**TREES**

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*l;

struct node \*r;

};

struct node \*t=NULL;

struct node \*t1;

struct node \*insert(int val, struct node \*t)

{

struct node \*newnode;

newnode=(struct node\*)malloc(sizeof(struct node));

if(t==NULL)

{

newnode->data=val;

newnode->l=newnode->r=NULL;

t=newnode;

}

else if(val < t->data)

t->l=insert(val, t->l);

else

t->r=insert(val, t->r);

return t;

}

struct node \*min(struct node \*n)

{

struct node \*ptr;

ptr=n;

while (ptr->l!=NULL)

ptr=ptr->l;

return ptr;

}

struct node \*delete1(int val, struct node \*t)

{

struct node \*temp;

if (t==NULL) return t;

if (val < t->data)

t->l = delete1 (val, t->l);

else if (val > t->data)

t->r = delete1 (val, t->r);

else

{

if(t->l==NULL)

{

temp=t->r;

free(t);

return temp;

}

if(t->r==NULL)

{

temp=t->l;

free(t);

return temp;

}

temp= min(t->r);

t->data=temp->data;

t->r=delete1(temp->data, t->r);

}

return t;

}

void preorder(struct node \*t)

{

if(t!=NULL)

{

printf("%d ", t->data);

preorder(t->l);

preorder(t->r);

}

}

void inorder(struct node \*t)

{

if(t!=NULL)

{

inorder(t->l);

printf("%d ", t->data);

inorder(t->r);

}

}

void postorder(struct node \*t)

{

if(t!=NULL)

{

postorder(t->l);

postorder(t->r);

printf("%d ", t->data);

}

}

int height (struct node \*t)

{

int h1=0, h2=0;

if(t==NULL)

return 0;

else

{

h1=height(t->l);

h2=height(t->r);

}

if(h1>=h2)

return (1+h1);

else

return (1+h2);

}

int nodes(struct node \*t)

{

if(t==NULL)

return 0;

else

return(nodes(t->l) + nodes(t->r) + 1);

}

struct node \*mirror(struct node \*t1)

{

struct node \*temp;

if (t1!=NULL)

{

temp=t1->l;

t1->l=t1->r;

t1->r=temp;

mirror(t1->l);

mirror(t1->r);

}

return t1;

}

int main()

{

int n,value,ch,ch1,h,c;

char a;

do{

printf("\n\n Enter choice:");

printf("1)INSERT AN ELEMENT IN A TREE\n2)DELETE AN ELEMENT FROM A TREE\n3)TRAVERSAL \n4)HEIGHT \n5)MIRROR \n6)NO. OF NODES");

scanf("%d",&ch);

switch(ch)

{

case 1: printf("ENTER A VALUE TO BE INSERTED");

scanf("%d",&value);

t=insert(value,t);

break;

case 2: printf("ENTER THE VALUE TO BE DELETED");

scanf("%d",&value);

t=delete1(value,t);

break;

case 3: printf("\n\n Enter choice:");

printf("1)PREORDER\n2)INORDER\n3)POSTORDER");

scanf("%d",&ch1);

switch(ch1)

{

case 1: preorder(t);

break;

case 2: inorder(t);

break;

case 3: postorder(t);

break;

}

case 4: h=height(t);

printf("\n Height of the tree is %d \n ",h);

break;

case 5: t1=t;

printf("\n Preorder traversal of original tree:");

preorder(t1);

t1=mirror(t1);

printf("\n Preorder traversal of mirror image of original tree: ");

preorder(t1);

break;

case 6: c=nodes(t);

printf("\n Number of nodes in the tree is %d ",c);

break;

}

printf("DO YOU WANT TO CONTINUE?(Y/N)");

scanf("%s",&a);

}while(a=='y'||a=='Y');

}

**OUTPUT**

Enter choice:1)INSERT AN ELEMENT IN A TREE

2)DELETE AN ELEMENT FROM A TREE

3)TRAVERSAL

4)HEIGHT

5)MIRROR

6)NO. OF NODES1

ENTER A VALUE TO BE INSERTED1

DO YOU WANT TO CONTINUE?(Y/N)y

Enter choice:1)INSERT AN ELEMENT IN A TREE

2)DELETE AN ELEMENT FROM A TREE

3)TRAVERSAL

4)HEIGHT

5)MIRROR

6)NO. OF NODES1

ENTER A VALUE TO BE INSERTED3

DO YOU WANT TO CONTINUE?(Y/N)y

Enter choice:1)INSERT AN ELEMENT IN A TREE

2)DELETE AN ELEMENT FROM A TREE

3)TRAVERSAL

4)HEIGHT

5)MIRROR

6)NO. OF NODES2

ENTER THE VALUE TO BE DELETED3

DO YOU WANT TO CONTINUE?(Y/N)y

Enter choice:1)INSERT AN ELEMENT IN A TREE

2)DELETE AN ELEMENT FROM A TREE

3)TRAVERSAL

4)HEIGHT

5)MIRROR

6)NO. OF NODES1

ENTER A VALUE TO BE INSERTED8

DO YOU WANT TO CONTINUE?(Y/N)y

Enter choice:1)INSERT AN ELEMENT IN A TREE

2)DELETE AN ELEMENT FROM A TREE

3)TRAVERSAL

4)HEIGHT

5)MIRROR

6)NO. OF NODES3

Enter choice:1)PREORDER

2)INORDER

3)POSTORDER1

1 8 DO YOU WANT TO CONTINUE?(Y/N)y

Enter choice:1)INSERT AN ELEMENT IN A TREE

2)DELETE AN ELEMENT FROM A TREE

3)TRAVERSAL

4)HEIGHT

5)MIRROR

6)NO. OF NODES4

Height of the tree is 2

DO YOU WANT TO CONTINUE?(Y/N)y

Enter choice:1)INSERT AN ELEMENT IN A TREE

2)DELETE AN ELEMENT FROM A TREE

3)TRAVERSAL

4)HEIGHT

5)MIRROR

6)NO. OF NODES6

Number of nodes in the tree is 2 DO YOU WANT TO CONTINUE?(Y/N)n