
Algorithms for Programming Contests

This problem set is due by

Thursday, 09.07.2015, 6:00 a.m.

Try to solve all the problems and submit them at

<https://judge.in.tum.de/>

This week's problems are:

SS15N10A Euler Line	2
SS15N10B Templonian Excavation	4
SS15N10C Area 51	6
SS15N10D Fallingwater	8
SS15N10E Fractals	11

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

Problem	SS15N10A	SS15N10B	SS15N10C	SS15N10D	SS15N10E
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.

SS15N10A Euler Line

Author: Stefan Toman

Sometimes, Lea likes to read math books that are written for people without mathematical background and is always fascinated how everything fits together. She recently read a chapter of such a book which was dealing with the so-called Euler line. The Euler line is part of the following theorem shown by Leonhard Euler in 1765: Given any triangle, its orthocenter (the intersection of its altitudes), circumcenter (the intersection of its perpendicular bisectors) and centroid (the intersection of its medians) are colinear, which means they are on one line, the Euler line.

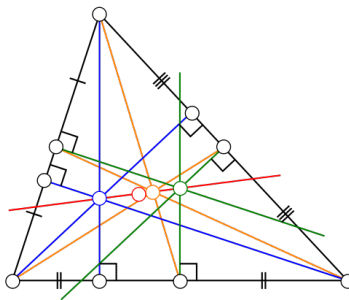


Figure 1: Euler's line (red) is a straight line through the centroid (orange), orthocenter (blue), circumcenter (green) and even the center of the nine-point circle, which is not considered in this problem (red). Source: wikipedia.org

Lea cannot believe this and thinks this is just a coincidence. The authors of the picture may just have chosen an example where it works. She wants to try this for some other examples and needs you to compute the coordinates of the points in question.

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 three lines, each of them containing two space-separated doubles: The x - and y -coordinates of the triangle's vertices.

Output

For each test case, output one line containing "Case # i :" where i is its number, starting at 1. Afterwards, print the x - and y -coordinates (separated by a space) of the triangle's centroid, orthocenter and circumcenter (in this order), each of them in a separate line. Each line of the output should end with a line break.

Your solution is considered correct if all coordinates are accurate to six decimal places.

Constraints

- $1 \leq t \leq 20$
- $-100 \leq x, y \leq 100$

Sample Data

Input

```
1 2
2 0.0 0.0
3 1.0 0.0
4 0.0 1.0
5
6 1.0 1.0
7 99.123 -12.5
8 58.54 -32.643
```

Output

```
1 Case #1:
2 0.3333333333333333 0.3333333333333333
3 -0.0 -0.0
4 0.5 0.5
5 Case #2:
6 52.88766666666666 -14.714333333333332
7 49.676013785025596 -97.06973476829143
8 54.49349310748721 26.463367384145705
```

SS15N10B Templonian Excavation

Author: Christian Müller

On her visit to Templonia, Lea comes by an old Templonian temple site, which has not yet been excavated completely. Fascinated, she spends the day talking to the archaeologists that are busily scouring the area, unearthing artifacts from ancient times.

In those times in Templonia, the number 3 was a very holy number. That is why they raised many monuments to it. The excavation site Lea visited is one of them.

It consists of several altars, each of them surrounded by 3 obelisks that form a triangle. Inside that triangle, they paved the floor with triangular floor tiles and for every 3 square meters of area of the triangle they set up a 3-armed candelabra (Since the area generally was not divisible by 3, they rounded the number of square meters to the nearest multiple of 3 and since they could never have enough 3s they also rounded the resulting number of candelabras to the nearest multiple of 3). They then held mass inside these triangles where they chanted songs about the holy 3 in which 3 viciously slaughtered every other number until it was the only number left. Then, they sacrificed 3 virgins, drank 3 different kinds of religious wine and danced until morning. All in all it sounded like a big party.

While she heard about all this, she watched the archaeologists uncover the floor tiles, candelabras, goblets, etc. As she really liked the old candelabras she wanted to know how many more the archaeologists would probably find.

Input

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

Each test case consists of 6 integers: $x_1y_1x_2y_2x_3y_3$. The points $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ specify the locations of the 3 obelisks (coordinates are given in meters).

