Problem Set

Think.

Create.

Solve.

2016 Asia-Manila Regional
Programming Contest
December 14–16, 2016
Ateneo de Manila University

ATENEO DE MANILA UNIVERSITY
acm International Collegiate Programming Contest
IBM. event sponsor
acer
THE GROVE by Rockwell
Smart
RCBC 56
accenture
frappe la rue
UBER
WHEN IN MANILA
ClickTheCity.com Around the City Online

Contents

Notes

- Many problems have large input file sizes, so we suggest using fast I/O. For example:
  - In Java, take your input with `BufferedReader` and buffer your output with `StringBuilder`.
  - In C/C++, take your input with `scanf` and write your output with `printf`.
- The problems are solvable in C, C++ and Java, but there's no guarantee that there's a solution in Python that passes the time limit.
- Good luck!

Problem A
AdoraBalls
Time Limit: 6 seconds

Since you had extra money, you decided to go to the local orphanage to donate some toys. After a few minutes of research, you discover that the children there like a new inexpensive kind of toy called AdoraBalls. These balls come in four different colors – azure, blue, cyan, and denim. One can obtain these adorable AdoraBalls by purchasing any of the following AdoraBall bundles:

- The **Bundle of Enjoyment** contains a_1 azure AdoraBalls, b_1 blue Adoraballs, c_1 cyan AdoraBalls, and d_1 denim AdoraBalls.
- The **Bundle of Festivity** contains a_2 azure AdoraBalls, b_2 blue Adoraballs, c_2 cyan AdoraBalls, and d_2 denim AdoraBalls.
- The **Bundle of Glee** contains a_3 azure AdoraBalls, b_3 blue Adoraballs, c_3 cyan AdoraBalls, and d_3 denim AdoraBalls.
- The **Bundle of Happiness** contains a_4 azure AdoraBalls, b_4 blue Adoraballs, c_4 cyan AdoraBalls, and d_4 denim AdoraBalls.

With this info, you decided to go to the local orphanage to donate AdoraBalls for them to play with. Each child in the orphanage has exactly one favorite color among the four. For example, some of the younger ones like to have cyan balls because of their brightness. Most of the mature ones prefer to have blue balls. As you are a thoughtful donor, you want to make sure that each child only receives AdoraBalls which are of their favorite color. Moreover, to prevent jealousy within themselves, you decide that you must give each child exactly the same number of AdoraBalls. Furthermore, since you are not cruel, you decide that each child should receive at least one AdoraBall.

Your plan is then to buy a combination of these bundles, unpack them and give the same number of AdoraBalls to each child. Now, you decide on buying E Bundles of Enjoyment, F Bundles of Festivity, G Bundles of Glee and H Bundles of Happiness. You want to determine if there exist E, F, G, H such that each child receives AdoraBalls which are exclusively their favorite color and each receives the same non-zero amount of AdoraBalls. Moreover, you require that none of them gets left over because you don't particularly find these AdoraBalls adorable.

If there are exactly a_0 children whose favorite color is azure, b_0 children whose favorite color is blue, c_0 children whose favorite color is cyan, and d_0 children whose favorite color is denim, is it possible to find non-negative integers E, F, G, H satisfying your constraints?

Input

The first line of input contains a single integer T, the number of test cases.

Each test case is described using five lines.

- The first line of each test case contains four space-separated integers a_0, b_0, c_0, and d_0, the number of children whose favorite colors are azure, blue, cyan and denim, respectively.
- The next four lines describe the contents of the bundles. The i^{th} line of these four lines give the values of a_i, b_i, c_i, and d_i, as indicated in the problem statement.

Output

For each test case, output a line containing a single string S, where $S =$ `POSSIBALL` if it is possible to find E, F, G, H satisfying the constraints described in the problem statement and $S =$ `IMPOSSIBALL` otherwise.

Constraints

$1 \leq T \leq 20000$
$0 \leq a_i, b_i, c_i, d_i \leq 55$

Sample Input	Sample Output
3 3 3 4 4 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 3 3 4 4 1 0 0 0 0 1 0 0 0 0 1 0 1 1 1 0 3 3 4 4 1 2 4 3 2 1 1 4 2 4 3 1 2 1 4 1	POSSIBALL IMPOSSIBALL POSSIBALL

Problem B
Balloon Distribution
Time Limit: 6 seconds

There are M contestants in a certain contest. The i^{th} contestant garnered P_i points, for $i = 1, 2, \ldots, M$.

The organizers have N balloons which shall be awarded to the contestants in the following manner:

1. The ratios $\frac{P_i}{j}$ are computed for each contestants $i = 1, 2, \ldots, M$ with divisors $j = 1, 2, 3, \ldots$ and so on.

2. These computed ratios are listed linearly and ranked from the highest value down to the lowest value. If two ratios are equal, they are ranked the same. To be more precise, the *rank* of a ratio is equal to one plus the number of ratios bigger than it.

3. A contestant shall receive one balloon for each of his or her computed ratios with rank that is less than or equal to N. In case there are ties in the last rank, all contestants tied in that rank will be awarded balloons. This means that more than N balloons may potentially be awarded.

The problem is to find the number of balloons that are awarded to each contestant.

