Lab 1 - lab.h

```
#ifndef LAB_H
# define LAB_H

# include <iostream>
# include <fstream>
# include <vector>
# include <catch.hpp>

struct Entry{
        Entry(std::string w, std::string t) : word(w), translation(t) {}
        bool        operator==(const Entry &e) const{
                return word == e.word && translation == e.translation;
        }
        std::string word;
        std::string translation;
};

bool        loadDictionary(std::string f, std::vector<Entry> &v);
bool        lookUpWord(std::string w, std::string &t, const std::vector<Entry> v);
bool        insertWord(std::string f, std::string w, std::string t);

#endif
```

```
#include <lab.h>

int                main(void){
        std::vector<Entry> v;
        std::string        file;

        std::cout << "Please, enter dictionary file name: ";
        getline(std::cin, file);

        if (loadDictionary(file, v)){
                std::string        input;
                std::string w, t;

                while (!std::cin.eof()){
                        bool                err = true;

                        std::cout << std::endl;
                        std::cout << "What do you want to do?" << std::endl
                                  << "[1] Look up a word" << std::endl
                                  << "[ctrl + D] Exit" << std::endl;
                        getline(std::cin, input);
                        while (input == "1" && !std::cin.eof()){
                                std::cout << std::endl;
                                std::cout << "[ctrl + D] Exit" << std::endl
                                          << "Please, enter a word: ";
                                getline(std::cin, w);
                                if (lookUpWord(w, t, v)){
                                        std::cout << std::endl;
                                        std::cout << "English: " << w << std::endl
                                                  << "Italian: " << t << std::endl;
                                        err = false;
                                }
                                else if (!std::cin.eof()){
                                        std::cout << std::endl;
                                        std::cout << "Cannot find the word." << std::endl
                                                  << "[1] Look up another word" << std::endl
                                                  << "[2] Make a new word" << std::endl;
                                        getline(std::cin, input);
                                        if (input == "2"){
                                                t = "";
                                                while (t.empty() && !std::cin.eof()){
                                                        std::cout << std::endl;
                                                        std::cout << "Please, enter the translation for "
                                                                          << w << ": ";
                                                        getline(std::cin, t);
                                                        if (t.empty()){
                                                                std::cout << std::endl;
                                                                std::cout << "Wrong input. Please try again."
                                                                                  << std::endl;
                                                        }
                                                }
                                                if (!insertWord(file, w, t) || !loadDictionary(file, v)){
                                                        std::cout << "Failed to add a new word."
                                                                          << std::endl;
                                                }
                                                err = false;
                                        }
                                }
                        }
                        if (err && !std::cin.eof()){
                                std::cout << "Sorry, wrong input. Please try again."
                                                  << std::endl;
                        }
                }
        }
        else{
                std::cout << "Failed to open file " << file << "." << std::endl;
        }
        std::cout << std::endl;
        return 0;
}
```

```
#include <lab.h>

bool loadDictionary(std::string f, std::vector<Entry> &v){
        std::string                title, w, t;
        std::ifstream        ifs(f);
        bool                        r = ifs;

        getline(ifs, title);
        while (ifs >> w >> t){
                v.push_back(Entry(w,t));
        }
        return r;
}

/*
TEST_CASE("Testing fill vector"){
        std::vector<Entry> v;

        REQUIRE(v.size() == 0);
        REQUIRE(loadDictionary("Dictionary", v));
        REQUIRE(v.size() == 4);

        Entry e1("want", "volere");
        Entry e2("a", "un");
        Entry e3("I", "io");
        Entry e4("this", "questo");

        std::vector <Entry> ev;
        ev.push_back(e1);
        ev.push_back(e2);
        ev.push_back(e3);
        ev.push_back(e4);
        REQUIRE(v == ev);
}
*/
```

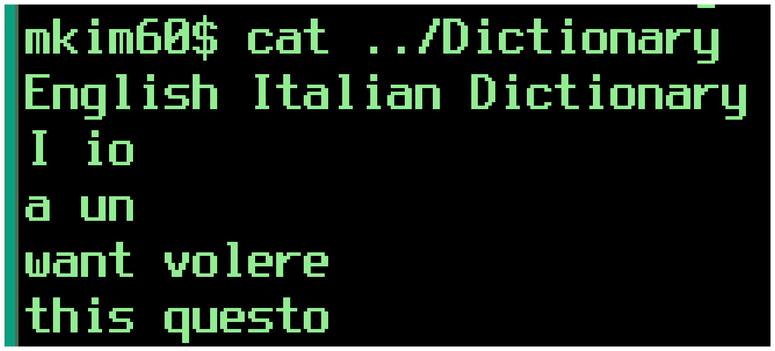
Lab 1 - lookUpWord.cpp

```
#include <lab.h>

bool        lookUpWord(std::string w, std::string &t, const std::vector<Entry> v){
        bool        r = false;
        for (size_t i = 0; i < v.size(); i++){
                if (w == v[i].word){
                        r = true;
                        t = v[i].translation;
                        break ;
                }
        }
        return r;
}

/*
TEST_CASE("Testing find word"){
        std::vector<Entry> v;
        std::string t;

        REQUIRE(loadDictionary(DICTIONARY, v));
        REQUIRE(v.size() == 4);

        REQUIRE_FALSE(lookUpWord("hello", t, v));
        REQUIRE(lookUpWord("I", t, v));

        REQUIRE(lookUpWord("this", t, v));
        REQUIRE(t == "questo");
}
*/
```

