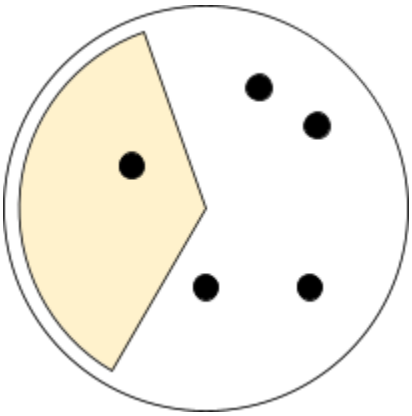
Projector Problems

Objective
Give practice with sorting in C.
Give practice with 2 pointer sweeps in C.

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 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).

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

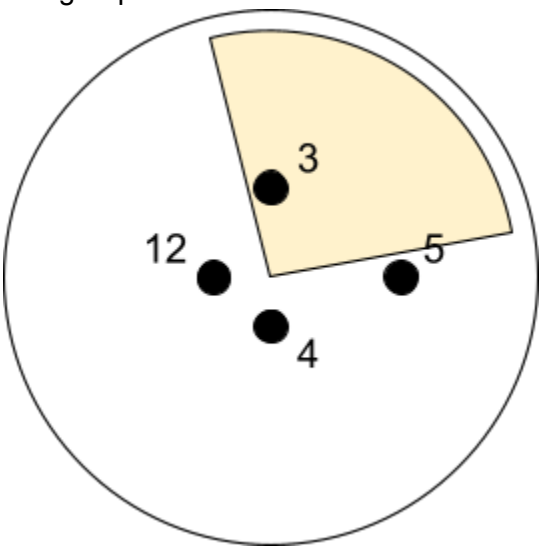
Output

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's guaranteed that the first answer would remain the change

Sample Input	**Sample Output**
4 91 0 5 3 10 0 5 -4 0 12 0 -3 4	90.0000 3
3 181 1 1 8 -2 1 5 2 10 10	251.5651 0

Explanation

Case 1
In the first test case there are 4 groups.

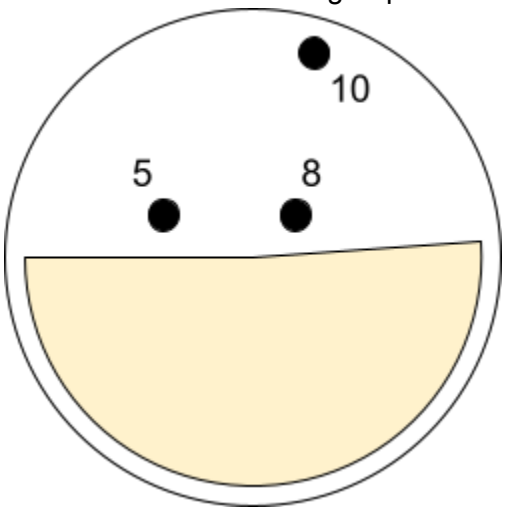


The 4 groups are laid out in a manner such that no matter what angle you project at least one of the groups will be in the projection. You can angle the projector so that only the group with 3 are in the cone.

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

Group Struct: The most critical information is the angle from the x axis the group lies at and the number of people in the group.

The struct would look something like the following.

```
struct Group {
   double angle;
   int size;
};
```

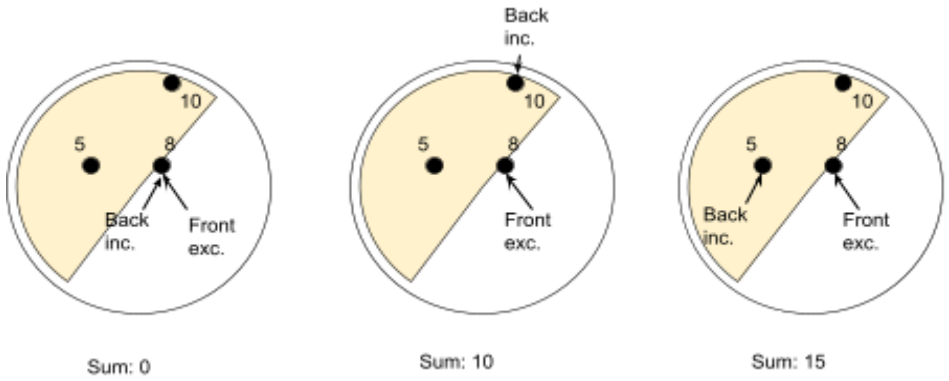
Sort: You should sort all the groups by the angle.

Trig: You can compute the angle most accurately using the atan2 function from the math header. Make sure to use -lm if you are compiling from the command line. See the command below as an example,

```
$ gcc my_code.c -o my_code.out -std=gnu11 -lm
```

Efficient Sort: You should use Merge Sort or Quick Sort to quickly sort the 500,000 values.

2 Pointers: You should use a 2 pointer sweep to efficiently find the minimum number of people in the cone. You can sweep through the groups in order of their angle with the first pointer. For each first pointer (exclusive) you can bump up the last pointer if the group should be in circle sector.



When the front is updated the sum should be decreased, and when the back is updated the sum should be increased.

I BETTER NOT FIND THIS ON THE INTERNET (OUTSIDE OF WEBCOURSES)!!!

Grading Criteria

- Good comments, whitespace, and variable names
  - 15 points
- No extra input output (e.g. input prompts, “Please enter the original sign’s message:”)
  - 5 points
- Standard IO
  - 5 points
- Use trigonometric functions to compute an angle
  - 10 points
- Sort the groups by their angle.
  - 5 points
- Use linear time sweeps to compute the answers
  - 5 points
- Compare only adjacent values to find the greatest angle that does not contain a group
  - 5 points
- Programs will be tested on 10 cases
  - 5 points each

No points will be awarded to programs that do not compile using “gcc -std=gnu11 -lm”.

Sometimes a requested technique will be given, and solutions without the requested technique will have their maximum points total reduced. For this problem use sorting. ***Without this programs will earn at most 50 points!***

Any case that causes a program to return a non-zero return code will be treated as wrong. Additionally, any case that takes longer than the maximum allowed time (the max of {5 times my solutions time, 10 seconds}) will also be treated as wrong.

No partial credit will be awarded for an incorrect case.