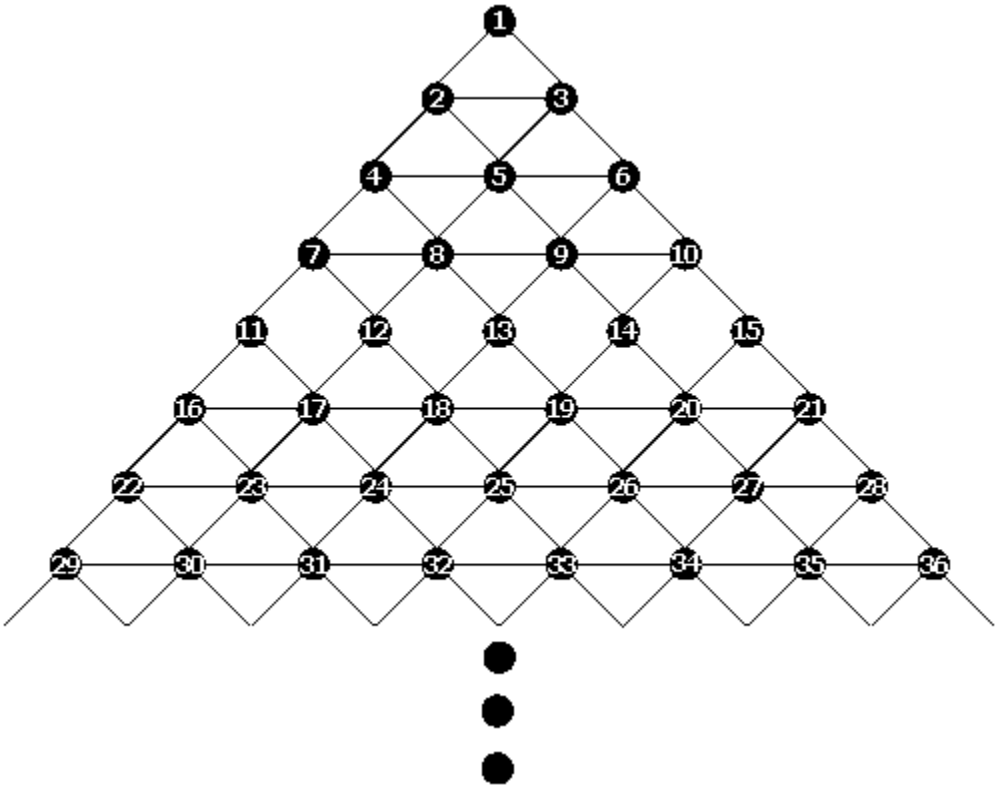
Appendix D - HHC 2008 Programming Contest

Gene Wright

Consider the points on a grid of equilateral triangles as shown below. Note that if the points are numbered from left to right and top to bottom, then groups of these points form the vertices of certain geometric shapes. For example, the sets of points 1, 2 ,3 and 7, 9, 18 are the vertices of triangles, the sets 11,13,24,26 and 2,7,9,18 are the vertices of parallelograms, and the sets 4, 5, 9, 13, 12, 7 and 8, 10, 21, 34, 32, 17 are the vertices of hexagons.



Write a program named A which will accept a set of points on this triangular grid, analyze it and determine whether the points are vertices of a triangle, parallelogram, hexagon or an illegal figure. In order for a figure to be acceptable, it must meet two conditions:

1) Each side of the figure must coincide with an edge in the grid, and

2) All sides of the figure must be of the same length.

INPUT: The input will consist of a series of point sets. Each point set will have at most six points in a set. The points in the set are limited to the range of 1 through 105.

35s (or other RPN model): Input will be done one point at a time. Each point will be keyed and R/S will be pressed. A -1 will be entered when all data points are entered and R/S will be pressed. The input for the data set { 1 2 3 } would be SHIFT CLEAR VARS then 1 STO A 2 STO B 3 STO C 1 CHS STO D. Variables A through G might be used in this manner. The last register used (in order) would contain the -1 value. Running the program will be done after storing the inputs by XEQ A ENTER.

50g (or other RPL model): The stack will be clear except for a list containing the data points. The input for the sample data set { 1 2 3 } would be { 1 2 3 } placed on a clear stack. Running the program will be done by pressing: VAR then the menu label A. USER RPL only. No unsupported entry points, System RPL, etc.

OUTPUT:

35s (or other RPN model): Display a 0 if the set is an invalid figure. Display a 1 if the set is a triangle. Display a 2 if the set is a parallelogram. Display a 3 if the set is a hexagon.

