Lab Assignment #1
Due 1/31/2020 at 11:59 pm on bCourses

For this assignment and future assignments, you will be tasked with creating a set of function files (.m) in Matlab (Matlab language syntax about function could be found here). Each of these functions will take a set of inputs and produce a set of outputs as specified by the function declaration table provided at the beginning of each problem statement. These functions will be graded with an autograder which checks the outputs for given inputs. For your functions to be scored properly, it is important that the names of your functions **exactly match** the names specified in the problem statements, and input and output variables to each function are in the correct order (i.e. use the given function header exactly). For this first assignment, templates are provided for each function on bCourses to help you follow the correct format. Instructions for submitting your assignment are included at the end of this document.

The due time is 11:59pm January 31st, 2020. NO LATE HOMEWORK WILL BE ACCEPTED. Instructions for submit homework in bcourse can be found here.

1 Using MATLAB as a Calculator

`function [d1,d2,d3,d4,d5,d6,d7,d8,d9,d10] = myMathCalculations(a, b, c, x)`		
Input	Type	Description
`a`	`1x1 double`	Variable a
`b`	`1x1 double`	Variable b
`c`	`1x1 double`	Variable c
`x`	`1x1 double`	Variable x
Output		
`d1,d2, ... d10`	`1x1 double`	Solution to the equations shown below.

Details

For this problem, you will add your code to the file `myMathCalculations.m` provided on bcourses. For each of the quantities d_1 through d_{10}, construct a **one-line** MATLAB expression that computes the value and assigns it to a variable. Your Matlab statements should give appropriate values for d_1 through d_{10} regardless of the values of a, b, c, and x (i.e. you should use the variable names instead of numerical values). Definitions of the variables `d1` through `d10` have been started for you in the template. You can check your code with a hand-held calculator and using the tester script (see Submission Guidelines).

1. $\mathrm{d1} = (x - \frac{1}{ab})^{-2}$

2. $\mathrm{d2} = c^{ab}$

3. $\mathrm{d3} = \sqrt[a]{c}$

4. $\mathrm{d4} = \log_{10}(a + b)$

5. d5 $= \frac{a^2+1}{(bx-1)(x+c^b)}$
6. d6 $= \ln(|\cos(a+b)|)$
7. d7 $= \log_2|\sin(\frac{a}{b})|$
8. d8 $= \cos(\frac{a^2+b^2}{\sqrt{(a^2+1)}})$
9. d9 $= \cos(\sin(\frac{x}{2}+1))$
10. d10 $= \cos(a)\sin(b) - \cos(b)\sin(a)$

Tips

- There is a file, `myMathCalculations.m`, on bCourses to help you get started on this problem.
- There is a MatlabCommands.pdf file in bcourse offers many examples of how to code different mathematical expressions, you may find it quite helpful for this problem!
- These trigonometric functions accept the argument in radians.

2 The Mean

`function [ave, abs_ave] = myMean(x1, x2, x3)`		
Input	Type	Description
`x1`	`1x1 double`	A quantity designated as $x1$
`x2`	`1x1 double`	A quantity designated as $x2$
`x3`	`1x1 double`	A quantity designated as $x3$
Output		
`ave`	`1x1 double`	The mean of $x1$, $x2$, and $x3$
`abs_ave`	`1x1 double`	The mean of $\|x1\|$, $\|x2\|$, and $\|x3\|$

Details

For this problem, you will code a function that finds the mean and the mean absolute value of the three quantities: $x1$, $x2$, and $x3$. You might find the following equations helpful.

$$Mean = \frac{1}{N}\sum_{i=1}^{N} xi$$

- In this problem, N should be 3.

- There is a file, `myMean.m`, on bCourses to help you get started on this problem.
- Note that the mean absolute value of a set of quantities is defined as the mean of the absolute values of those quantities. In other words, the order that the operations are applied is the absolute value first and then the mean second.
- There is a function `mean` that is provided in Matlab. However, do not use the built-in function.

3 The Variance

`function [sigma2, abs_sigma2] = myVar(x1, x2, x3)`		
Input	Type	Description
`x1`	`1x1 double`	A quantity designated as $x1$
`x2`	`1x1 double`	A quantity designated as $x2$
`x3`	`1x1 double`	A quantity designated as $x3$
Output		
`sigma2`	`1x1 double`	The variance of $x1$, $x2$, and $x3$
`abs_sigma2`	`1x1 double`	The variance of $\|x1\|$, $\|x2\|$, and $\|x3\|$

Details

For this problem, you will code a function that finds the variance and the variance absolute value of three quantities: $x1$, $x2$, and $x3$. You might find the following equations helpful.

$$Var = \frac{1}{N-1}\sum_{i=1}^{N} |xi - Mean|^2$$

- In this problem, N should be 3.
- There is a file, `myVar.m`, on bCourses to help you get started on this problem.
- Note that the variance absolute value of a set of quantities is defined as the variance of the absolute value of those quantities. In other words, the order in which the operations are applied is the absolute value first and then the variance.
- Note that you need `ave` and `abs_ave` to calculate `sigma2` and `abs_sigma2`, you can write the function `myVar` with the help of the function `myMean`. To use `myMean` in `myVar`, those two functions should be placed under the same directory (folder).
- There is a function `var` that is provided in Matlab. However, do not use the built-in function.

4 Deposit Calculator

`function Saving = myInterest(e, F)`		
Input	Type	Description
`e`	`1x1 double`	The monthly interest rate e in decimal format
`F`	`1x1 double`	The money($) available at the beginning of August
Output		
`Saving`	`1x1 double`	The account balance ($) at the end of the year

Details

A bank offers a monthly interest of e to its clients. If we have F dollars in the bank, with a monthly rate e, after i months, we will have $F \cdot (1 + e)^i$ dollars. In this case, we calculate the interest based on the balance at the end of each month. For example, if we begin August with F dollars, then at the end of the August, we will have $F \cdot (1+e)$ dollars.

A client begins August with F dollars and does the following transactions.

- He withdraws 3% at the beginning of September.
- He deposits $2,500 at the beginning of October.
- He withdraws 8% at the beginning of November.
- He deposits $5,000 at the beginning of December.

Calculate the bank value ($) at the end of the year and return this value as Saving.

Submission Guidelines

When you are ready to test one or more of your function(s), download the `Lab01Tester.m` script from bCourses and run it in the same directory as your function(s). This will run a series of test cases for your function(s) and display the outputs. You should check these results with the example outputs displayed. Note that the series of tests that the script will run is **not exhaustive and passing them does not guarantee full credit on the assignment**. It is up to you to make sure that your function fully satisfies the requirements of the problem. If you want to run all the test cases for one function, select that part of the tester script and push the `Run Section` button. If you wish to only check a single test case, you may do so by highlighting the appropriate lines in Matlab and then pressing F9 (on Windows and linux) and **`shift + fn + F7`** (on a Mac) to run the selected text in the Command Window.

The script will also pack your function into a zip file for final submission. Your function file names and headers should match the examples given in the problem statement. If they do not match, the tester script will not be able to find them and will issue a warning. If you do not correct this problem, your missing function(s) will not be able to be graded and **you will receive 0 points**, so be sure to check carefully! Make sure to upload the `Lab01.zip` file to bcourses before the deadline. You may resubmit your zip file by re-uploading to bcourses before the deadline, but please be aware that only the most recent zip submission is considered for grading. That said, it is critical that you include your most recent version of **all** of your functions in each new zip submission.

Your final zip file for this assignment should contain the following:

- myMathCalculations.m
- myMean.m
- myVar.m
- myInterest.m