Name: Abdulmuiz Khalid Shaikh
Roll no.:2101062
Subject: Data Structure and Algorithm Laboratory
Assignment No.5

```
#include<iostream>
#include<cstdlib>
usingnamespacestd;
classnode
{
public:
node*left;
stringword;
stringmeaning;
node*right;
};
classDict
{
public:
node*root;
Dict()
{
root=NULL;
}
voidcreate();
voidinsert(node*,node*);
voidinorder(node*);
voiddescend(node*);
voidsearch(node*,string);
voidmodify(node*,string);
node*minValue(node*);
node*deleteNode(node*,string);
};
voidDict::create()
{
node*temp;
temp=newnode;
cout<<"Enterword:";
cin>>temp->word;
getline(cin,temp->meaning);
cout<<"Entermeaning:";
getline(cin,temp->meaning);
temp->left=temp->right=NULL;
if(root==NULL)
root=temp;
else
insert(root,temp);
}
voidDict::insert(node*root,node*temp)
{
if(temp->word<root->word)
{
if(root->left==NULL)
root->left=temp;
else
insert(root->left,temp);
}
else
```

```
{
if(root->right==NULL)
root->right=temp;
else
insert(root->right,temp);
}
}
voidDict::inorder(node*temp)
{
if(temp!=NULL)
{
inorder(temp->left);
cout<<temp->word<<endl;
inorder(temp->right);
}
}
voidDict::descend(node*temp)
{
if(temp!=NULL)
{
descend(temp->right);
cout<<temp->word<<endl;
descend(temp->left);
}
}
voidDict::search(node*root,stringstr_key)
{
if(str_key.compare(root->word)<0)
{
if(root->left==NULL)
cout<<"Wordnotfound"<<endl;
else
search(root->left,str_key);
}
elseif(str_key.compare(root->word)>0)
{
if(root->right==NULL)
cout<<"Wordnotfound"<<endl;
else
search(root->right,str_key);
}
else
{
cout<<"Word:"<<root->word<<endl;
cout<<"Meaning:"<<root->meaning<<endl;
}
}
voidDict::modify(node*root,stringstr_key)
{
if(str_key.compare(root->word)<0)
{
if(root->left==NULL)
cout<<"Wordnotfound"<<endl;
else
modify(root->left,str_key);
}
elseif(str_key.compare(root->word)>0)
{
if(root->right==NULL)
cout<<"Wordnotfound"<<endl;
else
modify(root->right,str_key);
}
else
{
cout<<"Word:"<<root->word<<endl;
getline(cin,root->meaning);
cout<<"Enternewmeaning:";
getline(cin,root->meaning);
}
}
node*Dict::minValue(node*root)
{
node*curr;
curr=newnode;
curr=root;
while(curr->left!=NULL)
curr=curr->left;
returncurr;
}
