Problem J1: Boiling Water

Problem Description

At sea level, atmospheric pressure is 100 kPa and water begins to boil at 100°C. As you go above sea level, atmospheric pressure decreases, and water boils at lower temperatures. As you go below sea level, atmospheric pressure increases, and water boils at higher temperatures. A formula relating atmospheric pressure to the temperature at which water begins to boil is

$$P = 5 \times B - 400$$

where P is atmospheric pressure measured in kPa, and B is the temperature at which water begins to boil measured in °C.

Given the temperature at which water begins to boil, determine atmospheric pressure. Also determine if you are below sea level, at sea level, or above sea level.

Note that the science of this problem is generally correct but the values of 100° C and 100 kPa are approximate and the formula is a simplification of the exact relationship between water's boiling point and atmospheric pressure.

Input Specification

The input is one line containing an integer B where $B \geq 80$ and $B \leq 200$. This represents the temperature in °C at which water begins to boil.

Output Specification

The output is two lines. The first line must contain an integer which is atmospheric pressure measured in kPa. The second line must contain the integer `-1`, `0`, or `1`. This integer represents whether you are below sea level, at sea level, or above sea level, respectively.

Sample Input 1

```
99
```

Output for Sample Input 1

```
95
1
```

Explanation of Output for Sample Input 1

When $B = 99$, we can substitute into the formula and get $P = 5 \times 99 - 400$ which equals 95. Since 95 kPa is less than 100 kPa, you are above sea level.

Sample Input 2

```
102
```

Output for Sample Input 2

```
110
-1
```

La version française figure à la suite de la version anglaise.

Explanation of Output for Sample Input 2
When $B = 102$, we can substitute into the formula and get $P = 5 \times 102 - 400$ which equals 110. Since 110 kPa is greater than 100 kPa, you are below sea level.

Problem J2: Silent Auction

Problem Description

A charity is having a silent auction where people place bids on a prize without knowing anyone else's bid. Each bid includes a person's name and the amount of their bid. After the silent auction is over, the winner is the person who has placed the highest bid. If there is a tie, the person whose bid was placed first wins. Your job is to determine the winner of the silent auction.

Input Specification

The first line of input contains a positive integer N, where $1 \leq N \leq 100$, representing the number of bids collected at the silent auction. Each of the next N pairs of lines contains a person's name on one line, and the amount of their bid, in dollars, on the next line. Each bid is a positive integer less than 2000. The order of the input is the order in which bids were placed.

Output Specification

Output the name of the person who has won the silent auction.

Sample Input 1

```
3
Ahmed
300
Suzanne
500
Ivona
450
```

Output for Sample Input 1

```
Suzanne
```

Explanation of Output for Sample Input 1

The highest bid placed was `500` and it was placed by Suzanne. Suzanne wins the silent auction.

Sample Input 2

```
2
Ijeoma
20
Goor
20
```

Output for Sample Input 2

```
Ijeoma
```

La version française figure à la suite de la version anglaise.

Explanation of Output for Sample Input 2

The highest bid placed was 20 and it was placed by both Ijeoma and Goor. Since Ijeoma's bid was placed first, Ijeoma wins the silent auction.

La version française figure à la suite de la version anglaise.

Problem J3: Secret Instructions

Problem Description
Professor Santos has decided to hide a secret formula for a new type of biofuel. She has, however, left a sequence of coded instructions for her assistant.

Each instruction is a sequence of five digits which represents a direction to turn and the number of steps to take.

The first two digits represent the direction to turn:

- If their sum is odd, then the direction to turn is left.
- If their sum is even and not zero, then the direction to turn is right.
- If their sum is zero, then the direction to turn is the same as the previous instruction.

The remaining three digits represent the number of steps to take which will always be at least 100.

Your job is to decode the instructions so the assistant can use them to find the secret formula.

Input Specification
There will be at least two lines of input. Each line except the last line will contain exactly five digits representing an instruction. The first line will not begin with `00`. The last line will contain `99999` and no other line will contain `99999`.

Output Specification
There must be one line of output for each line of input except the last line of input. These output lines correspond to the input lines (in order). Each output line gives the decoding of the corresponding instruction: either `right` or `left`, followed by a single space, followed by the number of steps to be taken in that direction.

