**BE2080 Week#3**

**Submission:**

**Please copy your Matlab program (scripts or functions based on the requirements) and paste into your word document for submission.**

1) (10 points) Write a script that will prompt the user for a temperature in degrees Celsius, and then an ‘F’ for Fahrenheit or ‘K’ for Kelvin. The script will print the corresponding temperature in the scale specified by the user. For example, the output might look like this:

>>Enter the temp in degrees C: 29.3

>>Do you want K or F? F

>>The temp in degrees F is 84.7

The format of the output should be exactly as specified above. The conversions are:





%{

This script prompts user for temperature in degrees Celsius

and asks if user wants temperature converted to Fahrenheit or

Kelvin. Script then prints out corresponding temperature in

scale specified by user.

%}

temp\_C = input('Enter the temp in degrees C: ');

F\_or\_K = input('Do you want K or F? ', 's');

if F\_or\_K == 'K'

temp\_K = temp\_C + 273.15;

fprintf('The temp in degrees K is %.1f.\n', temp\_K)

elseif F\_or\_K == 'F'

temp\_F = (9/5)\*temp\_C + 32;

fprintf('The temp in degrees F is %.1f.\n', temp\_F)

else

F\_or\_K = input('Do you want K or F? ', ‘s’);

end

2) (10 points) Write a **for** loop that will print the column of real numbers from 1.0 to 2.0 in a step of 0.2 in exactly the following format:

Number 1 element is: 1.0

Number 2 element is: 1.2

Number 3 element is: 1.4

Number 4 element is: 1.6

Number 5 element is: 1.8

Number 6 element is: 2.0

%{

This prints a column of real numbers from 1.0 to 2.0

in steps of 0.2.

%}

count = 1;

for i = 1:0.2:2

fprintf('Number %d element is: %.1f\n', count, i);

count = count + 1;

end

3) (10 points) Write a function *PrintStars(n)* that has one integer input argument *n.* This function prints out n lines of the symbol ‘\*’ with 1 to n ‘\*’ in each line. Do this using a **for** loop.

For example, calling the function will look like this:

*>> PrintStars(5)*

*\**

*\*\**

*\*\*\**

*\*\*\*\**

*\*\*\*\*\**

function [] = PrintStars(rows)

%{

This function prints out n lines of star symbol in

each line using a for loop.

%}

% [] = PrintStars(integerInput)

for i = 1:rows

for j = 1:i

fprintf('\*')

end

fprintf('\n')

end

end

4) (10 points) Please convert the for-loop in the following codes to a while loop. Both versions (for-loop and while-loop) should produce the same result.

x=0;

y=randi(100,1,5);

for i = 1:5

x=x+y(i);

end

disp(x);

x = 0;

y = randi(100, 1, 5)

i = 1;

while i <= 5

x = x + y(i)

i = i + 1;

end

disp(x);

**5**) (10 points) Please convert the CalBMI script you wrote last week to a function. The function should have the weight [lb] and height [in] as the two input arguments and the BMI as the output argument.

An example execution of this function:

>> BMI = CalBMI(170,72)

BMI =

23.1

function [BMI] = CalBMI(weight, height)

%Calculate BMI based on person's height and weight.

if weight < 0 || height < 0

disp('BMI cannot be calculated with negative measures.')

else

BMI = weight/(height\*height) \* 703;

end

**6**) (10 points) Write a function *areaMenu()*  that has no input/output arguments. The function will first print a “menu” list consisting of “sphere*”*, “circle*”*, and “rectangle”. It prompts the user to choose one, and then prompts the user for the appropriate quantities (e.g., the radius of the circle) and then prints its area. If the user enters an invalid choice, the script simply prints an error message. Here are two examples of running it (units are assumed to be inches).

*>> areaMenu*

Menu

1. Sphere

2. Circle

3. Rectangle

Please choose one: 2

Enter the radius of the circle: 4.1

The area is 52.81

*>> areaMenu*

Menu

1. Sphere

2. Circle

3. Rectangle

Please choose one: 3

Enter the length: 4

Enter the width: 6

The area is 24.00

function [] = areaMenu()

%{

Prints menu which prompts user to choose a shape and enter

approprate quantieies and then prints area.

%}

menu\_choice = input('Menu\n1. Sphere\n2. Circle\n3. Rectangle\nPlease choose one: ');

if menu\_choice == 1

r = input('Enter the radius: ');

area = 4 \* pi \* r^2;

fprintf('The area is %.2f\n', area);

elseif menu\_choice == 2

r = input('Enter the radius: ');

area = pi \* r^2;

fprintf('The area is %.2f\n', area);

elseif menu\_choice == 3

length = input('Enter the length: ');

width = input('Enter the width: ');

area = length \* width;

fprintf('The area is %.2f\n', area);

else

disp('That was an incorrect menu choice.')

end

OR

\*This is better.

function [] = areaMenu()

%{

Prints menu which prompts user to choose a shape and enter

approprate quantieies and then prints area.

