UNIVERSITY OF
WATERLOO

Civil & Environmental Engineering

Midterm Examination

Spring 2019

Name:	
Signature:	**ID:**

AE 121:	**Computational Method**
Instructor:	**Chul Min Yeum**
Date/Time:	June 27, 2019 (**10:00-11:50** am)
Duration:	**110** minutes
Pages:	19 **(including cover) – Please check the number of pages.**

Notes:

1. This is an **individual open-note exam**. You can bring up to 10 pages (double-side) of your notes.
2. The maximum score that you can get is 100. You do not have to solve the extra question but it could give you extra points up to 100.
3. **Turn off your wireless device(s) and place it in your backpack/purse.**
4. **<u>The code section in each question is considered as a script file which is run in MATLAB.</u>**
5. Answer all questions in **the designated areas**. Do not use the back of pages for your answers.
6. **You can use any MATLAB built-in function at your own risk unless otherwise noted. Graders may run your script to check your answers.**
7. You need to include **a concise comment** in your scripts using '%' (comment symbol) to obtain partial grades. **<u>Note that if you do not put '%' in your script, the corresponding script is considered as a code line.</u>**
8. **Do not separate pages.**

Q	**Mark**
1	/25
2	/10
3	/15
4	/20
5	/15
6	/15
Extra	/15
Total	**/100**

Problem 1. There are five questions in this problem. You can choose **one answer** for each question. There is no partial grade for these questions, so you do not need to explain your answer. Please clearly write or mark the number of your choice in each question.

(a) Which of the following scripts have syntax errors?

(1)	`Val1 = 10`
(2)	`4val = 5`
(3)	`new@data = 8`
(4)	`val*2 = 3`
(5)	`Jason = 10+2`

1. (1), (2), and (4)
2. (2), (3), and (5)
3. (2), (3), and (4)
4. (1), (2), and (5)

(b) What value is assigned to 'sum1_3' when you run following script?

```
vec1 = [1 2; 3 4];
sum = sum(vec1);
sum1_3 = sum([1 2])
```

1. [4 6]
2. 3
3. 10
4. [1 2; 3 4]
5. Syntax or run-time error

(c) Which of the following scripts occur run-time or syntax error?

(1)	`mat1 = ones(10,10);`
	`mat1(1:2, 5:10) = [];`

(2)	`mat1 = ones(10,10);`
	`mat1(1, 5) = [];`

(3)	`mat1 = ones(10,10);`
	`mat1(1, 1:2) = 3;`

(4)	`mat1 = ones(10,10);`
	`mat1 = [1 2; 3 4];`

1. (1), (2)
2. (1), (2), (3)
3. (1), (2), (4)
4. (3), (4)
5. (1), (3), (4)

(d) Assume that x = [1 2 3 4 5 6 7 8 9 10 11 12]. You are going to extract all numbers that have 1 remainder when divided by 3 and then, assign those numbers to ‘y’. Which of the following scripts conduct this operation?

(1)	`y = x(3:1:12)`

(2)	`idx = rem(x, 1);`
	`y = x(idx);`

(3)	`idx = rem(x,3);`
	`y = x(idx==1);`

(4)	`idx = x/3;`
	`y = x(idx==1);`

(5)	`idx = rem(x,1);`
	`y = x(idx==3);`

1. (5)
2. (3)
3. (5), (1)
4. (5), (2), (1)
5. (5), (3)

(e) What variables are created when you run the following script?

```
a = 9
b = -5.2

if rem(a,4) == 2

    if a == 9
        d = 2;
    else
        e = 5;
    end

else

    if fix(b) == round(b)
        f = 6;
    elseif abs(b) == 5
        g = 7;
    end

end
```

1. ‘g’, ‘f’
2. ‘e’, ‘g’
3. ‘a’, ‘b’, ‘g’
4. ‘a’, ‘b’, ‘f’
5. ‘a’, ‘b’, ‘d’

Problem 2. Assume that matrix **A**, matrix **B**, row vector **r**, and column vector **c** are defined as follow:

$$\underset{\mathbf{A}}{\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{pmatrix}} \quad \underset{\mathbf{B}}{\begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{pmatrix}} \quad \underset{\mathbf{r}}{\begin{pmatrix} r_1 & r_2 & r_3 & r_4 \end{pmatrix}} \quad \underset{\mathbf{c}}{\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix}}$$

Then, **val_a**, **vec_b**, **vec_c**, **mat_d**, **mat_e**, **mat_f**, and **mat_g** are obtained from element-wise operations or matrix-matrix or matrix-vector multiplication using **A**, **B**, **r**, and **c**.

