C++: Practical session 12

1 Analysis of British Prime Ministers

Write a program that successively does the following:

1. Read in the list of names of British Prime Ministers from the file `https://www-internal.lsc.phy.cam.ac.uk/pmb39/PMs.txt` into two `std::vector`s, one for forename and one for surname.
2. Note that you only need to use: `inputFile >> firstName >> surname` as the separate names are guaranteed to be a string with no spaces.
3. Use `std::find` to find the first PM called George, and print his position in the list and his full name.
4. Use `std::find` to find the first PM called Richard, and print his position in the list and his full name.
5. Use `std::count` to count the number of PMs called Charles.
6. Use `std::adjacent_find` repeatedly to find all instances where successive PMs had the same first name.

You will need to use a C++ reference to find the correct syntax. Try `https://en.cppreference.com/w/cpp/algorithm/find` etc.

1.1 Extensions

Redo the exercise using a single `std::vector<std::pair<std::string, std::string> >` to store the first names and surnames. This will require the use of predicate functions to access only the first element of the `pair`.

2 Frequency analysis

(Somewhat harder; only do if you have the time and inclination)

For this practical you will need to look up the available algorithms implemented within the `std::string` class.

Write a program that conducts a frequency analysis on a string provided by the user, and outputs a data file for use in `gnuplot`.

All punctuation should be ignored, and capitals should be treated the same as lower-case. You should use a suitable STL algorithm. Download `http://www.gutenberg.org/files/829/829-0.txt` and perform the same analysis on this.

Write a program that counts the number of words in a given text (consider first the definition of a word). You may need to perform some whitespace transformations before doing the full analysis. Determine the frequency of the word "the" in the previous text. Comment on the computational complexity of the preceeding algorithms. Determine the frequency of all words found in the document. You should use a `std::map` to store the frequency of all words found.