

Problem A. AVL Tree

Time limit	1000 ms
Mem limit	1572864 kB
Code length Limit	50000 B
OS	Linux

This problem is an extension problem (with a little modification) from this problem: [SDITSBST](#).

In this problem you are given two type of query

1. Insert an integer to the list.
2. Given an integer x , you're about to find an integer k which represent x 's index if the list is sorted in ascending order. Note that in this problem we will use 1-based indexing.

As the problem title suggest, this problem intended to be solved using Balanced Binary Search Tree, one of its example is AVL Tree.

Input

The first line contains an integer Q , which denotes how many queries that follows.

The next Q lines will be one of the type queries which follow this format:

1 x means insert x to the list

2 x means find x 's index if the list is sorted in ascending order.

Output

For each query type 2, print a line containing an integer as the answer or print "Data tidak ada" no quotes if the requested number does not exist in the current lis.

Example

Input:

```
10
1 100
1 74
2 100
2 70
1 152
1 21
1 33
2 100
```

2 21

2 1

Output:

2

Data tidak ada

4

1

Data tidak ada

Explanation

Until the third query, the current list is {74, 100}. Therefore you must print 2 as 100 is on the first index.

Arriving at the fourth query we haven't add any other number so the list still consists of {74, 100}. Since 70 is not in the list you must print "Data tidak ada" remember no quotes.

For the last three queries the list looks like this {21, 33, 74, 100, 152}

So the answer for the eighth, ninth, and tenth query respectively are 4, 1, and "Data tidak ada".

Constraints

$$1 \leq Q \leq 200000$$

$$1 \leq x \leq 10^6$$

It is guaranteed that all integer that inserted in the list will be distinct.

Notes

There's no guarantee that the input will results a balanced tree i.e. you have to balanced it yourself :)