ENCMP 100 – Computer Programming for Engineers
Assignment #1
Due: Monday, Sept. 30, 2013 6:00pm MST

Objective

This assignment is designed to provide you with an introduction to MATLAB. You will create your own M-file to perform some basic matrix manipulations, and write a program to calculate mortgage interest.

Marking Scheme

This assignment is worth 3% of your final mark. You will get a total of 50 points for completing the following:

TASK	POINTS
Part A: Correct display of data	15
Part B: Correct code for calculation of monthly payments, total paid, and amount of remaining principal	25
Quality of code - 5 marks each for the submission	10
TOTAL	50

Breakdown for Quality of Code:

- Complete file header (see under the Submission heading for an example)
- Design (appropriate use and naming of variable
- Comments in the code
- Layout (indentation / spacing)

Submission

- Filename naming convention Assign1A_<UofA_ID_Number>.m
  Ex. U of A ID Number: 1234567890 filename for assignment #1 is Assign1A_1234567890.m
- Submit only your **.m** file under Assignment 1 section in your eClass/Moodle account.
- The assignment is due on Monday, Sept. 30, 2013 at 6:00 pm MST.
- A sample header is provided below, which must be included in all assignments:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Course: ENCMP 100
% Assignment: 1
% Lab Section: A2
% Name: Joe MacDonald
% CCID: jmac
% U of A ID: 1234567890
% Date: Sept. 3, 2013
%
% Acknowledgements:
% I received help from Jason Smith on matrix multiplication
%
% Description:
% This program will show some basic matrix manipulations.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

NOTE: When including the header from above, please change the details to contain your information. Also this header information must be included in all assignment, failure to do so will result in a failing assignment mark.

Background

The simplest way to define a matrix is to use a list of numbers. For example, the statement `x = [1 2 3 4]` will return the row vector:

```
x =
      1      2      3      4
```

A new row is indicated by a semicolon. An example of a matrix containing both rows and columns is created with the statement `y = [1 2 3 4; 2 3 4 5; 3 4 5 6]`, which returns:

```
y =
      1      2      3      4
      2      3      4      5
      3      4      5      6
```

The colon operator is very powerful for defining new matrices and modifying existing ones. When a colon is used in a matrix reference in place of a specific index number, the colon represents the entire row or column. For example, the statement `a = y(:,1)` will equate matrix `a` to column 1 of matrix `y`.

MATLAB contains many built-in functions to create and manipulate/analyze matrices. These include `sum, max, min, mean, median`, and `mode`, as well as many matrix algebra functions.

MATLAB supports two types of operations between arrays, known as array operations and matrix operations.

Array operations are operations that are performed between arrays on an element-by-element basis. For example, if

$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} -1 & 3 \\ -2 & 1 \end{bmatrix}$ then $\mathbf{A} + \mathbf{B} = \begin{bmatrix} 0 & 5 \\ 1 & 5 \end{bmatrix}$ and $\mathbf{A}.* \mathbf{B} = \begin{bmatrix} -1 & 6 \\ -6 & 4 \end{bmatrix}$.

Note: The number of rows and columns in both arrays must be the same.

Array operations may also occur between an array and a scalar. For example, if

$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, then $\mathbf{A} + 4 = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$.

Matrix operations, in contrast, follow the normal rules of linear algebra, such as matrix multiplication. In linear algebra, the product $\mathbf{C} = \mathbf{AB}$ is defined by the equation

$$C(i,j) = \sum_{k=1}^{n} A(i,k)B(k,j)$$

For example, if $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} -1 & 3 \\ -2 & 1 \end{bmatrix}$ then $a * b = \begin{bmatrix} -5 & 5 \\ -11 & 13 \end{bmatrix}$ and is represented in MATLAB by the expression `A*B`. Recall that matrix multiplication is not commutative – the order in which matrices are multiplied is important. For example, the result of `B*A` is $\begin{bmatrix} 8 & 10 \\ 1 & 0 \end{bmatrix}$.

Note: The number of columns in matrix a must be equal to the number of rows in matrix b.

Important: MATLAB uses a special symbol to distinguish array operations from matrix operations. A period is used before the symbol to indicate an array operation. For example,

- The MATLAB form for array multiplication is `A .* B`
- The MATLAB form for matrix multiplication is `A * B`

Assignment Description:

