DS Assignment-10

Implement the following functions in context to a binary tree:

a) insertion of element

b) deletion of element

c) Updation of element

d) calculate height of the tree

Source Code:

```
#include<stdio.h>

#include<stdlib.h>

struct node

{

        int data;

        struct node* left;

        struct node* right;

};

struct node*root=NULL;

struct node* min(struct node* p)

{

        while(p->left!=NULL)

        {
```

```
		p=p->left;
	}
	return p;
}
int height(struct node* root)
{
	if(root==NULL)
	{
		return -1;
	}
	else
	{
		return max(height(root->left),height(root->right))+1;
	}
}
int max(int a,int b)
{
	if(a>=b)
	{
		return a;
	}
	else
	{
```

```
                return b;
        }
}
void insert(int value){
                                struct node* temp, *rest;
                                temp=(struct node*)malloc(sizeof(struct node));
                        temp->data=value;
                        temp->left=NULL;
                        temp->right=NULL;
                        rest=root;
                        struct node* trav;
                        trav=root;
                        while(trav!=NULL)
                        {
                                rest=trav;
                                if(temp->data>trav->data)
                                {
                                        trav=trav->right;
                                        }
                                        else
                                        {
                                                trav=trav->left;
                                        }
                        }
                        if(temp->data>rest->data)
                        {
                                rest->right=temp;
                        }
                        else
                        {
                           rest->left=temp;
                        }


}
struct node* deletion(struct node* root, int data)
{
                if(root==NULL)
                        {
                                printf("No element Found!");
                                return 0;
                                }
                                else if(data<root->data)
                                {
                                        deletion(root->left,data);
                                }
                                else if(data>root->data)
```

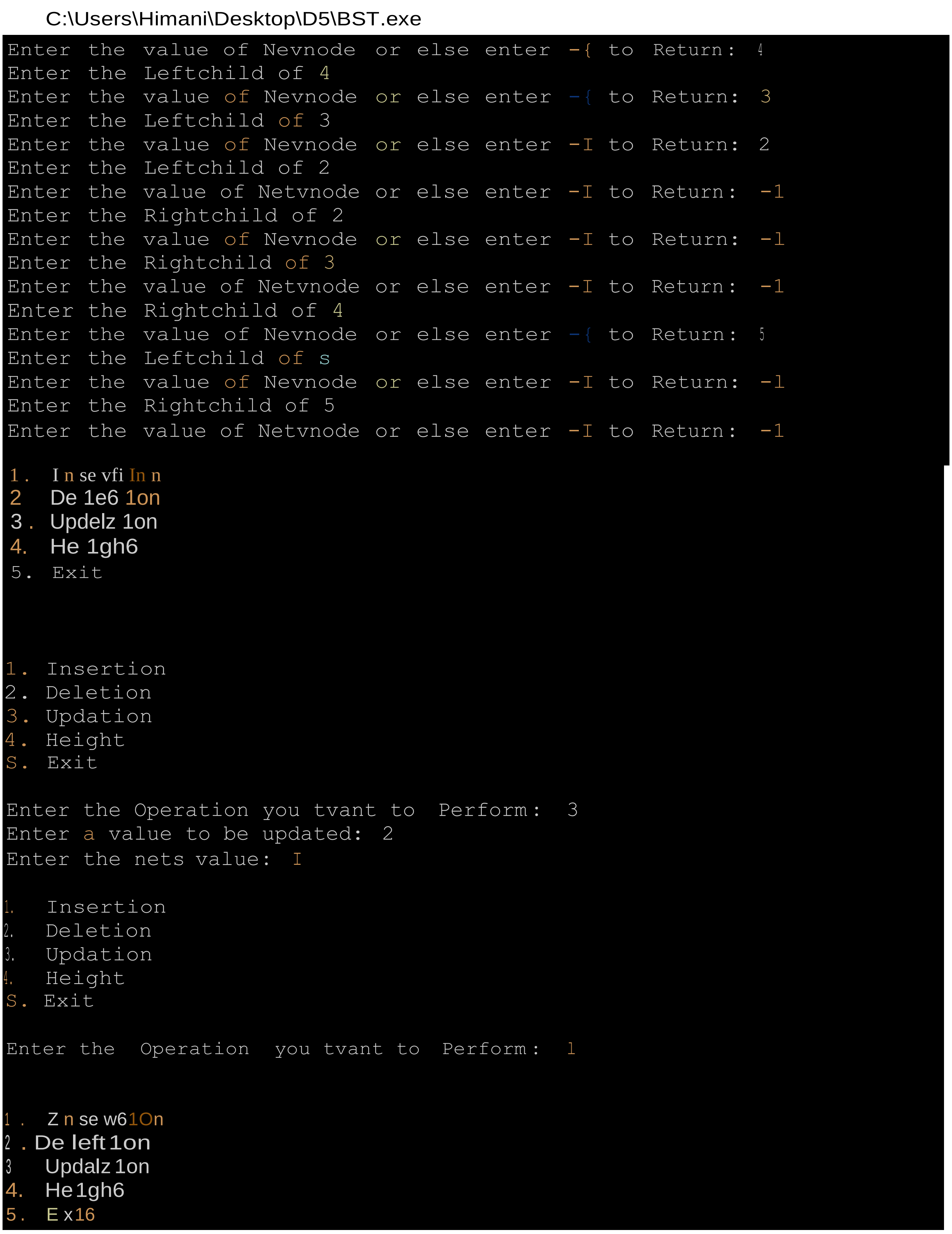
```
{
        deletion(root->right,data);
}
else if(root->right==NULL && root->left==NULL)
{
        free(root);
        root=NULL;
}
else if(root->right==NULL)
{
        struct node* tempp=root;
        root=root->left;
        free(tempp);
}
else if(root->left=NULL)
{
        struct node* tempp=root;
        root=root->right;
        free(tempp);
}
else
{
        struct node* tempp= min(root->right);
```

```
                                        root->data=tempp->data;
                                        root=deletion(root->right, tempp->data);
                                }
                                return root;
}

struct node* create(){
        int x;
        struct node* newnode;
        newnode=(struct node*)malloc(sizeof(struct node));
        printf("Enter the value of Newnode or else enter -1 to Return: ");
        scanf("%d", &x);
        if(x==-1)
        {
                return 0;
        }
        newnode->data=x;
        printf("Enter the Leftchild of %d\n", x);
        newnode->left=create();
        printf("Enter the Rightchild of %d\n", x);
        newnode->right=create();
        return newnode;
}
```

