

# HHC 2011 RPN Programming Contest

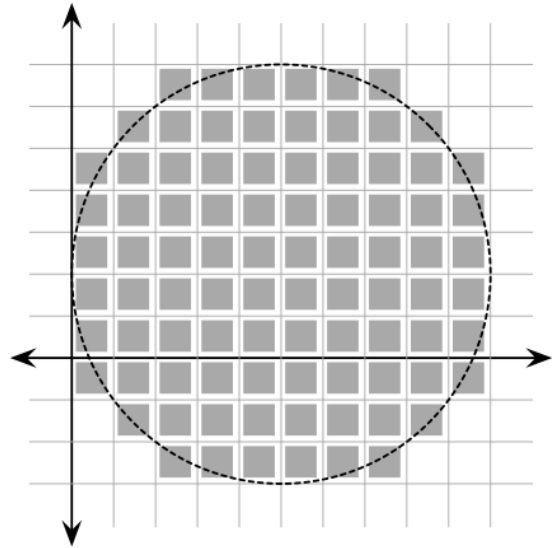
September 24-25 / 2011 San Diego

**Problem Description:** Did you know that if you draw a circle that fills the screen on your 1080p high definition display, almost a million pixels are lit? That's a lot of pixels! But do you know exactly how many pixels are lit? Let's find out!

Assume the display is set on a Cartesian grid where every pixel is a perfect unit square. For example, one pixel occupies the area of a square with corners (0,0) and (1,1). A circle can be drawn by specifying its center in grid coordinates and its radius. A pixel on the display is lit if any part of it is covered by the circle; pixels whose edge or corner are just touched by the circle, however, are not lit.

You must compute the exact number of pixels "lit" when a circle with a given position and radius is drawn.

**Input:** Each test case consists of three integers,  $x$ ,  $y$ , and  $r$  ( $1 \leq x, y, r \leq 5000$ ), specifying respectively the center  $(x, y)$  and radius of the circle drawn. The radius will be loaded into stack register Z, the  $y$  coordinate of the center of the circle into stack register Y, and the  $x$  coordinate of the circle into stack register X. Assume successive program runs are to be started by simply entering new values and pressing R/S. Assume that all circles fit on the display panel even if in reality they would not.



**Output:** Return the number of pixels that are lit when the specified circle is drawn.

**Sample Cases:** (A) Input of 1 ENTER 1 ENTER 1 R/S should return 4. This represents a circle with a center of (1,1) and a radius of 1. The display would have 4 pixels "on" to represent this circle. (B) Input of 5 ENTER 2 ENTER 5 R/S should return 88. This represents a circle with a center of (5,2) and a radius of 5. The display would have 88 pixels "on" to represent this circle. This is the circle shown in the figure above. 88 pixels are "on" in this picture.

**Machines Eligible:** This contest is open to any and all RPN machines: 15c, 15c+, 15c LE, 34S, 41CL, 42S, 67, 65, etc. RPL users are welcome to try the problem, but this is for RPN machines only.

## Rules: (aka the fine print)

- 1) The decision of the judge is FINAL. No appeals are allowed to anyone or anything.
- 2) The purpose of this contest is to have fun and learn.
- 3) At least two contestants must submit an entry.
- 4) No custom built ROM or machine code can be built and used for this problem. Any already existing functionality in the machine is ok.
- 5) You must submit a machine with your program already keyed in to the judge AS WELL as a legible listing of your program with your name on the listing AND the machine. Machines with no names that are given to the judge are assumed to be gifts to the judge. Thank you!
- 6) Submission must be made by the end of the contest (Time is TBA).
- 7) Assume the program will start running with step 001 and/or a R/S.
- 8) By submitting a program, you agree to allow it to be shared with the community.
- 9) This is a contest between individuals, not teams. One submittal <> one person.
- 10) You may not access the internet for any help in any fashion. Do not cheat in any way. Do not check the HP Museum Forum either.
- 11) You must be present to win.
- 12) If a point is unclear, ask immediately. No excuses for ignorance. Clarifications will be shared with the entire group during the conference.
- 13) Assume default machine settings. Your program must stop with the default settings in place.
- 14) Winner will be the program with the fastest times over the test cases giving correct results. If in the judge's sole discretion, two entries are "about the same speed," the winner will be the shortest routine. In case of a tie, the most elegant solution (according to the judge) wins.
- 15) The purpose of this contest is to learn and have fun. Happy Programming.