

CS112 Computation for the Sciences

Practice Exam 1

This exam is open book and open notes. There are 5 problems on the exam worth a total of 100 points. The number of points for each problem is shown below. **Try to do something on every problem, and show all of your work.**

Good luck!

Name _____

Problem 1 _____ (15/100 points)

Problem 2 _____ (20/100 points)

Problem 3 _____ (20/100 points)

Problem 4 _____ (20/100 points)

Problem 5 _____ (25/100 points)

TOTAL _____

Problem 1: Debugging (20 points)

The code below contains exactly *five* errors. For each error: (1) circle the error, (2) describe the nature of the error in your own words, and (3) write a modification of the code that “fixes” the error (there are multiple ways to fix each error – just make a change that allows the code to run without generating an error). You do not need to indicate the error message that MATLAB prints. **Please keep in mind that you will lose credit for missing an existing error, and for marking something as an error that is not a real error, so make your choices carefully!**

```
n = input('What''s your ''''favorite'''' number? ')
```

```
twiceYourFavoritePlusOne = 2n + 1
```

```
myFavorites = linspace(2,8,3)
```

```
yourFavorites = 2:3:6
```

```
myFavorites(yourFavorites)
```

```
yourFavorites(myFavorites) = 6
```

```
myFavoritesSquared = myFavorites * myFavorites
```

```
if (rem(n,2) = 0)
    disp('even')
else
    disp('odd')
end
```

```
ages4review = [19 7 4 2; 14 89 62; 2 3 2 12 ]
```

Problem 2: Simulating MATLAB (20 points)

Hand simulate the actions that MATLAB performs during execution of the following code. Your simulation work should clearly indicate what is printed after each statement that does not end with a semi-colon.

```
nums = [3 6 9; 2 4 6; 1 2 3]
```

```
mySums = sum(nums, 2)
```

```
nums(:,2) = mySums
```

```
nums(2:4,3:4) = 20
```

```
nums(nums > 17) = 1
```

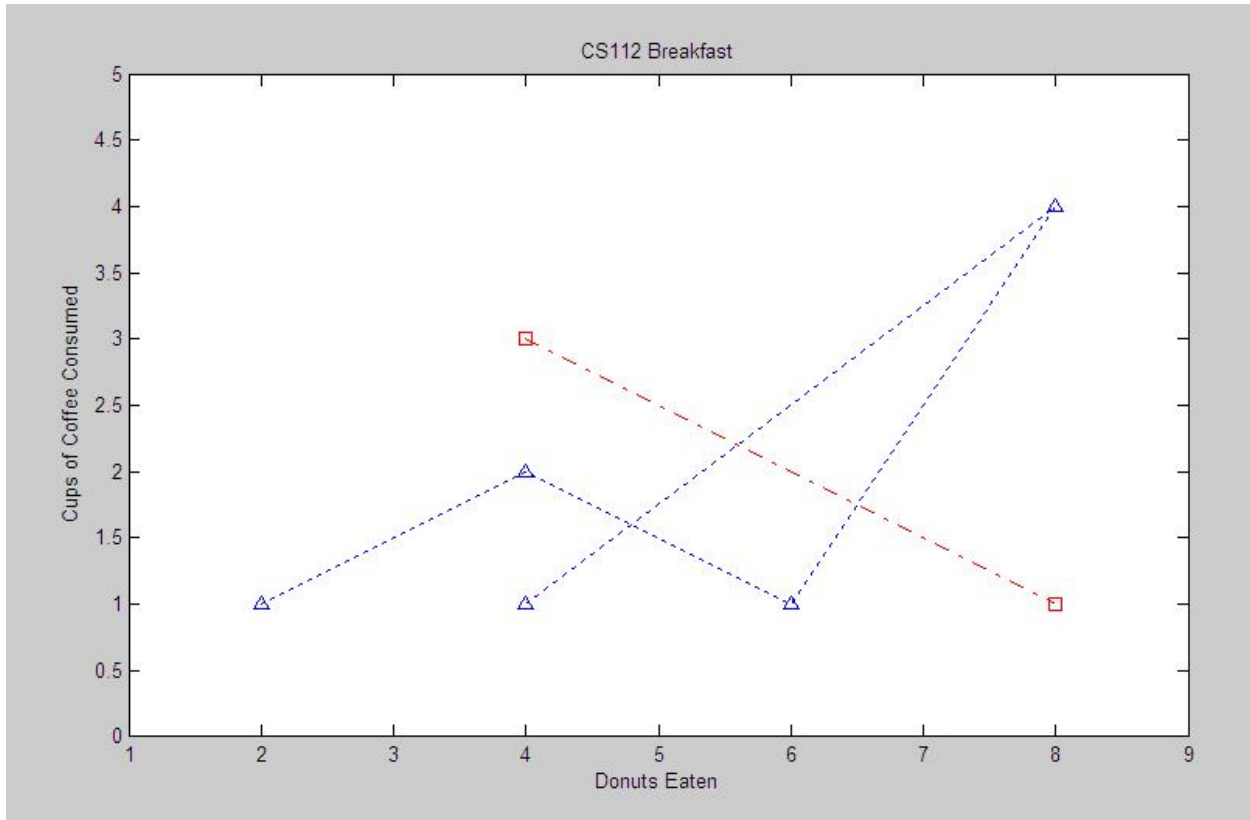
```
num = 9;
```

```
num + 2 * num / 3 : num / 3 : 2 * num
```

```
~9 ~= num & 3 + num < 13 == 1
```

Problem 3: Displaying Plots (20 points)

A MATLAB figure window is shown below. Write MATLAB code that would create the figure **exactly** as it is shown.



