**CPSC 3125 - Spring 2020**

**Project 3**

**Using the system call fork()**

**Due: 11:30 PM Thursday Feb. 20**

**This is an individual project**. **Write a C program using the system call fork() according to the following instructions:**

1. Your program should totally create **exactly 3 new processes**, each of which performs a different task.
2. The root parent process prints both process IDs of its own and its parent, and generates one random integer in the range of 200~2000 by using the function rand(). Then print **on the screen** all the positive multiples of 33 that are no more than the generated random number.
3. The first newly created process prints both process IDs of its own and its parent, **create an array of integers** and initialize it to {10, 20, 30, 40, 50, 60, 74, 89, 63}. Then calculates the average (stored in a float variable) of all the elements in the array, and outputs the average **on the screen**.
4. The second newly created process prints both process IDs of its own and its parent, and generates 10 random integers in the range of 1000~2000 by using the function rand(). Then **write these 10 random integers into an output file** named “data.txt” (created by your program). This output file MUST be included in your project submission.
5. The third newly created process prints its process ID, and **reads the integers from an input file** “input.txt”. The integers in the file are separated by a space. Then calculates the average (stored in a double variable) of all the integers in the file, and outputs the average and the number of integers on the screen (do NOT print all the integers to the screen in your final output).

A programming example of reading data from a file and writing data to a file using C is attached. Your program should be saved as “**Project3\_lw.c**”, where lw is your initials. Make the main() function be as simple as possible, and the code executed by each process **MUST** be in a separate function. Write one function at a time, test it and make sure everything in this function is correct before you work on the next function.

**How to Submit?**

**You must take screenshots for the output of a sample run with appropriate inputs** **for your program**. Save the screenshots as image or Word files. Your ZIP file should contain three items: (1) all the screenshots, (2) the output file “data.txt” generated at Step 4, and (3) the C source file.

**Grading Criteria**

1. **Correct output from the root parent process (20%)**
2. **Correct output from the first child process (20%)**
3. **Correct output from the second child process (20%)**
4. **Correct output from the third child process (20%)**
5. **Screenshots of the outputs (20%)**