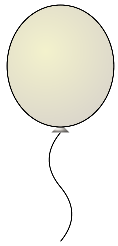
ICPC SouthWestern Europe Regional Contest 2024

Lisbon, Lyon, Pisa, Dec 1st, 2024

Judges and Problem Setters

- Guillaume Aubian (Université Paris-Cité)
- Alessandro Bortolin (University of Milan)
- Thomas Deniau
- Raphaël Marinier (Google), *chief judge*
- Henrique Navas (Técnico, ULisbon)
- Jaime Ramos (Técnico, ULisbon)
- Kevin Pucci (UPorto)
- Cheng Zhong (Google), *deputy chief judge*

This problem set consists of 13 problems, on 33 pages.



A: Titanomachy

Time limit: 6 seconds

In ancient Greek mythology, it is believed that the Olympian gods acquired their power from a long lasting war with the Titans.

After some recent excavations, an archaeological team found manuscripts explaining how people thought this war happened.

Zeus, the leader of the Olympian gods, and Kronos, the leader of the titans, each picked their best N soldiers in order to fight.

Then, the soldiers arranged themselves in two rows, one row containing only the Olympian gods, and the other containing the Titans, such that each soldier faced exactly one opponent.

At the beginning of the battle, each such pair i ($1 \leqslant i \leqslant N$) of soldiers had a score, A_i, which is an integer representing which side of the war had the advantage. A positive number represented an advantage for Zeus and the Olympian gods, a negative number an advantage for the Titans.

During the war, Q events occurred. These events could be of two kinds: either the relative strength of all pairs of soldiers changed due to the actions of Zeus and Kronos (for example, breaking down mountains with lightning), or Zeus conducted a strategic assessment of his army's advantage.

In the first case, the same integer X got added to every single A_i. If X was greater than 0, it meant that Zeus increased the strength of his army, while a negative value meant that the Titans increased their strength, and a value of 0 meant that nothing changed.

In the second case, Zeus focused on an interval $[L, R]$ of soldiers ($1 \leqslant L \leqslant R \leqslant N$): he considered every possible subinterval, that is, all l, r such that $L \leqslant l \leqslant r \leqslant R$, and computed the sum $A_l + A_{l+1} + \cdots + A_r$ for each of them. He computed the maximum value M of such sums over all possible subintervals, and declared that $V = \max(M, 0)$ was the strategic assessment of his army's advantage.

Obviously, Zeus's wisdom allowed him to compute such values instantaneously, but unfortunately, he did not record them anywhere.

Your task as a philologist is to provide the values of all the strategic assessments made by Zeus during the war.

Input

The first line of the input contains two space-separated integers N and Q. The second line contains N space-separated integers A_i, representing the state at the beginning of the war. The following Q lines describe the events of the war, in chronological order. Each of them is either:

- The word STRENGTH, followed by a space and an integer X, meaning that the strength of an army changed.
- The word ASSESS followed by space-separated integers L and R, meaning that Zeus made a strategic assessment of his army's advantage by focusing on the interval $[L, R]$ ($1 \leqslant L \leqslant R \leqslant N$).

Output

For every time Zeus made a strategic assessment of his army's advantage, and in the order they appear in the input, you must output one line with the value V of each assessment.

Limits

- $1 \leqslant N \leqslant 3 \times 10^5$
- $1 \leqslant Q \leqslant 3 \times 10^5$
- $-10^9 \leqslant A_i \leqslant 10^9$ for $i = 1, \ldots, N$.
- All X integers are between -10^9 and 10^9, inclusive.
- At all times during the war, $-2 \times 10^9 \leqslant A_i \leqslant 2 \times 10^9$ for all $i = 1, \ldots, N$ holds true.

Sample Input

```
4 6
1 -2 3 4
ASSESS 1 4
ASSESS 1 2
ASSESS 2 2
STRENGTH 2
ASSESS 1 4
ASSESS 1 2
```

Sample Output

```
7
1
0
14
3
```

Image credits: François Dumont, Titan foudroyé, Louvre Museum, Public domain, via Wikimedia Commons

B: Divine Gifting

Time limit: 5 seconds

A celestial scroll, detailing Zeus's latest whim unfurls before Hermes: the next few millenia will be a period of divine gifting for mortals. Hermes, the messenger god, is tasked with delivering these gifts. Not just any gifts, mind you, but exquisitely crafted items from the Olympian workshops: a lyre that plays melodies of pure joy, a quill that writes words of profound wisdom, and so on. Each of the N gifts is unique, and, to complicate matters, each has an optimal delivery date, a day when its magic would be most potent. But a divine law forbids delivering gifts before their optimal delivery day, lest mortals become complacent and entitled. Of course, all gifts must be delivered.

Adding to the challenge, Hermes, despite being the fastest god on Olympus, is always extremely busy. Between managing the celestial postal service and refereeing chariot races, he knows that he can only dedicate at most K days for the deliveries (each of those days, Hermes can deliver any number of gifts). Furthermore, late deliveries incur penalties: for each gift, the penalty is the square of the difference between its actual delivery day and its optimal delivery day. If a lyre is delivered a day or two late, a village might experience a few hours of slightly off-key music. A minor inconvenience, to be sure. However, if the lyre is delivered a month or a year late, the consequences are far more dire: a full year of discordant melodies, enough to drive even the most stoic musician to madness. The potential for chaos is immense.

And this is where you come in, mortal friend. Hermes, with his myriad responsibilities, could use a helping hand. Can you help him plan his divine calendar by determining the best days for delivering the gifts, so as to minimize the sum of late delivery penalties?

Input

On the first line of the input, two space-separated integers:

- N, the number of gifts;
- K, the maximum number of days Hermes dedicates to gifting.

On the second line, N space-separated integers d_i, representing the optimal delivery date of each gift.

Output

A single line of space-separated integers, where the i-th element is the day when Hermes should deliver the i-th gift. If there are multiple optimal solutions, they will all be accepted.

