***Álgebra Lineal***

***Práctica 1***

***Ejercicios realizados:***

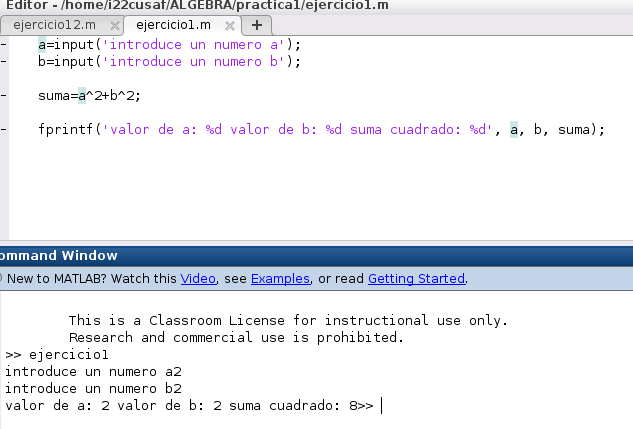
**1,2,3,4,5,6,7,8,9,10,11,12**

***Francisco Cuenca Salido***

**Ejercicio1:**

**Crea un programa (fichero .m) que te pida dos números usando la orden input de Matlab y nos muestre en pantalla el resultado de la suma de sus cuadrados. Nota: La orden input se introduce de la forma a=input(‘introduce un número’);**

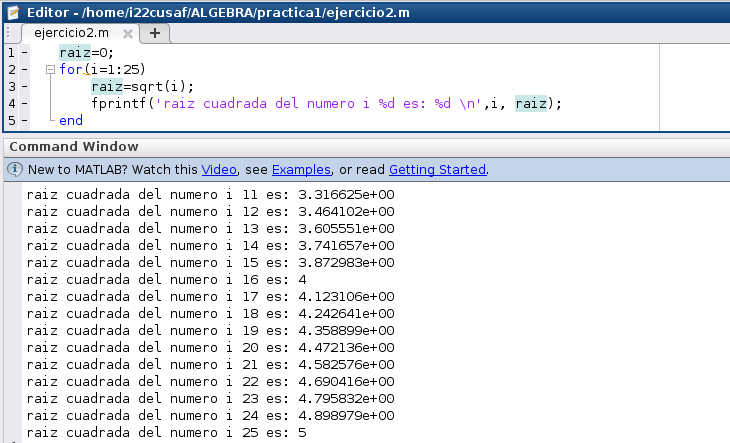
**Solución:**



**Ejercicio 2:**

**Escribe un script que permita escribir en pantalla todos los números enteros entre 1 y 25 y sus raíces cuadradas.**

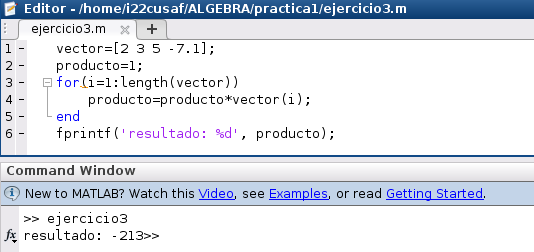
**Solución:**



**Ejercicio 3:**

**Escribe un programa que permita averiguar el producto de todos los elementos del vector [2 3 5 − 7.1]**

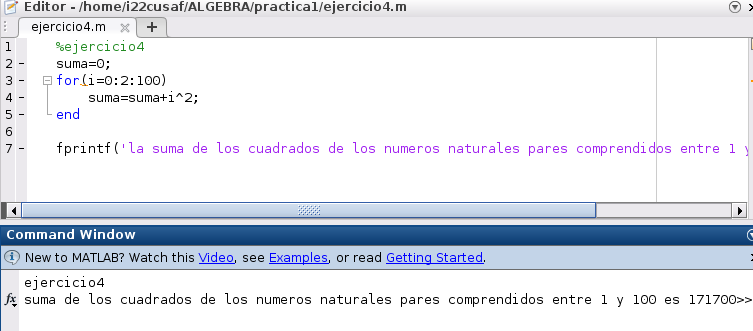
**Solución:**



**Ejercicio 4:**

**Escribe un programa que permita averiguar la suma de los cuadrados de los números naturales pares comprendidos entre 1 y 100.**