50g (or other RPL model): Display "ERROR", "TRIANGLE", "PARALLELOGRAM", or "HEXAGON", appropriately.

TEST: Fastest total time to evaluate a set of input test cases. Decision of the judge is ABSOLUTELY final.

==========

The contest generated only a handful of entries, perhaps because of the apparent complexity of the problem or perhaps because of the full schedule of the conference.

There were two classes of machines eligible: RPL (any) and RPN (any). Only four entries were received, two in each category.

Only one machine (a 50g) correctly solved all input problems. That was the winner of course. Allen Thomson. He correctly solved all 7 input cases in about 12 seconds. Timing wasn't as critical, since his program was the only one that worked in all cases.

One note: The hexagons in the problem are to be regular hexagons - no internal pointing sides.

Perhaps most interestingly, when this problem was posted to the HP Museum forum after the contest, quite a bit of interest ensued. Several of these were posted to the forum at this link:

http://www.hpmuseum.org/cgi-sys/cgiwrap/hpmuseum/archv018.cgi?read=141243

A couple of the more interesting programs generated there are reproduced below. Both of these are by Egan Ford.

Fast RPL solution #1

```
HHC 2008 Programming Contest (and a tip for future UserRPL optimization)
Message #40 Posted on the HP Museum websit3 by Egan Ford on 12 Oct 2008, 12:43 p.m.,

I got the time down to 1.25 seconds for all 7 problems. An incredible 2x+ increase in
performance. And that is Gjermund's UserRPL tip: Floats are faster than Ints.

Thanks Gjermund!

The changes:

   1. Added a . (dot) after each integer.
   2. Converted the input list to floats (I->R after SORT).

%%HP: T(3)A(R)F(.);
\<< SORT I\->R \-> p
  \<< "ERROR" p SIZE
    CASE DUP 3. ==
      THEN DROP p OBJ\-> \-> a b c r
        \<<
          b 8. * 1. + \v/ 1. - 2. / CEIL 'r' STO
          c b - b a - >
          IF
          THEN
```

```
      a r r 1. - * 2. / >
      IF
      THEN
        b a - DUP r 2. * + 1. - * 2. / b + c ==
        IF
        THEN DROP "TRIANGLE DOWN"
        END
      END
    ELSE
      c r r 1. + * 2. / \<=
      IF
      THEN
        b c b - DUP NEG r 2. * + 1. - * 2. / - a ==
        IF
        THEN DROP "TRIANGLE UP"
        END
      END
    END
  \>>
END DUP 4 ==
THEN DROP p OBJ\-> \-> a b c d r
  \<<
    c 8. * 1. + \v/ 1. - 2. / CEIL 'r' STO
    b r r 1. - * 2. / >
    IF
    THEN
      c b - DUP r 2. * + 1. - * 2. / c + d ==
      IF
      THEN b c b - DUP NEG r 2. * + 1. - * 2. / - a ==
        IF
        THEN DROP "PARALLELOGRAM DIAMOND"
        END
      END
    ELSE d c - b a - ==
      IF
      THEN d r r 1. + * 2. / \<=
        IF
        THEN
          b 8. * 1. + \v/ 1. - 2. / CEIL 'r' STO
          a r r 1. - * 2. / >
          IF
          THEN
            b a - DUP r 2. * + 1. - * 2. / b + d ==
            IF
            THEN DROP "PARALLELOGRAM LEFT"
            ELSE
              b a - DUP r 1. + 2. * + 1. - * 2. / b + d ==
              IF
              THEN DROP "PARALLELOGRAM RIGHT"
              END
            END
          END
        END
      END
    END
  \>>
END 6 ==
THEN p OBJ\-> \-> a b c d e f r
  \<< f e - b a - ==
    IF
    THEN f e - 2. * d c - ==
      IF
      THEN
        f 8. * 1. + \v/ 1. - 2. / CEIL 'r' STO
```

```
            e r r 1. - * 2. / >
            IF
            THEN
              b 8. * 1. + \v/ 1. - 2. / CEIL 'r' STO
              a r r 1. - * 2. / >
              IF
              THEN
                b a - DUP r 2. * + 1. - * 2. / a + c ==
                IF
                THEN
                  r b a - + 'r' STO
                  b a - DUP r 2. * + 1. - * 2. / d + f ==
                  IF
                  THEN DROP "HEXAGON"
                  END
                END
              END
            END
          END
        END
      \>>
    END
  END
\>>
\>>
```

