



Apellido y Nombre

DNI: _

Notificado:

Calificación :

Se dispone de un archivo de texto que ha sido codificado del siguiente modo:

Para las palabras con una cantidad par de caracteres se ha procedido a tomar para el primer carácter de la palabra el siguiente, para el segundo el siguiente del siguiente, y así sucesivamente. Teniendo en cuenta que el siguiente de la 'z' es la 'a' y el siguiente de la 'a' es la 'z'. Una vez modificados los caracteres en cuestión se ha invertido la palabra (p. ej.: **Mendez** -> **NgqghjF** -> **FjhqgN**).

En cuanto a las de una cantidad impar de caracteres se procedió primero a invertirla y luego modificarla de modo similar (p. ej.: **Perez** -> **zereP** -> **AguiU**).

Se requiere un proceso en que se genere el archivo decodificado, además de mostrar qué palabras se repiten tres o más veces ordenadas alfabéticamente indicando la palabra y la cantidad de veces que apareció en el texto.

Observaciones: Sólo se codificaron los caracteres de la **a** a la **zeta**, mayúsculas y minúsculas sin acento ni diéresis y tampoco la **ñ** ni la **Ñ**, considerando a cualquier otro carácter fuera de estos como separador de palabra (p. ej.: **Pérez** -> **Qézer** -> **QéAgu**, otro: **Méndez** -> **NéofhD** -> **NéDhfo**).

Escriba una class Caja con información privada para sus dimensiones.

Resuelva las sobrecargas de los operadores necesarios e imprescindibles para el correcto funcionamiento del siguiente trozo de código:

```
void calcular(int nUno, int nDos, int nTre, int nCua, int nCin)
{
    Caja      c0,                // caja de 1 x 1 x 1
              c1(nUno),          // cubo de nUno de arista
              c2(nUno, nDos),    // caja de base nUno x nDos y altura nDos
              c3(nTre, nCua, nCin);

    cout << "Valores iniciales:" << endl
         << "c0 = " << c0 << endl << "c1 = " << c1 << endl
         << "c2 = " << c2 << endl << "c3 = " << c3 << endl;

    c0 = nCin * c3;              // calcula nCin por cada dimensión
    c1 = c2++;                  // se incrementan en uno sus dimensiones
    c3 = --c2;                  // se decrementan en uno sus tres dimensiones

    cout << "Valores finales:" << endl
         << "c0 = " << c0 << endl << "c1 = " << c1 << endl
         << "c2 = " << c2 << endl << "c3 = " << c3 << endl;
}
```

Debe resolver la mínima cantidad de funciones miembro necesaria, y estas deben estar resueltas fuera de la declaración de la class.

EVALUACIÓN EN LABORATORIO

NOTA Ante cualquier duda consulte con los docentes.

Para la primera parte debería resolver las funciones indicadas en el proyecto con **_2** y valiéndose de una lista doblemente enlazada.

Salida por pantalla

Parte 1	
a	11
de	20
del	6
el	9
en	8
es	5
ese	3
est	3
la	12
las	6
les	3
Lo	3
los	7
lugar	3
M	4
n	8
no	8
Pero	3
por	6
pronombre	4
que	13
s	10
se	10
Sennett	4
un	4
una	3
y	8
Parte 2	
Valores iniciales:	
c0 = 1x1x1	
c1 = 6x6x6	
c2 = 6x5x5	
c3 = 4x3x2	
Valores finales:	
c0 = 8x6x4	
c1 = 6x5x5	
c2 = 6x5x5	
c3 = 6x5x5	