val_a = $c_1\,r_1 + c_2\,r_2 + c_3\,r_3$

vec_b = $\begin{pmatrix} c_1\,r_1 & c_2\,r_2 & c_3\,r_3 & c_4\,r_4 \end{pmatrix}$

vec_c =

$$\begin{pmatrix} a_{11}\,r_1 + a_{21}\,r_2 + a_{31}\,r_3 + a_{41}\,r_4 \\ a_{12}\,r_1 + a_{22}\,r_2 + a_{32}\,r_3 + a_{42}\,r_4 \\ a_{13}\,r_1 + a_{23}\,r_2 + a_{33}\,r_3 + a_{43}\,r_4 \end{pmatrix}$$

mat_d =

$$\begin{pmatrix} a_{11} + c_1 & a_{12} + c_1 & a_{13} + c_1 \\ a_{21} + c_2 & a_{22} + c_2 & a_{23} + c_2 \\ a_{31} + c_3 & a_{32} + c_3 & a_{33} + c_3 \\ a_{41} + c_4 & a_{42} + c_4 & a_{43} + c_4 \end{pmatrix}$$

mat_e =

$$\begin{pmatrix} b_{11}\,r_1 & b_{12}\,r_2 & b_{13}\,r_3 & b_{14}\,r_4 \\ b_{21}\,r_1 & b_{22}\,r_2 & b_{23}\,r_3 & b_{24}\,r_4 \\ b_{31}\,r_1 & b_{32}\,r_2 & b_{33}\,r_3 & b_{34}\,r_4 \end{pmatrix}$$

mat_f =

$$\begin{pmatrix} a_{11}\,c_1 + a_{12}\,c_2 + a_{13}\,c_3 \\ a_{21}\,c_1 + a_{22}\,c_2 + a_{23}\,c_3 \\ a_{31}\,c_1 + a_{32}\,c_2 + a_{33}\,c_3 \\ a_{41}\,c_1 + a_{42}\,c_2 + a_{43}\,c_3 \end{pmatrix}$$

mat_g =

$$\begin{pmatrix} a_{11}\,b_{11} + a_{12}\,b_{21} & a_{11}\,b_{12} + a_{12}\,b_{22} \\ a_{21}\,b_{11} + a_{22}\,b_{21} & a_{21}\,b_{12} + a_{22}\,b_{22} \end{pmatrix}$$

In MATLAB, variables named ‘A’, ‘B’, ‘r’, and ‘c’ are created and random values are assigned. Please write a script to compute ‘val_a’, ‘vec_b’, ‘vec_c’, ‘mat_d’, ‘mat_e’, ‘mat_f’, and ‘mat_g’. You need to create a single script to get all those values, like MATLAB grader problems.

Note that the dimensions of the variables are ‘A’: 4 x 3 matrix, ‘B’: 3 x 4 matrix, ‘r’: 1 x 4 vector, ‘c’: 4 x 1 vector, ‘val_a’: scalar, ‘vec_b’: 1 x 4 vector, ‘vec_c’: 3 x 1 vector, ‘mat_d’: 4 x 3 matrix, ‘mat_e’: 3 x 4 matrix, ‘mat_f’: 4 x 1 vector, and ‘mat_g’: 2 x 2 matrix.

```
A = randi(100, 4, 3);
B = randi(100, 3, 4);
r = randi(100, 1, 4);
c = randi(100, 4, 1);


%%%%%%%%%%%%%%%%%%%%%%%%%%% Your code starts here %%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem 3. Pressure data measurements have been taken during 80 days (Day 1, 2, ... 80) at 1 hr intervals throughout the entire day (there are 24 measurements for each day). Each row contains data collected at the same day. Each column contains data collected at the same time. Day 1 is Monday.

The pressure data is stored in an 80 x 24 matrix named ‘data_press’.

```
num_day = 80;
num_hr = 24;
data_press = randi(1000, num_day, num_hr);
```

Please write a script to create a variable named:
(a) ‘max_20’, which contains the maximum pressure value that occurred on **Day 20** in ‘data_press’. ‘max_20’ is a scalar value.
(b) ‘min_val’, which contains the minimum pressure value among **all pressure values** in ‘data_press’. ‘min_val’ is a scalar value.
(c) ‘min_mean_day’, which contains the day number having **the lowest average pressure of all data on each day** in ‘data_press’. ‘min_mean_day’ is a scalar value.
(d) ‘sum_sun’, which contains the sum of all pressure readings that happened on **all Sundays** in ‘data_press’. ‘sum_sun’ is a scalar value.
(e) ‘above_200’, which determines the number of pressure readings above 200 among all pressure values in ‘data_press’. ‘above_200’ is a scalar value.
Again, you need to create a single script to get all those values, like MATLAB grader problems.

