

ÁLGEBRA LINEAL

1º GRADO DE INGENIERÍA INFORMÁTICA

GREGORIO CORPAS PRIETO



PRÁCTICA 2

PROGRAMACIÓN EN MATLAB

Ejercicio 1.

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

```
clc;
max=0;
v=[2, 7.1, -1, 2.3, 44, 11];
for i=1:length(v)
    if(v(i)>max)
        max=v(i);
    end
end
fprintf('El mayor elemento del vector v es %.2f\n',max);
```

Ejercicio 2

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

$$1 + 1/2^2 + 1/3^2 + 1/4^2 + 1/5^2 + \dots = \pi^2 / 6$$

Comprueba la veracidad de esa fórmula, para ello deberás sumar, por ejemplo, los 10000 primeros términos y comprobar que el resultado de la suma es

muy próximo a $\pi^2 / 6 = 1.64493$

```
clc;
sumatorio=0;
for i=1:1000
    sumatorio=sumatorio+(1/(i^2));
end
fprintf('El resultado de dicha fórmula es %.6f que aproximadamente es
1.64493\n',sumatorio);
```

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 n.

```
clc;

n=input('Introduce la N posicion de la sucesion fibonacci: ');
numero=0;

anterior1=1;
anterior2=1;

for i=3:n
    numero=anterior1+anterior2;

    anterior2=anterior1;
    anterior1=numero;

end

fprintf('%d\n',numero);
```

Ejercicio 4

Observa estas matrices y haz un programa que construya una matriz A de dimensiones 20×20 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 $A^2 - 6A + 5I$

```
clc;

n=input('Introduce la dimension de la matriz: ');
m=zeros(n,n);
sumatorio=0;

for i=1:length(m)
    for j=1:length(m)
        if(j==i) %Si pasamos por la posicion de la diagonal le ponemos un 1
            m(i,j)=1;
            if(i<(length(m)))
                m(i,j+1)=2;
            end
            if(i>1)
                m(i,j-1)=-3;
            end
        end
    end
end

fprintf('Esta es la matriz resultante: \n');
disp(m);

identidad=eye(n);
```

```
b=(m^2- 6*m + 5*identidad);
```

```
fprintf('Esta es la matriz B resultante tras la operacion: \n');
```

```
disp(b);
```

```
for i=1:n
```

```
    for j=1:n
```

```
        sumatorio=sumatorio + b(i,j);
```

```
    end
```

```
end
```

```
fprintf('El sumatorio de los valores de la matriz resultante es  
%d\n',sumatorio);
```

Ejercicio 5

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

% Código 1. Es menos eficiente debido a que por cada iteración recalcula todas las operaciones de multiplicación hasta alcanzar la potencia (operador ^)

```
tic
```

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

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

```
for
```

```
i=1:10
```

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

```
end
```

```
toc
```

% Código 2. Es el más eficiente debido a que lleva un 'historico' de potencia guardado de modo que solo tiene que multiplicarlo de nuevo una única vez para conseguir el exponente + 1. Realiza menos multiplicaciones.

```
tic
```

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

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

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

```
for
```

```
i=1:10
```

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

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

```
end
```

```
toc
```

Ejercicio 6

Vamos a comprobar con un programa en Matlab el resultado obtenido en el apartado anterior.

El programa te pedirá (usando la orden input) qué cuántos sumandos se quiere sumar de la expresión anterior (los 100 primeros por ejemplo) y el programa debe devolver el valor de dicha suma. Lógicamente el resultado debe ser parecido al valor límite de S dado en la expresión (1).

Hazlo de la forma más eficiente posible (ver ejercicio 5).

$$a) A^2 = A \circ A = \begin{pmatrix} \emptyset & 1/3 & \emptyset \\ 1/3 & \emptyset & \emptyset \\ \emptyset & \emptyset & 1/3 \end{pmatrix} \cdot \begin{pmatrix} \emptyset & 1/3 & \emptyset \\ 1/3 & \emptyset & \emptyset \\ \emptyset & \emptyset & 1/3 \end{pmatrix} =$$

$$= \begin{pmatrix} 1/9 & \emptyset & \emptyset \\ \emptyset & 1/9 & \emptyset \\ \emptyset & \emptyset & 1/9 \end{pmatrix} = \begin{pmatrix} 1/3^2 & \emptyset & \emptyset \\ \emptyset & 1/3^2 & \emptyset \\ \emptyset & \emptyset & 1/3^2 \end{pmatrix} = 1/3^2 I$$

$$b) M = I + A^2 + A^4 + A^6 = I + \frac{1}{3^2} I + \frac{1}{3^4} I + \frac{1}{3^6} I =$$

$$= \left(1 + \frac{1}{3^2} + \frac{1}{3^4} + \frac{1}{3^6} \right) I = \left(\frac{1 \cdot 3^6 + 1 \cdot 3^4 + 1 \cdot 3^2 + 1}{3^6} \right) I =$$

$$= \left(\frac{729 + 81 + 9 + 1}{3^6} \right) I = \left(\frac{820}{3^6} \right) I$$

$$c) H = A + A^3 + A^5 = A + \frac{1}{3^2} I A + \frac{1}{3^4} I A =$$

$$= A + \frac{1}{3^2} A + \frac{1}{3^4} A = \left(1 + \frac{1}{3^2} + \frac{1}{3^4} \right) A =$$

$$= \left(\frac{1 \cdot 3^4 + 1 \cdot 3^2 + 1}{3^4} \right) A = \left(\frac{81 + 9 + 1}{3^4} \right) A = \left(\frac{91}{3^4} \right) A$$

$$d) S = H + M = \frac{91}{3^4} A + \left(\frac{820}{3^6} \right) I =$$

$$= \begin{pmatrix} 1'1248 & \emptyset'3745 & \emptyset \\ \emptyset'3745 & 1'1248 & \emptyset \\ \emptyset & \emptyset & 1'4993 \end{pmatrix}$$

```
clc;

A=[0 1/3 0; 1/3 0 0 ; 0 0 1/3];

I=eye(length(A),length(A));
sumandos=input('Introduce el numero de elementos de la expresión que quieres
sumar: ');
S=zeros(length(A),length(A));

potencia=I;

for i=1:sumandos
    potencia=potencia*A;
    S=S+potencia;
end
disp(S);
```