Sample Input
```
57234
00907
34100
99999
```

Output for Sample Input
```
right 234
right 907
left 100
```

La version française figure à la suite de la version anglaise.

Explanation of Output for Sample Input

The first instruction is 57234 which is decoded as right 234 because $5 + 7 = 12$ which is even and 57 is followed by 234.

The second instruction is 00907 which is decoded with the same direction as the previous instruction (right) but with 907 steps.

The third instruction is 34100 which is decoded as left 100 because $3+4 = 7$ which is odd and 34 is followed by 100.

The last line contains 99999 which tells us these are the only three instructions.

La version française figure à la suite de la version anglaise.

Problem J4: Arranging Books

Problem Description

Valentina wants books on a shelf to be arranged in a particular way. Every time she sees a shelf of books, she rearranges the books so that all the large books appear on the left, followed by all the medium-sized books, and then all the small books on the right. She does this by repeatedly choosing any two books and exchanging their locations. Exchanging the locations of two books is called a *swap*.

Help Valentina determine the fewest number of swaps needed to arrange a shelf of books as she wishes.

Input Specification

The input will consist of exactly one line containing at least one character.

The following table shows how the available 15 marks are distributed.

7 marks	at most 1000 characters	each character will be `L` or `S`
2 marks	at most 500 000 characters	each character will be `L` or `S`
4 marks	at most 1000 characters	each character will be `L`, `M`, or `S`
2 marks	at most 500 000 characters	each character will be `L`, `M`, or `S`

Output Specification

Output a single integer which is equal to the minimum possible number of swaps needed to arrange the books so that all occurrences of `L` come first, followed by all occurrences of `M`, and then all occurrences of `S`.

Sample Input 1

```
LMMMS
```

Output for Sample Input 1

```
0
```

Explanation of Output for Sample Input 1

The books are already arranged according to Valentina's wishes.

Sample Input 2

```
LLSLM
```

Output for Sample Input 2

```
2
```

La version française figure à la suite de la version anglaise.

Explanation of Output for Sample Input 2

By swapping the `S` and `M`, we end up with `LLMLS`. Then the `M` can be swapped with the `L` to its right. This is one way to use two swaps to arrange the books as Valentina wants them to be. It is not possible to use fewer swaps because both `S` and `M` need to be moved but using one swap to exchange them does not leave the books arranged as Valentina wants them to be.

La version française figure à la suite de la version anglaise.

Problem J5/S2: Modern Art

Problem Description

A new and upcoming artist has a unique way to create checkered patterns. The idea is to use an M-by-N canvas which is initially entirely black. Then the artist repeatedly chooses a row or column and runs their magic brush along the row or column. The brush changes the colour of each cell in the row or column from black to gold or gold to black.

Given the artist's choices, your job is to determine how much gold appears in the pattern determined by these choices.

Input Specification

The first line of input will be a positive integer M. The second line of input will be a positive integer N. The third line of input will be a positive integer K. The remaining input will be K lines giving the choices made by the artist. Each of these lines will either be `R` followed by a single space and then an integer which is a row number, or `C` followed by a single space and then an integer which is a column number. Rows are numbered top down from 1 to M. Columns are numbered left to right from 1 to N.

The following table shows how the available 15 marks are distributed.

1 mark	$M = 1$	$N = 1$	$K \leq 100$	only one cell, and up to 100 choices by the artist
4 marks	$M = 1$	$N \leq 100$	$K \leq 100$	only one row, and up to 100 choices by the artist
5 marks	$M \leq 100$	$N \leq 100$	$K \leq 100$	up to 100 rows, up to 100 columns, and up to 100 choices by the artist
5 marks	$MN \leq 5\,000\,000$		$K \leq 1\,000\,000$	up to 5 000 000 cells, and up to 1 000 000 choices by the artist

Output Specification

Output one non-negative integer which is equal to the number of cells that are gold in the pattern determined by the artist's choices.

Sample Input 1

```
3
3
2
R 1
C 1
```

Output for Sample Input 1

```
4
```

La version française figure à la suite de la version anglaise.

