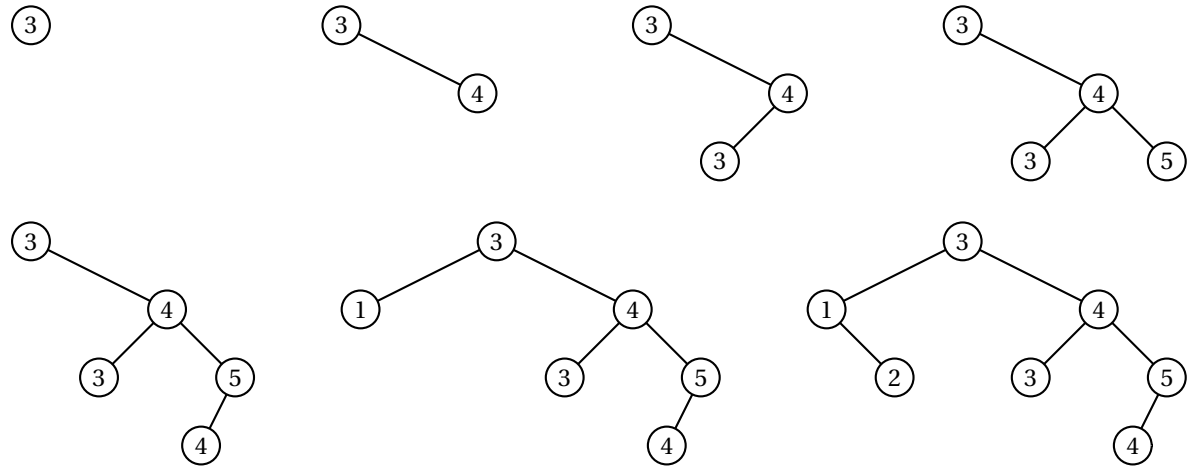
Tree Insertion

(problem given at CTU Open 2008)

All modern banks employ information systems to process their data. The amount of data is enormous. Imagine all the transactions, payments, e-banking, web services, etc. Therefore, advanced data structures must be used to store the data and allow to access them very quickly.

A Binary Search Tree (BST) is one such a data structure. It holds a collection of values that can be totally ordered with some comparison operation. A BST consists of nodes. A node contains one value and connects to at most two subtrees. The left subtree contains values strictly less than the node value, the right subtree contains values greater than or equal to the node value. As a consequence, values can be easily be looked up as in a binary search. We begin with the root node and compare its value with the value we are searching for. Depending on the result, we descend either into the left or into the right subtree, but we never need to go through both.

Inserting values is also simple. The first value in an empty tree goes to the root node. Then, to insert a value in a non-empty tree, we start with the root and descend through the tree as in the search procedure. When the descent leads to a missing subnode, we create a new leaf node with the new value at that position. The figures below show the tree after subsequently adding the following sequence of numbers: 3, 4, 3, 5, 4, 1, and 2.



Input

The input consists of several trees, each of them specified on two lines. The first line contains a single integer N ($1 \le N \le 100$), the number of values in the tree. The second line contains the N values separated by a space. These values, if inserted in the given order, form a BST to be examined. All values are between 0 and 1000. The last tree is followed by a line containing a single zero.

Output

For each tree, output the total number of different permutations that would generate the same Binary Search Tree. Notice that these numbers may well exceed 2^{32}.

Examples

Sample input 1

```
5
3 4 3 5 4
7
3 4 3 5 4 1 2
31
16 8 24 4 12 20 28 2 6 10 14 18 22 26 30 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
0
```

Sample output 1

```
3
45
7483682586183598080000 0
```

Limits

Time limit is 1 second.

Memory limit is 1024 megabytes.