

Álgebra Lineal

Práctica 2

Ejercicios realizados:

1,2,3,4,5,6a,6b,7,8,10

Francisco Cuenca Salido

Ejercicio1:

Hacer un programa que nos averigüe el máximo de un vector de números, por ejemplo del vector: $v = [2, 7.1, -1, 2.3, 44, 11]$.

Solución:

```
Editor - C:\Users\Paco\Desktop\2cuatrimestre\AL\practica2\ejer1.m
ejer1.m
1 - v=[2 7.1 -1 2.3 44 11];
2
3 - mayor=0;
4 - tamaño=6;
5
6 - for(i=1:tamaño)
7 -     if(mayor < v(i))
8 -         mayor=v(i);
9 -     end
10 - end
11
12 - fprintf('el numero mayor es %f: \n', mayor);
13

Command Window
>> ejer1
el numero mayor es 44.000000:
fx >>
```

Ejercicio 2:

Se sabe desde hace más de dos siglos que la suma de los inversos de los cuadrados de los números naturales es:

$$1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \frac{1}{6^2} + \frac{1}{7^2} + \frac{1}{8^2} + \dots = \frac{\pi^2}{6}$$

Comprueba la veracidad de esa fórmula, para ello deberás sumar, por ejemplo, los 10000 primeros términos y comprobar que el resultado de la suma está muy próximo a $\frac{\pi^2}{6} = 1.64493$

Solución:

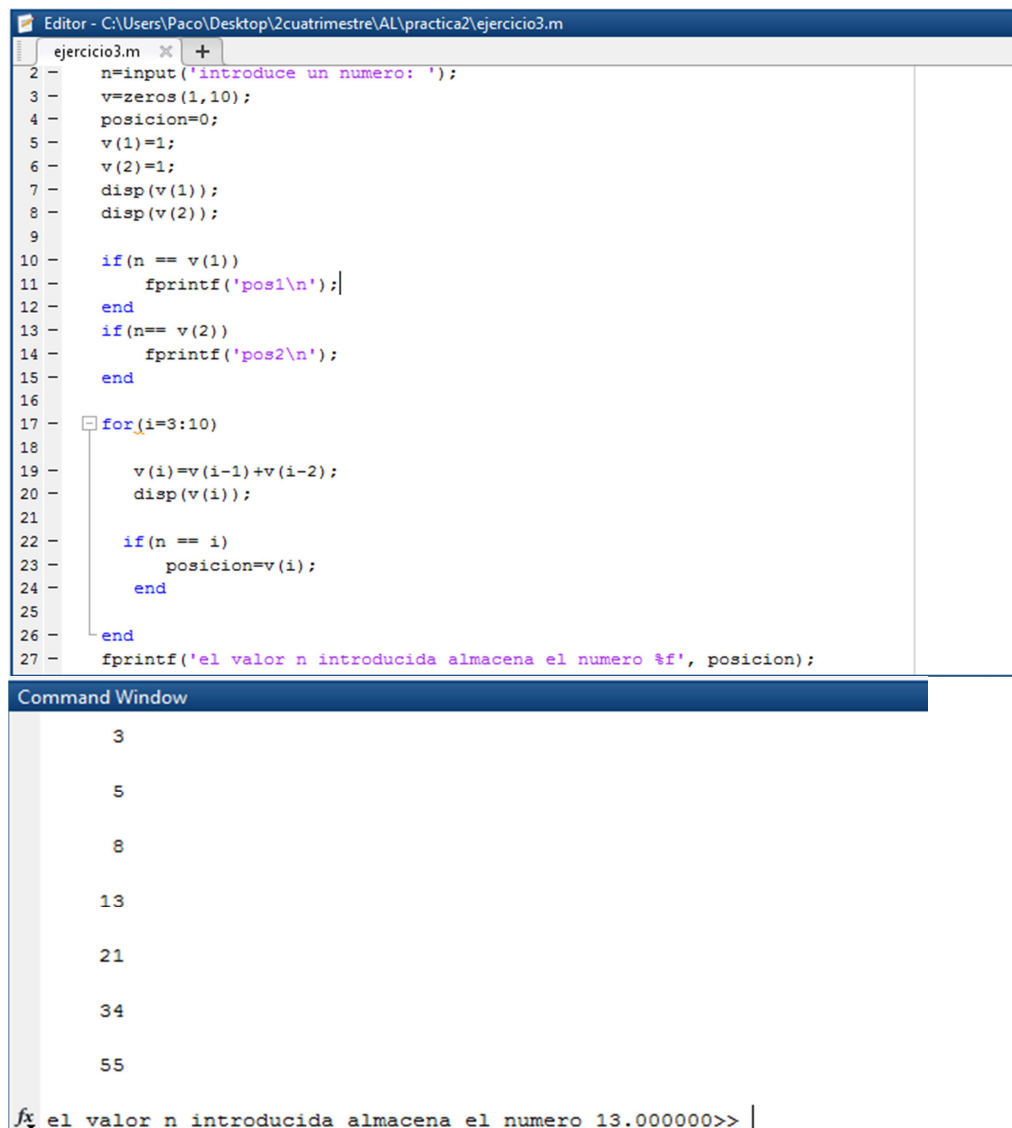
```
Editor - C:\Users\Paco\Desktop\2cuatrimestre\AL\practica2\ejer2.m
ejer2.m
1 - %ejercicio2
2 - inverso=1;
3 - %relleno de vector
4 - v=1:1000;
5
6 - for(i=2:1000)
7 -     inverso=inverso+1/(v(i)^2);
8 - end
9
10 - fprintf('resultado suma inversos de 1000 terminos es: %f\n', inverso);

Command Window
>> ejer2
resultado suma inversos de 1000 terminos es: 1.643935
fx >>
```