node*Dict::deleteNode(node*root,stringstr_key)
{
if(root==NULL)returnroot;
node*temp;
temp=newnode;
if(str_key.compare(root->word)<0)
{
if(root->left==NULL)
cout<<"Wordnotfound"<<endl;
else
root->left=deleteNode(root->left,str_key);
}
elseif(str_key.compare(root->word)>0)
{
if(root->right==NULL)
cout<<"Wordnotfound"<<endl;
else
root->right=deleteNode(root->right,str_key);
}
else
{
if(root->left==NULL)
{
temp=root->right;
delete(root);
returntemp;
}
if(root->right==NULL)
{
temp=root->left;
delete(root);
returntemp;
}
temp=minValue(root->right);
root->word=temp->word;
root->meaning=temp->meaning;
root->right=deleteNode(root->right,temp->word);
}
returnroot;
}
```

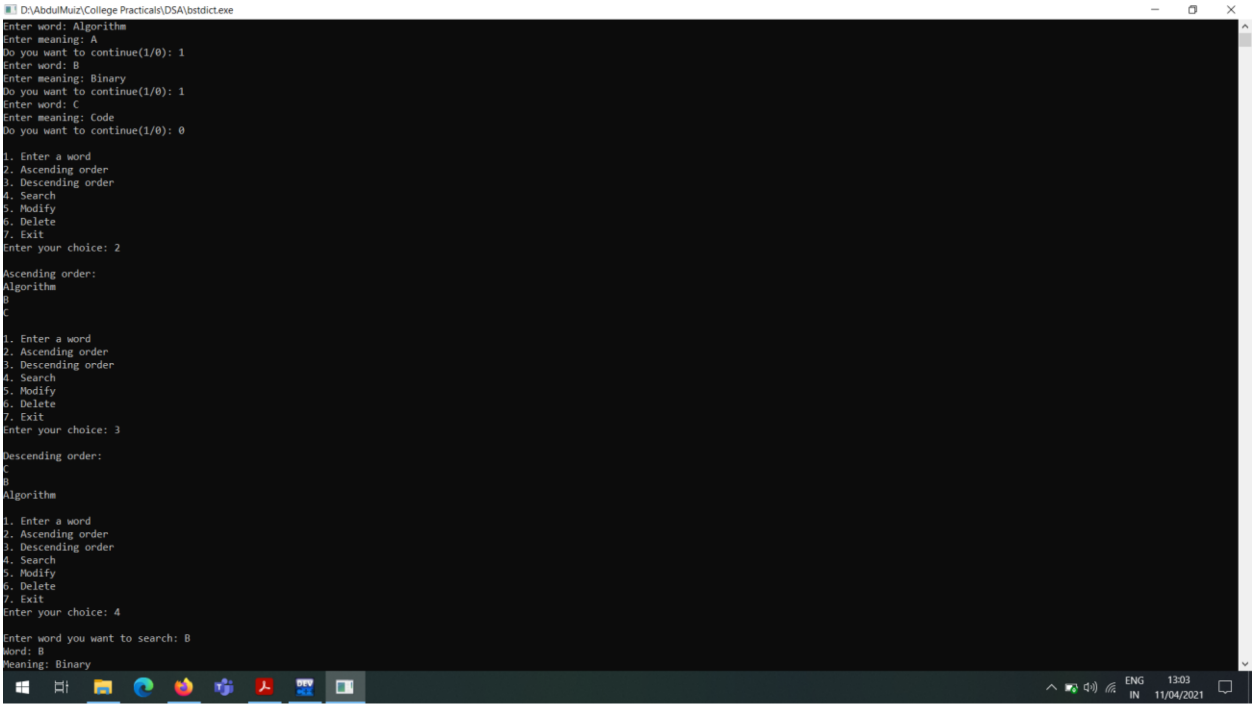
```
intmain(){
DictB;
stringkey;
intn,op;
x:B.create();
cout<<"Doyouwanttocontinue(1/0):";
cin>>n;
if(n==1)
gotox;
while(1)
{
cout<<"\n1.Enteraword";
cout<<"\n2.Ascendingorder";
cout<<"\n3.Descendingorder";
cout<<"\n4.Search";
cout<<"\n5.Modify";
cout<<"\n6.Delete";
cout<<"\n7.Exit";
cout<<"\nEnteryourchoice:";
cin>>op;
switch(op)
{
case1:B.create();break;
case2:cout<<"\nAscendingorder:"<<endl;B.inorder(B.root);break;
case3:cout<<"\nDescendingorder:"<<endl;B.descend(B.root);break;
case4:cout<<"\nEnterwordyouwanttosearch:";cin>>key;B.search(B.root,key);break;
case5:cout<<"\nEnterwordyouwanttomodify:";cin>>key;B.modify(B.root,key);break;
case6:cout<<"\nEnterwordyouwanttodelete:";cin>>key;B.root=B.deleteNode(B.root,key);
break;
case7:exit(1);break;
default:cout<<"\nInvalidchoice";break;
}
}
return0;
}
```

Output:

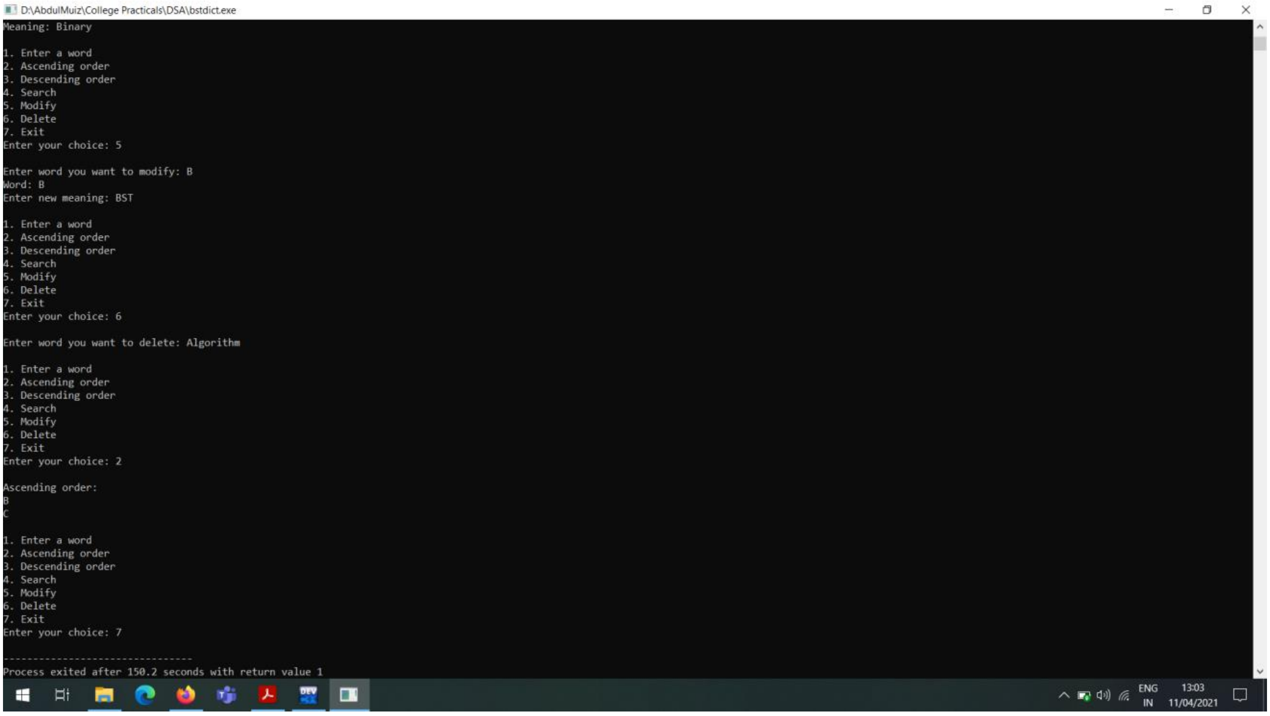
```
/*Enterword:Algorithm
Entermeaning:steps
Doyouwanttocontinue(1/0):1
Enterword:B
Entermeaning:binary
Doyouwanttocontinue(1/0):1
Enterword:C
Entermeaning:class
Doyouwanttocontinue(1/0):0
1.Enteraword
2.Ascendingorder
3.Descendingorder
4.Search
5.Modify
6.Delete
7.Exit
Enteryourchoice:2
Ascendingorder:
Algorithm
B
C
1.Enteraword
2.Ascendingorder
```

```
3.Descendingorder
4.Search
5.Modify
6.Delete
7.Exit
Enteryourchoice:3
Descendingorder:
C
B
Algorithm
1.Enteraword
2.Ascendingorder
3.Descendingorder
4.Search
5.Modify
6.Delete
7.Exit
