



Arrays y Matrices

EJERCICIO 1

Realizar un programa que lea los tiempos en los que 10 corredores han acabado una carrera. El programa debe determinar qué corredores tienen el primer, segundo y último puesto, así como cuál es el tiempo medio en que se ha corrido la carrera.

EJERCICIO 2

Diseña un programa que pida el valor de 10 números enteros distintos y los almacene en un vector. Si se da el caso y se trata de introducir un número repetido, el programa advertirá al usuario tan pronto sea posible, y solicitará nuevamente el número hasta que sea diferente de todos los anteriores. A continuación, el programa mostrará los 10 números por pantalla.

EJERCICIO 3

Realiza un programa que vaya pidiendo números enteros mientras que no se introduzca el cero y rellene dos vectores, uno con los números pares, y otro con los números impares. Al final, se debe mostrar por pantalla tanto el vector de números pares como el de impares, indicando la posición del vector y el valor que ha sido almacenado.

EJERCICIO 4

Mezcla de dos arrays. Escribe un programa que, dados dos arrays (de longitudes M y N respectivamente), que ya están ordenados ascendentemente, produzca un array (de longitud M + N) con el contenido de los dos arrays de entrada, también ordenado ascendentemente. Por ejemplo:

<1, 5, 8, 24>, <0, 6, 7, 20, 24, 25, 26> → <0, 1, 5, 6, 7, 8, 20, 24, 24, 25, 26>





EJERCICIO 5

Diseña un programa que lea los elementos de una matriz de 4×5 reales y genere un vector de tamaño 4 en el que cada elemento contenga el sumatorio de los elementos de cada fila. El programa debe mostrar la matriz original y el vector en este formato.

EJERCICIO 6

Realizar un programa que pida dos matrices y las sume (siempre que sea posible) o avise de que no se pueden sumar. Hacer el diseño descendente del problema (habrá seguro una función/procedimiento para leer la matriz y otra para el cálculo de la suma). Lo primero que se hará es pedir la dimensión de la matriz y luego se leerán sus datos.

EJERCICIO 7

Queremos un programa que nos permita jugar de forma sencilla al tres en raya a dos jugadores (A y B). El programa permitirá almacenar el estado del tablero (0 si nadie ha insertado ficha en una posición 1 para el jugador A y 2 para el jugador B). Realiza los procedimientos que:

- a. Muestren el estado actual del tablero.
- b. Pidan la posición de la ficha a introducir alternativamente a cada uno de los jugadores.
- c. Comprueben si el tablero está lleno. Para simplificar esta versión no comprobará si se ha hecho el tres en raya o no.

EJERCICIO 8

Completar el programa del tres en raya para que cada vez que se inserte una ficha se compruebe si en el tablero hay un tres en raya hecho por alguno de los dos jugadores.





EJERCICIO 9

Se pretende realizar un programa para gestionar la lista de participaciones en una competición de salto de longitud. El número de plazas disponible es de 10. Sus datos se irán introduciendo en el mismo orden que vayan inscribiéndose los atletas. Diseñar el programa que muestre las siguientes opciones:

- 1- Inscribir un participante.
- 2- Mostrar listado de datos.
- 3- Mostrar listado por marcas.
- 4- Finalizar el programa.

Si se selecciona 1, se introducirán los datos de uno de los participantes: Nombre, mejor marca del 2024, mejor marca del 2023 y mejor marca del 2022. Si se elige la opción 2, se debe mostrar un listado por número de dorsal. La opción 3 mostrará un listado ordenado por la marca del 2024, de mayor a menor. Tras procesar cada opción, se debe mostrar de nuevo el menú inicial, hasta que se seleccione la opción 4, que terminará el programa.

EJERCICIO 10

Una empresa tiene N sucursales en todo el país. Se formó una matriz de $N \times 12$ que contiene las ventas de cada sucursal durante 12 meses del año. Elaborar un programa que encuentre. (Generar datos aleatorios dentro de un rango 5000 y 20000)

- a. El total de ventas de la compañía
- b. El total de ventas por cada sucursal
- c. Sucursal que más vendió durante un año
- d. Mes que menos vendió la compañía





EJERCICIO 11

El youtuber Folagor03 contacta con nosotros para hacer un programa que nos muestre la tabla de tipos de Pokémon (tiene un problema serio con ello).

En la solución os dejo un fichero java con los datos necesarios: la tabla de tipos y los tipos de Pokemon. Es el fichero llamado Constantes.java, copialo a tu Proyecto, si lo quieres hacer de cero.

