

Name: _____

Date of demonstration: _____

Marks earned: /5

LAB 4 (Due by 10 PM on October 06)

Maximum Points: 5, Weight: 5%

Pre-requisites: Readings assigned thus far have been completed. Notes taken during lectures have been reviewed.

To complete the lab, please follow the instructions below. Not following these instructions may result in deduction of marks.

1. To receive any credit, you need to:
 - a. Demonstrate your work during the week 6 lab; Please print this handout, write your name and date of demonstration, and bring the printout with you for demonstration;
 - b. Upload your work to eConestoga as instructed in step 2 by the deadline.
2. Once you complete this lab, add weekly status report to the Visual Studio solution folder. Compress the entire solution folder to create a file with “.zip” extension. Upload the “zip” file to the appropriate assignment folder on eConestoga. You will not receive any credit if you fail to upload this file, even if you have finished and demonstrated your work. No credit is awarded without demonstration.
3. At the top of each C file, add your name and date of program creation.
4. Any variables or functions you create must be named following “camelCase” notation. Variables must be initialized before use. In case of multiple variables, define only one variable per line.
5. Program the following in one C file with one main function:
 - a. [3 points] For this part, your program prompts the user for four integers. The program finds the smallest integer and prints it on the console. Your program also prints whether the smallest integer is a multiple of 3. To achieve the above mentioned functionality, your program must use two C functions with the following prototypes:
 - i. **int smallest(int num1, int num2, int num3, int num4);** This function returns the smallest of **num1, num2, num3, and num4**;
 - ii. **int isMultipleOf3(int number);** This function returns 1, if **number** is a multiple of 3. Otherwise, it returns 0.
 - b. [2 points] For this part, your program prompts the user for two different integers. The program prints the average value of all the odd integers between the two values entered by the user. An odd integer is not divisible by 2. Average is calculated by dividing the sum of odd integers by the total number of odd integers. To achieve the above mentioned functionality, your program must use a function with the following prototype:
 - i. **double oddAverage(int a, int b);** This function returns the **average** of all odd integers between **a** and **b**.

Please be prepared to answer any questions during the demo. You are expected to correctly understand your programs. Any lack of understanding may result in deduction of points.