Computer Science I Program 4: Projector Problems (Sorting)

Please Check Webcourses for the Due Date

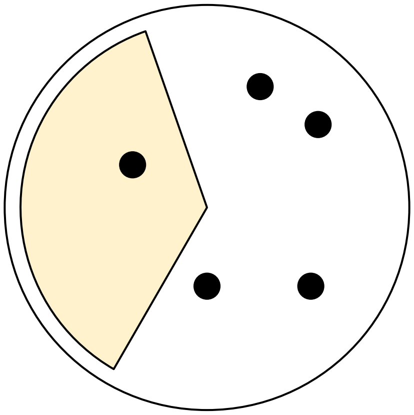
Please read the whole assignment before you start coding

Objective

Give practice with sorting in C.
Give practice with two pointer sweeps in C.

Background Story

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 does have some potential to damage organic entities.

You want to answer 2 main questions

1. What is the least number of people that will be in the projection assuming we choose the optimal projection location assuming the angle of projection is at the default setting?
2. What is the largest possible angle of projection that can be used such that no one is in the will stand in the projection?

Problem

Given the locations and number of people in the groups, find the least number of people that are required to be in some sector defined by a center and an angle. Additionally, determine the largest angle that can be used such that no one is in the projection.

Input

Input starts with a line containing 2 integers, N and A, ($1 \le N \le 500{,}000$; $1 \le A \le 359$) representing the number of groups of people standing in the room and the angle in degrees at which the screen projector initially projects.

The following N lines will each contain 3 space separated integers, x, y, and s, ($0 \le |x|, |y| \le 1{,}000{,}000$; $1 \le s \le 1{,}000$), 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).

All groups will be in distinct locations and no group will be located at (0, 0). However, it's possible that multiple groups may be on a ray that emanates from the origin. For example, it's possible that one group is located at (10, 6) while another group is located at (15, 9). We assume that all people in a group are located precisely at that point. (These are some very small people!!!)

Output

The output should consist of 2 lines. The first line will contain an integer representing the minimum number of people that would be in the projection. The second 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.

It is guaranteed that if the input angle were to change by ± 10^{-6} degrees, there would be no change to the output for the input case.

Sample Input	**Sample Output**

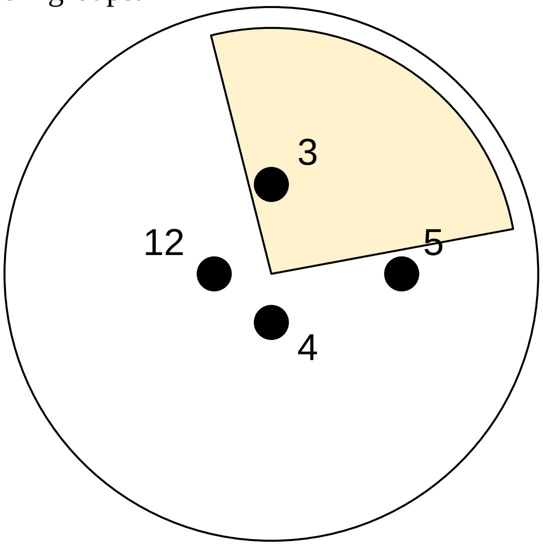
`4 91`
`0 5 3`
`10 0 5`
`-4 0 12`
`0 -3 4`
	`3`
	`90.0000`

`3 181`
`1 1 8`
`-2 1 5`
`2 10 10`
	`0`
	`251.5651`

Sample Explanation

Case 1

In the first test case there are 4 groups.

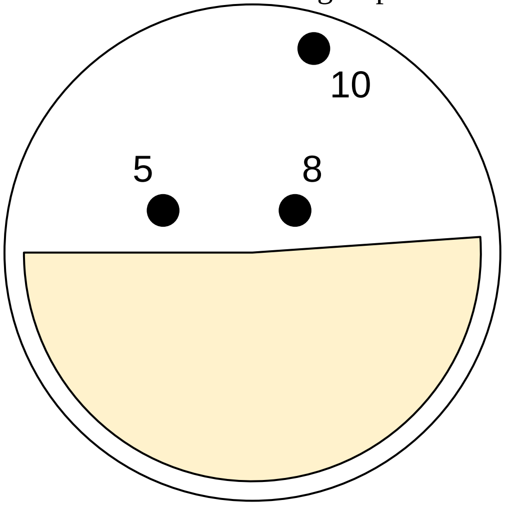


The 4 groups are laid out in a manner such that no matter what angle you project at least one of the groups will be in the projection. You can angle the projector so that only the group with 3 are in the cone.

You can use an angle slightly less than 90 degrees to project with no one in the projection. However, when rounded the answer becomes 90.

Case 2

In case 2 there are three groups and the projection is more than half the circle.



The bottom has no groups to project onto. Projecting in the negative y direction prevents projecting on anybody.

The angle can be quite large without projecting on anybody.

Hints

1. The relevant information about each group is the angle it forms with respect to the point (0, 0) and the number of people in the group. It probably makes sense to store this information in a struct.

2. To calculate the angle, **in radians**, formed by a point with respect to (0, 0), look up the atan2 function in the math library. This always returns an answer in between $-\pi$ and π.

3. Since all possible angles must be considered as a starting point, to make sure we include all possible ranges, the easiest solution is to include each point in a single array twice, once for the angle x and a second time for the angle $x + 2\pi$. This makes indicating the beginning and end of a range of "visible" groups easier.

Implementation Requirements/Run Time Requirements

1. Dynamic Memory allocation must be used to store any arrays of size ***N***, or larger, where ***N*** represents the number of groups for the input case.

2. You must use either Merge Sort of Quick Sort in your solution.

3. A two pointer sweep is recommended. This will be similar to the technique used in quiz 1 version b, question 2. For this problem, one index will represent the beginning angle where light is shone and the other index (higher one) will represent the end angle for the range where light is shone.

4. Your code must compile and execute on the Eustis system. The C compiler on this system is the one that the graders will be using to grade/evaluate your submissions.

Deliverables

1. Please submit a source file, `projector.c`, via Webcourses, for your solution to problem the problem.