



Mediante Octave/Matlab, realizar los siguientes programas:

**Práctica - Ejercicio 1:** Hacer un programa (circulo.m) que solicite el radio de una circunferencia y muestre por pantalla el diámetro, el perímetro y el área.

**Práctica - Ejercicio 2:** Hacer un programa para convertir una velocidad en m/s a km/h (velocidad.m).

**Práctica - Ejercicio 3:** Hacer un programa que muestre por pantalla la temperatura en grados Kelvin y en grados Fahrenheit a partir de la temperatura en grados Celsius introducida manualmente por el usuario. (temperatura.m).

**Práctica - Ejercicio 4:** Hacer un programa que te pida un número y te diga si es par o es non. (parImpar.m).

**Práctica - Ejercicio 5:** Hacer un programa que lea 2 números y escriba cual es el menor. (lessNumber.m).

**Práctica - Ejercicio 6:** Hacer un programa que pida 3 números y diga cual es el mayor. (greatestNumber.m).

**Práctica - Ejercicio 7:** Hacer un programa que pida la hora y muestre si es por la mañana (<12), por la tarde (<20) o de noche(>=20). (hour.m).

**Práctica - Ejercicio 8:** Hacer un programa que solicite 4 calificaciones y diga si está aprobado o no. (nota.m).

**Práctica - Ejercicio 9:** Hacer un programa que recoja dos valores (i y j). Que escriba “POSITIVOS” si los dos son positivos, “NEGATIVOS” si los dos son negativos o “MEZCLADOS” si uno es positivo y el otro no. (signos.m).

**Práctica - Ejercicio 10:** Hacer un programa que solicite los coeficientes de una ecuación cuadrática, la clasifique en función del discriminante y devuelva las soluciones. Las soluciones complejas se darán en función de su parte real e imaginaria. (equation.m).

**Práctica - Ejercicio 11:** Determine si un año es bisiesto. Un año es bisiesto si es múltiplo de 4 pero no de 100 a excepción de los múltiplos de 400 que siempre son bisiestos. (leapYear.m).

**Práctica - Ejercicio 12:** Hacer un programa que lea un valor x y devuelva f(x), siendo f(x) la función de la presentación. (funcionTrozos.m).

Los programas deben respetar los nombres definidos en el enunciado e incluir en el inicio un comentario con el nombre, grado y fecha.

**Tarea 1:** La enfermedad cardiovascular (ECV) continúa siendo la principal causa de mortalidad y morbilidad en todo el mundo. La puntuación Fuster-Bewat se recomienda para la prevención de la enfermedad cardiovascular basada en el estilo de vida definida en base a los siguientes factores: Presión arterial, Ejercicio, Peso, Alimentación y Tabaco.

Este índices recoge información clínica sobre el estilo de vida y los factores de riesgo, incluidos los de tabaquismo, actividad física, alimentación (consumo de frutas y verduras), peso corporal y presión arterial, por lo que no requiere resultados analíticos. Esto hace que su uso sea fácil y apropiado al ser poco invasivos y poco costosos..

La puntuación Fuster Bewat puede utilizarse con fines de educación sanitaria en entornos no médicos (es decir, en las escuelas) y para el autoseguimiento personal como instrumento para mejorar el autocuidado cardiovascular se basa en el pdf adjunto que podéis consultar también en el siguiente enlace.

En la presente tarea se realizará un programa de Octave/Matlab, `ecvScale.m`, el cual solicite al usuario los datos necesarios para cumplimentar los 5 apartados recogidos en la escala, estime la puntuación total en base a los óptimos definidos y, con ello, muestre por pantalla la interpretación final obtenida. A modo de ejemplo, un usuario con

PAS=10; PAD=7; %% Presiones arteriales

Ejercicio=5; %% Horas de ejercicio semanal

IMC=30; %% Índice de masa corporal

# Fundamentos de Computación

## Enunciados de Prácticas – Bloque 02

Fruta=3; %% Piezas de fruta al día

Fuma='NO'; %% Fumador o no.

tendría una puntuación total de 3 y, por tanto, se mostraría por pantalla que la persona se encuentra en un nivel intermedio de salud cardiovascular.

Debe respetarse el encabezado incluido en el Moodle. Es decir, el programa deberá incluir un comentario inicial con el nombre y apellidos del/la estudiante, así como una descripción de lo que hace el programa realizado.

**Práctica - Ejercicio 13:** Hacer un programa que solicite 3 valores asociados a los lados de un triángulo y que devuelva por pantalla que tipo de triángulo es (equilátero, isósceles o escaleno).

**Práctica - Ejercicio 14:** Hacer un programa que solicite 3 valores asociados a los ángulos de un triángulo y que devuelva por pantalla que tipo de triángulo es (rectángulo, obtuso o acutángulo).