%}

menu\_choice = input('Menu\n1. Sphere\n2. Circle\n3. Rectangle\nPlease choose one: ');

if menu\_choice == 1

r = input('Enter the radius: ');

area = 4 \* pi \* r^2;

elseif menu\_choice == 2

r = input('Enter the radius: ');

area = pi \* r^2;

elseif menu\_choice == 3

length = input('Enter the length: ');

width = input('Enter the width: ');

area = length \* width;

else

disp('That was an incorrect menu choice.')

return

end

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

end

7) (10 points) A machine cuts N pieces of a pipe. After each cut, each piece of pipe is weighed and its length is measured; these 2 values are then stored in a file called *pipe.dat* (first the weight and then the length on each line of the file). Ignoring units, the weight is supposed to be less than or equal to 2.3, and the length is supposed to be greater than 10.3.

Write a function to read in the data file ‘pipe.dat’ and count how many rejects there are. A reject is any piece of pipe that has an invalid weight OR length. The function shall have one input argument, which is the name of the data file. The function shall have one output argument, which shall be the number of rejects. Please implement this function using for-loops.

An example call of the function should look like this (the exact value may be different depending on data):

>> Num = CountRejects('pipe.dat')

Num =

3

function [Num] = CountRejects(file)

%{

This function counts how many rejects of pipes there are listed

in a file. Rejects have weights greater than 2.3 or lengths less

than or equal to 10.3.

%}

%[NumberofRejects] = CountRejects(FileName)

%FileName = 'pipe.txt' in this case \*When I downloaded the file, I noticed it was a .txt file.

load(file); \*should be pipe = importdata(file) so there won’t be an error if incorrect filename is inputted

weights = pipe(:, 1);

lengths = pipe(:, 2);

Num = 0;

for i = 1:size(pipe, 1)

if weights(i) > 2.3 || lengths(i) <= 10.3

Num = Num + 1;

end

end

end

8) (10 points) Write a script that will ask the user to input a number between 10 and 20, inclusive. The program will check the actual user input. If the user input is not between 10 and 20, the program will keep asking for the input until 8he user provides a correct one. Then the program will print out the number.

%{

This script asks user to input a # between 10 and 20. Program

checks input and keep asking for new input if the input is not

between 10 and 20. Then the program will print out the number.

%}

num = input('Enter a number between 10 and 20, inclusive: ');

while num < 10 || num > 20

fprintf('Your number (%d) is not between 10 and 20.\n', num)

num = input('Enter a number between 10 and 20, inclusive: ');

end

fprintf('Your number is %d.\n', num)

9) (10 points) Write a script *echoletters* that will prompt the user for letters of the alphabet and echo-print them until the user enters a character that is not a letter of the alphabet. At that point, the script will print the nonletter, and a count of how many letters were entered. Here are examples of running this script:

*>> echoletters*

Enter a letter: T

Thanks, you entered a T

Enter a letter: a

Thanks, you entered a a

Enter a letter: 8

8 is not a letter

You entered 2 letters

*>> echoletters*

Enter a letter: !

! is not a letter

You entered 0 letters

The format must be exactly as shown above.

%{

This script prompts user for letters of alphabet and echo-prints

them until user enters a character that isn't a letter of the

alphabet. The script will then print the nonletter and a count

of how many letters were entered.

%}

alphabet = {'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'};

letter = input('Enter a letter: ', 's');

count = 0;

while ismember(letter, alphabet)

count = count + 1;

fprintf('Thanks, you entered a %s\n', letter);

letter = input('Enter a letter: ', 's');

end

fprintf('%s is not a letter\nYou entered %d letters\n', letter, count)

OR

\*This is better. Or could do if (letter >= ‘A’ && letter <= ‘Z’ || (letter >= ‘a’ && letter <= ‘z’)

%{

This script prompts user for letters of alphabet and echo-prints

them until user enters a character that isn't a letter of the

alphabet. The script will then print the nonletter and a count

of how many letters were entered.

%}

letter = input('Enter a letter: ', 's');

count = 0;

while isletter(letter)

count = count + 1;

fprintf('Thanks, you entered a %s\n', letter);

letter = input('Enter a letter: ', 's');

end

fprintf('%c is not a letter\nYou entered %d letters\n', letter, count)

**10)** (10 points) Write a function that will receive as an input argument a number of kilometers (K). The function will convert the kilometers to miles and to U.S. nautical miles, and return both results. The conversions are: 1K = 0.621 miles and 1 US nautical mile = 1.852 K.

If your function is named as KConv, it should produce the following results in Matlab:

>> [Miles, NautMiles] = KConv(1)

Miles =

0.6210

NautMiles =

0.5400

function [Miles, NautMiles] = KConv(km)

%{

This function converts kilometers entered to miles and

U.S. nautical miles.

%}

Miles = 0.621 \* km;

NautMiles = km / 1.852;

end