Explanation of Output for Sample Input 1

After running the brush along the first row, the canvas looks like this:

```
GGG
BBB
BBB
```

Then after running the brush along the first column, four cells are gold in the final pattern determined by the artist's choices:

```
BGG
GBB
GBB
```

Sample Input 2

```
4
5
7
R 3
C 1
C 2
R 2
R 2
C 1
R 4
```

Output for Sample Input 2

```
10
```

Explanation of Output for Sample Input 2

Ten cells are gold in the final pattern determined by the artist's choices:

```
BGBBB
BGBBB
GBGGG
GBGGG
```

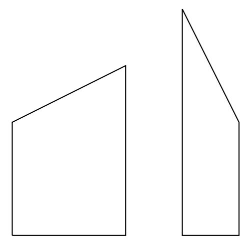
La version française figure à la suite de la version anglaise.

Problem S1: Crazy Fencing

Problem Description

You need to paint a wooden fence between your house and your neighbour's house. You want to determine the area of the fence, in order to determine how much paint you will use.

However, the fence is made out of N non-uniform pieces of wood, and your neighbour believes that they have an artistic flair. In particular, the pieces of wood may be of various widths. The bottom of each piece of wood will be horizontal, both sides will be vertical, but its top may be cut on an angle. Two such pieces of wood are shown below:



Thankfully, the fence has been constructed so that adjacent pieces of wood have the same height on the sides where they touch, which makes the fence more visually appealing.

Input Specification

The first line of the input will be a positive integer N, where $N \leq 10\,000$.

The second line of input will contain $N + 1$ space-separated integers $h_1, \ldots, h_{N+1}$ ($1 \leq h_i \leq 100$, $1 \leq i \leq N + 1$) describing the left and right heights of each piece of wood. Specifically, the left height of the i^{th} piece of wood is h_i and the right height of the i^{th} piece of wood is h_{i+1}.

The third line of input will contain N space-separated integers w_i ($1 \leq w_i \leq 100$, $1 \leq i \leq N$) describing the width of the i^{th} piece of wood.

Output Specification

Output the total area of the fence.

Sample Input 1

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

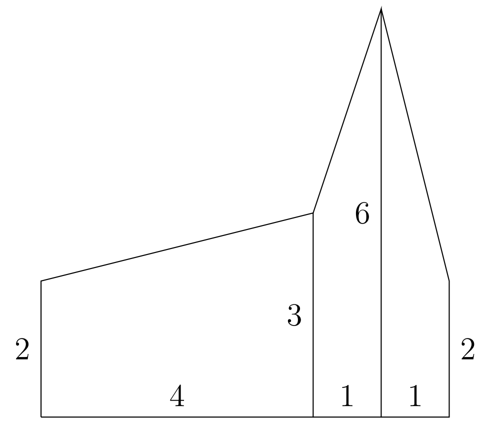
Output for Sample Input 1

```
18.5
```

Explanation of Output for Sample Input 1

La version française figure à la suite de la version anglaise.

The fence looks like the following:



When looking from left to right, the individual areas of the pieces of wood are $10 = 4 \cdot (2+3)/2$, $4.5 = 1 \cdot (3+6)/2$, and $4 = 1 \cdot (6+2)/2$, for a total area of 18.5.

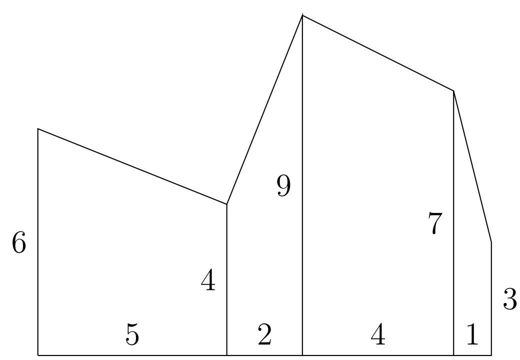
Sample Input 2

```
4
6 4 9 7 3
5 2 4 1
```

Output for Sample Input 2

```
75
```

Explanation of Output for Sample Input 2
The fence looks like the following:



When looking from left to right, the individual areas of the pieces of wood are 25, 13, 32, and 5, for a total area of 75.

La version française figure à la suite de la version anglaise.

Problem S3: Lunch Concert

Problem Description

It's lunchtime at your school! Your N friends are all standing on a long field, as they usually do. The field can be represented as a number line, with the ith friend initially at position P_i metres along it. The ith friend is able to walk in either direction along the field at a rate of one metre per W_i seconds, and their hearing is good enough to be able to hear music up to and including D_i metres away from their position. Multiple students may occupy the same positions on the field, both initially and after walking.

