Rajalakshmi Engineering College

*241901092*

*241901092*

Name: H.Sadhaa sivam

*241901092*

*241901092*

Email: 241901092@rajalakshmi.edu.in

Roll no: 241901092

Phone: 6383796684

Branch: REC

Department: l CSE (CS) FB

Batch: 2028

Degree: B.E - CSE (CS)

NeoColab\_REC\_CS23231\_DATA STRUCTURES

*241901092*

*241901092*

REC\_DS using C\_Week 5\_COD\_Question 4

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

John, a computer science student, is learning about binary search trees (BST) and their properties. He decides to write a program to create a BST, display it in post-order traversal, and find the minimum value present in the tree.

*241901092*

*241901092*

*241901092*

*241901092*

Help him by implementing 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 consists of N space-separated integers data, which is the data to be inserted into the BST.

*241901092*

*241901092*

*241901092*

*241901092*

*Output Format*

The first line of output prints the space-separated elements of the BST in postorder traversal.

*241901092*

*241901092*

*241901092*

*241901092*

The second line prints the minimum value found in the BST.

Refer to the sample output for formatting specifications.

*Sample Test Case*

Input: 3 5 10 15

*241901092*

Output: 15 10 5

The minimum value in the BST is: 5

*241901092*

*241901092*

*241901092*

*Answer*

#include <stdio.h>

#include <stdlib.h>

struct Node { int data; struct Node\* left;

struct Node\* right;

};

struct Node\* createNode(int data) {

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node)); newNode->data = data;

*241901092*

*241901092*

*241901092*

*241901092*

newNode->left = newNode->right = NULL; return newNode;

}

// You are using GCC

struct Node\* insert(struct Node\* root, int data) { if (root == NULL) { return createNode(data);

*241901092*

*241901092*

}

if (data < root->data) {

root->left = insert(root->left, data); } else {

*241901092*

*241901092*

root->right = insert(root->right, data);

*241901092*

*241901092*

void displayTreePostOrder(struct Node\* root) { if (root == NULL) return;

}

return root;

}

*241901092*

*241901092*

displayTreePostOrder(root->left); displayTreePostOrder(root->right); printf("%d ", root->data);

}

*241901092*

*241901092*

int findMinValue(struct Node\* root) { struct Node\* current = root; while (current->left != NULL) {

*241901092*

*241901092*

current = current->left;

}

return current->data;

} int main() {

struct Node\* root = NULL; int n, data;

scanf("%d", &n);

*241901092*

*241901092*

for (int i = 0; i < n; i++) { scanf("%d", &data);

*241901092*

*241901092*

root = insert(root, data);

}

displayTreePostOrder(root); printf("\n");

int minValue = findMinValue(root);

printf("The minimum value in the BST is: %d", minValue);

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

return 0;

}

*241901092*

*241901092*

*241901092*

*241901092*