

**Indian Institute of technology, Guwahati**  
**Department of Computer Science and Engineering**  
**Data Structure Lab: (CS210)**  
**Offline Assignment: 11**

**Date: 6<sup>th</sup> November 2017.**

**Total Marks: 20**

**Deadline: 10PM, 12th November 2017. (Hard Deadline)**

1. **[Print All BSTs]** You studied Binary Search Tree (BST) and its traversal methods. Now, as you are given a preorder traversal sequence of a BST. It is guaranteed that all the keys of this BST will be distinct positive integers.  
Somehow, you created actual BST on a paper from its given preorder traversal.  
Now, being a curious student, you want to know that how many different sequences of same keys (which are in given preorder traversal) can generate this BST. And you want to print all these sequences. For example, if you think about a new sequence, you take a new empty BST and then insert every key of the new sequence in this BST one by one. If this new BST is same as the original BST you had, then you will count this new sequence in your answer. Note that all your found sequences will be a permutation of initially given preorder traversal. Two sequences are different if they are different permutations.  
Your task is to first print the number of different possible such sequences. Then print all those sequences in a lexicographic order, one by one.

**Input Format:**

The first line will contain the number of keys  $N$  of BST.

The next line will contain  $N$  space separated positive integers, denoting preorder traversal sequence of the BST.

**Output Format:**

In the first line, output the number of possible sequences (which follow the problem statement).

Sort these  $C$  sequences in lexicographic order and print each sequence in a single line i.e. space separated keys of this sequence.

**Input Constraint:**

$1 \leq N \leq 1000$

Given preorder will be a valid preorder traversal sequence of a BST.

All keys of preorder traversal sequence will be distinct positive integers between 1 and 1000.

**Test:**

Input:

5  
3 1 2 4 5

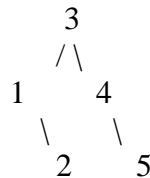
Output:

6  
3 1 2 4 5

3 1 4 2 5  
 3 1 4 5 2  
 3 4 1 2 5  
 3 4 1 5 2  
 3 4 5 1 2

**Explanation:**

From the given preorder traversal, the following unique BST can be created.



If we use same set of keys then there can be total 6 sequences, we will create same BST (after inserting keys in a new empty BST). In the output these sequences are in lexicographically sorted order.

2. **[Arrange the Sequence]** You are given a sequence S of positive integers. Say the distinct elements of this sequence are  $a_1, a_2, \dots, a_n$  and their corresponding number of occurrences in S are  $c_1, c_2, \dots, c_n$  respectively.

Your task is to arrange S, such that the element with highest number of occurrences comes first along with all its occurrences, then the element with second highest number of occurrences along with all its occurrences and so on...

If two elements have same number of occurrences, then all occurrences of smaller element will come first. Use only any version of binary tree to solve this problem.

**Input:**

In first line, an integer denoting size of sequence.

In the next line, the space separated integers denoting sequence.

**Output:**

Print the space separated integers of the desired sequence.

**Input Constraint:**

Size of sequence will be between 1 and 1000.

Each element of given sequence will be a positive integer between 1 and 1000.

**Test:**

Input

9

5 5 5 3 3 3 1 7 7

Output

3 3 3 5 5 5 7 7 1

**Explanation:**

Distinct elements are 5, 3, 1 and 7 with number of occurrences 3, 3, 1 and 2 respectively.  
5 and 3 have same number of occurrences but 3 is smaller than 5. So the output is 3 occurrences of 3, then 3 occurrences of 5, then 2 occurrences of 7 and then 1 occurrence of 1.