

ENGR 15100: SOFTWARE TOOLS FOR ENGINEERS
SPRING 2015

COMPUTER ASSIGNMENT #7

Due: Tuesday, March 31, 2015, 9am CST

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



1. OBJECTIVE

Continue working with conditional statements and become familiar with **for-end** statements.

2. PROCEDURE

Task I: Fun with Months of the Year [50 points]

In a script file named `LASTNAME_LAB7.m`, write a MATLAB program that determines the name of a calendar month given an integer representing the month's number. The program should carry out the general steps outlined below and utilize **if-end** statements. Unless otherwise specified, suppress output to the Command Window.

- (a) **[2 points]** At the beginning of your script, make sure to clear all variables defined in the MATLAB Workspace as well the contents of the MATLAB Command Window.
- (b) **[4 points]** Prompt the user to enter an integer that represents one of the twelve months in the standard calendar year (1 = January, 12 = December). *Assume the user always enters an integer.*
- (c) **[5 points]** In case the user enters an invalid month number, report back to the user that he/she must re-run the script because an invalid month number was entered. The program should terminate at this point and no "calculations" should be performed!
- (d) **[39 points]** For valid month numbers, display to the MATLAB Command Window the name of the month corresponding to the integer entered by the user.

Test your script for all possible valid calendar month numbers and at least two invalid calendar month numbers. Some sample executions of the program are shown below.

```
>> LASTNAME_LAB7_TASK1
Enter an integer corresponding to a calendar month: 3
Calendar month 3 corresponds to March.

>> LASTNAME_LAB7_TASK1
Enter an integer corresponding to a calendar month: 7
Calendar month 7 corresponds to July.

>> LASTNAME_LAB7_TASK1
Enter an integer corresponding to a calendar month: -2
Month #-2 is an invalid month!
Please re-run the script to try again.
```

Task II: Analyzing an Array with for-end Statements [50 points]

In a script file named `LASTNAME_LAB7_TASK2.m`, write a program that performs the following steps.
Unless otherwise specified, suppress output to the Command Window.

- (a) [1 point] At the beginning of your script, make sure to clear all variables defined in the MATLAB Workspace as well the contents of the MATLAB Command Window.
- (b) [4 points] Create a variable named `V` and assign to it a 250-element row vector whose elements are each randomly generated real numbers chosen uniformly from the open interval $(-4.5, 5.5)$.
- (c) [10 points] Create variables `numNegative`, `numPositive`, `sumPositive`, `meanPositive`, and `prodRange`. Initialize each variable to an appropriate scalar value.
 - `numNegative`, `numPositive`: number of negative and positive elements in `V`, respectively
 - `sumPositive`, `meanPositive`: sum of all the positive elements in `V`, respectively.
 - `prodRange`: the product of all real numbers in vector `V` in the range `[2.3, 2.6]`
- (d) [7 points] Declare a **for-end** statement using a loop variable named `k` that will be assigned to every element of a row vector whose elements represent the indices/positions of row vector `V`.
- (e) [20 points] The body of the **for-end** statement should update, when applicable, the values of `numNegative`, `numPositive`, `sumPositive`, and `prodRange`. *The use of built in functions `sum()`, `mean()`, and `prod()` is not allowed.*
- (f) [3 points] After the **for-end** statement, compute the mean of the positive real numbers in vector `V` and assign the mean to a variable named `meanPositive`.
- (g) [5 points] Using multiple instances of `fprintf()`, display the values contained in variables `numNegative`, `numPositive`, `sumPositive`, `meanPositive`, and `prodRange`, respectively.
 - Format variables `numNegative` and `numPositive` as integers.
 - Format variables `sumPositive`, `meanPositive` and `prodRange` as fixed-point real numbers, each showing a maximum of 3 digits after the decimal point.

Upon completing the above steps, the result of executing your program should look similar to the sample output shown below.

```
>> LASTNAME_LAB7_TASK2
Vector V has 119 negative elements.
Vector V has 131 positive elements.
The sum of vector V's positive elements is 363.659.
The mean of vector V's positive elements is 2.776.
The product of the elements in the range [2.3, 2.6] is 229.376.
```

Task III

Upload to Blackboard Learn the following MATLAB script files.

(a) `LASTNAME_LAB7_TASK1.m`

(b) `LASTNAME_LAB7_TASK2.m`