Function Templates

This assignment gives you practice with function templates, overloaded functions, pointers and arrays. The goal is to
implement several function Templates that work on arrays, or more precisely, ranges . You will implement functions that perform operations on ranges such as removing elements within the range, replacing elements within the range, searching for elements, copying one range to another range, etc. You will have a header file that looks like this:

```
//---------------------------------------------------------------------------
#ifndef FUNCTIONS_H
#define FUNCTIONS_H
//---------------------------------------------------------------------------
namespace CS225
{
template <typename T> void swap(T *left, T *right);
/*
* Other function templates declarations for count, remove, replace, etc.
* go here.
*
*/
}
#include "functions.cpp"
#endif
//---------------------------------------------------------------------------
```

Your .cpp file will contain several functions. The sample driver shows many of them. You will need to figure out what others are required. Many of the functions are going to be very similar in their implementation. However, for this assignment, don't be tempted to try and factor out the minimal common code into a separate function, because it will only complicate matters. Once you understand the concept of a range, you will see that the amount of code is not that great. (It never is.) The most complex function is the remove function, so you should work on that one last.
As you implement the functions, you should begin to see a pattern emerging in your code. This should help you understand the purpose of using pointers (a range) with these arrays instead of relying on the size.

Other criteria

- You must make sure that your functions can deal with the appropriate calls. This means you need to decide how and when to you const in your code. The driver isn't going to test all cases so you'll definitely need to add more tests.
- Remember: function templates are only generated if you call the function. Don't end up getting a 0 due to your code failing to compile because you didn't add the necessary test cases. Your code will fail to compile if you forgot a case.
- Do not include any header files other than iostream in your .cpp file. (You don't need any others.)
- You will need to use std::cout to print out the elements in the Display function. Make sure you format the output exactly as shown in the output-sample.txt file as I am not providing the output function for this assignment.
- You must include the .cpp file at the end of the .h file exactly as shown above.

Submit:
functions.cpp and functions.h