

Este ejercicio consiste en la implementación de un proyecto, que está formado por varios ficheros de código fuente (.cpp y .h), y su compilación mediante un fichero `makefile`. Se deberá implementar una clase `matriz` para representar las matrices cuadradas de dimensión  $n$  (utilizar una constante global para determinar la dimensión) con coeficientes enteros. Debe contener, al menos, los métodos miembro:

- Un constructor por defecto que genere la matriz cero (todas las componentes son cero).
- Una función miembro `get_component` para obtener una componente de la matriz.
- Una función miembro `set_component` que modifique una componente de la matriz.
- Una función miembro `imprimir` que muestre por pantalla la matriz.

Se pueden añadir otras funciones miembro que se consideren necesarias. Implementar las siguientes funciones externas a la clase para formar una biblioteca con la que operar con las matrices:

- Una función `suma` que sume dos matrices.
- Una función `producto` que multiplique dos matrices de la manera usual.
- Una función `traza` que devuelva la suma de los elementos de la diagonal.

Implementar otra biblioteca con las siguientes operaciones especiales:

- Una función `producto_especial` que multiplique dos matrices componente a componente (no de la forma usual). Por ejemplo,

$$\begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} * \begin{pmatrix} 1 & 3 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ 0 & 0 \end{pmatrix}$$

- Una función `twisted_mult` que multiplica dos matrices  $A$  y  $B$  como  $A \# B = APB$ , donde  $P$  es la matriz que tiene todas sus componentes a cero excepto la diagonal principal inversa que son unos. Por ejemplo,

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \# \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 1 \\ 1 & 1 \end{pmatrix}$$

Además se implementará un programa probando las funciones anteriores. Dividir el código en diferentes ficheros tal y como se ha explicado en clase y realizar un fichero `makefile` que compile todo el proyecto. Sigue el árbol usual de carpetas para el proyecto:

```
Matrices/  
|  
|----- bin/  
|----- src/  
|----- include/  
|----- obj/  
|----- lib/  
|  
|----- makefile
```