

	<p align="center"> 2º EVALUACION 1º DAW CURSO 2021 - 2022 Programación 1º Examen del 2º Trimestre 15/02/2022 – (15:00 a 18:00 h) </p>	
Nombre, apellidos y DNI:		Calificación

Puedes usar internet sin limitaciones, salvo para comunicarte con tus compañeros de clase durante el examen, estaremos vigilando, y la comunicación supondrá el suspenso. Solo puedes usar el ordenador, ni teléfonos ni smartwatch están permitidos. El examen se entregará a través de Edmodo en una entrega que se abrirá antes del final del examen. Si no te sale uno, pasa a hacer el siguiente, y luego vuelves a él: No pierdas el tiempo innecesariamente.

Realiza todo el examen en el mismo proyecto, pon TODOS los métodos en un archivo distinto al archivo donde guardes el main.

(-1,5 Puntos) Si algún método está en el mismo archivo que el main

(-0,2 Puntos) Por cada variable, método o clase que no se nombre con una notación correcta (UpperCamelCase o lowerCamelCase)

(-0,1 Puntos) Por cada variable que ocupe un tamaño mayor (o insuficiente) del necesario para almacenar los valores que puede tener dentro.

1 - **(0,75 Puntos)** Desde el main, pide por teclado un número, y construye un array de boolean de 20 posiciones, en la que se coloque en cada posición true si el índice es múltiplo de ese número, y false si no lo es. Por ejemplo, si se introduce el valor cuatro, estarían a true las posiciones 0 (4*0), 4 (4*1), 8 (4*2), 12 (4*3) y 16 (4*4):

```
{true, false, false, false, true, false, false, false, true, false, false, false, true, false, false, false, true, false, false, false}
```

Imprime el array por pantalla.

2 - **(1,5 Puntos)** Desde el main, pide dos números por teclado (a y b), y asegurate de que los dos números son distintos: Si son iguales, indicarlo por pantalla y volver a pedirlos tantas veces como sea necesario hasta que sean distintos. Después, crea una matriz `int [][]` matriz de tamaño `a * b`, y rellena la posición `matriz[0][0]` con un 1. Recorre todas las demás posiciones, poniendo en cada una, el doble del número anterior. Por ejemplo, en una matriz de 3*4, quedaría:

1	2	4	8
16	32	64	128
256	512	1024	2048

3 - **(0,25 Puntos)** Pide dos números por teclado. El primero será el número de filas de una matriz, y el segundo el número de columnas. Este ejercicio no incluye crear la matriz, eso se hace en el 5

4 - **(1 Punto)** Crea una función crearSopa, que reciba por argumentos dos números (numeroFilas y numeroColumnas), y devuelva una matriz de char. En su interior, usará esos dos argumentos para crear una matriz de char con el número de filas y columnas indicado en los valores de los argumentos. Rellenará todas las posiciones de esa matriz con letras aleatorias entre la a y la z. Devuélvela.

5 - **(0,5 Puntos)** En el main, tras pedir los dos números del ejercicio 3, llama a la función crearSopa usándolos como argumentos. Almacena el valor devuelto por la función en una variable llamada sopaLetras. Tras lo anterior, en el main, pide un String por teclado, que será la palabra a introducir en la sopa de letras, y guarda el resultado en una variable llamada palabra. Pide también dos números, introdúcelos en dos variables llamadas posX y posY.

6 - **(2,5 Puntos)** Crea una función llamada introducirPalabra. Recibirá como argumentos una matriz de char (sopa), un String (palabra) y dos enteros (posX y posY). Devolverá un booleano. Esta función intentará colocar la palabra, letra a letra en la matriz, en dirección horizontal, a partir de la posición (posX,posY). Si puede hacerlo y cabe la palabra entera, lo hará y devolverá true. Si no se puede hacer, porque no cabe la palabra entera, devolverá false, sin cambiar nada en la matriz.

Por ejemplo, si tenemos la matriz:

```
a z x t v
v g f d s
y u g l p
u b c a q
m v c z t
```

La palabra es “hola”, posX vale 2, y posY vale 1, colocará la palabra así, devolviendo true:

```
a z x t v
v g f d s
y h o l a
u b c a q
m v c z t
```

Si posX valiese 2, y posY 3, no haría nada, devolviendo false, porque en ese caso la palabra no cabe.

