% Clair Cunningham PSWC-01 Week 9 In\_Class-Exercises

%% In-Class Exercise #1

% Use relational operators to create a variable which contains the

% True/False answer to the questions

clc; clear all; close all;

%Is 2>1?

L = 2>1;

if L == 1

disp('2>1 True');

else

disp('2>1 False');

end

% Are both 2>1 and 3<3 true

J = 2>1 & 3<3;

if J == 1

disp('2>1 And 3<3 True');

else

disp('2>1 And 3<3 False');

end

% Is one, 2>1 or 3<3, true

K = 2>1 | 3<3;

if K == 1

disp('2>1 Or 3<3 True');

else

disp('2>1 True And 3<3 False');

end

%% In-Class Exercise #2

% Use relation operators to modify the array a = [1:10];

a = [1:10];

L = a < 5;

newa = L.\*a;

%% In-Class Exercise #3

% Plot only the positive part of the sin(x) curve from x=0 to 20

x = [0:0.1:20];

y = sin(x);

% Filter y array using relation operators

y = y.\*(y > 0);

% Plot positive half of sin(x)

plot(x,y)

xlabel('x'); ylabel('Positive half of sin(x)');

title('Filtered sin curve')

grid on

%%In-Class Exercise #4

clear all; close all; clc

x = input('Please enter a value for x :');

if x>= 0 & x<=3

g = 2\*x;

msg = ['g(x) = ' num2str(g)];

elseif x>3 & x <=10

g = (x-3)^2 + 6;

msg = ['g(x) = ' num2str(g)];

else

msg = ['X is out of Range.'];

end

msgbox(msg);

%% In-Class Exercise #5

% Evaluate a sum using both a loop and arrays ("vectorization")

%Using for loop

%initialize variables

tic;

loopsum=0;

%for loop to evaluate sum

for i = 1:1000000

term = 3\*sin(i\*pi/12)^2;

loopsum = loopsum+term;

end

%display sum & elapsed time

looptime=toc;

msg = ['Summation using loop = ' num2str(loopsum)];

disp(msg);

msg = ['Loop method elapsed time = ' num2str(looptime)];

disp(msg);

%Using vectorization

%initialize variables

tic;

vectorsum = 0;

%evaluate sum

i = [1:1000000];

terms = 3\*sin(i\*pi/12).^2;

arraysum = sum(terms);

%display sum & elapsed time

arraytime=toc;

msg = ['Summation using vectorization = ' num2str(arraysum)];

disp(msg);

msg = ['Array method elapsed time = ' num2str(arraytime)];

disp(msg);

%% In-Class Exercise #6

% Finds the first integer for which the factorial is 100 digits long

%initiaze varaible

n = 0;

factn = 1;

%Loop to run until a hundred digits has been reached.

while (factn<=10^99)

n = n+1;

factn = factorial(n);

end

msg = ['The first integer where its factorial is in the hundreds is ' num2str(n)];

disp(msg);

msg = ['The factorial is ' num2str(factn)];

disp(msg);

%% In-Class Exercise #7

% Use a for loop to find:

% max value, min value, sum, product, element closest to zero

%total number of elements

%initialize variables starting using i = 1 as base start value

y(1) = sin(1);

max = sin(1);

min = sin(1);

closest = abs(sin(1));

sum = sin(1);

prod = sin(1);

numneg = 0;

% One for loop to do everything starting from i = 2

for i = 2:100

y(i) = sin(i);

%Is the current y value larger than all past values

if y(i)>max

max = sin(i);

%Is the current y value smaller than all past values

elseif y(i)<min;

min = sin(i);

end

%Is the current y value the closest to 0

if abs(y(i))<closest

closest = sin(i);

end

% Sum of all y

sum = sum + y(i);

% Product of all y

prod = prod\*y(i);

% Total number of negative values;

if y(i)<0

numneg = numneg+1;

end

end

msgMax = ['Max value in the array ' num2str(max)];

msgMin = ['Min value in the array ' num2str(min)];

msgSum = ['Sum of values in the array ' num2str(sum)];

msgProd = ['Product of values in the array ' num2str(prod)];

msgElm = ['Element closest to 0 in the array ' num2str(closest)];

msgNum = ['Total number of negative elements in the array ' num2str(numneg)];

disp(msgMax);

disp(msgMin);

disp(msgSum);

disp(msgProd);

disp(msgElm);

disp(msgNum);

%% In-Class Exercise #8

% Use the array created in exercise #7 to repeat the exercise done in Vba

% to compute mean and standard deviation of a data set.

clc; clear all; close all

%Initialize variables

sum = 0;

sum2 = 0;

n = 0;

%For loop to find the sum of y and y^2

for i = 1:100

n = n+1;

y(i) = sin(i);

y2(i) = sin(i)^2;

sum = sum + y(i);

sum2 = sum2 + y2(i);

end

% Calculate the values of mean and standard deviation from the sums

% obtained

mean = sum/n;

std = sqrt((sum2-(sum^2)/n)/(n-1));

msgMean = ['The mean of the array of sin(i) is ' num2str(mean)];

msgStd = ['The standard deviation of the array is ' num2str(std)];

disp(msgMean);

disp(msgStd);

%% In-Class Exercise #9

% Evaluate the ratio A/B where A and B are double sums

clc;clear all; close all;

%Obtain the number of iterations;

n = input('Please enter the number of iterations ');

%Initialize variables

A = 0;

B = 0;

%For a Double loop to calculate sum A

for j = 1:n

for i = 1:j

A = A + i\*j^2;

end

%Calculates sum B using only the first for loop

B = B + j^3 + j^4;

end

ratioAB = A/B;

msgratio = ['The ratio of A/B is ' num2str(ratioAB) '.'];

disp(msgratio);

% The ratio comes out to always be 0.5 despite any change in the sums.

%% In-Class Exercise #10

% Create an array of the Fibbonacci numbers which are les than 1000

% Output the number of terms required.

clc; clear all; close all;

f(1) = 0; f(2) = 1;

i = 2;

nextf = 1;

while nextf<1000

f(i+1) = f(i)+f(i-1);

nextf = f(i+1) + f(i);

i = i+1;

end

msgterm = ['The total number of terms required is ' num2str(i);];

disp(msgterm);

disp(f);