Technische Universität München   Summer Term 2015
Fakultät für Informatik   Problem Set 9
Lehrstuhl für Effiziente Algorithmen (LEA)   25.06.2015
Prof. Dr. Ernst W. Mayr
Christian Müller, Philipp Hoffmann, Chris Pinkau, Stefan Toman

Algorithms for Programming Contests

This problem set is due by

Thursday, 02.07.2015, 6:00 a.m.

Try to solve all the problems and submit them at

<div align="center">https://judge.in.tum.de/</div>

This week's problems are:

The following amount of points will be awarded for solving the problems.

Problem	SS15N09A	SS15N09B	SS15N09C	SS15N09D	SS15N09E
Difficulty	easy	easy	medium	medium	hard
Points	4	4	6	6	8

If the judge does not accept your solution but you are sure you solved it correctly, use the "request clarification" option. In your request, include:

- the name of the problem (by selecting it in the subject field)

- a verbose description of your approach to solve the problem

- the time you submitted the solution we should judge

We will check your submission and award you half the points if there is only a minor flaw in your code.

If you have any questions please ask by using the judge's clarification form.

SS15N09A   Meteorite

Author: Christian Müller

Today, the LASER (**L**aboratory for **A**dvanced **S**cientific **E**mission of **R**ays) made a huge announcement - they discovered a new element. They even found out how to synthesize it: At first, you need a meteorite that is rapidly accelerating towards earth. Then you heat it up with a high-powered laser using a special focussing crystal (because lasers are totally awesome). This causes the meteorite to be rapidly condensed into one very small lump of the new element - aptly named "meteoritium".

However that process still leaves a small problem - a super dense lump of meteoritium rapidly falling through the earth's atmosphere. They now issued a safety warning to all people living close to the calculated impact site. To her excitement, Lea is among them.

She is now itching to know if there is a chance that the meteoritium will land on her parents' property (you can think of the property as a simple polygon with no intersecting edges and no holes) so she can be one of the first people on earth to see the new element.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case begins with a line consisting of 3 integers x_{impact}, y_{impact}, the coordinates of the calculated impact site and n, the number of sides that her parents' property has. n lines follow, each containing 4 integers x_1, y_1, x_2, y_2, describing a side of the polygon connecting the points (x_1, y_1) and (x_2, y_2).

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is "jackpot" if the impact site is contained in the given polygon and "too bad" otherwise. Each line of the output should end with a line break.

Constraints

- $1 \le t \le 20$

- $3 \le n \le 1000$

- $-1000 \le x_i, y_i \le 1000$

- Every coordinate of the polygon will have exactly 2 incident sides.

- The given polygon will always be a single, connected shape.

- (x_{impact}, y_{impact}) will never lie on a side or corner of the polygon.

Sample Data

Input

```
2
1 1 3
1 0 2 2
0 1 2 2
0 1 1 0

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

Output

```
Case #1: jackpot
Case #2: too bad
```

SS15N09B   Fence Posts

Author: Philipp Hoffmann

Lea has recently bought a new garden. The previous owner did not care very much for it, therefore it is quite a mess. The fencing is in an especially bad condition, fence posts are everywhere inside the garden and on unnecessary points of the boundary. Lea wants to fence in an area as big as possible using the existing fence posts and fences that run in straight lines between those posts. Help Lea by telling her which posts cannot be removed without altering the size of the garden.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with an integer n, the number of fence posts, n lines follow, The i-th line contains two integers x_i, y_i, the x- and y-coordinates of the i-th fence post.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is a space-separated, naturally ordered list of the indizes of those fence posts that cannot be removed without altering the size of the garden. Fence posts are indexed starting at 1. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$

- $3 \leq n \leq 10000$

- $0 \leq x_i, y_i \leq 1000$

- All points will be distinct.

- There will be at least three non-collinear points.

Sample Data

Input

```
 1  2
 2  4
 3  0 0
 4  1 1
 5  3 0
 6  0 3
 7
 8  4
 9  0 0
10  2 2
11  1 1
12  0 2
```

Output

