MATH 2221B Mathematics Laboratory II

Lab Assignment 9

Name: _____   Student ID.: _____

In this assignment, you are asked to run MATLAB demos to see MATLAB at work. The color version of this assignment can be found in your own **H:** drive.

Instructions

1. Start MATLAB, until you see a window with the MATLABprompt "**>>**". This window is called the **Command Window**.

2. After you started have MATLAB, you will automatically be in the directory **H:**. Please enter "`diary on`" after the MATLABprompt **>>** only once to record all your work in H:\diary. No marks will be given if no diary is found.

3. You should write your results on the lab sheet provided, and save the figures in the **H:** drive, in your personal drive.

4. Please read and sign the following declaration before handing in your assignment. Otherwise, no marks will be given.

1 (20 marks)		3 (25 marks)	
2 (25 marks)		4 (30 marks)	
		Total	/100 pts

Please read the following carefully:
General Guidelines for Lab Assignment Submission.

- Please sign and date the statement of Academic Honesty.

- Please go to the class and lab indicated by your registered course code via the CUSIS system. If you go to a different lab than the one you are registered for, you will not receive credit for the assignment even if you completed it.

- Write your COMPLETE name and student ID number legibly on the cover sheet (otherwise we will not take any responsibility for your lab). Please write your answers using a black or blue pen, NOT any other color or a pencil.

- Write your solutions on a double-sided printout of this pdf file. Try to fit your answers inside the available space.

- The use of computers/cellular phones/graphing calculators/iPads will NOT be permitted during tests and lab assignments. Please do not use our lab computer to recharge your cellar phone battery.

- In order to make it fair for all students, during the labs and tests, if you touch/press any icons on your cellular phone, our TA will check your phone to determine whether or not you are exchanging messages with another student. If you are found cheating (in the tests or in the lab or on homework assignments), you will automatically get an F grade in this course and your act will be reported to the Department for necessary disciplinary actions.

Exercises

Note: Remember to add ";" if you want to suppress the output. Marks will be deducted if there is any unnecessary output.

1. (20 marks) Write a script named `q1.m` to do the following:

   (a) Read the data in `q1.xlsx` into a matrix called `data`. In `q1.xlsx`, the first row corresponds to the values in x-axis, the second row corresponds to the values in y-axis.

   (b) Use a straight line $y = mx + b$ to fit in the data. Here

   $$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \quad m = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \quad b = \bar{y} - m\bar{x}$$

   (c) Plot the graph with the original data together with the best-fit line. The best-fit line should be in **black** color. Save it using `print('q1.eps','-depsc');`

   (d) Put the values of best-fit line into the third row of `data`. Save `data` as `q1.mat`.

   Remark: You are not allowed to use the built-in function `polyfit` to compute the best-fit line.

   Solution

```
clear all;

data = xlsread('q1.xlsx');

x = data(1,:);
y = data(2,:);

mX = sum(x)/numel(x);
mY = sum(y)/numel(y);

m = sum((x-mX).*(y-mY))/sum((x-mX).^2);
b = mY - m*mX;

figure;plot(x,y,x,m*x+b,'k-');
print('q1.eps','-depsc');

data(3,:) = m*x+b;
save q1.mat data
```

2. (20 marks) Create a MATLAB function `FormattedOutput(x,p)`. In this function, user is asked to input a real number x and precision p. If p is a positive integer, the program will round off x to p decimal place. If not, the program will ask the user to re-enter p, until a correct input is obtained. You may assume the user will always input real numbers as parameters.

Hint:

(a) Any integer is divisible by 1.

(b) The command

```
str = sprintf('Rounding off to %%2i decimal place,
x = %%%d.%df\\n', p+2, p)
```

followed by a `fprintf` may help. Note that the above command should be type in the same line. `%%` in the above command tells MATLAB that you want to print a `%`.