Output

For each test case, output one line containing “Case $\#i$: *number*” where i is its number, starting at 1, and *number* is the amount of candelabras the Templonians used to illuminate their sacrificial parties. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $0 \leq x_i, y_j \leq 10000$
- $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ are pairwise distinct
- $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ form a convex area

Sample Data

Input

1	4
2	0 0 3 0 0 3
3	0 0 6 0 0 6
4	0 0 5 0 0 4
5	15 16 17 14 30 40

Output

1	Case #1: 3
2	Case #2: 6
3	Case #3: 3
4	Case #4: 12

SS15N10C Area 51

Author: Stefan Toman

Lea saw an UFO yesterday night. You do not believe her? She is absolutely convinced that these red and blue flashing lights she saw were made by aliens. Now, she wants to find out more about this UFO, but where to find it? She had been thinking about this problems for some hours and finally found the answer: Area 51. If the UFO crashed it is now at Area 51 for sure, that secret military base somewhere in the desert. Nobody knows too much about this military base, so Lea decides to infiltrate it.

At first, she needs to know more about Area 51. How big is it, where are the weak points of the border control, what about the condition of the soil and so on. Her first step is to walk along the whole fence around the building. She logs her GPS coordinates, makes photos and takes notes about the environment.

Later, in her hotel, she wants to evaluate the new data. In particular, she wants to know the size of the area the military base is built upon. Can you compute it for her?

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 a line containing a single integer n , the number of GPS points she recorded in order. n lines follow describing the points. The i -th line contains two doubles x_i and y_i , the x and y coordinates of the point, respectively. After the n -th point she got back to her starting position. Between the points, Lea always walked straight lines.

Output

For each test case, output one line containing “Case # i : x ” where i is its number, starting at 1, and x is the area of the military base in square kilometers. Each line of the output should end with a line break.

You may assume that the ground is totally flat and that Lea never walked the same place twice, except for the start and end of her trip, in particular, none of the straight lines she walked intersect except at the vertices. One unit in the coordinates is 1km long. Note that there are no further restrictions, in particular, the area may be highly non-convex.

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

Constraints

- $1 \leq t \leq 20$
- $3 \leq n \leq 50$
- $-100 \leq x_i, y_i \leq 100$ for all $1 \leq i \leq n$

Sample Data

Input

1	3
2	3
3	0.0 0.0
4	1.0 0.0
5	0.0 1.0
6	
7	3
8	0.0 0.0
9	0.0 1.0
10	1.0 0.0
11	
12	5
13	0.0 0.0
14	0.0 2.0
15	1.0 1.0
16	2.0 2.0
17	2.0 0.0

Output

1	Case #1: 0.500000
2	Case #2: 0.500000
3	Case #3: 3.000000

SS15N10D Fallingwater

Author: Philipp Hoffmann

Most likely, you have heard of the famous architectural masterpiece, Fallingwater. Frank Lloyd Wright designed it in 1935 and beautifully integrated the natural flow of water into the house.



Figure 2: Fallingwater (Kaufmann Residence) by Frank Lloyd Wright.

Lea has planned something much like this for her own house. She will use a waterfall and guide its flow by building stone ledges. She has just one problem: She is unsure where the water will end up. Can you help her?

For this problem, consider the waterfall as two-dimensional. The ledges are given by their start- and endpoint. Water falls down from a given source towards the ground and can never flow upwards. If it touches a ledge it will flow downwards or horizontally along the edge. On horizontal edges, the water splits and continues in both directions. If the point where the water hit the ledge is exactly an endpoint and furthermore the ledge does not go upwards, the water also splits, some of it continues to fall while the rest flows along the edge. See the sketch below.

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 a line containing three integers $n\ x\ y$ where n is the number of ledges, and (x, y) is the location of the water source which is considered to be a single point. n lines follow describing the ledges. The i -th line contains four integers $x_{i,1}y_{i,1}x_{i,2}y_{i,2}$ meaning that the i -th ledge goes from $(x_{i,1}, y_{i,1})$ to $(x_{i,2}, y_{i,2})$ in a straight line.