Here's an example. Consider the case with $M = 5$ contestants having the following points:

$$P_1 = 274{,}771$$
$$P_2 = 344{,}854$$
$$P_3 = 773{,}780$$
$$P_4 = 627{,}629$$
$$P_5 = 386{,}890$$

Suppose there are $N = 10$ balloons. Then the allocation is (approximately) given as follows:

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	...
#1	274771.00	137385.50	91590.33	68692.75	54954.20	45795.17	39253.00	...
#2	344854.00	172427.00	114951.33	86213.50	68970.80	57475.67	49264.86	...
#3	773780.00	386890.00	257926.67	193445.00	154756.00	128963.33	110540.00	...
#4	627629.00	313814.50	209209.67	156907.25	125525.80	104604.83	89661.29	...
#5	386890.00	193445.00	128963.33	96722.50	77378.00	64481.67	55270.00	...

The ranks of each of these ratios near 10 are given as follows:

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	...
#1	**7**	15	*	*	*	*	*	...
#2	**5**	12	19	*	*	*	*	...
#3	**1**	**3**	**8**	**10**	14	16	*	...
#4	**2**	**6**	**9**	13	18	*	*	...
#5	**3**	**10**	16	*	*	*	*	...

Therefore, the number of balloons given to each of the 5 contestants is 1, 1, 4, 3 and 2, respectively.

Notice also that 11 balloons are awarded in total. This is because two ratios tied at rank 10.

Input

The first line of input contains T, the number of test cases.

The first line of each test case contains two integers, M and N, separated by a space. The second line contains M space-separated integers $P_1, P_2, \ldots, P_M$.

Output

For each test case, output M integers separated by single spaces, the i^{th} of which denotes the number of balloons the i^{th} contestant receives.

Constraints

$1 \leq T \leq 3000$
$1 \leq M \leq 10^5$
The sum of the Ms is $\leq 4 \cdot 10^5$
$1 \leq N \leq 10^9$
$1 \leq P_i \leq 10^9$

Sample Input	Sample Output
2 5 10 274771 344854 773780 627629 348561 5 10 274771 344854 773780 627629 386890	1 1 4 3 1 1 1 4 3 2

Problem C
Convex Quadrilateral
Time Limit: 9 seconds

Given n points on the plane, namely $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, your problem is to write a computer program that constructs a quadrilateral Q that satisfies all of the following conditions.

1. Each of the four sides of Q must pass through at least two of the n given points.
2. Q must be convex.
3. All of the n points must be inside Q or on Q.
4. Among all quadrilaterals satisfying (1) to (3), Q must have minimum area. It is possible that such quadrilateral is not unique; there may be several such quadrilaterals with minimum area. Your program should compute the value of this minimum area.

Input

The first line of input contains a single integer T indicating the number of data sets. The following lines describe the data sets.

The first line of each data set will contain a single integer n. This is followed by n lines, the i^{th} line of which contains x_i and y_i separated by a space, the (real-valued) x- and y-coordinates of the i^{th} point, respectively.

Output

For each data set, print a single line containing the minimum area, if such a quadrilateral exists. Otherwise print “`none`” (without quotes). Your answer should be accurate to within an absolute error of 10^{-6} from the correct answer.

Constraints

$1 \le T \le 60$
$1 \le n \le 300$
$|x_i|, |y_i| \le 1000.00$
Coordinates are given with at most two decimal places.
The points are distinct.

Sample Input

```
2
9
0 1
1 0
1 3
1 5
3 1
3 3
5 0
5 2
6 4
4
-1 -1
5 -1
0 0
-1 51
```

Sample Output

```
23.625
none
```

Problem D
Disco Dance Debacle
Time Limit: 5 seconds

Daryl owns a disco dance club named the Disco Dance Den. It has a disco dance floor which is in the shape of an $m \times n$ grid. At first, all of these cells are lit up. When the music starts playing, some cells of this grid become unlit.

A **disco dance** is a sequence of *at least two* steps done by a disco dancer while the music plays. The last step of a disco dance must be on the disco dancer's starting cell.

A **disco dancer** only steps on lit cells while doing a disco dance. At the exact moment he steps on a lit cell, it immediately becomes unlit. Moreover, a disco dancer always satisfies the following four conditions:

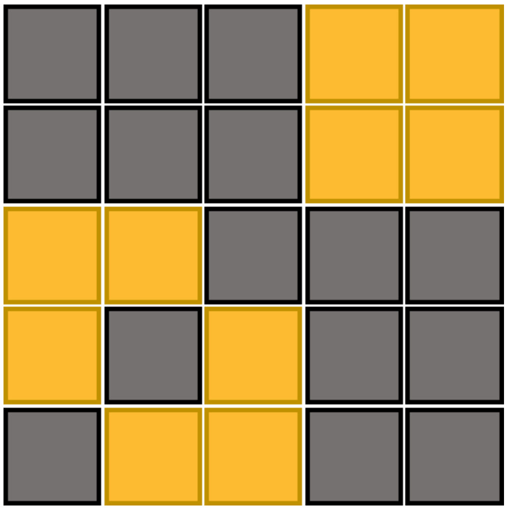
1. A disco dancer alternates using his left foot and right foot at each step. He never moves both feet at the same time.
2. Whenever he moves his left foot, it always steps on a lit cell which is on the same *column* as his right foot.
3. Whenever he moves his right foot, it always steps on a lit cell which is on the same *row* of as his left foot.
4. The foot he uses for his first and last step is different.

