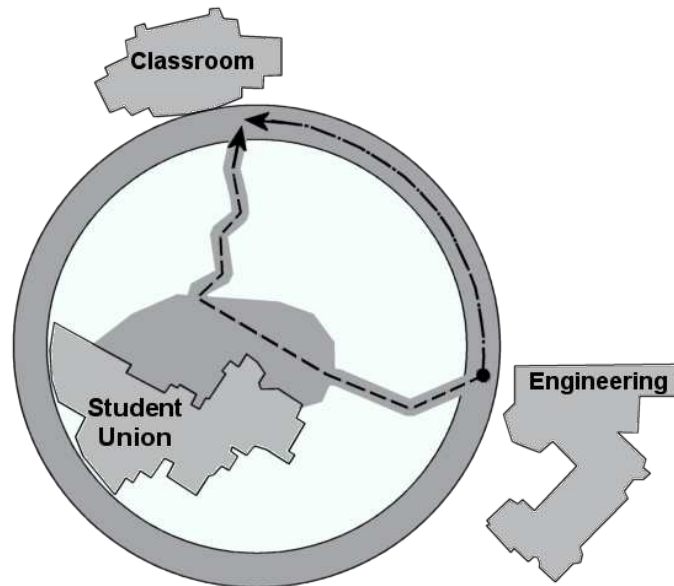


UCF “Practice” Local Contest — Aug 27, 2016

Pegasus Circle Shortcut

filename: shortcut
(*Difficulty Level:* Hard)

For the UCF High School Programming Tournament, the judges were located in the Engineering building, and most of the teams were in the Classroom building, which is on the other side of Pegasus Circle.



Chris was walking to the Classroom building for the first time, and was joined by Jeremy, who had made the hike a couple of times already.

“Jeremy, is it faster to stay on the circle, or to cut through the middle using the boardwalks that go to the Student Union?” asked Chris.

“I don’t know.” Jeremy answered. “I think it’s about the same, but it might be slightly faster to use the walkways.”

“Well, if it’s about the same, let’s stick to the circle. I don’t want to be attacked by squirrels.”*

The Problem:

Given two points on a circle, and two paths to get from one to the other—one following the perimeter of the circle, and the other by a sequence of connected straight line segments through the interior of the circle—determine the shorter of the two paths.

* No guarantee is made that these are exact quotes. Also, Chris is not really afraid of squirrels.

The Input:

The input will contain multiple test cases, each consisting of two lines. The first line of each test case contains six floating-point numbers: x_c , y_c , x_s , y_s , x_f , and y_f , where (x_c, y_c) is the center point of the circle, (x_s, y_s) is the start point for both paths (e.g., the Engineering building), and (x_f, y_f) is the finish point for both paths (e.g., the Classroom building). The circle will always have a radius greater than 1, and the start and finish points are both guaranteed to be at distinct points on its perimeter, with an accuracy of at least 3 places after the decimal. The path along the perimeter is always in the direction counter-clockwise around the circle.

The second line of each test case will start with an integer, n ($1 \leq n \leq 10$), followed by n pairs of floating-point numbers, $x_1, y_1, x_2, y_2, \dots, x_n, y_n$, where each pair (x_i, y_i) is a point inside the circle. The interior path traveled will be from point (x_s, y_s) to point (x_1, y_1) , then from (x_1, y_1) to (x_2, y_2) , then from (x_2, y_2) to (x_3, y_3) , ..., then from (x_n, y_n) to (x_f, y_f) .

The last test case will be followed by a line containing six zeros. All numbers on an input line will be separated from each other by one space, with no extra spaces at the beginning or end of lines. Assume that all the input floating point numbers will be less than 1000.0 and greater than -1000.0, with at most 6 places after the decimal.

The Output:

For each test case in the input, output a line in either the format

Case # n : Stick to the Circle.

if the perimeter path is shorter, or

Case # n : Watch out for squirrels!

if the interior path is shorter, where n is the number of the input test case, starting at 1.

Assume that the two paths will not be equal, i.e., it is guaranteed that the two distances will not be equal. In particular, assume that the two paths will differ in length by 0.001 or more.

Leave a blank line after the output for each test case.

Sample Input:

```
5.0 5.0 10.0 5.0 5.0 10.0
6 8.5 4.7 6.9 5.0 3.6 6.5 4.2 7.1 4.2 8.3 4.7 8.8
2.0 8.0 0.5 16.87412 7.5 0.8761
2 3.25 9.25 7.0 7.0
0 0 0 0 0 0
```

Sample Output:

Case #1: Stick to the Circle.

Case #2: Watch out for squirrels!