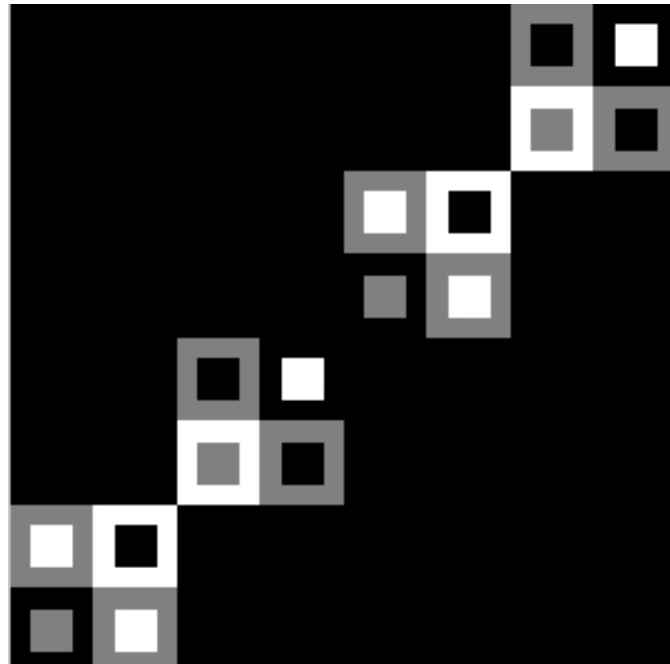
Write your MATLAB code here:

Problem 4: Creating Pictures (20 points)

You are given a MATLAB script called `makeBlock.m` that creates the 32x32 picture shown below on the left and stores it in a variable called `block`. You do not need to write the code to create `block`, you can simply use the `block` variable. Assume that the `block` matrix contains the values 0 (black), 0.5 (gray) and 1.0 (white).



`block`



`bigBlock`

Write MATLAB code to produce the 128x128 image (above right), using `block`. Store the image in a variable called `bigBlock`. Use comments to explain your strategy as you write your code.

Problem 5: Writing MATLAB Code (25 points)**Part a: Solving a Quadratic Equation (7 points)**

Given a quadratic equation of the form $ax^2 + bx + c = 0$, the roots of this equation are given by the quadratic formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Suppose the three coefficients a , b , c are stored in a 3-element vector named `cf`. For example, the vector `cf = [2 -3 1]` contains the coefficients of the equation $2x^2 - 3x + 1 = 0$. Write two MATLAB statements that assign the variables `x1` and `x2` to the two roots of the equation in terms of the `cf` vector and the formula above (do not refer to a , b and c in your expressions).

`x1 =`

`x2 =`

Part b: Calculating Vapor Pressure (7 points)

The variation of vapor pressure p of benzene with temperature T in the range from $0^\circ \leq T \leq 42^\circ$ (Centigrade) can be modeled with the equation:

$$\log_{10}p = b + (0.05223a)/T$$

where $a = 34.172$ and $b = 7.9622$ are material constants. Write a pair of MATLAB statements that create a sequence of temperatures from 0° to 42° , incrementing by 2° , and then determine the vapor pressure p for all of these temperatures. It is ok to substitute the numerical values of a and b directly in your expression for p .

Hint: If $\log_{10}m = n$, then $m = 10^n$

Part c: Computing Class Averages (11 points)

Wendy Wellesley has a theory that computer science professors become easier graders as they get older. She has collected statistics from five introductory courses taught at Wellesley College by five different professors into the table shown below.¹

Course	A	B	C	NC	Prof's age
CS100	8	7	1	0	55
CS110	25	57	33	2	40
CS111	9	12	4	1	33
CS112	12	2	0	0	55
CS199	3	6	3	0	38

Suppose the following statements are executed to create variables that store this information (note that `ages` is a column vector):

```
gradesGiven = [ 8  7  1  0;
                25 57 33  2;
                9 12  4  1;
                12 2  0  0;
                3  6  3  0 ];
```

```
ages = [55; 40; 33; 55; 38];
courses = {'CS100' 'CS110' 'CS111' 'CS112' 'CS199'};
grades = {'A' 'B' 'C' 'NC'};
```

(1) In order to test her theory, Wendy would like to calculate the weighted average of grades assigned in each class where an 'A' has a weight of 4, a 'B' has a weight of 3, a 'C' has a weight of 2, and an 'NC' has a weight of 0. Write a single MATLAB statement that assigns to the variable `classAverages`, a column vector containing the weighted average of the grades given in each class.