```
num_day = 80;
num_hr = 24;
data_press = randi(1000, num_day, num_hr);

%%%%%%%%%%%%%%%%%%%%%%%%%%% Your code starts here %%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem 4. Assume that there are 80 students in a class and they took 10 exams during the course. A variable named 'score' is a 10 x 80 matrix and each element contains a score for each exam and student. For instance, score(3,4) is a score of the third exam for student 4. Let's say that ID of this student is 4. Another example, score(2,1) is a score of the second exam for a student having ID 1.

Here, 'score' is randomly generated and it ranges from 51 to 100. All scores are integers.

```
num_st = 80;
num_ex = 10;
score = randi(50, num_ex, num_st) + 50;
```

(a) ‘is_st5_greater_avg’, which determines if the score of Student 5 on exam 2 is greater than or equal to the average score of all students on exam 2. If yes, assign **1** to ‘is_st5_greater_avg’, and otherwise assign **0** to ‘is_st5_greater_avg’. ‘is_st5_greater_avg’ is a logical (scalar) variable.

(b) ‘is_st2_5_greater’, which determines if the score of Student 2 on exam 3 is higher than the score of Student 5 on the same exam. If Student 2 has a higher score than Student 5, assign **1** to ‘is_st2_5_greater’. If Student 5 has a higher score, assign **-1** to ‘is_st2_5_greater’. If they have the same score, assign **0** to ‘is_st2_5_greater’. ‘is_st2_5_greater’ is a scalar variable.

(c) ‘above_95’, which contains the number of students whose scores on exam 4 are more than 95. ‘above_95’ is a scalar variable.

(d) ‘st_avg_score’, which contains the average score for all exams in each student. ‘avg_score_score’ is a 1 x 80 vector.

(e) ‘num_grade_A’, which contains the number of students who get ‘**A’** grade for this course. The grading scheme is in the table below. ‘num_grade_A’ is a scalar variable. You can re-use ‘st_avg_score’ to solve this question.

Average score for all the exams	Grade
100-90	‘A’
89-80	‘B’
79-70	‘C’
Under 69	‘D’

```
num_st = 80;
num_ex = 10;
score = randi(50, num_ex, num_st) + 50;

%%%%%%%%%%%%%%%%%%%%%%%%%%% Your code starts here %%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem 5. *AE121 Bulls and Cows* is a mind game, developed by Jason. In the game, a random, **3-digit** even number (**Possible digits are 0, 2, 4, 6, 8**) is chosen, called a true number, and its values are compared to those of a test number. All rules are exactly the same as the original Bulls and Cows game that we developed in the assignments except the digits used and the number of digits evaluated. All three digits of the number are different. If any digit in the test number is the exact same value and in the exact same position as any digit in the true number, this is called a **bull**. If the digit is present in both the trial number and chosen number but is not in the same location, this is called a **cow**.

(a) Please write a function of 'CompBullsCows' to evaluate bulls and cows of a test number. This function has two inputs ('true_num' and 'test_num') and two outputs ('bulls' and 'cows').

```
function [bulls, cows] = CompBullsCows(true_num, test_num)
```

(b) Assume that two players join this game. Player 1 and player 2 each guesses a three digit even number (which are the test numbers) and they are denoted as 'test_ply1' and 'test_ply2' respectively. You are going to compare their bulls and cows and decide who is **closer** to the true number. The player with more bulls wins. If both players have equal number of bulls, then the player with more cows wins. If both players have an equal number of bulls and cows, it is a tie.
Example 1: if bulls and cows for player 1 and 2 are 1,2 and 1,1, respectively, player 1 is **closer**.
Example 2: if bulls and cows for player 1 and 2 are 1,2 and 2,0, respectively, player 2 is **closer**.
Please write a function of '**TwoPlayerBullsCows**' which has three inputs: a true number ('true_num') and two test numbers ('test_num1' and 'test_num2'), and one output called 'result'. If the test number for Player **1** is **closer** to the true number, assign 1 to 'result'. If the test number for Player 2 is **closer** to the true number, assign **2** to 'result'. If their bulls and cows are identical, assign **0** to 'result'.

