

◁ Ejercicio 1 ▷ Palabras seudosimétricas

[2.5 puntos]

Una palabra se define como **seudosimétrica** si, cuando se parte por la mitad, da como resultado dos palabras que tienen las mismas letras con las mismas frecuencias. Si la longitud de la palabra es un número impar, la decisión se toma sin considerar la letra central. Por ejemplo gaga es una palabra *seudosimétrica*. Las dos mitades ga y ga tienen los mismos caracteres con la misma frecuencia. Otros ejemplos son rotor, aabccaba, xyzxy. La palabra abbaab NO es *seudosimétrica*.

En este ejercicio, una palabra se representa mediante la clase SecuenciaCaracteres (cuya descripción aparece en la tabla incluida más abajo). Supondremos que los caracteres almacenados pertenecen al conjunto de letras minúsculas consecutivas del alfabeto inglés (abcdefghijklmnopqrstuvwxyz).

Tareas a realizar

a) Implemente los siguientes métodos en la clase:

- **(0.5)** SubSecuencia que devuelve una secuencia de caracteres compuesta por un máximo de num caracteres, siendo el primero el que se encuentra en la posición inicio. Deberá considerar todas las situaciones posibles que puedan ocurrir con los valores de inicio y num y el tamaño de la secuencia, intentando siempre devolver una secuencia de caracteres (posiblemente vacía).
- **(1.25)** EsSeudoSimetrica que comprueba si la secuencia representa una palabra *seudosimétrica*.
- **(0.25)** En todos los métodos que escriba debe explicar claramente qué hacen, utilizando comentarios. Indique el tipo del resultado, así como los parámetros de entrada necesarios (si los hubiera). **Puede añadir todos los métodos que considere oportunos.** Justifique la necesidad y el ámbito (público, privado) de los mismos.

b) **(0.5)** Dispone de un objeto de la clase SecuenciaCaracteres llamado nombre que contiene una palabra formada por la concatenación de su nombre y primer apellido (todo en minúsculas, sin tildes ni espacios). Explique con claridad el funcionamiento de su método EsSeudoSimetrica. Por ejemplo, si su nombre es "Manuel Alonso Anguita", el objeto nombre contendrá 12 letras: manuelalonso.

SecuenciaCaracteres
<i>Datos miembros privados:</i>
static const int TAMANIO = 100 char vector_privado[TAMANIO] int total_utilizados
<i>Métodos públicos disponibles:</i>
SecuenciaCaracteres() SecuenciaCaracteres(string) int TotalUtilizados() int Capacidad() void Aniaide(char nuevo) char Elemento(int indice)

◁ Ejercicio 2 ▷ Tablero y sus Propiedades

[2.5 puntos]

Un tablero es un cuadrado de dimensiones  $N \times N$  donde se ubican valores enteros. Se dice que un tablero es **califragilístico** si sus filas, columnas y sus dos diagonales tienen sumas *diferentes*. Además, si un tablero es *califragilístico* entonces puede cumplir una propiedad adicional: ser **supercalifragilístico**. Para ello, se necesita que con los valores de

las sumas en filas, columnas y diagonales se pueda formar una secuencia creciente completa. Es decir, dada la secuencia  $S = \{s_1, s_2, \dots, s_k\}$ , decimos que es creciente y completa si  $s_{i+1} = s_i + 1$ .

A continuación se muestran dos ejemplos. A la izquierda, un tablero *califragilístico*. Podemos ver que las filas, columnas y diagonales dan lugar a sumas diferentes. A la derecha, un tablero *supercalifragilístico*. Todas las sumas son diferentes (es *califragilístico*) y además dan lugar a una secuencia creciente completa de valores  $\{30, 31, 32, \dots, 39\}$ .

$$\begin{bmatrix} & 9 & 8 & 7 & \rightarrow & 24 \\ & 2 & 1 & 6 & \rightarrow & 9 \\ & 3 & 4 & 5 & \rightarrow & 12 \\ \swarrow & \downarrow & \downarrow & \downarrow & \searrow & \\ 11 & 14 & 13 & 18 & 15 & \end{bmatrix}$$

$$\begin{bmatrix} & 15 & 2 & 12 & 4 & \rightarrow & 33 \\ & 1 & 14 & 10 & 5 & \rightarrow & 30 \\ & 8 & 9 & 3 & 16 & \rightarrow & 36 \\ & 11 & 13 & 6 & 7 & \rightarrow & 37 \\ \swarrow & \downarrow & \downarrow & \downarrow & \downarrow & \searrow & \\ 34 & 35 & 38 & 31 & 32 & 39 & \end{bmatrix}$$

Dado un tablero, se quiere implementar un programa para saber si es *califragilístico* y *supercalifragilístico*. Para ello, dispone de dos clases: Tablero y SecuenciaEnteros que puede utilizar.

Tablero
<i>Datos miembros privados:</i>
static const int MAX = 15 int t[MAX][MAX] int dimension
<i>Métodos públicos disponibles:</i>
Tablero() Tablero(int dimension) int GetDimension() int Elemento(int fil, int col) SecuenciaEnteros Fila(int fil) SecuenciaEnteros Columna(int col)

SecuenciaEnteros
<i>Datos miembros privados:</i>
static const int TAMANIO = 100 int vp[TAMANIO] int utiles
<i>Métodos públicos disponibles:</i>
SecuenciaEnteros() int TotalUtilizados() int Capacidad() void Aniaide(int nuevo) int Elemento(int indice) bool Existe(int valor) void OrdenaCreciente() int Minimo() int Maximo()

## Tareas a realizar

a) Implemente los siguientes métodos:

- **(0.5)** EsCalifragilistico que comprueba si el tablero cumple la propiedad de ser *califragilístico*
- **(0.75)** EsSuperCalifragilistico que comprueba si el tablero cumple la propiedad de ser *supercalifragilístico*.
- **(0.75)** En todos los métodos que escriba debe explicar claramente qué hacen, utilizando comentarios. Indique el tipo del resultado, así como los parámetros de entrada necesarios (si los hubiera). **Puede añadir todos los métodos que considere oportunos.** Justifique la necesidad y el ámbito (público, privado) de los mismos.

b) **(0.5)** Explique con claridad como funciona su método EsSuperCalifragilistico sobre un dato Tablero concreto. Construya un tablero de 3 x 3 utilizando los dígitos de su DNI y reemplazando la letra por el dígito 1. Por ejemplo, si su DNI fuera 12345678K, entonces el contenido del tablero asociado sería:

```
1 2 3
4 5 6
7 8 1
```

Escriba el tablero asociado a su DNI y úselo para la explicación.