

Ternary Search Tree

Time limit: 1.0s **Memory limit:** 256M

Antonio likes top trees, and hates all other types of trees. One day, his algorithms professor gives him an assignment to build a ternary search tree. The algorithm for building a ternary search tree is as follows:

A ternary search tree has a root node, and each node has three children, a left child, a middle child, and a right child. Each node also has an integer key associated with it. To insert a node with a given key into the tree, check if the given key is less than, equal to, or greater than the key of the root of the tree. If it is less than the given key, insert it into the left subtree (or make it the root of the left subtree if it doesn't exist). If it is greater than the given key, insert it into the right subtree (or make it the root of the right subtree if it doesn't exist). If it is equal to the given key, insert it into the middle subtree (or make it the root of the middle subtree).

Help Antonio build a ternary search tree!

Constraints

$$1 \leq N \leq 10^3$$

$$0 \leq v_i \leq 10^9$$

Input Specification

The first line contains a single integer, N , the number of integers to be inserted in the ternary search tree.

The next line contains a single integer, v_1 , the initial value at the root of the ternary search tree.

The next $N - 1$ lines each contain a single integer v_i , an integer to insert into the ternary search tree. These insertions are to be processed in the order they are presented.

Output Specification

Output $N - 1$ lines. On the i th line, output the key of the parent of the node that was just inserted.

Sample Input

```
6
2
1
3
1
2
3
```

Sample Output

2
2
1
2
3