IOITC 2015 Finals, Day 1

Chandragupta and the fun with BST

According to some historians, in ancient India, people already had discovered data structures like Binary Search Trees(BST) and used them to store and search data easily. Now, Chandragupta is fascinated with BSTs which store positive integers. He is thinking of testing Chanakya on BSTs.
First, let us see how integer BSTs work. A BST is a rooted binary tree(in a binary tree each node has at most two children), whose nodes each store a key and each have atmost two distinguished sub-trees, commonly denoted left and right. The tree additionally satisfies the binary search tree property, which states that the key in each node must be greater than all keys stored in the left sub-tree, and smaller than all keys in right sub-tree.

For inserting a number into an already existing BST, we use the following pseudo code:

```
insert(Node root, int data)
{
     if (root doesn't exist)
           root = new Node(data);

     else if (data < root->data)
           insert(root->left, data);

     else if (data > root->data)
           insert(root->right, data);
}
```

Note that you can assume that Chandragupta only inserts distinct values into the BST. Now, Chandragupta defines a new numbering criteria of nodes in a BST. He gives number 1 to the root node and for each node with number i, he gives numbers $2 \cdot i$ and $2 \cdot i + 1$ to it's left and right child respectively, if they exist. For example, the following shows the numbering criteria of a complete binary search tree with 7 nodes.

```
              1
             / \
            /   \
           /     \
          /       \
         2         3
        / \       / \
       /   \     /   \
      4     5   6     7
```

Now, he defines a function F where $F(x)$ returns the node number of the node where key x is stored. This function's domain includes only those values which are present in the BST.

Arriving to challenging Chanakya on BSTs, Chandragupta gives him N distinct integers $A_1, A_2, ..., A_N$, which he has to insert one by one into a BST(which is initially empty) in the order they arrive in the input. After all the insertions have been done, Chanakya has to report N integers $F(A_1), F(A_2), ..., F(A_N)$. Since these numbers can be large, output each of them after modulo $10^9 + 7$.

Input

The first line of input will contain integer N *i.e.*, the number of integers in array A.

The second line contains N space separated integers denoting array A.

Output

Output N space separated integers where i'th number denotes $F(A_i)$ modulo $10^9 + 7$.

Test Data

In all the subtasks,
$1 \leq A_i \leq 10^9$

Subtask 1 (30 Points): $1 \leq N \leq 1000$.

Subtask 2 (70 Points): $1 \leq N \leq 3 * 10^5$.

Sample Input1

```
3
4 2 7
```

Sample Output1

```
1 2 3
```

Sample Input2

```
3
4 6 5
```

Sample Output2

```
1 3 6
```

Limits

Time: 1 second

Memory: 256 MB