



## Grado en Ingeniería Informática y Grado en Estadística Fundamentos de Programación

Examen de convocatoria ordinaria. 12 de enero de 2024

Apellidos \_\_\_\_\_

Nombre \_\_\_\_\_ Grupo \_\_\_\_\_

--	--	--	--	--	--

DNI y Firma

Duración del examen: 3 horas.

Empezar cada problema en una cara distinta.

Poner nombre y apellidos en todas las páginas.

Las hojas del enunciado también deben entregarse.

Se valorará la presentación y la claridad en la exposición.

Se valorarán los criterios de calidad considerados en la asignatura y la adecuación de las estructuras utilizadas en cada problema.

No se calificarán las respuestas escritas a lápiz.

1. Elaborar los métodos Java siguientes, prestando especial atención a la documentación y en particular a sus precondiciones. Para cualquier apartado puede usar un apartado anterior, aunque no lo tenga resuelto.

- a) [1 pto] Dado un vector de caracteres, cuyos valores solamente pueden ser 'o', 'x' ó '-', devuelve un valor cierto o falso según el vector contenga o no tres oes o tres equis consecutivas.
- b) [0,5 ptos] Dada una matriz bidimensional de caracteres, con las mismas características del vector anterior, y un entero n, devuelve un valor cierto o falso según la *fila n* contenga o no tres oes o tres equis consecutivas.
- c) [0,5 ptos] Dada una matriz bidimensional de caracteres, con las mismas características del vector anterior, y un entero n, devuelve un valor cierto o falso según la *columna n* contenga o no tres oes o tres equis consecutivas.

2. [4 ptos] El método de cifrado César consiste en sustituir cada letra del abecedario por una letra desplazada un número determinado de posiciones (clave). Por ejemplo, al cifrar la cadena **IBM** con la clave **-1** se obtiene la cadena **HAL**. Denominaremos método de cifrado *CesarASCII* a la aplicación de esta forma de encriptado sobre cualquier carácter de la tabla ASCII (no solo las letras). Con este método, al cifrar la cadena **examen** con la clave 4 se obtiene la cadena **i|eqir**.

Elaborar un programa Java que lea palabras de teclado, formadas únicamente por letras minúsculas del alfabeto latino (sin tildes, ni diéresis ni eñes). Para cada palabra correcta leída, el programa debe obtener del usuario un valor entero no nulo en el intervalo [-65, 65] para garantizar que la codificación funciona sin salirse de rango. Ese valor será la clave con la que el programa cifrará esa cadena siguiendo el método *CesarASCII*. Como resultado, el programa debe escribir en un fichero de texto, de nombre **cifrado.txt**, cada clave y la cadena cifrada obtenida (separados por un espacio en blanco) a razón de una pareja (clave – cadena cifrada) por línea. El programa debe terminar cuando el usuario introduzca una cadena no vacía que no esté formada exclusivamente por letras minúsculas del alfabeto latino.

Suponga importadas las clases necesarias.

**RECORDATORIO:** Para declarar y abrir en modo lectura un fichero de texto:

```
Scanner <id_fich> = new Scanner (new File(<nombre_fich>)) ;
```

Al abrir el fichero se puede producir la excepción **FileNotFoundException**

Para declarar y abrir en modo escritura un fichero de texto:

```
PrintWriter <id_fich> = new PrintWriter (new FileWriter (<nombre_fich>)) ;
```

Al abrir el fichero se puede producir la excepción **IOException**.

3. [2 ptos] Para almacenar las cadenas cifradas con el método *CesarASCII* se utilizan registros de tipo **regCIFRADO**, que contienen tres campos: uno que almacena la cadena, otro que almacena la clave de cifrado (entero) y otro que guarda un valor que determina si la cadena almacenada está cifrada o es la original. Se pide:

- a) Definir el tipo de datos **regCIFRADO**
- b) Elaborar un método Java que, a partir de un vector de registros de tipo **regCIFRADO**, y una clave localice en el vector todos los elementos que tienen esa clave y para cada uno de ellos:
  - 1) Si la cadena está cifrada la cambie por la cadena descifrada.
  - 2) Si la cadena es la original la cambie por la cadena cifrada.En ambos casos, actualice el campo de cifrado.

4. [1 pto] Una *secuencia de enteros dinosaurio* de nivel 0 es, por definición, la secuencia vacía. Y para cualquier  $n > 0$ , la secuencia dinosaurio de nivel  $n$  es la formada por dos secuencias dinosaurio de nivel  $n-1$  separadas por el entero  $n$ .

Por ejemplo, las *secuencias de enteros dinosaurio* de nivel 1, 2 y 4 son, respectivamente:

Nivel 1: 1

Nivel 2: 1 2 1

Nivel 4: 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

Elabore un **método recursivo** Java que escriba en pantalla la secuencia dinosaurio del nivel de su argumento.

5. [1 pto] Se tiene **list** (de tipo **Nodo**), una referencia a una **lista dinámica** que contiene números enteros en su campo **dato**. Crear un método Java que elimine el elemento que ocupa la posición **pos** en **list** (el primer nodo de la lista ocupa la posición 1). Si la lista tiene menos de **pos** elementos, el método no debe modificarla. Supóngase dada la siguiente definición:

```
public class Nodo {
    int dato;
    Nodo sgte;
    // constructores, etc.
}
```