Last updated 31 October, 2014

Assignment 6:

Please see Submission Checklist (below) for submission requirements.

It is time to bring together some concepts we have been learning!

Remember that arrays can be tough to think about, so draw some out with all the parts of an array on paper to get an idea firmly in your mind about how they work!

(10 points) Programming Styles and Convention.
http://classes.engr.oregonstate.edu/~jessjo/CS161/OSU_IntroCodeStandards_v1.1.pdf

(30) Remember to submit your report! **Important – starting with this assignment, your report only needs to cover the project, not the exercises.**

(20) Exercise components:

1. (4) Write a function that takes as parameters an array of ints and the size of the array. It should return the (sum of the even values) – (sum of the odd values). In main you will need to query the user for the size of the array and the values in the array, then call the function and display the return value.

   File must be called: evenOdd.cpp

   Discussion idea: Why do you need to pass the size of the array to the function?

2. (4) Write a function that takes as parameters two **C-style** strings of the same length. Upon completion of the function, the second string should be a random rearrangement of the characters in the first string. The first string should remain unchanged. In main ask the user for a string, pass that and another (empty) string to the function, then display the value of the second (formerly empty) string. The function call in main should be in a loop that will keep asking the user if they want another random rearrangement (of the same string - it doesn't ask the user for a new string).

   File must be called: stringMix.cpp

   Discussion ideas: How is a c-style string different than an array of characters? Why don't we need to pass the size of the strings to the function?

3. (4) Generate a text-based histogram for a quiz given to a class of students. The quiz is graded on a scale from 0 to 5. Write a program that allows the user to enter grades for each student. As the grades are being entered, the program should count, using an array, the number of 0's, the number of 1's, the number of 2's, the number of 3's, the number of 4's, and the number of 5's. The program should be capable of handling an arbitrary number of student grades.

   You can do this by making an array of size 6, where each array element is initialized to zero. Whenever a zero is entered, increment the value in the array at index 0. Whenever a one is entered, increment the value in the array at index 1, and so on, up to index 5 of the array.

   Output the histogram count at the end. For example, if the input grades are 3, 0, 1, 3, 3, 5, 5, 4, 5, 4, then the program should output

   ```
   1 grade(s) of 0
   1 grade(s) of 1
   0 grade(s) of 2
   3 grade(s) of 3
   2 grade(s) of 4
   3 grade(s) of 5
   ```

   File must be called: grades.cpp

4. (4) Write a write a simple **function** that acts like a random number generator with the following behavior:
   - its definition looks like the following: **void rand_int(const int *min, const int *max, int *val)**
   - it accepts the addresses of three int values (one for the bottom of the random range, one for the top of the random range, and one to store the final value), generates a number within that range, and stores it in the variable pointed to by the third parameter.

   In main you should ask the user for the min and max values and store them in variables, pass the addresses of those variables (and that of a third variable) to the function, then print out the (new) value of the third variable.

   File must be called: randFun2.cpp

   Discussion Idea: The C language didn't have references. Why do you think they were added in C++, if you could already pass by reference using pointers?

5. (4) Write a function that takes as parameters three arrays-of-int (that are the same length) and the size of the arrays. It should add the first two arrays element by element, storing each sum in the corresponding element of the third array. All array accesses should be done with **pointer arithmetic**. In main you will need to query the user for the size of the arrays and the values in the first two arrays, pass them to the function, and then print out the third array. For example, if the first two arrays are [3,4,1,4,6] and [5,1,7,8,3], then the third array will be [8,5,8,12,9].

   File must be called: arrayAdd.cpp

6. **Be sure to test that each of your sections of code work correctly in a variety of cases.**
   After you think your code is good, trade executables with another person and try to break their code (share with the other person what you find works well and what does not).

(40) Project component

Write a program that asks the user for a **C-style** string and then allows the user to enter one of four commands. “rev” should reverse the string. “Lx”, where x is some integer, should shift all the characters in the string left x spaces, with any characters that fall off the beginning getting added on the end. For example, L3 would turn “hello world” into “lo worldhel”. “Rx”, where x is some integer, should shift all the characters in the string right x spaces, with any characters that fall off the end of the string getting added on the beginning. “quit” should end the program. Until the user enters “quit”, the program should loop, allowing the user to continue entering commands to manipulate the string and printing out the results of each command. You should implement the first three commands in functions (be careful of the null terminator). If the user enters a command string of the wrong format, you should print an error message and let them try again. For example:

```
Please enter a string.
perpendicular
Please enter a command.
L5
New string: ndicularperpe
Please enter a command.
rev
New string: eprepralucidn
Please enter a command.
F6
Invalid command.
Your string is: eprepralucidn
Please enter a command.
R21
New string: ralucidneprep
Please enter a command.
rev
New string: perpendicular
quit
```

File must be called: wordShift.cpp

Remember to submit your **report** and **source files** to TEACH before the end of Sunday.

Remember to keep discussion going!

Submission Checklist:

Exercises:

- evenOdd.cpp
- stringMix.cpp
- grades.cpp
- randFun2.cpp
- arrayAdd.cpp

Project:

- wordShift.cpp

Report, ***in PDF format***, must address these four sections (30 pts):

- Understanding
- Design
- Testing
- Reflection

The implementation part of the assignment is the .cpp file you submit.