ENGR 15100: Software Tools for Engineers
Spring 2015

Computer Assignment #10

Due: Tuesday, April 21, 2015, 9:00am CST

Departments of Engineering
School of Engineering, Mathematics, & Sciences
Purdue University Calumet

PURDUE
UNIVERSITY
CALUMET

1. OBJECTIVE

Continue working with iterative and conditional statements and begin becoming familiar with creating and using user-defined functions.

2. PROCEDURE

Task I : Counting Digits in an Integer Number [50 points]

In a script file named `LASTNAME_LAB10_TASK1.m` write a program according to specifications outlined below. Before starting, clear all variables in the MATLAB Workspace and the contents of the MATLAB Command Window. **Unless specified, suppress all MATLAB Command Window output**. After completing the program, the result of executing your program should look similar to the Command Window output shown below.

```
>> LASTNAME_LAB10_Task1
Enter an integer: 122333444455555
Enter a digit number to find (0 to 9): 3
Integer 122333444455555 contains 15 digits.
Occurrence frequency of digit 3 within 122333444455555 is 3.

>> LASTNAME_LAB10_Task1
Enter an integer: -122333444455555
Enter a digit number to find (0 to 9): 3
Integer 122333444455555 contains 15 digits.
Occurrence frequency of digit 4 within -122333444455555 is 4.

>> LASTNAME_LAB10_Task1
Enter an integer: 14.5
Invalid input.  Try again!
Enter an integer: 122333444455555
Enter a digit number to find (0 to 9): 20
Invalid input.  Try again!
Enter a digit number to find (0 to 9): -3
Invalid input.  Try again!
Enter a digit number to find (0 to 9): 5
Integer 122333444455555 contains 15 digits.
Occurrence frequency of digit 5 within 122333444455555 is 5.

>> LASTNAME_LAB10_Task1
Enter an integer: 0
Enter a digit number to find (0 to 9): 1
Integer 0 contains 1 digits.
Occurrence frequency of digit 1 within 0 is 0.

>> LASTNAME_LAB10_Task1
Enter an integer: 0
Enter a digit number to find (0 to 9): 0
Integer 0 contains 1 digits.
Occurrence frequency of digit 0 within 0 is 1.
```

(a) **[12 points]** Prompt the user to enter a scalar integer and store the integer in a variable named `integerInput`. *Assume the user will always enter a numerical scalar that does not contain any leading zeros*. Utilize a **`while-end`** statement to continuously prompt the user until the user actually enters an integer. You may use the built-in **`round()`** function to help you determine whether or not the user entered an integer scalar.

(b) **[10 points]** Prompt the user to enter a scalar integer representing a decimal digit (i.e. a positive integer number in the range 0 to 9, inclusive) and store the scalar integer in a variable named `digitToFind`. *Assume the user will always enter a numerical scalar*. Utilize a **`while-end`** statement to continuously prompt the user until the user actually enters a decimal digit in the range 0 to 9, inclusive.

(c) **[23 points]** Utilize a **`while-end`** statement in addition to other statements to perform the following:

- Compute the number of individual digits contained within the integer stored in `integerInput`. Store the result in a variable named `numOfDigits`.
- Compute the number of times (i.e. the frequency) at which the `digitToFind` appears within the integer entered by the user. Store the result in a variable called `digitToFindFreq`.

Hint: In each iteration, *compute the least significant digit (LSD) of the integer* `integerInput`. *Then, reduce the value of* `integerInput` *by a factor of 10 and keep the integer part of the reduced value.* You may use built-in function **`floor()`** to help you. *Continue to compute LSD's and reduce the value of* `integerInput` until all the individual digits have been computed.

(d) **[5 points]** Using multiple instances of the built-in **`fprintf()`** function, display `numOfDigits` and `digitToFindFreq`. Format each variable as an integer.

Thoroughly test your program for **integers containing less than or equal to 18 digits**. Use the sample output shown above to guide you in the testing process.

Task II: Creating/Using User Defined Functions – A Mathematical Function [50 points]

(a) **[20 points]** In a function file named `myMathFunction.m`, write a user-defined function named **`myMathFunction()`** that evaluates the single-variable mathematical function $f(x) = 5\,sin(x)cos(2x + \frac{\pi}{3})$. User defined function **`myMathFunction()`** accepts as an input parameter an array of x values named `xIn` and generates an output parameter named `fOut` representing the result of $f(x)$ evaluated at $x =$ `xIn`. Note that user-defined function **`myMathFunction()`** should operate on a scalar input parameter `xIn` or an array input parameter `xIn`. When input parameter `xIn` is an array, **`myMathFunction()`** should operate according to the principle of element-wise arithmetic like most built-in MATLAB mathematical functions.

(b) **[30 points]** In a script file named `LASTNAME_LAB10_TASK2b.m`, write a program that performs the following steps. Before starting, clear all MATLAB Workspace variables and clear the MATLAB Command Window contents.

i. **[15 points]** Calculate $f(x)$ one at a time for the following x values: $[-\frac{\pi}{3},\ -\frac{\pi}{2},\ 0,\ \frac{\pi}{6},\ \frac{\pi}{4}]$. Display each corresponding value of $f(x)$ in the MATLAB Command Window using several instances of the built-in **`fprintf()`** function. Utilize the user-defined function **`myMathFunction()`** in combination with a **`for-end`** statement with a loop variable named `xVal` that will represent each value of x.

ii. **[15 points]** In a Figure Window named Figure 3, plot the function $f(x)$ for 1000 equally-spaced values of x in the range $-2\pi \leq x \leq 2\pi$. Use a green dotted line for the curve. Utilize the user-defined function **`myMathFunction()`**.

Task III: Assignment Submission

Upload to Blackboard Learn the following MATLAB files:

(a) `LASTNAME_LAB10_TASK1.m`
(b) `myMathFunction.m`
(c) `LASTNAME_LAB10_TASK2b.m`