Fast RPN solution. About 44 seconds on an HP 41CX.
HHC 2008 Programming Contest -- anyone want to try an RPN solution? Message #22 Posted on HP Museum website by Egan Ford on 4 Oct 2008, 3:55 a.m.,

```
01 LBL "SETUP"        134 *                 267 1                 400 RCL 02
02 0                  135 2                 268 +                 401 8
03 SETSW              136 /                 269 SQRT              402 *
04 CLRG               137 RCL 01            270 1                 403 1
05 11                 138 X<=Y?             271 -                 404 +
06 STO 09             139 GTO 20            272 2                 405 SQRT
07 0                  140 RCL 02            273 /                 406 1
08 RTN                141 RCL 01            274 STO 07            407 -
09 LBL "FS"           142 -                 275 FRC               408 2
10 FIX 00             143 ENTER             276 X=0?              409 /
11 CF 29              144 ENTER             277 GTO 09            410 STO 07
12 CLA                145 RCL 07            278 RCL 07            411 FRC
13 >"NUM PTS?"        146 2                 279 1                 412 X=0?
14 PROMPT             147 *                 280 +                 413 GTO 13
15 INT                148 +                 281 INT               414 RCL 07
16 STO 00             149 1                 282 STO 07            415 1
17 1000               150 -                 283 LBL 09            416 +
18 /                  151 *                 284 RCL 07            417 INT
19 1                  152 2                 285 1                 418 STO 07
20 +                  153 /                 286 -                 419 LBL 13
21 STO 10             154 RCL 02            287 RCL 07            420 RCL 07
22 LBL 00             155 +                 288 *                 421 1
23 CLA                156 RCL 03            289 2                 422 -
24 >"PT "             157 X#Y?              290 /                 423 RCL 07
25 ARCL 10            158 GTO 20            291 RCL 01            424 *
26 >"?"               159 CLA               292 X<=Y?             425 2
27 PROMPT             160 >"TRIANGLE DOWN”  293 GTO 20            426 /
28 INT                161 2                 294 RCL 02            427 RCL 01
29 STO IND 10         162 STO 08            295 RCL 01            428 X<=Y?
30 ISG 10             163 GTO 20            296 -                 429 GTO 20
31 GTO 00             164 LBL 06            297 ENTER             430 RCL 06
32 RUNSW              165 0                 298 ENTER             431 RCL 05
33 FIX 04             166 4                 299 RCL 07            432 -
34 RCL 00             167 X#NN?             300 2                 433 ENTER
```

```
35 1000
36 /
37 1
38 +
39 SIGN
40 LBL 01
41 LASTX
42 LASTX
43 RCL IND L
44 LBL 02
45 X<=NN?
46 GTO 03
47 X<>Y
48 STO Y
49 RCL IND X
50 LBL 03
51 ISG Y
52 GTO 02
53 X<> IND L
54 STO IND Z
55 ISG L
56 GTO 01
57 CLA
58 >"ERROR"
59 0
60 STO 08
61 0
62 3
63 X#NN?
64 GTO 06
65 RCL 02
66 8
67 *
68 1
69 +
70 SQRT
71 1
72 -
73 2
74 /
75 STO 07
76 FRC
77 X=0?
78 GTO 04
79 RCL 07
80 1
81 +
82 INT
83 STO 07
