Exercise 8. *Deadline: 11 May 2009.*

COURSE OF C++ PROGRAMMING LANGUAGE

STACK AND QUEUE

University of Wrocław
Institute of Computer Science *Paweł Rzechonek*

Exercise

Define classes `MyStack` and `MyQueue`, which will represent respectively a stack (LIFO) and queue (FIFO) for real number (of type `double`). Use class `deque<double>` from STL as a base class.

```
class MyStack : protected deque<double>
{
    // ...
};

class MyQueue : public MyStack
{
    // ...
};
```

A stack is traditionally defined as any object that implements the following operations:

- `push(x)` — push a new element `x` onto the stack
- `pop()` — remove and return the topmost element from the stack
- `top()` — return (but do not remove) the topmost element in the stack
- `size()` — return number of elements in the collection
- `empty()` — return `true` if the collection is empty

A queue is similar to the stack and traditionally defined as any object that implements the following operations:

- `push(x)` — push a new element `x` on to the end of the queue
- `pop()` — remove and return the element at the front of the queue
- `front()` — return (but do not remove) the element at the front of the queue
- `back()` — return the element at the end of the queue
- `size()` — return number of elements in the collection
- `empty()` — return `true` if the collection is empty

You should define an operator `<<` for these classes.

Finally write a short program, which will test your implementation of the stack (class `MyStack`) and the queue (class `MyQueue`). Test their copy constructors and copy assignments too.

Suggestion

Partition your code into the header and source files.

Hint

Some information about the class `deque<>` from STL (a double–ended queue) can be found on the webpage:

`http://en.wikipedia.org/wiki/Deque`

Some information about the stack and the queue data structure can be found on the webpages:

`http://en.wikipedia.org/wiki/Stack_(data_structure)`

`http://en.wikipedia.org/wiki/Queue_(data_structure)`