```
1  Case #1: 1 3 4
2  Case #2: 1 2 4
```

SS15N09C   Burgers and Chips

Author: Chris Pinkau

As much as every other person, Lea loves fastfood. But as eating all by herself does make her feel guilty about all the calories, she needs some friends to accompany her to the nearest **TBOBT** (**T**he **B**est **O**ption for **B**urgers in **T**own). Lea is in a very good mood and invites her friends to some burgers and chips. She places her order at the bar and receives the burgers. The efficient person that she is, she does not want to walk back to the table with all the burgers more than once, so she asks for a big plate to put all the food on. She is very finicky when it comes to the arrangement on her plate, that is why she developed a system which is to arrange the burgers side by side at the edge of the plate in a ring around a big mountain of chips in the middle. Nothing must be lost on the way back to the table, so all burgers must lie fully on the plate. However, the height of the mountain of chips in the middle is no issue, Lea is a master of chips stacking. Can you tell Lea what size the plate must be at least to fit all the ordered burgers on it?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a line break.

Each test case consists of two integers r and b, with r being the radius of a burger (a round one) and b being the number of burgers ordered.

Assume that the plate is round. Moreover, because Lea is a chips stacking master, she can successfully stack chips on top of burgers, should the case occur (and it does occur with less than three burgers).

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the smallest radius of a plate that can hold all b burgers and the chips.

Your solution is considered correct if the area is accurate to six decimal places.

Constraints

- $1 \le t \le 100$
- $1 \le r \le 10000$
- $1 \le b \le 10000$

Sample Data

Input

```
1  2
2  3 2
3  4 9
```

Output

```
1  Case #1: 6.0
2  Case #2: 15.69521760065235
```

SS15N09D   Fragile Letters

Author: Stefan Toman

The company Lea is working at recently bought a new building to provide offices for all employees. The new building is a skyscraper situated in the city center and widely visible from all over the town. Since the company invests heavily in advertisements, the management decided to write its name on the outer wall of the new building, too. They bought big letters that glow in the dark. The letters got delivered today, but will only be mounted next week.

When the shipping company was about to unload the letters, Lea went outside to have a break in the company-owned park and stopped by to see the huge letters. The workers were debating loudly, so Lea joined them and asked what they are arguing about. It turned out they were not sure how to position the letters. They should stand vertically, due to technical reasons, but it is possible to rotate them or even turn them upside down. Obviously, the letters should be in a stable position and should not break, but they do not even know how many such positions there are. Can you help them together with Lea?

Input

Lea measured all of the letters. Each of the letters represents one test case. Since they have a very modern font, the letters are polygons (as seen when standing next to them), which means they do not contain holes and consist of straight lines only. Lea measured the position of all of the letter's vertices and computed their two-dimensional coordinates.

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with an integer n, the number of vertices. n lines follow describing the vertices. The i-th line contains two doubles x_i and y_i, the coordinates of the i-th vertex, The points are given in order, but Lea forgot whether she wrote them down clockwise or counter-clockwise. Note that due to the modern font the letters may not look like what you would expect a letter to look like. Consider them as a general simple polygon.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the number of stable positions of the letter. Each line of the output should end with a line break.

A position is considered stable if it touches the ground with exactly one edge and no vertex except the ones incident to that edge. Standing on an additional vertex or multiple edges would break the letter since it is not made for standing on the ground. Additionally, the center of mass of the letter's vertices (all mass of the letter is contained in the mountings at the vertices) needs to be above the lowermost edge or otherwise the letter would break. For instance, the letter "V" in normal fonts (Arial, for instance) has three stable positions, but a "T" has only two.
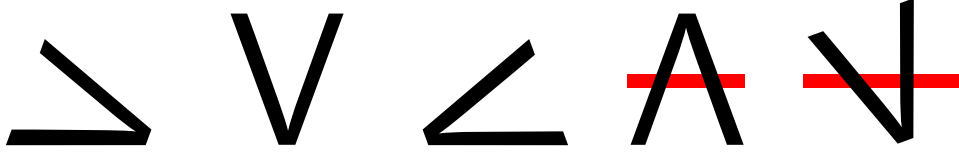
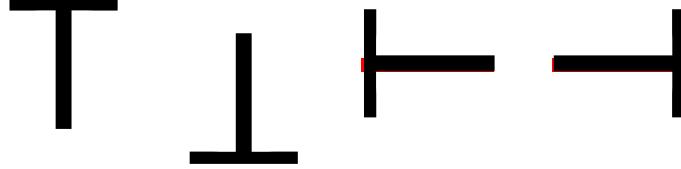Figure 1: The letter "V" has three stable positions.

Figure 2: The letter "T" has two stable positions.

Constraints

- $1 \leq t \leq 20$

- $3 \leq n \leq 100$

- $0 \leq x_i, y_i \leq 1000$ for all $1 \leq i \leq n$

Sample Data

Input

