

# Rajalakshmi Engineering College

Name: Tejeshwaran P

Email: 241801292@rajalakshmi.edu.in

Roll no: 241801292

Phone: 6383048813

Branch: REC

Department: I AI & DS AF

Batch: 2028

Degree: B.E - AI & DS

Scan to verify results



## NeoColab REC CS23231 DATA STRUCTURES

REC\_DS using C\_Week 5\_PAH\_Updated

Attempt : 1

Total Mark : 50

Marks Obtained : 42.5

### Section 1 : Coding

#### 1. Problem Statement

Joseph, a computer science student, is interested in understanding binary search trees (BST) and their node arrangements. He wants to create a program to explore BSTs by inserting elements into a tree and displaying the nodes using post-order traversal of the tree.

Write a program to help Joseph implement the program.

#### *Input Format*

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

The second line  
separated integers  
be inserted into the

#### *Output Format*

The output prints N  
integer values after

consists of N  
data, which  
BST.

space-  
the post-

space-  
is the data to

separated  
order

Refer to the sample output for formatting specifications.

#### *Sample Test Case*

Input: 4

10 15 5 3

Output: 3 5 15 10

#### *Answer*

```
// You are using GCC
#include <iostream>
using namespace std;
```

```
// Structure of a Binary Search Tree Node struct
Node {
    int data;    Node* left;
    Node* right;
    Node(int value) : data(value), left(nullptr),
};
```

right(nullptr)

```
// Function to insert a new node with given data
```

```
Node* insert(Node* root, int data) {
    if (root == nullptr)    return new Node(data);
    if (data < root->data)
        root->left = insert(root->left, data);    else
        root->right = insert(root->right, data);    return root;
```

```
}
```

```
// Function to  
post-order traversal
```

```
postOrder(Node*  
(root == nullptr)
```

```
postOrder(root->
```

```
postOrder(root->
```

```
cout << root->data <<
```

```
}
```

```
perform
```

```
void
```

```
root) { if
```

```
>left);
```

```
>right);
```

```
" ";
```

```
int main() { int N;
```

```
cin >> N; // Read the number of elements Node*
```

```
root = nullptr;
```

```
// Insert elements into the BST for (int i = 0; i <  
N; ++i) { int value; cin >> value;
```

```
root = insert(root, value);
```

```
// Output the post-order traversal  
postOrder(root); cout << endl;
```

```
return 0; }
```

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