Limits

- $1 \leqslant N \leqslant 5,000$
- $1 \leqslant K \leqslant 20$
- $0 \leqslant d_i \leqslant 1,000,000$ for all $i \leqslant N$

Sample Input

```
5 2
50 0 51 10 50
```

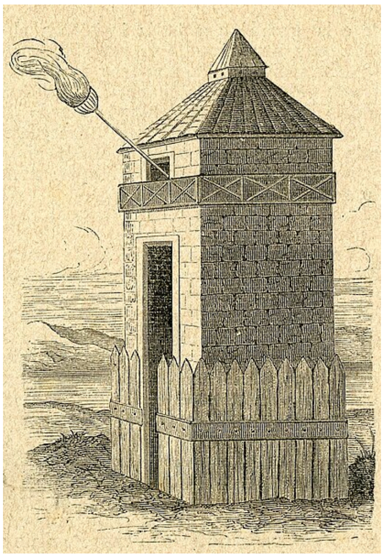
Sample Output

```
51 10 51 10 51
```

Image credits: Lawrence Alma-Tadema, Public domain, via Wikimedia Commons

C: Phryctoria

Time limit: 1 second



Lusius Quietus, one of the generals of Trajan, the Roman emperor, is sending signals to his army using signal towers like the one above, where the fires you lit can be seen from afar.

He wants to send the string S describing military manoeuvres. However, sending long messages via fire signals is time-consuming, so Lusius decides to use a shorthand system. To abbreviate the message, Lusius can replace any substring of S with the special character '*'. For example, if S is "swerc", then some possible abbreviations are: "swe*", "*sw*r*", "swerc", "*", etc. Note that '*' can match the whole string or no character at all, so "*sw*r*" is indeed considered a valid abbreviation of "swerc" even though it's longer (Lusius is not always clever).

However, there is a problem: the recipient might misinterpret Lusius's message. If the abbreviation he sends is also a valid abbreviation of the string T, which is a word used to signal retreat, it could lead to losing the war.

Help Lusius find the length of the shortest possible abbreviation of S that cannot be interpreted as an abbreviation of T.

Input

The first line contains two integers N and M, the lengths of the strings S and T, respectively. The second line contains the string S. The third line contains the string T.

Output

The length of the shortest abbreviation of S.

Limits

- $1 \leqslant N \leqslant 500$.
- $1 \leqslant M \leqslant 500$.
- $S \neq T$.
- S and T contain only lowercase English letters.

Sample Input

```
5 5
swerc
seerc
```

Sample Output

```
3
```

Sample Explanation

A possible solution is “sw⋆”.

Image credits: Roman signalling tower, from a bas-relief of the Trajan column, public domain. (from L’illustrazione popolare, Fratelli Treves Editori - Milano, 1895.), via Wikimedia Commons.

D: Temple Architecture

Time limit: 2 seconds

You are excavating what remains of a very old Indian temple. The architecture of the temple is very curious: you found a line of N towers of different heights (no two towers have the same height), all spaced by one meter (the radius of each tower is supposed to be negligible).

During the excavations, you find something which might explain this curious architecture: the tomb of the architect. You find the following epitaph on the tomb:

O thou temple-builder,
To achieve perfection, visit each tower;
For each, compute the distance to the closest tower that is taller[1]*;*
Add every such distance.
If thou can follow this guidance,
Enlightened shalt thou be by this result,
And great in thy temple shall be the cult.

[1] The closest taller tower can be on the left or on the right. For the tallest tower, this distance is undefined and should not be considered in the final sum.

You wonder how to compute this sum, this "enlightenment score" of the temple.

You are given a positive integer N corresponding to the number of towers, and an array H of N distinct non-negative integers corresponding to the heights of the towers. H_0 is the height of the leftmost tower, H_1 is the height of the tower to the right of H_0, and so on. Finally, H_{N-1} is the height of the rightmost tower. Observe that the distance between any two towers, in meters, is the absolute value of the difference of their respective indices in the array H. Let p denote the index of the tallest tower in H and define, for every $i \neq p$,

$$d(i) = \min\{|i-j|\colon \text{for every } j \text{ such that } H_j > H_i\}.$$

Note that $d(p)$ is not defined. The "enlightenment score" of the temple is then given by

$$\sum_{i=0, i\neq p}^{N-1} d(i).$$

Input

The first line contains the integer N. The second line contains the list of elements $H_0, \ldots, H_{N-1}$ of the array H of heights, separated by spaces.

Output

The output should contain one line with the enlightenment score of the temple.

Limits

- $2 \leqslant N \leqslant 200\,000$;
- $0 \leqslant H_i \leqslant 10^{18}$, for $i = 0, \ldots, N-1$;

Sample Input 1

```
5
7 3 2 100 1
```

Sample Output 1

```
6
```

Sample Explanation 1

- Distance of the 1st tower to the closest taller tower (4th tower): 3
- Distance of the 2nd tower to the closest taller tower (1st tower): 1
- Distance of the 3rd tower to the closest taller tower (2nd or 4th tower): 1
- Distance of the 5th tower to the closest taller tower (4th tower): 1

The tallest tower (the 4th) is not considered. Hence, 3+1+1+1=6.

Sample Input 2

```
8
45 13 18 10 8 56 17 19
```

Sample Output 2

```
13
```

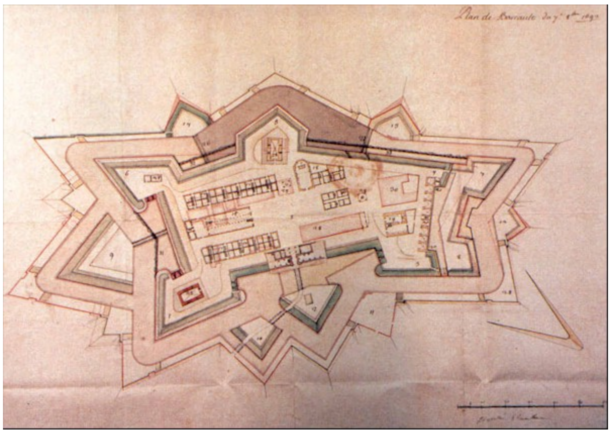
Sample Explanation 2

- Distance of the 1st tower to the closest taller tower (6th tower): 5
- Distance of the 2nd tower to the closest taller tower (1st or 3rd tower): 1
- Distance of the 3rd tower to the closest taller tower (1st tower): 2
- Distance of the 4th tower to the closest taller tower (3rd tower): 1
- Distance of the 5th tower to the closest taller tower (4th or 6th tower): 1
- Distance of the 7th tower to the closest taller tower (6th or 8th tower): 1
- Distance of the 8th tower to the closest taller tower (6th tower): 2

The tallest tower (the 6th) is not considered. Hence, 5+1+2+1+1+1+2=13.

