Exercise 11:

Submit your solutions (source code) to the questions below by email to your instructor and TA(s) by Monday, December 17th (16:30).

Question 1: A queue built from two stacks (50 points).

The goal of the exercise is to define a queue by using two stacks. You can use the stack adapter container provided by the STL. Please complete the following definition of a queue in a file named "Queue.h".

```
#ifndef QUEUE_H
#define QUEUE_H

#include <vector>
#include <stack>

template <typename T, typename Container=std::vector<T> > class Queue {
public:
    typedef typename Container::value_type value_type;
    typedef typename Container::size_type  size_type;
    typedef Container container_type;

    explicit Queue(const Container& contents = Container());
    bool empty() const;
    size_type size() const;
    value_type& front();
    const value_type& front() const;
    value_type& back();
    const value_type& back() const;
    void push(const value_type& x);
    void pop();
};
#endif
```

You need to implement each method above. For that purpose you need to use two intermediate stacks (you can use the stack provided by the STL).
To test your implementation, you can use the following code (to be typed in the file named "test_Queue.cpp"

```
#include <iostream>

#include "Queue.h"

int main (void) {
 Queue<int> q;
 q.push(1);
 q.push(2);
 q.push(3);

 while (!q.empty()) {
  std::cout << q.front() << std::endl;
  q.pop();
 }

 return 0;
}
```

The output for the previous program should be:

```
1
2
3
```

Question 2: Keyword counter (50 points).

C++ has a set of keywords (reserved words) that have a specific meaning to a compiler. Examples of such keywords are: int, double, if, else, switch, etc. We are interested in writing a program that answers the following question: given a C++ source file how many times does each keyword appear in the file.

The list of C++ keywords is given in the file keywords.txt that you need to download. The file is structured such that each line of the file contains a C++ keyword. Here are the first lines of the file keywords.txt:

```
keywords.txt:
and
and_eq
asm
auto
bitand
```

```
bitor
...
```

Given this list of keywords, your task is to write a program that prompts the user for a file name (corresponding to a C++ source file), load the content of the file in memory, then reports the frequency of each keyword that occurred in that source file. Only keywords that occur in the file should be reported. To simplify the problem, we make the following assumption: we count keywords wherever they appear, even if it is in comments or inside a string.

For example, the result obtained with the file Date.cpp is:

```
$ ./keyword_counter
Enter filename: Date.cpp
Keyword: int occurred: 3 times
Keyword: void occurred: 2 times
```