**Solución:**



**Ejercicio 5:**

**Escribe un programa que permita averiguar para la matriz del ejemplo anterior: - La suma de cada fila - La suma de cada columna - La suma de los cuadrados de todos los elementos de la matriz.**

**Solución:**

%ejercicio5

A=[2 3 5 -7.1; 1 2 3 4];

suma=0;

sumacuadrados=0;

[nfilas, ncolumnas]=size(A);

%suma de cada fila

fprintf('suma de las filas\n');

for(i=1:nfilas)

for(j=1:ncolumnas)

suma=suma+A(i,j);

sumacuadrados=sumacuadrados+A(i,j)^2;

end

fprintf('fila %d resultado: %f \n', i, suma);

suma=0;

end

fprintf('la suma de los cuadrados de todos los elementos de la matriz es %f', sumacuadrados)

suma=0;

fprintf('\nsuma de las columnas\n');

for(j=1:ncolumnas)

for(i=1:nfilas)

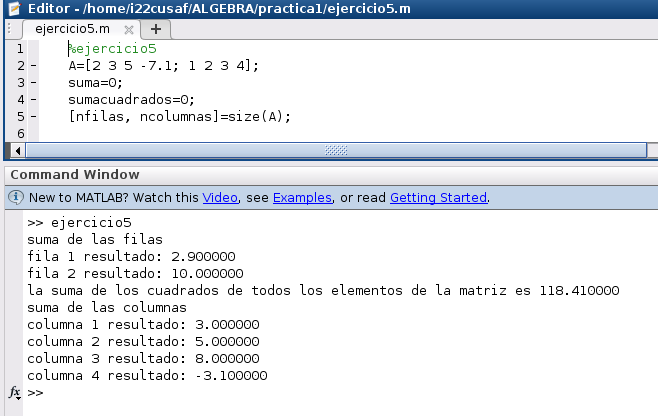
suma=suma+A(i,j);

end

fprintf('columna %d resultado: %f \n', j, suma);

suma=0;

end

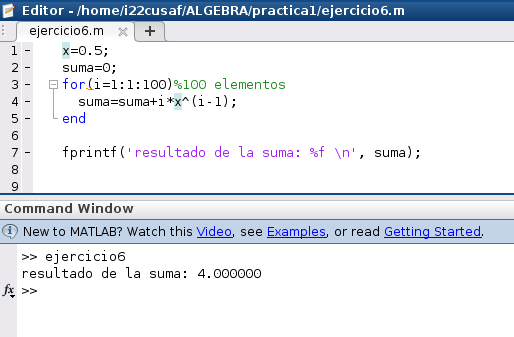


**Ejercicio 6:**

**Consideremos la sucesión 1, 2x1, 3x2 ,4x3 , 5x4 , …**

**Siendo x = 0.5, averiguar la suma de los 100 primeros términos de esta sucesión. (Solución: 4)**

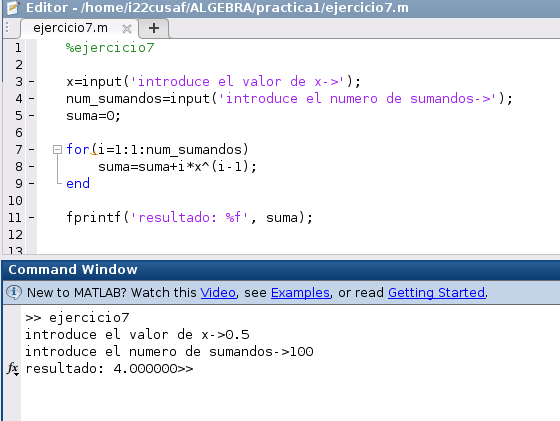
**Solución:**



**Ejercicio 7:**

**Modificar el programa anterior para que el usuario entre el valor de que desea y el número n de sumandos que desea sumar. Por ejemplo, si = 0.5 y $ = 100 debe devolver, evidentemente, lo mismo que en el ejercicio anterior. Para ello usarás la función de Matlab input.**

