

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_CY_Updated

Attempt : 1

Total Mark : 30

Marks Obtained : 28.5

Section 1 : Coding

1. Problem Statement

Arun is working on a Binary Search Tree (BST) data structure. His goal is to implement a program that reads a series of integers and inserts them into a BST.

Once the integers are inserted, he needs to add a given integer value to each node in the tree and find the maximum value in the BST.

Your task is to help Arun implement this program.

Input Format

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

The second line consists of N space-separated integers, each representing an element to be inserted into the BST.

The third line consists of an integer add, representing the value to be added to each node in the BST.

Output Format

The output prints the maximum value in the BST after adding the add value.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

10 5 15 20 25 5

Output: 30

Answer

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

```
// Define the structure of
a BST node struct Node {
    int val;
    Node* left;
    Node* right;

    Node(int value) {
        val = value;
        left = right = nullptr;
    }
};
```

```
// Function to insert a value into the BST Node*
insert(Node* root, int key) {
```

```
    if (root == nullptr)    return new Node(key);
    if (key < root->val)
```

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

```
// Function to add a given value to each node in
the BST
```

```
void addToEachNode(Node* root, int add) {
    if (root == nullptr)    return;    root->val +=
add;    addToEachNode(root->left, add);
    addToEachNode(root->right, add);
}
```

```
// Function to find the maximum value in the BST
```

```
int findMax(Node* root) {    Node* current = root;
while (current->right != nullptr) {
    current = current->right;
}
return current->val; }
```

```
// Main function
```

```
int main() {
    int n;    cin >> n;
```

```
    Node* root = nullptr;    for
    (int i = 0; i < n; ++i) {
        int val;
        cin >> val;
        root = insert(root,
val);
    }
```

```
    int add;
    cin >> add;
```

```
addToEachNode(root, add);
```

```
} return 0;
```

```
cout << findMax(root) << endl;
```

: **Correct** Marks : 10/10

Status