Fatima Jinnah Women University

Opening Portals of Excellence Through Higher Education

LAB #12

SUBMITTED BY: Tanzeela Asghar

ROLL NO: 2021-

BSE- 032 SECTION: 3(A)

SUBMITTED TO: SIR REHAN AHMED SIDDIQUI

COURSE: DATA STRUCTURE ANDALGORITHM(LAB)

Task 1 :

Give answers to the following.

1.Traverse the binary tree given above in pre, post and inorder.

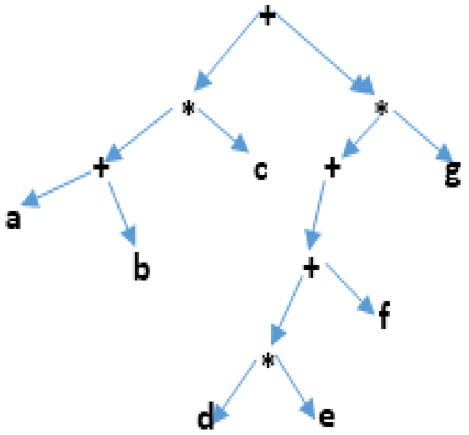
a. Preorder Traversal:**14 2 1 3 11 10 7 30 40**

b. Post Traversal: **1 3 2 7 10 40 30 11 14**

c. In-order Traversal: **1 2 3 14 7 10 11 40 30**

2. Draw the expression tree of the given algebraic expression and traverse the tree in pre, post and inorder.

(a+b*c)+((d*e+f)*g)



Code Task

Complete the given class to implement a binary search tree.

```
    class bst
    {
    struct node
    {
    node *left;
    node *right;
    int data;
};
    node* root;
    public:
    bst();
    bool isempty();
    void insert(int item);
    bool search(int item);
      void Preorder(node * ptr)
     void Postorder(node * ptr)
     void Inorder(node * ptr)

    };
```

ANSWER:

```
#include"stdafx.h"
#include<iostream>
using namespace std;
class bst
{
    public:
    struct node
    {
        int data;
        node *left;
        node *right;
    };node *root;
    bst(){root=NULL;}
    bool isempty()
    {
        if(root==NULL)
            return true;
        else
            return false;
    }
    void insert(int item)
    {
        node * ptr = root;
        node * prev = NULL;
        while(ptr!=NULL)
        {
            prev = ptr;
            if(item < ptr->data)
                ptr = ptr->left;
            else if(item > ptr->data)
                ptr = ptr->right;
            else
            {
                cout<<"Value already exist";
                return;
            }}
        node * temp =new node;
        temp->data=item;
        temp->left=NULL;
        temp->right=NULL;
        if(prev==NULL)
        root = temp;
        else if(item < prev->data)
            prev->left = temp;
        else
            prev->right = temp;
    }
    bool search(int item)
    {
```

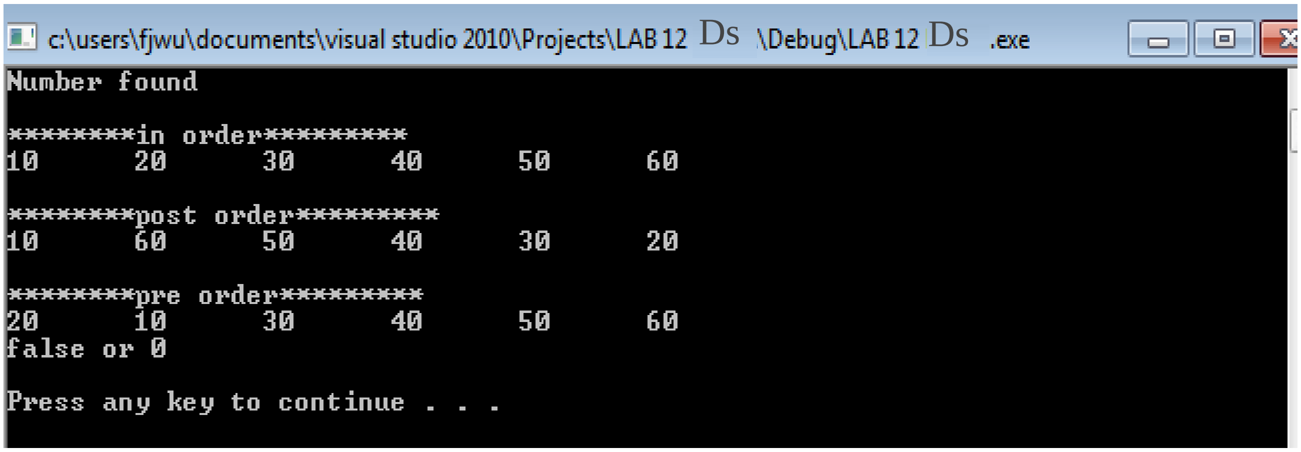
```
node * ptr = root;
            bool found =false;
            for(;;)
            {
                    if(found || ptr == NULL)
                            break;
                    if(item < ptr->data)
                            ptr = ptr->left;
                    else if(item> ptr->data)
                            ptr = ptr->right;

else
                    found =true;
            }
            return found;
    }
    void Search(node* temp,int item)
    {
            if(temp==NULL)
                    cout<<"Number not found";
            else if(temp->data ==item)
                    cout<<"Number found";
            else if(temp->data > item)
                    Search(temp->left, item);
            else if(temp->data < item)
                    Search(temp->right, item);
            }
                    void Preorder(node * ptr)
                    {
                    if(ptr!=NULL)
                    {
                    cout << ptr->data <<"\t";
                    Preorder(ptr->left);
                    Preorder(ptr->right);
                    }}
                    void Postorder(node * ptr)
                    {
                    if(ptr!=NULL)
                    {
                    Postorder(ptr->left);
                    Postorder(ptr->right);
                    cout << ptr->data <<"\t";
                    }
                    }
                    void Inorder(node *ptr)
                    {
                    if(ptr!=NULL)
                    {
                    Inorder(ptr->left);
                    cout << ptr->data <<"\t";
Inorder(ptr->right);
                    }}
    };
int main()
    {
            bst t;
            t.insert(20);
            t.insert(30);
            t.insert(10);
            t.insert(40);
            t.insert(50);
            t.insert(60);
            t.Search(t.root,30);
            cout<<"\n \n********in order********* "<<endl;
            t.Inorder(t.root);
            cout<<"\n \n********post order********* "<<endl;
            t.Postorder(t.root);
            cout<<"\n \n********pre order********* "<<endl;
            t.Preorder(t.root);
            cout<<endl;
            if(t.search(5)==true)

{


            cout<<"true or 1\n"<<endl;
            }
            else
            {

            cout<<"false or 0\n"<<endl;
    }
            system("pause");
    return 0;
}
```


c:\users\fjwu\documents\visual studio 2010\Projects\LAB 12 Ds \Debug\LAB 12 Ds .exe

```
Number found

*********in order**********
10      20      30      40      50      60

********post order**********
10      60      50      40      30      20

********pre order**********
20      10      30      40      50      60
false or 0

Press any key to continue . . .
```