**CETN 3010** **PROGRAMMING IN C**

**ASSIGNMENT #2 – FIVE NUMBER FUN**

**Due: Tuesday, Oct 13th, 2014 @ 10:30AM (ie. Before class begins)**
**Value: 15% of course mark**

# General Requirements

### REQ-001 RECREATE THE INCLUDED CONSOLE PROGRAM

Using the C programming language and Visual Studio, reverse engineer and re-create your own version of the accompanying program *FiveNumberFun.exe*. Your program should resemble and mimic exactly the functionality that is included in the sample application.

### REQ-002 NAMING YOUR PROJECT

You must name your Visual Studio with the following naming convention.

*[Your Name]*_CETN3010_Assignment2

### REQ-003 CODE REQUIREMENTS

Your recreation of this application must include the following techniques discussed in class:

1. Input and Output using printf() and scanf() functions.
2. Implementation of looping techniques discussed in class.
3. Appropriate use of arithmetic operators where appropriate.
4. One example of one *switch* statement and where appropriate other *if* statements.
5. Use of an array to store entered numbers.
6. Use of functions that receive an array as a parameter and either print to the console or return a numeric value.
   (Hints: many of these examples are in your book. You need to make them work in this application.
   The following link http://www.programmingsimplified.com/c/source-code/c-program-convert-decimal-to-binary shows an example of taking a number and printing it out as a 32-bit binary number.)
   Here are the suggested names of the functions:
   a. GetAverage – returns a decimal number indicating the average of the numbers in the array.
   b. Sort – sorts the numbers in the array from smallest to largest.
   c. PrintNumbers – prints the numbers in the array to the console.
   d. GetSum – returns an integer indicating the sum of the numbers in the array.
   e. PrintSumAsBinary – prints the 32-bit binary representation of the number to the console.

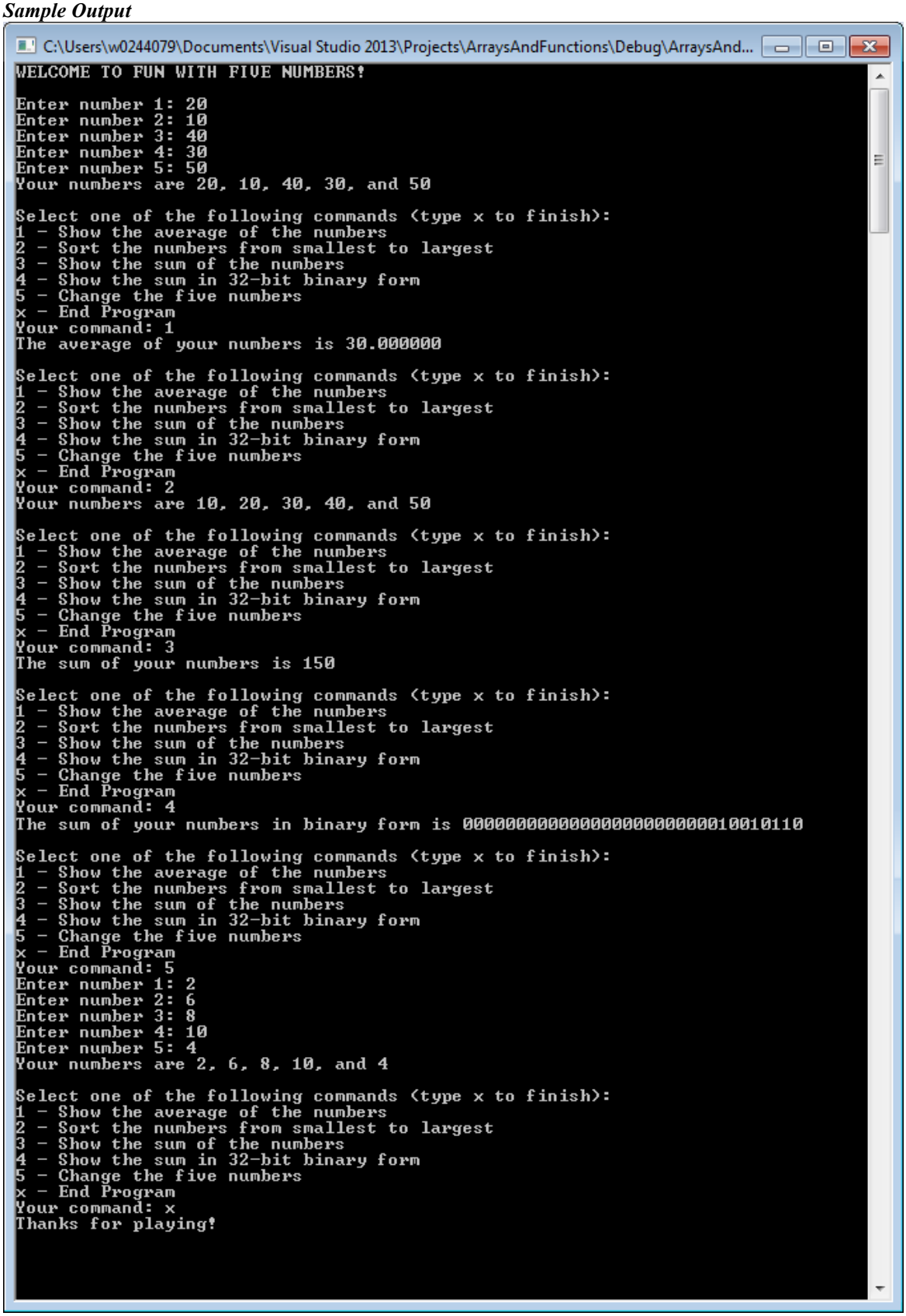
## Submission

You will submit this particular assignment via Github. Your submission will be time-stamped by Github as the latest commit you did to save your code. It is suggested that you commit often and confirm your final commit with your instructor.