Ejercicio 3:

La sucesión de Fibonacci: 1, 1, 2, 3, 5, 8, 13, ... es muy conocida, no sólo en el mundo de las Matemáticas, por ejemplo se habla de ella en el libro “El código Da Vinci”, en series de televisión como “Numbers”, etc. En esta sucesión cada elemento es la suma de los dos anteriores.

Hacer un programa que nos pida un número y nos devuelva el término que ocupa la posición.

Solución:


```

Editor - C:\Users\Paco\Desktop\2cuatrimestre\AL\practica2\ejercicio3.m
ejercicio3.m
2 - n=input('introduce un numero: ');
3 - v=zeros(1,10);
4 - posicion=0;
5 - v(1)=1;
6 - v(2)=1;
7 - disp(v(1));
8 - disp(v(2));
9
10 - if(n == v(1))
11 -     fprintf('pos1\n');
12 - end
13 - if(n== v(2))
14 -     fprintf('pos2\n');
15 - end
16
17 - for(i=3:10)
18 -
19 -     v(i)=v(i-1)+v(i-2);
20 -     disp(v(i));
21
22 -     if(n == i)
23 -         posicion=v(i);
24 -     end
25 - end
26
27 - fprintf('el valor n introducida almacena el numero %f', posicion);

```

Command Window

```

3
5
8
13
21
34
55
fx el valor n introducida almacena el numero 13.000000>>

```

Ejercicio 4:

Observa estas matrices:

$$\begin{pmatrix} 1 & 2 \\ -3 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 0 \\ -3 & 1 & 2 \\ 0 & -3 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 0 & 0 \\ -3 & 1 & 2 & 0 \\ 0 & -3 & 1 & 2 \\ 0 & 0 & -3 & 1 \end{pmatrix}$$

Hacer un programa que construya una matriz A de dimensiones 20 x 20 de este tipo, es decir, toda compuesta por ceros, salvo en la diagonal donde va siempre un 1, en la diagonal paralela inmediatamente superior donde va siempre un 2 y en la diagonal inmediatamente inferior donde va siempre un -3.

Después averigua el valor de la suma de todos los elementos de la matriz $A^2 - 6A + 5A$

Solución

```

Editor - C:\Users\Paco\Desktop\2cuatrimestre\AL\practica2\ejercicio4.m
ejercicio3.m  ejer1.m  ejer2.m  ejercicio4.m  +
1 %ejercicio4
2 matriz=eye(20)
3 for(i=1:20)
4     for(j=1:20)
5         %restamos las posiciones i y j y segun el resultado ponemos un
6         %numero otro
7         if(i-j == -1)
8             matriz(i,j)=2;
9         end
10        if(i-j == 1)
11            matriz(i,j)=-3;
12        end
13    end
14 end
15
16 disp(matriz);

