Rajalakshmi Engineering College

*241901104*

*241901104*

Name: Shafraz.R

*241901104*

*241901104*

Email: 241901104@rajalakshmi.edu.in

Roll no: 241901104

Phone: 6380360049

Branch: REC

Department: l CSE (CS) FB

Batch: 2028

Degree: B.E - CSE (CS)

NeoColab\_REC\_CS23231\_DATA STRUCTURES

*241901104*

*241901104*

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

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In his computer science class, John is learning about Binary Search Trees (BST). He wants to build a BST and find the maximum value in the tree.

*241901104*

*241901104*

*241901104*

*241901104*

Help him by writing a program to insert nodes into a BST and find the maximum value in the tree.

*Input Format*

The first line of input consists of an integer N, representing the number of nodes

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

in the BST.

*241901104*

*241901104*

*241901104*

*241901104*

*Output Format*

The output prints the maximum value in the BST.

*241901104*

*241901104*

*241901104*

*241901104*

Refer to the sample output for formatting specifications.

*Sample Test Case*

Input: 5

10 5 15 2 7 Output: 15

*Answer*

#include <stdio.h>

#include <stdlib.h>

struct TreeNode {

int data; struct TreeNode\* left;

*241901104*

*241901104*

*241901104*

*241901104*

struct TreeNode\* right;

};

struct TreeNode\* createNode(int key) {

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

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

}

// You are using GCC

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

*241901104*

*241901104*

*241901104*

*241901104*

if (key < root->data)

root->left = insert(root->left, key); else

root->right = insert(root->right, key); int findMax(struct TreeNode\* root) { if (root == NULL) return -1;

*241901104*

*241901104*

*241901104*

*241901104*

return root;

}

*241901104*

*241901104*

*241901104*

*241901104*

struct TreeNode\* current = root; while (current->right != NULL) current = current->right;

return current->data;

} int main() { int N, rootValue;

*241901104*

*241901104*

scanf("%d", &N);

struct TreeNode\* root = NULL;

for (int i = 0; i < N; i++) { int key; scanf("%d", &key); if (i == 0) rootValue = key;

*241901104*

*241901104*

root = insert(root, key);

}

int maxVal = findMax(root); if (maxVal != -1) {

*241901104*

*241901104*

*241901104*

printf("%d", maxVal);

}

return 0; }

*241901104*

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

*241901104*

*241901104*

*241901104*

*241901104*