



EJERCICIOS – Arrays y Matrices

Arrays o Vectores

1. Leer 10 números enteros y mostrarlos en el mismo orden introducido. A continuación, mostrarlos en el orden inverso al introducido.
2. Leer 10 números enteros y a continuación mostrar la media de los números introducidos, el mayor valor y el menor valor. Usar un array para almacenar los valores y sacar toda la información en una sola iteración del array.
3. Igual que el anterior, pero implementa una función para cada tarea (media, mayor y menor de los elementos almacenados en un array pasado como parámetro).
4. Leer 10 números enteros. Debemos mostrarlos en el siguiente orden: el primero, el último, el segundo, el penúltimo, el tercero, etc...
5. Leer por teclado dos tablas A y B de 5 números enteros y mezclarlas en una tercera de la forma: el 1º de A, el 1º de B, el 2º de A, el 2º de B, etc...
6. Leer por teclado una serie de 5 números enteros. El programa debe indicarnos si los números están ordenados de forma creciente, decreciente o si están desordenados. Crea una función para mejorar el código.
7. Diseñar una aplicación que declare una tabla de 10 elementos enteros. Leer mediante el teclado 8 números. Después se debe pedir un número y una posición, insertarlo en la posición indicada, desplazando los que estén detrás. Crea una función para mejorar el código.
8. Crear un programa que lea por teclado una tabla de 5 números enteros y la desplace una posición hacia la derecha: el primero pasa a ser el segundo, el segundo pasa a ser el tercero y así sucesivamente. El último pasa a ser el primero.
9. Igual que el anterior, pero desplazando N posiciones (N es un número introducido por el usuario).
10. Leer por teclado una tabla de 10 elementos numéricos enteros y una posición (entre 0 y 9). Eliminar el elemento situado en la posición dada sin dejar huecos.

Matrices

11. Crear una tabla bidimensional de tamaño 5x5 y rellenarla de forma que los elementos de la diagonal principal sean 1 y el resto 0. Mostrarla.
12. Crear una matriz “marco” de tamaño 8x6. Una matriz “marco” es aquella que todos sus elementos son 0 salvo los de los bordes que deben ser 1. Mostrarla.
13. Crear una tabla bidimensional de tamaño 3x4 de números enteros (leídos desde teclado). Mostrar la matriz y la suma de los valores de cada fila.
14. Crear una matriz 5x3 de números enteros (leídos desde teclado) y mostrar el menor, el mayor y la media de los elementos almacenados en la matriz.
15. Crear dos matrices 3x3 de números enteros (leídos desde teclado), sumarlas y mostrar su suma. El resultado será también una matriz (suma) donde cada elemento será la suma de los elementos de las otras matrices en la misma posición.
16. Crea una matriz de 3x6 números enteros aleatorios **no repetidos**.
17. **Ordenar matrices.** Crea una matriz de enteros de dos dimensiones de 50x50 elementos, rellénala con números aleatorios entre 1 y 500. A continuación, realiza dos funciones:
 - a. **ordenaFilas(matriz):** que ordene la matriz por filas, cada fila de la matriz quedará ordenada de menor a mayor.



- b. **ordenaColumnas**(matriz): que ordene la matriz por columnas, cada columna quedará ordenada de menor a mayor, independientemente de las demás.

18. Queremos realizar una **encuesta** a 10 personas, en esta encuesta indicaremos el sexo (1=masculino, 2=femenino), si trabaja (1=si trabaja, 2= no trabaja) y su sueldo (si tiene un trabajo, sino sera un cero) estará entre 600 y 2000 (valor entero). Los valores pueden ser generados aleatoriamente. Calcula y muestra lo siguiente:

- Porcentaje de hombres (tengan o no trabajo).
- Porcentaje de mujeres (tengan o no trabajo).
- Porcentaje de hombres que trabajan.
- Porcentaje de mujeres que trabajan.
- El sueldo promedio de las hombres que trabajan.
- EL sueldo promedio de las mujeres que trabajan.
-

Usa todos los métodos que veas necesarios, piensa que es aquello que se repite o que puede ser mejor tenerlo por separado.

19. Escribe un programa que genere al azar 20 números enteros comprendidos entre 0 y 9. Estos números se deben introducir en un array de 4 filas por 5 columnas.

