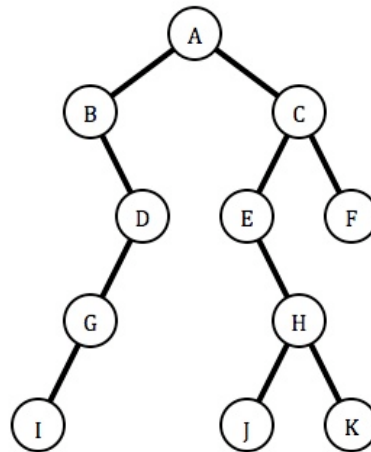


TREES

Problem 1. Write the pre-order, in-order and post-order traversals of the binary tree shown below.



Problem 2. Design an algorithm to test whether a binary tree is a binary search tree.

Problem 3. Create BST

Create a Binary Search Tree from list A containing N elements. Insert elements in the same order as given. Print the pre-order traversal of the subtree with root node data equal to Q (inclusive of Q), separating each element by a space.

Implement the following functions on binary trees:

- Count the number of nodes in a BST.
- Count the number of leaves in a BST.
- Find the height of the BST.
- Print the pre-order traversal of BST
- Print the post-order traversal of BST
- Print the in-order traversal of BST
- Print the pre-order traversal of the subtree with the root node data equal to Q (inclusive of Q), separating each element by a space.

Example:

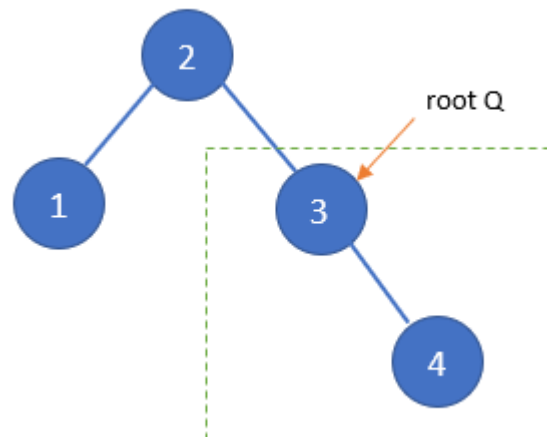
Input:

- First line contains a single integer N – number of elements
- Second line contains N space-separated integers
- Third line contains a single integer Q – the element whose subtree is to be printed in pre-order form.

Output: Print K space-separated integers – where K is the number of elements in the subtree of Q (inclusive of Q)

Input	Output
4 2 1 3 4 3	3 4

Explanation:



- h) Delete all leaves from a BST.
- i) Delete a node of BST.