1 Dictionary Program

Lab 1 for CS124 is designed to allow for review of concept and techniques learend from CS102 and CS116. We will also explore new ideas related to data structures. It is primarily based on Ch.2 of C++ for You ++.

1.1 Header File: ”lab.h”

Our ”lab.h” file includes the structure ”entry”. The string variables ”english” and ”spanish” will be used as placeholders to store the parced data from our ”dict.dat” dictionary file.

```
#ifndef MAIN_H
#define MAIN_H
#include <string>
#include <iostream>
#include <fstream>
#include <vector>

struct entry{
    std::string english;
    std::string spanish;
};
#endif
```

1.2 Reading the ”dict.dat” File: ”readFile.cpp”

In the ”readFile” function, we pass our vector ”store” as a parameter (through direct reference). We create the following: variable ”a” of type ”entry” (our structure); two strings (”en” and ”s”) to store and pass on value to the structure variables; our variable ”ifs” of data-type ”ifstream” which passes in our ”dict.dat” file.

Our control structure (if-statement) opens ”ifs” as the condition. We then use a while-loop inside the control structure to parse the file, checking for tab-spaces to separate the data. The parsed data is passed on to our variables ”en” and ”s”, which is set equal to the variables inside our structure ”a”.

The structure is then pushed into our vector ”store”, allowing us to close the ”dict.dat” file.

```
#include <iostream>
#include <fstream>
#include <vector>
#include "lab.h"

void readFile(std::vector <entry> &store){
    entry a ;
    std::string en;
    std::string s;
    std::ifstream ifs("dict.dat");
    /* Testing to check if file is being read by program.
    if(ifs){ std::cout << "Got it!" << std::endl;} */
    if (ifs.is_open()){
        while (!ifs.eof()){
            std::getline(ifs, en, '\t');
            std::getline(ifs, s);

            a.english = en;
            a.spanish = s;
            store.push_back(a);
        }
    }
    ifs.close();
    /*
    The following loop was used for testing purposes to
    print out the stored data after the file closes.

    int storeSize = store.size();
    for (int i = 0; i < storeSize; i++){
        std::cout << "English: " << store[i].english << "; Spanish: " << store[i].spanish <<
    }
    */
}
```

1.3 Taking User Input: ”userInput.cpp”

Our ”userinput.cpp” file will be displaying a prompt to take user input. The vector ”store” is passed in by reference in the function’s parameter.

We create a while-loop to prompt the user using a boolean conditional. A user input of ”q” or ”Q” will end the loop and end the program.

Another while-loop checks for correct user input. For example, an input of integers would repeat the prompt and clear the previous input.

A successful user input is then taken and compared to every position inside the vector. Because our vector is of the data-type ”entry” (declared in ”lab.h”), the user’s input is stored as the variable ”english”.

```
#include <iostream>
#include <vector>
#include <fstream>

void userInput(std::vector <entry> &store){
    std::string userSearch;
    bool menu = false;

    while(!menu){
        std::cout << "Please enter a word or ’q’ to quit: " << std::endl;
        std::cin >> userSearch;

        if(userSearch == "q" || userSearch == "Q"){
            break;
        }

        while(!std::cin){
            std::cout << "Please try again or press ’q’ to quit: " << std::endl;
            std::cin.clear();
            std::cin.ignore();
            std::cin >> userSearch;
            if(userSearch == "q" || userSearch == "Q"){
                menu = false;
                break;
            }
        }
        double storeSize = store.size();
        for (int i = 0; i < storeSize; i++){
            if (userSearch == store[i].english){
            std::cout << store[i].spanish << std::endl;
            }
        }
    }
}
```

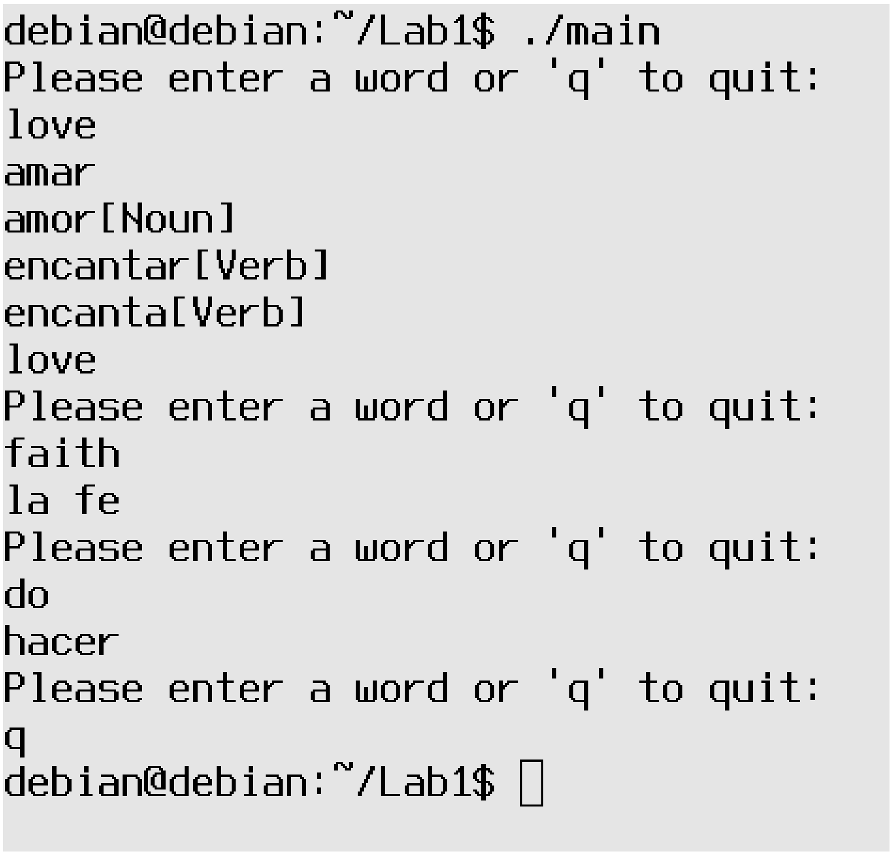
1.4 Main File: ”main.cpp”

The main function is used to bring all of our different files together. Inside, we declare our vector ”store” as a local variable, and we pass it on as a parameter into the functions ”readFile” and ”userInput”.

```
#include <iostream>
#include <fstream>
#include <vector>
#include "readFile.cpp"
#include "userInput.cpp"
#include "lab.h"

int main(){
    std::vector <entry> store;
    readFile(store);
    userInput(store);
    return 0;
}
```

1.5 Screenshots



```
debian@debian:~/Lab1$ ./main
Please enter a word or 'q' to quit:
love
amar
amor[Noun]
encantar[Verb]
encanta[Verb]
love
Please enter a word or 'q' to quit:
faith
la fe
Please enter a word or 'q' to quit:
do
hacer
Please enter a word or 'q' to quit:
q
debian@debian:~/Lab1$
```