El programa mostrará las **sumas parciales** de filas y columnas igual que si de una hoja de cálculo se tratara. La suma total debe aparecer en la esquina inferior derecha.

| | | | | | |
|--------------------|--------------------|--------------------|--------------------|--------------------|-----------------|
| | | | | | Σ fila 0 |
| | | | | | Σ fila 0 |
| | | | | | Σ fila 0 |
| | | | | | Σ fila 0 |
| Σ columna 0 | Σ columna 1 | Σ columna 2 | Σ columna 3 | Σ columna 4 | TOTAL |

Nota: se valorará en todos los ejercicios el control de excepciones en la lectura de teclado, así como el uso correcto de funciones (métodos).



Ejercicios extra: solo para subir nota aquellos alumnos que quieran probar algo más.

20. Implementación de una **Pila** con un vector de 20 elementos. Una Pila se basa en el concepto “Primero en entrar Último en salir”. De modo que se saca siempre el último que se añadió. Implementa métodos para crear una pila vacía, insertar elemento en la pila, sacar elemento de la pila (lo muestra y lo quita), comprobar si la pila está llena.
21. Vamos a realizar el típico juego del **3 en raya**, donde habrá dos jugadores que tengan que hacer el 3 en raya, los signos serán el X y el O, cuando haya una posición vacía habrá un –
- El tablero de juego, será una matriz de 3×3 de char.
- El juego termina cuando uno de los jugadores hace 3 en raya o si no hay más posiciones que poner.
- Primero juega una posición un jugador y luego el otro.
- El juego debe pedir las posiciones donde el jugador actual quiera poner su marca ([0,0], [0,1], [0,2], [1,0], [1,1], [1,2], [2,0], [2,1], [2,2]), esta debe ser validada y por supuesto que no haya una marca ya puesta.
22. **Dominó.** Realiza un programa que realice las siguientes acciones:
- Generar las piezas de un dominó en un array bidimensional llamado “fichas”. Puedes hacerlo por filas o por columnas. Ten en cuenta que son sólo 28 fichas y que no puedes generar una ficha dos veces (por ejemplo: 1|2 = 2|1).
 - Crea un array “juego” que solo tenga dos int, la cabeza juego[0] y la cola juego[1], inicialmente a -1, hasta que se ponga la primera ficha.
 - Realiza una función quitarUnaFicha() que se le pase como parámetro el array “fichas” y aleatoriamente tome una ficha del array y la devuelva como un vector ficha (x|y). En el array de fichas los valores en esa posición quedarán como -1|-1, o podéis desplazar el array a la izquierda y al hueco del final ponerle -1|-1. De modo que cuando busquemos una ficha aleatoriamente en “fichas” debes hacerlo mientras sea distinto de -1 lo que hay en la posición generada aleatoriamente, pues en esa posición no habría ninguna ficha.
 - Realiza una función pintarFichas() que se le pase por parámetro un array de fichas y pinte las piezas de dominó así (“Blanca | Blanca”, “Blanca | Uno”, ..., “Dos | Cinco”, “Dos | Seis”, ...). Servirá para pintar las fichas del juego y las de cada jugador. Si una casilla tiene -1|-1 significa que no hay ficha ahí y no se debe pintar.
 - Realiza una función repartirFichas() que se le pase como parámetro el array de fichas y devuelva un array con 7 fichas quitadas aleatoriamente del array “fichas” para asignárselas a un jugador.
 - Realice una función ponerFicha(). Se le pasa un jugador y el array del juego (cabeza|cola). Busca una ficha para poner en la cabeza o la cola del juego siempre que coincida por un extremo de la ficha. Se actualizan los valores de juego (cabeza|cola) según se ponga la ficha por un lado o por otro. Ponerla significa que los valores de la ficha en el jugador quedan a -1. Si no se puede poner devuelve false, si se puede poner se devuelve true.
 - Realice una función esGanador(). Se le pasa un array de un jugador y si tienes todas las fichas con -1 devuelve true, si no devuelve false.
 - Simula una partida de dominó. Primero crea cuatro arrays jugadores repartiéndoles las fichas. Después ve poniendo fichas hasta que un jugador se quede sin fichas (todas a -1|-1), será el ganador, di cuál de los cuatro ha ganado.



23. Crea un método que dada una determinada matriz nxm devuelva su **matriz traspuesta**.

$$A \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad A^T \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$
$$A \begin{bmatrix} 1 & 4 & 3 \\ 8 & 2 & 6 \\ 7 & 8 & 3 \\ 4 & 9 & 6 \\ 7 & 8 & 1 \end{bmatrix} \quad A^T \begin{bmatrix} 1 & 8 & 7 & 4 & 7 \\ 4 & 2 & 8 & 9 & 8 \\ 3 & 6 & 3 & 6 & 1 \end{bmatrix}$$