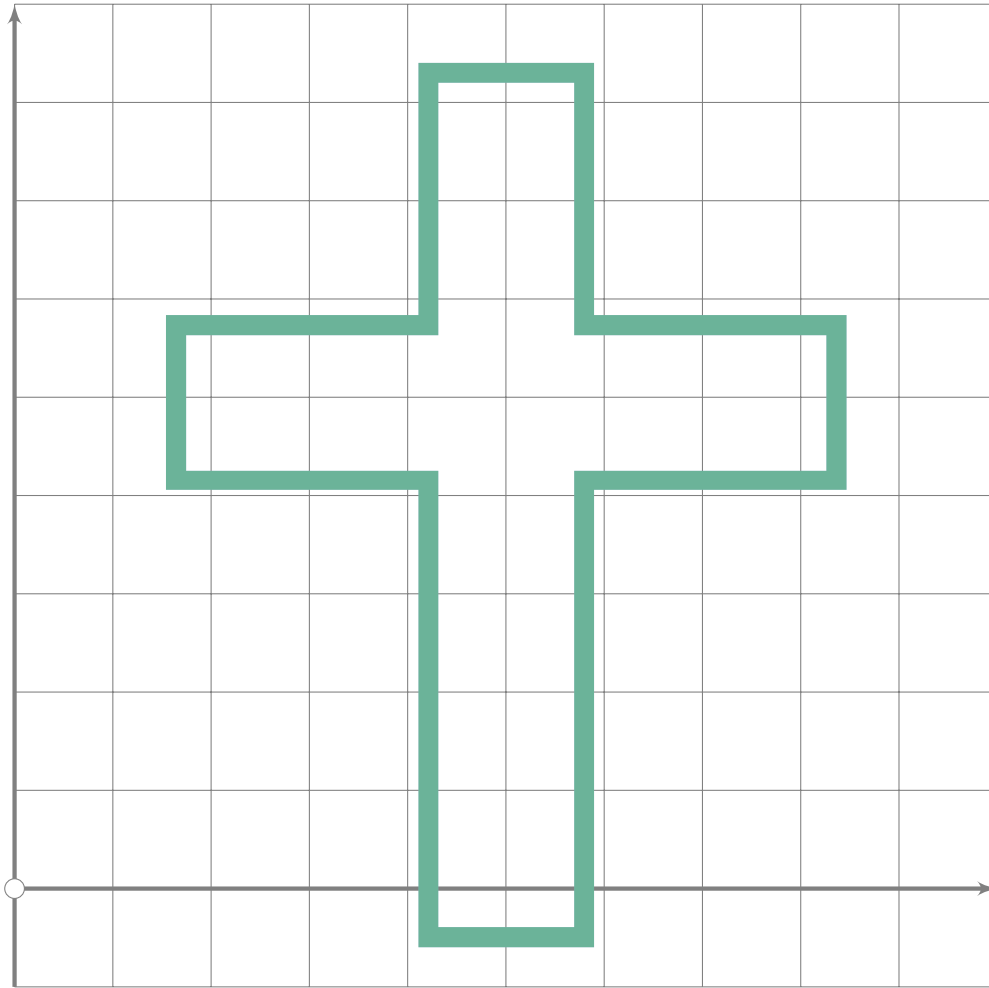
F5C



```

158
159 begintsuglyph("pentagram13",13);
160   begingroup
161     save lp;
162     path lp[];
163     my_path:=penta_ell(0.15,0);
164     lp1:=subpath (5,7) of my_path;
165     lp2:=subpath (2,4) of my_path;
166     default_nib:=fix_nib(17,17,0);
167     pen_stroke(tip(1)(1))(lp1 scaled 420 shifted centre_pt)(lp3);
168     pen_stroke(tip(1)(1))(lp2 scaled 420 shifted centre_pt)(lp4);
169     lp3:=regenerate(lp3);
170     lp4:=regenerate(lp4);
171     for i:=0 upto 4:
172       dangerousFill lp3 rotatedaround (centre_pt,i*72);
173       dangerousFill lp4 rotatedaround (centre_pt,i*72);
174     endfor;
175   endgroup;
176 endtsuglyph;
177
178

