Algorithms for Programming Contests WS15/16 - Week 12

Chair for Efficient Algorithms (LEA), TU München

Prof. Dr. Harald Räcke

Moritz Fuchs, Philipp Hoffmann, Christian Müller, Chris Pinkau, Stefan Toman

This problem set is due by

Wednesday, 20.01.2016, 6:00 a.m.

Try to solve all the problems and submit them at

https://judge.in.tum.de/conpra/

This week's problems are:

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

Problem	A	B	C	D	E
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.

This page is intentionally left blank.

Problem A
Fence Posts

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 Input 1

```
2
4
0 0
1 1
3 0
0 3

4
0 0
2 2
1 1
0 2
```

Sample Output 1

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

Sample Input 2

```
6
4
3 8
2 7
5 1
6 5

6
6 6
4 5
2 4
9 1
7 2
9 4

4
0 4
8 0
8 1
8 5

6
5 7
1 6
8 1
4 2
2 0
6 5

8
2 2
6 7
0 1
0 4
8 3
10 6
9 6
3 1

3
0 3
9 3
9 8
```

Sample Output 2

```
Case #1: 1 2 3 4
Case #2: 1 3 4 6
Case #3: 1 2 4
Case #4: 1 2 3 5
Case #5: 2 3 4 5 6 8
Case #6: 1 2 3
```

Problem B
Meteorite

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

- $1 \leq t \leq 20$
- $3 \leq n \leq 1000$
- $-1000 \leq x_i, y_i \leq 1000$
- 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 Input 1

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

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

Sample Output 1

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

Sample Input 2

```
3
-3 4 10
5 0 2 -2
1 -2 0 -5
2 -2 1 -2
-2 -1 -5 -1
0 -5 -1 -1
-5 -1 -5 0
-2 3 5 0
-5 0 -2 4
-2 4 -2 3
-1 -1 -2 -1

-2 0 10
-4 -1 -5 0
-1 1 0 4
-5 0 -5 1
-5 1 -1 1
5 0 3 -4
3 -4 2 -5
0 4 4 4
-2 -1 -4 -1
4 4 5 0
2 -5 -2 -1

3 -3 9
1 3 5 0
-5 0 -5 3
5 0 1 -4
-5 3 -3 3
0 3 1 3
-3 3 0 3
-3 -2 -5 -3
-5 -3 -5 0
1 -4 -3 -2
```

Sample Output 2

```
Case #1: too bad
Case #2: jackpot
Case #3: too bad
```

Problem C
Snowball Fight

Winter is a special time. The first real snowstorm hits, the lakes freeze over, the houses become snowcapped, trees have finally shed all their colors and become frosty skeletons. Everything outside becomes quiet.

Well, not entirely… Ever since they were small children, Lea and her friends have always loved this time. Why? Because once a year, the whole town comes together and celebrates the "Snowball Arena: Free-for-all". For a day, the whole town stands still and anyone who is spotted on the streets can be subject to snowball bombardment. It is an event of joy and of tears, where grown men cry, bombarded by endless blizzards of snowballs, thrown by the hands of children. To make it even more fun, the townspeople have dug trenches on the central square so there are now several fronts that can be besieged.

Of course, Lea takes part in all that. Right now, she is lying alone in one of the trenches and is bombarded by one particularly persistent fellow. To avoid being hit, she needs to keep her head low - thus, she cannot spot her attacker directly. All his snowballs came from the same direction however, so to retaliate, she only needs to know how far her attacker is away.

Luckily, she notices something - to the side of the central square, there is a huge new building[1]. There are no visible windows on the front wall, but rather the whole wall of that building is a huge mirror. Noticing that this could give her a substantial combat advantage, she watches out for movements in the mirror. As soon as she spots someone, she wants to lob a snowball over the siege lines and hit that person right in the face. Can you help her with the target computations?

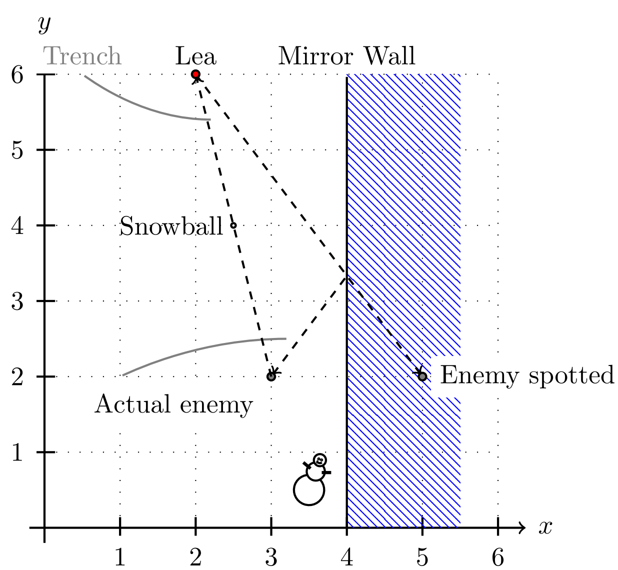

Figure C.1: Illustration of the sample input, case 1.

