Practical work nr. 11 - Files

Subjects

- Text files
- Binary files
- Command Line Arguments

Exercises

1. Develop a program which reads a list of real numbers stored in a file. The program should then calculate the average and standard deviation and print on screen the values larger than the average. You may reuse the functions developed in exercise 1 from class #10. Consider that the file contains a single value per line, as in the example below. The file's name should be requested to the user.

nums.txt

```
12.39
1.93
7.85
...
```

2. Return to problem 2 from class #10 and add an option to store the information about all the students in a text file. The file's name should be requested to the user.
3. In order to test the previous program and verify if the file was correctly created, add another option which allows reading student information stored in a file. This option should replace an eventual list of students previously existing in the program. The file's name should be requested to the user.
4. Write a program which determines the amount of times all letters appear in a text file. The file's name should be passed as an argument through the command line (explore the definition of the main function with two parameters - `int main (int argc, char *argv[])`. Download some text file examples from Project Gutenberg[1] and do the counting. Adjust the program to only consider letters and not distinguish lower case from upper case letters.
5. Write a program which verifies if two files are equal. In order to save time and memory, read and compare 1 KiB block as a time, and terminate as soon as a difference is detected. The file's name should be passed as a command line argument.

[1] https://www.gutenberg.org/

6. The file /etc/dictionaries-common/words contains a list of words (in english), one per line. Some of those words have equal consecutive letters, such as “accent", "access", "pool", etc. Develop a program to list and count all those words. Suggestion: start by creating the function which returns the number of letter pairs found in a string `s`:

```
int repeats(const std::string &s)
```