```
//============================================================================
// Name        : Assign5.cpp
// Author      : Shrikrushna S Zirape
// Version     :
// Copyright   : Your copyright notice
// Description : Hello World in C++, Ansi-style
//============================================================================

#include <iostream>
#include <bits/stdc++.h>
#include <string.h>
#include <cstring>
#define max 26
using namespace std;

class WordNode{
        string word, meaning;
        WordNode * next;
public:
        WordNode(){
                word = "";
                meaning = "";
                next = NULL;
        }
        WordNode(string wrd, string mean){
                word = wrd;
                meaning = mean;
                next = NULL;
        }
        friend class HT;
};

class HT{
        WordNode* dict[max];
public:
        HT(){
                for(int i=0; i<max; i++){
                        dict[i]=NULL;
                }
        }
        int hashFunction(string);
        void insert(string, string);
        void print();
        WordNode* search(string);
        void deleteWordNode(string);

};

int HT::hashFunction(string s){
        int len = s.length();
        int k=0;
        for(int i=0; i<len; i++){
                k+=s[i];
        }
        int x = k/len;
        x = x%max;
        return x;
}

void HT::insert(string wrd, string mean){
        WordNode *temp=new WordNode(wrd, mean);
        int k = hashFunction(wrd);
        if (dict[k] == NULL){
                dict[k]=temp;
                cout<<"\nNode Inserted Successfully";
                return;
```

```
        }
        else{ //condition for collision
                WordNode *itr = dict[k];
                while(itr->next != NULL){
                        itr = itr->next;
                }
                itr->next = temp;
                cout<<"\n Node Inserted Successfully";
                return;
        }
}

WordNode* HT::search(string key){
        int k = hashFunction(key);
        int comp = 1;
        WordNode *itr = dict[k];
                while(itr !=NULL){
                        if(key == itr->word){
                                cout<<"\nelement found\n";
                                cout<<"Meaning:- "<<itr->meaning;
                                cout<<"\nNo of comp :-"<<comp;
                                return itr;

                        }
                        comp ++;
                        itr=itr->next;
                }
                cout<<"\nKey not found";
                return NULL;


}

void HT::print(){
        cout<<"\nPrinting the Dictionary\n";
        for(int i=0; i<max; i++){
                cout<<i<<" - > ";
                WordNode *temp = dict[i];
                while(temp != NULL){
                        cout<<temp->word<<" = "<<temp->meaning<<"("<<this-
>hashFunction(temp->word)<<")"<< " | ";
                        temp = temp->next;
                }
                cout<<endl;
        }
}

void HT::deleteWordNode(string key){
        int index  = hashFunction(key);
        if(dict[index] == NULL){
                cout<<"\nKey Not Present";
        }
        else if(dict[index]->next == NULL){
                WordNode *temp = dict[index];
                dict[index]=NULL;
                delete temp;
                cout<<"\n Deleted";
                return;
        }
        else{
                WordNode *temp;
                temp = dict[index];
                WordNode *pre=NULL;
                while(temp != NULL){
                        if(key == temp->word){
                                if(pre == NULL){
```

```
                                        dict[index]=temp->next;
                                        delete temp;
                                        cout<<"\nNode Deleted";
                                        return;
                                }
                                pre->next=temp->next;
                                delete temp;
                                cout<<"\nNode Deleted";
                                return;
                        }
                        pre = temp;
                        temp = temp->next;
                }
                cout<<"\n key not found";
                return;
        }


}

int main() {
        HT h;
        int ch;
        string key, mean;
        do{

                cout<<"\n-------------------------------------
Menu-------------------------------------";
                cout<<"\n1.   Insert";
                cout<<"\n2.   Delete";
                cout<<"\n3.   Print";
                cout<<"\n4.   Search";
                cout<<"\n0.   Exit";
                cout<<"\n Enter your choice :-";
                cin>>ch;
                switch(ch){
                case 0:
                        cout<<"\n Ending the program";
                        break;
                case 1:
                        cout<<"\nEnter the key :-";
                        cin>>key;
                        cout<<"\nEnter the meaning :-";
                        cin>>mean;
                        h.insert(key, mean);
                        cout<<"\nInserted Successfully";
                        break;
                case 2:
                        cout<<"\nEnter key you want to delete :-";
                        cin>>key;
                        h.deleteWordNode(key);
                        cout<<"\nDeleted Successfully";
                        break;
                case 3:
                        h.print();
                        break;
                case 4:
                        cout<<"\nEnter the key you want to search";
                        cin>>key;
                        h.search(key);
                        break;
                default:
                        cout<<"\n Incorrect option";
                        break;
                }
```

```
        }while(ch!=0);

        return 0;
}
```