```
function result = TwoPlayerBullsCows(true_num, test_num1,…
test_num2)
```

Graders recommend using a function of 'CompBullsCows' in your code. Then, graders assume that the function produces correct bulls and cows values.

CompBullsCows.m

```
function [bulls, cows] =  CompBullsCows(true_num, test_num)

%%%%%%%%%%%%%%%%%%%%%%%% Your code starts here %%%%%%%%%%%%%%%%%%%%%%
```

TwoPlayerBullsCows.m

```
function result =  TwoPlayerBullsCows(true_num, test_num1,test_num2)

%%%%%%%%%%%%%%%%%%%%%%%%% Your code starts here %%%%%%%%%%%%%%%%%%%%%%%
```

Question 6. The standard 52-card deck has 13 numbers and four different suits. The suit order is 'Clubs', 'Diamonds', 'Hearts', and 'Spades'. Seven Card Stud is a classic poker game where seven cards are distributed to each player. Each integer from 1 to 52 will represent the value and suit of a card, where from 1 to 52, the value and suit of the cards will proceed in the following order:

Integer	Value and Suit
1	1 (Ace) of Clubs (C)
2	1 (Ace) of Diamonds (D)
3	1 (Ace) of Hearts (H)
4	1 (Ace) of Spades (S)
5	2 of Clubs (C)
.	.
.	.
49	13 (King) of Clubs (C)
50	13 (King) of Diamonds (D)
51	13 (King) of Hearts (H)
52	13 (King) of Spades (S)

(a) Jason makes a new rank called ‘Three Pairs’ that contains three sets of two cards of one number. For example, the following card sets are ranked as ‘Three Pairs’:
[2C, 2D, 3C, 3D, 5S, 5H, 11], [3H, 3S, 10H 10D, 11C, 11D, 1D], [3D, 3H, 10D 10S, 11H, 11S, 1C]

Please write a function of ‘ChckThrPrs, which has one input of a test card set and one output that tells you if the rank is ‘Three Pairs’ or not. Please assign 1 to ‘is_thr_prs’ if the rank of the test card is ‘Three Pairs’ and otherwise, assign 0 to ‘is_thr_prs’. The input variable named ‘your_card’ is a 1 x 7 vector including seven integers indicating cards with the above order.

```
function is_thr_prs =  ChckThrPrs(your_cards)
```

(b) Jason makes another new rank called ‘Two Three of a Kind’ that contains two sets of three cards of one number. For example, the following card sets are ranked as ‘Two Three of a Kind’: [2C, 2D, 2H, 5D, 5S, 5H, 11C], [3H, 10S, 10H 10D, 11C, 11H, 11D], [1C, 1H, 1S, 2C, 2D, 2H, 3C]

Please write a function of ‘Two Three of a Kind’ which has one input of a test card set and one output that tells you if the rank of the test card is ‘Two Three of a Kind’ or not. Please assign 1 to ‘is_tw_thr_knd’ if the rank is ‘Two Three of a Kind’ and otherwise, assign 0 to ‘is_tw_thr_knd’. The input variable named ‘your_card’ is a 1 x 7 vector including seven integers indicating cards with the above order.

```
function is_tw_thr_knd = ChckTwThrKnd(your_cards)
```

ChckThrPrs.m

```
function is_thr_prs =  ChckThrPrs(your_cards)

%%%%%%%%%%%%%%%%%%%%%%%%%%% Your code starts here %%%%%%%%%%%%%%%%%%%%%%%%%
```

ChckTwThrKnd.m

```
function is_tw_thr_knd = ChckTwThrKnd(your_cards)

%%%%%%%%%%%%%%%%%%%%%%%%%%% Your code starts here %%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem 7. 500 points with x, y, and z coordinates are provided row-wise (ie. [x_1, y_1, z_1; x_2, y_2, z_3; x_3, y_3, z_3; …; x_{500}, y_{500}, z_{500}] along with their corresponding ids, which are given as 1, 2, … 500 in order. Please write a script to find the minimum and maximum distance between a pair of two **different** points and their ids.

```
num_point = 500;
points_3D = randi(1000, num_point, 3);
point_ids = 1:num_point;
```

(a) Please compute the maximum distance between two points and assign the value to a variable named ‘max_dist’, and store one of the point ids to variable ‘id_max1’ and the other to ‘id_max2’.

(b) Please compute the minimum distance between two points and assign the value to a variable named ‘min_dist’, and store one of the point ids to variable ‘id_min1’ and the other to ‘id_min2’. For example, there are four 3D points, points = [1 1 1; 2 2 2; 10 10 10; 12 12 12].

‘max_dist’ becomes 19.0526 between two point 1 and 4. ‘id_max1’ and ‘id_max2’ becomes 1 and 4, respectively. ‘min_dist’ becomes 1.7321 between two point 1 and 2. ‘id_min1’ and ‘id_min2’ becomes 1 and 2, respectively.

Note that the distance (d) from the point (x_1, y_1, z_1) to the point (x_2, y_2, z_2) is

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

```
num_point = 500;
points_3D = randi(1000, num_point, 3);
point_ids = 1:num_point;

%%%%%%%%%%%%%%%%%%%%%%%%%%% Your code starts here %%%%%%%%%%%%%%%%%%%%%%%%%
```