Algorithms for Programming Contests
WS18 - Week 11

Chair for Foundations of Software Reliability and Theoretical Computer Science,
TU München

Tobias Meggendorfer, Philipp Meyer,
Christian Müller, Gregor Schwarz

Welcome to our practical course! This problem set is due by

Wednesday, 23.01.2019, 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
Beer Pipes

After a stressful work day, Lea enjoys a nice cold beverage while sitting on her couch in front of the TV. Like most of the people from the region she comes from, she usually enjoys a beer on these occasions. And after having a few sips of the exquisite golden liquid, she contemplates the work that is put behind brewing such a masterpiece. Thus, she decides to visit the **BIER** (**B**rewery of **I**nternational **E**xcellence and **R**elevance), one of the many local breweries, on the next day to learn a bit more about the process behind her favourite beverage. Apart from all the usual brewery tour, she meets Mr. Barley Hops, the CEO of **BIER**. Recognising Lea, he says (in a heavy German accent) "Guten Tag my dear Fräulein Lea. I have heard about you and your problem solving skills, maybe you can help us? The workers installed new pipes for the Bier. They were so drunk, every pipe has a different shape und we don't know how much Bier we can pump into the pipes." Lea immediately sees the problem: the beer is poured into a pipe on one end of the brewery and exits at one valve at the other end in a great cauldron. In between there is a whole system of pipes that are connected in a seemingly chaotic fashion and are all shaped very differently. The question at stake is to come up with the highest amount of beer that can be put through the system so that the beer cauldron at the end is filled with as much beer as possible. Unfortunately, Lea is very busy right now, so she wants you to take a look at the problem. Make sure you can help Mr. Hops because he will grant you a lifetime supply of **BIER** beer if your solution is optimal!

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 two integers n and m, n being the number of valves $\{v_1, \ldots, v_n\}$ that connect the pipes, and m the number of pipes in the system. m lines follow where line i consists of three integers a_i, b_i and k_i, and a double x_i, denoting that there is a pipe that connects the valves v_{a_i} and v_{b_i} whose cross section has the shape of a regular polygon with k_i sides and side length x_i. If k_i is equal to 0, then the pipe is cylindrical with radius x_i.

The maximal amount of beer that can flow through a pipe is measured by the area of its cross section.

The first valve, where the beer enters the pipe system, is v_1, the exit of the pipe system, at the large beer cauldron, is v_n.

Beer in the pipes can flow in both directions.

Output

For each test case, output one line containing "Case #i: y" where i is its number, starting at 1, and y is either the maximal amount of beer that can be poured into v_1 with an absolute error of up to 10^{-8}, or "impossible" if that amount is 0.

Constraints

- $1 \leq t \leq 20$
- $3 \leq n \leq 1000$
- $1 \leq m \leq 2000$
- $1 \leq a_i, b_i \leq n$ for all $1 \leq i \leq m$
- $3 \leq k_i \leq 20$ or $k_i = 0$ for all $1 \leq i \leq m$
- $1 < x_i \leq 100$ for all $1 \leq i \leq m$

Sample Input 1

```
2
4 5
1 2 0 3.3
1 3 3 1.5
2 3 0 2.2
2 4 5 4.1
3 4 4 2.5

3 2
1 2 0 1.2
1 2 4 2
```

Sample Output 1

```
Case #1: 35.17122510390053
Case #2: impossible
```

Sample Input 2

```
3
7 9
4 1 3 7.1373647190400025
1 2 4 3.619952082374767
7 4 4 3.9229060952100747
1 5 5 5.149342554193151
6 7 0 5.9289433379154115
7 3 3 4.480317767968532
5 7 4 9.175216600972856
3 5 3 5.809582781390198
3 5 6 1.5833251223822389

10 10
9 1 0 6.103567585805486
2 9 4 1.2239623087641212
3 3 3 5.74365487996366
4 3 5 5.509203206150823
5 1 5 4.490092275486687
1 5 3 3.8459000369208165
5 6 0 3.254413686940226
2 1 4 5.257332360634563
1 1 3 3.274553532434247
6 7 4 3.260082839139213

10 5
4 8 5 7.461085908138405
8 10 0 8.000823060758666
8 1 6 4.696393660473783
1 9 6 6.339265904299603
8 3 4 8.535902037766256
```

Sample Output 2

```
Case #1: 61.00890428988396
Case #2: impossible
Case #3: 57.30346357618359
```

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
Digging for Treasure

On her visit to Absurdistan, Lea met many strange and interesting people. Some evening in a shady bar, she met a guy who was missing a leg. He told her he had been a pirate all his life until, one day, he contracted a severe case of seasickness that just wouldn't stop. After that, he never set sail again.

Lea bought him a couple of drinks and he began telling her his stories, boasting of booty and plunder. He even showed her one of his old treasure maps - it was an island, littered with different marks for buried treasure. The pirates did not want to miss a single piece of treasure. So, he told her, together with his crew, he dug up one big, convex hole, 2 meters deep, containing all of the treasure markers.

Lea could not believe him. Surely, this would take ages. Can you help her estimate how much earth the crew would have to move to dig such a hole?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case begins with a line containing one integer n, the number of treasure markers on the island. n lines follow, each containing two integers x_i y_i detailing the location of one of the treasure markers.

Output

For each test case, print a line containing "Case #i: s" with i being the number of the test case, starting at 1, and s being the volume of the hole the crew would have to dig (in m^3). The answer should be correct up to an absolute or relative error of at most 10^{-6}.

Constraints

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

Sample Input 1	Sample Output 1
2 3 1 1 2 2 1 3 5 0 0 2 2 0 2 2 0 1 1	Case #1: 2.0 Case #2: 8.0

Sample Input 2	Sample Output 2
5 5 -6 -7 -4 8 -5 -5 5 -5 -2 5 3 0 -5 3 3 4 4 5 3 5 -5 8 9 8 -6 -6 0 -5 8 10 7 1 0 2 1 8 0 4 9 3 1 3 3 6 8 7 6 7 3 5 1 8 5 3 5 0 7 7 10 0	Case #1: 161.0 Case #2: 5.0 Case #3: 265.0 Case #4: 109.0 Case #5: 79.0

Problem D
Burgers and Chips

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

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

Sample Input 1

```
2
3 2
4 9
```

Sample Output 1

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

Sample Input 2

```
9
8 2
6 9
10 4
1 6
2 5
2 4
4 10
3 5
8 6
```

Sample Output 2

```
Case #1: 16.0
Case #2: 23.542826400978523
Case #3: 24.14213562373095
Case #4: 3.0
Case #5: 5.4026032334081595
Case #6: 4.82842712474619
Case #7: 16.94427190999916
Case #8: 8.10390485011224
Case #9: 24.0
```

This page is intentionally left blank.

Problem E
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 (i.e. the centroid of the polygon) 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.

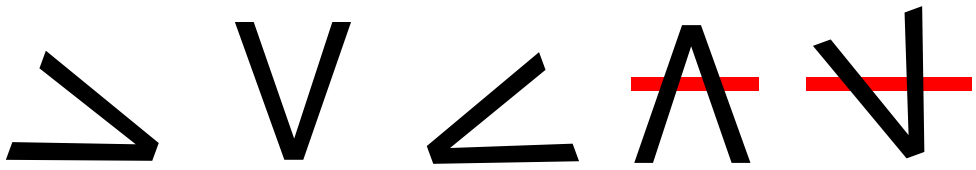

Figure E.1: The letter "V" has three stable positions.

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 E.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
```