Lab Experiment No.6

-----------------------------------------------------------------------------------------------------------------------------

Name: Prathamesh Sadashiv Gadekar

Roll No.:14

Class : SE-IT

-----------------------------------------------------------------------------------------------------------------------------

Implement In-order Threaded Binary Tree and traverse it in In-order and Pre-order.

-----------------------------------------------------------------------------------------------------------------------------

#include <iostream>

#include <cstdlib>

#define MAX\_VALUE 65536

using namespace std;

/\* Class Node \*/

class Node

{

public:

int key;

Node \*left, \*right;

bool leftThread, rightThread;

};

/\* Class ThreadedBinarySearchTree \*/

class ThreadedBinarySearchTree

{

private:

Node \*root;

public:

/\* Constructor \*/

ThreadedBinarySearchTree()

{

root = new Node();

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

root->leftThread = true;

root->key = MAX\_VALUE;

}

/\* Function to clear tree \*/

void makeEmpty()

{

root = new Node();

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

root->leftThread = true;

root->key = MAX\_VALUE;

}

/\* Function to insert a key \*/

void insert(int key)

{

Node \*p = root;

for (;;)

{

if (p->key < key)

{

if (p->rightThread)

break;

p = p->right;

}

else if (p->key > key)

{

if (p->leftThread)

break;

p = p->left;

}

else

{

/\* redundant key \*/

return;

}

}

Node \*tmp = new Node();

tmp->key = key;

tmp->rightThread = tmp->leftThread = true;

if (p->key < key)

{

/\* insert to right side \*/

tmp->right = p->right;

tmp->left = p;

p->right = tmp;

p->rightThread = false;

}

else

{

tmp->right = p;

tmp->left = p->left;

p->left = tmp;

p->leftThread = false;

}

}

/\* Function to print tree \*/

void printInorder()

{

Node \*tmp = root, \*p;

for (;;)

{

p = tmp;

tmp = tmp->right;

if (!p->rightThread)

{

while (!tmp->leftThread)

{

tmp = tmp->left;

}

}

if (tmp == root)

break;

cout<<tmp->key<<" ";

}

cout<<endl;

}

};

int main()

{

struct Node \*root,N;

ThreadedBinarySearchTree tbt;

int k,n;

root=NULL;

cout<<"\n How many nodes you want to insert in Threaded Binary Tree: ";

cin>>n;

for(int i=0;i<n;i++)

{

cout<<"\n Enter Key value: ";

cin>>k;

tbt.insert(k);

}

cout<<"\n Inorder Traversal of Threaded Binary Tree : ";

tbt.printInorder();

return 0;

}

**OUTPUT :**

How many nodes you want to insert in Threaded Binary Tree: 5

Enter Key value: 1

Enter Key value: 9

Enter Key value: 7

Enter Key value: 40

Enter Key value: 28

Inorder Traversal of Threaded Binary Tree : 1 7 9 28 40