Los datos que contiene la tabla de tipos es la siguiente:

Efect.	Tipo del Pokémon del oponente																	
	ROCA	AGUA	TIERRA	DRAGON	ELECTRO	FANTASMA	FUEGO	HILO	PLANTA	PSICO	LUCHA	NOCTURNO	PLANTA	PSICO	ROCA	SINIE	TIERRA	VENENO
ROCA	1/2	1/2	-	-	1/2	-	1/2	x2	x2	-	-	-	-	x2	-	-	-	-
AGUA	-	1/2	-	1/2	-	-	x2	-	-	-	-	-	1/2	x2	-	x2	-	-
TIERRA	1/2	-	-	-	-	1/2	1/2	1/2	-	1/2	-	x2	x2	-	x2	-	1/2	1/2
DRAGON	1/2	-	-	x2	-	-	-	x0	-	-	-	-	-	-	-	-	-	-
ELECTRO	-	x2	-	1/2	1/2	-	-	-	-	-	-	-	1/2	-	-	-	x0	x2
FANTASMA	-	-	-	-	-	x2	-	-	-	-	x0	-	x2	-	1/2	-	-	-
FUEGO	x2	1/2	x2	1/2	-	-	1/2	-	x2	-	-	x2	-	1/2	-	-	-	-
HILO	1/2	-	-	x2	-	-	1/2	-	-	-	x2	-	-	-	x2	-	1/2	-
PLANTA	1/2	-	-	x2	-	-	1/2	-	-	x2	-	-	-	-	x2	-	1/2	-
PSICO	1/2	1/2	-	x2	-	-	1/2	-	-	x2	-	-	-	-	x2	-	1/2	x2
LUCHA	x2	-	1/2	-	-	x0	-	1/2	x2	-	x2	-	1/2	x2	x2	-	1/2	1/2
NOCTURNO	1/2	-	-	-	-	x0	-	-	-	-	-	-	-	1/2	-	-	-	-
PLANTA	1/2	x2	1/2	1/2	-	-	1/2	-	-	-	-	-	1/2	x2	-	x2	1/2	1/2
PSICO	1/2	-	-	-	-	-	-	-	-	x2	-	-	1/2	-	x0	-	x2	-
ROCA	1/2	-	x2	-	-	-	x2	-	x2	1/2	-	-	-	-	-	1/2	-	x2
SINIE	-	-	-	-	x2	-	1/2	-	1/2	-	-	-	x2	-	-	1/2	-	-
TIERRA	x2	-	1/2	-	x2	-	x2	-	-	-	-	-	1/2	-	x2	-	x2	x0
VENENO	x0	-	-	-	1/2	-	x2	-	-	-	-	-	x2	-	1/2	-	1/2	-
DRAGON	1/2	-	x2	-	1/2	-	-	-	-	x2	-	x2	-	1/2	-	-	-	-

- 5: el atacante hace la mitad de daño al oponente (No es muy eficaz)
- 1: el atacante hace un daño neutro al oponente (No se da información) (en la imagen representa al -)
- 2: el atacante hace el doble de daño al oponente (Es muy eficaz)
- 0: el atacante hace un daño nulo al oponente (No afecta)





Las filas de la matriz es el Pokemon que ataca. Las columnas son el pokémon que recibe el ataque. Por ejemplo, Fila 4 (Eléctrico) ataca a un pokémon de la Columna 1 (Agua), habrá un 2 porque eléctrico es eficaz contra el agua.

Para mayor comodidad del usuario, al poner los tipos de pokemon, se los mostraremos de esta manera:

1. ACERO
2. AGUA
3. BICHO

Así hasta el final. Mirar el vector con los tipos.

Las opciones que le daremos a Folagor03 son:

- **Mostrar debilidades:** pedimos un tipo o dos (tenemos que preguntárselo al usuario) y nos mostrará todos los tipos que hacen un daño eficaz (recordar un 2) a este tipo.

Por ejemplo, si Folagor03 elige DEBILIDADES de FUEGO, le deberían aparecer estos tipos:

- AGUA
- ROCA
- TIERRA

Si elige dos tipos de pokémon, por ejemplo FUEGO y TIERRA, habrá que ver esos dos tipos y multiplicar las debilidades y mostrar el multiplicador de daño.

- AGUA : x4 ($2 * 2 = 4$, doblemente eficaz)
- TIERRA : x2 ($2 * 1 = 2$, la tierra es neutro con la tierra)





Fíjate que no aparece el tipo ROCA, ya que TIERRA tiene resistencia y hace que un ataque sea neutro ($0.5 * 2 = 1$)

- **Mostrar todas los tipos eficaces:** pedimos un solo tipo (un ataque al final solo es de un solo tipo) y mostramos que tipos son débiles contra ellos.

Por ejemplo, si elegimos ACERO, los pokemon a los que hace un “Es muy eficaz” son:

- HADA
- HIELO
- ROCA

- **Mostrar toda la información relativa de un tipo:** pedimos un tipo y nos mostrará el daño de cada uno. Según el valor poner los siguientes valores:
 - 1 : NEUTRO
 - 0 : NO AFECTA
 - 2 : MUY EFICAZ
 - 5 : NO MUY EFICAZ

Por ejemplo si elegimos AGUA, mostraría algo así:

- ACERO : NEUTRO
- AGUA: NO MUY EFICAZ
- BICHO: NEUTRO

Así con todos los tipos





- **Mostrar eficacia de un tipo a otro:** pedimos dos tipos de pokémon, el primero será el atacante y el segundo será el oponente. Simplemente diremos si es eficaz o no, neutro o no le afecta el ataque.

Por ejemplo, si elijo el tipo TIERRA de atacante y VOLADOR de oponente este me deberá mostrar:

- NO AFECTA

También tiene que pedir si quiere que el tipo del oponente sea de uno o dos tipos.