**Práctica - Ejercicio 15:** Hacer un programa que unifique los dos anteriores, de modo que solicite ángulos y lados, y clasifique el triángulo según ambos criterios.

Recordad que los ángulos de un triángulo deben sumar 180°. En caso contrario, el programa debe advertir por pantalla del error y no hacer nada.

Los nombres de los ficheros se han de llamar OBLIGATORIAMENTE, LadosTriangulo.m, AngulosTriangulo.m y ClasificacionTriangulo.m, respectivamente.

Además debéis comentar el programa e incluir en el inicio de cada script el nombre, grado y fecha. Los mensajes para el usuario han de ser suficientemente informativos.

**Práctica - Ejercicio 16:** Las ciclogénesis explosivas (<http://es.wikipedia.org/wiki/Ciclogenesis>) son eventos meteorológicos extremos con un gran impacto en las regiones afectadas provocando grandes pérdidas tanto económicas como humanas. Dichos eventos se caracterizan, entre otras cosas, por un descenso pronunciado de la presión atmosférica en las 24 horas anteriores, el cual puede estimarse con la fórmula:

$$NDR = ((Presion(d)-Presion(d-1))/24)*(\sin(\pi/3)/\sin(latitud));$$

Si, para una latitud y un día (d) dados, ese índice es mayor o igual que 1 se entiende que puede desarrollarse una ciclogénesis explosiva. Por ello un programa (ciclogenesisExplosiva.m) que:

Solicite la latitud en grados, la presión en el día anterior (Presion(d-1)) y en el mismo día ((Presion(d))), ambas en Pa.

En base a los valores anteriores calcule el índice NDR para el día y el punto dado. Para ello, deberá expresar la presión en hPa y la latitud en radianes realizando el correspondiente cambio de unidades.

Imprima por pantalla si hay o no posibilidad de ocurrencia de un evento de ciclogénesis explosiva, según sea el índice NDR menor o no que 1, y, en caso de poder darse una ciclogénesis explosiva, muestre por pantalla el valor del índice e indique si la ciclogénesis será leve ( $NDR < 2$ ), grave ( $NDR < 3$ ) ó muy grave ( $NDR \geq 3$ ).

**Práctica - Ejercicio 17:** Realizar un programa de Octave, proyectil.m, que muestre por pantalla si un proyectil lanzado desde una altura  $h_0$  es visto por un observador mirando al frente situado a una altura  $h_1$  una única vez, dos veces o ninguna. Para ello:

Solicite la altura  $h_0$  desde la cual se lanza el proyectil, la velocidad con la cual se lanza,  $v_0$ , y la altura  $h_1$  en la que se encuentra el observador. Notar que la aceleración es la gravitacional ( $g = 9.8$  m/s).

Plantea la ecuación del movimiento (ver la definición aquí).

Discutir la ecuación obtenida al igualar la ecuación del movimiento con la altura  $h_1$  en la que se encuentra el observador. Por ejemplo, si  $h_0 = 0$  (el proyectil parte del suelo),  $v_0 = 10$  m/s y  $h_1 = 10$ , la ecuación ( $h_1 = (g/2)*t^2 + v_0*t + h_0$ ) vendría dada por:  $10 = (-9.8/2)*t^2 + 10*t + 0$ .

Imprima por pantalla cada caso según la ecuación tenga una, dos o ninguna solución real.

Para el desarrollo de la tarea no se considerará la masa del proyectil.

# Fundamentos de Computación

## Enunciados de Prácticas – Bloque 02

**Práctica - Ejercicio 18:** Realizar los scripts correspondientes a los diagramas de flujo de la semana anterior utilizando ciclos (while y for):

**Práctica - Ejercicio 19:** Que solicite 4 calificaciones y diga si está aprobado o no.

**Práctica - Ejercicio 20:** Para calcular el factorial de un número.

**Práctica - Ejercicio 21:** Para imprimir por pantalla los primeros N términos de la sucesión de Fibonacci, donde el número será introducido manualmente por el usuario.

**Práctica - Ejercicio 22:** Que pida un número N y despliegue todas las combinaciones de dos números que sumados den N.

**Práctica - Ejercicio 23:** Que despliegue la tabla de multiplicar ( $0 \cdot X, \dots, 10 \cdot X$ ) de un número X.

**Práctica - Ejercicio 24:** Que recoja números del usuario hasta que se introduzca un 0 y devuelva la media de los números introducidos.

**Práctica - Ejercicio 25:** Que pida un número y vaya decrementando su valor en una unidad hasta que llegue a cero.

**Práctica - Ejercicio 26:** Que pida un número y compruebe si ese número es primo.

**Práctica - Ejercicio 27:** Que pida un número hasta que este número sea mayor que cero, indicando con un mensaje que el número introducido no es mayor de cero.

