## Lab work #5

**Topics**

* Containers
* Algorithms from standard template library

**Exercises**

1. Consider the file stocks.csv available on the base files folder. Each line has the following information:

Name, Date, OpeningPrice, MaxPrice, MinPrice, ClosingPrice, Volume

Implement functions that would calculate and print the following parameters:

* 1. mostTraded - the most traded company (with the largest total volume);
  2. highestDiff - company with the highest daily appreciation (difference between the opening price and the closing price);
  3. company with the highest valuation (stock growth) during the period to which the file refers
  4. day and value at which each share reached the highest value;
  5. Create a function that calculates the valuation of a given portfolio between two given dates. The portfolio should be a dictionary with the number of shares of each title, eg: {'NFLX': 100, 'CSCO': 80}.

1. Using the program stl\_analysis.cpp, compare the performance of the vector C++ container and some STL algorithms. Fill the following table with the execution times obtained for different dimensions of the vector (constant LEN). Include the information about the processor of your computer.

**Computer**: Intel Core i5 @ 2.00GHz (example)

| **LEN** | **fill** | **find** | **sort** | **bin\_search** | **pop** | **erase** |
| --- | --- | --- | --- | --- | --- | --- |
| 10000 |  |  |  |  |  |  |
| 20000 |  |  |  |  |  |  |
| 200000 |  |  |  |  |  |  |
| 400000 |  |  |  |  |  |  |
| 4000000 |  |  |  |  |  |  |
| 40000000 |  |  |  |  |  |  |

1. Based on the program stl\_analysis.cpp, implement the necessary code to test the following methods related to the Set container: insert, find and erase. Fill a table similar to the one of the previous exercise. Consider also the following recommendations:
   1. insert: if inserting random numbers, verify the set size at the end and compare it with the number of insertions made;
   2. find: compare the execution time of the find member function provided by the class with that provided by the STL Algorithms.
2. The file /etc/dictionaries-common/words contains a list of ordered words (in english), one per line. Read these words into a vector and, using a binary search function, find out how many words begin with "ea", without having to go through everything. Hint: Look for the first word with "ea" and the first with "eb" and subtract the indices. How many words start with "tb"? None? So what is the first letter, greater than 'b', that occurs after a "t" in English words?
3. Using the same principle, write a function that indicates all the letters that can succeed a certain prefix. You can use this function in an intelligent writing system that is presenting the possible letters to complete a certain prefix already entered. When the user enters another letter, it is added to the previous prefix and a new list of possibilities is presented, and so on.