Yet Another Contest 7 P6 - Arithmetic Sequence Data Structure

Time limit: 4.0s	**Memory limit:** 256M
Java: 7.0s	Java: 512M
Python: 8.0s	Python: 512M

Mike likes arithmetic sequences! So, he wants to create a data structure which maintains an array of N integers, each initially equal to 0, which supports three types of operations:

- `1 L R X V` Increase the elements at positions $L, L+X, L+2X, L+3X, \ldots, R$ by V. It is guaranteed that $R-L$ is a multiple of X.
- `2 L R X V` Set the elements at positions $L, L+X, L+2X, L+3X, \ldots, R$ to V. It is guaranteed that $R-L$ is a multiple of X.
- `3 Y` Output the value of the element at position Y.

Unfortunately, Mike does not know how to implement such a data structure. Can you help him implement this data structure and perform Q operations on it?

Constraints

$1 \leq N, Q \leq 5 \times 10^5$

$1 \leq L \leq R \leq N$

$1 \leq X < N$

$1 \leq Y \leq N$

$R-L$ is a multiple of X.

$-10^9 \leq V \leq 10^9$

Subtask 1 [15%]

$X = 2$

Subtask 2 [5%]

$1 \leq X \leq 2$

Subtask 3 [30%]

$1 \leq N, Q \leq 2 \times 10^5$

There are no operations of type 2.

Subtask 4 [25%]

There are no operations of type 2.

Subtask 5 [15%]

$1 \leq N, Q \leq 2 \times 10^5$

Subtask 6 [10%]

No additional constraints.

Input Specification

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

The i-th of the following Q lines denotes the i-th operation, in one of the three formats described above.

Output Specification

For each operation of type 3, output the value of the queried element on a separate line.

Sample Input

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

Sample Output

```
4
6
```

Explanation

The initial array is $[0, 0, 0, 0, 0]$.

The first operation increases the values of the elements at positions 1, 3 and 5 by 4. The array becomes $[4, 0, 4, 0, 4]$.

The second operation sets the values of the elements at positions 2, 3 and 4 to 6. The array becomes $[4, 6, 6, 6, 4]$.

The third operation queries the value of the element at position 1, which is 4.

The fourth operation queries the value of the element at position 2, which is 6.