

Nombre y apellidos: _____

Fecha y hora de entrega del examen en el aula virtual: _____

Ejercicio 1 (1 punto)

Diseña una función que reciba como parámetro el radio de un cilindro y su altura, y muestre por pantalla el área total y el volumen del cilindro.

Fórmulas:

- Área total = $2 \cdot \pi \cdot \text{radio} \cdot (\text{radio} + \text{altura})$
- Volumen = $\pi \cdot \text{radio}^2 \cdot \text{altura}$

Ejercicio 2 (1 punto)

Implementa la función:

double[] puntoMedio(double x1, double y1, double x2, double y2)

que devuelva las coordenadas del punto medio entre los puntos (x1, y1) y (x2, y2).

Fórmula:

- $x = (x1 + x2) / 2$
- $y = (y1 + y2) / 2$

Ejercicio 3 (1 punto)

Escribe una función que reciba una distancia expresada en kilómetros y metros, y devuelva la distancia total expresada únicamente en metros.

Ejercicio 4 (1 punto)

Diseña una función recursiva que calcule el n-ésimo número de la sucesión de Lucas.

La sucesión de Lucas se define como:

- $\text{lucas}(n) = \text{lucas}(n - 1) + \text{lucas}(n - 2)$
- $\text{lucas}(0) = 2$
- $\text{lucas}(1) = 1$

Ejercicio 5 (1 punto)

Escribe una función que determine si un número es un número capicúa. La función devolverá **true** si el número se lee igual de izquierda a derecha que de derecha a izquierda, y **false** en caso contrario.

Ejemplos:

- 121 → capicúa
- 3443 → capicúa
- 123 → no capicúa

Ejercicio 6 (1 puntos)

Pide al usuario que introduzca números enteros por teclado hasta que introduzca el número 0. Guarda los números introducidos en un array.

Una vez creado el array, itera sobre el array y calcula:

- La suma de los números positivos.
- La suma de los números negativos.
- El número de múltiplos de 5.

Ejercicio 7 (1,5 puntos)

Escribe la función:

int[] rellenaMultiplos(int longitud, int num)

que cree y devuelva un array de la longitud indicada, relleno con números aleatorios que sean múltiplos de **num**, comprendidos entre 1 y 100 (inclusive).

El array devuelto deberá estar ordenado de menor a mayor.

Usa la función en el **main** e imprime por pantalla el array resultante.

Ejercicio 8 (1,5 puntos)

Implementa la función:

int[] maximos(int t[], int numElementos)

que cree y devuelva un array con el valor máximo de cada grupo de **numElementos** consecutivos del array **t**.

Ejemplo:

Si **t = [4, 7, 2, 9, 5, 1]** y **numElementos = 3**, se obtendrán:

- $\max(4, 7, 2) \rightarrow 7$
- $\max(7, 2, 9) \rightarrow 9$
- $\max(2, 9, 5) \rightarrow 9$
- $\max(9, 5, 1) \rightarrow 9$

Resultado: **[7, 9, 9, 9]**

Ejercicio 9 (1 punto)

Dado un array bidimensional de enteros que representa una tabla de números, recorre la matriz y construye un array unidimensional que contenga la suma de cada fila del array bidimensional.

Imprime por pantalla el array resultante.

Ejemplo:

- Si la matriz es:
1 2 3
4 5 6
7 8 9
- El array resultante será:
[6, 15, 24]