ENGR 160 Algorithms and Applications Sec 02 Spring 2018

Homework Assignment #6 Due 9:10am Feb 16 (F)

Read Sections 3.7-3.8 and Chapter 5 of the MATLAB book. Work through the examples and practice problems given in the book. You do not have to submit any of this work.

Problem 6.1

Write a function ***prodby2*** that will receive a value of a positive integer n and will calculate and return the product of the odd integers from 1 to n (or from 1 to n-1 if n is even). Use a ***for*** loop. Test your function by calling it using the following values of n: 1, 2, 3, 9, 10, respectively. Attach your results.

Problem 6.2

The inverse of the mathematical constant e can be approximated as

$\frac{1}{e} \approx \left(1 - \frac{1}{n}\right)^n$, where n is a positive integer.

Create a MATLAB function named ***approx_inv_e*** to find the minimum positive integer n needed for the approximated value to have a specified accuracy. The function should have one input argument as the maximum tolerable difference between the real value of 1/e and its approximation, and the output argument is the minimum value of n required for achieving the required accuracy. Your function should still print the approximated value and the actual value, but not the number n, which is returned in the output argument. Test your function using 0.1, 0.01 and 0.001 as the input argument, respectively, and attach your results

Problem 6.3

Create a MATLAB function named ***find_max_N*** to find the largest integer N that satisfies the following inequality, where M is the upper bound. The function should take one input argument – the upper bound value, and output one single output argument - the largest N value.

$$2^2 - \frac{3}{1} + 4^2 - \frac{5}{3} + ... + (2N)^2 - \frac{2N+1}{2N-1} \le M$$

Use the following examples: M=0; 1; 10; 100 and 1000 to test your function. Attach the results.

Problem 6.4 (20 points)

A magic matrix is a square matrix (with N rows and N columns) constructed from the integers 1 through N^2, and the sums along each row, each column, and the two diagonal lines are all equal. For example, $\begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$ is a 3-by-3 magic matrix. The sum of each row, column, and diagonal line is 15. Write a MATLAB function called ***ismagic*** that accepts a square matrix as input and determine if this matrix is magic. Its output is '0' (logic false) if the matrix is not a magic matrix; and '1' (logic true) if it is a magic matrix. Test your function using $\begin{bmatrix} 16 & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{bmatrix}$ and then a randomly generated 4-by-4 integer matrix as input. Attach your results.

Note that not only do you need to check if the sums along rows, columns, and diagonal lines are equal, but also need to check if all the elements are from 1 to N^2.