The University of Nottingham

department of Mechanical, Materials and Manufacturing Engineering

A LEVEL 1 MODULE, SPRING SEMESTER 2020-2021

**MMME 1041 COMPUTER PROGRAMMING –RESIT COURSEWORK**

**NEED TO ANSWER ALL QUESTIONS**

**Introduction:**

This coursework is designed to help you revise your basic MATLAB knowledge that you have gained in autumn semester 2020-2021. For each question you should write a script that contains the answers. Save your scripts to disk with some sensible name for example, the file containing the code for question one could be called Q1.m.

In each .m file you generate, put on the first line your name and e-mail address –the markers will use this e-mail address to give you feedback. For questions which require you to write English sentences just include these as comments using the '%' character in a script file. Try to include comments in the script files describing what the program is doing.

Once you have finished all the questions, zip all your scripts up into a single zip file and the zip file must contain a word document where you can show each script as well as the solutions and graphs you have generated.

Name the zip file using the following format Firstname\_surname\_studentid.zip. So for example my zip file would be called: **yong\_ren\_01234567.zip.**

Finally, please remember this is individual work and not group work. Work handed in must be entirely your own and not copied from anyone else.

**Please submit the zip file by email to yong.ren@nottingham.edu.cn by the deadline of 4 pm, 20 August, 2021**

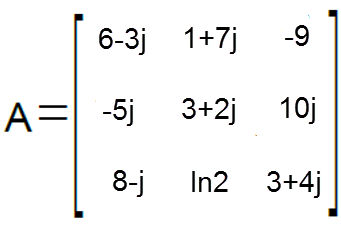
**Note that in this paper, variable-names and file-names are signified with chevrons.   
Thus, for example:**

<A> represents the variable *A*

<my\_points\_file.mat> represents the file *my\_points\_file.mat*

**SECTION 1: Command-line instructions**

For each of the following, state what MATLAB command-line you would use

1. Create a vector of numbers from -5 to 5 in steps of 1.5, and plot the results.
2. Establish the following array in the workspace: , where j is the imaginary unit, and find the non-conjugate transpose.
3. Create an array B of 3,000 random numbers between 0 and e5, then calculate the average value and standard deviation of the array B.
4. Calculate the following summation: 
5. Given that the array, <A>, exists, find out its dimension and add a row of zeros on the end of it.
6. Load the array of data from the asci file “mydata.dat” and find the average of all data in the array.

[60 marks in total – 10 per question]

**SECTION 2: Usage of MATLAB features**

For each of the following features within MATLAB, state how it might be used – using illustrations where possible.

1. save
2. fopen
3. sprintf
4. imshow
5. surf

12. switch

[60 marks in total – 10 per question]

**SECTION 3: Interpreting MATLAB code**

For the following piece of MATLAB code, provide an interpretation of the lines of code (either individually or in small groups) and then comment on the effect of the complete piece of code.

PLEASE NOTE: line numbers such as Q13a, Q13b, Q13c etc. are provided on the left hand side. These are not a part of the MATLAB code. You should use these in your response to indicate which line of code you are interpreting. You do not have to transcribe (i.e. copy-out) the lines of code.

13.

|  |  |  |
| --- | --- | --- |
| 13a | clear | %clear removes all variables from the current workspace, releasing them from system memory. |
| 13b | y=0; | %Assign y a value of 2 |
| 13c | x=1; | %Assign x a value of 1 |
| 13d | var=10; | %Assign var to 10 |
| 13e | for a=1:20 | % starts the loop, a=1,2,3,... ,20 |
| 13f | y=5\*x^3+log(x)+7\*x; | %Assign y with the value of the function 5\*x^3+log(x)+7\*x |
| 13g | dy=15\*x^2+1/x+7; | %Assign dy with the value of the function 15\*x^2+1/x+7 |
| 13h | new\_x=x-(y/dy); | %Assign new\_x with the value of the function x - (y/dy) |
| 13i | var=abs(x-new\_x); | %Assign var with the value of the function abs(x - new\_x) |
| 13j | if (var<1e-6) | %Determine if (var<1e-6) is true, if so, execute the following statement, otherwise do not execute the following statement |
| 13k | break; | %Jumping out of the loop |
| 13l | end | %End of judgement statement |
| 13m | x=new\_x; | %Assign x to new\_x |
| 13n | end | %End of loop statement |
| 13o | x | %The output window shows the value of x |

