VK CUP 2016

Codeforces Round #348 (VK Cup 2016 Round 2, Div. 2 Edition)

A. Little Artem and Presents

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

Little Artem got n stones on his birthday and now wants to give some of them to Masha. He knows that Masha cares more about the fact of receiving the present, rather than the value of that present, so he wants to give her stones as many times as possible. However, Masha remembers the last present she received, so Artem can't give her the same number of stones twice in a row. For example, he can give her 3 stones, then 1 stone, then again 3 stones, but he can't give her 3 stones and then again 3 stones right after that.

How many times can Artem give presents to Masha?

Input

The only line of the input contains a single integer n ($1 \le n \le 10^9$) — number of stones Artem received on his birthday.

Output

Print the maximum possible number of times Artem can give presents to Masha.

Examples

input
```
1
```
output
```
1
```

input
```
2
```
output
```
1
```

input
```
3
```
output
```
2
```

input
```
4
```
output
```
3
```

Note

In the first sample, Artem can only give 1 stone to Masha.

In the second sample, Atrem can give Masha 1 or 2 stones, though he can't give her 1 stone two times.

In the third sample, Atrem can first give Masha 2 stones, a then 1 more stone.

In the fourth sample, Atrem can first give Masha 1 stone, then 2 stones, and finally 1 stone again.

B. Little Artem and Grasshopper

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

Little Artem found a grasshopper. He brought it to his house and constructed a jumping area for him.

The area looks like a strip of cells $1 \times n$. Each cell contains the direction for the next jump and the length of that jump. Grasshopper starts in the first cell and follows the instructions written on the cells. Grasshopper stops immediately if it jumps out of the strip. Now Artem wants to find out if this will ever happen.

Input

The first line of the input contains a single integer n ($1 \leq n \leq 100\,000$) — length of the strip.

Next line contains a string of length n which consists of characters "<" and ">" only, that provide the direction of the jump from the corresponding cell. Next line contains n integers d_i ($1 \leq d_i \leq 10^9$) — the length of the jump from the i-th cell.

Output

Print "`INFINITE`" (without quotes) if grasshopper will continue his jumps forever. Otherwise print "`FINITE`" (without quotes).

Examples

input

```
2
><
1 2
```

output

```
FINITE
```

input

```
3
>><
2 1 1
```

output

```
INFINITE
```

Note

In the first sample grasshopper starts from the first cell and jumps to the right on the next cell. When he is in the second cell he needs to jump two cells left so he will jump out of the strip.

Second sample grasshopper path is 1 - 3 - 2 - 3 - 2 - 3 and so on. The path is infinite.

C. Little Artem and Matrix

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

Little Artem likes electronics. He can spend lots of time making different schemas and looking for novelties in the nearest electronics store. The new control element was delivered to the store recently and Artem immediately bought it.

That element can store information about the matrix of integers size $n \times m$. There are $n + m$ inputs in that element, i.e. each row and each column can get the signal. When signal comes to the input corresponding to some row, this row cyclically shifts to the left, that is the first element of the row becomes last element, second element becomes first and so on. When signal comes to the input corresponding to some column, that column shifts cyclically to the top, that is first element of the column becomes last element, second element becomes first and so on. Rows are numbered with integers from 1 to n from top to bottom, while columns are numbered with integers from 1 to m from left to right.

Artem wants to carefully study this element before using it. For that purpose he is going to set up an experiment consisting of q turns. On each turn he either sends the signal to some input or checks what number is stored at some position of the matrix.

Artem has completed his experiment and has written down the results, but he has lost the chip! Help Artem find any initial matrix that will match the experiment results. It is guaranteed that experiment data is consistent, which means at least one valid matrix exists.

Input

The first line of the input contains three integers n, m and q ($1 \le n, m \le 100, 1 \le q \le 10\,000$) — dimensions of the matrix and the number of turns in the experiment, respectively.

Next q lines contain turns descriptions, one per line. Each description starts with an integer t_i ($1 \le t_i \le 3$) that defines the type of the operation. For the operation of first and second type integer r_i ($1 \le r_i \le n$) or c_i ($1 \le c_i \le m$) follows, while for the operations of the third type three integers r_i, c_i and x_i ($1 \le r_i \le n$, $1 \le c_i \le m$, $-10^9 \le x_i \le 10^9$) are given.

Operation of the first type ($t_i = 1$) means that signal comes to the input corresponding to row r_i, that is it will shift cyclically. Operation of the second type ($t_i = 2$) means that column c_i will shift cyclically. Finally, operation of the third type means that at this moment of time cell located in the row r_i and column c_i stores value x_i.

