Bubble Cup 14 - Finals Online Mirror (Unrated, ICPC Rules, Teams Preferred, Div. 2)

A. Weights

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an array A of length N weights of masses A_1, A_2...A_N. No two weights have the same mass. You can put every weight on one side of the balance (left or right). You don't have to put weights in order A_1,...,A_N. There is also a string S consisting of characters "L" and "R", meaning that after putting the $i-th$ weight (not A_i, but $i-th$ weight of your choice) left or right side of the balance should be heavier. Find the order of putting the weights on the balance such that rules of string S are satisfied.

Input

The first line contains one integer N $(1 \leq N \leq 2 * 10^5)$ - the length of the array A The second line contains N distinct integers: A_1, A_2,...,A_N $(1 \leq A_i \leq 10^9)$ - the weights given The third line contains string S of length N consisting only of letters "L" and "R" - string determining which side of the balance should be heavier after putting the $i-th$ weight of your choice

Output

The output contains N lines. In every line, you should print one integer and one letter - integer representing the weight you are putting on the balance in that move and the letter representing the side of the balance where you are putting the weight. If there is no solution, print -1.

Example

input

```
5
3 8 2 13 7
LLRLL
```

output

```
3 L
2 R
8 R
13 L
7 L
```

Note

Explanation for the test case:

after the 1st weight: 3 L (left side is heavier)

after the 2nd weight: 2 R (left side is heavier)

after the 3rd weight: 8 R (right side is heavier)

after the 4th weight: 13 L (left side is heavier)

after the 5th weight: 7 L (left side is heavier)

So, the rules given by string S are fulfilled and our order of putting the weights is correct.

B. Mars

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In the year 2420 humans have finally built a colony on Mars thanks to the work of Elon Tusk. There are 10^9+7 cities arranged in a circle in this colony and none of them are connected yet. Elon Tusk wants to connect some of those cities using only roads of the same size in order to lower the production cost of those roads. Because of that he gave a list on N cites where some cites can appear more than once and Q queries that you need to answer. For the query you need to determine if it is possible to connect all the cities from L_i to R_i on that list using only roads of length D_i.

Input

The first line contains two integers N and Q $(1 \leq N, Q \leq 2 \cdot 10^5$ $)$ — the length of the array of cities and the number of queries you need to answer.

The second lines contains N integers representing the array of cites. Next Q lines contain three integers L, R and D ($1 \leq L_i, R_i \leq N, 0 \leq D_i \leq 10^9+6)$ — the range of cities that needs to be connected and the length of the road that you can use.

Output

The output contains Q lines. If it is possible to connect all the cities from the i-th query can be connected with roads of length D_i the i-th line should contain the word "Yes", otherwise it should contain the word "No".

Examples

input

```
9 8
17 0 12 6 10 8 2 4 5
2 3 12
2 3 6
2 4 6
4 6 2
2 8 2
1 2 17
1 8 2
9 9 14
```