[30 marks]

14.

|  |  |  |
| --- | --- | --- |
| 14a | clear; | %clear removes all variables from the current workspace, releasing them from system memory. |
| 14b | a = input(' Enter the first integer : '); | %The command line prompts for the first integer |
| 14c | b = input(' Enter the second integer : '); | %The command line prompts for the second integer |
| 14d | c = 1; | %Assign c a value of 1 |
| 14e | while c==1 | %Loop statement, determine if c is equal to 1, if c is equal to 1, execute the following statement, otherwise, jump out of the loop |
| 14f | if (b > a) | %Determine if b is greater than a |
| 14g | b = rem(b, a); | %assign b to rem(b, a) |
| 14h | else | %If b is less than or equal to a, the following statement is executed |
| 14i | a= rem(a, b); | %assign a to rem(a, b) |
| 14j | end | %End of judgement statement |
| 14k | if (min([a, b])==0) | %Determine whether the smallest of a and b is equal to 0. If it is 0, execute the following statement |
| 14l | result = max([a, b]); | %assign result to the maximum of a and b |
| 14m | c=0; | %Assign a value of 0 to c |
| 14n | end | %End of judgement statement |
| 14o | end | %End of loop statement |
| 14p | disp([' The result is = ' int2str(result)]); | %Command line window showing the resulting values |

[30 marks]

**SECTION 4: Writing MATLAB code**

15. A sensor has been developed to monitor the level of organic pollutants present in water reservoir by measuring the concentration of the pollutants every hour. The data is saved sequentially into a text file called 'pollutant.dat', where the first column contains the time in hours since the beginning of the observation and the second column shows the value of concentration in ppm (parts per million) every hour. Note that both two columns of “pollutant.dat” contain integer values. The data file is cleared before the first observation is conducted.

(a) Write a MATLAB script to read the data file into memory as two variables called x and y, where x holds the time and y holds the concentration. Determine how long the observation has been made. Plot a graph of y versus x. Label the x- and y- axes appropriately. The y-axis should use a logarithmic scale and the x-axis a linear scale.

[30 marks]

(b) Write a function called ***alert\_signal*** which takes a single concentration value as the input argument. If the concentration value is below 10, the program should print “pollutant level very low” to the screen. If the value given to the function is between 10 and 1000, the function should write “water quality within safe limits” to the screen. If the input value to the function is above 1000, the program will print a warning message: “reaching dangerous level-abc, water treatment is required!”, where ‘abc’ here refers to the exact value of input concentration. If none of these conditions are met the code should print “value out of range”.

[30 marks]

(c) Write a function called ***weekly\_average*** which calculates the average concentration value of pollutants detected by the sensor during the first week. The function should take two arrays x and y as input, and return the result as a single number. The vectors x and y here have each got over 168 entries.

[30 marks]

(d) Photocatalytic oxidation technology is used for water treatment, as it can lead to degradation of organic contaminants when the photocatalysts are under either UV light or solar irradiation. The concentration of pollutants decays over time after the treatment is implemented. Assume the concentration decreases at a rate in accordance to the equation,

where t is time, m and n are two constants. Write a program which simulates the exponential decay of concentration using this equation. First, prompt the user for the value of the constants m and n. Then generate an array <t> representing time per hour between 1 and 120 hours, and the array contains 120 evenly distributed numbers. Use the above equation and the array <t> to calculate the corresponding values of C(t). When C(t) is below 1000, the program should write “water quality within safe limits again at t=abc hour” to the screen, where ‘abc’ here refers to the exact value of hours, and then the program is terminated.

[30 marks]

**End**