```
3
7
0.0 2.0
1.0 2.0
2.0 1.0
3.0 2.0
14.0 2.0
13.0 0.0
1.0 0.0

5
0.0 0.0
1.0 0.0
2.0 1.0
3.0 0.0
3.0 5.0

8
1.0 0.0
1.0 2.0
0.0 2.0
0.0 3.0
3.0 3.0
3.0 2.0
2.0 2.0
2.0 0.0
```

Output

```
Case #1: 1
Case #2: 2
Case #3: 2
```

SS15N09E   Meteorite Revisited

Author: Philipp Hoffmann

Lea has finally fenced in her garden and computed that the meteorite will not hit her house. But then, everything went wrong! The laser that heated up the meteorite had too much power and the meteorite exploded into thousands of shards. These shards came down everywhere in Lea's garden and the surroundings.

Of course, Lea is excited and wants to start looking for meteorite shards right away. To her amazement, she discovers a pattern in the shard distribution: The shards exactly hit every integer coordinate in the area. Lea, quite the archaeologist, wants to rope in the area inside her garden where shards might have landed. Help her by telling her at which integer locations inside her garden she should locate posts such that all shard location (inside the garden) are inside the convex hull of the posts. Furthermore, use the minimal number of posts. Points on the boundary of the garden are considerd to be inside the garden.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with an integer n, the number of fence posts that define the border of Lea's garden, n lines follow. The i-th line contains two floating point numbers x_i, y_i, the x- and y-coordinates of the i-th fence post. The fence posts are given in counter clockwise order and form a convex area.

Output

For each test case, output one line containing "Case #i:" where i is its number, starting at 1. In the next line output a single number m, the number of posts Lea has to place inside her garden. She may not use existing fence posts (but may place a post at the same location as an existing fence post). Output m more lines. The j-th line should contain two integers u_j, v_j, the x- and y-coordinates of the j-th post Lea has to place. Each line of the output should end with a line break. Any permutation of the correct post locations will be considered correct.

Should all shard locations lie on a line, give the two end points, should only one shard location be inside the garden, give that point. If no shard location lies inside the garden, Lea will be very sad and will not place any posts.

Constraints

- $1 \le t \le 20$
- $3 \le n \le 100$

- $0 \leq x_i, y_i \leq 300$ for all $1 \leq i \leq n$

- No two points given will be identical.

- The points are given in counter clockwise order and are the vertices of a convex area.

Sample Data

Input

```
1  4
2  4
3  0 0
4  1 0
5  1 1
6  0 1
7
8  4
9  0.5 0.5
10 2.5 0.5
11 1.75 1.75
12 0.5 2.5
13
14 3
15 0.5 0.5
16 1 0.5
17 0.5 1
18
19 3
20 0.5 0.5
21 1.5 0.5
22 0.5 1.5
```

Output

```
1  Case #1:
2  4
3  0 0
4  1 0
5  1 1
6  0 1
7  Case #2:
8  3
9  1 1
10 2 1
11 1 2
12 Case #3:
13 0
14 Case #4:
15 1
16 1 1
```