15-295 Spring 2019 #11 Data Structures

A. Heap Operations **1 point**

1 second, 256 megabytes

Petya has recently learned data structure named "Binary heap".

The heap he is now operating with allows the following operations:

- put the given number into the heap;
- get the value of the minimum element in the heap;
- extract the minimum element from the heap;

Thus, at any moment of time the heap contains several integers (possibly none), some of them might be equal.

In order to better learn this data structure Petya took an empty heap and applied some operations above to it. Also, he carefully wrote down all the operations and their results to his event log, following the format:

- `insert` x — put the element with value x in the heap;
- `getMin` x — the value of the minimum element contained in the heap was equal to x;
- `removeMin` — the minimum element was extracted from the heap (only one instance, if there were many).

All the operations were correct, i.e. there was at least one element in the heap each time `getMin` or `removeMin` operations were applied.

While Petya was away for a lunch, his little brother Vova came to the room, took away some of the pages from Petya's log and used them to make paper boats.

Now Vova is worried, if he made Petya's sequence of operations inconsistent. For example, if one apply operations one-by-one in the order they are written in the event log, results of `getMin` operations might differ from the results recorded by Petya, and some of `getMin` or `removeMin` operations may be incorrect, as the heap is empty at the moment they are applied.

Now Vova wants to add some new operation records to the event log in order to make the resulting sequence of operations correct. That is, the result of each `getMin` operation is equal to the result in the record, and the heap is non-empty when `getMin` ad `removeMin` are applied. Vova wants to complete this as fast as possible, as the Petya may get back at any moment. He asks you to add the least possible number of operation records to the current log. Note that arbitrary number of operations may be added at the beginning, between any two other operations, or at the end of the log.

Input

The first line of the input contains the only integer n ($1 \leq n \leq 100\,000$) — the number of the records left in Petya's journal.

Each of the following n lines describe the records in the current log in the order they are applied. Format described in the statement is used. All numbers in the input are integers not exceeding 10^9 by their absolute value.

Output

The first line of the output should contain a single integer m — the minimum possible number of records in the modified sequence of operations.

Next m lines should contain the corrected sequence of records following the format of the input (described in the statement), one per line and in the order they are applied. All the numbers in the output should be integers not exceeding 10^9 by their absolute value.

Note that the input sequence of operations must be the **subsequence** of the output sequence.

It's guaranteed that there exists the correct answer consisting of no more than $1\ 000\ 000$ operations.

input

```
2
insert 3
getMin 4
```

output

```
4
insert 3
removeMin
insert 4
getMin 4
```

input

```
4
insert 1
insert 1
removeMin
getMin 2
```

output

```
6
insert 1
insert 1
removeMin
removeMin
insert 2
getMin 2
```

In the first sample, after number 3 is inserted into the heap, the minimum number is 3. To make the result of the first `getMin` equal to 4 one should firstly remove number 3 from the heap and then add number 4 into the heap.

In the second sample case number 1 is inserted two times, so should be similarly removed twice.

B. Knight Tournament **1.5 points**

3 seconds, 256 megabytes

Hooray! Berl II, the king of Berland is making a knight tournament. The king has already sent the message to all knights in the kingdom and they in turn agreed to participate in this grand event.

As for you, you're just a simple peasant. There's no surprise that you slept in this morning and were late for the tournament (it was a weekend, after all). Now you are really curious about the results of the tournament. This time the tournament in Berland went as follows:

- There are n knights participating in the tournament. Each knight was assigned his unique number — an integer from 1 to n.
- The tournament consisted of m fights, in the i-th fight the knights that were still in the game with numbers at least l_i and at most r_i have fought for the right to continue taking part in the tournament.
- After the i-th fight among all participants of the fight only one knight won — the knight number x_i, he continued participating in the tournament. Other knights left the tournament.
- The winner of the last (the m-th) fight (the knight number x_m) became the winner of the tournament.

You fished out all the information about the fights from your friends. Now for each knight you want to know the name of the knight he was conquered by. We think that the knight number b was conquered by the knight number a, if there was a fight with both of these knights present and the winner was the knight number a.

Write the code that calculates for each knight, the name of the knight that beat him.