Now, we say that Daryl's Disco Dance Den **is dark** if all the cells of the disco dance floor becomes unlit.

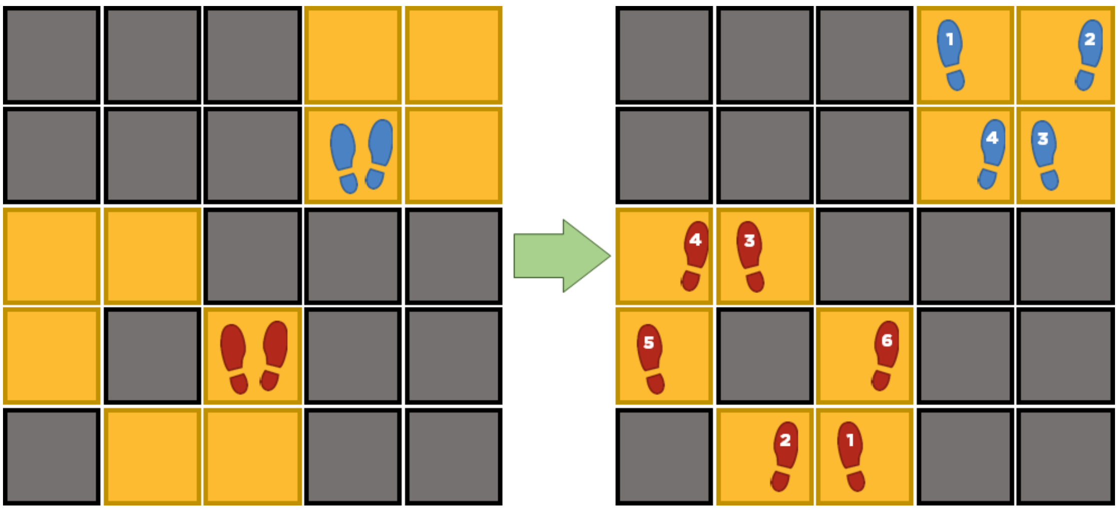
Daryl discusses a disco dare with a group of disco dancers in his den. The disco dare goes as follows:

1. Daryl tells the disco dancers which cells will remain lit and which cells will become unlit when the music starts.
2. The disco dancers choose how many of them go on the dance floor. They can choose to let none of them go on the dance floor.
3. The disco dancers decide their starting cells and stand on them before the music starts. We clarify that standing on a cell is not the same as stepping on it.
4. The disco dancers on the dance floor must each perform a disco dance.
5. The disco dancers on the dance floor must do their disco dances one-by-one.
6. After the last disco dancer finishes his disco dance, the Disco Dance Den must be dark.

As an example, if this is the setup:



Then the disco dare can be done by having two disco dancers start at the cells indicated in the left side of the figure below:



After all the disco dancers finish their disco dance, Daryl's Disco Dance Den is expected to go dark since all the lit cells have been stepped on. In addition, the two disco dancers on the dance floor satisfied all four conditions a disco dancer must satisfy.

Is it possible for them to do the disco dare they discussed with Daryl? If not, what is the smallest number of cells that must change their state (from lit to unlit, or vice-versa) so that it becomes possible for them to do the disco dare?

Input

The first line of input contains a single integer T, the number of test cases. The following lines describe the test cases.

The first line of each test case contains three space-separated integers m, n and k, in that order, specifying the number of rows and columns of the grid, respectively, and the number of *groups of cells* which will become unlit once the music starts.

The next k lines describe which cells of the disco dance floor become unlit when the music starts. Each of these lines contain four space-separated integers i_0, j_0, i_1, j_1. This tells us that for all i and j that satisfy $i_0 \le i \le i_1$ and $j_0 \le j \le j_1$, the cell on the i^{th} row and j^{th} column becomes unlit when the music starts.

Note: The groups of unlit cells may overlap.

Output

For each test case, one line containing a single integer N, which is the smallest number of cells whose states must change so that the disco dancers can complete Daryl's disco dare.

Constraints

$1 \le T \le 2000$
$0 \le k \le 10^5$
$1 \le m, n \le 10^5$
$1 \le i_0 \le i_1 \le m$
$1 \le j_0 \le j_1 \le n$
The sum of the ks is $\le 3 \cdot 10^5$

Sample Input

```
3
5 5 4
1 1 2 3
1 3 3 3
3 4 5 5
5 1 5 1
4 4 8
1 1 1 1
1 3 1 3
2 2 2 2
2 4 2 4
3 1 3 1
3 3 3 3
4 2 4 2
4 4 4 4
1 2 0
```

Sample Output

```
1
0
2
```

Explanation

The first sample input shows that the groups of unlit cells may overlap. The solution is to flip just one cell, namely $(4,2)$, so it becomes just like the grid shown in the images above.

