

# Rajalakshmi Engineering College

Name: KAVYA SRIRAM

Email: 241901045@rajalakshmi.edu.in

Roll no: 241901045

Phone: 8939657782

Branch: REC

Department: I CSE (CS) FA

Batch: 2028

Degree: B.E - CSE (CS)

Scan to verify results



## 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**

```
#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 {  
        root->right = insert(root->right, value);  
    }
```

```
    return root;
```

```
}
```

```
void printPreorder(struct Node* node) {  
    if (node == NULL) return;
```

```
    printf("%d ", node->data);  
    printPreorder(node->left);  
    printPreorder(node->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