You're going to hold a little concert at some position c metres along the field (where c is any integer of your choice), and text all of your friends about it. Once you do, each of them will walk along the field for the minimum amount of time such that they end up being able to hear your concert (in other words, such that each friend i ends up within D_i units of c).

You'd like to choose c such that you minimize the sum of all N of your friends' walking times. What is this minimum sum (in seconds)? Please note that the result might not fit within a 32-bit integer.

Input Specification

The first line of input contains N.
The next N lines contain three space-separated integers, P_i, W_i, and D_i $(1 \leq i \leq N)$.

The following table shows how the available 15 marks are distributed.

4 marks	$1 \leq N \leq 2000$	$0 \leq P_i \leq 2000$	$1 \leq W_i \leq 1000$	$0 \leq D_i \leq 2000$
9 marks	$1 \leq N \leq 200\,000$	$0 \leq P_i \leq 1\,000\,000$	$1 \leq W_i \leq 1000$	$0 \leq D_i \leq 1\,000\,000$
2 marks	$1 \leq N \leq 200\,000$	$0 \leq P_i \leq 1\,000\,000\,000$	$1 \leq W_i \leq 1000$	$0 \leq D_i \leq 1\,000\,000\,000$

Output Specification

Output one integer which is the minimum possible sum of walking times (in seconds) for all N of your friends to be able to hear your concert.

Sample Input 1

```
1
0 1000 0
```

Output for Sample Input 1

```
0
```

Explanation of Output for Sample Input 1

If you choose $c = 0$, your single friend won't need to walk at all to hear it.

La version française figure à la suite de la version anglaise.

Sample Input 2

```
2
10 4 3
20 4 2
```

Output for Sample Input 2

```
20
```

Explanation of Output for Sample Input 2

One possible optimal choice of c is 14, which would require your first friend to walk to position 11 (taking $4 \times 1 = 4$ seconds) and your second friend to walk to position 16 (taking $4 \times 4 = 16$ seconds), for a total of 20 seconds.

Sample Input 3

```
3
6 8 3
1 4 1
14 5 2
```

Output for Sample Input 3

```
43
```

La version française figure à la suite de la version anglaise.

Problem S4: Daily Commute

Problem Description

Toronto has N subway stations, numbered from 1 to N. You start at station 1, and every day, you need to reach station N to get to school.

There are W *one-way* walkways running amongst the stations, the i^{th} of which allows you to walk *from* station A_i *to* a different station B_i ($1 \le A_i, B_i \le N$, $A_i \ne B_i$) in 1 minute. There may be multiple walkways connecting any given pair of stations.

The subway line follows a certain route through the N stations, starting at station 1 and visiting each station once. Initially, this route consists of stations $S_1, S_2, ..., S_N$, in that order. $S_1 = 1$, and $S_2, \ldots, S_N$ is a permutation of the integers $2, \ldots, N$. Only one subway train runs along this route per day, departing from station 1 at 6am in the morning and taking 1 minute to reach each subsequent station. This means that, m minutes after 6am, the train will be at station S_{m+1} (or at station S_N if $m \ge N - 1$).

Over a period of D days, however, the subway line's route will keep changing. At the start of the i^{th} day, the ${X_i}^{\text{th}}$ station and ${Y_i}^{\text{th}}$ station ($2 \le X_i, Y_i \le N$, $X_i \ne Y_i$) in the route will be swapped. Note that, after each such change, the route will still begin at station 1 and will visit all N stations once each. Changes will carry over to subsequent days – the route will not automatically reset itself back to $S_1, \ldots, S_N$.

On each of these D days, you'd like to determine how quickly you can get to school so you can begin learning things. On the i^{th} day, starting at 6am in the morning (after the i^{th} update to the subway line's route), you'll begin your daily trip to station N. Each minute, you may either ride the subway to its next stop (if you're currently at the same station as the train and it hasn't already completed its route), take a walkway from your current station to another one, or wait at your current station. Note that your trip begins at the same time as the train's route, meaning that you may choose to immediately ride it if you'd like to, and that you may choose to leave and then get back on the train during your trip.

Input Specification

The first line contains three space-separated integers, N, W, and D.
The next W lines each contain two space-separated integers, A_i and B_i ($1 \le i \le W$).
The next line contains the N space-separated integers, $S_1, \ldots, S_N$, which form the initial permutation of stations.
The next D lines each contain two space-separated integers, X_i and Y_i ($1 \le i \le D$).

