

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

Examen convocatoria ordinaria. 14 de enero de 2022.

Apellidos \_\_\_\_\_

DNI y Firma \_\_\_\_\_

Nombre \_\_\_\_\_ Grupo: **T2**

--	--	--	--	--

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 valorará la calidad de las soluciones.

No se calificarán las respuestas escritas a lápiz.

No se deben entregar las hojas que se usen como borrador.

1. [4 ptos] Elaborar un programa Java que simule el bombo del juego del bingo. Para ello:

- El programa debe generar y mostrar en pantalla valores enteros aleatorios entre 1 y 90 (sin que se repitan) hasta que el usuario teclee la palabra **bingo** (*cante bingo*) o hasta que se hayan generado todos los valores posibles.
- Si el usuario teclea la palabra *bingo*, el programa debe mostrar en pantalla los números que han salido hasta ese momento para que se pueda comprobar que el bingo es correcto.
- Si el bingo es correcto, el usuario avisará al programa introduciendo un **1** y el programa terminará.
- Si el bingo no es correcto, el usuario tecleará cualquier valor entero distinto de **1** y el programa deberá continuar emitiendo números.
- En cualquier caso, el programa finalizará de forma automática cuando se hayan generado todos los valores entre 1 y 90.

2. [4 ptos] Una consultoría guarda los datos del personal de una empresa de los últimos 22 años. Para ello, usa una matriz en la que cada columna representa un puesto de trabajo y cada fila un año (la fila 0 sería el año 2000, la 1 el año 2001 y así sucesivamente). Los elementos de la matriz contienen los datos de cada trabajador, para lo que utiliza registros de tipo **regTRAB**, que contienen tres campos: uno de tipo **String** que guarda el nombre de ese trabajador, otro de tipo **doublé** para almacenar el sueldo y otro de tipo **char** que almacena el sexo (**M** para mujer y **H** para hombre). Así, el elemento **[i][j]** de la matriz contiene los datos del empleado que ocupaba el puesto de trabajo **j** el año **2000+i**. Dadas estas condiciones, se pide:

- Definir el tipo de datos **regTRAB**.
- Definir la matriz.
- Elaborar un método Java que, a partir de una matriz como la descrita, un determinado año entre 2000 y 2022 (**anyo** de tipo **int**) y un valor entero positivo que determina un puesto de trabajo (**pst** de tipo **int**), escriba en el fichero de texto **datos.txt** cinco líneas en las que debe aparecer la siguiente información:
  - Línea 1: Media total del sueldo de los trabajadores que son mujeres.
  - Línea 2: Media total del sueldo de los trabajadores que son hombres.
  - Línea 3: Diferencia entre las dos medias anteriores.
  - Línea 4: Media del sueldo de todos los trabajadores del año **anyo**.
  - Línea 5: Media del sueldo de todos los trabajadores que han ocupado el puesto **pst**.

**RECORDATORIO:** 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. [1 pto] Sean las cadenas de caracteres *imaginarias* aquellas que:

- Están formadas exclusivamente por letras minúsculas del alfabeto inglés.
- Tienen longitud par.
- Y además, los caracteres en posiciones simétricas respecto al centro de la cadena son simétricos respecto al centro del abecedario.

Por ejemplo, la cadena "**bolazoly**" es imaginaria porque las letras '**b**' e '**y**' ocupan respectivamente la segunda posición desde el principio y desde el final del abecedario, y '**l**' y la '**o**' lo mismo para la duodécima posición y el par '**a**', '**z**' para la primera.

La **cadena vacía** también es *imaginaria*.

Escriba y documente una función recursiva, que pueda utilizarse en un programa para determinar si una cadena es o no *imaginaria*.

**PISTA:** Dada la distribución del abecedario inglés en el código ASCII, para los caracteres simétricos respecto al centro del abecedario se cumple que la suma de sus códigos es constante. Es fácil ver que, para que una cadena sea *imaginaria*, esa cadena sin el primer y el último carácter también tiene que ser *imaginaria*.

**RECORDATORIO:** La documentación de la función **substring**, de la clase **String**, es la siguiente:

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

Returns a new string that is a substring of this string. The substring begins at the specified beginIndex and extends to the character at index endIndex - 1. Thus the length of the substring is endIndex-beginIndex.

Examples: "hamburger".substring(4, 8) returns "urge". "smiles".substring(1, 5) returns "mile"

**Parameters:**

beginIndex - the beginning index, inclusive.

endIndex - the ending index, exclusive.

**Returns:**

the specified substring.

**Throws:**

**IndexOutOfBoundsException** - if the beginIndex is negative, or endIndex is larger than the length of this String object, or beginIndex is larger than endIndex.

4. [1 pto] Crear un método Java que, dada **list** (de tipo **Nodo**), una referencia a una **lista dinámica** que contiene números enteros en su campo **dato**, devuelva la posición de la última aparición en esa lista de un número entero dado. Para calcular la posición se supone que el primer nodo es el 1, el segundo el 2, y así sucesivamente. Si el número no estuviese en la lista, el método devolverá el valor 0.

Supóngase definida la clase **Nodo** como:

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