**Solución:**

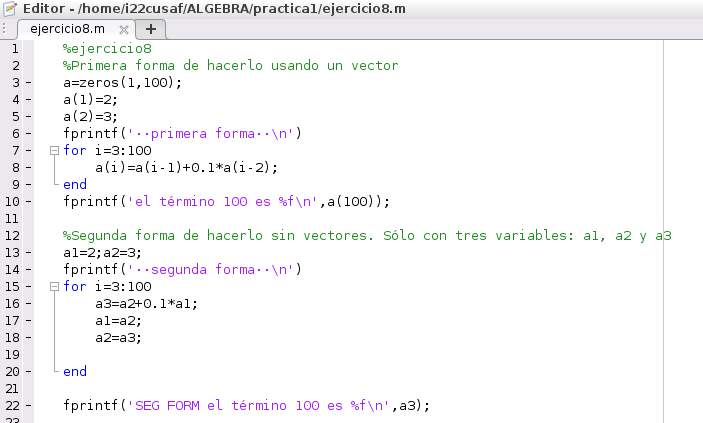


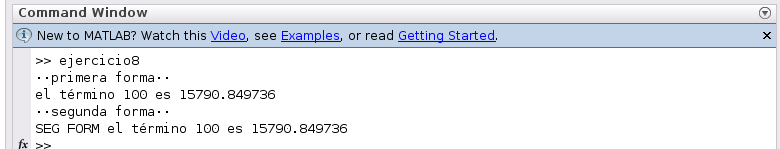
**Ejercicio 8:  
 Supongamos que tenemos una sucesión de la que sabemos:**

**Queremos averiguar el valor del término que ocupa la posición 100.**

**Para ello te proporciono dos códigos incompletos que tú debes completar para hacer que funcione correctamente.**

**Solución:**





**Ejercicio9:**

**Escribe el siguiente programa y mira el resultado:**

x = 0;

contador = 0;

while (x<=0.95)

contador = contador + 1;

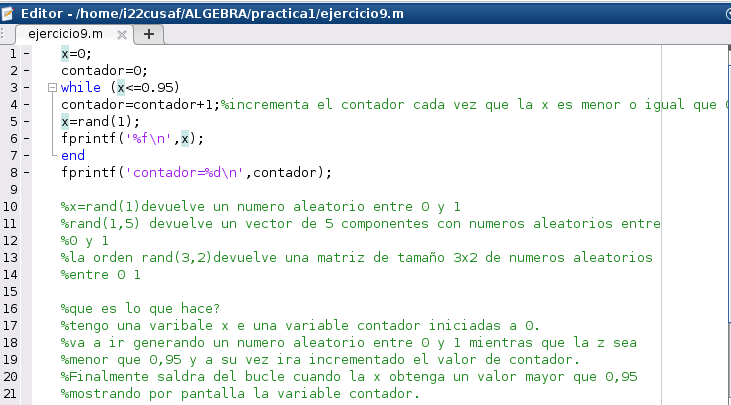
x=rand(1);

fprintf('%f\n',x);

end

fprintf('contador = %d\n',contador);

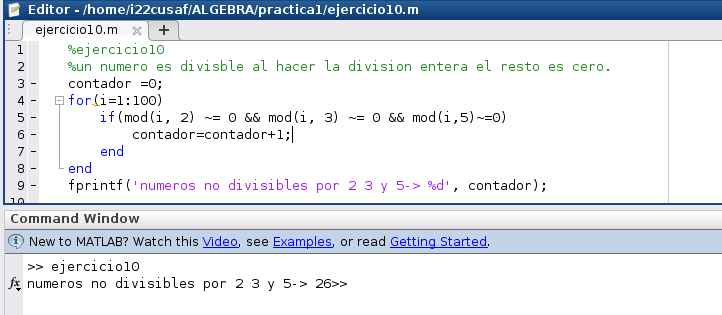
**Solución:**



**Ejercicio 10:**

**Recordar que un número es divisible por otro cuando al hacer la división entera resulta que el resto de la división es cero. Por ejemplo, 8 es divisible por 2, 12 es divisible por 3, etc. Hay una función en Matlab que nos devuelve el resto de la división entera: mod(a, n) devuelve el resto de la división entera de a entre n. Por ejemplo, mod(13, 3) devuelve un 1. Hacer un programa que nos permita averiguar cuántos números naturales hay entre 1 y 1000 que no sean divisibles ni por 2 ni por 3 ni por 5.**

