#include<iostream>

using namespace std;

class node

{

public:

int data;

int rth,lth;

node \*left,\*right;

};

class thread

{

public:

node \*head;

node \*newnode, \*root, \*temp, \*parent;

thread()

{

root=NULL;

}

void create();

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

void display\_inorder(node\*,node\*);

void display\_preorder(node\*,node\*);

};

void thread::create()

{

newnode=new node;

newnode->left=NULL;

newnode->right=NULL;

newnode->lth=0;

newnode->rth=0;

cout<<"Enter the element:";

cin>>newnode->data;

if(root==NULL)

{

root=newnode;

head=new node;

head->left=root;

root->left=head;

root->right=head;

}

else

insert(root,newnode);

}

void thread::insert(node \*ptr,node \*newnode)

{

if(newnode->data<ptr->data)

{

if(ptr->lth==0)

{

newnode->left=ptr->left;

newnode->right=ptr;

ptr->lth=1;

ptr->left=newnode;

}

else

insert(ptr->left,newnode);

}

if(newnode->data>ptr->data)

{

if(ptr->rth==0)

{

newnode->right=ptr->right;

newnode->left=ptr;

ptr->rth=1;

ptr->right=newnode;

}

else

insert(ptr->right,newnode);

}

}

void thread::display\_inorder(node \*temp,node \*d)

{

while(temp!=d)

{

while(temp->lth==1)

{

temp=temp->left;

}

cout<<temp->data;

while(temp->rth==0)

{

temp=temp->right;

if(temp==d)

return;

cout<<" "<<temp->data;

}

temp=temp->right;

}

}

void thread::display\_preorder(node \*temp, node \*head)

{

int flag=0;

while(temp!=head)

{

if(flag==0)

cout<<" "<<temp->data;

if((temp->lth==1)&&(flag==0))

{

temp=temp->left;

}

else

{

while(1)

{

if(temp->rth==1)

{

flag=0;

temp=temp->right;

break;

}

else

{

if(temp!=head)

{

flag=1;

temp=temp->right;

break;

}

}

}

}

}

}

int main()

{

int choice;

thread th;

do

{

cout<<"\n Program for threaded binary tree";

cout<<"\n1. Create \n2. Display Inorder \n3. Display Preorder \n4. Exit\n";

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

cin>>choice;

switch(choice)

{

case 1:

th.create();

break;

case 2:

th.display\_inorder(th.root,th.head);

break;

case 3:

th.display\_preorder(th.root,th.head);

break;

}

}while(choice!=4);

return 0;

}

OUTPUT:

Program for threaded binary tree

1. Create

2. Display Inorder

3. Display Preorder

4. Exit

Enter your choice: 1

Enter the element:24

Program for threaded binary tree

1. Create

2. Display Inorder

3. Display Preorder

4. Exit

Enter your choice: 1

Enter the element:34

Program for threaded binary tree

1. Create

2. Display Inorder

3. Display Preorder

4. Exit

Enter your choice: 1

Enter the element:44

Program for threaded binary tree

1. Create

2. Display Inorder

3. Display Preorder

4. Exit

Enter your choice: 1

Enter the element:54

Program for threaded binary tree

1. Create

2. Display Inorder

3. Display Preorder

4. Exit

Enter your choice: 1

Enter the element:64

Program for threaded binary tree

1. Create

2. Display Inorder

3. Display Preorder

4. Exit

Enter your choice: 2

24 34 44 54 64

Program for threaded binary tree

1. Create

2. Display Inorder

3. Display Preorder

4. Exit

Enter your choice: 3

24 34 44 54 64

Program for threaded binary tree

1. Create

2. Display Inorder

3. Display Preorder

4. Exit

Enter your choice: 4