Lab 1 - insertWord.cpp

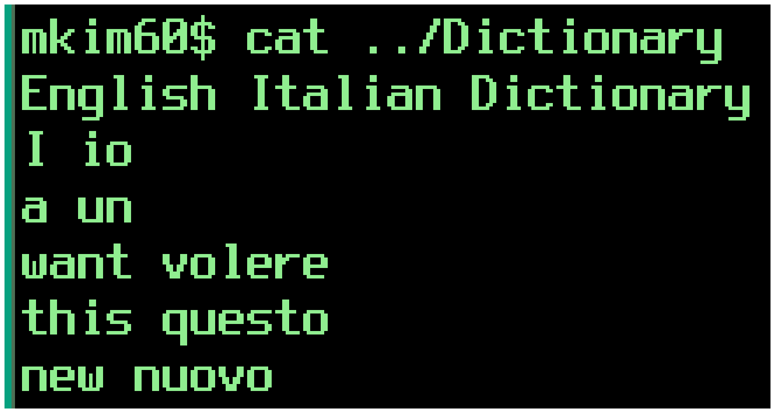
```
#include <lab.h>

bool        insertWord(std::string f, std::string w, std::string t){
        std::ofstream        ofs(f, std::ofstream::app);
        bool                        r = ofs;

        if (r){
                ofs << w << " " << t << std::endl;
        }
        return r;
}

/*
TEST_CASE("Testing insert word"){
        std::vector<Entry> v;
        std::string w, t;

        REQUIRE(loadDictionary("Dictionary", v));
        REQUIRE(v.size() == 4);

        REQUIRE_FALSE(lookUpWord("hello", t, v));
        REQUIRE(insertWord("Dictionary", "hello", "world"));

        REQUIRE(loadDictionary("Dictionary", v));
        REQUIRE(lookUpWord("hello", t, v));
        REQUIRE(t == "world");
}
*/
```

Before



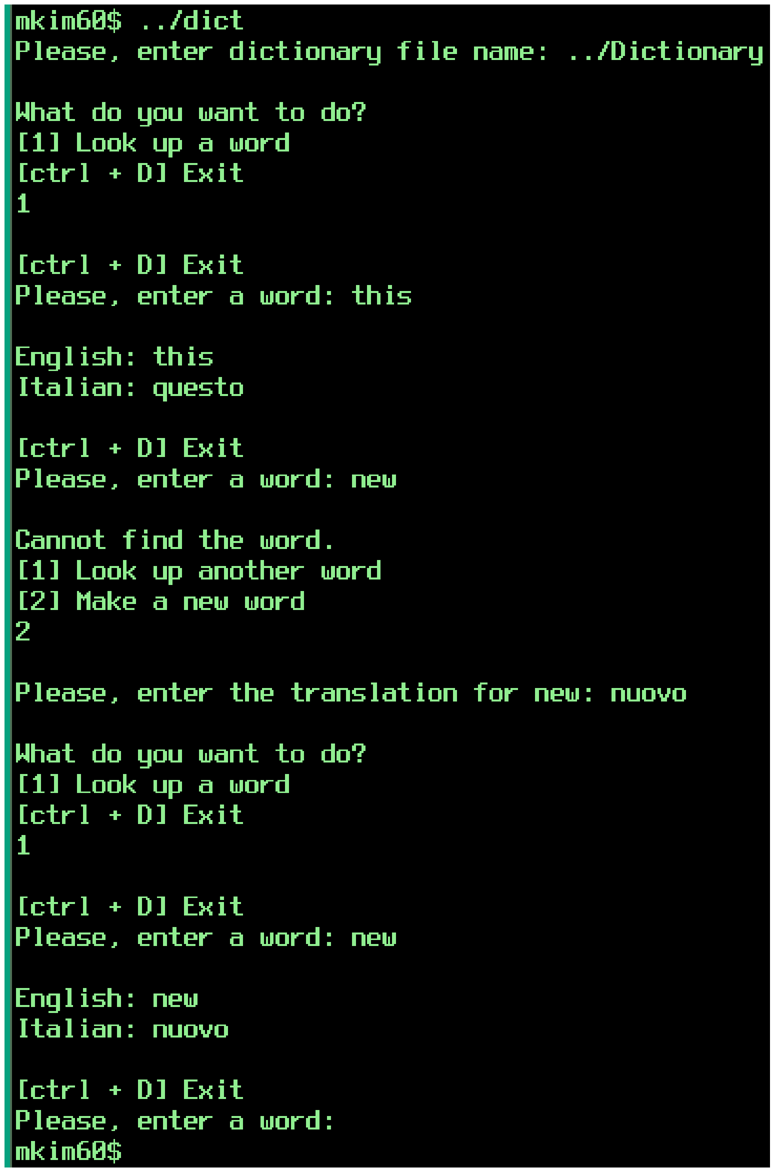
```
mkim60$ cat ../Dictionary
English Italian Dictionary
I io
a un
want volere
this questo
```

After



```
mkim60$ cat ../Dictionary
English Italian Dictionary
I io
a un
want volere
this questo
new nuovo
```

Result



```
mkim60$ ../dict
Please, enter dictionary file name: ../Dictionary

What do you want to do?
[1] Look up a word
[ctrl + D] Exit
1

[ctrl + D] Exit
Please, enter a word: this

English: this
Italian: questo

[ctrl + D] Exit
Please, enter a word: new

Cannot find the word.
[1] Look up another word
[2] Make a new word
2

Please, enter the translation for new: nuovo

What do you want to do?
[1] Look up a word
[ctrl + D] Exit
1

[ctrl + D] Exit
Please, enter a word: new

English: new
Italian: nuovo

[ctrl + D] Exit
Please, enter a word:
mkim60$
```