Problem E
Expression
Time Limit: 4 seconds

Here's a cool puzzle: Can you write a mathematical expression that evaluates to 113, but only contains common mathematical symbols and four instances of the digit 4?

One possible answer is the following:

$$113 = \frac{\sqrt{4} + (\sqrt{4} + 4!)\%}{\sqrt{4}\%}$$

Here, ! is the factorial symbol, and % is the percentage symbol, e.g., $11\% = 0.11$.

Four fours is an arithmetic puzzle/exercise where the goal is to produce a given number using only four fours and common mathematical symbols. For some numbers, this is easy, e.g. 16 can be written as $4+4+4+4$. But for others, it isn't always so, as you may have seen from the example 113 above. As another example, can you produce 2016 using only four fours?

In this problem, you will solve a variation of this puzzle. Given x and y, find a valid mathematical expression that evaluates to y, subject to the following constraints:

- The only constant appearing in the expression must be the integer x.
- The only operations allowed are addition, subtraction, multiplication and division.
- The number of operations must be at most 28.
- Intermediate results cannot exceed 10^{18} in absolute value.
- Division must be valid and must always yield integers.

If no such expression exists, output `IMPOSSIBLE`.

Input

The first line of input will contain an integer T, the number of test cases.

The first and only line of each test case will contain two space-separated integers, x and y, as discussed in the problem statement.

Output

For each test case output a single line containing a single string.

- If no such expression exists, output the string `IMPOSSIBLE`.
- Otherwise, output a string containing only the characters `x`, `+` `-`, `*`, or `/`. This represents your expression written in *postfix notation*.

An expression in postfix notation is evaluated with a stack, initially empty. Then, for each letter from left to right:

- If it is `x`, then the number x is pushed onto the stack.
- If it is `+`, then two numbers b and a (in that order) are popped from the stack, and then $a+b$ is pushed onto the stack.
- If it is `-`, then two numbers b and a (in that order) are popped from the stack, and then $a-b$ is pushed onto the stack.
- If it is `*`, then two numbers b and a (in that order) are popped from the stack, and then $a \times b$ is pushed onto the stack.
- If it is `/`, then two numbers b and a (in that order) are popped from the stack, and then a/b is pushed onto the stack.

At the end of this process, the stack must contain exactly one number, which we define as the result of the evaluation. If the stack doesn't contain exactly one number, or if an empty stack is popped, then the evaluation fails.

There may be multiple correct answers; any one will be accepted.

Constraints

$1 \leq T \leq 212121$
$0 \leq x, y < 1212$

Sample Input	Sample Output
4 4 32 3 21 3 21 3 21	xx*xx*+ xxxx+*+ xxxx+*+ xxx**x-x-

Problem F
Frog Pushers
Time Limit: 5 seconds

Alpha Task Force X has been assigned to a very special mission. It is to eradicate all frogs and frog pushers in their country within the next three to six months. Frogs are dangerous as they attack communication channels between people. Once a frog attacks a communication channel, the channel is disabled and is rendered unusable.

The task force consists of N people. There exists C channels of communication among them. Any two people can only send and receive messages to and from each other if there still exists an active communication channel between them.

A pair of people may initially have more than one channel of communication between them. However, communicating through a channel has a certain risk involved. The channel might be not secure enough for hackers, or might be physically tampered with, thereby ruining the communication. This risk is quantified by the *risk index (RI)* and each communication channel has a given RI value.

Occassionally, there are important messages that need to be sent to the whole group. The message originates from one member of the task force and he sends it to some other members. These members then send it to other members, and so on. Note that a member may choose not to send the message to others especially if there is a less risky way to reach that person. But in the end, every member must receive the message. The *total risk index (TRI)* of sending such a group message is computed by getting the sum of all RI of the communication channels used to send the message.

Naturally, the task force wants the lowest possible TRI, which we refer to as the LTRI, whenever it sends a group message. It can be shown that this LTRI does not depend on the original sender of the message. Moreover, the task is made even harder by the frogs attacking and disabling the channels. Luckily, the intelligence squad of Alpha Task Force X has found out the order by which the frogs will be attack the communication channels.

Help Alpha Task Force X eradicate all frogs and frog pushers by creating communication plans for the coming months. Figure out the LTRI of the remaining active communication channels before each frog attack!

Input

The first line of input contains a single integer T, the number of test cases.

The first line of each test case consists of two integers N and C, indicating how many members are there in the task force and exactly how many communication channels are originally active.

The next C lines of input describes the original communication channels, in the order which the frogs will attack them. Each line consists of three integers — a, b, and r

— indicating a communication channel between task force member a and task force member b with risk index r.

Output

For each test case, output a single line containing the string `Case #X:`, where X is the case number.

Then output C lines, the i^{th} of which contains a single integer containing the LTRI immediately before the i^{th} communication channel is attacked by frogs. If it is impossible to send a group message due to the lack of communication channels, output `FAIL` for that line instead.

