

## EXAMEN PRIMER TRIMESTRE

### 9 DE DICIEMBRE DE 2022

#### Instrucciones de entrega

Una vez realizado el examen, comprimirás todas las carpetas de los proyectos Netbeans en un único archivo en formato ZIP que enviarás para su corrección.

El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

**Apellido1\_Apellido2\_Nombre\_PROG\_ExamenDiciembre.zip**

#### Programas/métodos/funciones

**Nota:** Para cada ejercicio realiza un **Main** que pruebe el correcto funcionamiento del código desarrollado.

#### 1. (3 puntos) Tirada de datos

Crea un proyecto en Netbeans llamado **TiradaDados** en donde se implementará un método que reciba un array de 5 números enteros que simule la tirada de 5 dados. La función debe comprobar que el array tiene 5 elementos (ni más, ni menos), y además, que esos números se corresponden con las caras del dado, es decir, tendrán los valores 1, 2, 3, 4, 5 o 6.

El método/función devolverá:

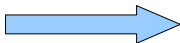
- -1 si la tirada no es válida (no cumple con los criterios anteriores).
- 0 si no hay ninguna pareja.
- 1 si hay una **Pareja** (dos dados iguales).
- 2 si hay **Dobles Parejas** (dos parejas de dos dados iguales).
- 3 si hay **Trío** (tres dados iguales).
- 4 si hay **Full** (un trío y una pareja).
- 5 si hay **Poker** (cuatro dados iguales).
- 6 si hay **Repoker** (cinco dados iguales).

**Orientaciones sobre cómo realizar la implementación.** Este ejercicio se puede plantear usando un array auxiliar en dónde sumemos cuántos dados de cada tipo hay.

0	1	2	3	4	5
Cantidad de 1	Cantidad de 2	Cantidad de 3	Cantidad de 4	Cantidad de 5	Cantidad de 6

Así por ejemplo para un array que simula la siguiente tirada de dados:

0	1	2	3	4
3	4	3	5	5



Se podría crear el siguiente array con la suma de cada tipo:

0	1	2	3	4	5
0	0	2	1	2	0

Esto es así puesto que la posición 0 guardamos el número de dados con valor 1, en la posición 1 el número de dados con valor 2, etc. En este ejemplo la posición 3 del array tiene el valor 1 porque hay solo un dado con el valor 4. Una vez tenemos el array auxiliar ya podríamos recorrerlo para decidir qué jugada hay (en este caso sería dobles parejas porque hay dos 3 y dos 5).

## 2. (3 puntos) Criba de Eratóstenes

La Criba de Eratóstenes es un algoritmo que permite hallar todos los números primos menores a un número dado (de manera más eficiente). Crea un proyecto en Netbeans llamado **CribaEratostenes** en dónde se implementarán los siguientes dos métodos:

- **(2 puntos) public static boolean[] cribaEratostenes(int n).** Devolverá un array de booleanos de tamaño  $n$  marcando como *true* todas las posiciones que sean números primos. Por ejemplo para  $n = 20$ . Devolverá:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
false	false	true	true	false	true	false	true	false	false	false	true	false	true	false	false	false	true	false	true

Este método deberá realizar la Criba de Eratóstenes por lo que debe funcionar de la siguiente manera:

- Se deberá declarar un array de booleanos con  $n$  elementos (inicializados todos a *true*, excepto el 0 y el 1)
- Se toma el primer elemento marcado como *true* y se marcarán como *false* todos sus múltiplos.
- Se continúa con el siguiente elemento marcado como *true* y se marcan como *false* todos sus múltiplos.
- Se repite el proceso mientras el cuadrado del elemento a evaluar sea menor que  $n$ .

**Ejemplo de funcionamiento.** Situación inicial para  $n = 20$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
false	false	true	true	true	true	true	true	true	true	true	true	true	true	true	true	true	true	true	true

Se toma el primer elemento marcado como *true* (en este caso el 2, cuyo cuadrado es menor que 20) y se marcan como *false* todos sus múltiplos (en este caso 4, 6, 8, 10, 12, 14, 16 y 18).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
false	false	true	true	false	true	false	true	false	true	false	true	false	true	false	true	false	true	false	true

Se toma el siguiente elemento marcado como *true* (en este caso el 3, cuyo cuadrado es menor que 20) y se marcan como *false* todos sus múltiplos que no hayan sido ya marcados como *false* (en este caso 9 y 15).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
false	false	true	true	false	true	false	true	false	false	false	true	false	true	false	false	false	true	false	true

El siguiente elemento marcado como *true* es el 5, pero como el cuadrado de 5 ( $5 * 5 = 25$ ) es mayor que 20, el algoritmo termina.