The following table shows how the available 15 marks are distributed.

2 marks	$3 \le N \le 10$	$0 \le W \le 10$	$1 \le D \le 10$
2 marks	$3 \le N \le 200$	$0 \le W \le 200$	$1 \le D \le 200$
3 marks	$3 \le N \le 2000$	$0 \le W \le 2000$	$1 \le D \le 2000$
8 marks	$3 \le N \le 200\,000$	$0 \le W \le 200\,000$	$1 \le D \le 200\,000$

La version française figure à la suite de la version anglaise.

Output Specification

The output is D lines, with one integer per line. The i^{th} line is the minimum number of minutes required to reach station N on the i^{th} day ($1 \leq i \leq D$).

Sample Input

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

Output for Sample Input

```
1
2
3
```

Explanation of Output for Sample Input

At the start of the first day, the subway line's route will be updated to visit stations $[1, 4, 2, 3]$, in that order. On that day, you should simply take the subway to station 4, taking 1 minute.

On the second day, the route will become $[1, 3, 2, 4]$, and you should take the subway to station 3 (taking 1 minute) and then walk to station 4 (taking 1 more minute).

On the third day, the route will become $[1, 2, 3, 4]$. One choice of optimal trip involves walking to station 2 (taking 1 minute), then boarding the train there and taking it through station 3 and finally to station 4 (taking another 2 minutes).

La version française figure à la suite de la version anglaise.

Sample Input 2

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

Output for Sample Input 2

```
Impossible
```

Explanation of Output for Sample Input 2

There exists no sequence $[A_1, A_2]$ such that the GCD of $[A_1, A_2]$ is 2 and the GCD of $[A_2]$ is 5.

La version française figure à la suite de la version anglaise.

Problem S5: Math Homework

Problem Description

Your math teacher has given you an assignment involving coming up with a sequence of N integers $A_1, \ldots, A_N$, such that $1 \leq A_i \leq 1\ 000\ 000\ 000$ for each i.

The sequence A must also satisfy M requirements, with the i^{th} one stating that the GCD (Greatest Common Divisor) of the contiguous subsequence $A_{X_i}, \ldots, A_{Y_i}$ $(1 \leq X_i \leq Y_i \leq N)$ must be equal to Z_i. Note that the GCD of a sequence of integers is the largest integer d such that all the numbers in the sequence are divisible by d.

Find *any* valid sequence A consistent with all of these requirements, or determine that no such sequence exists.

Input Specification

The first line contains two space-separated integers, N and M.
The next M lines each contain three space-separated integers, X_i, Y_i, and Z_i $(1 \leq i \leq M)$.

The following table shows how the available 15 marks are distributed.

3 marks	$1 \leq N \leq 2000$	$1 \leq M \leq 2000$	$1 \leq Z_i \leq 2$ for each i
4 marks	$1 \leq N \leq 2000$	$1 \leq M \leq 2000$	$1 \leq Z_i \leq 16$ for each i
8 marks	$1 \leq N \leq 150\,000$	$1 \leq M \leq 150\,000$	$1 \leq Z_i \leq 16$ for each i

Output Specification

If no such sequence exists, output the string `Impossible` on one line. Otherwise, on one line, output N space-separated integers, forming the sequence $A_1, \ldots, A_N$. If there are multiple possible valid sequences, *any* valid sequence will be accepted.

Sample Input 1

```
2 2
1 2 2
2 2 6
```

Output for Sample Input 1

```
4 6
```

Explanation of Output for Sample Input 1

If $A_1 = 4$ and $A_2 = 6$, the GCD of $[A_1, A_2]$ is 2 and the GCD of $[A_2]$ is 6, as required. **Please note that other outputs would also be accepted.**

La version française figure à la suite de la version anglaise.

2021 Canadian Computing
Problem 1
Swap Swap Sort

Time Limit: 3 seconds

Problem Description

There is an array of N integers, each with a value between 1 and K. Your friend has a an algorithm that can sort this array according to any ordering of the numbers from 1 to K. The algorithm performs a sequence of **swap** operations, that exchange two **adjacent** elements of the array. The algorithm performs exactly the minimum number of such swaps to sort the array.

