COMP2045 Coursework 2

Introduction

This coursework is worth **10% of the module mark**. It requires you to write a C++ program that solves the task described below. The deadline for this exercise is **23:55pm on Sunday 31th March 2019**.

Read the entire document before beginning the exercise.

If you have any questions about this exercise, please ask in the Q&A forum on Moodle, after a lecture, in a lab, or during the advertised office hours. Do not post your program or parts of your program to Moodle as you are not allowed to share your coursework programs with other students. If any questions require this exercise to be clarified then this document will be updated and everyone will be notified via Moodle.

Submission

You must submit two files: a report and a single C++ source code file containing all your code for this exercise. The code file must be called fuzzysearch.cpp and it should contain all the source code.

The first line of your program should be a comment which contains your student ID number, username, and full name, of the form:

```
// 6512345 zy12345 Joe Blogs
```

The program must compile without warnings or errors when I use the command

```
g++ -std=c++11 fuzzysearch.cpp -o fuzzysearch
```

This command will be run on our Linux server cslinux. If it does not compile, for any reason, then you will lose all the marks for testing. The completed source code file and report should be uploaded to the Coursework 2 Submission link on the Moodle page.

Late submissions: Late submissions will lose 5 percentage points per hour, rounded up to the next whole hour. No late submissions will be accepted more than 24 hours after the exercise deadline. If you have extenuating circumstances you must file them before the deadline.

Task

Your task is to write a C++ program that makes inaccurate search. Given an arbitrary string and a text file, your program should output how many occurrence if the string is contained in the file. If the string is not contained in the file, your program should output the most similar string contained in the file.

The filename of the text file will be specified as a command line argument. Your program should load this file and then prompt the user to enter a string to search for. There should not be limitation on the length of input string.

After the user has entered a string, the program should output the occurrence of the string or the most similar string contained in the file. It should repeatedly do this until the user enters an empty string.

If the program is run without the correct number of command line parameters, it should exit with the message
"Invalid command line arguments. Usage: ./fuzzysearch textfile". If the file cannot be opened, print the error message "Cannot open file".

In your report, you should explain what method or algorithm is used for string comparison, how you implement the algorithm in class/objects, how your program performs in making fuzzy search. The report should be in pdf format and it should be **not more than** one page.

Example input/output

Given the following text file (assume that the filename is helloworld.txt, with the file in the current directory)

This is an example text file.
Hello World! Hello C++! Hello UNNC!

Running the program and just pressing return:

```
zlizpd3 $ ./fuzzysearch
Invalid command line arguments. Usage: ./fuzzysearch textfile
zlizpd3 $
```

Running the program:

```
zlizpd3 $ ./ fuzzysearch helloworld.txt
Search for:
zlizpd3 $
```

Running the program:

```
zlizpd3 $ ./ fuzzysearch helloworld.txt
Search for: Hello
3 occurance found in the file
Search for: hello
0 occurance found in the file. Most similar string: Hello
Search for: Hello C++
1 occurance found in the file
Search for: hello world
0 occurance found in the file.
Most similar string: Hello World
Search for: I like C++ programming
0 occurance found in the file.
Most similar string:  Hello C++! Hello UNNC
Search for:
zlizpd3 $
```

If the text file cannot be opened:

```
zlizpd3 $ ./ fuzzysearch missing.txt
Cannot open file
zlizpd3 $
```

Note: The above examples simply show one of possible solutions. It is not necessary that a ‘correct’ program must output the same results as the above examples.

Marking

Your program should correctly implement the task requirements. A number of tests will be run against your program with different input data designed to test if this is the case for each individual requirement. The marking criteria include:

- Does the program find similar strings effectively?
- Does the program deal with different input correctly?
- Does the program output match the required format?
- Does the program work with class, object, pointer, and reference properly?

If your program does not compile then you will lose all marks. Each warning message leads to -1 mark.

Plagiarism

You should complete this coursework on your own. Anyone suspected of plagiarism will be investigated and punished in accordance with the university policy on plagiarism (see your student handbook and the University Quality Manual). This may include a mark of zero for this coursework.

You should write the source code required for this assignment yourself. If you use code from other sources (books, web pages, etc), you should use comments to acknowledge this. You must not copy or share source code with other students. You must not work together on your solution. You can informally talk about higher-level ideas but not to a level of detail that would allow you all to create the same source code.

If I have concerns about a submission, I may ask you to come to my office and explain your work in your own words.