7 - **(0,25 Puntos)** En el main, con todo lo que habías pedido por teclado, llama a la función introducirPalabra. Si devuelve false, no hacer nada. Si devuelve true, imprimir la matriz con la palabra insertada por consola.

8 - **(2,5 Puntos)** Haz que tu programa admita los argumentos de programa -filas, -columnas, -palabra, -posX y -posY. Todos vendrán seguidos en la línea de comandos de un número, excepto -palabra, que vendrá seguido de una palabra. Estos argumentos podrán estar o no estar, o ponerse en cualquier orden, y el programa debería funcionar. Un ejemplo de la llamada desde consola sería:

```
java -jar miPrograma.jar -columnas 12 -filas 8 -posX 5 -palabra hola
```

Si recibe el argumento -filas, ya no pedirá el número de filas por teclado en el main, como hacías en el ejercicio 3, y utilizarás el valor dado por argumentos. A no ser que ese valor no esté entre 5 y 127. En ese caso, pide el valor por teclado de todas formas, como en el ejercicio 3. el argumento -columnas hará lo mismo, pero con las columnas. Ocurrirá lo mismo -posX, -posY y -palabra del ejercicio 5. Si alguno de ellos está en argumentos, no los pedirá por teclado inicialmente. Pero si los pedirá si utilizándolos, la función introducirPalabra devuelve false. No hagas argumentos para los valores que se piden por teclado en los ejercicios 1 y 2. Estos se pedirán por teclado de todas formas

9 - **(0,75 Puntos)** Documenta correctamente todo tu código fuente. No tienes por qué tener todo el examen terminado para hacer este ejercicio. Si lo que tienes está bien documentado, este ejercicio tendrá la máxima nota.

DEPENDENCIAS DE EJERCICIOS

El 5 depende del 3

El 5 depende del 4, pero si no te sale la función del ejercicio 4, lo puedes hacer sin llamarla

El 7 depende del 6

El 8 depende del 3 y del 5

CRITERIOS DE CALIFICACIÓN

Los ejercicios que los alumnos no hayan hecho, o estén comentados, o tan errados que no tienen nada que ver con lo pedido, no contarán para la nota. Cada ejercicio que hayan hecho, o al menos intentado los alumnos, cuenta su nota completa, y se van restando esta cantidad de puntos según los fallos cometidos:

- **-0,5 puntos** Si hay algún error que impide compilar o que hace reventar la ejecución del programa, o lo mete en una situación de bloqueo permanente, o bucle infinito.
- **-0,25 puntos** Si hay algún error que puede ocasionar fallos en ciertas situaciones muy específicas, o hacen que la salida del programa sea errónea aunque este termine sin reventar.
- **+0,25 puntos** Es lo que vale en el 2 tener al menos los numeros pedidos distintos
- **+0,75 puntos** Es lo que vale en el 2 tener todo menos la cuenta bien hecha, y vas de 1 en 1.
- **-0,75 puntos** La matriz de crearSopa no se rellena en el ejercicio 4
- **-0,5 puntos** La matriz de crearSopa no se rellena con letras correctas en el ejercicio 4
- **-0,35 puntos** La condición para los true-false es incorrecta en el ejercicio 1 , pero el resto está bien.
- **-0,25 puntos** Ejercicio 2: empiezas en 2, y no en 1 la matriz.
- **-0,5 puntos** Si no se vuelven a pedir ninguno de los datos del ejercicio 8 cuando son incorrectos por argumentos
- **-0,5 puntos** Si en el 6 modificas la matriz si la palabra no cabe
- **-0,75 puntos** Si en el 6 modificas más posiciones de la necesaria, pero al menos en algunas situaciones la palabra se pone bien
- **-1,25 puntos** Si en el 6 modificas más posiciones de la necesarias, y no se pone bien la palabra
- **-0,5 puntos** Si en el 6 no devuelves false en caso de que la matriz no quepa
-
- **-0,25 puntos** Si no se vuelven a pedir algunos datos del ejercicio 8 cuando son incorrectos por argumentos
- **-0,5 puntos** En el 8 por cada argumento no hecho
- **+2 puntos** Es lo que vale en el 6 Si haces todo bien menos elegir el valor que insertar