The desired ordering of the numbers from 1 to K is given by a **target permutation**. A target permutation is a sequence of each number from 1 to K appearing exactly once, in the same order that is desired by the corresponding ordering.

For example, the array `[1 4 2 1 2]` sorted by the target permutation $4, 1, 2, 3$ results in the array `[4 1 1 2 2]`.

You are interested in the number of swaps performed by your friend's algorithm for different target permutations. To explore this, you start with a target permutation of $1, 2, \ldots, K$ and perform Q operations on it. Each operation swaps two **adjacent** elements of the target permutation. After performing each operation, find the number of swaps your friend's algorithm would make if it was run with the current target permutation. The Q operations cumulatively change the target permutation, but do not affect the array.

Input Specification

The first line contains the three integers N, K, and Q ($1 \leq K \leq N \leq 100\,000$, $1 \leq Q \leq 1\,000\,000$).

The next line contains N integers $a_1, a_2, \ldots, a_N$ ($1 \leq a_i \leq K$) specifying the array .

The next Q lines each contains a single integer j ($1 \leq j \leq K-1$), representing the operation of swapping the elements of the target permutation at indices j and $j+1$.

For 3 of the 25 available marks, $N, Q \leq 5\,000$.

For an additional 3 of the 25 available marks, $Q \leq 100$.

For an additional 5 of the 25 available marks, $K \leq 500$.

Output Specification

For each of the Q operations, output a line containing a single integer; the answer for the current target permutation.

Sample Input

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

Output for Sample Input

```
4
2
2
```

Explanation of Output for Sample Input

The three target permutations are $1, 2, 4, 3$, then $1, 4, 2, 3$, then $4, 1, 2, 3$. For the final target permutation, your friend's algorithm uses two swaps to sort the array `[1 4 2 1 2]` to `[4 1 1 2 2]`.

2021 Canadian Computing
Problem 2
Weird Numeral System

Time Limit: 1.5 seconds

Problem Description

Alice enjoys thinking about base-K numeral systems (don't we all?). As you might know, in the standard base-K numeral system, an integer n can be represented as $d_{m-1}\ d_{m-2}\ \ldots\ d_1\ d_0$ where:

- Each digit d_i is in the set $\{0, 1, \ldots, K-1\}$, and
- $d_{m-1}K^{m-1} + d_{m-2}K^{m-2} + \cdots + d_1K^1 + d_0K^0 = n$.

For example, in standard base-3, you would write 15 as `1 2 0`, since $(1)\cdot 3^2 + (2)\cdot 3^1 + (0)\cdot 3^0 = 15$.

But standard base-K systems are too easy for Alice. Instead, she's thinking about **weird-base-K** systems.

A weird-base-K system is just like the standard base-K system, except that instead of using the digits $\{0, \ldots, K-1\}$, you use $\{a_1, a_2, \ldots, a_D\}$ for some value D. For example, in a weird-base-3 system with $a = \{-1, 0, 1\}$, you could write 15 as `1 -1 -1 0`, since $(1)\cdot 3^3 + (-1)\cdot 3^2 + (-1)\cdot 3^1 + (0)\cdot 3^0 = 15$.

Alice is wondering how to write Q integers, n_1 through n_Q, in a weird-base-K system that uses the digits a_1 through a_D. Please help her out!

Input Specification

The first line contains four space-separated integers, K, Q, D, and M ($2 \le K \le 1\,000\,000$, $1 \le Q \le 5$, $1 \le D \le 5001$, $1 \le M \le 2500$).

The second line contains D distinct integers, a_1 through a_D ($-M \le a_i \le M$).

Finally, the i-th of the next Q lines contains n_i ($-10^{18} \le n_i \le 10^{18}$).

For 8 of the 25 available marks, $M = K - 1 \le 400$, $K = D \le 801$.

Output Specification

Output Q lines, the i-th of which is a weird-base-K representation of n_i. If multiple representations are possible, any will be accepted. The digits of the representation should be separated by spaces. Note that 0 must be represented by a non-empty set of digits.

If there is no possible representation, output `IMPOSSIBLE`.

