



Task Klasika

In the beginning there was a node denoted as 1 and it represented the root of a tree. Your task is to support Q queries of the form:

- **Add** x y – Adds a new node to the tree as a child of node x . The newly added node and node x are connected with an edge of weight y . The newly added node is denoted by a number equal to the number of nodes that the tree consists of after its addition.
- **Query** a b – Finds the longest path in a tree which starts in node a and ends in some node from the subtree of node b (which itself is considered to be in its own subtree). The length of the path is defined as exclusive or (xor) of weights of all edges that the path consists of.

Input

The first line contains an integer Q ($1 \leq Q \leq 200\,000$) from the task description.

The i -th of the next Q lines contains the i -th query whose format corresponds to one of the queries from the task description. Values x , a and b will refer to an existing node at that moment and value y will not be greater than 2^{30} .

Output

You should output an answer to each query of type **Query**. Each answer should be printed in a separate line in the order in which corresponding queries appear in the input.

Scoring

Subtask	Score	Constraints
1	11	$Q \leq 200$
2	22	$Q \leq 2\,000$
3	33	In all queries of type Query it holds $b = 1$
4	44	No additional constraints.



Examples

input

```
4
Add 1 5
Query 1 1
Add 1 7
Query 1 1
```

output

```
5
7
```

input

```
6
Add 1 5
Add 2 7
Add 1 4
Add 4 3
Query 1 1
Query 2 4
```

output

```
7
2
```

input

```
10
Add 1 4
Add 1 9
Add 1 10
Add 2 2
Add 3 3
Add 4 4
Query 4 2
Query 1 3
Add 6 7
Query 1 3
```

output

```
14
10
13
```