E: Building the Fort

Time limit: 0.5 second



You are a Roman general setting up a defence fort against the Barbarians.

You only know how to build straight walls, so your fort is going to be polygon-shaped.

Due to the shape of the land, which has multiple hills located at all the integer coordinates, you know that the enemy artillery can only be installed in very specific ways, and that higher positions inside your fort are more vulnerable, which means that:

- all the vertices of the polygon must have integer coordinates between 1 and 10^9 (inclusive);
- N known points $(x_i, y_i), i = 1, \ldots, N$, with integer coordinates between 1 and 10^9 (inclusive), initially given, must be among the vertices of the polygon;
- no point with integer coordinates can be located strictly inside the polygon, as it would be vulnerable to the enemy artillery otherwise;
- the polygon has to be simple[1] (obviously, the fort needs to be closed, and you don't know how to build intersecting walls).

In addition, due to the cost of building this fort and the limited materials, you can only afford to build a polygon with a number of vertices smaller than or equal to $3N$.

It can be shown that such a polygon always exists.

Input

The first line contains the integer N. The next N lines contain two space-separated integers x_i, y_i, the points that must be among the vertices of the polygon.

Output

The first line should contains K, the number of vertices of the polygon.

The next K lines should contain two space-separated integers x'_i, y'_i, the coordinates of the vertices of the polygon. They must be in an order that forms a closed and non-intersecting path that defines the outline of the polygon.

If there are multiple solutions, you can output any of them.

Limits

- $3 \leqslant N \leqslant 1\,000$;

[1]A simple polygon is a polygon formed by a single closed path that does not intersect itself or overlap itself.

- $1 \leqslant x_i \leqslant 10^9$ for $i = 1, \ldots, N$;
- $1 \leqslant y_i \leqslant 10^9$ for $i = 1, \ldots, N$;
- All (x_i, y_i) are unique;
- $1 \leqslant x'_i \leqslant 10^9$ for $i = 1, \ldots, K$;
- $1 \leqslant y'_i \leqslant 10^9$ for $i = 1, \ldots, K$;
- All (x'_i, y'_i) must be unique.

Sample Input

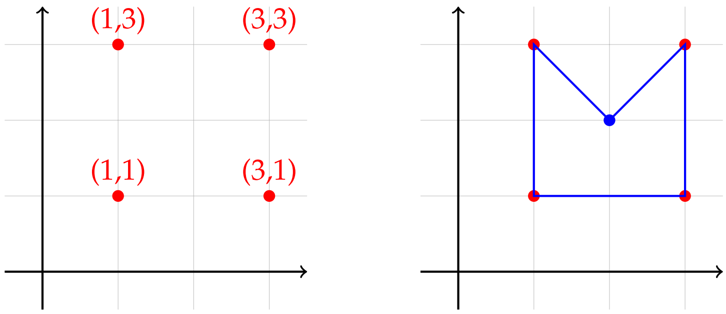
```
4
1 1
1 3
3 1
3 3
```

Sample Output

```
5
1 1
3 1
3 3
2 2
1 3
```

Sample Explanation

The following is a possible solution for the sample input:



On the other hand, the following polygons are **not** valid solutions:

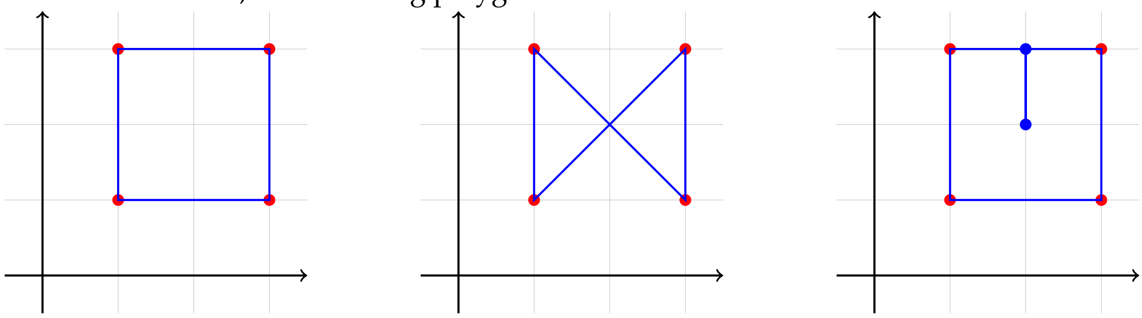


Image credits: Plan de fort Barraux par Vauban, Public domain, via Wikimedia Commons

F: Yaxchilán Maze

Time limit: 5 seconds

Years spent deciphering Mayan glyphs had led Dr Wood, a famous archeologist, to this very moment: inside a chamber of the Yaxchilán maze, in the middle of the Lacandon Jungle, the last undiscovered Mayan codex appeared before her. It was beautiful, protected by jaguar pelt and adorned with jade. However, as soon as she touched it, all the corridors of the maze closed. She was now a prisoner of the maze. Worse, she was not alone, but had a full team of archeologists with her that were also held prisoner in different chambers of the maze.

As she examined the codex, a series of intricate diagrams caught her eye. They depicted the maze, its corridors morphing into different configurations, each aligned with a specific position of the sun. Another set of diagrams showed intricately carved holes in the walls together with a chilling depiction of a giant wasp. A realization dawned on her – the Maya had designed this maze to change with the sun's position, and some chambers were booby-trapped with a giant nest of deadly wasps.

