Bahria University,

Karachi Campus

LAB EXPERIMENT NO. _12_

LIST OF TASKS

TASK NO	OBJECTIVE
Task 1	Create a data structure to store a binary search. Implement a method to insert and search an element in BST.
Task 2	Create a data structure to store a binary search. Implement a method to insert and traverse in BST (inorder, postorder, preorder).
Task 3	N/A
Task 4	N/A
Task 5	**N/A**
Task 6	**N/A**
Task 7	N/A
Task 8	N/A

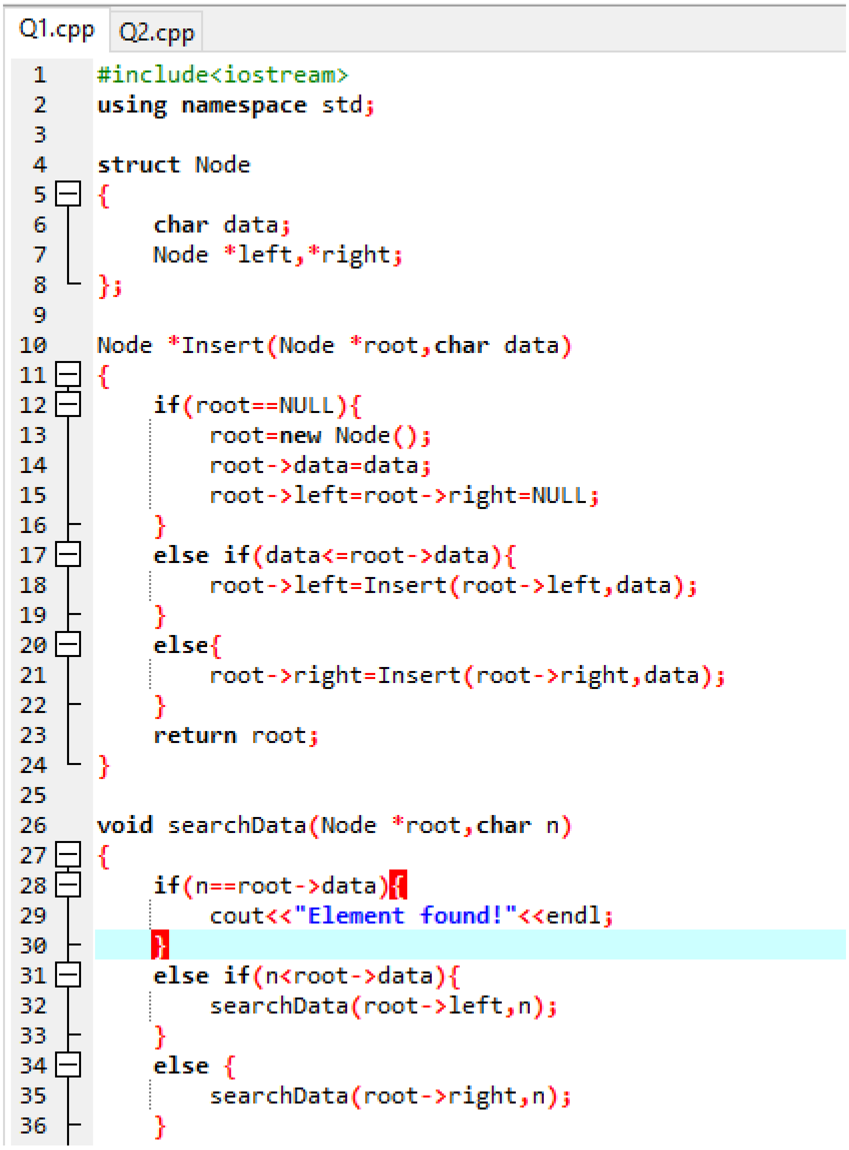
Submitted On:

__17/06/2020__

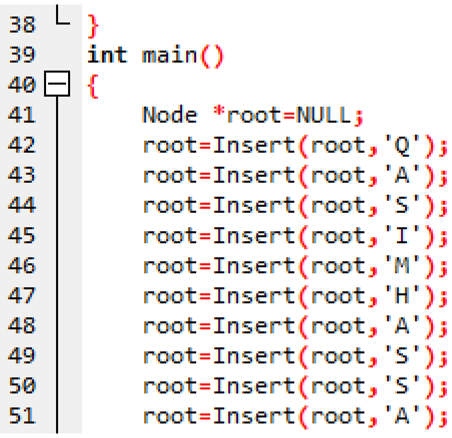
(Date: DD/MM/YY)

Task No. 1: Create a data structure to store a binary search. Implement a method to insert and search an element in BST.