**Práctica - Ejercicio 28:** Que calcule la raíz en el intervalo  $[0, 2]$  del polinomio  $x^3 - 3x^2 - 2x + 6$  con una precisión de  $1e-5$  a través del método de la bisección.

**Práctica - Ejercicio 29:** Que calcule la raíz del polinomio anterior con el método de Newton-Raphson con la misma tolerancia tomando como condición inicial los extremos del intervalo  $[0, 2]$ . ¿Qué soluciones obtienes? ¿Coinciden con las anteriores?

**Práctica - Ejercicio 30:** Realizar un programa (SucesionCondicional.m) que calcule los términos de la sucesión:

$$x_{n+1} = \begin{cases} a \cdot x_n & \text{si } x_n < 1/2, \\ a \cdot (1 - x_n) & \text{si } 1/2 \leq x_n \end{cases}$$

preguntando por el valor inicial  $x_0$ , el coeficiente  $a$  y el número de términos. El valor de  $a$  ha de ser mayor que cero y el de  $x_0$  ha de estar en el intervalo  $[0, 1]$

**Práctica - Ejercicio 31:** Realizar un programa (SucesionCondicional2.m) que sume todos los términos que se muestren por pantalla en el ejercicio anterior, y calcule también su producto. La suma y el producto se mostrarán al final.

**Práctica - Ejercicio 32:** Realizar un programa (SucesionCondicional3.m) que dirá si la sucesión converge a un valor si después de 2 iteraciones/repeticiones seguidas, el valor no cambia.

**Práctica - Ejercicio 33:** Considerar las funciones det (determinante), rank (rango), poly (polinomio característico) y roots (raíces de un polinomio) para hallar la dimensión del subespacio imagen, el núcleo y calcular la expresión diagonal de la aplicación lineal dada por la matriz (linearApplication.m):

$$A = \begin{bmatrix} 2 & 1 & 4 \\ 3 & 0 & -1 \\ 4 & -1 & 5 \end{bmatrix};$$

**Práctica - Ejercicio 33:** Implementa con bucles el producto matricial (matrixProd.m) y evalúa la diferencia entre el tiempo de cálculo con bucles y con el operador \* de Octave (funciones tic y toc).

**Práctica - Ejercicio 34:** Resuelve la ecuación matricial (matrixEquation.m):

$$A \cdot X \cdot B^{-1} = 2 \cdot E;$$

donde  $A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & 0 \\ 2 & 2 & 1 \end{bmatrix};$

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix};$$

$$E = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix};$$

**Práctica - Ejercicio 35:** Dada la matriz  $A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{bmatrix}$  y el escalar  $r=2$ , comprueba las propiedades del determinante (propDeterminant.m).

# Fundamentos de Computación

## Enunciados de Prácticas – Bloque 02

**Práctica - Ejercicio 36:** Considerar una matriz  $A$  en la cual se recoge el resultado, expresado como diferencia de tantos, entre los 20 equipos de una competición, representando cada fila los resultados de un equipo frente a los otros 19. Por lo tanto,  $A$  debe ser una matriz cuadrada, con la diagonal de ceros y con valores enteros en el resto de posiciones (p.e.  $A = \text{floor}(\text{randn}(20,20)*5)$ ; for  $i=1:20, A(i,i)=0$ ; end). Calcular la puntuación de cada equipo de la competición considerando que:

- Una victoria ( $A(i,j) > 0$ ) supone 3 puntos.
- Un empate supone 1 punto.
- Una derrota por 1 tanto supone un punto.

**Tarea 2:** Realizar un programa de Octave/Matlab (extremeEvents.m) que cuente cuantos días hay de cada uno de los tipos siguientes, Ice days (Id - days with  $T_X < 0^\circ\text{C}$ ), Wind-day (Wd - day with wind speed greater than  $10.8 \text{ m}\cdot\text{s}^{-1}$ ), Calm-day (Cd - days with wind speed lower than  $2 \text{ m}\cdot\text{s}^{-1}$ ), Wind-Ice day (Wd-Id), Calm-Ice day (Cd-Id) y normales (si no se encuentran en ninguna de las clases anteriores), en base a la definición dada en el siguiente enlace <https://www.ecad.eu/indicesextremes/indicesdictionary.php>, en un periodo de tiempo solicitado al usuario para que lo introduzca manualmente.

Para ello, el programa debe solicitar el número de días considerado (Ndays) y generar (ver la ayuda de las funciones rand y randn) las series de viento (p.e.  $\text{Wind} = 6 * \text{rand}(N\text{days}, 1)$ ;) y temperatura máxima (p.e.  $T_x = 20 + 10 * \text{randn}(N\text{days}, 1)$ ;) a partir de las cuales realizar la clasificación y la cuenta de los días de cada clase.

Finalmente, el programa mostrará por pantalla la media de viento, la media de temperatura máxima y el número de días de cada clase.