Namely, there are N chambers in the maze numbered $0, \ldots, N-1$. Dr Wood and the members of her team start in chambers $0, 1, \ldots, A-1$ (one person per chamber). There are E exits: chambers $N-E, \ldots, N-1$. Initially, the maze contains no open corridors. At each hour (00:00, 01:00, 02:00 and so on, represented by integers 0, 1, 2 and so on), a new corridor between two chambers opens. This corridor stays open for exactly M hours minus one minute. Corridors are bi-directional.

Among the N chambers, B are booby-trapped. The trap of a chamber triggers as soon as it is connected to at least K other chambers. When triggered, a gigantic swarm of deadly wasps appears in the chamber and immediately spreads to all chambers connected to the booby-trapped chamber. Furthermore, as time progresses, the wasps never disappear from a chamber, and worse, they continue to spread instantaneously from chambers that contain wasps to newly connected chambers. Two chambers are considered connected at a given time when there exists a path of one or multiple open corridors that allows going from one chamber to the other.

Thanks to the codex and to her knowledge of the maze (and smartphones!), Dr Wood and each member of her team have full information on the maze at hour 0, including all future events: in particular, they each know where they are in the maze, where corridors lead to, the exact time, when corridors open and close and which chambers they connect, the location of the booby-traps, exits, K, and can deduce which chambers are or will be filled with deadly wasps. All the archeologists can move freely and independently from each other using the open corridors at all times. They run fast and can move to any reachable chamber in less than 59 minutes. If an archeologist ends up in a chamber filled with deadly wasps, he or she dies and cannot exit the maze.

The exit chambers behave as the other chambers: they can be filled with wasps and be booby-trapped.

As soon as an archeologist reaches any exit chamber not filled with wasps, she or he exits the maze and its dangers. Can you tell the earliest time when each archeologist can exit the maze?

Input

- The first line contains the integer A, the number of archeologists.
- The second line contains the integer N.

- The third line contains the integer M.
- The fourth line contains the integer E.
- The fifth line contains the integer T, the number of hours at which corridors open.
- The sixth line contains B, followed by the list of the B space-separated booby-trapped chambers b_i.
- The seventh line contains K.
- In the next T lines, line t contains two space-separated integers u_t, v_t representing a corridor that opens at hour t and connects chambers u_t and v_t. t goes from 0 to $T-1$ (inclusive). Multiple corridors can connect the same two chambers. A corridor can connect a chamber to itself.

Output

The output should contain A lines. The i-th line represents the i-th archeologist. If the i-th archeologist can exit the maze, it should be the integer representing the earliest hour at which she or he can exit. Otherwise, it should be "IMPOSSIBLE" (without quotes).

Limits

- $1 \leqslant A \leqslant 50$;
- $2 \leqslant N \leqslant 50\,000$;
- $1 \leqslant M \leqslant 100\,000$;
- $1 \leqslant E \leqslant 100$;
- $A + E \leqslant N$ (no chamber is both a starting chamber and an exit chamber);
- $1 \leqslant T \leqslant 500\,000$;
- $0 \leqslant B \leqslant N$;
- $0 \leqslant b_i < N$ for $i = 0, \ldots, B-1$;
- The b_i's are unique;
- $0 \leqslant K < N$;
- $0 \leqslant u_t < N$ for $t = 0, \ldots, T-1$;
- $0 \leqslant v_t < N$ for $t = 0, \ldots, T-1$;

Sample Input 1

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

Sample Output 1

```
3
```

Sample Explanation 1

There are no booby-traps. Dr Wood (the only archeologist) starts at 0 and the exit is 4. After the corridor `0 2` opens, she goes from chamber 0 to chamber 2. She stays there until the corridor `2 4` opens, which allows her to reach chamber 4 and exit at hour 3.

Sample Input 2

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

Sample Output 2

```
IMPOSSIBLE
```

Sample Explanation 2

There are no booby-traps. Dr Wood (the only archeologist) starts at 0 and the exit is 3. Unfortunately, Dr Wood can only reach chamber 2 after the corridor `2 3` leading to the exit closed. She cannot exit the maze.

Sample Input 3

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

Sample Output 3

```
9
9
9
IMPOSSIBLE
```

Sample Explanation 3

The fastest path to an exit for the first archeologist starting at 0 is to go to chamber 5 at hour 6 using corridors `0 6` and `5 6`, then to chamber 7 at hour 7 using the corridor `5 7`, and finally reach chamber 8 (an exit) at hour 9 using the corridor `7 8`. Note that chamber 7 is booby-trapped, but the wasps only appear at hour 10, after the first archeologist exited the maze.

The second archeologist starting at 1 can move to chamber 0 at hour 0 and then follow the path of the first archeologist.

The third archeologist starting at 2 can move to chamber 0 at hour 1 and then follow the first two archeologists.

The last archeologist starting at 3 is killed by the wasps at hour 2.

At the end, chambers 1, 2, 3, 4, 6, 7, 8, 9 are filled with wasps.

G: Guess How the Ballet Will End

Time limit: 0.5 second

In ancient Greek theater, the chorus would often dance.

The dances were a pleasure to watch; all the dancers on stage executed the same movements at the same time, going across the theater, left to right or right to left. Of course, no dancer could ever go past the borders of the scene. So if the instructions would have made them fall off the scene, they would simply have carried out the move until they reached the border and then stayed there until the next move. For instance, if the move said 5 steps to the right but the dancer only had space for two steps, he/she would have taken those two steps and would then have waited on the border for the next move.

You are excavating a theater and have found extraordinarily well-preserved papyri detailing the sequence of moves the dancers would make during many theater plays, a real treasure trove. The initial positions of the dancers is not known to you, since you have only found the papyri with the moves, and the rest of the instructions and the rest of the plays have been lost.

