	Examen 5_A: ARRAYS		
	1º DAW - PRO		
	Departamento de Informática	18/12/2023	Página 1 de 4

Ejercicio 1

Se tienen las notas de los alumnos guardadas en un array 2D de forma que para cada alumno se guardan las notas de todos los exámenes que ha realizado.

Realiza un programa que calcule las notas medias para cada alumno y la media total del curso. Ejecuta el programa con los siguientes datos de estudio:

Notas[][] = {{6,2,4,9,1}, {0,8,8,10,2},{9,2,5,9,3}, {5,1,1,1,4}};

Alumnos[] = {"Juan", "Pedro", "Antonio", "Luis"};

Nº	Nombre	Nota 1	Nota 2	Nota 3	Nota 4	Nota 5
1	Juan	6	2	4	9	1
2	Pedro	0	8	8	10	2
3	Antonio	9	2	5	9	3
4	Luis	5	1	1	1	4

Las notas de cada alumno se asociarán según la tabla superior, se podrá saber la nota de un alumno atendiendo al número de la fila donde se encuentra.

El programa funcionará de la siguiente manera:

- El programa principal llamará a una función llamada **notamedia()** a la que le pasará los valores enteros de la matriz con las notas totales, y calculará la nota media para cada alumno llamando a la función **mediaDescartar()**. La función **notamedia()** devolverá un vector con las notas medias de cada alumno, respetando el orden de los alumnos en clase. Las notas medias se muestran con sólo un decimal.
- Función **mediaDescartar()**: Desde la función **notamedia()** se llamará a función **mediaDescartar()** para cada alumno y calculara la media de todas las notas disponibles para dicho alumno, descartando siempre la mayor y la menor nota que haya sacado. Una vez realizada la media con los valores no excluidos devolverá la media de cada alumno a la función **notamedia()**.
- Desde el programa principal se mostrará el nombre de cada alumno junto con la media calculada. Todas las funciones deben funcionar para cualquier número de alumnos y de notas parciales que tenga cada alumno.
- Además, el programa principal calculará la media total del curso llamando a la función **mediaDescartar()** y se mostrará por pantalla. Para que puedas llamar a **mediaDescartar()** debes sobrecargar esta función.




Ejercicio 2

Tenemos una agenda para registrar la actividad de un alumno durante el mes de febrero y guardamos esta información en un array 3D. La agenda guarda las 4 semanas de febrero en formato de números enteros (ej: 1ª semana en posición 0, la 2ª en la posición 1, etc), cada semana tiene guardados los días en los que estudio programación (el lunes lo guarda en la posición 0, martes en la 1, ...), y en cada día guardo las horas de estudio (posición 0) y las horas de juego (posición 1). Pongo un ejemplo:

```
int[][][] agenda = {  
    {  
        {1,2},  
        {2,3},  
        {4,1},  
        {0,2},  
        {1,0},  
        {0,2},  
        {3,2}  
    },  
    {  
        {1,2},  
        {2,3},  
        {4,3},  
        {0,5},  
        {1,0},  
        {0,0},  
        {0,7}  
    },  
    {  
        {0,2},  
        {0,3},  
        {5,1},  
        {0,2},  
        {0,0},  
        {3,2},  
        {3,2}  
    },  
    {  
        {1,2},  
        {2,3},  
        {4,1},  
        {0,2},  
        {1,0},  
        {0,4},  
        {3,3}  
    }  
};
```

La matriz sería un **int[4][7][2]**. Para saber cuántas horas hemos estudiado el martes de la semana 3, veríamos el elemento: **agenda[2][1][0]=0**. Para saber cuántas horas hemos jugado el miércoles de la 1ª semana, veríamos el elemento: **agenda[0][2][1]=1**.


	Examen 5_A: ARRAYS		
	1º DAW - PRO		
	Departamento de Informática	18/12/2023	Página 3 de 4

Realiza las siguientes operaciones a través de funciones que se llamen del programa principal.

- Calcula las horas que se estudian en una semana mediante la función **estudioSemana()**, la función debe recibir como parámetro la matriz agenda y la semana que se quiere estudiar y debe devolver al programa principal las horas totales estudiadas en esa semana. Se mostrará dicho número de horas desde el programa principal.
- Usa la misma función **estudioSemana()** para que calcule las horas totales que estudiaría el alumno sólo en un fin de semana de una semana en concreto (sobrecarga la función como creas conveniente). En este caso la función no devolverá nada al programa principal, mostrará las horas desde la función.
- Calcula el número total de horas que juega el alumno en el mes mediante la función **juegoMes()**, la función sólo recibe como parámetro el array agenda. En este caso la función no devolverá nada al programa principal, mostrará las horas totales de juego desde la función.
- Calcula el número de horas que juega el alumno en cada semana del mes mediante la función **juegoSemana()**, la función sólo recibe como parámetro el array agenda. En este caso la función devolverá un vector con las horas que juega cada semana. Se imprimirán los valores de dicho vector desde el programa principal.
- Usa la función **fiestaSemana()** para poner a 5 las horas de juego de todos los días que eran lectivos de la semana que se quiera. La función recibe como parámetro el array agenda y el índice de la semana festiva. Deberá fijar en 5 las horas de juego de cada día “lectivo” de esa semana y mostrar la agenda por pantalla.

