ENGR 15100: Software Tools for Engineers
Spring 2015

Computer Assignment #9

Due: Tuesday, April 14, 2015, 9am CST

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

PURDUE
UNIVERSITY
CALUMET

1. OBJECTIVE

Continue working with iterative statements and become familiar with **while-end** loops.

2. PROCEDURE

Task I

[40 points] In a script file named LASTNAME_LAB9_TASK1.m, write a program according to the 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_LAB9_TASK1
Enter a value for x (in radians): 2
Enter a threshold in the range (0, 1): -1
Invalid threshold. Please try again!
Enter a threshold value in the range (0, 1): 0
Invalid threshold. Please try again!
Enter a threshold value in the range (0, 1): 1.3
Invalid threshold. Please try again!
Enter a threshold value in the range (0, 1): 1e-5
cos(2.000000) = -0.4161468365
cosApprox(2.000000) = -0.4161552028
Number of terms = 6

>> LASTNAME_LAB9_TASK1
Enter a value for x (in radians): pi/2
Enter a threshold in the range (0, 1): 1e-4
cos(1.570796) = 0.0000000000
cosApprox(1.570796) = 0.0000247373
Number of terms = 5

>> LASTNAME_LAB9_TASK1
Enter a value for x (in radians): 0.0
Enter a threshold in the range (0, 1): 1e-7
cos(0.000000) = 1.0000000000
cosApprox(0.000000) = 1.0000000000
Number of terms = 1
```

(a) [4 points] Create a scalar variable named x and assign to it a value obtained by prompting the user with the prompt string ‘Enter a value for x (in radians): ’.

(b) [10 points] Create a scalar variable named threshold and assign to it a value obtained by prompting the user with the prompt string 'Enter a threshold in the range (0, 1): '. Use a **while-end** statement to continuously prompt the user until the user enters a valid value for the threshold. *You may assume the user will always enter a numerical value.*

(c) **[25 points]** Using another **`while-end`** statement to perform the following:

- Compute an approximate value for $\cos(x)$ using the summation formula given below.

$$cosApprox(x) \approx +\frac{x^0}{0!} - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots$$

  For the value of `x` entered by the user, accumulate terms of the formula for $cosApprox(x)$ until the absolute value of the difference between the summation of $cosApprox(x)$ and the actual value of $\cos(x)$ is less than or equal to the `threshold`.

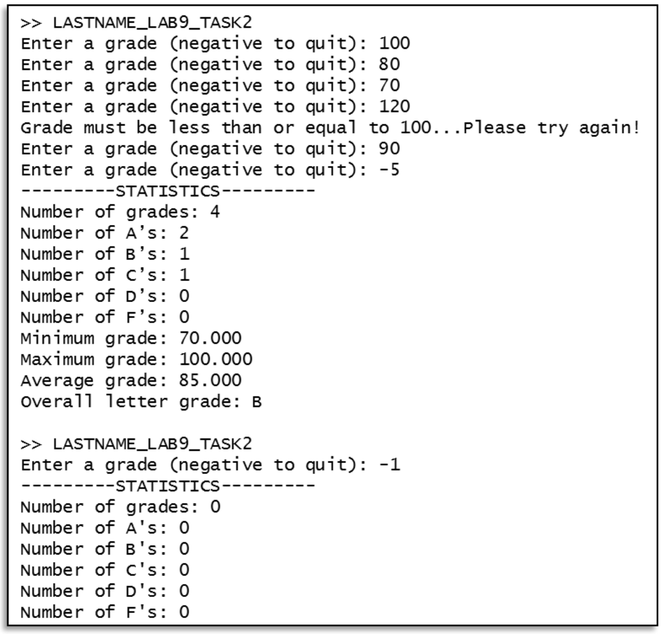
- Compute how many terms were accumulated to achieve the accuracy as determined by the `threshold` value entered by the user.

(d) **[6 points]** Use multiple instances of the built-in **`fprintf()`** function to display to the Command Window the value of $cos(x)$, the value of $cosApprox(x)$, and the number of terms used for the accumulation. Format the values of $cos(x)$ and $cosApprox(x)$ as fixed-point real numbers showing a maximum of 10 digits after the decimal point. Format the number of accumulated terms as an integer.

Thoroughly test your program with more than those input sequences shown in the sample output above.

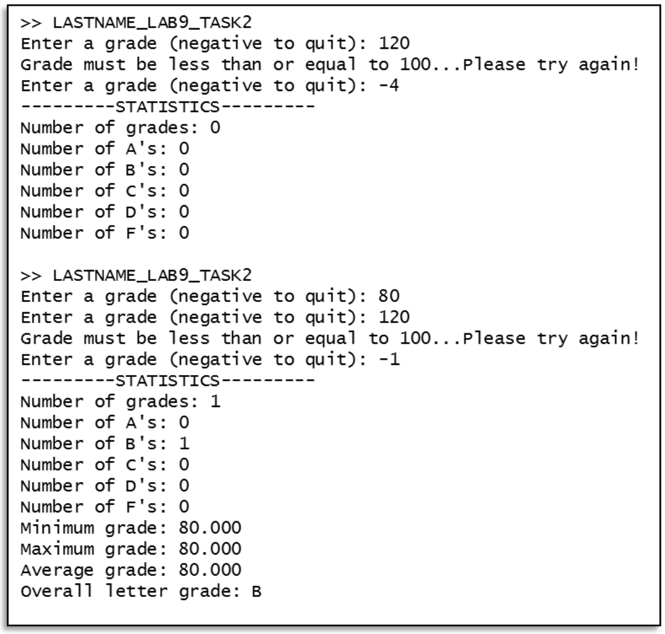
Task II

[60 points] In a script file named `LASTNAME_LAB9_TASK2.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_LAB9_TASK2
Enter a grade (negative to quit): 100
Enter a grade (negative to quit): 80
Enter a grade (negative to quit): 70
Enter a grade (negative to quit): 120
Grade must be less than or equal to 100...Please try again!
Enter a grade (negative to quit): 90
Enter a grade (negative to quit): -5
---------STATISTICS---------
Number of grades: 4
Number of A’s: 2
Number of B’s: 1
Number of C’s: 1
Number of D’s: 0
Number of F’s: 0
Minimum grade: 70.000
Maximum grade: 100.000
Average grade: 85.000
Overall letter grade: B

>> LASTNAME_LAB9_TASK2
Enter a grade (negative to quit): -1
---------STATISTICS---------
Number of grades: 0
Number of A's: 0
Number of B's: 0
Number of C's: 0
Number of D's: 0
Number of F's: 0
```



```
>> LASTNAME_LAB9_TASK2
Enter a grade (negative to quit): 120
Grade must be less than or equal to 100...Please try again!
Enter a grade (negative to quit): -4
---------STATISTICS---------
Number of grades: 0
Number of A's: 0
Number of B's: 0
Number of C's: 0
Number of D's: 0
Number of F's: 0

>> LASTNAME_LAB9_TASK2
Enter a grade (negative to quit): 80
Enter a grade (negative to quit): 120
Grade must be less than or equal to 100...Please try again!
Enter a grade (negative to quit): -1
---------STATISTICS---------
Number of grades: 1
Number of A's: 0
Number of B's: 1
Number of C's: 0
Number of D's: 0
Number of F's: 0
Minimum grade: 80.000
Maximum grade: 80.000
Average grade: 80.000
Overall letter grade: B
```