- **(1 puntos) public static int[] nPrimosEratostenes(boolean[] criba).** Realiza otra función que dado un array de booleanos marcados con *true*, devuelva un array de enteros con los números de los elementos marcados como *true*.

Siguiendo con el ejemplo anterior, para un array de entrada como el siguiente:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
false	false	true	true	false	true	false	true	false	false	false	true	false	true	false	false	false	true	false	true

La función debe devolver el siguiente array de enteros:

0	1	2	3	4	5	6	7
2	3	5	7	11	13	17	19

### 3. (4 puntos) ShiftRows

El método ShiftRows es usado en criptografía para cifrar datos. Crea un proyecto en Netbeans llamado **Desplazar** en dónde se implementarán los siguientes dos métodos:

- **(2 puntos) public static void rotarIzquierda(int[] elementos, int n).** Este método recibe un array de enteros y el número de posiciones que deben rotarse a la izquierda los elementos del array. Como resultado de la aplicación del método al array pasado por parámetro, este array deberá quedar rotado esas  $n$  posiciones.

Por ejemplo para el siguiente array:

0	1	2	3	4	5	6	7
2	4	8	16	32	64	128	256

Si lo rotamos sólo una posición a la izquierda ( $n = 1$ ). Deberemos obtener:

0	1	2	3	4	5	6	7
4	8	16	32	64	128	256	2

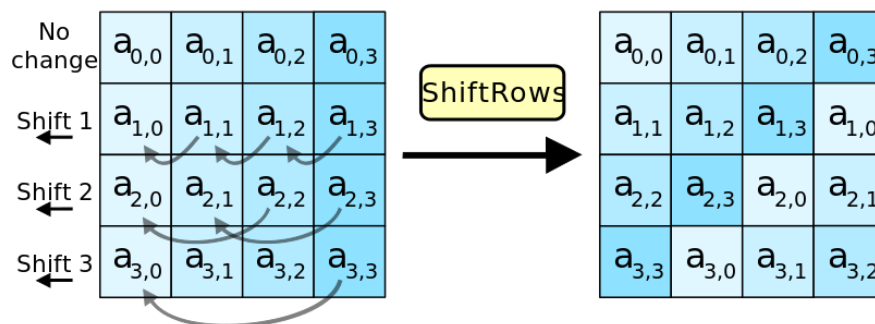
En cambio si lo rotamos 3 posiciones ( $n = 3$ ). Deberemos obtener:

0	1	2	3	4	5	6	7
16	32	64	128	256	2	4	8

Nótese que, por supuesto, después de las rotaciones no se pierden ni se duplican valores contenidos en el array.

- **(2 puntos) public static void desplazarFilas(int[][] matriz).** Este método deberá rotar una matriz cuadrada recibida por parámetro de la siguiente manera. La primera fila se queda como está. La segunda fila se rota hacia la izquierda una posición, la tercera fila se rota dos posiciones, y así sucesivamente (la  $n$ -ésima fila se rotará  $n-1$  veces). Se deberá usar el método **rotarIzquierda()** implementado en el apartado anterior.

La siguiente ilustración muestra el funcionamiento que acabamos de explicar.



## Resultados de Aprendizaje, Criterios de Evaluación y Rúbrica

En cada ejercicio / apartado se tocan los siguientes resultados de aprendizaje y criterios de evaluación:

Resultado de Aprendizaje	Criterios de evaluación
RA1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.	d, e, f, g, h, i
RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.	a, b, c, d, e, f, g
RA3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.	a, b, e, f, g
RA5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.	a
RA6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.	a

Cada ejercicio / apartado se valorará con la siguiente rúbrica.

#	Criterio	Porcentaje
1	El programa/función implementado no cumple los requisitos, no soluciona de forma algorítmica el ejercicio o las soluciones obtenidas por el programa no son las esperadas, el código no compila o contiene errores.	0%
2	El programa/función implementado soluciona de forma algorítmica el ejercicio pero falla con datos de entrada no permitidos.	25%
3	El programa/función implementado tiene fallos inesperados en situaciones específicas o concretas, es decir, falla para un determinado caso o valor de entrada, pero en general el resultado obtenido es válido.	50%
4	El programa/función implementado cumple los requerimientos pero: <ul style="list-style-type: none"> <li>El código no es legible o no está bien estructurado.</li> </ul>	75%
5	El programa/función implementado se ajusta perfectamente a la especificación: <ul style="list-style-type: none"> <li>Se validan los datos de entrada.</li> <li>El resultado obtenido es válido para cualquier dato de entrada.</li> <li>El código es modular y se emplean funciones/métodos adecuadamente.</li> <li>El código es legible y usa comentarios relevantes y/o Javadoc.</li> </ul>	100%