**Solución:**

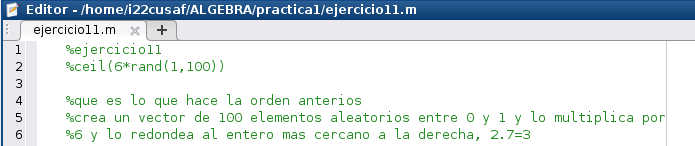


**Ejercicio 11:**

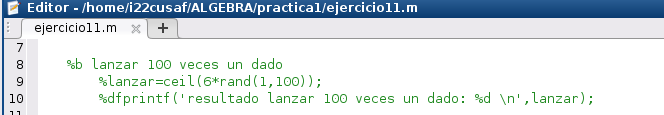
**Supongamos que deseamos simular el lanzamiento de un dado. Usaremos la orden: ceil(6\*rand(1,100)) a) Explica razonadamente qué hace la orden anterior. b) Úsala para simular 100 lanzamientos de un dado. c) Cuenta el número de veces que ha salido un 1, el número de veces que ha salido un 2, etc.**

**Solucion:**

**a) Explica razonadamente qué hace la orden anterior.**



**b) Úsala para simular 100 lanzamientos de un dado.**



**c) Cuenta el número de veces que ha salido un 1, el número de veces que ha salido un 2, etc.**

**Código:**

%ejercicio11

%ceil(6\*rand(1,100))

%que es lo que hace la orden anterior

%crea un vector de 100 elementos aleatorios entre 0 y 1 y lo multiplica por

%6 y lo redondea al entero más cercano a la derecha, 2.7=3

%b lanzar 100 veces un dado

%lanzar=ceil(6\*rand(1,100));

%dfprintf('resultado lanzar 100 veces un dado: %d \n',lanzar);

%c cuenta el número de veces que ha salido un 1 un 2 ...

lanzar=ceil(6\*rand(1,100));

apariciones1=0;

apariciones2=0;

apariciones3=0;

apariciones4=0;

apariciones5=0;

apariciones6=0;

for(i=1:100)

if(lanzar(i) == 1)

apariciones1=apariciones1+1;

end

if(lanzar(i) == 2)

apariciones2=apariciones2+1;

end

if(lanzar(i) ==3)

apariciones3=apariciones3+1;

end

if(lanzar(i) == 4)

apariciones4=apariciones4+1;

end

if(lanzar(i) == 5)

apariciones5=apariciones5+1;

end

if(lanzar(i) == 6)

apariciones6=apariciones6+1;

end

end

fprintf('veces que aparece el 1: %d\n', apariciones1);

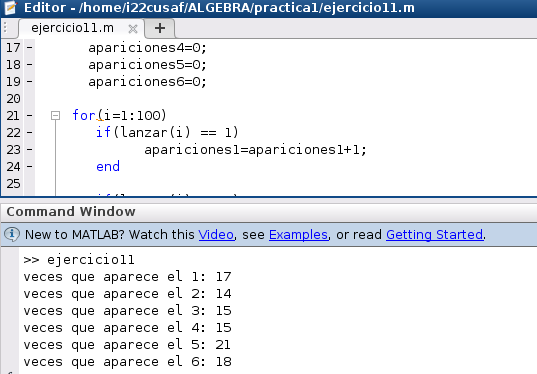
fprintf('veces que aparece el 2: %d\n', apariciones2);

fprintf('veces que aparece el 3: %d\n', apariciones3);

fprintf('veces que aparece el 4: %d\n', apariciones4);

fprintf('veces que aparece el 5: %d\n', apariciones5);

fprintf('veces que aparece el 6: %d\n', apariciones6);



**Ejercicio 12:**

**Vamos a simular el lanzamiento de una moneda. Se va a ir lanzando una moneda sucesivas veces hasta que consigamos obtener 20 caras. En ese momento pararemos y deberemos sacar en pantalla el número de tiradas que han sido necesarias.**

**Solución:**