## Instructions

1. **Don't forget that a code review is part of this project. You will need to show your code to the instructor in class on the due date while going through an evaluation of the application's functionality. You will need to explain how the code works and complete the code review part of the rubric.**
2. ***Late submissions will be subject to the late penalties laid out in the course outline.***

*Sample Output*



| | Developing | Competent | Professional | Mark Awarded |
|---|---|---|---|---|
| **Console Input and Output** | Developing (0 pts.)<br>Little or no elements of this requirement are met. | Competent (1 pts.)<br>Some elements of this requirement are met. Some errors exist. | Professional (2 pts.)<br>All elements of this requirement are met. | |
| **Array Implementation** | Developing (0-1 pts.)<br>Little or no elements of this requirement are met. | Competent (1-2 pts.)<br>Some elements of this requirement are met. Some errors exist. | Professional (3 pts.)<br>Numbers are stored in a correctly defined array the first time the user is prompted to enter numbers. | |
| **GetSum Function** | Developing (0-1)<br>Little or no elements of this requirement are met. | Competent (2-3)<br>Function is defined but does not accept an array and an item number count as parameters. Errors in the implementation exist. | Professional (4-5 pts.)<br>All elements of this requirement are met. Output is correctly displayed on the console screen. | |
| **GetAverage Function** | Developing (0-1)<br>Little or no elements of this requirement are met. | Competent (2-3)<br>Function is defined but does not accept an array and an item number count as parameters. Errors in the implementation exist. | Professional (4-5 pts.)<br>All elements of this requirement are met. Output is correctly displayed on the console screen. | |
| **Sort Function** | Developing (0-1)<br>Little or no elements of this requirement are met. | Competent (2-3)<br>Function is defined but does not accept an array and an item number count as parameters. Errors in the implementation exist. | Professional (4-5 pts.)<br>All elements of this requirement are met. Output is correctly displayed on the console screen. | |
| **PrintNumbers Function** | Developing (0-1)<br>Little or no elements of this requirement are met. | Competent (2-3)<br>Function is defined but does not accept an array and an item number count as parameters. Errors in the implementation exist. | Professional (4-5 pts.)<br>All elements of this requirement are met. Output is correctly displayed on the console screen. | |
| **PrintSumAsBinary Function** | Developing (0-1)<br>Little or no elements of this requirement are met. | Competent (2-3)<br>Function is defined but does not accept an array and an item number count as parameters. Errors in the implementation exist. | Professional (4-5 pts.)<br>All elements of this requirement are met. Output is correctly displayed on the console screen. Function makes use of the implemented GetSum function. | |
| **Changing of numbers in the array** | Developing (0-1) | Competent (2-3)<br>Numbers can be changed but errors in the implementation exist. | Professional (4-5 pts.) | |

| | | | | |
|---|---|---|---|---|
| | Little or no elements of this requirement are met. Numbers cannot be changed. | | All elements of this requirement are met. Output is correctly displayed on the console screen. | |
| **Looping Structures** | Developing (0-1 pts.)<br>Little or no elements of this requirement are met. | Competent (2-3 pts.)<br>An implementation of a looping structure exists but is not the best choice for the functioning of the application. Some functions lack loops where it would be appropriate. Much code duplication exists. | Professional (4-5 pts.)<br>All elements of this requirement are met. Appropriate looping structure is used. Users can do as many commands as they wish before terminating the program. No unnecessary code duplication. | |
| **Conditional Statements** | Developing (0 pts.)<br>Little or no elements of this requirement are met. | Competent (1-2 pts.)<br>No switch statement is implemented in the program. | Professional (3 pts.)<br>A switch statement is implemented in the program. | |
| **Handle invalid commands** | Developing (0 pts.)<br>Little or no elements of this requirement are met. | Competent (1 pts.)<br>Program handles invalid command entries most of the time. | Professional (2 pts.)<br>Program consistently handles invalid command entries. | |
| **Code Commenting** | Developing (0 pts.)<br>Little or no elements of this requirement are met. | Competent (1 pts.)<br>Some elements of this requirement are met. | Professional (2 pts.)<br>All elements of this requirement are met. Code is adequately commented with the programmer's intention. | |
| **Formatting/ Readability** | Developing (0 pts.)<br>Little attention to correct spacing, indenting and adequate use of curly braces. Code very difficult to read | Competent (1-2 pts.)<br>Some attention to correct spacing, indenting and adequate use of curly braces but could be improved. Code slightly difficult to read | Professional (3 pts.)<br>Code is well spaced and attention is paid to the overall layout of the code. Spacing, indentation and correct use of curly braces are all implemented in a professional manner. | |
| **Student Number:** | | **Student Name:** | | Total:<br>( /50) |