CSC781M Assignment #2
De La Salle University
by: Eric John G. Emberda
October 7, 2018

DESCRIPTION:
This program demonstrates basic Octave programming. To run this program, type the following in the Octave Command Window and press enter:

run("/path/to/the/.m_file")

CONTENTS:
PART I Generating signals and plotting them

1. The following code generates a series of values starting from -100 to 100, in increments of 4. The values will be stored in a variable x. . There are three (3) parameters. First is the starting value (-100), the middle (4) is the interval, and the last (100) is the ending value.

x=[-100:4:100]

2. The following code reverses the values of x. Thus, the value of y will be 100 to -100 with an interval of 4.

y=flip(x)

3. The following code performs an element-by-element multiplication of x and y and stores it in a variable z.

*z=x.*y*

4. The following demonstrates how to plot x, y, and z with graph on top of the other. The subplot function divides the current figure into a specified number of rows and columns. The title function adds a label to the particular plot.

```
subplot(3,1,1)
plot(x,"r")
title("X")
subplot(3,1,2)
plot(y,"g")
title("Y")
subplot(3,1,3)
plot(z,"b")
title("Z")
axis("tight");
```

PART II Plotting functions and finding the absolute maximum and absolute minimum in Octave

1. The following plots the function $y(x) = x^3 - 5x^2 - 4x + 20$ for values of x ranging from -5 to 5, in increments of 0.5.

```
x=[-5:0.5:5];
y=x.^3-5.*x.^2-4.*x+20
```

2. From the graph of y(x), we use the *find()* function to determine the values of x where particular values occur (e.g. absolute maximum and absolute minimum values).

```
figure(2)
plot(y,"g")
max_index=find(y==max(y))
min_index=find(y==min(y))
zero_index=find(y==0)
```

3. The statement *zero_index=find(y==0)* above is how we automatically determine the zeros of y(x).

PART III The following code asks the user for a numeric value N, and returns the first N elements of a Fibonacci series.

```
N=input('Enter a value N: ');
fib = ones (1, N);
for i = 3:N
  fib(i) = fib(i-1) + fib(i-2);
endfor
disp(fib);
```