# Rajalakshmi Engineering College

Name: Adhithya varun
Email: 240801008@rajalakshmi.edu.in
Roll no: 240801008
Phone: null
Branch: REC
Department: I ECE FA
Batch: 2028
Degree: B.E - ECE

## NeoColab_REC_CS23231_DATA STRUCTURES

## REC_DS using C_Week 5_COD_Question 2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

## Section 1 : Coding

1.  Problem Statement

Mike is learning about Binary Search Trees (BSTs) and wants to implement various operations on them. He wants to write a basic program for creating a BST, inserting nodes, and printing the tree in the pre-order traversal.

Write a program to help him solve this program.

*Input Format*

The first line of input consists of an integer N, representing the number of values to insert into the BST.

The second line consists of N space-separated integers, representing the values to insert into the BST.

*Output Format*

The output prints the space-separated values of the BST in the pre-order traversal.

Refer to the sample output for formatting specifications.

*Sample Test Case*

Input: 5
3 1 5 2 4

Output: 3 1 2 5 4

*Answer*

```c
#include <stdio.h>
#include <stdlib.h>

struct Node {
    int data;
    struct Node* left;
    struct Node* right;
};

struct Node* createNode(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->left = newNode->right = NULL;
    return newNode;
}
struct Node* insert(struct Node* root, int value){
    if (root == NULL)
{
        return createNode(value);
}
    if (value < root->data)
{
        root->left = insert(root->left, value);
} else if (value > root->data)
{
        root->right = insert(root->right, value);
}
```

```c
        return root;
    }
    void printPreorder(struct Node* root)
    {
        if (root != NULL)
        {
            printf("%d ", root->data);
            printPreorder(root->left);
            printPreorder(root->right);
        }
    }

    int main() {
        struct Node* root = NULL;

        int n;
        scanf("%d", &n);

        for (int i = 0; i < n; i++) {
            int value;
            scanf("%d", &value);
            root = insert(root, value);
        }

        printPreorder(root);
        return 0;
    }
```

*Status :* Correct                                                    *Marks : 10/10*