```

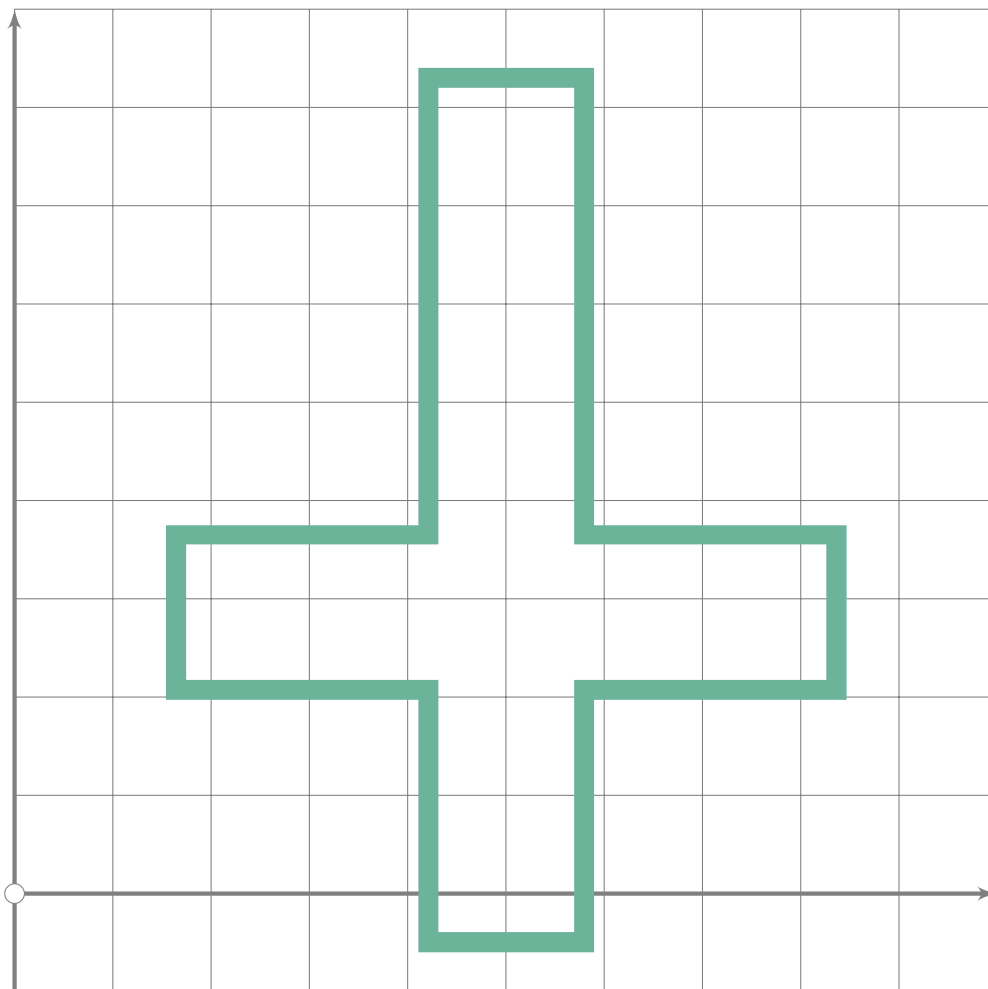


F5C

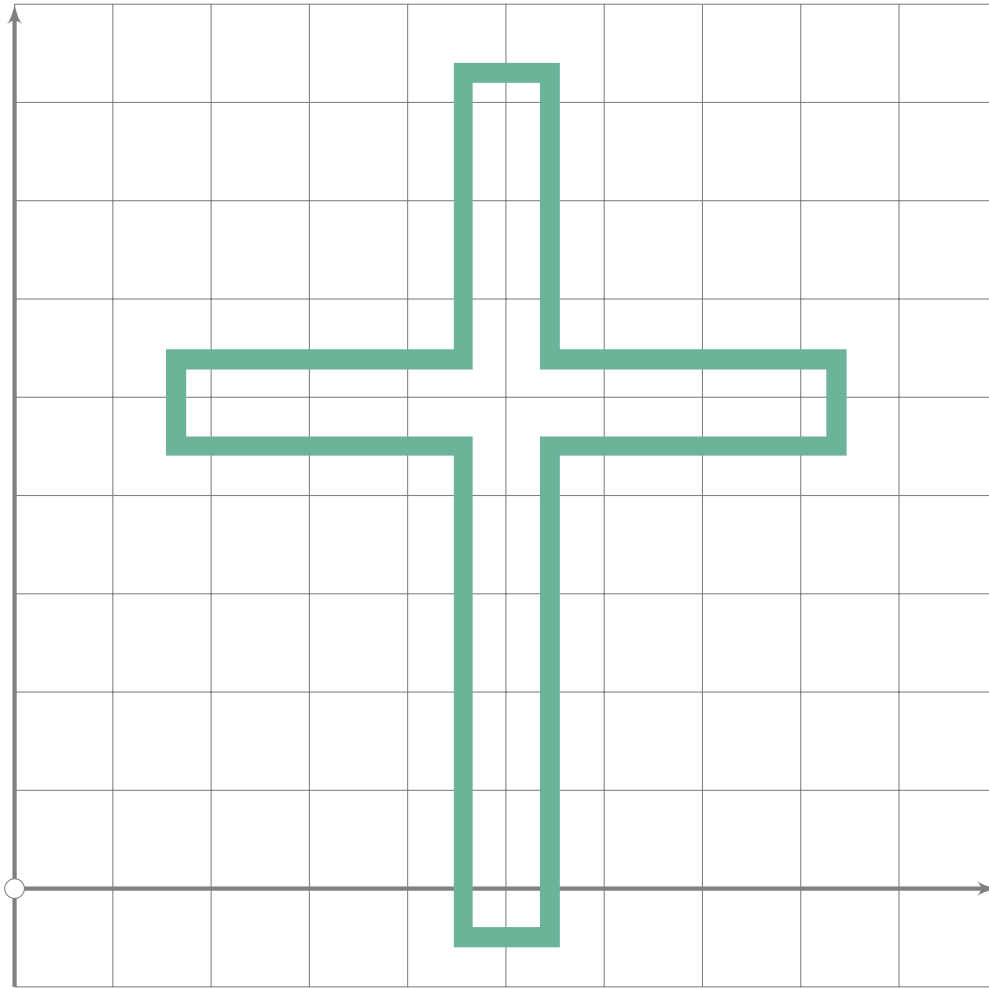
```
179
180 begint Suglyph("cross01",129);
181 draw_stroked(20,1)(cross_path(0.09) scaled 880 shifted centre_pt);
182 endtsuglyph;
```

U+F5C82
bll.cross02

F5C



```
183
184 begintsuglyph("cross02",130);
185   draw_stroked(20,1)(cross_path(0.09)
186     rotated 180 scaled 880 shifted centre_pt);
187 endsuglyph;
```