Ejercicio 3

Realiza un programa que llame a una función a la que se le pasa como parámetros dos números enteros. La función calculará todos divisores del primer número que sean múltiplos del segundo número y devolverá al programa principal todos esos divisores mediante un vector que contenga dichos divisores. Por ejemplo, si le paso a la función los parámetros (48, 3), el programa devolverá el vector [3, 6, 12, 24]. Otro ejemplo: si le paso a la función los parámetros (250, 5), el programa devolverá el vector [5, 10, 25, 50, 125]. La función debe devolver un vector con la dimensión exacta del número de divisores hallado.

	Examen 5_A: ARRAYS		
	1º DAW - PRO		
	Departamento de Informática	16/11/2023	Página 4 de 4

Consideraciones Generales

Escribir el código de los tres ejercicios planteado, usando ayudándote del IDE Netbeans 19 junto con la versión 17 de JDK, como hemos visto en clase. Usar los elementos y estructuras vistos hasta ahora en clase.

A la hora de evaluar se aplicará la rúbrica adjunta, donde además de comprobar el correcto funcionamiento de los programas, se evaluarán los siguientes puntos:

- Código eficiente: uso apropiado de los recursos. No hay código innecesario o realiza acciones superfluas.
- Realizar el menor número de operaciones posibles.
- Crear una interfaz de usuario sencilla e intuitiva.
- Código ordenado: con sangrías en los bloques, espaciados apropiados, etc.
- Nombre de variables que tienen significado.
- Establece comentarios en el código.

Rúbrica

PRO Examen 5.1: ARRAYS

Acciones Cancelar Guardar

		Excelente	10 Avanzado	7.5 Medio	5 Básico	2.5 Nulo	0
1a. NOTAS medias	35 %	Funcionan bien los 4 apartados	Sólo funcionan bien 3 de los 4 apartados	Sólo funcionan bien 2 de los 4 apartados	Sólo funciona bien uno de los 4 apartados	El programa no compila, o hace algo totalmente diferente a lo que se pide en el enunciado.	
Programas que devuelve 6a 6b 2b 2c 2g 2i							
1b. Código-eficiencia	5 %	Cumple todos los criterios sobre código y eficiencia	Falla en un criterio sobre código y eficiencia.	Falla en dos criterios sobre código y eficiencia.	Falla en tres criterios sobre código y eficiencia.	El programa falla en más de 3 criterios de código y eficiencia.	
Código ordenado, eficiente, 2i							
2a. ARRAY 3D	30 %	Sólo funcionan bien 4 de los 5 apartados	Sólo funcionan bien 3 de los 5 apartados	Sólo funcionan bien 2 de los 5 apartados	Sólo funciona bien uno de los 5 apartados	El programa no compila, o hace algo totalmente diferente a lo que se pide en el enunciado.	
Cálculo del productorio de los n 2b 2c 2g 2i 4e 4f 4g 6a 6b							
2b. Código-eficiencia	10 %	Cumple todos los criterios sobre código y eficiencia + y no falla ningún apartado.	Falla en 1 criterio sobre código y eficiencia o en un apartado	Falla en dos criterios sobre código y eficiencia o 1 criterio y eficiencia o 2 criterios y un apartado	Falla en tres criterios sobre código y eficiencia o 2 criterios y un apartado	El programa falla en más de 3 criterios de código y eficiencia y en un apartado	
Código ordenado, eficiente, 2i							
3. Divisores	20 %	El programa funciona y tal y como se pide en el enunciado y el código está impecable.	Medio + el código está mejor en eficiencia y apariencia que en la casilla "medio".	El programa funciona correctamente pero tiene fallos de indentación, sin comentarios, variables innecesarias...	El programa funciona pero el resultado no es correcto aunque se aproxima al resultado solicitado.	El programa no compila, o hace algo totalmente diferente a lo que se pide en el enunciado.	
Programa para calcular divisores 4e 4f 4g 6a 6b 2b 2c 2g 2i							

Total: 100 %