

Ejercicios Matlab

Nombre: _____

1. Almacénese en memoria principal la siguiente matriz, en una variable que se llame M1.

$$\begin{pmatrix} -7 & 2 & -1 \\ 3 & 4 & 2 \\ -5 & -2 & 1 \end{pmatrix}$$

Con esta matriz, se pide:

- a) Calcular la traspuesta de M1 y guardarla en M2.

```
M1=[-7 2 -1;3 4 2;-5 -2,1];  
M2=M1 .';
```

- b) Calcular el producto elemento a elemento de M1 y M2

```
M1.*M2;
```

- c) Calcular el producto matricial de M1 y M2 y guardarlo en prodM1M2

```
prodM1M2=M1*M2
```

- d) Cambiar el valor del elemento central de M1 a 9

```
M1(2,2)=9;
```

- e) Guardar en un vector fila v los elementos de la diagonal principal de M1

```
v=diag(M1);
```

```
v=v';
```

- f) Guardar en un vector columna w los elementos de la diagonal secundaria de M2

```
W=[-5;4;-1];
```

- g) Calcular el producto escalar y el producto vectorial de v y w

```
dot(v,w);
```

```
cross(v,w);
```

- h) Guardar en fila1 los elementos de la primera fila de la matriz M1

```
fila1=M1(1,:);
```

- i) Convertir fila1 en un vector columna.

```
fila1= fila1';
```

2. Se pide definir dos vectores de tres elementos (x, y, z), que representen las coordenadas 3D de dos puntos en el espacio, y calcular la distancia que hay entre ambos puntos.

```
A=(x1,y1,z1);
B=(x2,y2,z2);
Distancia=norm(A-B);
```

3. Sea el vector **b=[521 134 -45 67]**, se pide calcular el máximo y la posición que ocupa dicho elemento del vector **b**.

```
B=[512 134 -45 67];
[y,k]=max(b); //máximo es 'y' y la posición 'k'
```

4. Sea el vector **c=[31 13 42 -5 7 33 -18 3 0]**, se pide calcular el menor y el mayor de los elementos del vector, así como guardar en **COrden** el vector ordenado de **c**.

```
c=[31 13 42 -5 7 33 -18 3 0];
[m,k]=max(b); //máximo es 'm' y la posición 'k'

[n,h]=min(b); //mínimo es 'n' y la posición 'h'
COrden= sort(c);
```

5. Sea **x=17.641**, se pide calcular todos los posibles redondeos de **x** disponibles en Matlab.

```
x=17.641;
round(x);
fix(x);
floor(x);
ceil(x);
```

6. Generar una matriz de ceros de tamaño 100x100. Colocar unos en la posición (3,4), (52,36) y (89,89). Buscar a continuación en esta matriz todos los elementos distintos de cero.

```
M=zeros(100,100);
M(3,4)=1;
M(52,36)=1;
M(89,89)=1;
[f,c]=find(M);
```

7. Generar un vector entre 0 y 2π con un salto de $\pi/8$. Calcular e imprimir todas las magnitudes trigonométricas disponibles en Matlab.

```
V=[0:pi/8:2*pi];
seno=sin(V);
seno;
cos=cos(V);
cos;
tan=tan(V);
tan;
asin= asin (V);
asin;
sinh= sinh (V);
sinh;
asinh= asinh (V);
asinh;
asind= asind (V);
asind;
asindh= asindh (V);
asindh;
```

8. Dadas las longitudes de tres lados (a, b y c), decidir si forman un triángulo y de qué tipo es:

- Es triángulo si la suma de dos lados cualesquiera es mayor que el otro lado.
- Es equilátero si los lados son iguales.
- Es isósceles si dos de los lados son iguales.
- Es escaleno si los tres lados son distintos.

```
a=2.1;
b=2;
c=2.2;
if a+b>c && a+c>b && b+c>a % Es triangulo?
    if a==b && b==c % Todos los lados iguales ?
        disp('Equilatero');
    else if a~=b && b~=c && a~=c % Todos los lados diferentes?
        disp('Escaleno');
    else
        disp('Isosceles');
    end
end
else
disp('No es un triangulo');
end
```

9 Hacer un script que imprima los 20 primeros números y sus cuadrados. Hacer otro script que imprima los números impares y sus cuadrados que hay entre 1 y 20.

A)

```
for num=1:20
    disp('numero:')
    disp(num);
    disp('cuadrado');
    disp(num^2);
end
```

B)

```
for num=1:20
    b = mod(num,2);
    if b==0
        disp('numero:');
        disp(num);
        disp('cuadrado:');
        disp(num^2);
    end
end
```

10 Escribe un programa que lea números enteros procedentes del teclado y que cada vez que el número sea par lo eleve al cuadrado. El programa terminará cuando el usuario introduzca 0.

```
n = input('Teclee el numero');
while n ~=0
    if mod(n,2)==0
        disp(n^2);
    end
    n = input('Teclee el numero');
end
```

- 11 El programa solicita un número entero, comprueba si es primo y lo indica. Para comprobar la divisibilidad de un número por otro se utilizará la función mod. A partir de este script, escribe una función lógica que recibe un número entero e indica si el número es primo.

```
n = input('Teclee el numero');
if isprime(n)==1
    disp('Es primo');
else
    disp('No es primo');
end
```

Respuesta alternativa:

```
n = input('Teclee el numero');
aux=0;
for num=2:n-1
    b=mod(n, num)
    if b==0
        aux=1;
    end
end
if aux==0
    disp('Es primo');
else
    disp('No es primo');
end
```

- 12 Suponiendo que no existe el operador potencia en Matlab, escribir una función denominada potencia que acepte dos valores enteros, llamados base y exponente y devuelva base elevado a la potencia exponente. Elaborar dos versiones. Una de ellas utilizando la instrucción FOR y la otra utilizando la instrucción WHILE.

FOR

```
function [z] = potencia (num,exp)
z=1;
for n=1:exp
    z=z*num;
end
end
```

WHILE

```
function [z] = potencia (num,exp)
z=num;
n=1;
while n<exp
    z=z*num;
    n=n+1;
end
end
```