QUIZ 2 – MATLAB

Directions:

1. This is a limited open book quiz, meaning that the ***only*** sources of information allowed are *lecture notes, tutorial problems, textbook and your handwritten notes* reiterating/reinforcing the material presented in class. This information can be accessed in soft or hard copy form.
2. Please note that the access of files, documents, or information ***other than*** that explicitly listed at Point 1 will be treated as cheating and will be dealt with accordingly.
3. The access of social media, text messages, email or any other form of electronic communication will be treated as cheating and will be dealt with accordingly.
4. "Teamwork" and/or discussions are not permitted during the quiz.
5. The use of cell phones during the quiz is prohibited.
6. Only your owl submission will be marked.
7. **Quiz duration: 90 minutes (9:30 am – 11:00 am) + 10 minutes upload time**
8. **Total Duration: 100 minutes (9:30 am – 11:10 am)**

Submission Directions:

9. Code all your solutions in MATLAB. Save each problem as a separate '.m' file.
10. Ensure your upload to OWL is successful and that all files are functional. If there is a problem with your upload you must let the professor or TA know **BEFORE** leaving the zoom room. After leaving the quiz missing or corrupt files will not be accommodated for.

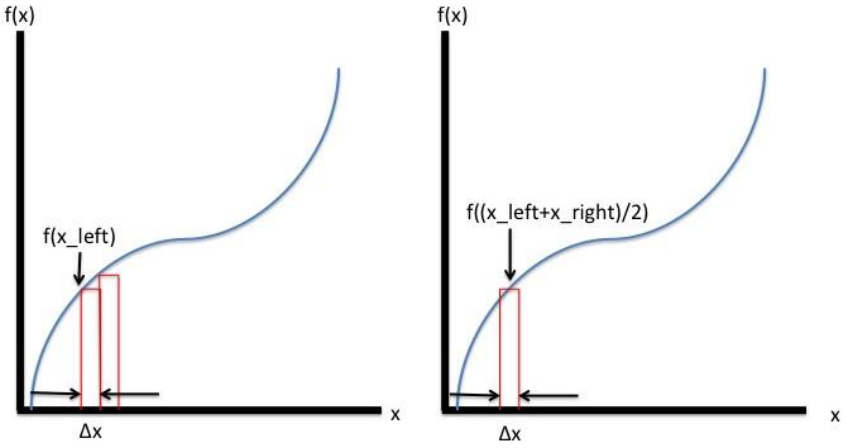
Note: While performing problems you can use the help function built into MATLAB to help you with syntax, and the proper use of commands.

Problem 1 [10 marks]

*** This is the same problem you solved in Quiz 1, Question 1 ***

Integrals can be estimated numerically by using various discretization methods. Two of these methods are shown below. The left method calculates the area under the curve by multiplying the function value at the left data point by the discretization width, all these areas are then summed. The right method is the same except it uses the function value at the average x value of the section.

$$\int f(x) \approx \sum f(x_{left}) * \Delta x \qquad \int f(x) \approx \sum f((x_{left} + x_{right})/2) * \Delta x$$



Given the function:

$$f(x) = -x^4 + 30x^2 + 25x - 100$$

Where $-6 \leq x \leq 6$

a) Write a MATLAB script that calculates the function values using a discretization with of Δx = 0.1 and write your results to a 1xN array named '`fx1`'. **[3 marks]**

b) Using the left numerical integration method described above, compute the definite integral within the bounds given using a discretization width of Δx = 0.1 and write the resulting value to a variable called '`left_int`'. Remember to not use the last value (at x=6) since the resulting area would be to the right of the interval of interest. **[5 marks]**

c) Define an anonymous function called '`fx2`' using the function provided above. **[1 mark]**

d) Plot the function using the '`fplot`' function from $-6 \leq x \leq 6$. **[1 mark]**

Save your script as '`Q1.m`', which you will upload to OWL during submission.

