PRACTICA DE LABORATORIO 1 - MATLAB

Ej.1:

En la línea de comandos, ejecute las siguientes instrucciones:

a = 5

b = a ^ 2

x = 0:1:9

y = x .^ 2

z = x \* 2

c = 10:2:20

d = c - 1

clear

sin (pi / 4)

x = 0:0.01:2 \* pi

y = sin (x)

plot (x)

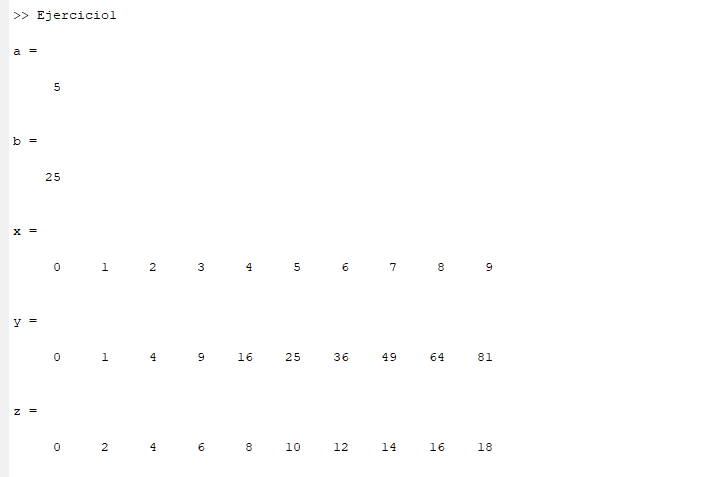
plot (y)

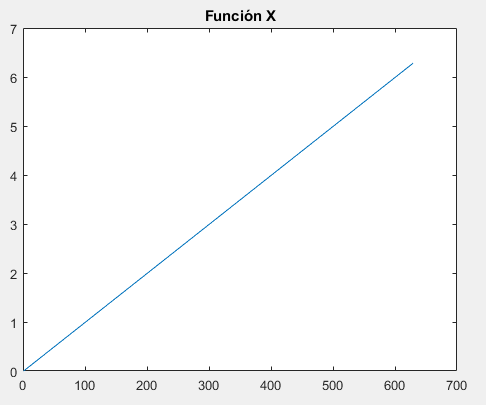
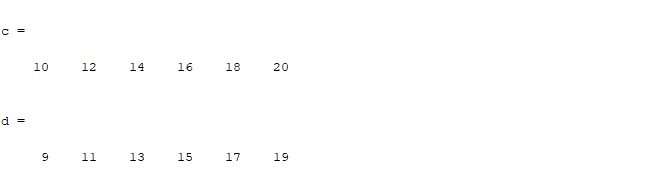
plot (x, y)

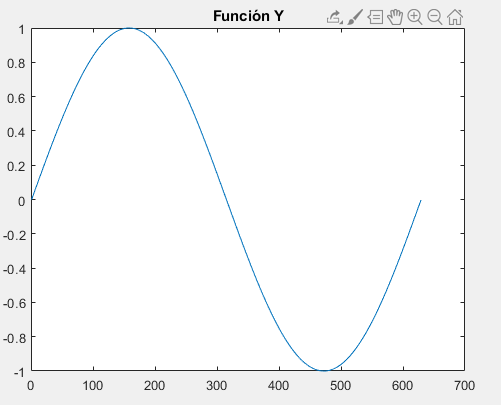
z = cos (x)

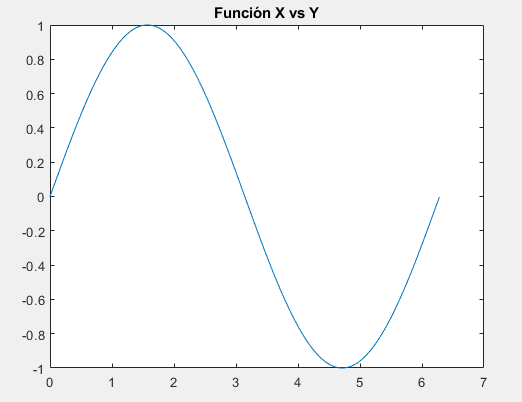
plot (x, y, 'r', x, z, 'g')