The instructions are a sequence of numbers, telling all the dancers to dance 3 steps to the left, 5 steps to the right, etc., all carrying out the same move at the same time.

Adding a coordinate system to the scene, each dancer i starts at the position (X_i, i) and a move only impacts their X coordinate (each dancer dances on his/her own line), so they cannot bump into each other. They also all move at the same speed; moving by n steps means the same change of X coordinate for all dancers, unless they hit the borders as explained above.

You have figured out that in some cases, because some dancers may need to cut some movements short due to the borders, it may happen that they will all end up in a perfect line (all at the same X position) at the end of the dance. There are so many papyri that you have decided to write a program to find out if this is the case. Maybe you will learn interesting things about Greek dancing, depending on the number of plays where all the dancers end up all aligned.

Of course, you don't know the starting positions of the dancers, nor their number (only that they were at least a dozen), but you think there are some cases where your program will be able to tell for sure that the dancers end up all aligned at the same X position and if so, where.

Input

On the first two lines of the input:

- R, an integer, the length of the scene. The valid X positions of the dancers range from 0 (left of the scene) to R (right of the scene). Going beyond that would make the dancers fall of the scene.

- N, the number of movements the dancers shall carry out.

On the third line, N integers d_i, separated by spaces, the signed lengths of the moves the dancers shall carry out (a positive number means the dancer is moving to the right, a negative number that he/she is moving to the left).

Output

If you can be sure that all the dancers will end up being aligned at the same X position, a single integer, corresponding to the position (distance from the left of the scene) where all the dancers will end up at the end of the moves.

Otherwise, output the string `uncertain` .

Limits

- $1 \leqslant R \leqslant 10,000,000,000$
- $1 \leqslant N \leqslant 1000$
- $-10,000,000,000 \leqslant d_i \leqslant 10,000,000,000$

Sample Input 1

```
100
2
110 -20
```

Sample Output 1

```
80
```

Sample Input 2

```
100
1
50
```

Sample Output 2

```
uncertain
```

Sample Input 3

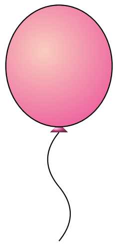
```
100
8
50 30 -40 10 -30 -50 30 -10
```

Sample Output 3

```
20
```

H: The king of SWERC

Time limit: 0.5 second



That's it! After years of research, you finally managed to find who was the king of the infamous South Western Egyptian River City (SWERC)!

Well, not exactly. You have found the actual votes. But considering that the person who got the most votes got elected, and considering that there wasn't a tie for first place, can your code deduce who was the actual king of SWERC?

Input

The first line of the input is an integer, N, the number of votes. The next N lines describe the votes, and each contains a name V_i in capital English letters, without spaces.

It is guaranteed one name appears strictly more than the others.

Output

The output should contain a single string, the name of the king of SWERC.

Limits

- $1 \leqslant N \leqslant 500$.
- $1 \leqslant |V_i| \leqslant 20$, for $i = 1, \ldots, N$.

Sample Input 1

```
1
RAMSES
```

Sample Output 1

```
RAMSES
```

Sample Input 2

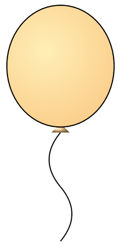
```
4
JON
JOFFREY
TYWIN
JON
```

Sample Output 2

```
JON
```

I: Divination

Time limit: 1 second



In Yinxu, the archaeological site of the late capital of the Shang Dynasty, there are N divination papers written in oracle bone script, numbered $1, 2, ..., N$. Some papers may cite other papers, but no paper can cite itself. Additionally, there are no circular citations, meaning it's not possible to see the following situation: A_1 cites A_2, A_2 cites A_3, ..., A_{K-1} cites A_K, A_K cites A_1 (where $2 \leqslant K \leqslant N$).

As per myth, a complete set of divination papers can predict the wars and peace of the next century, and it should have a complete citation chain, i.e., A_1 cites A_2, A_2 cites A_3, ..., A_{N-1} cites A_N, without any papers missing. Please determine whether these N divination papers constitute a complete set.

Input

The first line contains an integer N, represents the number of papers. Then N lines follow, the i^{th} of them represents the citations of the i^{th} paper: the first integer c_i represents the number of its citations, followed by c_i integers $p_{i,1}, p_{i,2}, ..., p_{i,c_i}$ that represent the papers that it cites.

Output

A single integer, 1 if they constitute a complete set of divination papers, or 0 otherwise.

Limits

- $2 \leqslant N \leqslant 100\,000$;
- $0 \leqslant c_i \leqslant N - 1$ for all $i \leqslant N$;
- $0 \leqslant c_1 + c_2 + ... + c_N \leqslant 500\,000$;
- $1 \leqslant p_{i,j} \leqslant N$ for all $i \leqslant N$ and $j \leqslant c_i$.
- $p_{i,j} \neq i$ for all $i \leqslant N$ and $j \leqslant c_i$.

Sample Input 1

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

Sample Output 1

```
1
```

Sample Explanation 1

In this sample, paper 3 cites paper 2, paper 2 cites paper 4, paper 4 cites paper 1. Thus, we find a complete citation chain, which makes them a complete set of divination papers.

Sample Input 2

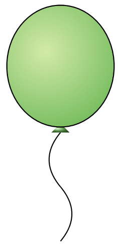
```
4
0
1 1
2 2 4
1 1
```

Sample Output 2

```
0
```

J: Recovering the Tablet

Time limit: 5 seconds



A long time ago, before our modern civilization arose and before any of you were born, was the year -1966 (3961 Before SWERC). During this dark period, there were no streaming services nor any programming contests. Thus, to entertain themselves, humans had rudimentary games that they played on clay tablets.

This year, a mysterious game was created: Kakurus. We know close to nothing about Kakurus (the Internet Archive had not yet been created), except for a few rules described on an artifact you have discovered:

1. The game is played on a $M \times N$ grid;
2. Each cell is either black or white;
3. White cells are originally empty, but you will have to put an integer from 1 to 9 (inclusive) inside each white cell;
4. **Horizontal constraint**: A black cell can contain an integer corresponding to the sum of the consecutive white cells to its right (until the first black cell or the limit of the grid);
5. **Vertical constraint**: A black cell can contain an integer corresponding to the sum of the consecutive white cells below it (until the first black cell or the limit of the grid).