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 consists of 4 lines. The first line contains the integers x_{Lea} y_{Lea}, Lea's coordinates. The second line contains the doubles x_{Snow} y_{Snow}, some in-flight coordinates of one of the snowballs that came from the direction of the attacker. The third line contains the integers x_{Wall1} y_{Wall1} x_{Wall2} y_{Wall2}, the coordinates of the wall. The fourth line contains the doubles x_{enemy} y_{enemy}, the projected coordinates of the enemy she saw in the mirror.

[1]It is owned by some huge tech corporation. Lea forgot the name, but it was something like "Mirror's Ledge".

Output

For each test case, print a line containing "Case #i: x y!" where i is its number, starting at 1, and (x, y) are the coordinates of the enemy. Your solution is considered correct if the area is accurate to four decimal places. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $y_{Wall1} = 0$
- $y_{Wall2} = 30$
- All given coordinates are between 0 and 30.
- All points are at least 10^{-4} apart.
- All points are at least 10^{-4} away from the wall.

Sample Input 1

```
1
2 6
2.5 4.0
4 0 4 30
5 2
```

Sample Output 1

```
Case #1: 3.0 2.0
```

Sample Input 2

```
5
14 10
16.5 9.5
20 30 20 0
21.0 9.0

16 15
11.5 10.0
17 30 17 0
27.0 5.0

5 13
10.0 9.5
19 30 19 0
23.0 6.0

19 19
17.0 10.5
20 30 20 0
25.0 2.0

6 17
11.0 13.0
18 30 18 0
20.0 9.0
```

Sample Output 2

```
Case #1: 19.0 9.0
Case #2: 7.0 5.0
Case #3: 15.0 6.0
Case #4: 15.0 2.0
Case #5: 16.0 9.0
```

Problem D
Fragile Letters

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.

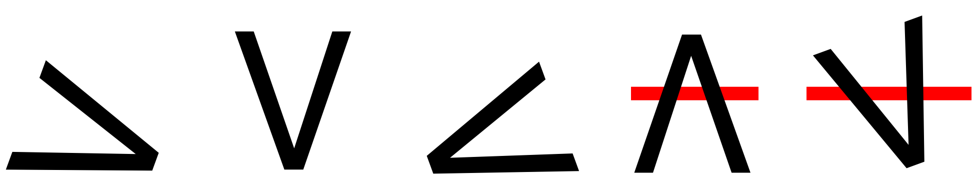

Figure D.1: The letter "V" has three stable positions.

Constraints

- $1 \leq t \leq 20$
- $3 \leq n \leq 50$
- $0 \leq x_i, y_i \leq 1000$ for all $1 \leq i \leq n$

Figure D.2: The letter “T” has two stable positions.

Sample Input 1

```
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
```

Sample Output 1

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

Sample Input 2

```
3
3
69.65178077847345 476.5867758189318
821.1853328040016 88.77793357104213
647.0223390027927 271.8248148626079

4
181.44458803946185 940.3664887563629
9.073553867733452 668.9316108351632
300.5477457248187 232.847850879327
736.0950978507178 376.24666402600525

6
589.537247768286 277.42135454463437
894.9913870012382 124.39634658639598
45.423403877217 250.24312860783692
509.6408941478913 630.1328761488896
166.30830193549738 595.7205262511858
826.6580132415487 795.3440293977031
```

Sample Output 2

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

Problem E
Meteorite Revisited

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

- $1 \leq t \leq 20$
- $3 \leq n \leq 100$
- $0 \leq x_i, y_i \leq 300$ 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 Input 1

```
4
4
0 0
1 0
1 1
0 1

4
0.5 0.5
2.5 0.5
1.75 1.75
0.5 2.5

3
0.5 0.5
1 0.5
0.5 1

3
0.5 0.5
1.5 0.5
0.5 1.5
```

Sample Output 1

```
Case #1:
4
0 0
1 0
1 1
0 1
Case #2:
3
1 1
2 1
1 2
Case #3:
0
Case #4:
1
1 1
```

Sample Input 2

```
4
4
1.375 7.578125
1.484375 7.046875
9.546875 0.203125
9.4375 0.734375

6
2.15625 5.125
5.375 1.859375
8.109375 2.65625
9.0 4.796875
5.78125 8.0625
3.046875 7.265625

6
2.296875 9.140625
2.5 6.828125
4.875 5.28125
8.34375 3.234375
8.140625 5.546875
5.765625 7.09375

4
1.75 8.875
3.359375 6.609375
9.109375 3.796875
7.5 6.0625
```

Sample Output 2

```
Case #1:
3
2 7
4 5
9 1
Case #2:
6
3 5
5 3
8 3
8 5
6 7
3 7
Case #3:
7
3 7
4 6
8 4
8 5
7 6
4 8
3 8
Case #4:
5
3 8
5 6
7 5
8 5
7 6
```