

Examen 2 Programación (variables, arrays (uni y bidi), funciones)	
Nombre del alumno/a:	Calificación

1._ (2 puntos) Números pares repetidos

En esta pregunta se pide realizar un bucle en el que se vayan introduciendo números enteros. Nos saldremos del bucle en el momento en que introduzcamos un número igual al anterior y que éste sea par. Al salir del bucle me deberá mostrar la suma de los números introducidos. Recordad la orden para limpiar el buffer por si os hiciera falta. Un ejemplo de salida por pantalla sería el siguiente:

```
put - Run (EXA2_Nov_23_24) x
cd C:\Users\Rut\Documents\NetBeansProjects\EXA2_Nov_23_24; "JAVA_HOME=C:\\Pro
Introduce números y sal del bucle sólo si repites dos números pares seguidos
Introduce un número: 2
Introduce un número: 2
La suma es: 4
```

PUNTUACIÓN	
Bucle correcto	1.7 ptos
Suma correcta	0.3 ptos

2._ (2 puntos) Función buscaNum

Esta función se encargará de buscar un número dentro de un array y decir si este número está en el array o no. **USAD UN BUCLE WHILE, PROHIBIDO CUALQUIER BUCLE FOR**

En la función ejercicio2() ya hay creado un array y he generado un número aleatorio entre 1 y 25 incluido. Hay que llamar a la función buscaNum para saber si este número está dentro del array o no. El resultado se mostrará por pantalla.

Un ejemplo de salida por pantalla sería:

```
[2, 4, 23, 8, 13, 9, 10, 5]      [2, 4, 23, 8, 13, 9, 10, 5]
El número 8 está en el array     El número 17 NO está en el array
```

PUNTUACIÓN	
Firma función correcta	0.7 ptos
Llamada a la función correcta	0.3 ptos
Función correcta con while	1 ptos
Si usáis for	-1 pto

3._ (2,8 puntos) Función creaArray

Esta función se encargará de pedirle al usuario un tamaño para el array (suponemos que será mayor que 1). Luego creará y devolverá un array de este tamaño y rellenará la primera parte (hasta la mitad) de números positivos generados aleatoriamente entre 1 y 100. El resto del array (desde la mitad hasta el final) lo rellenará de números negativos generados aleatoriamente entre -100 y -1.

Si el array tiene un tamaño impar, se meterá un número negativo más que de positivos.

Un ejemplo de salida por pantalla sería:

```
Por favor, introduzca un tamaño: 3
[50, -12, -6]
```

PUNTUACIÓN	
Firma función correcta	0.5 ptos
Creación correcta del array	1.3 ptos

4._ (3,2 puntos) Función divideArray

Esta función creará un array como el del ejercicio anterior, llamando a la función creaArray. Luego se encargará de dividir este array en un array de números positivos y otro de números negativos. La función devolverá un array bidimensional en el que, en la primera posición habrá un array con todos los números positivos y en la segunda posición habrá un array con todos los números negativos.

A la hora de dividir el array original haremos uso de la función

'Arrays.copyOfRange(array,from,to)' que se encarga de devolverme un array con una copia de los datos del array original desde las posición **from** hasta la posición **to** incluida.

He aquí un pantallazo de la documentación oficial de Oracle sobre esta función de Java.

copyOfRange

```
public static int[] copyOfRange(int[] original,
                               int from,
                               int to)
```

Copies the specified range of the specified array into a new array. The initial index of the range (*from*) must lie between zero and *original.length*, inclusive. The value at *original[from]* is placed into the initial element of the copy (unless *from == original.length* or *from == to*). Values from subsequent elements in the original array are placed into subsequent elements in the copy. The final index of the range (*to*), which must be greater than or equal to *from*, may be greater than *original.length*, in which case 0 is placed in all elements of the copy whose index is greater than or equal to *original.length* - *from*. The length of the returned array will be *to* - *from*.

Un ejemplo de la salida por pantalla sería:

```
Por favor, introduzca un tamaño: 9
El array completo es [15, 23, 19, 96, -2, -83, -76, -76, -50]
15 23 19 96
-2 -83 -76 -76 -50
```

PUNTUACIÓN	
Firma función correcta	0.75 ptos
División del array con copyOfRange	1.2 ptos
Creación del array bidimensional	1.25 ptos

Criterios de evaluación del examen (en cada ejercicio se tocan todos)			
RA1.	CE a, b, c, d, e, f, g, h, i	RA3.	CE a, b, c, d, e, f, g

Para cada ejercicio se seguirá la siguiente rúbrica:

Porcentaje	Comentario
100%	El programa/clase/función implementado se ajusta perfectamente a las especificaciones/requerimientos. La ejecución es correcta para todas las circunstancias. El código es legible (debidamente comentado, estructurado, modular y con nombres de variables/funciones/clases significativos)
75%	El programa/clase/función implementado se ajusta perfectamente las especificaciones/requerimientos. La ejecución es correcta para todas las circunstancias. El código no es lo suficientemente legible (comentarios, estructura, nombres, etc.)
50%	El programa/clase/función implementado se ajusta mayoritariamente las especificaciones/requerimientos y el código es legible, pero en algunos casos no funciona correctamente o tiene algún error menor.
25%	El programa/clase/función implementado no se ajusta totalmente las especificaciones/requerimientos, el código no es legible y no funciona para la mayoría de los casos.
0%	El programa/clase/función implementado no se ajusta a las especificaciones, no resuelve el problema de forma algorítmica o atenta contra la programación estructurada (saltos indiscriminados, uso de break, continue o goto)

Una vez terminado, el código deberá subirse a moodle en la actividad creada a tal efecto