Input

The first line contains two integers n, m $(2 \le n \le 3 \cdot 10^5;\ 1 \le m \le 3 \cdot 10^5)$ — the number of knights and the number of fights. Each of the following m lines contains three integers l_i, r_i, x_i $(1 \le l_i < r_i \le n;\ l_i \le x_i \le r_i)$ — the description of the i-th fight.

It is guaranteed that the input is correct and matches the problem statement. It is guaranteed that at least two knights took part in each battle.

Output

Print n integers. If the i-th knight lost, then the i-th number should equal the number of the knight that beat the knight number i. If the i-th knight is the winner, then the i-th number must equal 0.

input

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

output

```
3 1 4 0
```

input

```
8 4
3 5 4
3 7 6
2 8 8
1 8 1
```

output

```
0 8 4 6 4 8 6 1
```

Consider the first test case. Knights 1 and 2 fought the first fight and knight 1 won. Knights 1 and 3 fought the second fight and knight 3 won. The last fight was between knights 3 and 4, knight 4 won.

C. Pashmak and Parmida's problem **1.5 points**

4.5 s, 256 megabytes

Parmida is a clever girl and she wants to participate in Olympiads this year. Of course she wants her partner to be clever too (although he's not)! Parmida has prepared the following test problem for Pashmak.

There is a sequence a that consists of n integers $a_1, a_2, ..., a_n$. Let's denote $f(l, r, x)$ the number of indices k such that: $l \le k \le r$ and $a_k = x$. His task is to calculate the number of pairs of indicies i, j $(1 \le i < j \le n)$ such that $f(1, i, a_i) > f(j, n, a_j)$.

Help Pashmak with the test.

Input
The first line of the input contains an integer n $(1 \le n \le 10^6)$. The second line contains n space-separated integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$.

Output
Print a single integer — the answer to the problem.

input
```
7
1 2 1 1 2 2 1
```
output
```
8
```

input
```
3
1 1 1
```
output
```
1
```

input
```
5
1 2 3 4 5
```
output
```
0
```

D. A Lot of Games **2 points**

1 second, 256 megabytes

Andrew, Fedor and Alex are inventive guys. Now they invent the game with strings for two players.

Given a group of n non-empty strings. During the game two players build the word together, initially the word is empty. The players move in turns. On his step player must add a single letter in the end of the word, the resulting word must be prefix of at least one string from the group. A player loses if he cannot move.

Andrew and Alex decided to play this game k times. The player who is the loser of the i-th game makes the first move in the $(i+1)$-th game. Guys decided that the winner of all games is the player who wins the last (k-th) game. Andrew and Alex already started the game. Fedor wants to know who wins the game if both players will play optimally. Help him.

Input

The first line contains two integers, n and k ($1 \le n \le 10^5$; $1 \le k \le 10^9$).

Each of the next n lines contains a single non-empty string from the given group. The total length of all strings from the group doesn't exceed 10^5. Each string of the group consists only of lowercase English letters.

Output

If the player who moves first wins, print "`First`", otherwise print "`Second`" (without the quotes).

input
```
2 3
a
b
```
output
```
First
```

input
```
3 1
a
b
c
```
output
```
First
```

input
```
1 2
ab
```
output
```
Second
```

E. MEX Queries **2.5 points**

2 seconds, 256 megabytes

You are given a set of integer numbers, initially it is empty. You should perform n queries.

There are three different types of queries:

- `1` l r — Add all missing numbers from the interval $[l, r]$
- `2` l r — Remove all present numbers from the interval $[l, r]$
- `3` l r — Invert the interval $[l, r]$ — add all missing and remove all present numbers from the interval $[l, r]$

After each query you should output *MEX* of the set — the smallest positive (*MEX* ≥ 1) integer number which is not presented in the set.

Input

The first line contains one integer number n ($1 \leq n \leq 10^5$).

Next n lines contain three integer numbers t, l, r ($1 \leq t \leq 3, 1 \leq l \leq r \leq 10^{18}$) — type of the query, left and right bounds.

Output

Print *MEX* of the set after each query.

input

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

output

```
1
3
1
```

input

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

output

```
4
4
4
1
```

Here are contents of the set after each query in the first example:

1. $\{3, 4\}$ — the interval $[3, 4]$ is added
2. $\{1, 2, 5, 6\}$ — numbers $\{3, 4\}$ from the interval $[1, 6]$ got deleted and all the others are added
3. $\{5, 6\}$ — numbers $\{1, 2\}$ got deleted