E7 Midterm Examination 2
November 2, 2018

NAME : ____________________________

SID : ____________________________

<table>
<thead>
<tr>
<th>Part</th>
<th>Points</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>46</td>
<td></td>
</tr>
</tbody>
</table>

Notes: 1. Write your full name and SID in the spaces provided above.
2. Write your name on the top right corner of each page.
3. Record your answers only in the spaces provided.
4. This is a closed-book exam.
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 (cell phones, computers) during the exam.
**Part A - Problem 1** (2 points)

Consider the following code:

```matlab
a = [2 8 8 2 3 0 7 0 6
     6 5 8 1 5 9 3 2 4
     6 9 0 1 4 9 2 3 5
     7 0 3 8 0 4 4 8 2
     4 4 2 5 2 4 0 0 7
     0 1 8 5 1 3 1 0 1
     2 9 4 2 1 9 9 1 6
     9 0 9 8 2 3 9 6 1
     1 7 1 6 4 1 5 7 3];

b = 0;
n = size(a, 2);
while b < 15
    b = b + a(4, n);
    n = n - 1;
end
```

After execution of the above code, fill out in the spaces provided the results of the following statements:

A.1 (1 point)

```matlab
>> b
ans =
```

A.2 (1 point)

```matlab
>> n
ans =
```
**Part A - Problem 2** (4 points)

Consider the following code:

```matlab
a = zeros(1,100);
for n = 1:100
    for m = 2:sqrt(n)
        if mod(n,m)==0
            a(n) = a(n) + 1;
        end
    end
end
```

After execution of the above code, fill out in the spaces provided the results of the following statements:

A.3 (1 point)

```matlab
>> a(2)
```

`ans =` ________

A.4 (1 point)

```matlab
>> a(24)
```

`ans =` ________

A.5 (2 point)

```matlab
>> sum(a(10:20)==0)
```

`ans =` ________
**Part B** (2 points) Consider the following code:

```plaintext
n = 0;
count = 0;
while n < 1e3
    count = count + 1;
    n = 2^n;
if n >= 4
    n = n - 1;
break
end
end
```

After running the above code, fill out in the spaces provided the results of the following statements:

B.1 (1 point)

```plaintext
>> count

ans =
```

B.2 (1 point)

```plaintext
>> n

ans =
```
Part C (9 points)

Consider a function called `convert`, which is intended to convert letter grades to grade points, such that:

- `convert('A')` returns 4.0
- `convert('B')` returns 3.0
- `convert('C')` returns 2.0
- `convert('D')` returns 1.0
- `convert('F')` returns 0.0

The function presently reads as follows:

```plaintext
1  function point = convert(letterGrade)
2
3      if letterGrade == 'A'
4          point = 4.0;
5
6      elseif letterGrade == 'B'
7          point = 3.0;
8
9      elseif letterGrade == 'C'
10         point = 2.0;
11
12     else    letterGrade == 'D'
13         point = 1.0;
14
15     elseif letterGrade == 'F'
16         point = 1.0;
17
18     end
19
20 end
```
C.1 (5 points) Trying out this function, it is observed that it does not work properly. Identify and fix all the errors in the function, by writing below the line numbers (using the enumeration of the code lines as above) and the corresponding corrections.


C.2 (4 points) Below, you are given the function `convertArray` that changes an array of letter grades `letterArray` to an array of grade points `pointArray` using the above `convert` function. The `letterArray` function has incomplete parts. Fill them out in the spaces provided.

```matlab
function pointArray = convertArray(letterArray)
    % letterArray: an input array of letter grades
    % pointArray: an output array of point grades
    % e.g., calling
    %     >> convertArray(['A','B','C','D'])
    % returns
    %     >> ans =
    %     [4.0, 3.0, 2.0, 1.0]
    %
    pointArray = [];

    while ____________
        pointArray = [pointArray, convert(_______)];

        letterArray(1) = [];
    end

end
```
Part D (4 points) Consider the following function functionD:

```matlab
function [output] = functionD(input, count)
    if count < 10 && input >= 2
        output = functionD(input/2, count + 1);
        count = count + 1;
    else
        output = count + input;
    end
end
```

Answer the following questions:

D.1 (1 point) What is the output of the following call?

```matlab
>> functionD(32, 3)
```

ans =  

D.2 (1 point) How many times is functionD called in part D.1?

D.3 (1 point) What is the output of the following call?

```matlab
>> functionD(256, 8)
```

ans =  

D.4 (1 point) How many times is functionD called in part D.3?
**Part E** (6 points)

We want to find the minimum in a given double array of numbers recursively. We have written the following code:

E.1 (4 points) Fill in the blank lines in the code below.

```matlab
function num = findmin(array)
    if length(array) == 1
        num = array;
    elseif array(1) < findmin(array(2:end))
        num = ________________
    else
        num = ________________
    end
```

E.2 (1 point) What does this function return if we type the following command:

```matlab
>> findmin([2,1,3,1,5])
ans = ________________
```

E.3 (1 point) How many times the `findmin` function is called in part E.2?

________________________
Part F (3 points)

F.1 (1 point) What is the output of the following code (true/ false)?

```
>> (4 + eps)/4 == 1
```

ans = ______________________

F.2 (1 point) Given

```
>> realmax
```

ans =

```
1.7977e+308
```

What is the output of the following code?

```
>> [10,2].^10000
```

ans = ______________________

F.3 (1 point) Convert the number 121 in the base-3 system to a number in the decimal (base-10) system. The answer is:

ans = ______________________
Part G (8 points) Consider a quadratic polynomial of the form

\[ y(x) = c_2 x^2 + c_1 x + c_0. \]

Assume that the polynomial \( y(x) \) satisfies the following equations:

\[ y(0) = 4 \]
\[ y(1) = 1 \]
\[ y(2) = 0 \]

The coefficients \( c_2, c_1, c_0 \) are determined by solving the system of linear algebraic equations

\[ Ac = b. \]

Complete the matrix \( A \) and the vector \( b \) in the spaces provided below (note that the vector \( c \) has coefficients stored in the order \( [c_2; c_1; c_0] \)):

\[
\text{>> A = [} \\
\text{]} \;
\]

\[
\text{>> b = [} \\
\text{]} \;
\]

\[
\text{>> c = A}\backslash b ;
\]
Part H (8 points)

H.1 (6 points)

Consider the nonlinear function

\[ f(x) = 1 + \sin(x) - x. \]

We want to use fixed-point iteration method to find a root of \( f(x) \). Complete the code below, in the four designated lines, to accomplish this.

\[
\% \text{ fixed-point iteration to find the root} \\
\% \text{ of function } f = 1 + \sin(x) - x \\
tol = 1e-3; \quad \% \text{ tolerance (relative)} \\
xprev = pi;
\]

\[ xnext = \quad \quad \quad \quad \quad \quad \] \\
while \[ \quad \quad \quad \quad \quad \quad \] \\
\[ xprev = \quad \quad \quad \quad \quad \quad \] \\
\[ xnext = \quad \quad \quad \quad \quad \quad \] \\
end

H.2 (2 points)

Given a nonlinear function

\[ f = 2x^{11} + 3x^4 + 2x + 8, \]

we want to use the bisection method to find a root with an initial search range [-2, 2]. After one iteration, what is the new search range?

\[ \text{ans} = \quad \quad \quad \quad \quad \quad \]