MATH.2720 Introduction to Programming with MATLAB
Homework on Programming Structures (Due 4/5)

Please email me a script file containing the commands you used to answer these questions.

1. Write a script file that asks the user to input three real numbers $a$, $b$, and $c$ and then calculates and displays the real root(s), if any, of the quadratic $ax^2 + bx + c$ . If there are 2 distinct real roots, the output should be "There are two distinct real roots:" followed by the values of the roots. If there is a single real root, the output should be "There is one real root:" followed by the value of the root. If there are no real roots, the output should be "There are no real roots." Test your code on the quadratics $x^2 + 2x + 5$, $x^2 - 5x + 6$, and $x^2 - 6x + 9$.

2. Write a script file that asks the user for a positive integer $n$ greater than 2 and then generates and prints an array of length $n$ containing the first $n$ Fibonacci numbers. The Fibonacci numbers are the numbers $0, 1, 1, 2, 3, 5, 8, 13, \ldots$. Each Fibonacci number after the second is the sum of the previous two.

3. (From Gilat, *MATLAB: An Introduction with Applications*)

   The Sierpinski triangle can be implemented in MATLAB by plotting points iteratively according to one of the following three rules which are selected randomly with equal probability.

   Rule 1: $x_{n+1} = 0.5x_n,\ y_{n+1} = 0.5y_n$

   Rule 2: $x_{n+1} = 0.5x_n + 0.25,\ y_{n+1} = 0.5y_n + \dfrac{\sqrt{3}}{4}$

   Rule 3: $x_{n+1} = 0.5x_n + 0.5,\ y_{n+1} = 0.5y_n$

   Write a script file that calculates the $x$ and $y$ vectors and then plots $y$ versus $x$ as individual points. (Use `plot(x,y,'^')`) Start with $x_1 = 0$ and $y_1 = 0$. Run the program four times, first with 10 iterations, then 100 iterations, then 1,000 iterations, and finally with 10,000 iterations. (Hints: The command `rule = randi([1 3])` will generate a 1, 2, or 3 with equal probability. You might want to use a `switch - case` structure inside a `for` loop.)

4. Approximate the value of the sum $\displaystyle\sum_{n=1}^{\infty} \frac{-1^{n+1}}{n}$ by computing a partial sum $\displaystyle\sum_{n=1}^{N} \frac{-1^{n+1}}{n}$. Use a `while` loop that terminates when the difference between two successive approximations is less than $10^{-5}$ (1.e-5 in MATLAB notation).

   Compare the value you obtain with the number $\ln(2)$.