Sample Input 1

```
3 3 3 1
-1 0 1
15
8
-5
```

Output for Sample Input 1

```
1 -1 -1 0
1 0 -1
-1 1 1
```

Explanation of Output for Sample Input 1

We have:
$(1) \cdot 3^3 + (-1) \cdot 3^2 + (-1) \cdot 3^1 + (0) \cdot 3^0 = 15$,
$(1) \cdot 3^2 + (0) \cdot 3^1 + (-1) \cdot 3^0 = 8$, and
$(-1) \cdot 3^2 + (1) \cdot 3^1 + (1) \cdot 3^0 = -5$.

Sample Input 2

```
10 1 3 2
0 2 -2
17
```

Output for Sample Input 2

```
IMPOSSIBLE
```

2021 Canadian Computing
Problem 3
Through Another Maze Darkly

Time Limit: 8 seconds

Problem Description
There is a maze that is formed by connecting N rooms via $N-1$ corridors. The rooms are numbered 1 to N and each room has the shape of a circle. The corridors have the following constraints:

- Each corridor forms a connection between two distinct rooms.
- Every pair of rooms is connected by exactly one path of connected corridors.

One difficulty in navigating through this maze is that the lights are all out, so you cannot see where you are. To help with navigation, each room contains a laser pointer that initially points to a corridor. Consider the following strategy:

- Rotate the room's laser pointer clockwise until it points to another corridor.
- Follow the room's laser pointer and traverse the corridor.
- Repeat the previous two steps indefinitely.

You created Q queries to investigate this strategy. For each query, you are given an integer K and you start at room 1. Determine the final room after traversing exactly K corridors. All laser pointers will reset to their original orientation after each query.

Input Specification
The first line contains the integers N and Q ($2 \leq N \leq 800\,000$, $1 \leq Q \leq 800\,000$).

The next N lines describe the layout of the maze, with the i^{th} of these lines describing room i. Specifically, it contains an integer k, the number of corridors connecting to room i, followed by k integers, $c_1\ c_2 \ldots c_k$, indicating the rooms that these k corridors lead to, in clockwise order. Lastly, room i's laser pointer initially points to the corridor leading to room c_1.

The final Q lines describe a query, with each line containing an integer K ($1 \leq K \leq 10^{15}$).

For 4 of the 25 available marks, the i^{th} corridor forms a connection between room i and room $i+1$.

For an additional 4 of the 25 available marks, $N \leq 2000$ and $Q \leq 2000$.

For an additional 12 of the 25 available marks, $Q = 1$.

Output Specification

Output Q lines answering the queries in order.

Sample Input

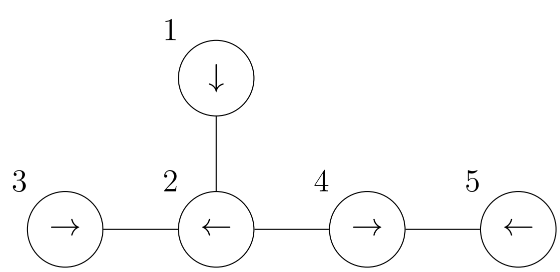
```
5 6
1 2
3 3 1 4
1 2
2 5 2
1 4
1
2
3
4
5
6
```

Output for Sample Input

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

Explanation of Output for Sample Input

This is the initial state of the maze.



The strategy will visit these rooms in order:

$$1, 2, 1, 2, 4, 2, 3, \ldots$$

2021 Canadian Computing

Problem 4

Travelling Merchant

Time Limit: 1 second

Problem Description

A merchant would like to make a business of travelling between cities, moving goods from one city to another in exchange for a profit. There are N cities and M trading routes between them.

The i-th trading route lets the merchant travel from city a_i to city b_i (in only that direction). In order to take this route, the merchant must already have at least r_i units of money. After taking this route, the total amount of money the merchant has will increase by p_i units, with a guarantee that $p_i \geq 0$.

For each of the N cities, we would like to know the minimum amount of money required for a merchant to start in that city and be able to keep travelling forever.

Input Specification

The first line contains the two integers N and M $(2 \leq N, M \leq 200\,000)$.

The i-th of the next M lines contains the four integers a_i, b_i, r_i, and p_i $(1 \leq a_i, b_i \leq N$, $a_i \neq b_i$, $0 \leq r_i, p_i \leq 10^9)$. Note that there may be any number of routes between a pair of cities.

For 4 of the 25 available marks, $N, M \leq 2\,000$.

For an additional 5 of the 25 available marks, $p_i = 0$ for all i.

