Departamento de Informática
Universidad de Valladolid

Grado en Ingeniería Informática y Grado en Estadística
Fundamentos de Programación

Examen de convocatoria ordinaria. 5 de febrero 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 se puede usar un apartado anterior, aunque no lo tenga resuelto.

a) **[1 pto]** Un método Java que, dados dos vectores de enteros, devuelve un valor cierto o falso según los vectores sean o no iguales.

b) **[0,5 ptos]** Un método Java que, dada una matriz bidimensional de enteros, y dos enteros `m` y `n`, devuelve un valor cierto o falso según las filas `m` y `n` coincidan o no.

c) **[0,5 ptos]** Un método Java que, dada una matriz bidimensional de enteros, y dos enteros `m` y `n`, devuelve un valor cierto o falso según las columnas `m` y `n` coincidan o no.

d) **[1 pto]** Dada una matriz bidimensional de elementos de tipo `double` y un entero `n`, suma todos los elementos de la línea inclinada `45°` que comienza en el elemento de la posición `(n, 0)` de la matriz.

Por ejemplo: Dada la siguiente matriz, y siendo `n = 2:`

1	2	3	4
8	6	4	2
10	30	50	70

El valor devuelto sería: `19 (10+6+3)`

2. **[4 ptos]** Para estudiar ciertas propiedades de un historial de mensajes, que se encuentra en el fichero de texto `entrada.txt`, es necesario almacenar las siguientes informaciones para cada longitud posible de los mensajes (desde 1 hasta 150, que es el máximo permitido):

- El número de mensajes que tienen esa longitud.
- El número de orden del primer mensaje que tiene esa longitud (el primer mensaje es el número 1).
- El primer mensaje de esa longitud.
- El porcentaje de mensajes de esa longitud en el total del texto (número real).

La estructura idónea para realizar este almacenamiento es un vector de registros, cada uno de los cuales almacena los cuatro valores citados para cada longitud. Esta estructura **debe definirse.**

Elaborar un programa Java que, una vez almacenada esta información, permita que el usuario solicite sucesivamente (hasta que el usuario pulse 0) información sobre longitudes de mensajes (introduciendo un número entero cuyo valor debe estar en el rango permitido) y el programa le proporcione en pantalla los datos correspondientes.

3. **[1 pto]** Se dice que una cadena de caracteres es *contrapalindrómica* si cada carácter es distinto del que ocupa la posición simétrica a él respecto al centro, si es que lo hay. Por ejemplo, **`abcac`** lo es, porque el primer carácter es distinto del último, y el segundo distinto del penúltimo. Considérese que la cadena vacía es *contrapalindrómica*.

Elabore un **método recursivo** Java que determine si una cadena es o no *contrapalindrómica*.

NOTA:

`public String substring(int beginIndex, int endIndex)`

Devuelve la subcadena formada por los caracteres desde **`beginIndex`** hasta **`endIndex-1`** (ambos inclusive) de una cadena dada. Asi, su longitud es endIndex-beginIndex

Ejemplos:

`"hamburger".substring(4, 8)` devuelve `"urge"`

`"smiles".substring(1, 5)` devuelve `"mile"`

4. **[2 ptos]** Elaborar un método Java que determine cuántas letras mayúsculas del alfabeto latino contiene una cadena de caracteres. Utilizando ese método y el del ejercicio anterior, implemente un programa Java que lea de teclado cadenas de caracteres, hasta que se introduzca una no *contrapalindrómica*, y escriba en un fichero, cuyo nombre debe solicitar al usuario, el número de letras mayúsculas contenidas en cada cadena leída incluida la última (que no es *contrapalindrómica*), a razón de uno por línea.

NOTA: Puede usarse el método del ejercicio anterior, aunque no se haya implementado.

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.`**