

Help Me Please!

One day while chilling in college, Deepankar and Rishabh were discussing some extremely complex topics which are beyond the comprehension of normal people. Just then, Amit meets up with them and listens in to what they have to say. It turns out they were discussing a problem. Though (as expected), the problem blew Amit's mind. He had no clue what to do with it or how to go about it. So it's up to you to help Amit in solving this problem.

Problem Statement

You are initially given a single node (numbered as 1) having a value of 0. Now you will be given queries of the following types :

1. **ADD** : Add a new node to the graph having some integer value , by making an edge from a pre-existing node to this node.
2. **RETURN** : Return the net value added to a subtree rooted at a node X between time A and B (inclusive) .

The initial node is considered the root of the tree that is formed.

Each ADD query will take 1 time unit while each RETURN query will take 0 time.

The clock will start from time 1, i.e the current time before all queries is 1 , after first ADD it will be 2 and so on.

The number of any node inserted into the graph is based on the order the nodes were put in the graph. The initial node is 1, the next inserted node is 2 , the next is 3 and so on.

The value of any node inserted into the graph is $V + \text{answer for previous RETURN query}$, where V is given in ADD query. If no RETURN queries have been done yet then we take only value of V as value of node.

Since the net value may become large , print the answer to each query as the remainder on its division by **1000000007**.

Input

The first line contains a single integer **Q** , denoting the number of queries.

The remaining lines represent each query. The first number in each query line denotes the type of query , **1** for **ADD** and **2** for **RETURN**. In case of an **ADD** query you will have 2 more numbers in the line , **X** and **V** , where **X** is the number of the node to which the current node is joined and meaning of **V** is as mentioned in the problem statement. In case of a **RETURN** query you will have 3 more numbers in the line , **A** , **B** and **X** , where their meaning is as defined in the problem statement.

It can be assumed that for an **ADD** query node **X** will always be a node which exists in the graph.

Output

For each **RETURN** query output a single line containing the answer for that query.

Constraints

$1 \leq Q \leq 200000$

$0 \leq V \leq 1000000000$

Example

Input:

3

1 1 10

2 1 2 1

1 1 5

Output:

10