Constraints

$1 \leq N \leq 100$
$1 \leq C \leq 10^5$
$1 \leq T \leq 1000$, but the sum of the Cs is $\leq 2 \cdot 10^5$
$1 \leq a, b \leq N$
$1 \leq r \leq 10^9$

Sample Input	Sample Output
3 2 4 1 2 50 1 2 60 1 2 80 1 2 70 3 2 1 2 90 1 3 600 3 5 1 2 4000 1 2 2000 1 3 6000 2 3 9000 3 2 11000	Case #1: 50 60 70 70 Case #2: 690 FAIL Case #3: 8000 8000 15000 FAIL FAIL

Problem G
Go Go Go Special Action Force!
Time Limit: 3 seconds

You currently lead a special action force composed of 81 highly trained personnel. The force advances towards enemy forces using a 9×9 formation. Depending on the attack strategy, the force will break into 9 companies. Each company will be composed of 9 personnel having different positions. The positions in each company are as follows:

- Officer (`O`)
- Warrant Officer (`WO`)
- Weapons Sergeant (`WS`)
- Engineer Sergeant (`ES`)
- Medical Sergeant (`MS`)
- Communication Sergeant (`CS`)
- Intelligence Sergeant (`IS`)
- Operations Sergeant (`OS`)
- Senior Sergeant (`SS`)

There are 3 attack formations, namely *infiltrate*, *flanking maneuver* and *deploy*. In *infiltrate*, the personnel in front take the lead in infiltrating the enemy. This will result in 9 companies advancing in single file, as illustrated in Figure 1, towards the enemy to set up 9 strategic posts behind enemy lines.

In *flanking maneuver*, the alternate personnel on both sides of the formation takes lead to form 9 companies to attack the enemy from the sides. The left and right flanks will always have 4 and 5 companies respectively as shown in Figure 2.

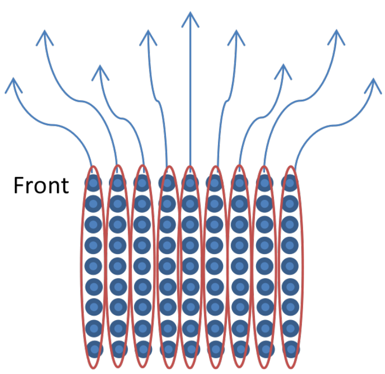


Figure 1: Infiltrate

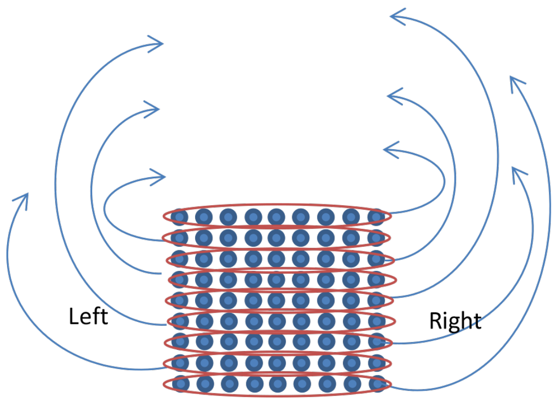


Figure 2: Flanking Maneuver

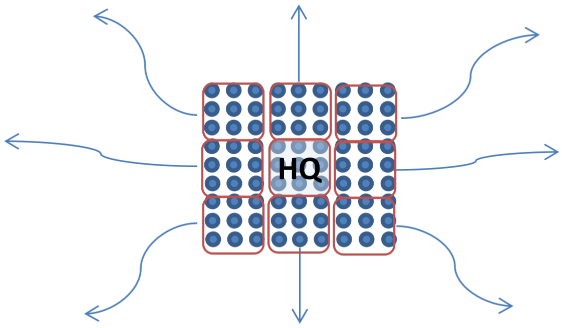


Figure 3: Deploy

Finally, in deploy, the force breaks out into 9 company formations of 3×3, with the center company forming the headquarters and the 8 companies forming satellite posts. Figure 3 shows that 9 companies is formed with the company at the center keeping their position.

Input

The first line of input contains a single integer G, the number of forces.

Each force is composed of 9 lines representing the force formation. Each line is composed of 9 personnel represented by their 1- or 2-letter positions delimited by single spaces.

Output

For each force, output a single line containing the string "`all 3`" if the formation allows all 3 attack strategies, or "`not`" if not. (Both without quotes.)

Constraints

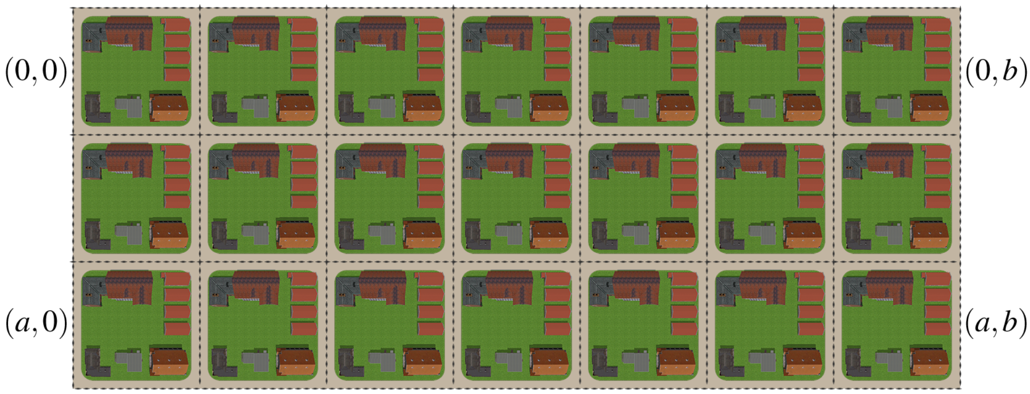
$1 \leq G \leq 350$
Each string in the input is one of the (1- or 2-letter) position codes above.

