

Closest Apartments

locked

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

На Иванчо му писнало от условията в Студентски град и решил да се изнесе на квартира. Тъй като той е много претенциозен и не обича да пътува много, Иванчо искал апартаментът му да е сравнително близо до ФМИ. След като отворил сайт за квартири и му излезли цели N апартамента, той разбрал че няма как да разгледа всеки един по отделно и решил да си направи кратък списък с K апартамента, за които има време да разгледа. За да отговарят на неговите изисквания, той избрал тези K , които са най-близо до ФМИ.

Можем да си представим картата на София като равнина, където ФМИ е координатното начало - т.е с координати $(0, 0)$. Всеки от N -те апартамента също има две координати X и Y , които могат да са положителни или отрицателни. Разстоянието между две точки е стандартно Евклидово разстояние. Как се изчислява то - [тук](#).

Напишете програма, които намира тези K квартири, които са най-близки до ФМИ(центъра на координатната система).

Input Format

На първия ред се въвеждат две цели числа - N и K .

Следват N реда, всеки от които съдържа две цели числа - X и Y координатите на текущия апартамент.

Constraints

$1 \leq N \leq 5000000$

$K \leq N$

$1 \leq K \leq 100$

$-1000000 \leq X_i, Y_i \leq 1000000$

Output Format

Изкарайте K реда с по две числа - K -те най-близки апартамента до ФМИ. Изкарайте ги от най-близкия към най-далечния. Ако има няколко квартири на еднакво разстояние от центъра, изкарайте първо тези с по-малка X координата. Ако и X координатите им съвпадат - изкарайте първо тези с по-малка Y координата.

Sample Input 0

```
5 3
1 1
2 2
0 2
-1 -1
-3 2
```

Sample Output 0

```
-1 -1
1 1
0 2
```

Explanation 0

Имаме 5 апартамента, разстоянията до тях са съответно:

1) ≈ 1.41


2) ≈ 2.83

3) 2



4) ≈ 1.41

5) ≈ 3.61

Трите най-близки апартамента са съответно първият, третият и четвъртият. Изкарваме ги от най-близък към най-далечен като тъй като първият и четвъртият са еднакви близки, изкарваме първи този, с по-малка X координата.

Current Buffer (saved locally, editable)  

C++

```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
5 #include <algorithm>
6 #include <set>
7 #include <iterator>
8 #include<math.h>
9 using namespace std;
10 struct Point{
11     int x;
12     int y;
13     Point(int x,int y){
14         this->x=x;
15         this->y=y;
16     }
17     friend bool operator<(const Point& point1,const Point& point2);
18 }
19 };
20 bool operator <(const Point& point1,const Point& point2){
21     if(sqrt(pow(point1.x,2)+pow(point1.y,2))<sqrt(pow(point2.x,2)+pow(point2.y,2))){
22         return true;
23     }
24     else if(sqrt(pow(point1.x,2)+pow(point1.y,2))>sqrt(pow(point2.x,2)+pow(point2.y,2))){
25         return false;
26     }
27     else {
28         if(point1.x<point2.x){
29             return true;
30         }
31         else if(point1.x==point2.x){
32             return point1.y<point2.y;
33         }
34         else{
35             return false;
36         }
37     }
38 }
39 int main() {
40     int apartments;
41     int K;
42     cin>>apartments;
43     cin>>K;
44     int x;
45     int y;
46     set<Point> nearestPoints;
47     set<Point>:: iterator it;
48     for(int i=0;i<apartments;i++){
49         scanf("%d",&x);
50         scanf("%d",&y);
51         Point current(x,y);
52
53         nearestPoints.insert(current);
54     }
55     int counter=0;
56     for(it=nearestPoints.begin();it!=nearestPoints.end();it++){
57         printf("%d ", it->x);
58         printf("%d\n", it->y);
59
60         counter++;
61         if(counter==K){
62             break;
63         }
64     }
65
66     return 0;
67 }
68
69
```

Line: 1 Col: 1