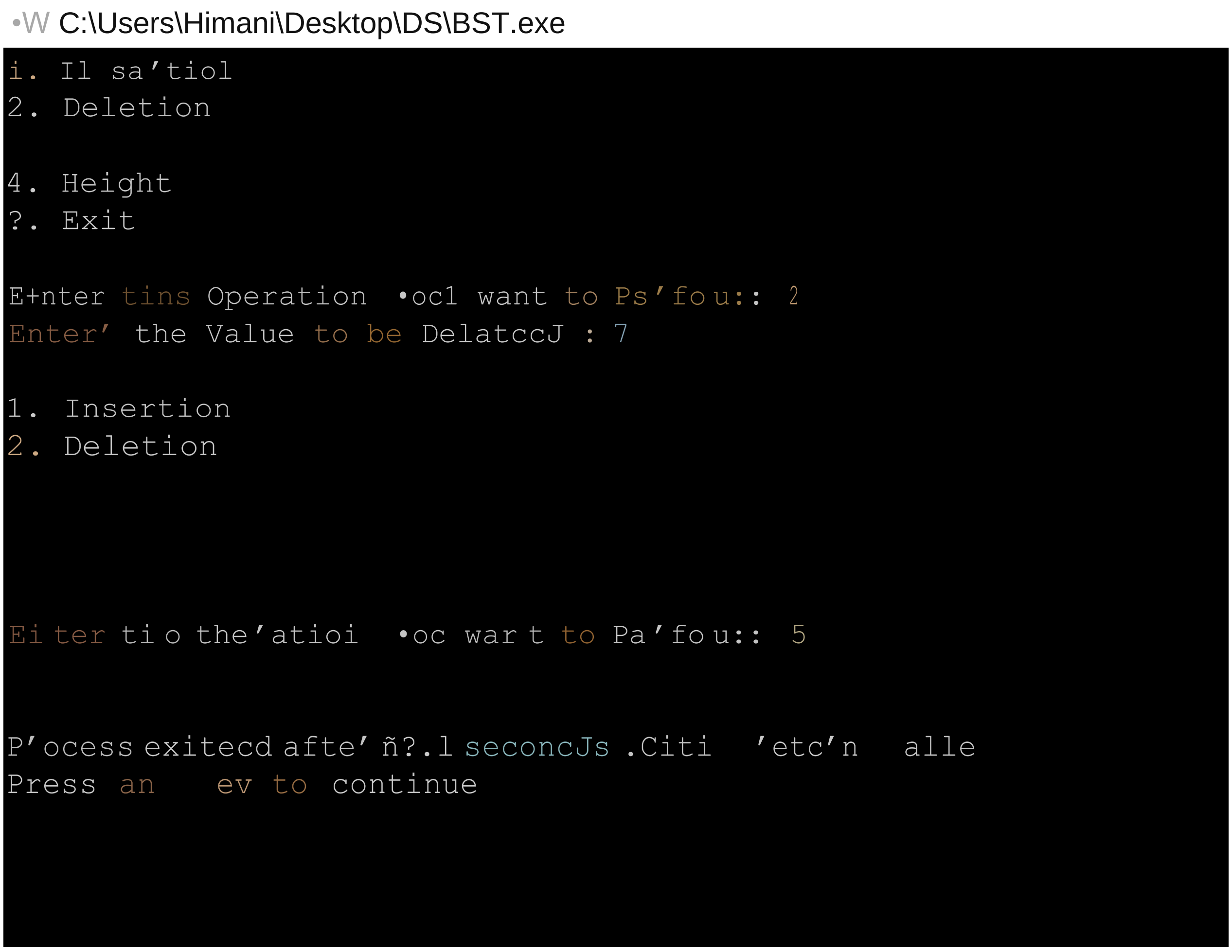
```
int main()
{
        int choice,value,data, update,naya;
   root=create();
   while(1)
   {
        printf("\n1. Insertion\n2. Deletion\n3. Updation\n4. Height\n5.
Exit\n");
        printf("\nEnter the Operation you want to Perform: ");
        scanf("%d", &choice);
        switch(choice)
        {
                case 1:
                        printf("Enter the data to be inserted: ");
                        scanf("%d", &value);
                        insert(value);
                        break;
                case 2:
                        printf("Enter the Value to be Deleted: ");
                        scanf("%d", &data);
                        deletion(root,data);
                        break;
                case 3:
```

```
                printf("Enter a value to be updated: ");
                scanf("%d", &update);
                deletion(root,update);
                printf("Enter the new value: ");
                scanf("%d", &naya);
                insert(naya);
                break;
            case 4:
                printf("The Overall height is %d", height(root));
                break;
            case 5:
                exit(0);
            default:
                printf("Enter a Valid Number!");
            }
        }
}
```