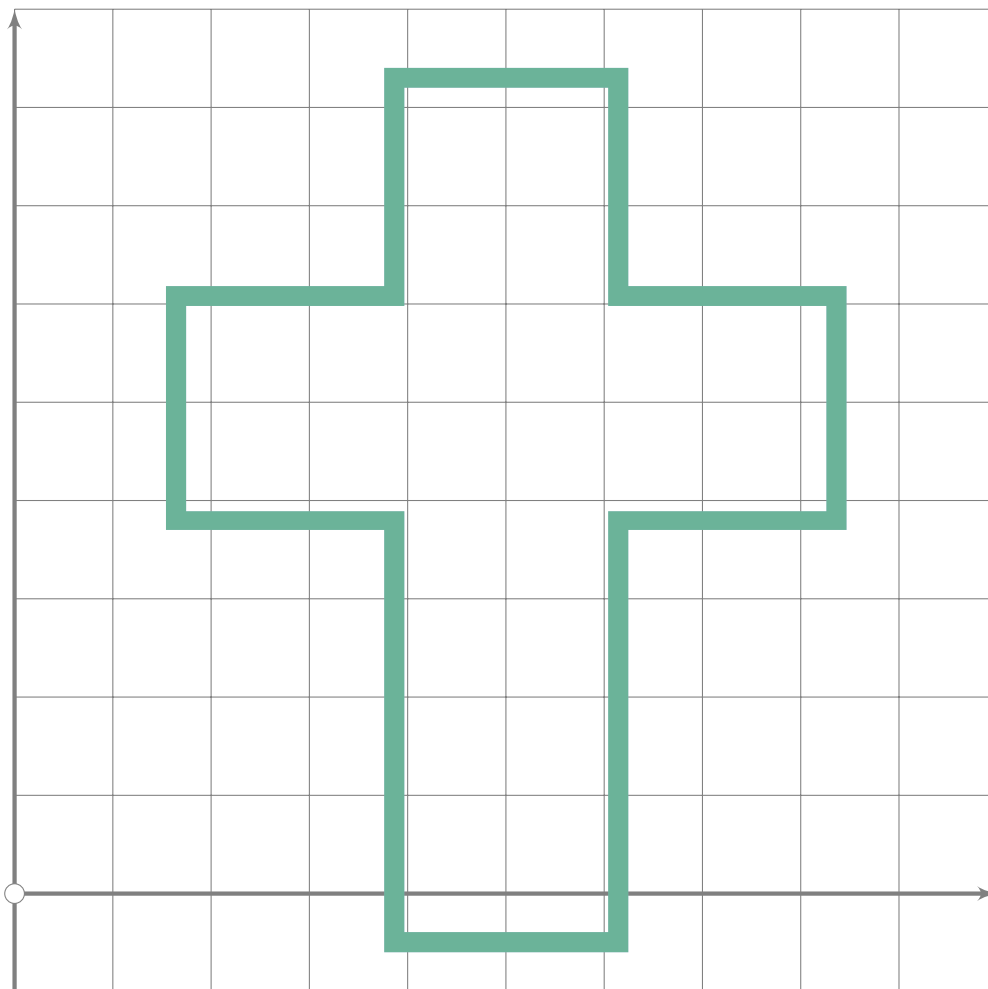


F5C

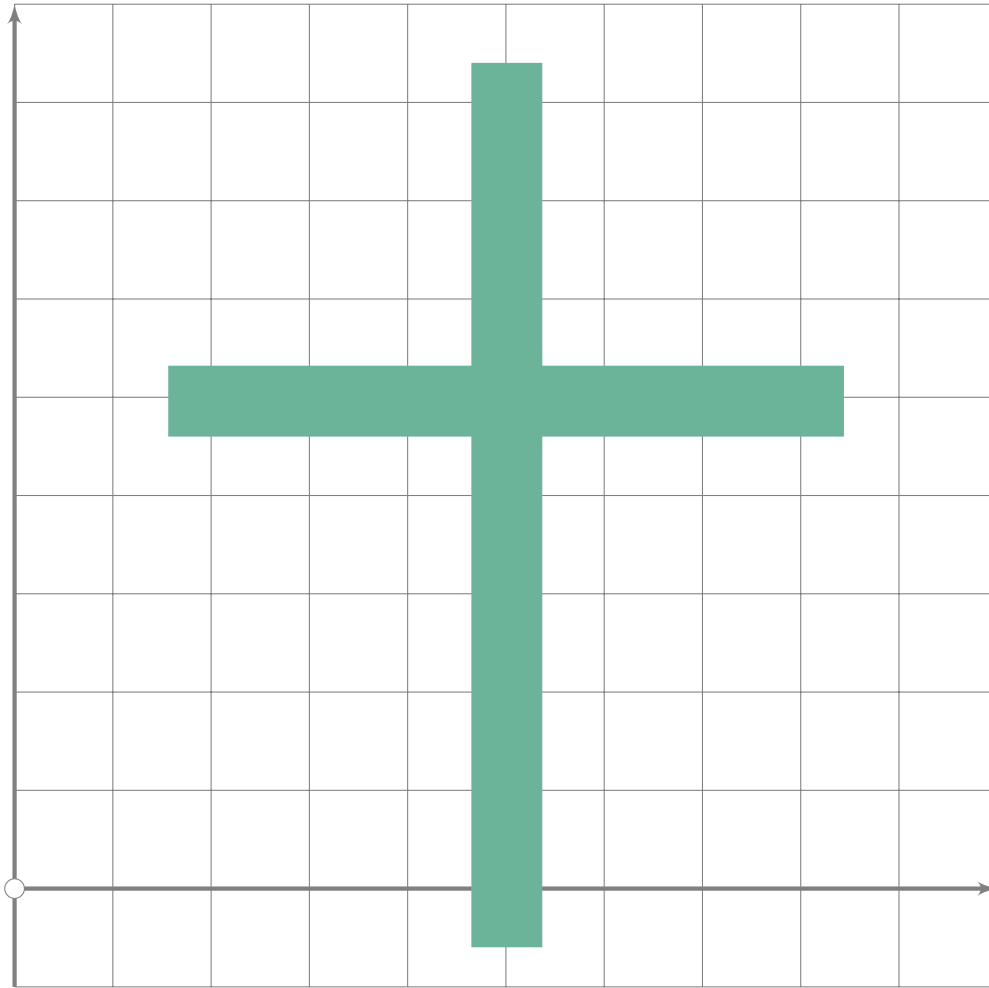
```
188
189 begintsuglyph("cross03",131);
190 draw_stroked(20,1)(cross_path(0.05) scaled 880 shifted centre_pt);
191 endsuglyph;
```

