Initially, there is an empty tree. You add n nodes to the tree, one by one.

After each node is added, print the number of accessible node pairs.

Two different nodes i and j are accessible if and only if $dist(i, j) \leq r(i) + r(j)$, where $dist(i, j)$ is the length of unique path from i and j.

Note that a node and itself is NOT an accessible node pair.

Nodes are numbered 1, 2, 3, ... in the same order as they are added.

Input

The first line contains n ($2 \leq n \leq 100000$), the number of total nodes.

There are n lines followed. The i-th line contains three integer $a(i)$, $c(i)$, $r(i)$, that means node i is connected with node $f(i) = a(i) XOR (last\_ans \bmod 10^9)$, edge weight is $c(i)$, range value is $r(i)$ ($1 \leq r(i) \leq 10^9$).

Note that node 1 is not connected with any node, so we define $a(1) = c(1) = 0$. For other nodes (i.e. $i \geq 2$), $1 \leq f(i) < i$, $1 \leq c(i) \leq 10000$, $0 \leq a(i) \leq 2 * 10^9$. For each test case, $last\_ans$ is initially 0.

Output

The output for each test case contains $n+1$ lines. The first line contains the case number, the $(i+1)$-th line is the number of accessible pairs after node i is added. Print a blank line after each test case (including the last one).

Sample Input

```
5
0 0 6
1 2 4
0 9 4
0 5 5
0 2 4
5
0 0 6
1 2 4
0 9 4
0 5 5
0 2 4
0
```

Sample Output

```
Case 1:
0
1
2
4
7

Case 2:
0
1
2
4
7
```