

A binary search tree is a tree in which every node has **at most** two children nodes (a left and a right child). Each node has an integer written inside it. If the number  $X$  is written inside a node, then the numbers in its left subtree are less than  $X$  and the numbers in its right subtree are greater than  $X$ .

You will be given a sequence of integers between 1 and  $N$  (inclusive) such that each number appears in the sequence exactly once. You are to create a binary search tree from the sequence, putting the first number in the root node and inserting every other number in order. In other words, run  $\text{insert}(X, \text{root})$  for every other number:

```
insert( number X, node N )
    increase the counter C by 1
    if X is less than the number in node N
        if N has no left child
            create a new node with the number X and set it to be the left child of node N
        else
            insert(X, left child of node N)
    else (X is greater than the number in node N)
        if N has no right child
            create a new node with the number X and set it to be the right child of node N
        else
            insert(X, right child of node N)
```

Write a program that calculates the value of the counter  $C$  after every number is inserted. The counter is initially 0.

### **INPUT**

The first line contains the integer  $N$  ( $1 \leq N \leq 300000$ ), the length of the sequence.

The remaining  $N$  lines contain the numbers in the sequence, integers in the interval  $[1, N]$ . The numbers will be distinct.

### **OUTPUT**

Output  $N$  integers each on its own line, the values of the counter  $C$  after each number is inserted into the tree.

### **SCORING**

In test cases worth 50% of points,  $N$  will be at most 1000.

**EXAMPLES**

<pre>input 4 1 2 3 4 output 0 1 3 6</pre>	<pre>input 5 3 2 4 1 5 output 0 1 2 4 6</pre>	<pre>input 8 3 5 1 6 8 7 2 4 output 0 1 2 4 7 11 13 15</pre>
---	---	--