FALL 2018 PRACTICE EXAM

1: TRUE OR FALSE QUESTIONS

1. T [✓ F] The **break** command is used to end a program when something bad happens

(Return or error command)

2. T [✓ F] **If** statements and **for** loops can be used inside of a function-handler variable

y = @(x) x.^2

3. [✓ T] F MATLAB® will help you find syntax errors in you program, but cannot find logical (or math) errors

bad code bad math

4. [✓ T] F The **length** function returns the number of values in a 1D array.

5. T [✓ F] When an **If-elseif-else** construct is used, the **else** option should include a logical condition.

should not

2: MULTIPLE CHOICE QUESTIONS

1. What does the **clear** command do?

   A. Closes all the open figure windows close all
   [✓ B.] Clears all the variables saved in the workspace
   C. Clears all the output in the command window clc
   D. Stops a program that is running ctrl - c

2. For a 2D array called cake, the command

```
CAKE(:,1) = pi
```

   A. Change the values in CAKE that are equal to 1
   [✓ B.] Changes all the values in the 1st column to 3.14159
   C. Changes all the values in the 1st row to 3.14159
   D. Outputs a program error.

3. For the code

```
soda = [41  23  13  45];
pop = abs(soda(1) - soda(4));
```

|41 - 45| = |-4| = 4

What is the value of variable pop?

   A. pop = -4
   B. pop = 14
   C. pop = 3
   [✓ D.] pop = 4

4. Consider the summation

$$R = \sum_{n=1}^{\infty} \frac{1}{n^2}$$

which can be coded as shown on the right.

```
R = 0;
n = 1;
while 1
    R = R+1/n^2;
    n = n + 1;
end
```

What is the final value of variable **result** *from the code*?

   [✓ A.] Ummm… the loop doesn't stop. No final value.
   B. This is a p-series sum, R=1.6449.
   C. The loop doesn't start. R = 0.
   D. The loop runs one time. R = 1

5. For the code in problem 4, what commands would make the code run better?

   A. Add a **break** command if **n** gets too big. ✓
   B. Change the **while** loop to a **for** loop, with n=1:1:100. ✓
   C. Add a condition to the **while** loop that stops when R stops changing ✓
   D. It runs fine. No change is needed. ✗
   [✓ E.] A, B, or C will make the code run better.

3: Matching

Suppose that two 1D arrays are typed into MATLAB®:

```
A = [1 2 3];
B = [4 5 6];
```

} 1 row , 3 columns

Match the operation to the correct output.

b 1) A*B [1x3] * [1x3] error	a) ans = 4 10 18
a 2) A.*B element-wise [1*4 2*5 3*6]	b) Error
e 3) A./B element-wise [1/4 2/5 3/6]	c) ans = 32
c 4) A*B' transpose [1x3] * [3x1] = [1x1]	d) ans = 4 5 6 8 10 12 12 15 18
d 5) A'*B transpose [3x1] * [1x3] = [3x3]	e) ans = 0.2500 0.4000 0.5000

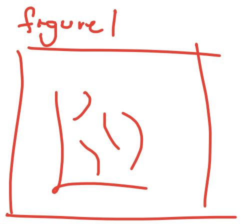
4: 2D Array Commands

The following array has been entered into the MATLAB® workspace:

```
A =
    -1     2    -1
    -7    -2     8
     6    -7     3
```

What is the output from for each command?

1. `>> A(2,:)` Row 2 all columns ans = [-7 -2 8]
2. `>> size(A)` number of rows and number of columns ans = [3 3]
3. `>> max(A)` maximum for each row ans = [2 8 6]
4. `>> max(max(A))` maximum of the maximum for each row ans = 8
5. `>> contour(A)` Creates a 2D figure showing the contour lines based on the values of A



Note: does not have to be an exact match to the data

5: Understanding Matlab

In lab, you wrote a code that computes $\cos(x)$, using an infinite sum

$$\cos(x) = 1 + \sum_{n=1}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}$$

Below is a 'broken' program of the solution.

```
 1  clear; close all; clc
 2
 3  x = input('Pick a value for x: ','s');
 4  N = input('Number in series (even): ');
 5
 6  xpow = 1;
 7  posneg = 1;
 8  result = 0;
 9
10  for itar = 2:2:
11    xpow = xpow*x*x
12    faq = faq*itar*(itar-1);
13    posneg = posneg*(-1);
14    result(itar) = result(itar-1)+posneg*xpow/faq;
15    break;
16  end
17
18  funCos = cosine(x);
19
20  fprintf('The approximate value is %i \n', result);
21  disp('The function value is %0.6f \n', funCos);
22  fprintf('The difference is %0.6f \n', funCos-result);
```

Is result an array?

The code has 10 errors (GPP, syntax, or math). In the space below, identify 5 of these errors and say how to fix them. (+5 bonus if you identify all of them)

Line 3: delete 's', x is a number, not a text

Line 8: should size the array : result = zeros(1, N+1) result(1) = 1 or result = ones(1, N+1)
or result = 1 ; (single value)

Line 10: bad for Loop notation
□ needs end value 2: 2: 2N
or □ use 1: N

Line 11: missing ; to stop screen output

Line 12: faq does not have a starting value

Line 14: depending on how Line 8 and 10 are fixed...
□ by 2's result(itar-1) is always zero or one
or □ by 1's no error on □ single value: remove index

Line 15: Break will execute on the first time through the Loop

Line 18: No cosine function exists.
must use cos (x)

Line 20: Using % i to output a real number (decimal)
□ if result is an array: using one %i to show entire array

Line 21: Bad use of disp command
- should use another fprintf

6: SMALL CODE

Using the variables below. Write a code completes the following:

- □ sums the value of each element of **book** into **total**
- □ uses **count** to determine the number of positive values in **book**
- □ uses an if statement set all the negative values to 0
- □ shows the total at the end of the program

```
Clear; clc; close all;
book = randi([-100 100],200,300);


[R  C] = size(book);

total =  sum(sum(book)) ;
count  =  0 ;
for  i = 1:R
   for j = 1:C
        if  book(i,j) > 0
            count =  count + 1;
        else
            book(i,j) = 0
        end
    end
end
fprintf('the total of the values is %i \n', total)
```