Assignment 1: Library

Goals: Write a C++ class with separate .h and .cpp file, use partially filled arrays,

Create a new class called Library that uses a private string array to store the list of books and supports the following functions:

```
// Add a new book,
// return true for success, false if book already in library
bool AddBook(const std::string &name);

// Remove a book,
// return true for success, false if book not in library
bool RemoveBook(const std::string &name);

// List all books in library
void ListAllBooks() const;

// Return true if book in library, false otherwise
bool IsInLibrary(const std::string &name) const;


// friend function
std::ostream &operator<<(std::ostream &out, const Library &lib);
```

Sample ass1.cpp, expand this as necessary to fully test your class:

```
#include <iostream>
#include <cassert>
#include "library.h"

void Test1() {
   Library libs("UWB");
   libs.AddBook("Don Quixote");
   libs.AddBook("In Search of Lost Time");
   libs.AddBook("Ulysses");
   libs.AddBook("The Odyssey");
   libs.ListAllBooks();
   // should generate already in library message and return true
   bool result = libs.IsInLibrary("The Odyssey");
   assert(result);

   // cannot add book twice, result should be false
   result = libs.AddBook("The Odyssey");
   assert(!result);
```

```
    // test remove, result should be true
    result = libs.RemoveBook("The Odyssey");
    assert(result);

    // not in library, result should be false
    result = libs.IsInLibrary("The Odyssey");
    assert(!result);

    // cannot remove twice, result should be false
    result = libs.RemoveBook("The Odyssey");
    assert(!result);
}


void TestAll() {
    Test1();
    std::cout << "Successfully completed all tests." << std::endl;
}

int main() {
    TestAll();
    return 0;
}
```

Under unix, compile your code using

```
g++ -std=c++14 -g -Wall -Wextra ass1.cpp library.cpp -o ass1
```

You need to submit `ass1.zip` with the following files in it. See course assignments page for instructions on how to create it

`library.h` - the prototypes of Library class functions (must support at least the functions shown in main above)
`library.cpp` - the implementation of Library class functions
`ass1.cpp` - tests demonstrating the Library constructor and functions
`output.txt` - See course assignments page for instructions on how to create it
`selfassessment.txt` - See course assignments page for the template

Tips & Hints

- You can set the maximum number of books in the library to be 100 and use that to set the size of your array. Use a private variable in the library class rather rather than hard coding the number 100 into your code, such as `static const int MAX = 100;`

- Since the library does not always have 100 books, you should have a private variable, such as `numberOfBooks`, to keep track of how many books are in the library (i.e. how much of the array is used)
- Removing an element from an array requires you know the index of that element, so your Library class might need a private function such as `int findBook(const std::string& name) const;` This would return the valid index if book is in library or -1 if book is not in library. Once you have the index of the book you want to remove, you can copy the last book in the array to overwrite the book being removed and then decrease the `numberOfBooks` variable. For example, your library might have 4 books ["A", "B", "C", "D"] with `numberOfBooks` as 4. If we are removing “B”, we find its index which is 1. We then copy the last book at `numberOfBook`s - 1 to that index, making the library ["A", "D", "C", "D"]. We decrease the `numberOfBooks` variable, so the library is now ["A", "D", "C"]. The array still has “D” at index 3, but that part of the array is not being used.