Software Engineering 1: Introduction to Computing

Programming test A – Spring 2015

Max Cattafi (m.cattafi@imperial.ac.uk)

It's in your interest to fully read this document at least once before writing any code.

A shop has two warehouses where items are stored. The respective content of the warehouses is kept in two text files, see for instance the following (which might be the content for warehouse A):

```
apple 12
banana 14
mango 4
orange 15
pineapple 10
```

and the following (which might be for warehouse B):

```
avocado 9
banana 22
papaya 5
```

Notice that the files contain on each line the name of an item and the quantity available (expressed in some unit of measure) and that they are alphabetically sorted.

We would like to merge the information contained in the two files. The merging should keep the alphabetical order. Moreover, when the same item is present in both inputs, in the output the associated quantity should be the sum of the quantities in input.

For instance if the previous two files are given in input the output on screen should be:

```
apple 12
avocado 9
banana 36
mango 4
orange 15
papaya 5
pineapple 10
```

Write a program in C++ which:

- Reads from the user the names of two text files with the content of the two warehouses (for testing you will have to prepare the files with a text editor).
- Reads from the files and stores the data into two C++ vectors (define a suitable structured type).
- Merges the content of the vectors. For this purpose, define and use a function `merge_items`.
- Prints on the screen the outcome.

Guidelines

Remember that you can use the `<` operator on C++ strings (as you would do with e.g. integer variables) which returns `true` if the left hand side comes before the right hand side in alphabetical order and `false` otherwise.

None of the functions, except for the `main`, should contain user or file input or output (`cin`, `cout`, `ifstream` etc) in their implementation: `cin`, `cout`, `ifstream` etc should only be in the `main`.

All the variables should be declared in the scope of a function (either the `main` or some other one): there should be no global variables.

All the loops should be controlled either by the loop condition or by `return`: there should be no `break`, `continue` or `goto` statements anywhere in the program.

Functions should not *return* vectors provided as output, vectors should be provided in output using output parameters (i.e. passing by reference).

Only headers from the standard library are allowed.

The use of the header `<algorithm>` is not allowed.

Do not use C++11 features or non-standard features.