Note that the last two rules are independent from each other: a black cell can have 0, 1 or 2 integers inside it. **Note also that no constraint is placed on repetitions of numbers.** Finally, since we want this problem to be interesting, every white cell is covered by exactly one vertical constraint and by exactly one horizontal constraint.

At the bottom of your artifact lies a grid of Kakurus. It is already filled, but not necessarily with a correct solution – probably a deterioration due to the age of this antique artifact. Can you find a valid solution that is as close as possible to this proposed solution?

If the number you write on a white cell is X and the value of the proposed solution for this cell is T, then the closeness score is $|X - T|$. The final closeness score of the grid is the sum of all closeness scores of the cells. Your objective is to find the minimum closeness score that can be achieved.

Input

The first line of the input consists of three integers, M, N and S, respectively the number of lines and columns of this grid, and the number of sum constraints.

Then, M lines follow. The ith of these lines only contains digits from 0 to 9. The jth character equals 0 if and only if the cell at line i and column j ($1 \leqslant i \leqslant M$, $1 \leqslant j \leqslant N$) is black, and otherwise is equal to the value of this cell (which is thus white) in the proposed solution.

Then S lines follow. Each line is of the form c i j s where c is equal to either `H` or `V`, $1 \leqslant i \leqslant M$, $1 \leqslant j \leqslant N$ and s is an integer between 1 and 135, inclusive. The sum of the consecutive white cells at the right of (when $c =$ `H`) or below (when $c =$ `V`) the cell at line i and column j must be equal to s in your solution.

It is guaranted that every white cell is covered by exactly one vertical and one horizontal constraint.

Output

If the grid has no solution, you must output `IMPOSSIBLE`. Otherwise, your output should consist of the minimum closeness score that can be achieved.

Limits

- $1 \leqslant M \leqslant 16$;
- $1 \leqslant N \leqslant 16$.
- $0 \leqslant S \leqslant 2 \times M \times N$

Sample Input 1

```
4 4 7
0000
0032
0233
0204
H 3 1 9
V 1 3 6
H 2 2 5
V 2 2 5
V 1 4 9
H 4 1 2
H 4 3 4
```

Sample Output 1

```
1
```

Sample Input 2

```
3 4 5
0000
0012
0345
H 2 2 3
H 3 1 12
V 2 2 3
V 1 3 5
V 1 4 6
```

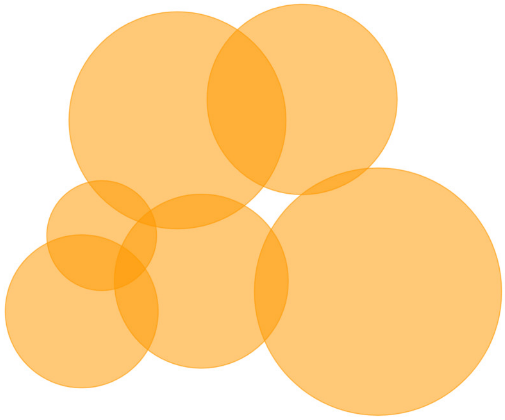
Sample Output 2

```
IMPOSSIBLE
```

Page intentionally left blank

K: Disk Covering

Time limit: 0.5 second



On a vast, flat green meadow, there are several golden disks in the shape of perfect circles from ancient times. According to legend, if one chants a spell, the area covered by the disks will turn into flames, fending off enemy attacks. When the enemy comes, you can hide in a place completely surrounded by disks, yet not on the disks, thus isolated from the outside world by the flames.

Given the positions and sizes of the disks, determine whether such a hiding place exists.

Input

The first line contains an integer N, representing the number of disks. In the following N lines, the i^{th} line contains three integers that describe disk i: the x-coordinate x_i, the y-coordinate y_i of its center, and its radius r_i.

Output

A single integer, 1 if such a place exists, or 0 otherwise.

Limits

- $1 \leqslant N \leqslant 250$;
- $-10^9 \leqslant x_i, y_i \leqslant 10^9$ for all $i \leqslant N$;
- $1 \leqslant r_i \leqslant 10^9$ for all $i \leqslant N$;
- There are no three disks whose circular outlines intersect at a common point;
- Among all intersection points of the circular outlines of any two disks, the distance between any two intersection points is greater than or equal to 1;
- There are no two disks whose circular outlines are tangent to each other (i.e. have exactly one intersection point);
- For two disks whose circular outlines do not intersect, the distance between any point on the circular outline of one disk and any point on the circular outline of the other disk is always greater than or equal to 1.