output

```
Yes
No
Yes
Yes
Yes
Yes
No
Yes
```

input

```
4 1
7 21 14 0
1 4 1000000000
```

output

```
Yes
```

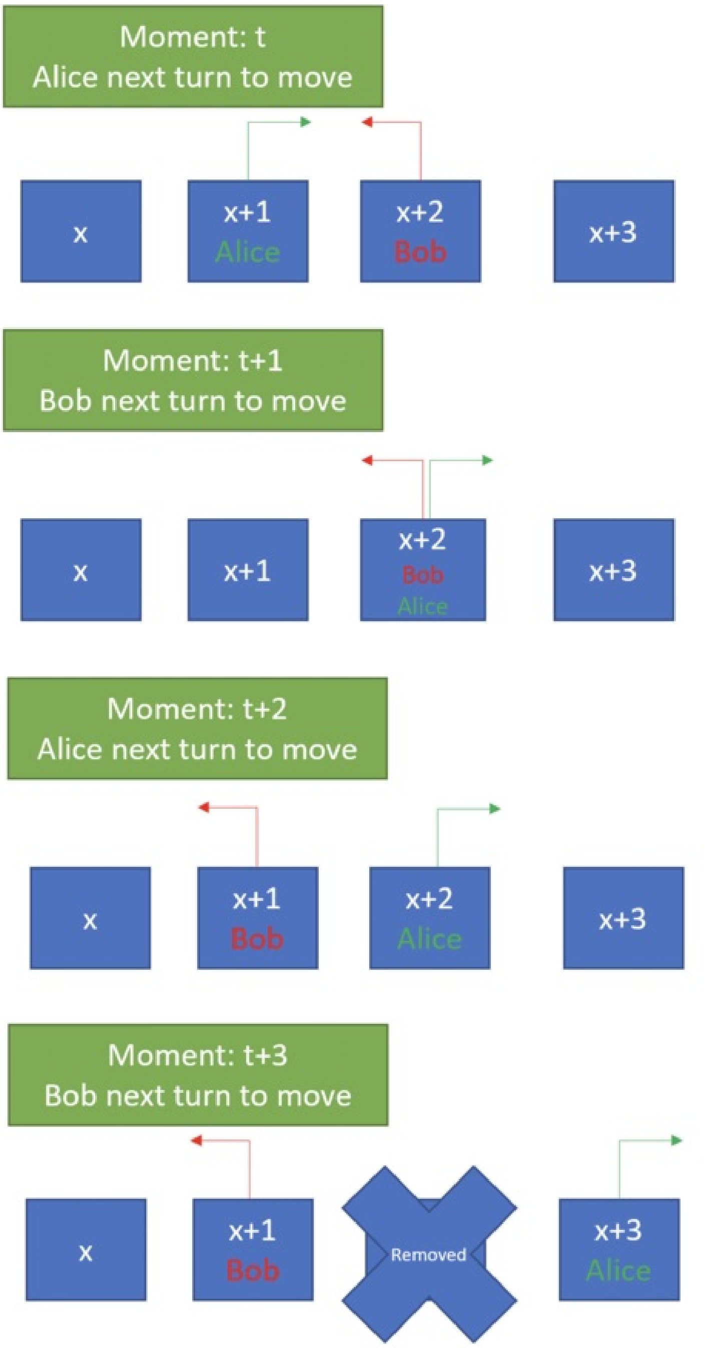
Note

In the 5^{th} query of the first test case we can connect cities in this order 0-2-4-6-8-10-12 this way distance between any two connected cities is 2. In the second test case we can connect cities in this order 21-14-7-0 this way distance between any two connected cities is 10^9 module $10^9 + 7$.

C. Restaurant Game

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice and Bob always had hard time choosing restaurant for the dinner. Previously they performed Eenie Meenie Miney Mo game, but eventually as their restaurant list grew, they had to create a new game. This new game starts as they write restaurant names on N cards and align the cards in one line. Before the game begins, they both choose starting card and starting direction they are going to. They take turns in order one after another. After each turn, they move one card in their current direction. If they reach the end or beginning of the line of cards they change direction. Once they meet in a card, the card is marked for removal and is removed the first moment they both leave the card.



Example of how card is removed

They repeat this process until there is only one restaurant card left. Since there are a lot of restaurant cards, they are bored to simulate this process over and over and need your help to determine the last card that remains. Can you help them?

Input

The first line of the input is one integer T $(1 \leq T \leq 10^4)$ representing number of test cases. Each test case contains 3 lines: The first line contains an integer N representing initial number of cards. Next line contains two integer values A, B $(0 \leq A, B < N, 2 \leq N \leq 10^{18})$ representing starting 0-based index of the card in the array. Last line contains two strings $D_A, D_B \in$ {"left", "right"} representing starting direction of their movement.

Output

The output contains T integer number – the 0-based index of the last card that remains for every test case in order.

Example

input
1 4 0 1 left right

output
0

Note

Note that since Alice is starting at the beginning of the line even though her initial direction is left, on her next move she will go right.

D. Hidden Fortress

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is an interactive problem!

As part of your contribution in the Great Bubble War, you have been tasked with finding the newly built enemy fortress. The world you live in is a giant $10^9 \times 10^9$ grid, with squares having both coordinates between 1 and 10^9.

You know that the enemy base has the shape of a rectangle, with the sides parallel to the sides of the grid. The people of your world are extremely scared of being at the edge of the world, so you know that the base doesn't contain any of the squares on the edges of the grid (the x or y coordinate being 1 or 10^9).

To help you locate the base, you have been given a device that you can place in any square of the grid, and it will tell you the manhattan distance to the closest square of the base. The manhattan distance from square (a, b) to square (p, q) is calculated as $|a - p| + |b - q|$. If you try to place the device inside the enemy base, you will be captured by the enemy. Because of this, **you need to make sure to never place the device inside the enemy base**.