Sample Input	Sample Output
2 OS MS IS O WS CS SS ES WO SS WO CS OS ES MS O IS WS O ES WS SS IS WO MS OS CS IS O WO MS CS ES OS WS SS CS WS OS IS WO SS ES O MS MS SS ES WS OS O WO CS IS ES IS SS WO O WS CS MS OS WS CS MS ES SS OS IS WO O WO OS O CS MS IS WS SS ES CS OS ES MS IS O WS WO SS MS SS WS WO ES OS CS IS O IS O WO SS WS MS OS ES MS O WO OS WS CS IS MS SS ES ES CS SS O OS MS IS WS WO WS MS IS ES SS WO O OS CS SS IS O CS WO WS ES MS OS WO WS MS OS O ES SS CS IS OS ES CS IS MS SS WO O WS	all 3 not

Problem H
Handbags
Time Limit: 11 seconds

Handbags are a very popular merchandise across the city of Manhattanila. This is because they are manufactured there.

The city of Manhattanila is composed of $(a+1)(b+1)$ villages (locally called *barangays*) neatly arranged in an $(a+1)\times(b+1)$ rectangular lattice. A village is identified by two integers (x,y), $0 \le x \le a$ and $0 \le y \le b$, denoting its row and column number (both starting from 0).



Some s of the villages *manufacture* these handbags (let's call these villages *sources*), while the remaining ones take their supply from one of their *neighboring* villages. Two villages are *neighbors* if they share boundaries, so each village has up to four neighboring villages. (One for each cardinal direction.) Note that these neighboring villages are not necessarily sources, so they may also take their handbag supply from another neighboring village, and so on.

The price of a handbag varies across multiple villages. A source village always sells their handbags at a fixed price. However, for a non-source village, the price depends on the price of the handbag from the village they're supplying from. Specifically, if they get their handbags from a neighboring village which sells them at p pesos, then they will sell it at $p+1$ pesos. Note that their neighbors' prices are not necessarily the same, so naturally they only take their supply from the neighbor which sells them the *cheapest*.

Also, a source village *never* takes their supply of handbags from a neighboring village, even if they might sell for less. They're loyal to their own, after all!

It can be seen that all these rules uniquely determine the price of the handbag for all villages.

A tourist visits Manhattanila, hoping to buy some souvenirs (or *pasalubong*). To impress her friends, she intends to buy handbags and tell her friends that she bought it from the village where it is *most expensive*, while secretly buying it from the village where it is cheapest!

Please help her with her plan by answering this question: what is the most expensive price for this handbag, and how many villages sell it for this price?

Input

The first line of input contains T, the number of test cases.

The first line of each test case contains three integers a, b and s. Here s is the number of sources. Each of the next s lines contains three integers x, y and p, denoting that (x,y) is a source village and p is the price it sells its handbags. No (x,y) pair will appear more than once in a test case.

Output

For each test case, output a single line containing two integers separated by a space:

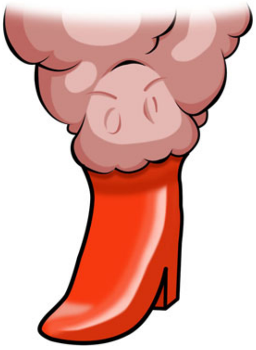
- the most expensive price for the handbag, and
- the number of villages that sell it for this price.

Constraints

$1 \le T \le 50000$
$0 \le a,b \le 10^6$
$0 \le x \le a$
$0 \le y \le b$
$1 \le s \le 30$
$1 \le p \le 10^9$
The sum of the s is ≤ 50000

Sample Input	Sample Output
3 5 6 2 0 0 5 5 6 1 10 10 4 0 10 5 1 6 8 2 3 2 5 10 5 10 20 2 0 0 5 10 20 16	8 9 13 3 25 21

Problem I
Imelda's Shopping Spree
Time Limit: 5 seconds



Shoe Wariwap is the most prestigious shoe boutique in the world. Everyday, the company showcases N of the most luxurious pairs of shoes ever created. These pairs of shoes are showcased by putting each of them in its own pedestal and arranging these pedestals in a row. For ease of notation, we index each pedestal from 1 to N, where Pedestal 1 is the leftmost pedestal.

By the time Shoe Wariwap opens, the pair originally on pedestal i is worth p_i pesos. However, within the day, the store manager might choose to increase the price of the shoes or shuffle the order in which they are displayed, hoping to get the less-appreciated ones more exposure. In particular, at any point during the day, the manager can choose a set of pedestals

$$S_{i,j} := \{i, i+1, \ldots, j-1, j\}$$

and do one of the following:

1. **INC**REASE the price of the shoes which stand on the chosen pedestals. That is, the shoes on pedestal i to j will increase their price by k pesos. We emphasize that the price is associated to the *shoe on the pedestal* and not on the pedestal itself.
2. **REV**ERSE the order of the shoes on the chosen pedestals. That is, the shoe which previously was on pedestal i will be moved to pedestal j, the shoe which was on pedestal $i+1$ will be moved to pedestal $j-1$, and so on.

