**PROGRAM FOR THREADED BINARY TREE:**

#include <iostream>

#include<cstdlib>

using namespace std;

class tbt

{

private:

struct node

{

int data,lbit,rbit;

node \*left;

node\*right;

}\*dummy,\*root;

public:

tbt()

{

dummy=NULL;

root=NULL;

}

void create();

void preorder();

void inorder(node \*trav, node \*dummy);

void postorder(node \*trav,node\*dummy);

void display();

void insert(node \*trav,node\*temp);

};

void tbt::create()

{

int num;

node \*trav,\*temp;

temp=new node();

cin>>num;

temp->data=num;

temp->lbit=0;

temp->rbit=0;

if (root==NULL)

{

dummy=new node;

dummy->data=-0000;

dummy->lbit=1;

dummy->rbit=1;

dummy->left=temp;

dummy->right=dummy;

temp->left=dummy;

temp->right=dummy;

root=temp;

}

else

{

trav=root;

insert(trav,temp);

}

}

void tbt:: insert(node \*trav,node\*temp)

{

if(temp->data < trav->data)

{

if (trav->lbit==0)

{

temp->left=trav->left;

temp->right=trav;

trav->left=temp;

trav->lbit=1;

}

else

{

insert(trav->left,temp);

}

}

if(temp->data > trav->data)

{

if(trav->rbit==0)

{

temp->right=trav->right;

temp->left=trav;

trav->right=temp;

trav->rbit=1;

}

else

{

insert(trav->right,temp);

}

}

}

void tbt::preorder()

{

node \*trav;

trav=root;

if (root==NULL)

{

cout<<"Tree is empty";

}

else

{

while(trav!=dummy)

{

cout<<endl<<trav->data;

if (trav->lbit==1)

{

trav=trav->left;

}

else

{

while(trav->rbit==0 && trav->right !=dummy)

{

trav=trav->right;

}

trav=trav->right;

}

}

}

}

void tbt::inorder(node \*trav, node \*dummy)

{

trav=root;

if (root==NULL)

{

cout<<"Tree is empty";

}

else

{

while(trav !=dummy)

{

while (trav->lbit==1)

{

trav=trav->left;

}

cout<<endl<<trav->data;

while(trav!=dummy)

{

if(trav->rbit==1)

{

trav=trav->right;

while(trav->lbit==1)

{

trav=trav->left;

}

cout<<endl<<trav->data;

}

else

{

while(trav->rbit==0)

{

trav=trav->right;

if(trav==dummy)

{

break;

}

cout<<endl<<trav->data;

}

}

}

}

}

}

void tbt::postorder(node \*trav,node\*dummy)

{

trav=root;

if (root==NULL)

{

cout<<"Tree is empty";

}

else

{

int a[20],i=0,n;

while(trav!=dummy)

{

a[i]=trav->data;

i++;

if(trav->rbit==1)

{

trav=trav->right;

}

else

{

while(trav->lbit==0 && trav->left!=dummy)

{

trav=trav->left;

}

trav=trav->left;

}

}

n=i-1;

for(i=n;i>=0;i--)

{

cout<<endl<<a[i];

}

}

}

void tbt::display()

{

node \*trav;

trav=root;

int ch,i,j;

do

{

cout<<"\n##MENU##\n";

cout<<"\n1.Insert\n2.Preorder\n3.Inorder\n4.Postorder\n5.Exit\n";

cout<<"\nEnter your choice:\n";

cin>>ch;

switch(ch)

{

case 1:cout<<"Enter the no of elements to be inserted:";

cin>>j;

cout<<"\nEnter the elements to be inserted\n";

for(i=0;i<j;i++)

{

create();

}

break;

case 2:

cout<<"\nThe preorder traversal of the entered data is";

preorder();

break;

case 3:

cout<<"\nThe inorder traversal of the entered data is";

inorder(trav,dummy);

break;

case 4:

cout<<"\nThe postorder traversal of the entered data is";

postorder(trav,dummy);

break;

case 5:

exit(0);

default:

cout<<"\nInvalid choice please enter a valid choice"<<endl;

break;

}

}

while(i!=5);

}

int main()

{

tbt t;

t.display();

}

**OUTPUT**:

##MENU##

1.Insert

2.Preorder

3.Inorder

4.Postorder

5.Exit

Enter your choice:

1

Enter the no of elements to be inserted:10

Enter the elements to be inserted

50

40

70

45

60

80

42

48

75

85

##MENU##

1.Insert

2.Preorder

3.Inorder

4.Postorder

5.Exit

Enter your choice:

22

Invalid choice please enter a valid choice

##MENU##

1.Insert

2.Preorder

3.Inorder

4.Postorder

5.Exit

Enter your choice:

2

The preorder traversal of the entered data is

50

40

45

42

48

70

60

80

75

85

##MENU##

1.Insert

2.Preorder

3.Inorder

4.Postorder

5.Exit

Enter your choice:

3

The inorder traversal of the entered data is

40

42

45

48

50

60

70

75

80

85

##MENU##

1.Insert

2.Preorder

3.Inorder

4.Postorder

5.Exit

Enter your choice:

4

The postorder traversal of the entered data is

42

48

45

40

60

75

85

80

70

50

##MENU##

1.Insert

2.Preorder

3.Inorder

4.Postorder

5.Exit

Enter your choice:

5