MATLAB Week 2: 1D Arrays Practice

Submit one .m file for this activity on Canvas. Use an appropriate commented header and separate the problems by sections (%%). Your file name should be *"inclass2_lastname."*

All problems (both Tuesday and Thursday) due on Mon. 1/25 by 11:59 pm

Tuesday Problems (due on Mon. 1/25 by 11:59 pm)

Example: Given the data below, write a script that can analyze the temperatures in Breckenridge and find days that are above 0^{o}C (above freezing). We'll use the following Fahrenheit temperatures from a five-day period:

Mon: 19.43, Tue: 34.37, Wed: 41.2, Thu: 30.78, Fri: 25.95

First, we'll define a day of the week string array, enter the given temperatures into an array, and convert those temperatures to Celsius. Next, we'll use the max, min, and mean functions to find the highest, lowest, and mean temperatures and print the results using fprintf. Finally, we'll find the temperatures from this week that are greater than 0^{o}C using a logical array operation and use fprintf to display days of the week corresponding to the 1D array index values for days above 0^{o}C (days above freezing).

--

Problem 1: Write a script that asks the user to enter a list of 5 separate numbers (using the input function, *i.e. list(1) = input('Enter the first number: ');*) and then calculates the mean, min, and max of the list of numbers using the mean, min, and max functions. The script should print out the result of the calculation on the screen to two decimal places using fprintf.

Problem 2: Write a script which prompts the user to enter a list of three separate numbers (using the input function, *i.e. list(1) = input('Enter the first number: ');*) and a fourth number which is a "minimum value" for comparison. The code should print to the screen any of the first three numbers in the list that are greater than this fourth "minimum value."

Problem 3: A class has students with the following ID numbers:

104 156 126 178 101

The respective course scores for these students are:

93 86 75 96 89

Write a script that determines which students got an A in the course (a score greater than or equal to 90 – use a logical array operation like in the example). The script should print the ID numbers of the students that got an A to the screen using logical array indexing.

(Thursday problems below)

Thursday Problems (due on Mon. 1/25 by 11:59 pm)

Example: Plot the polynomial y=x^2-10x+15 with x starting at 0 going to 10 in increments of 0.1 with red circles. Then find the minimum y value and corresponding x value and print both to the user (hint: type ‘help min’ to get the syntax for the min function to give the min value and corresponding array index). Find the x values where y is less than 0 and store these in an array, then plot the negative y values corresponding to these x values with blue circle on the same plot (hint: use ‘hold on’).

--

Problem 1:

Left: A center pivot field viewed from above Right: A close-up of the irrigation machinery

Many farmers use center-pivot irrigation systems which rotate around a circle of crops to easily water their crops. Center-pivot irrigation systems are typically up to 100 m long (the circle radius). The typical corn plant requires about 0.1 m^2 of area to grow properly for maximum yield, and each corn plant yields about 1.5 ears of corn on average. Calculate the total value of the field for each radius from 0 to 100 m if there are 50 ears of corn in a bushel and the current value of a bushel is $3.8575/bushel. Plot the total value of the crop (vertical axis) with a center-pivot irrigation system with a radius (horizontal axis) from 0 to 100 m (question: will this plot be linear?). Make sure to include an appropriate title and axis labels. Use fprintf to display how many 100 m radius fields it would take to raise $1 million worth of corn. *Hint*: use the ceil function to round up to the nearest integer.

Problem 2: In some cultures, people celebrate by shooting a bullet straight up into the air, which can be a dangerous practice. If the muzzle velocity of the projectile is 715 m/s, write a script to tell the user how high the bullet travels, when the bullet reaches max height, and when the bullet will return back to the ground (take cover!). First, create a time vector t from 0 to 200 s in increments of 0.1 s. Second, use the projectile motion equation y = v_0*t - ½*g*t^2 to calculate the project heights over this time interval (let g = 9.81 m/s). Third, plot the height of the projectile over time just to see what the trajectory looks like. Fourth, find the positive height values greater than 0 and plot the data so only the positive height values are included in the plot (the projectile can’t go through the ground), and add a title and axis labels. Fifth, find the maximum height and corresponding time when the bullet reaches the maximum height, then print both values to the user (i.e. ‘The projectile will reach a max height of y(t) m at t s’). Finally, find the time when the projectile will hit the ground and print this value to the user.

Extra Credit Exercises

The following two problems are optional extra credit problems. If you plan to submit work for these problems, you should submit a single .m file to be turned under the separate extra credit assignment for week 2.

Extra Credit Exercise 1:

Extra Credit Exercise: Download the Matlab data file fibonacci.mat from Canvas and use the load function to load the data it into your Matlab workspace. This file contains the first 1000 terms of the Fibonacci sequence. Find the first term of the Fibbonacci sequence where the value of that term is greater than 1E100. Display the term and the number to the user, and verify that you found the correct number.

--

Extra Credit Exercise 2:

The amount, A, which will be in a bank account that earns compound interest, is calculated with the following formula:

$$A = P\left(1 + \frac{r}{n}\right)^{nt}$$

where P is the principle (the amount originally deposited in the account), r is the annual interest rate (as a decimal, i.e. 3%=.03), n is the number of times the interest is compounded (how often interest earnings are paid into the account), and t is the number of years that the account earns interest.

Simple interest will accumulate lower interest earnings over time because the interest is not compounded one or more times per year (i.e. you’re not earning interest on accumulated prior interest). The equation for simple interest is:

$$A = P(1 + rt)$$

Write a script which prompts the user for a principle dollar amount P and the interest rate r (i.e. 0.03 for 3%). The script should calculate the amount that will be in the savings account from 1 to 30 years in increments of 1 year assuming that the interest is compounded monthly, or n = 12 (compounded 12 times per year, or monthly). Note that since t will be a vector of years from 1 to 30, you’ll need to use the .^ element-wise operator in the equation above to calculate the amount saved A for each year t (element-wise 1D array operation). Plot the amount saved A versus the year t using a solid red line. Next, add a dotted blue curve to the same plot using the simple interest equation for comparison. Calculate and print out how much your savings has multiplied relatively to the principle after 30 years in each case for comparison.