

Problem: Projector

Objective

Give practice with merge sort and quick sorting in C.

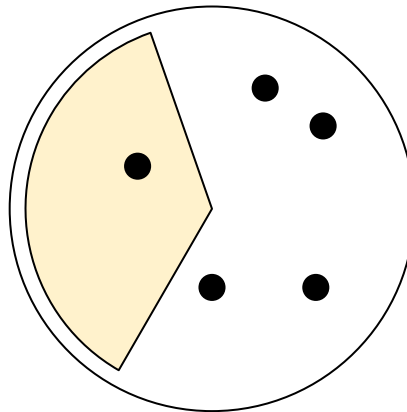
Practice with compareTo function strategy to compare objects in C while sorting

Practice with floating point comparison strategy in C

Story

Before starting to read the assignment, you can probably watch the class recording where I have briefly discussed about the requirements.

Your movie theater is trying out a new projector. The projection will be in the center of a circular room and will project a movie onto a part of the wall. The screen will project a sector of light at some angle width which comes at some default value, but can be updated based on a setting. The projector can be rotated such that the projection starts at any angle in the room.



However, you have several locations that groups of people have decided to stand at in the room. These groups of people are relatively small compared to the distance they are from the projector and can be treated as points with 0 radius. Although the projection won't be disturbed by our infinitely thin customers, the light emitted from the projector will be harmful to the people in the group.

You want to answer 2 main questions.

1. What is the largest possible angle of projection that can be used such that no group will be standing in the projection, thus will not harm anyone?
2. Find all the ways you can project this largest angle and show the list of pairs of groups that will be closest to the edge of those projections. It means, if there are multiple ways to get that largest angle, you need to list all the pairs of groups that are closest to the projection edge.

Problem

Given the locations and number of people in the groups, determine the largest angle that can be used such that no one is in the projection. Additionally, find all the ways you can project this largest angle and show the list of pairs of groups will be closest to the edge of the projection.

Input (no file i/o. Only standard input)

Input starts with a line containing 2 integers, **N** and **A**, ($1 \leq N \leq 500,000$; $1 \leq A \leq 359$) representing the number of groups standing in the room and the angle in degrees at which the screen projector initially projects (However, we will change this default setting).

The following N lines will each contain 3 space separated integers, **x**, **y**, and **s**, ($0 \leq |x|, |y| \leq 1,000,000$; $1 \leq s \leq 1,000$), representing x and y coordinates respectively of the location of a group and the number of people in that given group. Assume that the projector will be located at location (0, 0). Also, you can consider they are entered in a sequence of group numbers starting from 0 to ending at N-1. It means the first x, y, and s represent the data for group 0, the second one is group 1 and so on.

Note since the customers are so small multiple customers could stand at the same location.

Output

The output should consist of two items:

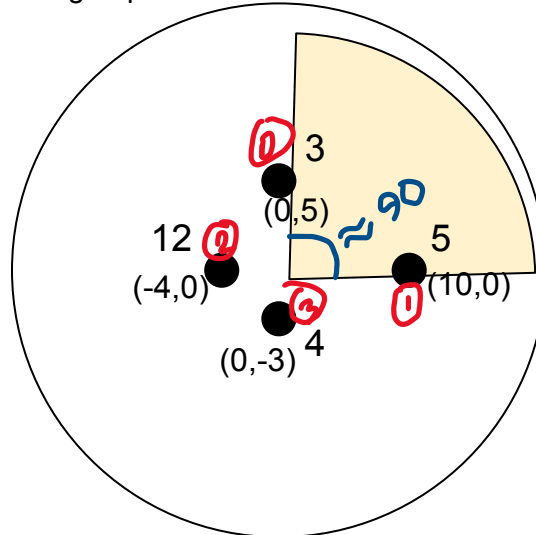
The first line will contain a floating-point value that specifies the maximum angle in degrees that can be used such that no person will be in the projection rounded to 4 digits after the decimal. Assuming, there are P different ways that maximum angle can be projected, print a list of P pairs, one per line containing the group numbers that will be closest to the projection edge. While printing a pair, the first group number must be smaller than the second group number. Also, the list of pairs must be printed as a sorted list. If multiple pairs starts with same group number, then it must be sorted based on the second group number in the pair.

Sample Input	Sample Output
4 91 0 5 3 10 0 5 -4 0 12 0 -3 4	Max projection degree without harming 90.0000 Closest possible group pairs in order: 0 1 0 2 1 3 2 3
3 181 1 1 8 -2 1 5 2 10 10	Max projection degree without harming 251.5651 Closest possible group pairs in order: 0 1

Explanation

Case 1

In the first test case there are 4 groups. The red colored numbers represent the group numbers.



You can use an angle slightly less than 90 degrees to project with no group in the projection. See the example image where these 90 degrees is achieved between group 0 and 1. You can get this 90 degrees in 4 different ways. Between group 0 and 2, between 2 and 3, and between 1 and 3. It is possible for this case as all the points are either in the x or y axis directly. Also, if you observe the output, it is in ordered based on the group number of the pair. If there is a tie in the first number of the pair, it is sorted base on the second number.

Example:

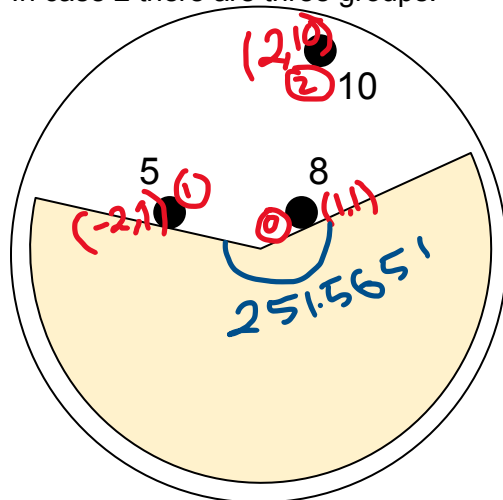
0 1

0 2

(as the first item is zero for both of them, it is ordered by the second item for those cases)

Case 2

In case 2 there are three groups.



The angle can be quite large without projecting on anybody. We can get this angle keeping group pair 0 and 1 close the edge.