```

Command Window

```

0
1

Columns 1 through 19

    1    2    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   -3    1    2    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0   -3    1    2    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0   -3    1    2    0    0    0    0    0    0    0    0    0    0    0    0    0

```

```

Editor - C:\Users\Paco\Desktop\2cuatrimestre\AL\practica2\ejercicio4.m
ejercicio5.m*  ejercicio4.m  +
1 %ejercicio4
2 matriz=eye(20)
3 for(i=1:20)
4     for(j=1:20)
5         %restamos las posiciones i y j y segun el resultado ponemos un
6         %numero otro
7         if(i-j == -1)
8             matriz(i,j)=2;
9         end
10        if(i-j == 1)
11            matriz(i,j)=-3;
12        end
13    end
14 end
15 % A^2-6A+5I
16 m=matriz^2-6*matriz+5*eye(20);
17
18 resultado=sum(sum(m));
19 disp(m);
20 disp(resultado);
21
22

```

Command Window

```

82

```

Ejercicio 5:

En este ejercicio se trabaja con potencias de matrices.

Presentamos dos códigos:

Código 1:

```
tic
```

```
A = rand(1000, 1000);
```

```
S=zeros(1000,1000);
```

```
for i=1:10
```

```
    B=S+A^i;
```

```
end
```

```
toc
```

La orden tic toc de Matlab sirve para que nos salga por pantalla el tiempo empleado en ejecutarlo.

Código 2:

```
tic
```

```
A = rand(1000, 1000);
```

```
S=zeros(1000,1000);
```

```
potencia=eye(1000);
```

```
for i=1:10
```

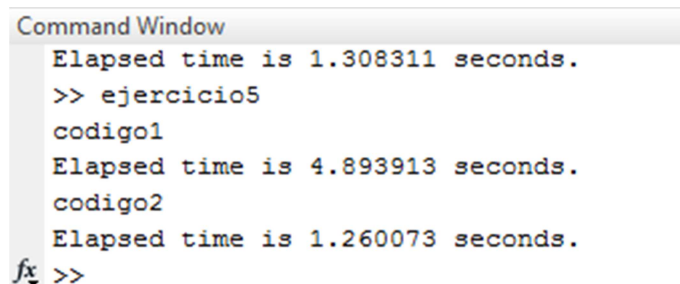
```
    potencia=potencia*A;
```

```
    B=S+potencia;
```

```
end
```

```
toc
```

Prueba estos códigos en Matlab. ¿Qué hacen estos programas? ¿Qué código es más eficiente? Razona la respuesta

Solución:

```
Command Window
Elapsed time is 1.308311 seconds.
>> ejercicio5
codigo1
Elapsed time is 4.893913 seconds.
codigo2
Elapsed time is 1.260073 seconds.
fx >>
```

Estos dos código rellenan una matriz A con 1000 números aleatorios.

Se rellena una matriz S de 1000 elementos todo a 0.

En el Código 1:

Se recorren 10 elementos y en la matriz S se almacena A^i ($A^1 + A^2 + A^3 \dots$)

En el Código 2:

Se crea una matriz potencia donde almacena una matriz Identidad.

Se recorren 10 elementos y en potencia se guarda $\text{potencia} * A$ y en S $S + \text{potencia}$.

El código 2 es mas eficiente por que solamente realiza un modificación en cambio en el código 1 es menos eficiente por que va a tener que multiplicar siempre los elementos anteriores.

```

Editor - C:\Users\Paco\Desktop\2cuatrimestre\AL\practica2\ejercicio5.m
ejercicio5.m
1  %ejercicio5
2  %Codigo1
3  tic
4      fprintf('codigo1\n');
5      A = rand(1000, 1000);
6      S=zeros(1000,1000);
7      for i=1:10
8          B=S+A^i;
9      end
10 toc
11
12 %codigo2
13 tic
14     fprintf('codigo2\n');
15     A = rand(1000, 1000);
16     S=zeros(1000,1000);
17     potencia=eye(1000);
18     for i=1:10
19         potencia=potencia*A;
20         B=S+potencia;
21     end
22 toc
Command Window
Elapsed time is 4.893913 seconds.
codigo2
Elapsed time is 1.260073 seconds.
fx >>

```

Ejercicio 7:

Hacer un programa que genera una lista aleatoria de 1000 números enteros entre 1 y 100. A continuación debe preguntarnos (con la orden input) por un número entre 1 y 100 y el programa debe devolvernos cuántas veces aparece en la lista y en qué posiciones se encuentra (las dos cosas).

