

# Algorithm Lab (Course Code: MC504)

## Assignment - 5

**Submission Deadline:** Within class timing, (10/02/2023)

**Total Marks:** 50

---

### Instructions:

- Proper indentation is mandatory.
  - Program files **must** be compiled using **linux gcc compiler**.
  - **VERY IMPORTANT:** You must add comments whenever necessary, to make the code understandable.
  - Markings will be based on the correctness and soundness of the outputs. Marks will be deducted in case of plagiarism.
  - Take inputs from users. Make necessary assumptions if required.
  - **ANSWER FILE:** Source code: (file name) e.g. A5\_Q1.c, A5\_PP.c
- 

### Q1.

Given two integer arrays `preorder` and `inorder` where `preorder` is the preorder traversal of a binary tree and `inorder` is the inorder traversal of the same tree, **construct and return the binary tree**. A binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child.

In a Preorder sequence, the leftmost element is the *root* of the tree. By searching that *root* in the Inorder sequence, we can find out that all elements on the left side of *root* are in the left subtree, and elements on right in the right subtree. Now using this logic perform the above task.

Definition for a binary tree node to be followed is:

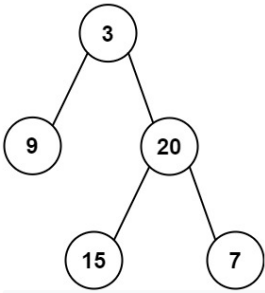
```
struct TreeNode {
    int val;
    struct TreeNode *left;
    struct TreeNode *right;
};
```

**Example:****Input:**

```
preorder = [3,9,20,15,7], inorder = [9,3,15,20,7]
```

**Output:** `[3,9,20,null,null,15,7]`

**Explanation:** The output array is the binary tree with root as 3. The next two elements(9 and 20) are the children of root. The next two elements (null and null) are children of 9. Again the next two elements (15 and 7) are children of 20. So the tree is represented by an array in this form.

**Input:**

```
preorder = [-1], inorder = [-1]
```

**Output:** `[-1]`**Q2.**

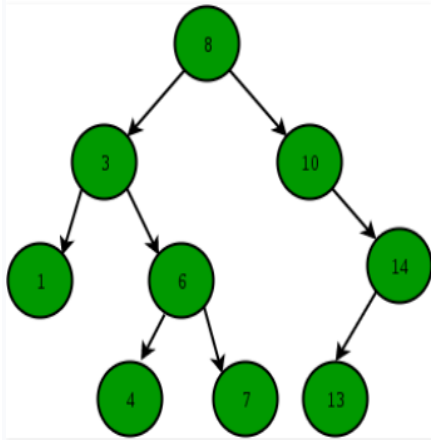
Write a C program to implement inorder traversal of a Binary Search Tree(BST) using linked list. Take the input from the user.

The properties that separate a binary search tree from a regular binary tree is

1. All nodes of left subtree are less than the root node
2. All nodes of right subtree are more than the root node
3. Both subtrees of each node are also BSTs i.e. they have the above two properties

**Example:****Input:** `[8,3,10,1,6,14,4,7,13]`**Output:** `[1,3,4,6,7,8,10,13,14]`

**Explanation:** From the input array the first element is taken as root. Then a BST is constructed



Now its Inorder traversal is displayed as output.

### **PRACTICE PROBLEM:**

In Q2 extend the functionality of BST. Your code should support the insertion of a new node to the BST and deletion of the existing node from BST. Design, Develop and Implement these functionality as menu driven Program for end user. For visualization of changes, implement display/print functionality also. Make assumptions wherever necessary with proper comments.