Part A: Basic Operations with Matrices

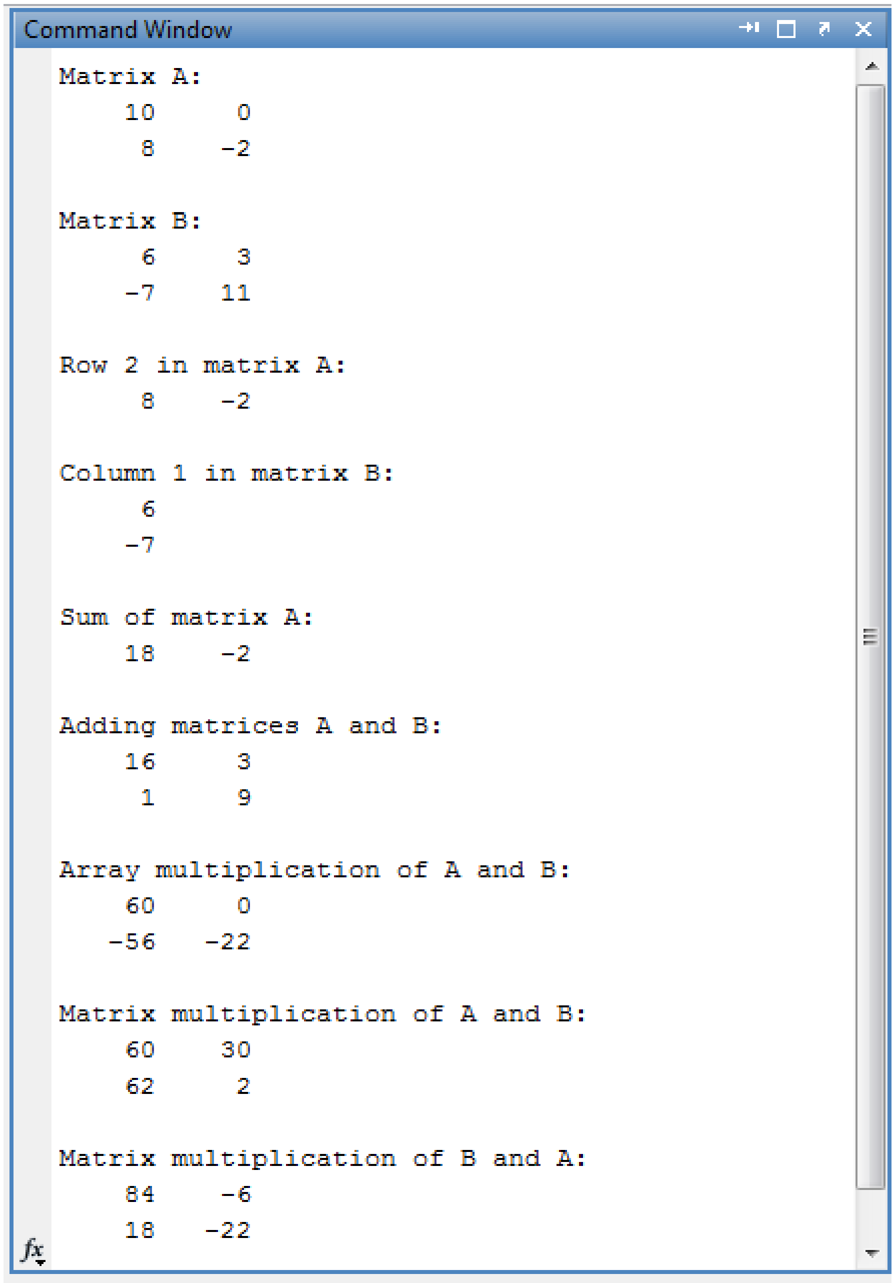
In this portion of the assignment you will be asked to create two arrays, perform some manipulations on the arrays and print the results to the screen.

Important: For the testing of this portion of the assignment, the automatic echoing of values in the command window must be suppressed. Please ensure you put a semicolon (;) at the end of each statement in order to suppress the echoing of values.

The program must perform the following steps:

1. Populate an array `A`. Display the array `A` contents to the command window with the title “`Matrix A:`”. The array `A` should represent the following:
   $\mathbf{A} = \begin{bmatrix} 10 & 0 \\ 8 & -2 \end{bmatrix}.$
2. Populate an array `B`. Display the array `B` contents to the command window with the title “`Matrix B:`”. The array `B` should represent the following:
   $\mathbf{B} = \begin{bmatrix} 6 & 3 \\ -7 & 11 \end{bmatrix}.$
3. Select row 2 in matrix `A` using the colon operator. Display the output to the command window with the title “`Row 2 in matrix A:`”.
4. Select column 1 in matrix `B` using the colon operator. Display the output to the command window with the title “`Column 1 in matrix B:`”.
5. Use the built-in sum command to find the sum of each column in matrix `A`. Display the results to the command window with the title “`Sum of matrix A:`”.
6. Add the matrices `A` and `B`. Display the results to the command window with the title “`Adding matrices A and B:`”.
7. Find the array (element-by-element) multiplication of `A` and `B`. Display the results to the command window with the title “`Array multiplication of A and B:`”.
8. Find the matrix multiplication of `A` and `B`. Display the results to the command window with the title “`Matrix multiplication of A and B:`”.
9. Find the matrix multiplication of `B` and `A`. Display the results to the command window with the title “`Matrix multiplication of B and A:`”.

