**Practical Assignment :: 06**

**Github:**

**Implement inorder threaded binary tree and traverse it in inorder and preorder**

#include<bits/stdc++.h>

using namespace std;

struct Node

{

struct Node \*left, \*right;

int data;

bool thread\_left;

bool thread\_right;

};

Node \*insert(struct Node \*root, int data\_to\_insert)

{

Node \*temp = root;

Node \*parent = NULL;

while (temp != NULL)

{

if (data\_to\_insert == (temp->data))

{

cout<<"Duplicate Key !\n";

return root;

}

parent = temp;

if (data\_to\_insert < temp->data)

{

if (temp -> thread\_left == false)

temp = temp -> left;

else

break;

}

else

{

if (temp->thread\_right == false)

temp = temp -> right;

else

break;

}

}

Node \*new\_node = new Node;

new\_node -> data = data\_to\_insert;

new\_node -> thread\_left = true;

new\_node -> thread\_right = true;

if (parent == NULL)

{

root = new\_node;

new\_node -> left = NULL;

new\_node -> right = NULL;

}

else if (data\_to\_insert < (parent -> data))

{

new\_node -> left = parent -> left;

new\_node -> right = parent;

parent -> thread\_left = false;

parent -> left = new\_node;

}

else

{

new\_node -> left = parent;

new\_node -> right = parent -> right;

parent -> thread\_right = false;

parent -> right = new\_node;

}

return root;

}

void preorder(struct Node \*root )

{

struct Node \*ptr;

if(root==NULL)

{

cout << "Tree is empty" << endl;

return;

}

ptr=root;

while(ptr!=NULL)

{

cout<<ptr->data<<" ";

if(ptr->thread\_left==false)

ptr=ptr->left;

else if(ptr->thread\_right==false)

ptr=ptr->right;

else

{

while(ptr!=NULL && ptr->thread\_right==true)

ptr=ptr->right;

if(ptr!=NULL)

ptr=ptr->right;

}

}

}

struct Node \*inorderSuccessor(struct Node \*temp)

{

if (temp -> thread\_right == true)

return temp->right;

temp = temp -> right;

while (temp -> thread\_left == false)

temp = temp -> left;

return temp;

}

void inorder(struct Node \*root)

{

if (root == NULL)

cout<<"Tree is empty";

Node \*temp = root;

while (temp -> thread\_left == false)

temp = temp -> left;

while (temp != NULL)

{

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

temp = inorderSuccessor(temp);

}

}

int main()

{

Node \*root = NULL; int ch,n,size;

do{

cout<<"\n1. Insert";

cout<<"\n2. Display Inorder Threaded binary tree";

cout<<"\n3. Display Preorder Threaded binary tree ";

cout<<"\n4. Exit\nEnter Choice :";

cin>>ch;

switch(ch){

case 1:cout<<"How many nodes you want to enter :";

cin>>size;

for(int i=1;i<=size;i++){

cout<<"#"<<i<<" Insert node value : ";

cin>>n;

root=insert(root,n);

}

break;

case 2:

cout<<"Inorder binary threaded tree: ";

inorder(root); ;

cout<<endl;

break;

case 3:

cout<<"Preorder binary threaded tree: ";

preorder(root); ;

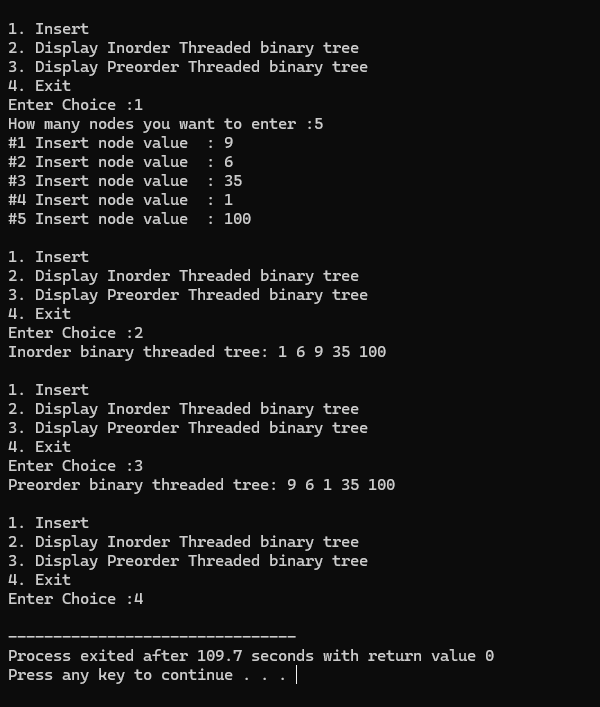
cout<<endl;

break;

}}while(ch!=4);

return 0;

}

****