Levve Loves Segment Trees

Time limit: 1.0s **Memory limit:** 1G
Java: 2.0s
Python: 4.0s

> Levve loves segment trees, so he has given you the following task:

You are given an array of N **zeroes** and Q operations of the following forms:

- `C x v`: Change the element's value at index x to v.
- `S l r`: Output the sum of all elements between indices l and r, inclusive.
- `M l r`: Output the maximum of all elements between indices l and r, inclusive.

Levve doesn't like cheesing, so you will not be given l_i, r_i, x_i or v_i directly. Instead of l_i, you will be given l'_i, which can be decrypted using the formula $l_i = l'_i \oplus lastAns$ where $lastAns$ represents the output to the previous `S` or `M` query, and $\oplus$ represents the bitwise xor operation. If there is no previous output, $lastAns = 0$.

Similarly:

- $r_i = r'_i \oplus lastAns$
- $x_i = x'_i \oplus lastAns$
- $v_i = v'_i \oplus lastAns$

Constraints

$1 \leq N \leq 10^{18}$

$1 \leq Q \leq 4 \times 10^5$

$1 \leq x \leq N$

$0 \leq v \leq 10^9$

$1 \leq l \leq r \leq N$

Input Specification

The first line contains two space-separated integers, N and Q.

The following Q lines each contain a query of one of the previously described forms.

Output Specification

For each `S` or `M` query, output its result on a new line.

Sample Input

```
10 5
C 2 1
C 7 3
S 1 5
C 2 9
M 2 8
```

Sample Input (Unencrypted)

```
10 5
C 2 1
C 7 3
S 1 5
C 3 8
M 3 9
```

Sample Output

```
1
8
```