**Class 25 Group Practice Problem Solutions**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Partner Names: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Work in MATLAB. Cut and paste your code solutions into this document.

1) When would you use **disp** instead of **fprintf**? When would you use **fprintf** instead of **disp**?

The disp function is used when no formatting is required. It is also easier to print vectors and matrices using disp. The fprintf function is used for formatted output.

2) When do you pass just the prompt to the **input** function and when do you also pass ‘s’?

You only pass ‘s’ when a character or character vector input is desired.

3) Write an **input** statement that will prompt the user for the name of a chemical element as a character vector. Then, find the length of the character vector.

elemname = input('Enter a chemical element: ', 's');

length(elemname)

4) Write an **input** statement that will prompt the user for a real number, and store it in a variable. Then, use the **fprintf** function to print the value of this variable using 2 decimal places.

realnum = input('Enter a real number: ');

fprintf('The number is %.2f\n', realnum)

5) The atomic weight is the weight of a mole of atoms of a chemical element. For example, the atomic weight of oxygen is 15.9994 and the atomic weight of hydrogen is 1.0079. Write a script that will calculate and print the molecular weight of hydrogen peroxide, which consists of two atoms of hydrogen and two atoms of oxygen. Assign the values for the atomic weights to variables. Include comments in the script. The output should be in this format:

The molecular weight is 34.01

% Calculates the molecular weight of hydrogen peroxide

% Initialize the atomic weights for oxygen and hydrogen

atWtOxygen = 15.9994;

atWtHydrogen = 1.0079;

% Hydrogen peroxide is 2 atoms of hydrogen and 2 of oxygen

molWtHydPer = 2\*atWtHydrogen + 2 \* atWtOxygen;

fprintf('The molecular weight is %.2f\n', molWtHydPer)

6) Write a script that will prompt the user for the results of a classification model (the numbers of true positives, true negatives, false positives, and false negatives). Calculate and print the accuracy in a nice sentence format. For example:

Enter the number of true positives: 200

Enter the number of true negatives: 30

Enter the number of false positives: 10

Enter the number of false negatives: 12

The accuracy is 91.27%

The accuracy is the sum of the correctly classified (TP + TN) divided by the total number (sum of all).

TP = input('Enter the number of true positives: ');

TN = input('Enter the number of true negatives: ');

FP = input('Enter the number of false positives: ');

FN = input('Enter the number of false negatives: ');

accuracy = (TP + TN) / (TP + TN + FP + FN);

fprintf('The accuracy is %.2f%%\n', accuracy\*100)

7) Write a script that would create a 5x5 matrix of random integers in the range from 0 to 50. From this, create another matrix variable, which is the “middle” 3x3 part of the original matrix.

mat = randi([0 50], 5)

newmat = mat(2:4, 2:4)

8) Write a script that will plot the temperature at noon, 1pm, 2pm, 3pm, and 4pm today. Create an x vector that stores the hours, and a y vector that stores the temperatures, and then plot the data with appropriate annotations on the graph. Create your own data.

x = 0:4;

y = [55 61 63 62 59];

plot(x,y,'\*')

xlabel('Time')

ylabel('Temperature')

title('Temps this afternoon (0 == noon)')

9) Atmospheric properties such as temperature, air density, and air pressure are important in aviation. Assume that you have a matrix “alttemps” that stores temperatures in degrees Kelvin at various altitudes. The altitudes are in the first column and the temperatures in the second. For example, it may look like this:

1000 288

2000 281

3000 269

Write a script that will separate it into vectors, and then plot the data with appropriate axis labels and a title.

altitudes = alttemps(:,1);

temps = alttemps(:,2);

plot(altitudes,temps,'k\*')

xlabel('Altitudes')

ylabel('Temperatures')

title('Atmospheric Data')

10) Write a script *lumin* that will calculate and print the luminosity L of a star in Watts. The luminosity L is given by L = 4 π d2 b where d is the distance from the sun in meters and b is the brightness in Watts/meters2. Here is an example of executing the script:

>> lumin

This script will calculate the luminosity of a star.

When prompted, enter the star's distance from the sun

in meters, and its brightness in W/meters squared.

Enter the distance: 1.26e12

Enter the brightness: 2e-17

The luminosity of this star is 399007399.75 watts

lumin.m

% Calculates the luminosity of a star

disp('This script will calculate the luminosity of a star.')

disp('When prompted, enter the star''s distance from the sun')

fprintf(' in meters, and its brightness in W/meters squared.\n\n')

d = input('Enter the distance: ');

b = input('Enter the brightness: ');

L = 4\*pi\*d^2\*b;

fprintf('The luminosity of this star is %.2f watts\n', L)

11) Consider the following script:

yesorno = input('Are you an engineering student: ', 's');

yesorno = yesorno(1);

if yesorno == 'y' || 'Y'

disp('Engineers rock!')

elseif yesorno == 'n' || 'N'

disp('Yay non-engineers!')

else

disp('Ummmm....')

end

What would the script print if the user enters ‘y’?

Engineers rock!

What would the script print if the user enters ‘N’?

Engineers rock!

What would the script print if the user enters ‘x’?

Engineers rock!

Yes, it is true, Engineers rock! This will always be printed. ***EXPLAIN WHY.***

Since ‘Y’ is not zero, it is a way of representing **true** – so it doesn’t matter what character is stored in *yesorno*. This is probably a logical error. The if statement should say:

if yesorno == 'y' || yesorno == 'Y'

Also, what is the purpose of the second statement?

This is in case the user enters words, such as ‘yes’ or ‘no’

12) When would you use just an **if** statement and not an **if-else**?

In any case in which if the condition is false, no action is required.

13) Re-write the following nested **if-else** statement as four separate **if** statements (no **else**) that together will accomplish exactly the same thing. Assume that a variable *n* has been initialized.

if n >= 3

disp('a')

elseif n >= 1

disp('b')

elseif n == 0

disp('c')

else

disp('d')

end

if n >= 3

disp('a')

end

if n == 1 || n == 2

disp('b')

end

if n == 0

disp('c')

end

if n < 0

disp('d')

end

14) Write a script that will prompt the user for a character, (assume that the user enters just one character) and then print whether or not it is a letter of the alphabet. Do NOT use ASCII equivalents; just use characters in the relational expressions. Also, do this two ways: with and without the **isletter** function.

mychar = input('Enter a character: ', 's');

mychar = mychar(1);

if isletter(mychar)

fprintf('It IS a letter!\n')

else

fprintf('It is NOT a letter :-(\n')

end

if mychar >= 'a'&& mychar <= 'z' ...

|| mychar >= 'A' && mychar <= 'Z'

fprintf('It IS a letter!\n')

else

fprintf('It is NOT a letter :-(\n')

end

15) Write a script that presents the user with a menu of choices, using the **menu** function. You decide what the choices are! Do something based on the menu, two ways: using a **switch** statement, and using a nested **if-else**.

16) Write your own script to accomplish something. It must prompt the user for at least two inputs (one character or character vector and one number), and then either calculate and print a result based on the inputs, or plot. Please post this to the Discussion Board.