ECE 175 Homework Assignment 6

Due Date: by 11.59 pm on October 19, 2021

Submission Instructions: Submit only .c flies in the designated Assignment Dropbox on D2L.
Conventions: Name your C programs as *hw**x**p**y**.c*
where ***x*** corresponds to the homework number and ***y*** corresponds to the problem number.
For example, the C program for homework 6, problem 1 should be named as *hw**6**p**1**.c*
Write comments in your programs. Programs with no comments will receive PARTIAL credits.
For each program, the following info should at least be included at the top of the C file:
Author and Date created and Brief description of the program.

Problem 1 (35 points): Luhn Algorithm
https://www.geeksforgeeks.org/luhn-algorithm/

The *Luhn Algorithm** is a formula that is used to determine whether the identification number provided by a user is valid. The formula is widely used in validating credit card numbers, as well as other number sequences such as government Social Security Numbers (SSNs), and mobile device manufacturer identification numbers.

Given an *N* digit identification number, the *Luhn Algorithm* is applied as follows:
1. S_{odd} =Sum all digits in the odd positions (start counting from the right).
2. The digits in the even positions are multiplied by 2, then added in the following way: S_{even} =
- if the digit multiplied by 2 is < 10 then it is added, as is, to the value of S_{even}.
- if the digit multiplied by 2 is ≥ 10 then the sum of the digits is added to the value of S_{even}. For example, if an even positioned digit is 7 then 2 · 7 = 14. The digits of 14 sum to 1 + 4 = 5 so add 5 to S_{even}.

3. $Sum_{Total} = S_{odd} + S_{even}$
4. if Sum_{Total} %10 = 0 then the identification number is valid.

For example, a security code of 3 5 6 7 4 4 9 8, contains the number 8 in the first position. The integer positions should be counted from right to left.
• S_{odd} = 5 + 7 + 4 + 8 = 24. Since 5 7 4 and 8 are in the odd positions of the security code.
• The values in the even positions are 3 6 4 9. Multiply each digit by two to get 6 12 4 18. Since 12 and 18 are greater than 9 then they become 1 + 2 = 3 and 1 + 8 = 9. So S_{even} = 6 + 3 + 8 + 9 = 26
• $Sum_{Total} = S_{even} + S_{odd}$ = 26 + 24 = 50.
• 50%10 = 0 so 3 5 6 7 4 4 9 8 forms a valid security code.

The file *SecurityCodes.txt* contains a list of 20 digit security codes, write a C program that
• Reads the digits from the *SecurityCodes.txt* file into an array (one row at a time)
• Determines whether the security code in the array is valid according to the *Luhn Algorithm*.
• If it is valid, display the code and 'valid'. If t is not valid, list the *first digit* that makes the entire code valid. See sample code execution.

Your program shall include at least the following user defined functions:

```
// Reads a line of 20 security code values ,
        int ReadSecurityCode ( FILE *I , int SC []);

// Tests to see if an array of 20 values forms a valid security code.
        int IsValidSecurityCode ( int SC []);
```

Sample code execution:

Given that the first four lines of SecurityCodes.txt are

```
8 0 4 7 3 9 0 1 3 8 1 2 7 9 7 1 9 2 5 2
2 9 1 5 5 5 8 2 1 7 9 9 5 6 4 8 3 2 0 6
7 7 9 8 1 7 2 6 7 4 7 7 4 6 1 4 6 7 1 0
5 6 7 3 5 0 4 9 8 5 8 4 0 4 6 9 6 5 0 0
```

The output of your program should be

```
8 0 4 7 3 9 0 1 3 8 1 2 7 9 7 1 9 2 5 2          valid
2 9 1 5 5 5 8 2 1 7 9 9 5 6 4 8 3 2 0 6          invalid,
        The digit in the first position for valid code should be 7
7 7 9 8 1 7 2 6 7 4 7 7 4 6 1 4 6 7 1 0          invalid,
        The digit in the first position for valid code should be 9
5 6 7 3 5 0 4 9 8 5 8 4 0 4 6 9 6 5 0 0          valid
```

Check if 8 0 4 7 3 9 0 1 3 8 1 2 7 9 7 1 9 2 5 2 is a valid security code:

1. S_odd = 0 + 7 + 9 + 1 + 8 + 2 + 9 + 1 + 2 + 2 = 41
2. S_even = $(2 \cdot 8) + (2 \cdot 4) + (2 \cdot 3) + (2 \cdot 0) + (2 \cdot 3) + (2 \cdot 1) + (2 \cdot 7) + (2 \cdot 7) + (2 \cdot 9) + (2 \cdot 5)$
   $= (16) + (8) + (6) + (0) + (6) + (2) + (14) + (14) + (18) + (10)$
   $= (1 + 6) + (8) + (6) + (0) + (6) + (2) + (1 + 4) + (1 + 4) + (1 + 8) + (1 + 0)$
   $= 7 + 8 + 6 + 0 + 6 + 2 + 5 + 5 + 9 + 1 = 49$
3. S_Total = S_odd + S_even = 41 + 49 = 90
4. 90%10 = 0. Thus 8 0 4 7 3 9 0 1 3 8 1 2 7 9 7 1 9 2 5 2 is a valid security code.

Problem 2 (35 points) Develop a C program that processes a file called "data1.txt" and "data2.txt" containing a sequence of real-valued numbers. Your program should calculate and display

- Mean of data from data1.txt
- Mean of data from data2.txt
- Standard Deviation of data from data1.txt
- Standard Deviation of data from data2.txt
- Correlation between the data contained in those two files.

You must use the following user-defined functions in your code.
Note: length is the number of data points in each of the files.

```
void read_file(FILE *inp, double z[], int length);
//read_file - read data from the text file pointed to by inp and keep all data points in the array z

void calc_mean(int length, double z[], double *mean);
//calc_mean - find mean of data points in the array z with number of elements = length

void calc_std(int length, double z[], double *std);
//calc_std - find standard deviation of data points in the array z with number of elements = length
```

If x and y are arrays containing the sequence of number that were read from data1.txt and data2.txt, respectively, then

$$corr(x,y) = \frac{\sum_{i=0}^{499}(x_i - m_X)(y_i - m_y)}{500\, s_y s_x}$$

where

$$s_y = \sqrt{\left(\frac{1}{500}\sum_{i=0}^{499} y_i^2\right) - \left(\frac{1}{500}\sum_{i=0}^{499} y_i\right)^2} \qquad s_x = \sqrt{\left(\frac{1}{500}\sum_{i=0}^{499} x_i^2\right) - \left(\frac{1}{500}\sum_{i=0}^{499} x_i\right)^2}$$

$$m_y = \frac{1}{500}\sum_{i=0}^{499} y_i \qquad m_x = \frac{1}{500}\sum_{i=0}^{499} x_i$$

Output (code execution):

```
Mean of data in data1.txt is 2.03439916
Mean of data in data2.txt is 3.01058324
Standard deviation of data in data1.txt is 0.97672088
Standard deviation of data in data2.txt is 1.00189554
The correlation is 0.25396397
```

Lab 6 (30 points): complete this assignment when you attend the lab session after HW 6 is due.