U+F5C84
bll.cross04

F5C



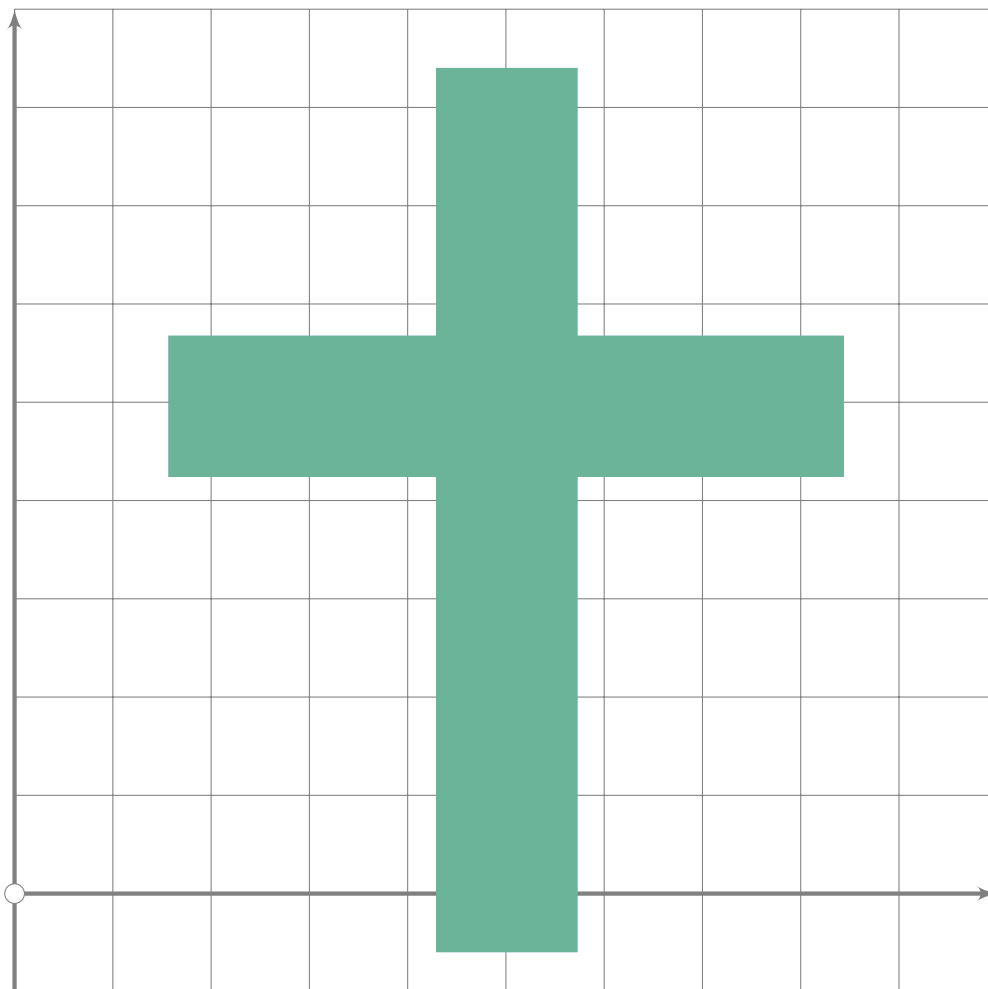
```
192  
193 begintsuglyph("cross04";132);  
194 draw_stroked(20,1)(cross_path(0.13) scaled 880 shifted centre_pt);  
195 endsuglyph;
```

F5C

```
196  
197 beginsuglyph("cross05",133);  
198   dangerousFill cross_path(0.04) scaled 900 shifted centre_pt;  
199 endsuglyph;
```

U+F5C86
bll.cross06



F5C

```
200
201 begintsuglyph("cross06",134);
202   dangerousFill cross_path(0.08) scaled 900 shifted centre_pt;
203 endsuglyph;
204
205 _____
206
207 endfont;
208
209 _____
```