Today is actually the birthday of one of Shoe Wariwap's best customers – Imelda. If it were up to her, she would buy all N pairs of shoes. However, her loving husband, Marshall, tells her that they are under a lot of debt and so they agree on a system to decide which pairs of shoes she will buy.

The system goes as follows. Imelda picks two integers a and b such that $a \leq b$. This means that Imelda's set of chosen shoes are the shoes which currently stand on Pedestals a through b. However, Marshall has an additional condition, which the couple fondly calls and remembers as Marshall's Law. Marshall's Law dictates that Imelda's chosen shoes, viewed from left to right, must be increasing in price. That means that if $m < n$, then the shoe on pedestal n must be more expensive than the shoe on pedestal m. If

Imelda's chosen shoes do not satisfy Marshall's Law, then the couple goes home buying nothing and Marshall vows that they will not return to Shoe Wariwap.

In spite of this condition, Imelda is still excited about her big shopping day. She thinks about all the possibilities. In particular, every time the manager changes something, she wants to know the number of ways she can fill her shopping bag such that she follows Marshall's Law. Take note that Imelda would never want to go home without purchasing anything for her birthday and so we assume she buys at least one pair of shoes. For example, consider the case when there are $N = 5$ shoes, whose prices are as follows:

Shoe A Php 69000	Shoe B Php 1000	Shoe C Php 1000	Shoe D Php 2000	Shoe E Php 3000
Pedestal 1	Pedestal 2	Pedestal 3	Pedestal 4	Pedestal 5

Here are all the different possibilities on what might Imelda's shopping bag contain.

- There are five possible instances where Imelda's shopping bag will contain exactly one shoe. Indeed, in this case Imelda chooses $a = b$. And there are five cases where this is true.
- There are two possible instances where Imelda's shopping bag will contain exactly two shoes. She could choose $a = 3, b = 4$ (which corresponds to buying Shoes C and D) or choose $a = 4, b = 5$ (which corresponds to buying Shoes D and E).
- There is only one possible instance where Imelda's shopping bag will contain exactly three shoes. This is when she chooses $a = 3, b = 5$ (which corresponds to buying Shoes C, D, and E).

It can be shown that all the other combinations of shoes will break Marshall's Law. Hence, there are 8 different possibilities.

You, as her loyal servant, are tasked to answer her question at any point during the day.

Input

The first line of input contains a single integer T, the number of test cases.

The first line of each test case contains two space-separated integers N, the number of shoes, and Q. The next line contains N space-separated integers denoting the initial prices of the shoes: $p_1, p_2, \ldots, p_N$.

The next Q lines will describe what the manager changes. Each of these lines will be one of the following formats:

- `INC i j k`
  This describes an **INC**REASE operation, done by the manager, as discussed in the problem statement.
- `REV i j`
  This describes a **REV**ERSE operation, done by the manager, as discussed in the

problem statement.

Output

Every time the manager makes a change (price or order), output a line containing a single integer X such that X is the number of ways Imelda can fill her shopping bag (right after the manager makes that change) such that she follows Marshall's Law.

Constraints

$1 \leq T \leq 2$
$1 \leq N \leq 10^5$
$1 \leq Q \leq 10^5$
$1 \leq i \leq j \leq N$
$1 \leq k, p_i \leq 10^9$

Sample Input	Sample Output
2 5 1 69000 500 500 1500 3000 INC 2 4 500 4 4 3333 1111 4444 5555 REV 1 2 REV 1 3 INC 3 4 2222 INC 3 4 1111	8 10 5 5 7

Problem J

Jack and Jill and Joe

Time Limit: 3 seconds

Jill and Jack are about to purchase separate plots of land. Jill would like her plot to be a square one while Jack's is a rectangular plot with one side longer than the other by whatever the sidelength of Jill's plot is.

Joe, their father, insisted that the shorter sidelengths of their plot cannot exceed N feet and that the square of the difference of the area of Jack and Jill's plot is at most 900 square feet. Furthermore the sum of the areas should be the maximum possible.

Joe insists that the sidelengths of the plots are at least 100 feet and are integers.

Input

The input starts with a positive integer T, followed by T test cases.

Each test case consists of one positive integer N on a line by itself.

Output

For each case, output a single line. If the case is not solvable, output `impossible`, else output a and b, separated by a space, where a is the sidelength of Jill's square plot and b is the shorter sidelength of Jack's rectangular plot.