Sample Input 1

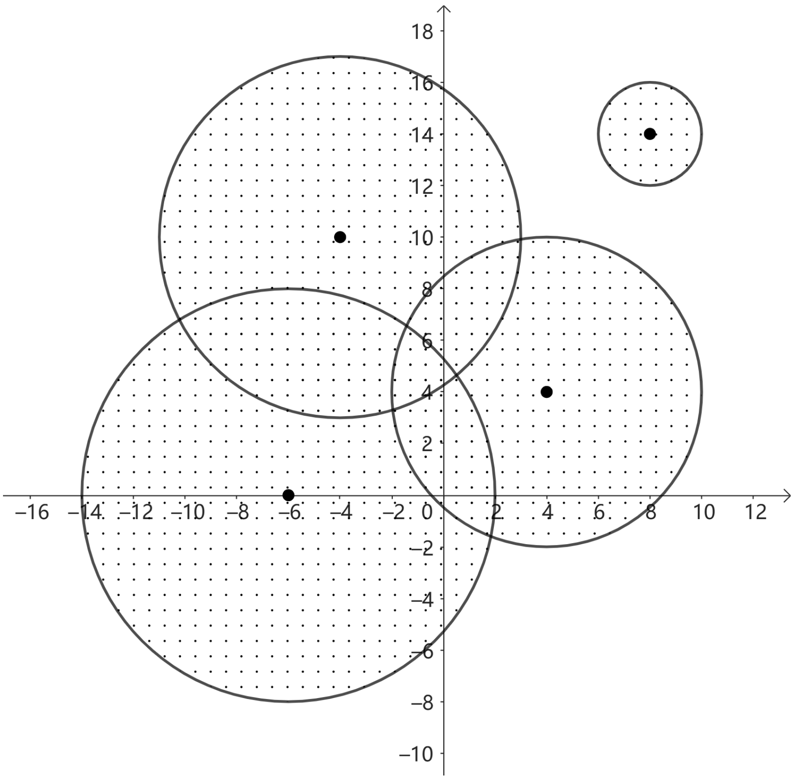
```
4
-6 0 8
-4 10 7
4 4 6
8 14 2
```

Sample Output 1

```
0
```

Sample Explanation 1

In this sample, there isn't any place that is completely surrounded by disks, yet not on the disks.



Sample Input 2

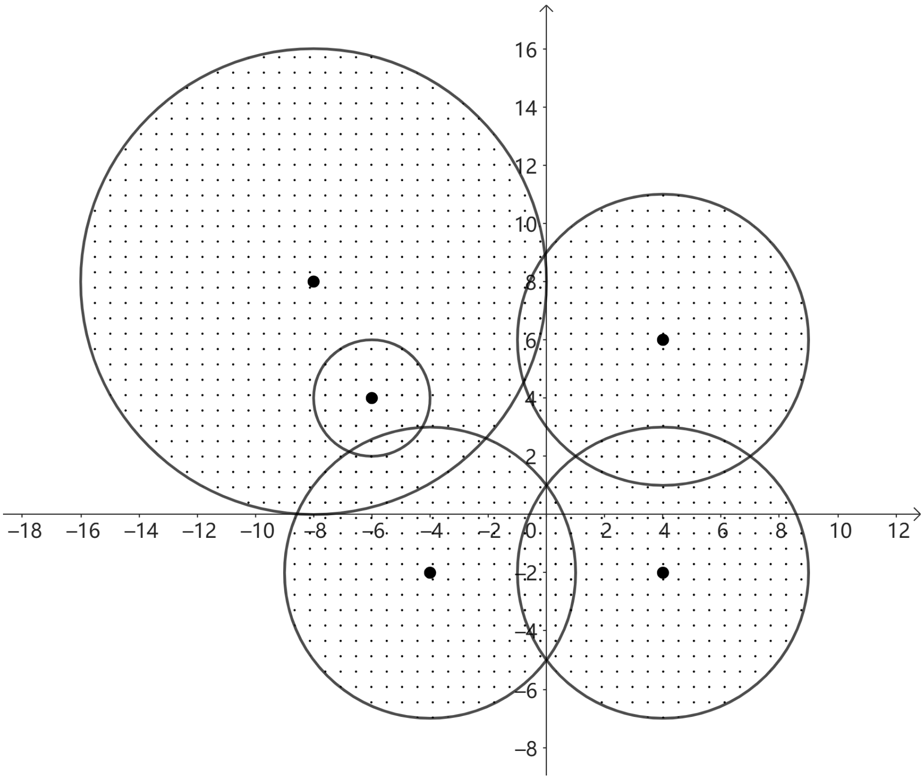
```
5
4 -2 5
-4 -2 5
-8 8 8
4 6 5
-6 4 2
```

Sample Output 2

```
1
```

Sample Explanation 2

In this sample, $(-0.5, 3)$ is one of the places we can hide. It is surrounded by disks, yet not on the disks. Note that although all the inputs are integers, the hiding place does not necessarily have to be an integer point.



Sample Input 3

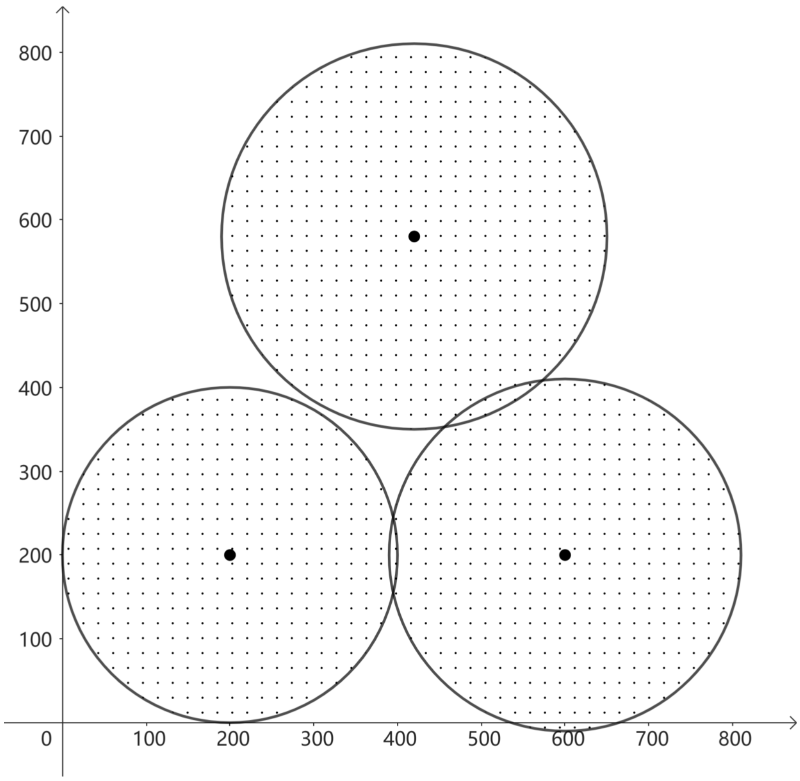
```
3
420 580 230
200 200 200
600 200 210
```

Sample Output 3

```
0
```

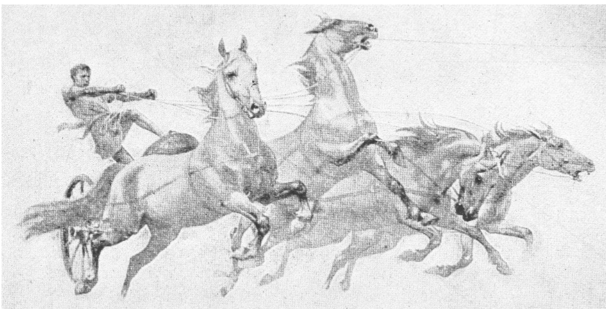
Sample Explanation 3

In this sample, there isn't any place that is completely surrounded by disks, yet not on the disks.


