

# LAB-09

Name: K V Jaya Harsha

Roll no: CS23B1034

Date: 25-10-2024

Q1. Binary tree (in cpp)

```
//CS23B1034
//K V Jaya Harsha
#include <iostream>
using namespace std;
struct Node{
    int data;
    Node *left;
    Node *right;

    Node(int value){
        data = value;
        left = NULL;
        right = NULL;
    }
};
int main(){
    struct Node *root = new Node(1);
    root->left = new Node(2);
    root->right = new Node(3);
    root->left->left = new Node(4);
    root->left->right = new Node(5);
    root->right->left = new Node(6);
    root->right->right = new Node(7);
    root->right->right->right = new Node(8);
    root->right->right->left = new Node(9);
    root->left->left->left = new Node(10);
    root->left->left->right = new Node(11);
    root->left->left->right = new Node(12);
    return 0;
}
```

## Q2. Traversal (in cpp)

```
// CS23B1034
// K V Jaya Harsha
#include <iostream>
using namespace std;
struct Node{
    int data;
    Node *left;
    Node *right;
    Node(int value){
        data = value;
        left = NULL;
        right = NULL;
    }
    void inorder(Node *root){
        if (root == NULL){
            return;
        }
        inorder(root->left);
        cout << root->data << " ";
        inorder(root->right);
    }

    void preorder(Node *root){
        if (root == NULL){
            return;
        }
        cout << root->data << " ";
        inorder(root->left);
        inorder(root->right);
    }
    void postorder(Node *root){
        if (root == NULL){
            return;
        }
        inorder(root->left);
        inorder(root->right);
        cout << root->data << " ";
    }
};
```

```

int main(){
    struct Node *root = new Node(1);
    root->left = new Node(2);
    root->right = new Node(3);
    root->left->left = new Node(4);
    root->left->right = new Node(5);
    root->right->left = new Node(6);
    root->right->right = new Node(7);
    root->right->right->right = new Node(8);
    root->right->right->left = new Node(9);
    root->left->left->left = new Node(10);
    root->left->left->right = new Node(11);
    root->left->left->right->right = new Node(12);
    cout << "inorder: ";
    root->inorder(root);
    cout << '\n';
    cout << "preorder: ";
    root->preorder(root);
    cout << '\n';
    cout << "postorder: ";
    root->postorder(root);
    return 0;
}

```

```

($?) { .\binary-tree-in-pre-post-order }
inorder: 10 4 11 12 2 5 1 6 3 9 7 8
preorder: 1 10 4 11 12 2 5 6 3 9 7 8
postorder: 10 4 11 12 2 5 6 3 9 7 8 1
PS C:\Users\harsh\OneDrive\Documents\Deskt

```

Q3. Binary Search. (in cpp)

```
// CS23B1034
//K V Jaya Harsha
#include <iostream>
#include <stack>
using namespace std;
struct Node
{
    int data;
    Node *left;
    Node *right;
    Node(int value){
        data = value;
        left = NULL;
        right = NULL;
    }
    void tosearch(Node *root, int target){
        if (root == NULL){
            cout << "Tree is empty." << endl;
            return;
        }
        stack<Node *> s;
        s.push(root);
        while (!s.empty()){
            Node *current = s.top();
            s.pop();
            if (current->data == target){
                cout << "True.\nElement found." << endl;
                return;
            }
            if (current->right != NULL){
                s.push(current->right);
            }
            if (current->left != NULL){
                s.push(current->left);
            }
        }
        cout << "False.\nElement not found." << endl;
    }
};
```

```

int main(){
    struct Node *root = new Node(1);
    root->left = new Node(2);
    root->right = new Node(3);
    root->left->left = new Node(4);
    root->left->right = new Node(5);
    root->right->left = new Node(6);
    root->right->right = new Node(7);
    root->right->right->right = new Node(8);
    root->right->right->left = new Node(9);
    root->left->left->left = new Node(10);
    root->left->left->right = new Node(11);
    root->left->left->right = new Node(12);
    int n;
    cout << "Please enter the target: ";
    cin >> n;
    root->tosearch(root, n);
    return 0;
}

```

```

?) { .\binary-tree-binary-search }
Please enter the target: 5
True.
Element found.
PS C:\Users\harsh\OneDrive\Documents\Desktop\challenge\c
a\labs\lab-9>

```

Q4. No of Node in tree (in cpp)

```
// CS23B1034
// K V Jaya Harsha
#include <iostream>
using namespace std;
struct Node
{
    int data;
    Node *left;
    Node *right;
    Node(int value)
    {
        data = value;
        left = NULL;
        right = NULL;
    }
};

int countNodes(Node *root)
{
    if (root == NULL)
    {
        return 0;
    }
    int leftCount = 0;
    if (left != NULL)
    {
        leftCount = left->countNodes(root);
    }
    int rightCount = 0;
    if (right != NULL)
    {
        rightCount = right->countNodes(root);
    }
    return 1 + leftCount + rightCount;
}

int main()
{
    Node *root = new Node(1);
    root->left = new Node(2);
    root->right = new Node(3);
    root->left->left = new Node(4);
    root->left->right = new Node(5);
    root->right->left = new Node(6);
    root->right->right = new Node(7);
    root->right->right->right = new Node(8);
    root->right->right->left = new Node(9);
    root->left->left->left = new Node(10);
    root->left->left->right = new Node(11);
    root->left->left->right->right = new Node(12);

    int totalNodes = root->countNodes(root);
    cout << "Total number of nodes in the tree: " << totalNodes << endl;

    return 0;
}
```

inary-tree-search-node }

Total number of nodes in the tree: 12

PS C:\Users\harsh\OneDrive\Documents\Des