

Assignment No-6

Name- PRAFUL SHEWALE

Roll no=SI-48

Class=SE-IT

Aim- 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; //Mark temp node as root as initially tree is
empty
head->left=root;//dummy node
root->right=head; //check if 1st node
root->left=head;
root->rbit=0;
root->lbit=0;
}
else //If root is already created
{
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; //root left is pointing to head so
now temp left will point to head

```

```

temp->right=root; // temp right to root
root->left=temp; // root left to temp
root->lbit=1;
}
else
get_insert(root->left,temp); //for next insertions root
left i.e. temp will become root
}
else // (temp->data>root->data)
{
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; //start traversing from root ,take temp
pointer t
do
{
while(t->lbit!=0)
{
t=t->left;
}
cout<<" "<<t->data;
while(t->rbit==0)
{
t=t->right; //follow the thread back now check whether
it is pointing to head
if(t==head) // check if right points to head stop
{
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\tt-----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\tInordered traversal :";
t.inorder(t.head);
break;
case 2:
cout<<"\n\tPreordered 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);
}

```

OUTPUT

-----THREADED BINARY TREE OPERATIONS-----

Creation of Binary Tree

Enter the value of node: 100

Do you want to continue? (Press Y/N): y

Enter the value of node: 500

Do you want to continue? (Press Y/N): y

Enter the value of node: 200

Do you want to continue? (Press Y/N): y

Enter the value of node: 300

Do you want to continue? (Press Y/N): n

- 1> Inorder Traversing.
- 2> Preorder Traversing.
- 3> Creating Tree Again.
- 4> Exit.

Enter your choice: 1

Inorder traversal: 100 200 300 500

- 1> Inorder Traversing.
- 2> Preorder Traversing.
- 3> Creating Tree Again.
- 4> Exit.

Enter your choice: 2

Preorder traversal: 100 500 200 300

- 1> Inorder Traversing.
- 2> Preorder Traversing.
- 3> Creating Tree Again.
- 4> Exit.

Enter your choice: 3

Creation of Binary Tree

Enter the value of node: 10

Do you want to continue? (Press Y/N): n

1> Inorder Traversing.
2> Preorder Traversing.
3> Creating Tree Again.
4> Exit.

Enter your choice: 4

=== Code Execution Successful ===