800
700
600
500
400
300
200
100
0
100
200
300
400
500
600
700
800

L: The Charioteer

Time limit: 1 second



As you are studying ancient greek mythology, you stumble upon the legend of Phaethon, son of Helios. After being recognized by Helios as his son, he asks him to drive his chariot. Despite the warnings of Helios that only he can control the horses, Helios obliges, giving Phaethon the control of the chariot.

Tragically, Phaethon loses control of the chariot, which gets too close to the Earth and burns it.

However, you wonder if maybe the story could have ended differently. If Phaethon reaches the temple of Helios, then the chariot can be stopped by Helios himself. For that, you create a model that fits the story.

Greece can be represented as a 2D infinite grid, with Phaethon and its chariot starting at coordinate $(0,0)$, facing the Ox axis in the direction of increasing X. The temple of Helios is placed in an unknown position (X, Y) on this 2D grid. Initially, the velocity V of the chariot is 1.

At each timestep, the following happens in order:

1. Phaethon does one of the following 3 actions: turn the chariot left 90 degrees, continue facing front, or turn the chariot right 90 degrees;
2. The chariot moves V units in the direction it is facing;
3. V is increased by 1;
4. The Oracle tells Phaethon the Manhattan distance between his position and the position of the temple of Helios.

The Manhattan distance between $(X1, Y1)$ and $(X2, Y2)$ is $|X1 - X2| + |Y1 - Y2|$.

If the velocity ever reaches 2×10^4, the chariot gets uncontrolled and too close to the earth. If at the end of a step, the chariot is in (X, Y), the chariot is stopped.

Input and Output

This is an interactive problem. As such, no initial input is provided in this problem, and you get to ask a question first.

At each step, you must print a line to act on the direction of the chariot. This line must follow the description "? c" followed by a newline, where c is either L to rotate counterclockwise, R to rotate clockwise, or F to keep the direction unchanged. If it does not respect this format, you will receive a WRONG-ANSWER verdict.

After sending a line, you must always flush the output. Otherwise, you will get the verdict TIMELIMIT.

The oracle then prints a single integer, that you can read on the standard input, denoting the Manhattan distance between your new position (after moving V units in the specified direction) and the temple. V is increased after updating your position.

If this distance is 0, it means you arrived to the temple of Helios and thus should stop interaction immediately (exiting your program), receiving an ACCEPTED verdict.

Notice that you must be in the temple after your move, just flying above the temple with the chariot does not count.

If your velocity ever reaches 2×10^4, you will receive the verdict WRONG-ANSWER.

Limits

- $|X| \leqslant 10^6$
- $|Y| \leqslant 10^6$
- $|V| \leqslant 2 \times 10^4$

Sample interaction

In this section, to clarify what the input and output should be, “< ” is printed before what your program can read on standard input and “> ” is printed before what your program should output. Do not include these characters in your real input/output.

In this example, the temple of Helios (whose position you must guess) is located at X=1, Y=5. The left column is the I/O. The right column is the state of the chariot after the requested move happened.

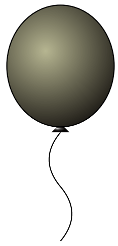
`> ? F`	Chariot moves to (1, 0), orientation = right, $V = 2$
`< 5`	Distance from (1,0) to (1,5) is 5
`> ? L`	Chariot moves to (1, 2), orientation = up, $V = 3$
`< 3`	Distance from (1,2) to (1,5) is 3
`> ? F`	Chariot moves to (1, 5), orientation = up, $V = 4$
`< 0`	Success

An interactive Python script where you can enter commands and see how the judge would respond is also available on the Web version of this problem.

Image credits: Phaëthon Driving the Chariot of Apollo, Max Francis Klepper, Public domain, via Wikimedia Commons.

M: Ook? Ook!

Time limit: 0.5 second



When Arthur Evans excavated Knossos, along with the famous Linear A and Linear B tablets, he also found evidence of a much earlier language, the Ook language, whose only letters are O and K.

As he was deciphering this language, he telegraphed to London lists of Ook words that he was gathering. At the time, the Morse language was different from the one you know, and letters had Pseudo-Morse translations different from the ones they have today. Unfortunately, he was so excited by this discovery that he forgot to include pauses between letters when telegraphing.

As the operator in London, you have received this message from sir Arthur: `.-.-.-`, and through a side-channel, you have learnt this sequence can represent two different Ook words: OK and KO. You have also received the sequence `.-.-.-.-.-`, and have learnt it means OOK.

You need to ask sir Arthur to translate a few more Ook words for you into English. Can your program send them as Pseudo-Morse over the wire? You are as excited as sir Arthur, so you forget pauses in what you send as well.

Input

The input consists of a single line. This line contains a single Ook word W, which only contains letters `O` and `K`, followed by an end-of-line character.

Output

The output should contain a single string, the translation of W in the pseudo-Morse language, without pauses.

Limits

- $1 \leqslant |W| \leqslant 1\,000$.

Sample Input 1

```
OK
```

Sample Output 1

```
.-.-.-
```

Sample Input 2

```
KO
```

Sample Output 2

```
.-.-.-
```

Sample Input 3

```
OOK
```

Sample Output 3

```
.-.-.-.-.-
```

Image: Photo of a Linear A tablet by Olaf Tausch, CC BY 3.0, background removed, via Wikimedia Commons