Your results must match exactly with the screenshot below:



```
Matrix A:
    10     0
     8    -2

Matrix B:
     6     3
    -7    11

Row 2 in matrix A:
     8    -2

Column 1 in matrix B:
     6
    -7

Sum of matrix A:
    18    -2

Adding matrices A and B:
    16     3
     1     9

Array multiplication of A and B:
    60     0
   -56   -22

Matrix multiplication of A and B:
    60    30
    62     2

Matrix multiplication of B and A:
    84    -6
    18   -22
```

Part A Submission:

For part A of this assignment, please submit your solution to eClass under Assignment 1. With the following naming convention:

Assign1A_<UofA_ID>.m
Ex. U of A ID Number: 1234567890 filename will be
Assign1A_1234567890.m

Part B: Mortgage Calculator

In this portion of the assignment, you are required to develop an m-file that calculates mortgage payments based on compound interest formulas.

The m-file should prompt a user to input the amount borrowed (principal), p, the number of monthly payments, n, and an annual interest rate R in percent.

The program should convert the annual interest rate R into a monthly interest rate $r = R/1200$. The division includes the number of months in a year and a factor of 100 which takes care of the conversion between % and decimal.

Next the program should calculate monthly payments m given as:

$$m = \frac{rp}{1-(1+r)^{-n}}$$

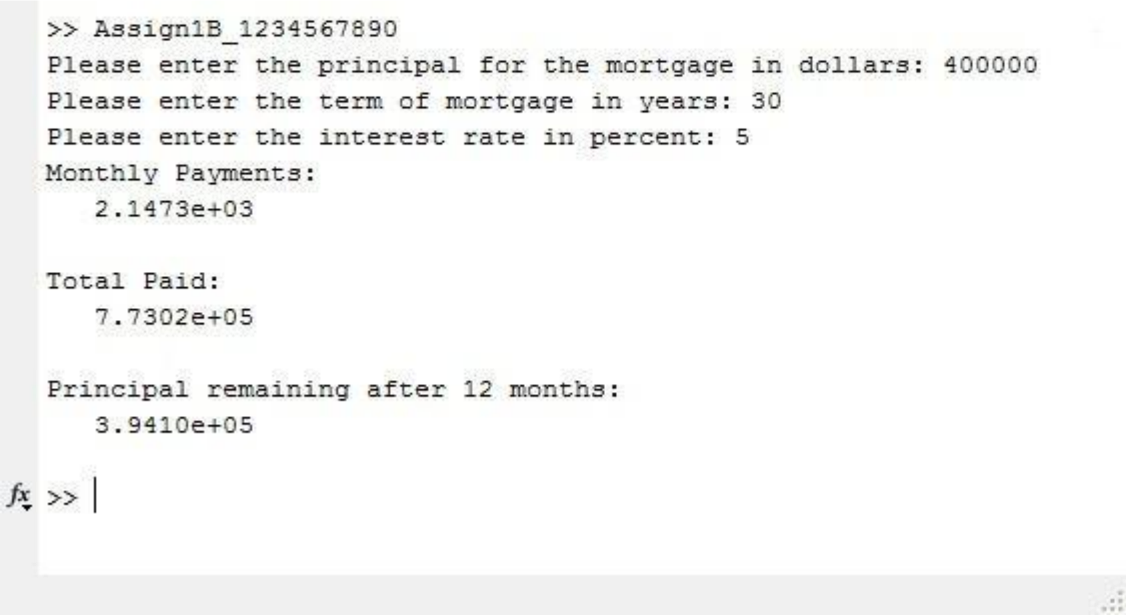
The total paid (calculated by multiplying the monthly payment by the number of monthly payments)

The amount of principal remaining after 12 months, p_{12}, calculated as:

$$p_{12} = p(1+r)^{12} - \left(\frac{m}{r}\right)[(1+r)^{12} - 1].$$

Your program should print the results of these computations to the command window with text explaining each value.

Run your code assuming that you want to buy a $400,000 home and a 30-year mortgage at a fixed annual interest rate of 5%. Your results should match exactly like this:



```
>> Assign1B_1234567890
Please enter the principal for the mortgage in dollars: 400000
Please enter the term of mortgage in years: 30
Please enter the interest rate in percent: 5
Monthly Payments:
   2.1473e+03

Total Paid:
   7.7302e+05

Principal remaining after 12 months:
   3.9410e+05

>> |
```

Hints:

1. A power operation in MATLAB is accomplished by the ^ operator. For example 2^3 is written `2^3` in MATLAB.
2. Type `help input` to learn about how to use the command `input` to prompt users for input and to record their entries from the command window.

Part B Submission:

For part B of this assignment, please submit your solution to eClass under Assignment 1B. With the following naming convention:

Assign1B_<UofA_ID>.m
Ex. U of A ID Number: 1234567890 filename will be
Assign1B_1234567890.m