84 LBL 04
85 RCL 02
86 RCL 01
87 -
88 RCL 03
89 RCL 02
90 -
91 X>Y?
92 GTO 05
93 RCL 07
94 1
95 +
96 RCL 07
97 *
98 2
168 GTO 11
169 RCL 03
170 8
171 *
172 1
173 +
174 SQRT
175 1
176 -
177 2
178 /
179 STO 07
180 FRC
181 X=0?
182 GTO 07
183 RCL 07
184 1
185 +
186 INT
187 STO 07
188 LBL 07
189 RCL 07
190 1
191 -
192 RCL 07
193 *
194 2
195 /
196 RCL 02
197 X<=Y?
198 GTO 08
199 RCL 03
200 RCL 02
201 -
202 ENTER
203 ENTER
204 RCL 07
205 2
206 *
207 +
208 1
209 -
210 *
211 2
212 /
213 RCL 03
214 +
215 RCL 04
216 X#Y?
217 GTO 20
218 RCL 03
219 RCL 02
220 -
221 ENTER
222 ENTER
223 CHS
224 RCL 07
225 2
226 *
227 +
228 1
229 -
230 *
231 2
301 *
302 +
303 1
304 -
305 *
306 2
307 /
308 RCL 02
309 +
310 RCL 04
311 X#Y?
312 GTO 10
313 CLA
314 >"PARALLELOGRAM"
315 >" LEFT"
316 4
317 STO 08
318 GTO 20
319 LBL 10
320 1
321 ST+ 07
322 RCL 02
323 RCL 01
324 -
325 ENTER
326 ENTER
327 RCL 07
328 2
329 *
330 +
331 1
332 -
333 *
334 2
335 /
336 RCL 02
337 +
338 RCL 04
339 X#Y?
340 GTO 20
341 CLA
342 >"PARALLELOGRAM"
343 >" RIGHT"
344 5
345 STO 08
346 GTO 20
347 LBL 11
348 0
349 6
350 X#NN?
351 GTO 20
352 RCL 06
353 RCL 05
354 -
355 RCL 02
356 RCL 01
357 -
358 X#Y?
359 GTO 20
360 RCL 06
361 RCL 05
362 -
363 2
364 *
434 ENTER
435 RCL 07
436 2
437 *
438 +
439 1
440 -
441 *
442 2
443 /
444 RCL 01
445 +
446 RCL 03
447 X#Y?
448 GTO 20
449 RCL 06
450 RCL 05
451 -
452 ST+ 07
453 ENTER
454 ENTER
455 RCL 07
456 2
457 *
458 +
459 1
460 -
461 *
462 2
463 /
464 RCL 04
465 +
466 RCL 06
467 X#Y?
468 GTO 20
469 CLA
470 >"HEXAGON"
471 6
472 STO 08
473 LBL 20
474 STOPSW
475 RCL 08
476 STO IND 09
477 1
478 ST+ 09
479 AVIEW
480 RTN
481 LBL "PRINT"
482 RCL 09
483 1
484 -
485 1000
486 /
487 11
488 +
489 STO 10
490 LBL 21
491 CLA
492 RCL IND 10
493 30
494 +
495 INT
496 GTO IND X
497 LBL 30
```

```
99 /
100 RCL 03
101 X>Y?
102 GTO 20
103 RCL 03
104 RCL 02
105 -
106 ENTER
107 ENTER
108 CHS
109 RCL 07
110 2
111 *
112 +
113 1
114 -
115 *
116 2
117 /
118 RCL 02
119 X<>Y
120 -
121 RCL 01
122 X#Y?
123 GTO 20
124 CLA
125 >"TRIANGLE UP"
126 1
127 STO 08
128 GTO 20
129 LBL 05
130 RCL 07
131 1
132 -
133 RCL 07
232 /
233 RCL 02
234 X<>Y
235 -
236 RCL 01
237 X#Y?
238 GTO 20
239 CLA
240 >"PARALLELOGRAM"
241 >" DIAMOND"
242 3
243 STO 08
244 GTO 20
245 LBL 08
246 RCL 04
247 RCL 03
248 -
249 RCL 02
250 RCL 01
251 -
252 X#Y?
253 GTO 20
254 RCL 07
255 1
256 +
257 RCL 07
258 *
259 2
260 /
261 RCL 04
262 X>Y?
263 GTO 20
264 RCL 02
265 8
266 *
365 RCL 04
366 RCL 03
367 -
368 X#Y?
369 GTO 20
370 RCL 06
371 8
372 *
373 1
374 +
375 SQRT
376 1
377 -
378 2
379 /
380 STO 07
381 FRC
382 X=0?
383 GTO 12
384 RCL 07
385 1
386 +
387 INT
388 STO 07
389 LBL 12
390 RCL 07
391 1
392 -
393 RCL 07
394 *
395 2
396 /
397 RCL 05
398 X<=Y?
399 GTO 20
498 >"ERROR"
499 GTO 40
500 LBL 31
501 >"TRIANGLE UP"
502 GTO 40
503 LBL 32
504 >"TRIANGLE DOWN"
505 GTO 40
506 LBL 33
507 >"PARALLELOGRAM"
508 >" DIAMOND"
509 GTO 40
510 LBL 34
511 >"PARALLELOGRAM"
512 >" LEFT"
513 GTO 40
514 LBL 35
515 >"PARALLELOGRAM"
516 >" RIGHT"
517 GTO 40
518 LBL 36
519 >"HEXAGON"
520 LBL 40
521 AVIEW
522 ISG 10
523 GTO 21
524 FIX 06
525 RCLSW
526 CLA
527 >"TIME: "
528 ATIME24
529 AVIEW
530 RTN
531 END
```