CS3520 Programming in C++

Assignment 1

Objectives:

- Understand how to use variables and arithmetic operations
- Work with string and cmath libraries
- Understand how to use loops and conditional statements
- Practice debugging and number conversions
- Work on clarifying requirements, then design before coding.

Submission:

Create a folder named "hw1" in your github repository and submit all of your source code (*.cpp file) on GitHub, along with Makefile **AND** then submit the github commit link on Canvas under Assignment 1. Please make sure that your code is readable.

Note: *You are welcome to use any environment/IDE to write/test your code. Once you are ready to submit the work, you will need to first transfer it (using ftp or WinSCP) to your Khoury Linux account, compile and test it in that environment {with Makefile} before committing it to your github repository. See course policies for details (given on Canvas -> Syllabus)*

Grading scheme: 68 pts

- Source Code: Q1 (15pts =3x5) + Q2 (20pts) + Q3 (20pts) + Q4 (11pts=4+3+4) => 66 pts
- Code Readability / Style (Comments, Indentations, etc.) = Q2+Q3 (1pt+1pt) = 2 pts

Coding Assignments:

1) On course repo, we have uploaded a new folder named "exercises/hw1_debug/" which includes **3 programs** (file*.cpp) which have several bugs. Please help us fix all of the errors (including any logical errors) to ensure that the programs work as intended. *For each file, provide a list of most important bugs that were found/fixed (as comments)*

2) **Random paragraph generator:**
   A string variable often consists of a few characters, each with an ascii value. ASCII table uses values from 65-90 and 97-122 to represent English alphabets and it also contains a whole range of symbols as well as numbers between ascii values 33-96.

   Let's start by writing a program that uses a random number generator to create 2000 ascii values (*for the sake of simplicity, let's focus on ascii values for Space to ~ i.e., [32-126] only).* Insert (or pack) them in a string variable and then *print* it. You will notice that it really doesn't look anywhere close to a paragraph from your college essay. Let's see if we can try to fix it. But first:
   - Let's find out how many alphanumeric characters (i.e., number or letter) do we have? How many spaces do you have? How many numbers do we have? *Print* these statistics neatly.

   Now,

- Write code that would replace all non-alphanumeric characters with a space but leave the period (ascii value 46), comma (ascii value 44), quotation marks (ascii codes 34 and 39) and different brackets untouched. *Print* the updated version.
- Now *remove* any numeric digits from your string and *print* the new version (notice that your string will now be shorter as many of the numeric values have been eliminated and it will only have letters, spaces, quotations, different brackets, periods and commas).
- Now replace every 20th space by a LF marker (ascii value 10) and *print* this version.
- Finally insert a space before and after every bracket (i.e., (), {}, []) and *print* the final version.
- Now **read** the final version yourself and manually count any valid words.

Run your program **3 times** and report {*in comments*} the number of valid words your program generated each time.

- Now suggest {*in comments*} something that can help generate better valid words. Write code to test if your suggestion helps or not.

3) **Hexadecimal to Octal Convertor**:
During our lecture this week, we saw how each hexadecimal digit can be converted into binary by writing 4-digit binary equivalent number. Similarly, an octal number can be written by combining 3 binary digits. Write a program that mimics that manual conversion operation.

Your program will first convert a user provided (*via console*) hexadecimal number into its binary equivalent and then use the binary version to write the equivalent octal number. This hexadecimal input will be a string and must have 0x preceding the hex digits and must have minimum of 6 hex digits. It will generate (and print) a new string in the following format: "Your number XXXXXX (in hex) is equivalent to YYYYYYYYYYYYYYYYYY (in binary) and is equivalent to ZZZZZZZZ (in octal)" where XXXXXX is the positive hex integer value and YYYYYYYYYYYYYYYYYY is the binary equivalent number and ZZZZZZZZ is the octal equivalent number.
For example: for user-provided input 0x00000A, the result is: ***"Your number 0x00000A (in hex) is equivalent to 1010 (in binary) and 12 (in octal)"***

Note: You may not use libraries that allow string to integer conversions. Instead use member functions of string and cmath libraries with loops. *It might be useful for you to first write a simple algorithm or pseudo code detailing the steps that your program will follow and then write your program.*

4) Please show all work. **If you do not show your work, you will not receive any credit even if your final answer is correct**.
   a) Assume that your computer uses 10-bits registers to represent numbers. If your computer can, what would be its 2's complement representation for -11, 56, -512, 512
   b) Convert the hexadecimal number 5C1 to binary, octal and decimal representations.
   c) Given two numbers in different formats (one is hex number 1B05 and second is a decimal number 8040), what would be the best way to compute the sum and product to get the results in hexadecimal format. Discuss your approach first then perform the calculations and show your work. Notice that you can probably convert them to any format and then perform the computation in that format. Choose the approach that is the easiest and doesn't require much effort.

Note: *For Q4, you are welcome to complete your work by hand or type it and then submit it as docx/pdf/text file or screen grab on Canvas or push the file(s) to your repo.*

Change Log:

- 9/18/2022: Initial Version