#include<iostream>

#include<stdlib.h>

using namespace std;

class node

{

public:

node \*left,\*right;

int lbit,rbit;

int data;

node(int d)

{

data = d;

}

friend class list;

};

class list

{

public:

node \*root;

list()

{

root=new node(-1);

root->lbit=0;

root->rbit=1;

root->left=root->right=root;

}

node \*presuc(node \*t);

node \*insuc(node \*t);

void inorder();

void preorder();

void create1(node \*,int );

void create()

{

create1(root,0);

}

};

void list::create1(node \*fatherof,int leftorright)//leftorright=0 for left child

{

node \*p;

int x;

cout<<"\n\nEnter a data (-1 for no data): ";

cin >> x;

if(x==-1)

return ;

p=new node(x);

if(leftorright==0)

{

p->lbit=fatherof->lbit;

p->left=fatherof->left;

fatherof->left=p;

fatherof->lbit=1;

p->rbit=0;

p->right=fatherof;

}

else

{

p->rbit=fatherof->rbit;

p->right=fatherof->right;

fatherof->right=p;

fatherof->rbit=1;

p->lbit=0;

p->left=fatherof;

}

cout<<"\n\n\tEnter left child of"<<x;

create1(p,0);

cout<<"\n\n\tEnter right child of"<<x;

create1(p,1);

}

node \* list::presuc(node \*t)

{

if(t->lbit==1)

return(t->left);

if(t->rbit==1)

return(t->right);

while(t->rbit==0)

t=t->right;

return(t->right);

}

void list::preorder()

{

node \*t;

t=root->left;

cout <<"\n\n";

while(t!=root)

{

cout<<" "<<t->data;

t=presuc(t);

}

}

node \* list::insuc(node \*t)

{

if(t->rbit==1)

{

t=t->right;

while(t->lbit==1)

t=t->left;

return(t);

}

else

return(t->right);

}

void list::inorder()

{

node \*t;

t=root->left;

cout<<"\n\n";

while(t->lbit==1)

t=t->left;

while(t!=root)

{

cout<<" "<<t->data;

t=insuc(t);

}

}

int main()

{

list tbt;

int choice;

cout<<"\n\n\t\t\t\t\t\t\t\t\t\t\t\t THREADED BINARY TREE";

do

{

cout<<"\n\n MENU: ";

cout<<"\n\n\t1.Create\n\n\t2.Preorder Traversal\n\n\t3.Inorder Traversal\n\n\t4.Postorder Traversal\n\n\t5.Exit";

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

cin>>choice;

switch(choice)

{

case 1: tbt.create();break;

case 2: tbt.preorder();break;

case 3: tbt.inorder();break;

//case 4: tbt.postorder();break;

}

}while(choice!=5);

}