E7 Midterm Exam 1

NAME
SID
LAB

| #11: TuTh 8-10 | #12: TuTh 10-12 | #13: TuTh 12-2 |
| #14: TuTh 2-4 | #15: TuTh 4-6  | #16: MW 10-12  |
| #17: MW 12-2 | #18: MW 2-4  | #19: MW 4-6 |

Carefully read and follow these instructions:
1. Write your name on the top left corner of each page.
2. Start answering the exam only when instructed to do so.
3. Record your answers only in the spaces provided.
4. You may not ask questions during the exam.
5. You may not leave the exam room before the exam ends.
6. You may not use any electronic devices.
7. You may use one 8.5×11 sheet (2 pages) of handwritten notes.
8. Count the number of pages before the start of the exam.
   There should be 10 pages.
1. (a) (2 points) A MATLAB user tries to execute a script file *trash2.m* and receives an error (see Figure 1). What is the cause for this error? How can the user fix the error?

```
>> trash2
Undefined function or variable 'trash2'.
```

Figure 1: Screenshot from MATLAB for Problem 1 part (a)

Answer:

2. **Arrays**

Given

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$$C = \begin{bmatrix} 4 & 1 \\ 2 & 3 \\ 3 & 2 \end{bmatrix}$$

(a) (1 point) Write a line of MATLAB code to multiply B and C to create a 3-by-2 array.

Answer:

(b) (2 points) Write a line of MATLAB code to evaluate the following mathematical expression, resulting in a 3-by-2 array. The values of $x$ are contained in the variable B and the values of $y$ are contained in the variable C.

$$e^{x^2+y^2} + 1$$

Answer:
3. Arrays

(a) (4 points) Use built-in MATLAB functions and other methods discussed in class to construct the 2-D array \( A \) on a row-by-row basis. For this problem, manual assignment of each element in \( A \) will receive 0 points.

\[
A = \begin{bmatrix}
1 & 2 & 3 & 4 \\
3 & 3 & 3 & 3 \\
1.00 & 2.67 & 4.33 & 6 \\
0 & -2 & -4 & -6 \\
\end{bmatrix}
\]

Answer:

(b) (3 points) What are the values of variables \( C \), \( D \), and \( E \) after the following code block is executed?

```
begin code
B = logical([1 0 1 0]);
C = A([1, 2, 4], 2:4);
D = A(B,B);
E = A(9)+A(11);
end code
```

Answer:

```
begin code
C =
D =
E =
end code
```
4. **Relational Operators**

(a) (1 point) What is displayed on the command window upon executing the following?

```
begin code

>> all('pass'>'fail')
```

**Answer:**

(b) (3 points) What is displayed on the command window upon executing the following?

```
A = [1 0 10 42];
B = logical(A);
C = 'isequal('pass','fail')
if C & all(A ~=B)
    disp('esta prueba es facil')
else
    disp('esta prueba es dificil')
end
```

**Answer:**
(c) (5 points) What is displayed on the command window upon executing the following?

```matlab
A = reshape(1:25,5,5);
B = A';
C = A([1 3], [2 4])
D = B([2:2:4],[1:2:3])
C == D
```

Answer:

```matlab
A =

B =

A([1 3], [2 4]) =

B([2:2:4],[1:2:3])

ans =
```

...
5. **Conditional Statements**

Consider the following code:

```plaintext
begin code

if W<0
    if (ceil(D)==floor(D)) (ceil(D)>=abs(E))
        B = 16;
    else
        if E<0
            B = 4;
        else
            B = abs(E);
    end
else
    A = B - sqrt(B);
elseif E*D*W<0
    if round(W)==W
        B = 5.5*E;
    elseif B<-10
        A = 5;
    else
        A = 1;
    end
else
    DD = 25;
    A = sqrt(DD);
end
else
    A = 0;
end

end code
```

For each of the following values of the variables \(W, D,\) and \(E\) stored in the workspace before executing the code, what would be the resultant value of the variable \(A\) after the code block is executed?

(a) (2 points) \(W = 2; D = 3.9; E = -9;\)
   
   **Answer:**

(b) (2 points) \(W = 2; D = 3.9; E = 9;\)
   
   **Answer:**

(c) (2 points) \(W = -2; D = 3.9; E = -9;\)
   
   **Answer:**

6 of 10
6. **Loops**

Part (a) is based on the following code block.

```matlab
A = [3 1 -1 3.5 5];
for i = 2*A
    if i > 8
        B = 8
    elseif (i > 2) & (i <= 6)
        B = 6
    elseif i >= -1
        B = 5
    else
        B = 4
    end
end
```

(a) (5 points) What values of B will be displayed on the command window after the code block is executed?

**Answer:**

```matlab
B =
```

```matlab
4
B
6
B
9
B
10
B
13
B
17
B
```
Parts (b) and (c) are based on the following code block.

```
1  count = 1;
2  steps = 0;
3  while steps < 5
4      if mod(count,2) == 0
5          steps = steps + 2;
6      else
7          steps = steps - 1;
8      end
9  count = count + 1;
10 end
```

(b) (2 points) What is the value of the variable `count` after the code block is executed?
Answer: 

(c) (2 points) What is the value of the variable `steps` after the code block is executed?
Answer: 5
7. Functions (11 points)
Answer the following subproblems in the space provided on the next page.

(a) In this problem, you will create a function called StandUP that has one input variable called
currentTime and one output variable called MilTime.

(b) The function should first try to open a file in the current directory called last_run.mat.
The file last_run.mat contains the variable last_run, which represents time at which the
function was last executed. If there is no last_run.mat file, and therefore, no variable called
last_run, set the variable last_run to 0.

(c) The input currentTime is the MATLAB built-in variable clock. See the abbreviated MAT-
LAB help documentation below for the built-in variable clock. Note that class(clock) is
double and the time displayed is in military notation (24-hour clock).

```matlab
begin code
>> help clock
  clock  Current date and time as date vector.
  C = clock returns a six element date vector containing the
  current time and date in decimal form:
  [year month day hour minute seconds]
  The sixth element of the date vector output (seconds) is accurate
  to several digits beyond the decimal point. FIX(clock) rounds to
  integer display format.
end code
```

The output MilTime is the last 3 elements of currentTime converted into a scalar double
(round down to the nearest minute). For example, if currentTime = [2015 10 9 13 30
42], then MilTime = 1330.

(d) Then, if it has been over an hour since the program was last executed, the function should
display a message for the user to Stand UP!. This could be determined by checking the
value of the variable last_run, which stores the time at which the function was last executed
in the same format as MilTime (see part (e)).

(e) The function should assign the variable MilTime to last_run and save the updated last_run
to a .mat file in the current folder so the information can be retrieved the next time that
the function is called.

(f) Finally, you should create a function handle called handleStandUP in the command window
that points MATLAB to the function StandUP.
Answer: Content of StandUP.m

%Part (a): Define the function header

    function MILTime = StandUP (CurrentTime)

%Part (b): Try to load last_run.mat

    try
        load (last_run)
    catch
        last_run = 0
    end

%Part (c): Convert vector input to a scalar military time

    MILTime = CurrentTime (4) * 1000 + CurrentTime (5)

%Part (d): Check if an hour has passed

    if MILTime = (last_run (4) * 1000 + CurrentTime (5)) ≥ 100
        disp ('Stand UP! ')
    end

%Part (e): Save the current time as a variable called last_run.

    last_run = MILTime

    MILTime = last_run

Command window display

%Part (f): Assign a function handle called handleStandUP.

    >> handleStandUP = @StandUP

end code