Unfortunately, the device is powered by a battery and you can't recharge it. This means that you can use the device at most 40 times.

Input

The input contains the answers to your queries.

Interaction

Your code is allowed to place the device on any square in the grid by writing "? i j" $(1 \le i, j \le 10^9)$. In return, it will recieve the manhattan distance to the closest square of the enemy base from square (i, j) or -1 if the square you placed the device on is inside the enemy base or outside the grid.

If you recieve -1 instead of a positive number, exit immidiately and you will see the `wrong answer` verdict. Otherwise, you can get an arbitrary verdict because your solution will continue to read from a closed stream.

Your solution should use no more than 40 queries.

Once you are sure where the enemy base is located, you should print "! x y p q" $(1 \le x \le p \le 10^9, 1 \le y \le q \le 10^9)$, where (x, y) is the square inside the enemy base with the smallest x and y coordinates, and (p, q) is the square inside the enemy base with the largest x and y coordinates. **Note that answering doesn't count as one of the 40 queries.**

After printing a query or printing the answer, do not forget to output end of line and flush the output. Otherwise, you will get `idleness limit exceeded`. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- See the documentation for other languages.

Example

input

```
1
1
2
1
```

output

```
? 2 2
? 5 5
? 4 7
? 1 5
! 2 3 4 5
```

E. Array Game

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice and Bob are playing a game. They are given an array A of length N. The array consists of integers. They are building a sequence together. In the beginning, the sequence is empty. In one turn a player can remove a number from the left or right side of the array and append it to the sequence. The rule is that the sequence they are building must be strictly increasing. The winner is the player that makes the last move. Alice is playing first. Given the starting array, under the assumption that they both play optimally, who wins the game?

Input

The first line contains one integer N $(1 \leq N \leq 2 * 10^5)$ - the length of the array A.

The second line contains N integers A_1, A_2,...,A_N $(0 \leq A_i \leq 10^9)$

Output

The first and only line of output consists of one string, the name of the winner. If Alice won, print "Alice", otherwise, print "Bob".

Examples

input

```
1
5
```

output

```
Alice
```

input

```
3
5 4 5
```

output

```
Alice
```

input

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

output

```
Bob
```

F. Party Organization

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

On the great island of Baltia, there live N people, numbered from 1 to N. There are exactly M pairs of people that are friends with each other. The people of Baltia want to organize a successful party, but they have very strict rules on what a party is and when the party is successful. On the island of Baltia, a party is a gathering of exactly 5 people. The party is considered to be successful if either all the people at the party are friends with each other (so that they can all talk to each other without having to worry about talking to someone they are not friends with) or no two people at the party are friends with each other (so that everyone can just be on their phones without anyone else bothering them). Please help the people of Baltia organize a successful party or tell them that it's impossible to do so.

Input

The first line contains two integer numbers, N $(5 \leq N \leq 2 * 10^5)$ and M $(0 \leq M \leq 2 * 10^5)$ – the number of people that live in Baltia, and the number of friendships. The next M lines each contains two integers U_i and V_i $(1 \leq U_i, V_i \leq N)$ – meaning that person U_i is friends with person V_i. Two friends can not be in the list of friends twice (no pairs are repeated) and a person can be friends with themselves $(U_i \neq V_i)$.

Output

If it's possible to organize a successful party, print 5 numbers indicating which 5 people should be invited to the party. If it's not possible to organize a successful party, print -1 instead. If there are multiple successful parties possible, print any.

Examples

input

```
6 3
1 4
4 2
5 4
```

output

```
1 2 3 5 6
```

input

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

output

```
-1
```

G. Bob's Beautiful Array

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bob really likes playing with arrays of numbers. That's why for his birthday, his friends bought him a really interesting machine – an array beautifier.

The array beautifier takes an array A consisting of N integers, and it outputs a new array B of length N that it constructed based on the array given to it. The array beautifier constructs the new array in the following way: it takes two numbers **at different indices** from the original array and writes their sum to the end of the new array. It does this step N times - resulting in an output array of length N. During this process, the machine can take the same index multiple times in different steps.

Bob was very excited about the gift that his friends gave him, so he put his favorite array in the machine. However, when the machine finished, Bob was not happy with the resulting array. He misses his favorite array very much, and hopes to get it back.