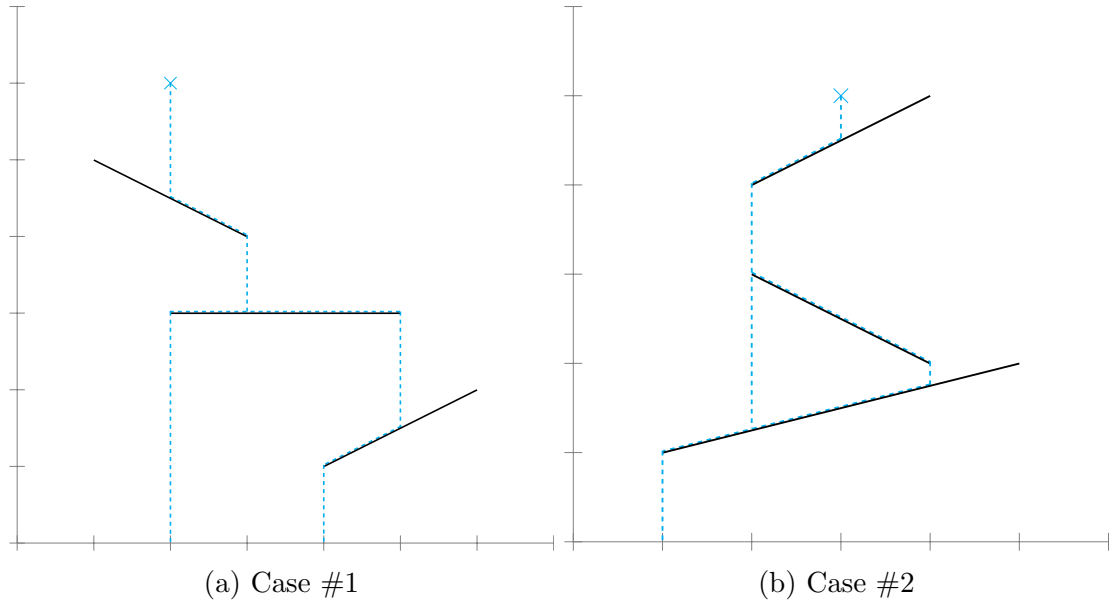


Figure 3: Illustration of the sample inputs.

Output

For each test case, print a line containing “Case # i : x ” where i is its number, starting at 1, and x is a space-separated, naturally ordered list of the x -coordinates at which water will arrive at the ground ($y = 0$). Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 100$
- $1 \leq x, y, x_{i,j}, y_{i,j} \leq 100$ for all $1 \leq i \leq n, 1 \leq j \leq 2$.
- $x_{i,1} \neq x_{i,2}$ for all $1 \leq i \leq n$.
- No two ledges will intersect. Every two points on different ledges are at least 10^{-4} apart.
- The water source is at least 10^{-4} away from each ledge.

Sample Data

Input

1	2
2	3 2 6
3	1 5 3 4
4	2 3 5 3
5	4 1 6 2
6	
7	3 3 5
8	2 4 4 5
9	2 3 4 2
10	1 1 5 2

Output

1	Case #1: 2 4
2	Case #2: 1

SS15N10E Fractals

Author: Stefan Toman

Since Lea likes to draw and she also likes mathematics, there is nothing more obvious than the fact that Lea absolutely loves fractals. Her newest project is a huge fractal in her garden: She wants to walk around the lawn and her footsteps should form a nice fractal that can be seen from her roof terrace and even from outer space!

The problem is that she needs clear instructions on how to walk since it is hard to see the fractal's structure if you stand inside. Lea found a solution for this problem. This is how she creates her walking instructions: She starts by choosing an integer d , some word s , an angle a and a set of n productions p . A production is a mapping from a letter to a word containing other letters or the characters “+” or “-”. Now, she replaces all letters in s by the strings given by the corresponding productions, “+” and “-” are not replaced. She repeats this process d times.

Afterwards, she starts walking using the string she just computed. She reads the characters one after another from left to right. If she reads a “+” sign, she turns a degrees to the left. If she reads a “-” sign, she turns a degrees to the right. If she reads some other character, she walks 1 meter in her current direction.

Lea wants to see the result of her inputs first without walking long distances. Can you tell her how the resulting structure will look like?

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 a single line containing space-separated integers n , d , a and the string s . n lines follow describing the productions. Each line has the form $x \Rightarrow y$ where x is a character and y is a string meaning that x is replaced by y .

Output

For each test case, print a line containing “Case # i :” where i is its number, starting at 1. Afterwards, print each point Lea will move to starting at $(0,0)$ in a single line. Each point should be given as a space-separated list of its coordinates. Each line of the output should end with a line break. Your answer will be considered correct if the absolute error of each number in the output is at most 10^{-2} .

