

Ejercicio 1 – Cifrado César

• 1 Enunciado

Escribe un programa que **cifre** el texto "CIFRADOCESAR" usando un desplazamiento de **+5**.

• 2 Condiciones

- Nombre de programa: `ejercicio_1.xxx`
- Lenguaje libre (C++, Python, Java, etc.).
- No usar librerías de cifrado ni utilidades externas.
- Usar **solo estructuras y funciones básicas** del lenguaje.
- Buscar la **menor cantidad de líneas posible**.

• 3 Sugerencias

- Convertir a mayúsculas y quitar espacios.
- Trabajar con el valor ASCII de cada letra ('A' = 65).
- Aplicar la fórmula:

`Cifrado = (LetraOriginal + Desplazamiento) mod 26`

• Puntos: 1.5 Puntos

Ejercicio 2 – Cifrado Playfair

• 1 Enunciado

Escribe un programa que **cifre** el mensaje "CIFRADOPLAYFAIR" usando la clave "CLAVE".

• 2 Condiciones

- Nombre de programa: `ejercicio_2.xxx`
- Lenguaje libre.
- No usar librerías que generen automáticamente matrices de cifrado.
- Implementar la **matriz 5x5** manualmente a partir de la clave.
- Respetar la regla de unir I/J en una celda.
- Procesar pares de letras (añadir `x` si es necesario).

• Puntos: 1.5 Puntos

Ejercicio 3 – Cifrado Vigenère

• 1 Enunciado

Escribe un programa que **cifre** el texto "CIFRAVOVIGENERE" usando la clave "CLAVE".

• 2 Condiciones

- Nombre de programa: `ejercicio_3.xxx`
- Lenguaje libre.
- No usar tablas de Vigenère ya hechas.
- Implementar el cálculo usando:
$$\text{Cifrado} = (\text{ValorTexto} + \text{ValorClave}) \bmod 26$$
- Trabajar únicamente con funciones y operadores básicos.

• Puntos: 1.5 Puntos

Ejercicio 4 – Cifrado Rail Fence

• 1 Enunciado

Escribe un programa que **cifre** el texto `"CIFRADORAILFENCE"` usando **3 rieles**.

• 2 Condiciones

- Nombre de programa: `ejercicio_4.xxx`
- Lenguaje libre.
- No usar funciones de cifrado ya implementadas.
- Construir manualmente el patrón de rieles en zig-zag.
- Leer el resultado riel por riel para formar el texto cifrado.

• Puntos: 1.5 Puntos

• Objetivo

-  **Pensar en cómo hacerlo con el mínimo de instrucciones.**
-  **Aprovechar solo las características básicas del lenguaje.**
-  **Evitar cualquier función de cifrado de librerías externas.**
-  **Probar que el resultado sea reversible para descifrarlo.**

• Formato de salida

- i.  **Presentar el texto plano.**
- ii.  **Desplegar el texto cifrado.**
- iii.  **Exponer el texto descifrado.**
- iv.  **Comparar el texto plano con el texto descifrado.**