#include <iostream>

using namespace std;

class tbt

{

int data;

int lbit, rbit;

tbt\* left, \* right;

friend class thread;

};

class thread

{

private:

tbt\* dummynode;

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

public:

thread()

{

root = dummynode = newnode = temp = parent = NULL;

}

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

void inorder(tbt\*, tbt\*);

void preorder(tbt\*, tbt\*);

void create();

void display();

};

void thread::create() //Create

{

tbt\* newnode = NULL;

//void insert(tbt \*, tbt \*);

newnode = new tbt;

newnode->left = NULL;

newnode->right = NULL;

newnode->lbit = 1;

newnode->rbit = 1;

cout << "\n Enter data : ";

cin >> newnode->data;

if (root == NULL)

{

root = newnode;

dummynode = new tbt;

dummynode->data = -999;

dummynode->left = root;

root->left = dummynode;

root->right = dummynode;

root->lbit = 1;

root->rbit = 1;

}

else

{

insert(root, newnode);

}

}

void thread::display()

{

int n;

cout << "1) For Inorder" << endl;

cout << "2) For Preorder" << endl;

cin >> n;

if (root == NULL)

{

cout << "Tree is empty";

}

else

{

cout << "\n The Tree is: ";

if (n == 1)

{

inorder(root, dummynode);

}

else

{

preorder(root, dummynode);

}

}

}

void thread::insert(tbt\* root, tbt\* newnode) //Function creates BST and then add node.

{

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

{

if (root->lbit == 1)

{

newnode->left = root->left;

newnode->right = root;

root->left = newnode;

root->lbit = 0;

}

else

{

insert(root->left, newnode);

}

}

if (newnode->data > root->data)

{

if (root->rbit == 1)

{

newnode->right = root->right;

newnode->left = root;

root->rbit = 0;

root->right = newnode;

}

else

insert(root->right, newnode);

}

}

void thread::inorder(tbt\* temp, tbt\* dummynode)

{

while (temp != dummynode)

{

while (temp->lbit == 0)

{

temp = temp->left;

}

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

while (temp->rbit == 1)

{

temp = temp->right;

if (temp == dummynode)

{

return;

}

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

}

temp = temp->right;

}

}

void thread::preorder(tbt\* temp, tbt\* dummynode)

{

while (temp != dummynode)

{

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

while (temp->lbit == 0)

{

temp = temp->left;

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

}

while (temp->rbit == 1)

{

temp = temp->right;

}

temp = temp->right;

}

}

int main()

{

int choice;

char ch = 'y';

thread th;

do

{

cout << "\n\t Threaded Binary Tree";

cout << "\n 1.Create \n 2.Display \n";

cin >> choice;

switch (choice)

{

case 1:do

{

th.create();

cout << "\n Enter Data:(y/n)";

cin >> ch;

} while (ch == 'y');

break;

case 2:

th.display();

break;

}

cout << "\n\n Do you want to continue?(y/n)";

cin >> ch;

} while (ch == 'y');

}