Output:



```
C:\Users\Himani\Desktop\D5\BST.exe
Enter the value of Nevnode or else enter -{ to Return: 4
Enter the Leftchild of 4
Enter the value of Nevnode or else enter -{ to Return: 3
Enter the Leftchild of 3
Enter the value of Nevnode or else enter -I to Return: 2
Enter the Leftchild of 2
Enter the value of Netvnode or else enter -I to Return: -1
Enter the Rightchild of 2
Enter the value of Nevnode or else enter -I to Return: -l
Enter the Rightchild of 3
Enter the value of Netvnode or else enter -I to Return: -1
Enter the Rightchild of 4
Enter the value of Nevnode or else enter -{ to Return: 5
Enter the Leftchild of s
Enter the value of Nevnode or else enter -I to Return: -l
Enter the Rightchild of 5
Enter the value of Netvnode or else enter -I to Return: -1

1.  In sevfi In n
2   De 1e6 1on
3.  Updelz 1on
4.  He 1gh6
5.  Exit



1. Insertion
2. Deletion
3. Updation
4. Height
S. Exit

Enter the Operation you tvant to  Perform:  3
Enter a value to be updated: 2
Enter the nets value:  I

1.  Insertion
2.  Deletion
3.  Updation
4.  Height
S. Exit

Enter the  Operation  you tvant to  Perform:  1


1 .  Z n se w61On
2 . De left1on
3    Updalz1on
4.  He1gh6
5.  Ex16
```



•W C:\Users\Himani\Desktop\DS\BST.exe

```
i. Il sa’tiol
2. Deletion

4. Height
?. Exit

E+nter tins Operation  •oc1 want to Ps’fo u:: 2
Enter’ the Value to be DelatccJ : 7

1. Insertion
2. Deletion




Ei ter ti o the’atioi  •oc war t to Pa’fo u:: 5


P’ocess exitecd afte’ ñ?.l seconcJs .Citi  ’etc’n  alle
Press an   ev to continue
```