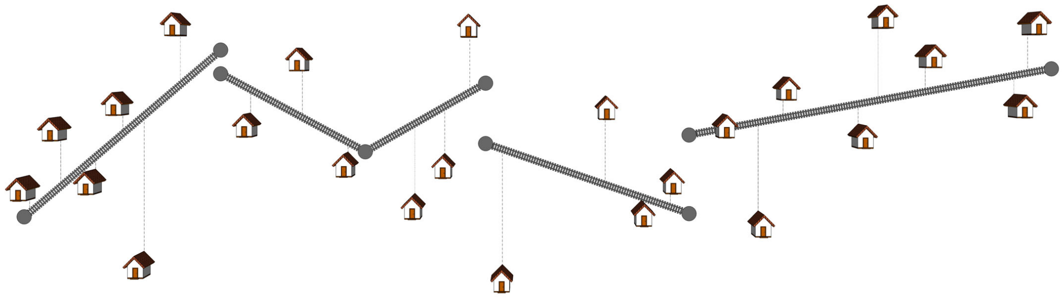
Constraints

$1 \leq T \leq 12000$
$100 < N < 2^{31}$

Sample Input	Sample Output
2 333 900	322 199 898 555

Problem K
Off the Rails
Time Limit: 5 seconds

The Country of Everlasting is planning to build a rail line that will connect its cities. During their planning stage, they have decided that the route where the rail line will pass should minimize the distance between the cities and the rail line. During the canvassing of materials, the engineers involved found out that it will be best if they buy pre-fabricated guideways from the country of Forever. However, the only available pre-fabricated guideways are the straight guideways. If the selected route is not a straight line (for example, the figure below), then they will need to buy several pre-fabricated guideways. They can buy guideways of different lengths.



The problem now is that there is an overhead cost C for each pre-fabricated guideway imported from Forever. So they have to design the route such that the value $a + bC$ is minimized, where:

- a is the sum of the squares of the lengths of the *vertical* segments from each city to the rail line.
- b is the number of pre-fabricated lines.

Also, remember the following:

- The sequence of segments need not be connected.
- No pre-fabricated line must be positioned vertically.
- No vertical line intersects two pre-fabricated lines at different points in their interior.

Input

The first line of input contains T, the number of test cases.

The first line of each test case contains two numbers separated by a space: the integer n and the real number C. The next n lines describe the cities.

The i^{th} subsequent line contains two integers x_i and y_i denoting the coordinates of the i^{th} city.

Output

For each test case, output a single line containing a single real number: the optimal value of $a+bC$. Your solution must be accurate to within an absolute error of 10^{-2} from the correct answer.

Constraints

$1 \leq T \leq 200$
$1 \leq n \leq 1000$
$-1000 \leq x_i, y_i \leq 1000$
$0 < C \leq 10000.000$
C will be given with at most 3 decimal places
$x_1 < x_2 < x_3 < \ldots < x_n$

Sample Input	Sample Output
2 10 1.0 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10 8 2.0 1 1 2 2 3 1 4 2 5 1000 6 999 7 1000 8 999	1.0000 5.6000

Problem L
LoL Tournament
Time Limit: 5 seconds

There are n contestants in this year's LoL tournament. They are numbered from 1 to n. You decided to join since you are very good at Dota and you laugh a lot.

The format of the LoL tournament is very weird. Instead of common formats like single-elimination, round robin, etc., they have their own unique system of determining the winner.

The tournament consists of $n-1$ rounds, and the winner of the last round is the overall champion. Here is how it goes:

1. In the 1^{st} round, the contestants numbered a_1 and b_1 will compete. The loser is eliminated, and the winner advances and is renumbered $n+1$.
2. In the 2^{nd} round, the contestants numbered a_2 and b_2 will compete. The loser is eliminated, and the winner advances and is renumbered $n+2$.
3. In the 3^{rd} round, the contestants numbered a_3 and b_3 will compete. The loser is eliminated, and the winner advances and is renumbered $n+3$.
4. In the 4^{th} round, the contestants numbered a_4 and b_4 will compete. The loser is eliminated, and the winner advances and is renumbered $n+4$.
5. etc.

After $n-1$ rounds, only one contestant will remain, and he/she will be numbered $2n-1$. He/she is the winner of the tournament.

It's not so hard to see why this is unfair! But, despite the apparent unfairness of the system, the contest is still well-formed, i.e., a contestant wins if he/she wins all his/her rounds, and everyone else loses to exactly one other contestant.

Given this, you decided to game the system. You would like to choose your initial number (from 1 to n) such that your probability of winning the tournament is maximized. Which number should you initially be? There may be multiple such initial numbers, so you have to output all of them in increasing order.

Your probability of winning a single round is p, regardless of your opponent.

Input

The first line of input contains T, the number of test cases.

The first line of each test case contains n and p separated by a space. The next $n-1$ lines describe the tournament. The i^{th} subsequent line contains two integers a_i and b_i separated by a space.

Output

For each test case, output a single line containing a list of integers: the list of initial numbers that maximize your chances of winning, in increasing order, separated by single spaces.

Constraints

$1 \le T \le 5$
$1 \le n \le 10^5$
The list $[a_1, a_2, \ldots, a_{n-1}, b_1, b_2, \ldots, b_{n-1}]$ contains no duplicates.
$1 \le a_i, b_i < n + i$
$0 \le p \le 1$
p is written with 7 digits after the decimal point.

Sample Input	Sample Output
1 5 0.9111111 2 3 1 6 5 4 8 7	1 4 5