

Name	Komal Tarachandani
UID no.	2021600065
Experiment No.	5

AIM:	Implement binary tree operations and its application
-------------	--

Program5

PROBLEM STATEMENT :	<p>Perform the following operations on a binary tree:</p> <p>1-Creation of binary tree and display using any one traversal</p> <p>2- counting no. of ndes in a binary tree</p> <p>3- counting no of leaf nodes in a binary tree</p> <p>4- counting height of a given node in a binary tree</p> <p>5- create an Arithmetic expression tree from given postfix expression</p> <p>show the intermediate stages of output for each function</p>
PROGRAM:	<p><u>BASIC TREE OPERATIONS:</u></p> <pre>#include <iostream> using namespace std; class Node { public: int data; Node *left; Node *right; }; int count=1,cl=0; int find_height(Node *root,int height) { if(root==NULL) return -1; int leftheight=find_height(root->left,height);</pre>

```

int rightheight=find_height(root->right,height);
int ans=max(leftheight,rightheight)+1;
return ans;
}
void display(Node *cur)
{
    if(cur==NULL)
        return;
    display(cur->left);
    cout<<cur->data<<"-->";
    if(cur->right==NULL && cur->left==NULL)
        cl++;
    display(cur->right);
}
void insert_node_left(Node *cur,Node *newnode)
{
    if(cur->left==NULL)
    {
        count+=1;
        cur->left=newnode;
        cout<<"Added to the left ";
    }
    else
    {
        cur=cur->left;
        insert_node_left(cur,newnode);
    }
}
void insert_node_right(Node *cur,Node *newnode)
{
    if(cur->right==NULL)
    {
        count+=1;
        cur->right=newnode;
        cout<<"Added to the right ";
    }
    else
    {
        cur=cur->right;
        insert_node_right(cur,newnode);
    }
}

```

```

    }
}
int main()
{
    Node *root,*newnode,*root1,*newnode1;
    root=NULL;
    cout<<"CREATE THE TREE\n";
    int ch=1,num;
    cout<<"Enter data for root";
    newnode=new Node();
    cin>>num;
    newnode->left=NULL;
    newnode->right=NULL;
    newnode->data=num;
    root1=newnode;
    root=newnode;
    while(ch==1)
    {
        cout<<"Enter data for left node: ";
        cin>>num;
        newnode1=new Node();
        newnode1->left=NULL;
        newnode1->right=NULL;
        newnode1->data=num;
        insert_node_left(root,newnode1);
        cout<<"\nEnter data for right node";
        cin>>num;
        newnode=new Node();
        newnode->left=NULL;
        newnode->right=NULL;
        newnode->data=num;
        insert_node_right(root,newnode);
        cout<<"\nEnter 1 to continue";
        cin>>ch;
        root=newnode1;
    }
    cout<<"Exited loop";
    cout<<"\nNUMBER OF NODES ARE "<<count;
    cout<<"\nThe TREE INORDER TRAVERSAL IS:\n";
    display(root1);
}

```

```
cout<<"\nThe number of leaf nodes is "<<cl;
cout<<"\nthe height of root node is "<<find_height(root1,0);
```

```
return 0;
```

```
}
```

ARITHMETIC TREE OPERATION:

```
#include <iostream>
```

```
using namespace std;
```

```
class Node
```

```
{
```

```
public:
```

```
char data;
```

```
Node *right,*left;
```

```
};
```

```
int n=0,top=-1;
```

```
Node *root,*arr[10];
```

```
void push(Node *cur)
```

```
{
```

```
    top++;
```

```
    arr[top]=cur;
```

```
}
```

```
Node* pop()
```

```
{
```

```
    Node *cur;
```

```
    cur=arr[top];
```

```
    top--;
```

```
    return cur;
```

```
}
```

```
void display(Node *cur)
```

```
{
```

```
    if(cur==NULL)
```

```
        return ;
```

```
    display(cur->left);
```

```
    cout<<cur->data<<" ";
```

```
    display(cur->right);
```

```
}
```

```
int main()
```

```
{
```

```
    root=NULL;
```

```
Node *newnode,*cur,*leftnode,*rightnode;
int ch=1;
char ele;
while(ch==1)
{
    cout<<"Enter next element: ";
    cin>>ele;
    newnode=new Node();
    newnode->left=NULL;
    newnode->right=NULL;
    newnode->data=ele;
    if(ele=="*"||ele=="+"||ele=="-"||ele=="/")
    {
        rightnode=pop();
        leftnode=pop();
        newnode->left=leftnode;
        newnode->right=rightnode;
        push(newnode);
    }
    else
        push(newnode);
    cout<<"\nEnter 1 to continue : ";
    cin>>ch;
}
root=arr[top];
display(root);
return 0;
}
```

```

input
CREATE THE TREE
Enter data for root5
Enter data for left node: 2
Added to the left
Enter data for right node3
Added to the right
Enter 1 to continue1
Enter data for left node: 4
Added to the left
Enter data for right node5
Added to the right
Enter 1 to continue0
Exited loop
NUMBER OF NODES ARE 5
The TREE INORDER TRAVERSAL IS:
4-->2-->5-->5-->3-->
The number of leaf nodes is 3
the height of root node is 2

...Program finished with exit code 0
Press ENTER to exit console.

```

RESULT:

```

Enter element: 4
Enter 1 to continue : 1
Enter element: 3
Enter 1 to continue : 1
Enter element: *
Enter 1 to continue : 1
Enter element: 5
Enter 1 to continue : 1
Enter element: +
Enter 1 to continue : 1
Enter element: 6
Enter 1 to continue : 1
Enter element: /
Enter 1 to continue : 0
4 * 3 + 5 / 6

...Program finished with exit code 0
Press ENTER to exit console.

```

CONCLUSION:

Hence I was able to learn the application of BINARY TREE and its basic operations.