Given the array that the machine outputted, help Bob find an array that could be the original array that he put in the machine. Sometimes the machine makes mistakes, so it is possible that no appropriate input array exists for the array it has outputted. In such case, let Bob know that his array is forever lost.

Input

The first line contains one positive integer N $(2 \leq N \leq 10^3)$ – the length of Bob's array.

The second line contains N integers B_1, B_2, ..., B_N $(1 \leq B_i \leq 10^6)$ – the elements of the array the machine outputted.

Output

If an appropriate input array exists, print "YES", followed by the input array A_1, A_2, ..., A_N $(-10^9 \leq A_i \leq 10^9)$ in the next line. Otherwise, print "NO".

Examples

input

```
2
5 5
```

output

```
YES
2 3
```

input

```
3
1 2 3
```

output

```
YES
0 1 2
```

input

```
3
2 4 5
```

output

```
NO
```

input

```
4
1 3 5 7
```

output

```
YES
6 -3 4 1
```

H. Shortest path

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given N points on an infinite plane with the Cartesian coordinate system on it. $N-1$ points lay on one line, and one point isn't on that line. You are on point K at the start, and the goal is to visit every point. You can move between any two points in a straight line, and you can revisit points. What is the minimum length of the path?

Input

The first line contains two integers: N $(3 \leq N \leq 2 * 10^5)$ - the number of points, and K $(1 \leq K \leq N)$ - the index of the starting point.

Each of the next N lines contain two integers, A_i, B_i $(-10^6 \leq A_i, B_i \leq 10^6)$ - coordinates of the $i - th$ point.

Output

The output contains one number - the shortest path to visit all given points starting from point K. The absolute difference between your solution and the main solution shouldn't exceed 10^-6;

Example

input

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

output

```
7.478709
```

Note

The shortest path consists of these moves:

2 -> 5

5 -> 4

4 -> 1

1 -> 3

There isn't any shorter path possible.

I. Bubble Strike

time limit per test: 0.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Johnny Bubbles enjoys spending hours in front of his computer playing video games. His favorite game is Bubble Strike, fast-paced bubble shooting online game for two players.

Each game is set in one of the N maps, each having different terrain configuration. First phase of each game decides on which map the game will be played. The game system randomly selects three maps and shows them to the players. Each player must pick one of those three maps to be discarded. The game system then randomly selects one of the maps that were not picked by any of the players and starts the game.

Johnny is deeply enthusiastic about the game and wants to spend some time studying maps, thus increasing chances to win games played on those maps. However, he also needs to do his homework, so he does not have time to study all the maps. That is why he asked himself the following question: "What is the minimum number of maps I have to study, so that the probability to play one of those maps is at least P"?

Can you help Johnny find the answer for this question? You can assume Johnny's opponents do not know him, and they will randomly pick maps.

Input

The first line contains two integers N $(3 \leq N \leq 10^3)$ and P $(0 \leq P \leq 1)$ – total number of maps in the game and probability to play map Johnny has studied. P will have at most four digits after the decimal point.

Output

Output contains one integer number – minimum number of maps Johnny has to study.

Example

input

```
7 1.0000
```

output

```
6
```

J. Robot Factory

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input

output: standard output

You have received data from a Bubble bot. You know your task is to make factory facilities, but before you even start, you need to know how big the factory is and how many rooms it has. When you look at the data you see that you have the dimensions of the construction, which is in rectangle shape: N x M.

Then in the next N lines you have M numbers. These numbers represent factory tiles and they can go from 0 to 15. Each of these numbers should be looked in its binary form. Because from each number you know on which side the tile has walls. For example number 10 in it's binary form is 1010, which means that it has a wall from the **North** side, it doesn't have a wall from the **East**, it has a wall on the **South** side and it doesn't have a wall on the **West** side. So it goes North, East, South, West.

It is guaranteed that the construction always has walls on it's edges. The input will be correct.

Your task is to print the size of the rooms from biggest to smallest.

Input

The first line has two numbers which are N and M, the size of the construction. Both are integers:

n $(1 \leq n \leq 10^3)$

m $(1 \leq m \leq 10^3)$

Next N x M numbers represent each tile of construction.

Output

Once you finish processing the data your output consists of one line sorted from **biggest** to **smallest** room sizes.

Example

input

```
4 5
9 14 11 12 13
5 15 11 6 7
5 9 14 9 14
3 2 14 3 14
```

output

```
9 4 4 2 1
```