Output Specification

On a single line, output N space-separated integers, where the i-th is the answer if the merchant were to start at city i. This answer is either the minimum amount of money, or -1 if no amount of money can be sufficient.

Sample Input

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

Output for Sample Input

```
2 3 3 1 -1
```

Explanation of Output for Sample Input

Starting from city 2 with 3 units of money, the merchant can cycle between cities 2, 1, and 3.

2021 Canadian Computing

PROBLEM 5

Bread First Search

Time Limit: 1 second

Problem Description

There are N towns in a network of M undirected roads. Each road connects one pair of towns. The government has decided to conduct a breadth first search, which means finding an ordering of the N towns such that if the ordering begins with r:

- Each town except for r is adjacent to another town given earlier in the order.
- The towns are given in non-decreasing order of distance to r. Here, distance means the minimum number of roads that need to be traversed to reach a town.

However, someone mistakenly did a *bread* first search. They found the ordering $1, 2, \ldots, N$ (this was obtained by sorting the towns in increasing order of bread production).

To cover up this embarrassment, the government would like to build new roads such that $1, 2, \ldots, N$ is also a possible breadth first search ordering. The new roads can be built between any two towns. What is the minimum possible number of roads that need to be built?

Input Specification

The first line contains the two integers N and M ($1 \leq N \leq 200\,000$, $0 \leq M \leq \min(200\,000, \frac{N(N-1)}{2})$).

The i-th of the next M lines contains the two integers a_i and b_i ($1 \leq a_i, b_i \leq N$), representing the two endpoints of the i-th road. It is guaranteed that $a_i \neq b_i$ and there is at most one road between any two towns.

For 5 of the 25 available marks, $N \leq 200$.

For an additional 6 of the 25 marks available, $N \leq 5\,000$.

Output Specification

On a single line, output the minimum number of roads that must be constructed.

Sample Input 1

```
2 0
```

Output for Sample Input 1

```
1
```

Explanation of Output for Sample Input 1
For $1, 2$ to be a breadth first search ordering, a road between towns 1 and 2 must be built.

Sample Input 2
```
6 3
1 3
2 6
4 5
```

Output for Sample Input 2
```
2
```

Explanation of Output for Sample Input 2
By building a road between 1 and 2 and a road between 1 and 4, the sequence of distances becomes $0, 1, 1, 1, 2, 2$ which is in non-decreasing order.

2021 Canadian Computing

Problem 6

Loop Town

Time Limit: 4 seconds

Problem Description

Some cities have complicated road networks that require advanced graph theory to analyze. But not Loop Town! Loop Town has a single circular road that loops around the town. It has N residents that live in N distinct houses located around the road. The town also has N offices, and each resident works at a distinct office.

The road in Loop Town has length L. The location of each building will be represented by an integer between 0 and $L - 1$. Since the road is circular, the positions 0 and $L - 1$ are adjacent. It is guaranteed that the locations of all $2N$ buildings will be distinct.

Every morning, all N residents simultaneously exit their houses onto the road. They then need to walk along the road to the entrance of the office where they work. When each resident has reached the entrance of their office, they all enter simultaneously.

However, a pandemic has now come to Loop Town, disrupting this usual routine. To prevent the spread of disease, residents must now observe social distancing while walking to work. Since the loop road is rather narrow, this means that it is far more inconvenient for two people to cross each other on their way to work (one person must temporarily step off the path to let the other pass). What is the minimum total number of crossings, assuming all the residents work together to achieve this?

Input Specification

The first line contains the two integers N and L ($1 \le N \le 1\,000\,000$, $1 \le L \le 10^9$).

The i-th of the next N lines contains the two integers a_i and b_i ($0 \le a_i, b_i < L$), where a_i and b_i represent the locations of the i-th resident's house and office respectively. It is guaranteed that all $2N$ locations are distinct.

For 12 of the 25 available marks, $N \le 5\,000$.

For an additional 6 of the 25 available marks, $N \le 100\,000$.

Output Specification

On a single line, output the minimum total number of crossings.

Sample Input 1

```
3 100
10 50
```

```
30 20
60 40
```

Output for Sample Input 1

```
0
```

Explanation of Output for Sample Input 1

Since the road is circular, nobody needs to cross each other.

Sample Input 2

```
4 100
30 70
10 12
60 75
90 50
```

Output for Sample Input 2

```
1
```