**Program: Threaded Binary Tree**

#include<stdio.h>

#include<conio.h>

#include <malloc.h>

struct tree

{

int val;

struct tree \*left;

struct tree \*right;

int thread;

};

typedef struct tree tree;

tree \*root=NULL;

tree \* insert\_node(tree \*root,tree \*ptr,tree \*rt)

{

if (root==NULL)

{

root=ptr;

if(rt != NULL)

{

root->right = rt;

root->thread = 1;

}

}

else if (ptr->val < root->val)

root->left = insert\_node(root->left,ptr,root);

else

if(root->thread==1)

{

root->right=insert\_node(NULL,ptr,rt);

root->thread=0;

}

else

root->right=insert\_node(root->right,ptr,rt);

return root;

}

tree \* createThreadedTree()

{

tree \*ptr;

int num;

printf("\n Enter elements ,press -1 to terminate");

scanf("%d",&num);

while(num!=-1)

{

ptr=(tree\*)malloc(sizeof(tree));

ptr->val=num;

ptr->left=ptr->right=NULL;

ptr->thread=0;

root=insert\_node(root,ptr,NULL);

printf("Enter next ");

scanf("%d",&num);

}

return root;

}

void inorder(tree \*root)

{

tree \*ptr=root,\*prev;

do

{

while(ptr !=NULL)

{

prev=ptr;

ptr=ptr->left;

}

if(prev !=NULL)

{

printf("%d \t ",prev->val);

ptr=prev->right;

while (prev !=NULL && prev ->thread)

{

printf("%d \t",ptr->val);

prev=ptr;

ptr=ptr->right;

}

}

} while (ptr !=NULL);

}

int main()

{

createThreadedTree();

printf("Inorder Traversal is ");\

inorder(root);

return 0;

}

**Output:**

Enter elements ,press -1 to terminate 5

Enter next 8

Enter next 2

Enter next 3

Enter next 7

Enter next -1

Inorder Traversal is 2 3 5 7 8