(2) Write a single MATLAB expression that has the value "true" if and only if all class(es) with the highest grade point average are also the class(es) taught by the oldest professor(s). Your expression may use the variable `classAverages` assigned in (1).

¹ These data are, of course, totally factious

CS112 Computation for the Sciences

Practice Exam 1 Solutions

Problem 1: Debugging

The following five errors occur in this code, and there are multiple ways to fix each one:

(1) `twiceYourFavoritePlusOne = 2n + 1`

The multiplication operator in the expression `2n` is missing – it should be written as `2*n`.

(2) `myFavorites (yourFavorites)`

The variable `myFavorites`, created by the expression `linspace(2, 8, 3)` is assigned to the vector `[2 5 8]`. The variable `yourFavorites`, created by the expression `2:3:6` is assigned to the vector `[2 5]`. When the vector `[2 5]` is supplied as indices for `myFavorites`, the index 5 is out of bounds for the `myFavorites` vector, which has only three elements. The following is one way to fix this error:

`myFavorites (yourFavorites(1))`

(3) `myFavoritesSquared = myFavorites * myFavorites`

This statement uses the matrix multiplication operator, but we cannot multiply two row vectors in this way, so the `*` operator should be replaced with the element-by-element multiplication operator, `.*`

(4) `if (rem(n,2) = 0)`

When comparing two values in a conditional expression, the `==` operator must be used to test for equality, as in `(rem(n,2) == 0)`.

(5) `ages4review = [19 7 4 2; 14 89 62; 2 3 2 12]`

The first and third rows of this matrix have 4 elements, whereas the middle row has only 3 elements, so there are inconsistent dimensions in the rows of the matrix. A fourth number should be added to the middle row.

Problem 2: Simulating MATLAB

The following printout is produced by the execution of these statements:

```

nums =
     3     6     9
     2     4     6
     1     2     3
mySums =
    18
    12
     6
nums =
     3    18     9
     2    12     6
     1     6     3
nums =
     3    18     9     0
     2    12    20    20
     1     6    20    20
     0     0    20    20

```

```

nums =
    3     1     9     0
    2    12     1     1
    1     6     1     1
    0     0     1     1
ans =
    15    18
ans =
    1

```

In the case of the following expression:

```
num + 2 * num / 3 : num / 3 : 2 * num
```

the operations are performed in the order indicated by the parentheses in the expression below, which also has the variable `num` replaced with its value, 9:

```

(9 + ((2 * 9) / 3)) : (9 / 3) : (2 * 9)
15:3:18
[15 18]

```

For the last expression: `~9 ~= num & 3 + num < 13 == 1`

the following parenthesized expression indicates the order in which operations are performed:

```

((~9) ~= 9) & (((3 + 9) < 13) == 1)
(0 ~= 9) & ((12 < 13) == 1)
1 & (1 == 1)
1 & 1
1

```

Problem 3: Displaying Plots

The following code produces the figure shown:

```

xcoords = [2 4 6 8 4];
ycoords = [1 2 1 4 1];
plot(xcoords, ycoords, 'b^:');
hold on
plot([4 8], [3 1], 'm-.s')
hold off
axis([1 9 0 5])
xlabel('Donuts Eaten')
ylabel('Cups of Coffee Consumed')
title('CS112 Breakfast')

```

Problem 4: Creating Pictures

The following code constructs the **bigBlock** image:

```

% create 128x128 image with black background
bigBlock = zeros(128,128);
% copy block as is onto bottom left corner
bigBlock(97:128,1:32) = block;
% copy block with brightness flipped to mid-lower-left
bigBlock(65:96,33:64) = 1 - block;

```

```
% copy block as is to mid-upper-right
bigBlock(33:64,65:96) = block;
% copy block with brightness flipped to upper right corner
bigBlock(1:32,97:128) = 1 - block;
```

Problem 5: Writing MATLAB Code

Part a: Solving a Quadratic Equation

```
x1 = (-cf(2) + sqrt(cf(2)^2 - 4*cf(1)*cf(3)))/(2*cf(1))
x2 = (-cf(2) - sqrt(cf(2)^2 - 4*cf(1)*cf(3)))/(2*cf(1))
```

Part b: Calculating Vapor Pressure

```
T = 0:2:42
P = 10^(7.9622 + 0.05223*34.172/T)
```

Part c: Calculating Class Averages

(1) The following code statements show two ways to construct classAverages:

```
classAverages = (4*gradesGiven(:,1) + 3*gradesGiven(:,2) + ...
                2*gradesGiven(:,3)) ./ sum(gradesGiven,2)
classAverages = (gradesGiven*[4; 3; 2; 0]) ./ sum(gradesGiven,2)
```

(2) `sum((max(classAverages) == classAverages) & (max(ages) == ages)) > 0`