**//Threaded Binary Tree**

#include<iostream>

#include<stdlib.h>

//#define MAX 10

using namespace std;

struct node

{

int lbit;

node \*left;

int data;

node \*right;

int rbit;

};

class tree

{

public :

node \*root;

node \*head;

void get\_create();

void get\_insert(node \*,node \*);

void inorder(node \*);

void preorder(node \*);

};

void tree::get\_create()

{

node \*temp;

char ch;

head=new node;

head->data=999;

head->right=head;

head->lbit=1;

head->rbit=0;

root=NULL;

do

{

temp=new node;

cout<<"\n\tEnter the value of node :";

cin>>temp->data;

temp->left=NULL;

temp->right=NULL;

temp->rbit=0;

temp->lbit=0;

if(root==NULL)

{

root=temp;

head->left=root;

root->right=head;

root->left=head;

root->rbit=0;

root->lbit=0;

}

else

{

get\_insert(root,temp);

}

cout<<"\n\tDo you want Continue ? (Press Y/N)";

cin>>ch;

}

while(ch=='Y'||ch=='y');

}

void tree::get\_insert(node \*root,node \*temp)

{

char op;

if(temp->data<=root->data)

{

if(root->lbit==0)

{

temp->left=root->left;

temp->right=root;

root->left=temp;

root->lbit=1;

}

else

get\_insert(root->left,temp);

}

else

{

if(root->rbit==0)

{

temp->right=root->right;

temp->left=root;

root->right=temp;

root->rbit=1;

}

else

get\_insert(root->right,temp);

}

} void tree::inorder(node \*head)

{

if(root==NULL)

return;

node \*t=head->left;

do

{

while(t->lbit!=0)

{

t=t->left;

}

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

while(t->rbit==0)

{

t=t->right;

if(t==head)

{

return;

}

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

}

t=t->right;

}while(t!=head);

}

//-----------------------------------------------------------------------------------------

void tree::preorder(node \*head)

{

if(root==NULL)

return;

node \*t=head->left;

do

{

while(t->lbit!=0)

{

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

t=t->left;

}

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

while(t->rbit==0)

{

t=t->right;

if(t==head)

return;

}

t=t->right;

}while(t!=head);

}

int main()

{

int op;

tree t;

// clrscr();

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

cout<<"\n\n\tCreation of Binary Tree\n ";

t.get\_create();

do

{

cout<<"\n\n\t1>Inorder Traversing.\n\t2>Preorder Traversing.\n\t3>Creating Tree Again.\n\t4>Exit.";

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

cin>>op;

switch(op)

{

case 1:

cout<<"\n\t Inordered traversal :";

t.inorder(t.head);

break;

case 2:

cout<<"\n\t Preordered traversal :";

t.preorder(t.head);

break;

case 3:

cout<<"\n\tCreation of Binary Tree\n ";

t.get\_create();

break;

case 4:

exit(0);

break;

}

}while(op!=4);

}