Solución:

```

Editor - C:\Users\Paco\Desktop\2cuatrimestre\AL\practica2\ejercicio7.m
ejercicio7.m
1  %ejercicio7
2  %generar lista aleatoria de 1000 num entre 1 y 100
3  lista=ceil(100*rand(1,1000));
4
5  apariciones=0;
6
7  numero=input('introduce numero entre 1 y 100: ');
8  for(i=1:1000)
9      if(numero == lista(i))
10         apariciones=apariciones+1;
11         fprintf('posicion-> %d \n', i);
12     end
13 end
14 fprintf('aparece %d veces', apariciones);
Command Window
>> ejercicio7
introduce numero entre 1 y 100: 2
posicion-> 92
posicion-> 119
posicion-> 142
posicion-> 395
posicion-> 528
posicion-> 709
posicion-> 752
posicion-> 850
fx aparece 8 veces>>

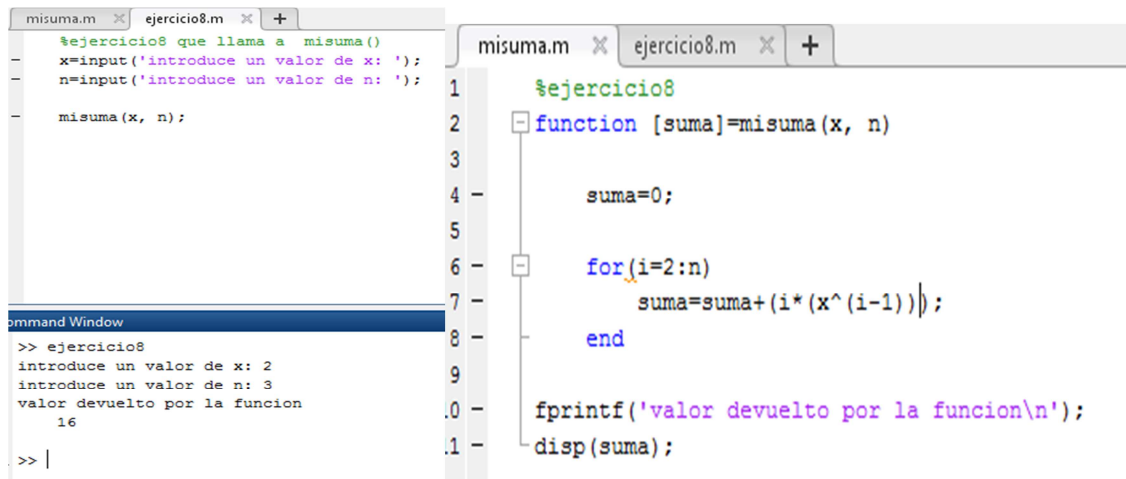
```

Ejercicio 8:

Consideremos la sucesión: $1, 2x, 3x^2, 4x^3, 5x^4, \dots$

Hacer una función de dos variables llamada `misuma()`, cuyas variables de entrada sean x y n y la función debe devolver el valor de la suma: $1, 2x, 3x^2, 4x^3 + \dots + nx^{n-1}$.

Hacer un programa principal donde se le pida al usuario un valor de la x y un valor de la n y devuelva la suma anterior usando la función `misuma()`.

Solución:

```
misuma.m  ejercicio8.m  +
%ejercicio8 que llama a misuma()
x=input('introduce un valor de x: ');
n=input('introduce un valor de n: ');

misuma(x, n);

Command Window
>> ejercicio8
introduce un valor de x: 2
introduce un valor de n: 3
valor devuelto por la funcion
    16
>> |

misuma.m  ejercicio8.m  +
1 %ejercicio8
2 function [suma]=misuma(x, n)
3
4     suma=0;
5
6     for(i=2:n)
7         suma=suma+(i*(x^(i-1)));
8     end
9
10    fprintf('valor devuelto por la funcion\n');
11    disp(suma);
```