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 26$

- $1 \leq d \leq 15$
- $1 \leq a \leq 359$
- All strings of the input consist of 1 to 20 upper case letters, “+” or “-”.
- There is a production vor every letter that appears in the input.
- The output of each test case contains at most 10^6 points.

Sample Data

Input

```

1 3
2 1 3 30 L
3 L=>LL+
4
5 2 3 45 L
6 R=>+R--L+
7 L=>-R++L-
8
9 2 1 60 L
10 R=>R+L++L-R--RR-L+
11 L=>-R+LL++L+R--R-L

```

Output

```

1 Case #1:
2 0.0 0.0
3 1.0 0.0
4 2.0 0.0
5 2.8660254037844393 0.5
6 3.7320508075688776 1.0
7 3.732050807568879 2.0
8 3.7320508075688785 3.0
9 3.2320508075688785 3.8660254037844384
10 2.7320508075688785 4.732050807568877
11 Case #2:
12 0.0 0.0
13 0.7071067811865476 0.7071067811865475
14 1.4142135623730951 0.0
15 0.7071067811865479 -0.7071067811865478
16 1.4142135623730954 -1.4142135623730951
17 2.121320343559643 -0.7071067811865474
18 2.8284271247461903 -1.414213562373095
19 3.535533905932738 -0.7071067811865472
20 2.8284271247461916 8.881784197001252E-16
21 Case #3:

```

```

22 0.0 0.0
23 0.5000000000000001 -0.8660254037844386
24 1.5 -0.8660254037844386
25 2.5 -0.8660254037844386
26 2.0000000000000004 -1.1102230246251565E-16
27 1.0000000000000004 3.3306690738754696E-16
28 1.5000000000000007 0.8660254037844389
29 2.5000000000000001 0.8660254037844389

```

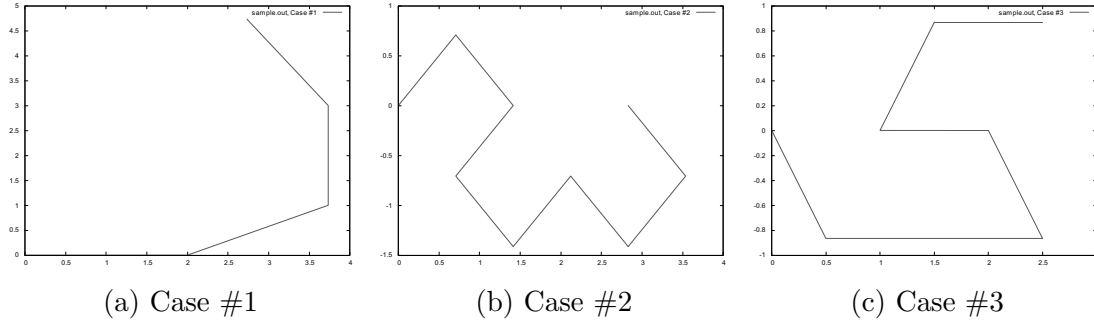


Figure 4: Illustration of the sample inputs.

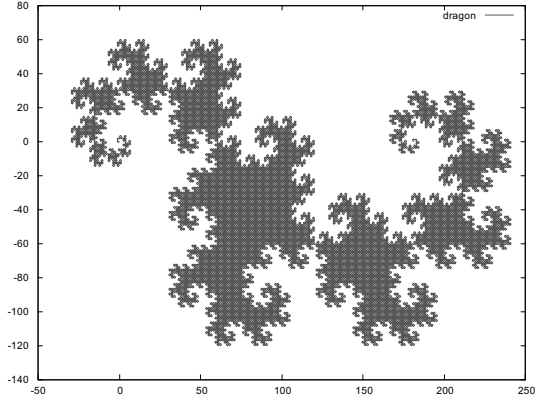
These pictures were generated with GNUplot. You can create the first picture with the following command:

```

1 gnuplot -e "plot 'data.out' every 1::1::9 \
2     title 'sample.out, Case \#1' with lines; pause -1"

```

Adjust the line numbers to draw the other test cases. Here are some other examples not mentioned in the sample input:



