Homework 3

Create a single MATLAB script with each problem below as a separate section (hint: %%).

1. Create a new script
2. Create a comment section at the top with your name, date, HW#, class, etc.
3. The first script commands should erase all the workspace data, command window output, and close all figures.
4. Create separate sections (%%) for each problem
5. For problem 1-4, provide an input, call the function and then publish the result.

Problem 1 (20 pts)

The Fibonacci numbers are generated by the sequence:

$$1, 1, 2, 3, 5, 8, 13, \ldots$$

Can you work out what the next term is? Write a recursive function `f(n)` to compute the Fibonacci numbers F_0 to F_{20}, using the relationship:

$$F_n = F_{n-1} + F_{n-2},$$

given that $F_0 = F_1 = 1$.

Problem 2 (20 pts)

The first three Legendre polynomials are $P_0(x) = 1$, $P_1(x) = x$, and $P_2(x) = (3x^2 - 1)/2$. There is a general *recurrence* formula for Legendre polynomials, by which they are defined recursively:

$$(n+1)P_{n+1}(x) - (2n+1)xP_n(x) + nP_{n-1}(x) = 0.$$

Define a recursive function `p(n,x)` to generate Legendre polynomials, given the form of P_0 and P_1. Use your function to compute `p(2,x)` for a few values of `x`, and compare your results with those using the analytic form of $P_2(x)$ given above.

Problem 3 (20 pts)

The lump sum S to be paid when interest on a loan is compounded annually is given by $S = P(1 + i)^n$, where P is the principal invested, i is the interest rate, and n is the number of years. Write a program that will plot the amount S as it increases through the years from 1 to n. The main script will call a function to prompt the user for the number of years (and error-check to make sure that the user enters a positive integer). The script will then call a function that will plot S for years 1 through n. It will use 0.05 for the interest rate and $10,000 for P.

Problem 4 (20 pts)

Write your own MATLAB function to compute the exponential function directly from the Taylor series:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots.$$

The series should end when the last term is less than 10^{-6}. Test your function against the built-in function `exp`, but be careful not to make x too large—this could cause a rounding error.

Problem 5 (20 pts)

Hurricanes are categorized based on the winds. The following table shows the category number for hurricanes with varying wind ranges and what the storm surge is (in feet above normal).

1	74–95	4–5
2	96–110	6–8
3	111–130	9–12
4	131–155	13–18
5	>155	>18

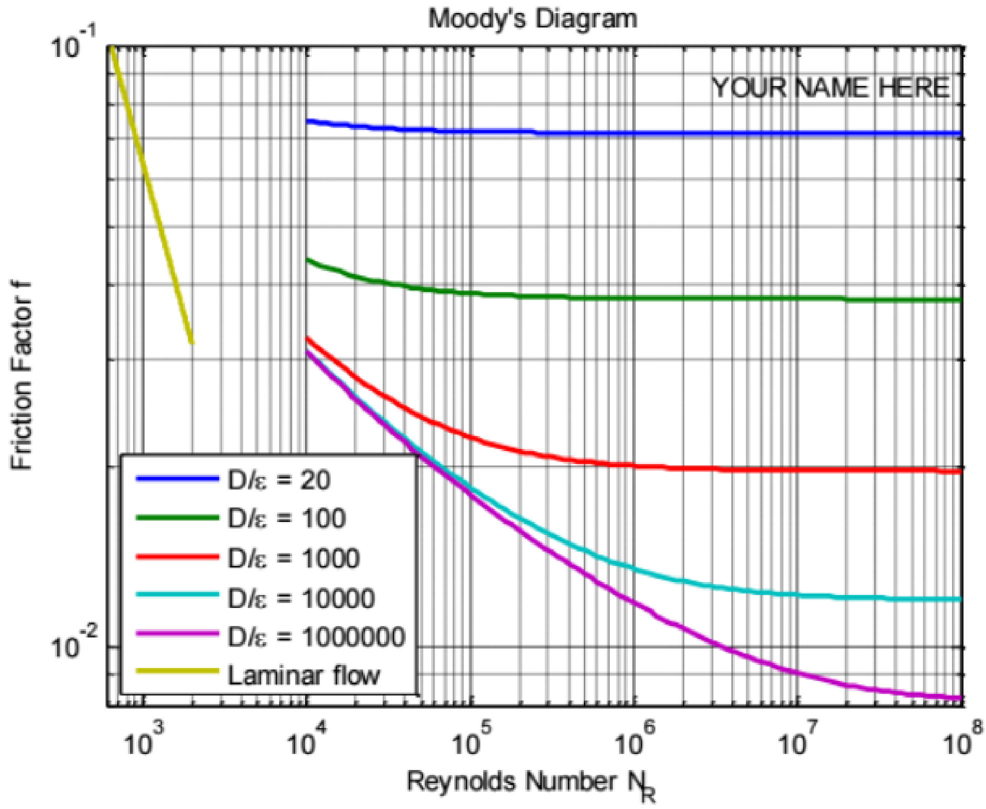
Write a function that will receive as an input argument the wind speed, and will return the category number and the minimum value of the typical storm surge.

Bonus question (20 pts)

Moody's diagram as shown on the next page is a famous plot used to determine the effect internal friction (surface roughness) has on fluids flowing in pipes. The equation below has been developed by Jain and Swamee for the friction factor (f) for turbulent pipe flow.

$$f = \frac{0.25}{\left[\log\left(\frac{1}{3.7(D/\varepsilon)} + \frac{5.74}{N_R^{\;0.9}}\right)\right]^2}$$

a. Plot the friction factor following function for values Reynolds number (N_R) between 10^4 and 10^8 for D/ε values of 20, 100, 1000, 10,000, and 100,000. Your plot should look similar to example below. A couple of hints:
   a. Notice that both axes are log scaled
   b. log( ) in MATLAB does not do what you think it does
   c. The logspace( ) function may be a helpful in generating data that is evenly spaced on log-scaled axes (similar to linspace( )).
b. Add a line for $f = 64/N_R$ for smooth pipes to the same plot (be sure to match the N_R range shown in the figure).
c. Add a complete title and x and y axis labels (including any of the subscripts or superscripts).
d. Add a legend with the actual Greek character epsilon for each trace (e.g. D/ε = 1000, etc.). It can be in any position on the figure.
e. Adjust the axis limits so it looks like the figure on the last page.
f. Use the "text" command to print your name anywhere on the plot.



Turn in the following:

1. A Word (.doc, .docx) document created using the MATLAB publish feature to publish your script.
2. Your .m script file(s) (these are separate files from the document above). Make sure you use plenty of comments. Before submitting, rename all *.m files to have a .txt extension. For example, rename MyHW6.m to MyHW6.txt before submitting.

Submit all files electronically on Blackboard. See syllabus for late assignment policy.

Late submissions will receive a 10% deduction!
No submissions will be accepted after one day!