Sample Output of `FormattedOutput(pi,1.1)`

```
>> FormattedOutput(pi,1.1)
Precision should be a positive integer, please re-enter: 1.2
Precision should be a positive integer, please re-enter: 4
Rounding off to  4 decimal place, x = 3.1416
```

Write down the output of `FormattedOutput(1/pi,5.2)`, with first re-entered p equals to 3.

```
>> FormattedOutput(1/pi,5.2)


```

Solution

```
function FormattedOutput(x,p)

    while mod(p,1) ~= 0 || p <= 0
       p = input('Precision should be a positive integer, please re-enter:
    end

    str = sprintf('Rounding off to %%2i decimal place,
    x = %%%d.%df\\n', p+2, p); %They are in the same line

    fprintf(str,p,x);

end
```

3. (25 marks) In this question, you need to create a MATLAB function `PrintToFile(x,n)` to write into a text file. In the file, you have to write down different values of the function $(2x)^n$. Here x, n are both integers. Follow the instructions below:

   (a) Use `fopen` with suitable parameters so that you can write into a file named `q2.txt`.

   (b) Use 2 `for` loops to write the values into the file. Write $(2j)^i$ to the i-th row, j-th column. Remember to open a new line after filling in all entries of a row. In this question, set `x = 4`, `n = 9`.

   (c) Close the file using `fclose`.

Sample Output of `PrintToFile(4,9)`

```
   2.00e+00     4.00e+00     6.00e+00     8.00e+00
   4.00e+00     1.60e+01     3.60e+01     6.40e+01
   8.00e+00     6.40e+01     2.16e+02     5.12e+02
   1.60e+01     2.56e+02     1.30e+03     4.10e+03
   3.20e+01     1.02e+03     7.78e+03     3.28e+04
   6.40e+01     4.10e+03     4.67e+04     2.62e+05
   1.28e+02     1.64e+04     2.80e+05     2.10e+06
   2.56e+02     6.55e+04     1.68e+06     1.68e+07
   5.12e+02     2.62e+05     1.01e+07     1.34e+08
^^
```

Remark: `^^` is used to remind you what `m,n,L` should be in `%m.nL`. It is not an output.

Solution

```
function PrintToFile(x,n)
    datafile = fopen('q2.txt','w');
    for i = 1:n
        for j = 1:x
            fprintf(datafile,'%10.2e  ', (2*j)^i);
        end
        fprintf(datafile,'\n');
    end
    fclose(datafile);

end
```

4. (25 marks) Create a function `AppendToFile(m)`. Follow the instructions below.

   (a) Open `AppendToFile.m`.

   (b) Obtain the data in `q3.txt`, and save it as a row vector

   $$\mathbf{n} = (1, 2, 3, \cdots, 210)$$

   (c) Use `numel`, `mod` and `find` to find values in $\mathbf{n}$ that are divisible by `m`.

   (d) Let `m = 2`, append the result into `q3.txt`. Your answer should look like the sample output.

Sample Output of `AppendToFile(2)`

```
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
 16  17  18  19  20  21  22  23  24  25  26  27  28  29  30
 31  32  33  34  35  36  37  38  39  40  41  42  43  44  45
 46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
 61  62  63  64  65  66  67  68  69  70  71  72  73  74  75
 76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
 91  92  93  94  95  96  97  98  99 100 101 102 103 104 105
106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135
136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
151 152 153 154 155 156 157 158 159 160 161 162 163 164 165
166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
181 182 183 184 185 186 187 188 189 190 191 192 193 194 195
196 197 198 199 200 201 202 203 204 205 206 207 208 209 210

There are 105 numbers that are not divisible by  2. They are

1    3    5    7    9    11   13   15   17   19   21   23   25
27   29   31   33   35   37   39   41   43   45   47   49   51
53   55   57   59   61   63   65   67   69   71   73   75   77
79   81   83   85   87   89   91   93   95   97   99   101  103
105  107  109  111  113  115  117  119  121  123  125  127  129
131  133  135  137  139  141  143  145  147  149  151  153  155
157  159  161  163  165  167  169  171  173  175  177  179  181
183  185  187  189  191  193  195  197  199  201  203  205  207
209
```

Remark: As long as you can print all the numbers that are not divisible by `m`, your answer is regarded as correct.

Solution

```
function AppendToFile(m)
    datafile = fopen('q3.txt','r');
    n = fscanf(datafile,'%g %g', [1, inf]);
    fclose(datafile);

    loc = find(mod(n,m)~=0);
    s = numel(loc);
    num = n(loc);
    datafile = fopen('q3.txt','a');
    fprintf(datafile,'\nThere are %3d numbers that are not divisible by %2d
    They are\n\n' ,s,m); %They are in the same line

    fprintf(datafile, num2str(num));
    fclose(datafile);
end
```