Output

Print the description of any valid initial matrix as n lines containing m integers each. All output integers should not exceed 10^9 by their absolute value.

If there are multiple valid solutions, output any of them.

Examples

input

```
2 2 6
2 1
2 2
3 1 1 1
3 2 2 2
3 1 2 8
3 2 1 8
```

output

```
8 2
1 8
```

input

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

output

```
0 0 0
0 0 5
0 0 0
```

D. Little Artem and Dance

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

Little Artem is fond of dancing. Most of all dances Artem likes rueda — Cuban dance that is danced by pairs of boys and girls forming a circle and dancing together.

More detailed, there are n pairs of boys and girls standing in a circle. Initially, boy number 1 dances with a girl number 1, boy number 2 dances with a girl number 2 and so on. Girls are numbered in the clockwise order. During the dance different moves are announced and all pairs perform this moves. While performing moves boys move along the circle, while girls always stay at their initial position. For the purpose of this problem we consider two different types of moves:

1. Value x and some direction are announced, and all boys move x positions in the corresponding direction.
2. Boys dancing with even-indexed girls swap positions with boys who are dancing with odd-indexed girls. That is the one who was dancing with the girl 1 swaps with the one who was dancing with the girl number 2, while the one who was dancing with girl number 3 swaps with the one who was dancing with the girl number 4 and so one. It's guaranteed that n is even.

Your task is to determine the final position of each boy.

Input

The first line of the input contains two integers n and q ($2 \le n \le 1\,000\,000$, $1 \le q \le 2\,000\,000$) — the number of couples in the rueda and the number of commands to perform, respectively. It's guaranteed that n is even.

Next q lines contain the descriptions of the commands. Each command has type as the integer 1 or 2 first. Command of the first type is given as x ($-n \le x \le n$), where $0 \le x \le n$ means all boys moves x girls in clockwise direction, while $-x$ means all boys move x positions in counter-clockwise direction. There is no other input for commands of the second type.

Output

Output n integers, the i-th of them should be equal to the index of boy the i-th girl is dancing with after performing all q moves.

Examples

input

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

output

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

input

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

output

```
1 2
```

input

```
4 2
2
1 3
```

output

```
1 4 3 2
```

E. Little Artem and Time Machine

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

Little Artem has invented a time machine! He could go anywhere in time, but all his thoughts of course are with computer science. He wants to apply this time machine to a well-known data structure: *multiset*.

Artem wants to create a basic multiset of integers. He wants these structure to support operations of three types:

1. Add integer to the multiset. Note that the difference between set and multiset is that multiset may store several instances of one integer.
2. Remove integer from the multiset. Only one instance of this integer is removed. Artem doesn't want to handle any exceptions, so he assumes that every time remove operation is called, that integer is presented in the multiset.
3. Count the number of instances of the given integer that are stored in the multiset.

But what about time machine? Artem doesn't simply apply operations to the multiset one by one, he now travels to different moments of time and apply his operation there. Consider the following example.

- First Artem adds integer 5 to the multiset at the 1-st moment of time.
- Then Artem adds integer 3 to the multiset at the moment 5.
- Then Artem asks how many 5 are there in the multiset at moment 6. The answer is 1.
- Then Artem returns back in time and asks how many integers 3 are there in the set at moment 4. Since 3 was added only at moment 5, the number of integers 3 at moment 4 equals to 0.
- Then Artem goes back in time again and removes 5 from the multiset at moment 3.
- Finally Artyom asks at moment 7 how many integers 5 are there in the set. The result is 0, since we have removed 5 at the moment 3.

Note that Artem dislikes exceptions so much that he assures that after each change he makes all delete operations are applied only to element that is present in the multiset. The answer to the query of the third type is computed at the moment Artem makes the corresponding query and are not affected in any way by future changes he makes.

Help Artem implement time travellers multiset.

Input

The first line of the input contains a single integer n ($1 \le n \le 100\,000$) — the number of Artem's queries.

Then follow n lines with queries descriptions. Each of them contains three integers a_i, t_i and x_i ($1 \le a_i \le 3$, $1 \le t_i, x_i \le 10^9$) — type of the query, moment of time Artem travels to in order to execute this query and the value of the query itself, respectively. It's guaranteed that all moments of time are distinct and that after each operation is applied all operations of the first and second types are consistent.

Output

For each ask operation output the number of instances of integer being queried at the given moment of time.

Examples

input

```
6
1 1 5
3 5 5
1 2 5
3 6 5
2 3 5
3 7 5
```

output

```
1
2
1
```

input

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

output

```
0
```