**Homework 3: Practice with recursion**
**ECE 2574, Spring 2018**
**Due before 5:00 pm on March 2**

**Honor Code:** You must work *independently* on this assignment. You may copy and/or modify any code that you see in the textbook, and you may copy/modify any code provided to you by the instructor or GTA. Referring to other source code is not permitted. Please review the statement in the syllabus.

**Objectives:** The main goal of this assignment is to gain experience with *recursion*. You will also reinforce important concepts related to testing and debugging of code.

**Project specification:** Working from the starter code, develop and test several recursive functions.

1) multiplication by using repeated addition

$m * n = m + m + \ldots + m$

Use recursion to implement the repetition needed for the addition operations.

2) exponentiation by using repeated multiplication

$m^n = m * m * \ldots * m$

Use recursion to implement the repetition needed for the multiplication operations.

3) palindrome detection

A palindrome is a string that is the same if it is reversed. Examples are "mom" and "racecar" and "able was I ere I saw elba". Write a function to determine whether a given string is a palindrome. Your function should be case-sensitive, which implies that "Mom" is not a palindrome. Do not give special treatment to white-space characters. It is acceptable for you to use methods provided by the C++ string library, such as `std::string::length()`. More of these library methods are described in Appendix A, section A.7, and in Appendix H of the textbook.

4) binary search of an array

This problem is described extensively in section 2.4.2 of the textbook.

5) copy one array to another in reverse order

See section 2.4.1 of the textbook for hints on how to proceed.

More details are provided in the interface file, `hw3_utilities.h`. This assignment does not rely on OO principles. In particular, you do not need to create any classes. Write C++ functions that could be called from a `main()` function. Place the functions in `hw3_utilities.cpp`.

**Starter code** is provided in file `hw3_starter.zip`. This zip archive contains several files similar to those that you received for previous assignments. Here is a summary of files that are provided to you:

`CMakeLists.txt`: Do not change this file, and do not submit it. This file is used by CMake to create a VS project named `hw3_test`. It creates other VS projects also, but `hw3_test` is the one that you will work with to develop code.

`catch.hpp`: Do not change this file, and do not submit it. It is needed by the Catch utility. (The Inginious auto-grader appears to be sensitive to particular versions of this file.)

`hw3_utilities.h`: Do not change this file, and do not submit it. It contains a specification of the functions that you need to write.

`hw3_utilities.cpp`: Write code in this file to implement the functions that are described above. Submit this file as part of your solution.

`hw3_test.cpp`: Write code in this file to test your recursive functions. Submit this file as part of your solution. This file uses the Catch utility to create a main() function, and to implement test cases of your code.

`hw3_main.cpp`: This file is for your (optional) use only. Do not submit this file. It is here in case you would like to create your own main() function and write client code to experiment with your recursive functions.

Notice that file `hw3_utilities.h`. contains the following line:

```
#include "hw3_utilities.cpp"
```

The reason for this is to support templates, which are needed by two of the functions that you will write. Because of this `#include`, the file `hw3_utilities.cpp` must not be associated as a Source File within Visual Studio; instead, it should show up automatically under External Dependencies.

**Additional requirements:** At the beginning of your test file `hw3_test.cpp`, place a comment block similar to the following:

```
//////////////////////////////////////////////////////
// ECE 2574, Homework 3, Jane Doe ← your name here
//
// File name:   hw3_test.cpp
// Description: (Describe the purpose of this file)
// Date:        2/24/2018  ← completion date here
//
```

Add appropriate comments throughout all of the code that you modify or create.

**Online testing:** We will continue to use the INGInious autograding system to determine part of your grade for this assignment. To use the autograder, place the following C++ source files into a single zip file: `hw3_utilities.cpp` and `hw3_test.cpp`. The zip archive must contain these files only, without any directories. A suggested file name for the zip archive is `hw3_`*name*`.zip`, where *name* is your family name.

Submit your zip file to the INGInious autograder at https://grader.ece.vt.edu. It will attempt to compile and run the tests that you have formulated in `hw3_test.cpp`, and the autograder will also run some "instructor" tests. A grade will be reported to you, proportional to the number of tests that have executed correctly.

You can submit to the autograder as many times as you like, but it will limit you to 4 submissions every hour. This limit is to discourage people from using it as their only compiler and overloading the machine.

**Submission to Canvas:** After you are satisfied with your code based on your autograde results and addressing the requirements, upload the same a zip file (as described above) to Canvas at the HW3 assignment link. You do not need to submit any other files or directories.

Be careful to verify that you have uploaded the correct files to Canvas. After you have uploaded your zip file, it is suggested that you also download it from Canvas and verify that it is correct.

**Grading:** There are 100 points allocated to this assignment, and most of the grade will be determined by the autograder.

- Correctly submitting the required files to INGInious and to Canvas: 5 points
- Your tests compile: 0 points
- Your tests pass: 20 points (proportional)
- Instructor's tests compile with your code: 0 points
- Instructor's tests pass: 50 points (proportional)
- No memory leaks (checked by autograder using `valgrind`): 10 points
- Good coding style, and your tests are reasonably thorough: 15 points (assessed manually by GTA)

Please note that your most recent submission to INGInious determines the autograded portion of your grade.