Dynamic Sum

Time limit: 1.0s **Memory limit:** 512M

Given an array of N numbers (initially, each number is set to 0), you need to answer Q queries of 2 types:

`1 i' v'`: Add v to the i^{th} number.

`2 l' r' x'`: Query the sum $[l, r]$ after undoing the x most recent type 1 queries (note: the undo queries are NOT persistent).

Constraints

$1 \leq N, Q \leq 300\,000$

$1 \leq i \leq N$

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

$0 \leq v \leq 10^9$

It is guaranteed that there have been at least x type 1 queries.

Input Specification

The first line will contain N and Q, the number of elements in the array and the number of queries.

The next Q lines will contain a query from those listed above.

Note that this problem will be **online enforced**, meaning that input will be given in an encrypted format. To encrypt the data, the values l, r, i, v, x in queries will be given as $l' = l \oplus \text{lastAns}, r' = r \oplus \text{lastAns}$, $i' = i \oplus \text{lastAns}$, $v' = v \oplus \text{lastAns}$, and $x' = x \oplus \text{lastAns}$, where $\oplus$ denotes the bitwise XOR operation. Note that $|\text{lastAns}|$ at any time is defined as the answer to the latest query. If no queries have occurred so far, $\text{lastAns} = 0$.

Output Specification

Output the answer to each type 2 query.

Sample Input (Encrypted)

```
5 6
1 1 5
1 5 9
2 1 5 0
2 15 11 15
1 6 7
2 4 0 5
```

Sample Input (Unencrypted)

```
5 6
1 1 5
1 5 9
2 1 5 0
2 1 5 1
1 3 2
2 1 5 0
```

Sample Output

```
14
5
16
```