Problem 2 [10 marks]

The code below has a user guess a secret number that has been passed into the function and gives them 5 tries to do so. It has been provided in `Q2_GIVEN.m`, feel free to try running it.

```
%Quiz 2, Problem 2 (Provided Code)

secret_num(5);  % calling the function, passing in the num2guess

function secret_num(num2guess)
% this function gives the user 5 tries to guess the number passed in
% num2guess = number to guess

fprintf('You have 5 attempts to guess the secret number between 1 and 10\n');

for num = 1:1:5  % loop from 1 to 5 (for each try)

    fprintf('Attempt # %d\n',num);  % telling user the attempt #
    guess = input('Guess :');  % prompting the use for a guess

    if guess == num2guess  % checking if guess is correct
    disp('You guessed the secret number!');  % telling the user they are correct
        return  % 'return' from the function as we have finished
    end

    if num == 5  % checking to see if the user if out of attempts
        disp('You failed')  % telling the user they are out of tries
        return  % 'return' from the function as we have finished
    end

end
end
```

You have been asked to modify the code to accomplish the following:

a) Change the function so that number to guess between 0 and 10 is randomly generated inside the function, so that the user does not need to pass in the number to be guessed. To do this, you can use `randi([0 10])` which will generate a random integer between 0 and 10. Be sure to use a '`;`' to make sure this is not visible to the user! **(3 marks)**

b) Modify the function so that the user can control the range of the random integer generated so that the function call is now `function secret_num(min,max)` where min and max represent the lowest and highest possible values of the random integer. **(4 marks)**

c) Now, add the ability to pass in the number of tries given, so that the function call is now `function secret_num(min,max,tries)` **(2 marks)**

d) Finally, if the user fails to guess the correct number in the maximum number of guesses, modify the function to tell the user the secret number. **(1 mark)**

Save your modified file as '`Q2.m`', which you will upload to OWL during submission.

Problem 3 [10 marks]

You are an avid day trader and have managed to clear $100,000 from the recent short squeeze that occurred on GameStop (NYSE: GME). You have since decided it is a smart idea to invest in something less risky like Ontario real estate (which some argue is currently at a plateau), and plan to use the investing windfall as a down payment on an $800,000 house. The table below lists the *Simplified Year 1 Cost of Borrowing* for different loan amounts (rows) and interest rates (columns).

Simplified Year 1 Cost of Borrowing

		Interest Rate (%)								
		1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
Loan Amount	500000	5000	7500	10000	12500	15000	17500	20000	22500	25000
	550000	5500	8250	11000	13750	16500	19250	22000	24750	27500
	600000	6000	9000	12000	15000	18000	21000	24000	27000	30000
	650000	6500	9750	13000	16250	19500	22750	26000	29250	32500
	700000	7000	10500	14000	17500	21000	24500	28000	31500	35000
	750000	7500	11250	15000	18750	22500	26250	30000	33750	37500
	800000	8000	12000	16000	20000	24000	28000	32000	36000	40000
	850000	8500	12750	17000	21250	25500	29750	34000	38250	42500
	900000	9000	13500	18000	22500	27000	31500	36000	40500	45000

The values in the red outlined section of the table can be calculated using the following formula:

$$Simplified\ Year\ 1\ Cost\ of\ Borrowing = Loan\ Amount \times \left(\frac{Interest\ Rate\ (\%)}{100}\right)$$

a) Recreate the portion of the table above outlined in red as a matrix called '`costs`' in a MATLAB script. You do not need to worry about the greyed-out sections. Arrays for loan amount and interest have been provided in '`Q3_GIVEN.m`' **(5 marks)**

Note: You cannot simply 'hard code' the values in the table, they must be calculated mathematically using the given loan amount and interest arrays. One way to do this would be to utilize nested for loops which index the two arrays for the calculation to be performed.

b) Your friends are now also shopping for real estate and would like to use a similar analysis. Turn your code into a function which permits the user to pass in any arrays containing loan amounts and interest rates, which the function then uses to generate a table as you did in part (a). The format of the function call is given below. **(5 marks)**

```
function [costs] = LoanCostCalculator(loan_amount,interest)
```

Note: You can test your function using the given arrays, but it should be able to handle other sizes of arrays which are passed in.

Save the modified file as '`Q3.m`' for upload to OWL during submission. If you made your function standalone (not embedded in the above script), then be sure to upload it as well.