

A&CP w/matlab. 2PE-MMXVIII-JULY-2

P#1. Quizzes. The external file **quiz.m** is constructed with the score of 15 quizzes plus the total points allowed, e.g.: 1, 3, 2, 0, 2, 5, 13, 13, 5, 8, 1, 2, 3, 0, 4, 100, in which the last value is the maximum points allowed. Write a driver program which read quiz.m (e.g., **load**) and calls the function QZS which in turn utilizes the formula:

$$QZS = 10 \left[\frac{\text{points obtained}}{\text{maximum points}} \right]$$

The driver program reports the final quizzes. Write separately the QZS function too.

P#2 Movie rating. Below is a table of reviewers versus movie ratings. Review scale is 1 to 5. Find the average rating for each movie. For instance, the average review of movie number 1 (i.e., in column 1) is $(4+3+4)/3 = 3.7$. You can use the statement `[m,n]=size(argument)` for sizing the array.

reviewer					
	1	2	3	4	← movie
1	4	5	2	4	
2	3	5	4	5	
3	4	4	3	4	

Movies are: 'Hellboy', 'Pans Labyrinth', 'Call me by your name', and 'Crimson Peak'. The expected output:

```
The average of each movie :
      Movie  Average
      Hellboy      3.7
Pans Labyrinth      4.7
Call me by your name      3.0
      Crimson Peak      4.3
```

P#3 Series. Using the concept of accumulation, write a program to compute the series:

$$S = \sum_{i=0}^{N=10} i! = 0! + 1! + \dots + 10!$$

Hint: compute the factorial first and then accumulate each term. Can't use the **sum** nor **factorial** library functions.

P#4. Nested Loops. Implement matlab code for addition of all values in the series below. The total sum is printed.

$$5+5+5+5+5+4+4+4+4+3+3+3+2+2+1$$

Notice that the above data could be arranged in a triangular shape and nested loops could be used for a solution.

5 + 5 + 5 + 5 + 5 +	There may be different arrangement or 'shapes' to add these numbers.
+ 4 + 4 + 4 + 4 +	Addition is commutative, so different orders in the sequences may also be
+ 3 + 3 + 3 +	used. Finally, using mathematics, there may be some other ways to get
+ 2 + 2 +	these numbers and get the same result
+ 1	

P#5. Heat Transfer Problem & Your Debugging Skills

```
% HeatTransfer1D_V2.m
% Documentation
clc, clear
load toDATA.m
to=toDATA;
n=numel(to);  tol=1e-6;  iteration =0;

%% Initialize extremes, which are constant
tn(1)=to(1);  tn(n)=to(n);

%% Main loop computes interior knots
```

HERE INSERT (A) , (B) , (C) or (D)

```
%% Print the old and new (final) temperatures
fprintf('\n The old and new (final) temperatures are :\n');
for i=1:n
    fprintf(' %.1f %.1f \n',toDATA(i) tn(i));
end
fprintf('\n Iterations to obtain new temperatures is %d \n', iteration);
```

Analyze the PREVIOUS code by inserting (A), (B), (C) or (D) below. Write YES if the code is correct. There are not syntax errors. There may be more than one correct algorithm. The **max** function computes the largest element of an array.

(A)	(B)
<pre> %% Main loop computes interior knots maxerror=1.0; while maxerror>=tol for i=2:1:n-1 tn(i)=(to(i-1)+to(i+1))/2.0; error(i)= abs(to(i)-tn(i)); end % Calculate maximum error maxerror = error(2); for i=2:n-1 if error(i) >= maxerror maxerror = error(i); end end % Transfer 'new' to 'old' temp. for i=2:n-1 to(i)=tn(i); end iteration=iteration+1; end </pre>	<pre> %% Main loop computes interior knots maxerror=1.0; while maxerror>=tol for i=2:n-1 tn(i)=(to(i-1)+to(i+1))/2.0; error(i)= abs(to(i)-tn(i)); to(i)=tn(i); % 'new' to 'old' end % calculate maximum error maxerror = max(error); iteration=iteration+1; end </pre>
(C)	(D)
<pre> %% Main loop computes interior knots error=1.0; while max(error)>=tol i=2:n-1; tn(i)=(to(i-1)+to(i+1))/2.0; error= abs(to-tn); to=tn; iteration=iteration+1; end </pre>	<pre> %% Main loop computes interior knots error=1.0; i=1; while max(error)>=tol tn(i)=(to(i-1)+to(i+1))/2.0; error(i)= abs(to(i)-tn(i)); to(i)=tn(i); i=i+1; iteration=iteration+1; end </pre>