

C++ for Numerical Methods: Class Test

December 2010

Time allowed: 2 hours.

Attempt all parts. The test is open-book and you may use online resources, but **exam conditions still apply**. You **must not** copy source code from other students or online resources. Please make sure you are not signed on to any instant messaging applications (e.g. MSN or Facebook).

Consider the problem of shuffling a deck of cards which are labelled 1 through n . We define two operations, called *reversing* and *riffling* which manipulate the deck as follows:

- **Reversing:** reverse the order of cards in the deck;
- **Riffling:** if n is even, split the deck into two equally sized halves; otherwise, split the deck into two pieces of size $(n+1)/2$ and $(n-1)/2$. Then, interleave the cards from each pile.

As an example, consider a deck of 5 cards labelled 1 through 5. The effect of these operations is:

- **Reverse:** 1 2 3 4 5 \rightarrow 5 4 3 2 1
- **Riffle:** 1 2 3 4 5 \rightarrow 1 4 2 5 3

Your task is to write a C++ program which allows a user to choose a deck of arbitrary size and run these shuffling operations on that deck. In this program you will represent the deck of cards by a vector of `ints`.

To begin, download the source file:

<http://go.warwick.ac.uk/ma903/exam-template.cpp>

This is a template file – the structure of the code is written, but the implementation is missing. Your aim should be to complete all the functions in this file; however, **do not change any of the function definitions in this file**. You should only write code wherever you see the comment “insert your code here”.

There are five functions in this file, four of which you need to complete (including `main`). The one exception is `printDeck`, which you can use to print out the deck. You should then:

1. Implement the `main` function. This should ask the user for n , the length of the deck, create a `vector<int>` of this length and fill it with integers from 1 to n . You should ensure that $n \geq 1$ before allowing the user to continue.
2. Fill in the `reverseDeck` function, which takes a `vector<int>` by reference and apply the reverse operation. Test your implementation using `main` before moving on.
3. Similarly, complete and test the `riffleDeck` function, which applies the riffle operation.
4. The final function you need to complete is `shuffleDeck`. This takes two parameters: the deck and a string `order`. `order` is a string containing the characters `i` and `r`, where `i` denotes a riffle and `r` a reverse. `shuffleDeck` should shuffle the deck according to the order of the characters in `order`. For example, `"ir"` first riffles the deck, then reverses it.
5. Finally, extend your `main` function to ask the user for the `order` string from the user. It should check that the only characters in the string are `r` and `i` before passing it to `shuffleDeck`.

Bonus marks: Extend `shuffleDeck` to handle strings of the form `"12ir3i"` – i.e. 12 riffles, a reverse and 3 more riffles.

At the end of the test: e-mail your solution code to d.mcdougall@warwick.ac.uk.