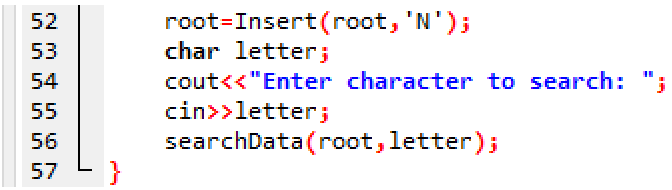
Coding:



```cpp
#include<iostream>
using namespace std;

struct Node
{
    char data;
    Node *left,*right;
};

Node *Insert(Node *root,char data)
{
    if(root==NULL){
        root=new Node();
        root->data=data;
        root->left=root->right=NULL;
    }
    else if(data<=root->data){
        root->left=Insert(root->left,data);
    }
    else{
        root->right=Insert(root->right,data);
    }
    return root;
}

void searchData(Node *root,char n)
{
    if(n==root->data){
        cout<<"Element found!"<<endl;
    }
    else if(n<root->data){
        searchData(root->left,n);
    }
    else {
        searchData(root->right,n);
    }
```

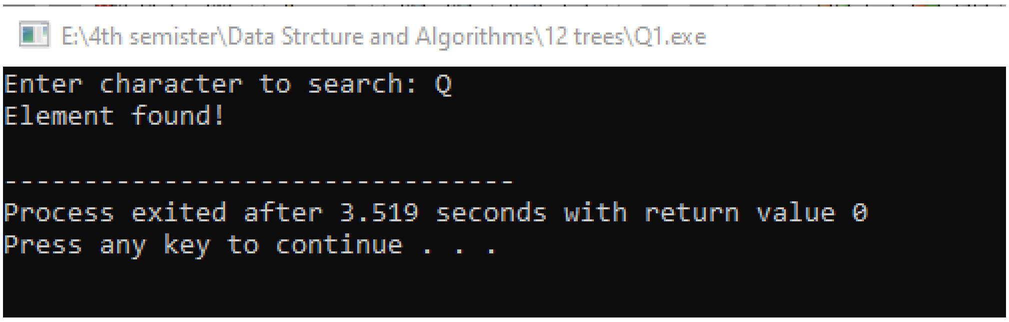


```cpp
}
int main()
{
    Node *root=NULL;
    root=Insert(root,'Q');
    root=Insert(root,'A');
    root=Insert(root,'S');
    root=Insert(root,'I');
    root=Insert(root,'M');
    root=Insert(root,'H');
    root=Insert(root,'A');
    root=Insert(root,'S');
    root=Insert(root,'S');
    root=Insert(root,'A');
```



```
    root=Insert(root,'N');
    char letter;
    cout<<"Enter character to search: ";
    cin>>letter;
    searchData(root,letter);
}
```

Output:

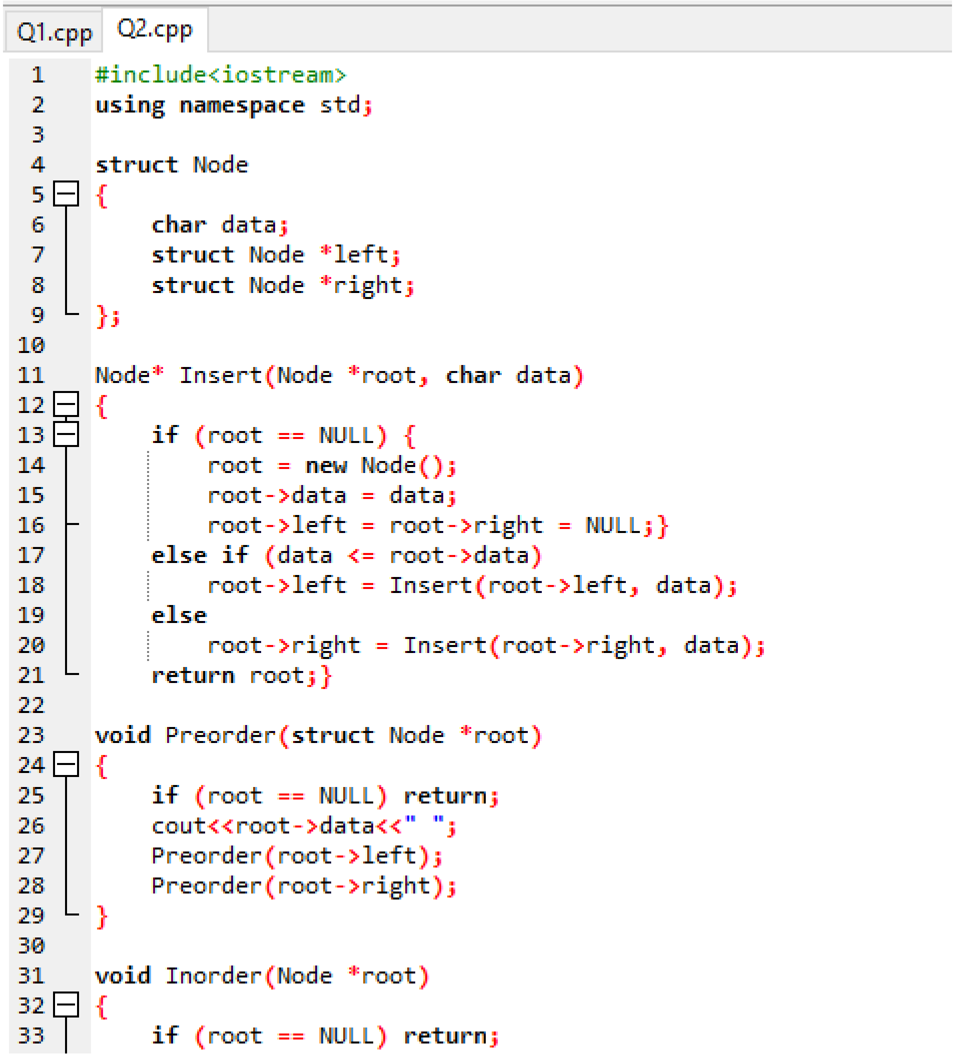


E:\4th semister\Data Strcture and Algorithms\12 trees\Q1.exe

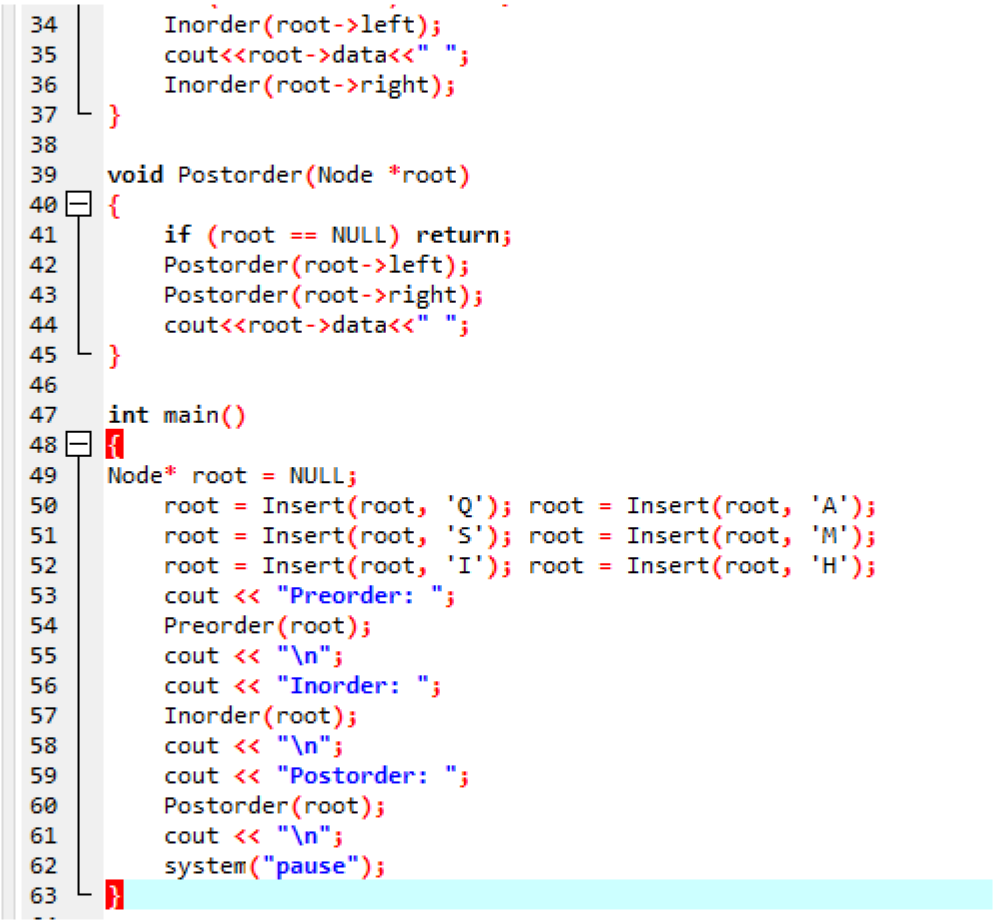
```
Enter character to search: Q
Element found!

--------------------------------
Process exited after 3.519 seconds with return value 0
Press any key to continue . . .
```

Task No. 2: Create a data structure to store a binary search. Implement a method to insert and traverse in BST (inorder, postorder, preorder).

Coding:

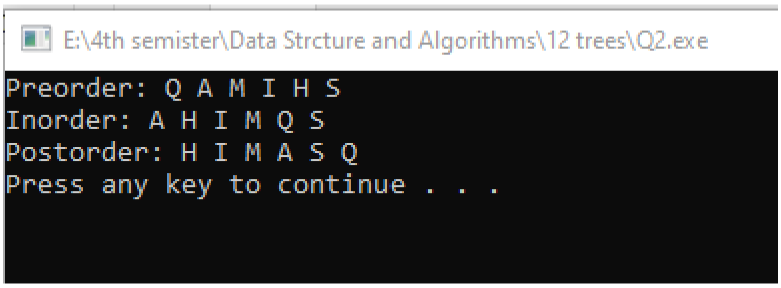


```
#include<iostream>
using namespace std;

struct Node
{
    char data;
    struct Node *left;
    struct Node *right;
};

Node* Insert(Node *root, char data)
{
    if (root == NULL) {
        root = new Node();
        root->data = data;
        root->left = root->right = NULL;}
    else if (data <= root->data)
        root->left = Insert(root->left, data);
    else
        root->right = Insert(root->right, data);
    return root;}

void Preorder(struct Node *root)
{
    if (root == NULL) return;
    cout<<root->data<<" ";
    Preorder(root->left);
    Preorder(root->right);
}

void Inorder(Node *root)
{
    if (root == NULL) return;
```



```
    Inorder(root->left);
    cout<<root->data<<" ";
    Inorder(root->right);
}

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

int main()
{
Node* root = NULL;
    root = Insert(root, 'Q'); root = Insert(root, 'A');
    root = Insert(root, 'S'); root = Insert(root, 'M');
    root = Insert(root, 'I'); root = Insert(root, 'H');
    cout << "Preorder: ";
    Preorder(root);
    cout << "\n";
    cout << "Inorder: ";
    Inorder(root);
    cout << "\n";
    cout << "Postorder: ";
    Postorder(root);
    cout << "\n";
    system("pause");
}
```

Output:



E:\4th semister\Data Strcture and Algorithms\12 trees\Q2.exe

```
Preorder: Q A M I H S
Inorder: A H I M Q S
Postorder: H I M A S Q
Press any key to continue . . .
```