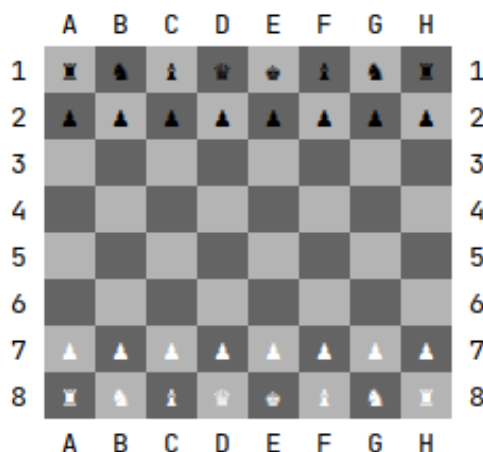


NOMBRE		ETAPA / CICLO	CURSO
		CFGS DAM/DAW	1º
APELLIDOS		ASIGNATURA/MÓDULO	CONVOCATORIA
		Programación	ORDINARIA 1EVA
DNI	FECHA	NOTA	
	25-11-2024		

1. (2p) Se desea crear una función donde se obtenga unas coordenadas para un tablero de ajedrez.



Dicha función debe devolver una coordenada del tipo {'3','C'}, donde el primer valor debe ser un numero entre 1 y 8, y el segundo valor debe ser una letra entre A y H. Por lo que se debe controlar que el usuario solo introduzca 2 caracteres y siga la política descrita. Para ello se pide solucionar el problema mediante dos enfoques:

- Resuelve el problema mediante un enfoque iterativo. Elige la estructura iterativa que mejor se adapte al problema.
- Resuelve el problema mediante un enfoque recursivo. Define el caso base, así como las llamadas recursivas sin la utilización de ninguna estructura iterativa.

Introduce una coordenada:

3CC

Debes introducir 2 caracteres. Introduce una coordenada:

3

Debes introducir 2 caracteres. Introduce una coordenada:

9C

El primer carácter debe estar entre 1 y 8. Introduce una coordenada:

1J

El segundo carácter debe estar entre A y H. Introduce una coordenada:

3c

[3, C]

2. (2p) Crea una función que devuelva la suma de los elementos de una matriz dentro de un determinado rango. La función tendrá 3 parámetros, el array, la posición inicial desde la cual se quiere comenzar a sumar elementos y cuántos elementos (rango) quieres sumar (será inclusive). Debes tener en cuenta que, si el rango supera la longitud del array, se sumarán los elementos hasta el último. Puedes usar la siguiente cabecera de la función.

public static int sumaN(int[] array, int inicio, int rango)

- Resuelve el problema mediante un enfoque iterativo. Elige la estructura iterativa que mejor se adapte al problema.
- Resuelve el problema mediante un enfoque recursivo. Define el/los caso/s base/s, así como las llamadas recursivas sin la utilización de ninguna estructura iterativa.

Ejemplos de ejecución:

sumaN(new int[]{0}, 1, 3)	→ resultado 0
sumaN(new int[]{0, 1, 2, 3, 4}, 1, 3)	→ resultado 6
sumaN(new int[]{1, 2, 3}, 1, 3)	→ resultado 5
sumaN(new int[]{1, 2, 3, 4, 5, 6, 7, 8}, 1, 0)	→ resultado 0
sumaN(new int[]{1, 2, 3, 4, 5, 6, 7, 8}, 1, 11)	→ resultado 35

3. (3p) Dado el siguiente array, donde se almacenan los nombres de alumnos de una clase, y los partes asociados a cada uno de ellos,

```
String[][] partes = new String[][]{  
    {"Juan", "Pedro", "Carlos", "Jose", "Vane", "Jenni", "Lore"},  
    {"3", "5", "10", "0", "1", "10", "4"}  
};
```

se solicita:

- a. Una función que calcule la media de partes del aula

Media de partes: 4.714286

- b. Una función que devuelva el nombre de quien o quienes tienen el mayor número de partes.

Nombres con más partes: Carlos Jenni

- c. Una función que indique si un alumno está en el array. Para ello, se deberá devolver su posición si se encuentra, o -1 si no.

Posición de Jenni: 5

Posición de Tomas: -1

- d. Una función que devuelva, dado el nombre de un alumno, la cantidad de partes que tiene asociado. Si el alumno no se encuentra, se deberá devolver un mensaje indicándolo.

Partes de Jenni: 10

Partes de Tomas: No se encuentra Tomas en el array.

Puedes crear las funciones auxiliares que consideres necesarias para dar respuesta a cada pregunta.

4. **(1.5p) Crear una función que, dado un número, creará una matriz de dimensión n x n con 1's formando una X y 0's en caso contrario.**

Por ejemplo, si la función recibe por parámetro 6, creará una matriz con el siguiente contenido:

```
[1, 0, 0, 0, 0, 1]
[0, 1, 0, 0, 1, 0]
[0, 0, 1, 1, 0, 0]
[0, 0, 1, 1, 0, 0]
[0, 1, 0, 0, 1, 0]
[1, 0, 0, 0, 0, 1]
```

Si la función recibe 5, creará una matriz con el siguiente contenido:

```
[1, 0, 0, 0, 1]
[0, 1, 0, 1, 0]
[0, 0, 1, 0, 0]
[0, 1, 0, 1, 0]
[1, 0, 0, 0, 1]
```

5. **(1.5p) Crear dos funciones** para realizar las siguientes funcionalidades:

- a. La primera **rellenará un array unidimensional con números enteros aleatorios del 1 al 10**

Ejemplo:

```
[5, 9, 4, 7, 10, 5, 9, 1, 4]
```

- b. La segunda **invertirá un array unidimensional de enteros.**

Si le pasáramos el array del ejemplo 5a daría el siguiente resultado:

```
[4, 1, 9, 5, 10, 7, 4, 9, 5]
```