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).

```
clc;

coincidencias=0;
elementos=1000;
lista=ceil(100*rand(elementos,1));
%redondea al alza un aleatorio entre 0 y 1 multiplicado por 100

numero=input('Introduce el valor a buscar en esta lista: ');

for i=1:elementos
    if(lista(i,1)==numero)
        fprintf('\n\tCoincidencia encontrada en la posicion (%d,1)',i);
        coincidencias=coincidencias+1;
    end
end
fprintf('\n\nHubo un total de %d resultados\n',coincidencias);
```

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: $1, 2x, 3x^2, 4x^3, 5x^4 \dots + nx^{n-1}$

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()`

```
% PROGRAMA PRINCIPAL

clc;

x=input('Introduce el valor de x: ');
n=input('Introduce el valor de n: ');

resultado=misuma(x,n);

fprintf('\nEl valor devuelto por misuma(%.2f,%.2f) es %.2f\n',x,n,resultado);

%FUNCION MISUMA
function[retorno] = misuma(x,n)
    retorno=0;
    for i=1:n
        retorno=retorno + (i * x^(i-1));
    end
end
```

Ejercicio 9

Un número entero entre 0 y 31 se escribe con 5 bits:

0 → 00000

1 → 00001

2 → 00010

3 → 00011

...

30 → 11110

31 → 11111

Hacer un programa que nos pida un número entero en ese rango (con la orden input) y nos calcule ese número escrito en binario con 5 bits. Crea una función para ello

```
% PROGRAMA PRINCIPAL

clc;

valor=input('Introduce un numero a transformar en binario: ');

binario=zeros(1,5);

binario=abinario(valor);

fprintf('El numero %d en binario es:',valor);

disp(binario);
```

```
%FUNCION ABINARIO
```

```
function[retorno] = abinario (numero)
```

```
    retorno=zeros(1,5);
```

```
    divisor=0;
```

```
    posicion=5;
```

```
    while(posicion~=1)% Mientras estemos rellenoando hasta la segunda posicion...
```

```
        retorno(1,posicion)=mod(numero,2); % Estos valores serán los restos de  
las divisiones
```

```
        numero=floor(numero/2); % Y el dividendo se actualizará a el valor  
entero de la división por 2
```

```
        posicion=posicion-1; % Decrementando la posicion
```

```
    end
```

```
    retorno(1,1)=numero;%El bit mas significativo es el ultimo cociente
```

Ejercicio 10

En este ejercicio vamos a ver cómo extraer un mensaje oculto en una imagen. Esto se conoce como esteganografía. La imagen se llama “estegolsb1.png ” y debes bajártela de Moodle.

Vamos a usar a continuación un alfabeto compuesto de 32 caracteres que incluye las 27 letras del abecedario además del punto, la coma, los símbolos de interrogación y el espacio en blanco

```
% PROGRAMA PRINCIPAL

clc;

a=imread('estegolsb1.png');

imshow(a); %muestra la imagen

whos a; % Nos especifica la dimensión y codificación de la imagen

%whos a nos dice que la imagen es una matriz de 256 x 256
%sabiendo esto podemos recorrer linea a linea la matriz
%y guardando el valor del bit menos significativo
% de cada posicion

b=mod(a(1:256,1:256),2);

% con la sentencia anterior guardamos el bit menos significativo
% de cada celda de la matriz, ya que la recorremos desde el
% elemento 1 hasta el 256 tanto en i como j
anterior='';

paquete='';
contador=0;
for i=1:256
    for j=1:256
        contador=contador+1;
        paquete=strcat(num2str(paquete),num2str(b(i,j)));
```

```
        if(mod(contador,5)==0)
            letra=imprimeletra(paquete);

            if((strcmp(letra,anterior)) && (strcmp(letra,'A')))

                else
                    fprintf('%c',letra);
                end
                anterior=letra;

                paquete='';

            end
        end
    end
end
```



```
% FUNCION IMPRIMELETRA
function [letra] = imprimeletra( paquete)
letra='';
switch paquete
    case '00000'
        letra='A';
    case '00001'
        letra='B';
    case '00010'
        letra='C';
    case '00011'
        letra='D';
    case '00100'
        letra='E';
    case '00101'
        letra='F';
    case '00110'
        letra='G';
    case '00111'
        letra='H';
    case '01000'
        letra='I';
    case '01001'
        letra='J';
    case '01010'
        letra='K';
    case '01011'
        letra='L';
    case '01100'
        letra='M';
    case '01101'
        letra='N';
    case '01110'
```

```
        letra='Ñ';
case '01111'
        letra='O';
case '10000'
        letra='P';
case '10001'
        letra='Q';
case '10010'
        letra='R';
case '10011'
        letra='S';
case '10100'
        letra='T';
case '10101'
        letra='U';
case '10110'
        letra='V';
case '10111'
        letra='W';
case '11000'
        letra='X';
case '11001'
        letra='Y';
case '11010'
        letra='Z';
case '11011'
        letra='.';
case '11100'
        letra=',';
case '11101'
        letra='&';
case '11110'
        letra='?';
```

```
case '11111'  
    letra=' '  
end
```