

D. Zayin and Forest

We define $B(x)$ as the number of digit 1 in the binary representation of x . For example, $B(7) = B((111)_2) = 3$, $B(8) = B((1000)_2) = 1$, $B(9) = B((1001)_2) = 2$.

We define $F(x) = \min\{y \mid (y > x) \wedge (B(y) \leq B(x))\}$. For example, $F(4) = 8$, $F(5) = 6$, $F(6) = 8$.

Zayin has a forest (a set of trees), whose nodes are numbered from 1 to n . For each node (e.g. node x), if $F(x)$ is not greater than n , then the father of node x is $F(x)$, else, node x is the root of a tree.

We use $A[i]$ to denote the number of apples on the node i . Initially, all the $A[i](1 \leq i \leq n)$ are equal to zero. In order to make his girlfriend happy, Zayin is going to perform magic on the tree. The magic consists of two types of operations:

1. Add $x v$: For every ancestor of node x (including itself), put v more apples on it. In other words, for every node (e.g. node y) on the path between node x and the root of its tree, let $A[y] = A[y] + v$.
2. Sum $L R$: Count the total number of apples on the nodes whose index is between L and R . In other words, you need to calculate $\sum_{i=L}^R A[i]$.

Input

The first line contains two integers n and m ($1 \leq n \leq 10^{18}, 1 \leq m \leq 10^5$), where n is the number of nodes and m is the number of operations.

The next m lines describe the m operations in order. Each line contains three integers. The first of them is op . If $op = 1$, then the next two integers will be x and v , representing an **Add** operation. If $op = 2$, then the next two integers will be L and R , representing a **Sum** operation. ($1 \leq x \leq n, 1 \leq v \leq 10^9, 1 \leq L \leq R \leq n$)

Output

For each **Sum** operation, output one line including one number, denoting the corresponding result.

Sample

Input	Output
8 6 1 1 1 1 3 2 1 5 3 1 7 4 2 1 5 2 4 8	10 23
1000000000000000000 2 1 1 1 2 1 1000000000000000000	60