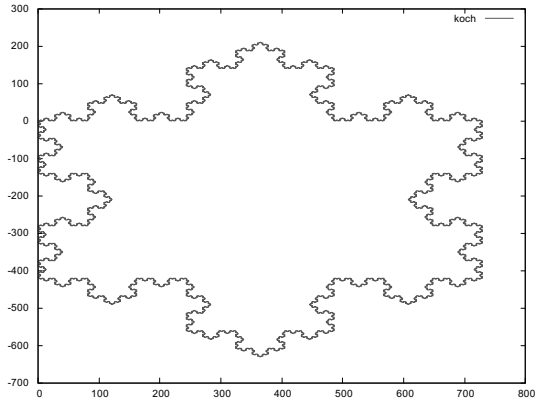
(a)

1

2 15 45 L

R=>+R-L+

L=>-R++L-

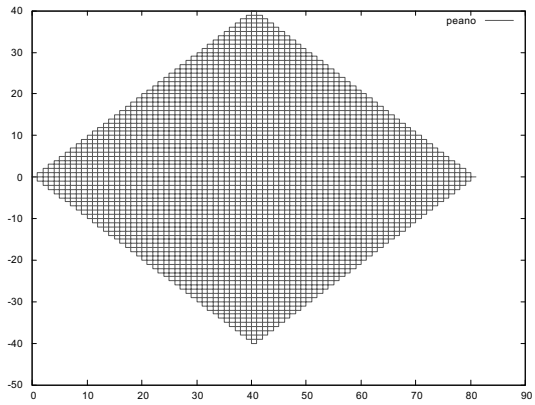


(c)

1

1 6 60 F-F-F

F=>F+F-F+F

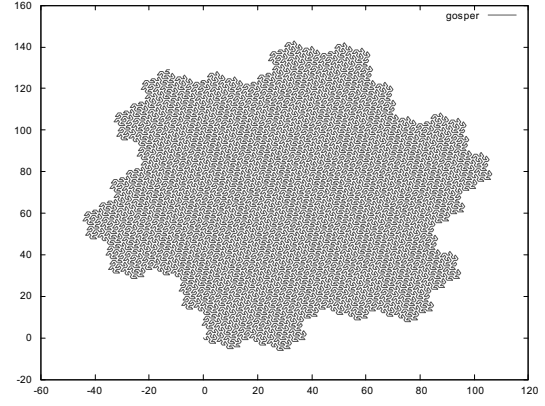


(e)

1

1 4 90 F

F=>F-F+F+F+F-F-F-F+F



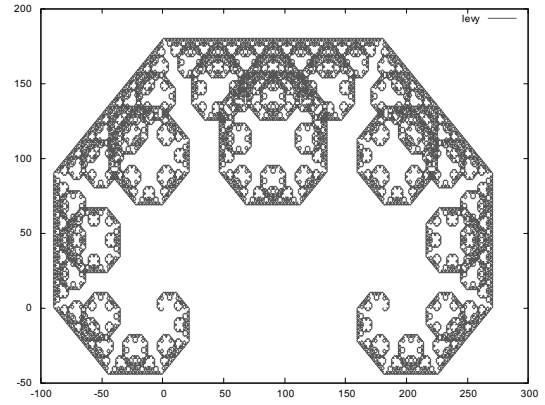
(b)

1

2 5 60 L

R=>R+L++L-R-RR-L+

L=>-R+LL++L+R-R-L

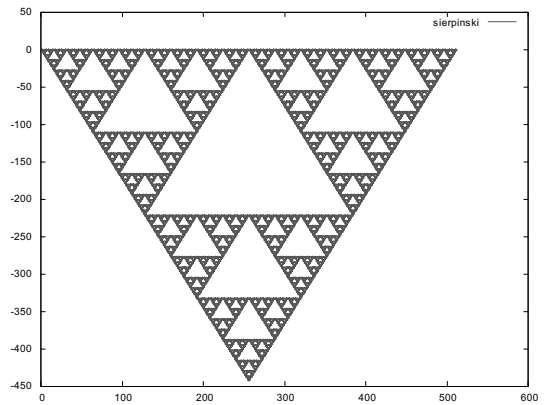


(d)

1

1 15 45 F

F=>+F-F+



(f)

1

2 8 60 FXF-FF-FF

X=>-FXF++FXF++FXF-

F=>FF

Figure 5: Illustration of additional inputs.