Enteryourchoice:2
Ascendingorder:
Algorithm
B
C
1.Enteraword
2.Ascendingorder
3.Descendingorder
4.Search
5.Modify
6.Delete
7.Exit
Enteryourchoice:4
Enterwordyouwanttosearch:B
Word:B
Meaning:binary
1.Enteraword
2.Ascendingorder
3.Descendingorder
4.Search
5.Modify
6.Delete
7.Exit
Enteryourchoice:5
Enterwordyouwanttomodify:B
Word:B
Enternewmeaning:BST
1.Enteraword
2.Ascendingorder
3.Descendingorder
4.Search
5.Modify
6.Delete
7.Exit
Enteryourchoice:6
Enterwordyouwanttodelete:Algorithm
1.Enteraword
2.Ascendingorder
3.Descendingorder
4.Search
5.Modify
6.Delete
7.Exit
```

```
Enteryourchoice:2
Ascendingorder:
B
C
1.Enteraword
2.Ascendingorder
3.Descendingorder
4.Search
5.Modify
6.Delete
7.Exit
Enteryourchoice:7
*/
```



```
Enter word: Algorithm
Enter meaning: A
Do you want to continue(1/0): 1
Enter word: B
Enter meaning: Binary
Do you want to continue(1/0): 1
Enter word: C
Enter meaning: Code
Do you want to continue(1/0): 0

1. Enter a word
2. Ascending order
3. Descending order
4. Search
5. Modify
6. Delete
7. Exit
Enter your choice: 2

Ascending order:
Algorithm
B
C

1. Enter a word
2. Ascending order
3. Descending order
4. Search
5. Modify
6. Delete
7. Exit
Enter your choice: 3

Descending order:
C
B
Algorithm

1. Enter a word
2. Ascending order
3. Descending order
4. Search
5. Modify
6. Delete
7. Exit
Enter your choice: 4

Enter word you want to search: B
Word: B
Meaning: Binary
```



```
Meaning: Binary

1. Enter a word
2. Ascending order
3. Descending order
4. Search
5. Modify
6. Delete
7. Exit
Enter your choice: 5

Enter word you want to modify: B
Word: B
Enter new meaning: BST

1. Enter a word
2. Ascending order
3. Descending order
4. Search
5. Modify
6. Delete
7. Exit
Enter your choice: 6

Enter word you want to delete: Algorithm

1. Enter a word
2. Ascending order
3. Descending order
4. Search
5. Modify
6. Delete
7. Exit
Enter your choice: 2

Ascending order:
B
C

1. Enter a word
2. Ascending order
3. Descending order
4. Search
5. Modify
6. Delete
7. Exit
Enter your choice: 7

--------------------------------
Process exited after 150.2 seconds with return value 1
```