Rajalakshmi Engineering College

*241901092*

*241901092*

Name: yogeshwaran T

*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 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.

*241901092*

*241901092*

*241901092*

*241901092*

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.

*241901092*

*241901092*

*241901092*

*241901092*

*Output Format*

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

traversal.

*241901092*

*241901092*

*241901092*

*241901092*

Refer to the sample output for formatting specifications.

*Sample Test Case*

Input: 5

3 1 5 2 4

*241901092*

*241901092*

*241901092*

Output: 3 1 2 5 4

*Answer*

#include <stdio.h>

#include <stdlib.h>

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

*241901092*

struct Node\* right;

};

struct Node\* createNode(int value) {

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

*241901092*

*241901092*

*241901092*

*241901092*

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

}

// You are using GCC

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

}

if (value < root->data) { root->left = insert(root->left, value); root->right = insert(root->right, value); void printPreorder(struct Node\* node) { if (node == NULL) {

} else {

}

*241901092*

*241901092*

*241901092*

*241901092*

return root;

}

*241901092*

*241901092*

*241901092*

*241901092*

return;

}

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

printPreorder(node->right);

*241901092*

*241901092*

*241901092*

} int main() {

*241901092*

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;

}

*241901092*

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

*241901092*

*241901092*

*241901092*

*241901092*

*241901092*

*241901092*

*241901092*