(a) **[40 points]** Using **`if-end`** and **`while-end`** statements, perform the following:

- Prompt the user to enter numerical grades, one grade at a time (i.e. do not use vectors). Any number of grades can be entered, however, each grade must be non-negative. A numerical grade larger than 100 is considered invalid and the user should be prompted to re-enter the grade. *A negative numerical grade indicates the user has finished entering numerical grades*.
- Compute the number of grades in each grade range using the standard grading policy: A [90 – 100], B [80 – 90), C [70 – 80), D [60 – 70), F [0 – 60).
- Compute the minimum grade, maximum grade, and average grade. You are not allowed to use built-in functions **`min()`**, **`max()`**, **`sum()`**, or **`mean()`**.

(b) **[20 points]** Using multiple instances of the built-in **`fprintf()`** function, display the following items in the Command Window:

- the number of valid grades entered by the user formatted as an integer (always display)
- the number of valid grades belonging to each grade range, each formatted as an integer (always display)
- the minimum grade, maximum grade, and average grade, each as a fixed-point real number showing a maximum of 3 digits beyond the decimal point (display when at least one valid grade is entered)
- the overall letter grade according on the average formatted as a character (display when at least one valid grade is entered).

Thoroughly test your program with more than those input sequences shown in the sample output above.

Task III

Upload the following MATLAB script files to Blackboard Learn.

(a) `LASTNAME_LAB9_TASK1.m` (b) `LASTNAME_LAB9_TASK2.m`