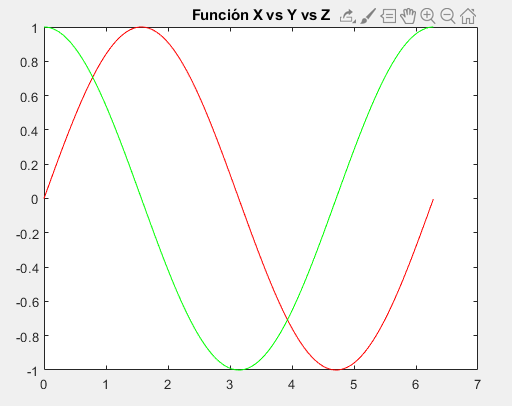
Memoria:











Ej.2:

Cree un archivo con el nombre facto1.m que contenga el siguiente código:

function f = facto1 (N)

f = 1;

for n = 2:N

f = f \* n;

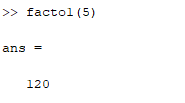
end

Ejecute la función anterior escribiendo su nombre en la línea de comandos. Si

no se encuentra la función, utilice el comando cd para llegar al directorio dónde se

halle el archivo creado.

Memoria:



Ej.3:

Cree un archivo con el nombre facto2.m que contenga el siguiente código:

function f = facto2 (N)

if N == 0

f = 1;

else

f = N \* facto2 (N - 1);

end

Ejecute la función anterior.

Memoria:



Ej.4:

Ejecute los siguientes comandos:

clear

a = [1 2 3 4 5]

b = [1 2; 3 4; 5 6]

c = 1:20

length (a)

size (b)

size (b, 1)

size (b, 2)

rem (c, 4)

¿Qué cree usted que devuelven las funciones length y size? ¿Y la función rem?

Si no lo sabe, ejecute los siguientes comandos:

help length

help size

help rem

Memoria:

La función length devuelve el número de elementos del vector a

La función size en el caso: size (b) nos devuelve dos valores 3 y 2 (Número de filas y número de columnas).

En el caso de size (b, 1) y size (b, 2) nos devuelve número de filas y columnas (en ese orden)

La función rem devuelve el módulo en base 4 (rem (c, 4)) de cada uno de los elementos de la matriz c.

Ej.5:

Existen una serie de funciones nativas que permiten trabajar con imágenes digitales

de forma sencilla. Estas son:

- imread: para leer una imagen desde un archivo.

- imshow: para mostrar una imagen previamente leída en pantalla.

- imwrite: para escribir una imagen a un archivo.

Utilice estas funciones para visualizar en pantalla algunas de las imágenes suministradas en el paquete de laboratorio (cada banda está separada en un archivo

PNG monocromo independiente).

Memoria:

- imread: para leer una imagen desde un archivo.

- imshow: para mostrar una imagen previamente leída en pantalla.

- imwrite: para escribir una imagen a un archivo.

Ej.6:

Cree un archivo con el nombre qsort.m que ordene ascendentemente un vector

utilizando el algoritmo Quicksort. Para ordenar ascendentemente una secuencia *x*

utilizando dicho algoritmo, es necesario seguir un proceso recursivo:

El caso base de la recursión es aquel en el que la longitud de *x* es cero, por

lo que no hay nada que ordenar. Para comprobar esto puede utilizar la función

isempty o la función length.

En el caso recursivo, se elige un elemento cualquiera de *x* (denominado pivote,

*p*) y se divide la secuencia *x* en dos subsecuencias: *a* que contiene al resto

de elementos menores o iguales que *p* y *b* que contiene al resto de elementos

mayores que *p*. A continuación se ordenan estas dos subsecuencias de manera

recursiva (supongamos que el resultado de esta ordenación es *aord* y *bord*).

Por último, se obtiene la secuencia ordenada mediante la concatenación de las

dos subsecuencias ordenadas y el pivote: xord = [aord, p, bord].

El comando nativo sort también ordena vectores. Compruebe si su función es

correcta ejecutando los siguientes comandos:

x = rand (1, 100);

y = sort (x);

z = qsort (x);

b = (y == z);

min (b)

¿Qué debe devolver min(b) para saber que ambos resultados son idénticos?

Memoria:

function y = qsort (x)

longitud=length(x);

if longitud == 0

y = x;

else

x1=[];

x2=[];

for i=1:longitud-1

if x(i) < x(longitud)

x1=[x1 x(i)];

else

x2=[x2 x(i)];

end

end

y=[qsort(x1) x(longitud) qsort(x2)];

end

end

min(b) debe devolver 1 ya que los dos vectores son iguales, por lo que la ordenación del algoritmo sort y el algoritmo diseñado qsort dan el mismo resultado.