

# Super Dice Game

Alice and Bob are playing a game. The game consists of a circular track of  $M$  ( $2 \leq M \leq 1,000,000,000$ ) cells labeled 0 through  $M - 1$ . Initially both players start at cell 0. The game progresses by having each player take turns rolling one of  $N$  ( $1 \leq N \leq 10,000$ ) 'super-dice' labeled 0 through  $N - 1$ . The actual mechanics of the 'super-dice' is not very well understood; however, it is known that they will only ever turn up a number between 0 and 1,000,000,000 inclusive after a roll. After rolling the super-dice the number of spaces a player moves is determined by the product of a contiguous subsequence of the values shown on the dice (There are special rules for determining the range that vary each move that will not be discussed).

To make matters more complicated, after any turn if Alice and Bob land on the same cell the value shown on all dice is multiplied by the label of the cell they are on. Note in this way it is possible for some dice to show numbers greater than 1,000,000,000. This multiplier does not apply to future rolls.

After playing this game for a while, Alice and Bob have grown frustrated because the calculations became too difficult. Given the series of  $R$  ( $1 \leq R \leq 100,000$ ) dice rolls and ranges, help Alice and Bob determine their position after each move. Assume that all dice start out showing 1.

## Input

The first line contains  $R$ ,  $N$ , and  $M$  each separated by a space.  $R$  lines follow. Each line will contain  $d$   $v$   $a$   $b$  separated by a space.  $d$  indicates the label of the dice rolled.  $v$  indicates the value shown on the dice.  $a$  and  $b$  indicate the range of dice used to determine the move distance.

## Output

$R$  lines containing the position of the player that just rolled after their roll.

## Example

### Input:

```
6 4 20
1 5 1 1
3 10 2 3
2 3 0 3
1 2 0 3
1 5 1 2
0 7 0 1
```

### Output:

```
5
10
15
10
10
0
```

### Output Explanation:

For your assistance, here is the state of the dice after each turn:

[1, 5, 1, 1]  
[1, 5, 1, 10]  
[1, 5, 3, 10]  
[1, 2, 3, 10]  
[10, 50, 30, 100]  
[7, 50, 30, 100]

**Warning: large Input/Output data, be careful with certain languages**