

CS1371 Fall13 Test 1 – Sept 20th, 2013 **VERSION A**

Name: KEY

GTID: KEY

Section: _____

- You will have 50 min for this exam
- This is a closed note/closed computer exam
- You are allowed one piece of paper for a crib sheet, front side only
- You are allowed to print off Appendix A to use as a reference during the exam

Question 1:

The following code is run in MATLAB:

```
str1 = 1371
str1 = num2str(str1)
A = length(str1)
B = class(str1 + 1)
```

```
arr1 = [6 3 8; 4 1 9; 3 1 0]
arr1(4,2) = 7
[C D] = size(arr1)
```

```
arr2 = [6 3 8; 4 1 9; 3 1 0]
arr2(arr2<4 | arr2>7) = arr2(arr2<4 | arr2>7) + 1
E = arr2
```

After the above script is run, what are the values of the following variables? If a line produces an error, answer ERROR for that variable and assume the rest of the code is still run.

Enter the values as they would be entered into MATLAB :

- vectors and arrays in [square brackets]
- strings in 'single quotes'
- logicals as true or false
- if the answer is error write ERROR

A = 4

B = 'double'

C = 4

D = 3

E = [6 4 9; 4 2 10; 4 2 1]

Question 2:

Function Name: cupcake

Inputs (3):

- (double) A 3x3 array containing the scores of the cupcakes from judge 1
- (double) A 3x3 array containing the scores of the cupcakes from judge 2
- (double) A 3x3 array containing the scores of the cupcakes from judge 3

Outputs (2):

- (double) The highest scoring total score for a cupcake
- (double) An array with the scores in each category from each judge added together

Function Description:

You are judging a cupcake competition that you also entered a cupcake in. During cupcake competitions, cupcakes are judged by 3 different judges on their flavor, presentation, and originality, as shown in the form below.

Judge 1:

	Cupcake 1	Cupcake 2	Cupcake 3
Flavor	[number	number	number;
Presentation	number	number	number;
Originality	number	number	number]

Judge 2:

	Cupcake 1	Cupcake 2	Cupcake 3
Flavor	[number	number	number;
Presentation	number	number	number;
Originality	number	number	number]

Judge 3:

	Cupcake 1	Cupcake 2	Cupcake 3
Flavor	[number	number	number;
Presentation	number	number	number;
Originality	number	number	number]

First, return the best score received by a cupcake in all three of the categories combined. This should be a single number. Finally, return an array containing added up totals for each cupcake in each category from all 3 judges.

Test Case:

```
arr1 = [2 1 3; 4 7 8; 3 4 5]
arr2 = [8 3 1; 7 7 3; 3 10 5]
arr3 = [3 1 2; 6 7 7; 1 4 5]
```

```
[bestScore, newArr] = cupcake(arr, vec)
bestScore ==> 44
newArr ==> [13 5 6; 17 21 18; 7 18 15];
```

function [highestScore totalScores] = cupcake(arr1, arr2, arr3)
totalScores = arr1 + arr2 + arr3 % Sums the cupcake scores for each judge
highestScore = cupcakeScores = sum(totalScores) % Finds the total sum for each cupcake
highestScore = max(cupcakeScores) % Finds highest total score

Question 3:

Function Name: newTV

Function Inputs: (3)

- (double) display quality/resolution the customer wants
- (double) maximum length of wall for the TV
- (double) maximum height of wall for the TV

Function Outputs: (2)

- (double) Model number of highest-rated TV
- (double) Cost of highest-rated TV

Function Description: Write a function 'newTV' that outputs the model number and cost of the highest-rated TV. The TV must have the required resolution and fit the wall whose length and height are provided. You must use the following helper functions in your solution. Consider them to be a part of your directory but their implementation is not required.

You are given two functions:

[<TV diagonal>] = findMaxDiag(<length of wall>, <height of wall>)
Returns max possible diagonal of TV

[<model number>, <cost> , <ratings>] = getTVInfo(<TV Diagonal>, <TV Resolution>)

Returns three different vectors with list of TV model number, cost and rating respectively that fit the required resolution and size.

```
function [bestmodel bestcost] = newTV(resolution, lengthTV, heightTV)
    diagonal = findMaxDiag(lengthTV, heightTV); % Find diagonal
    [models, costs, ratings] = getTVInfo(diagonal, resolution) % Get info
    [maximum ind] = max(ratings) % Gets the highest rating and
    % the index where it occurs
    bestmodel = models(ind); % Indexes the models and costs
    bestcost = costs(ind); % at that index
```