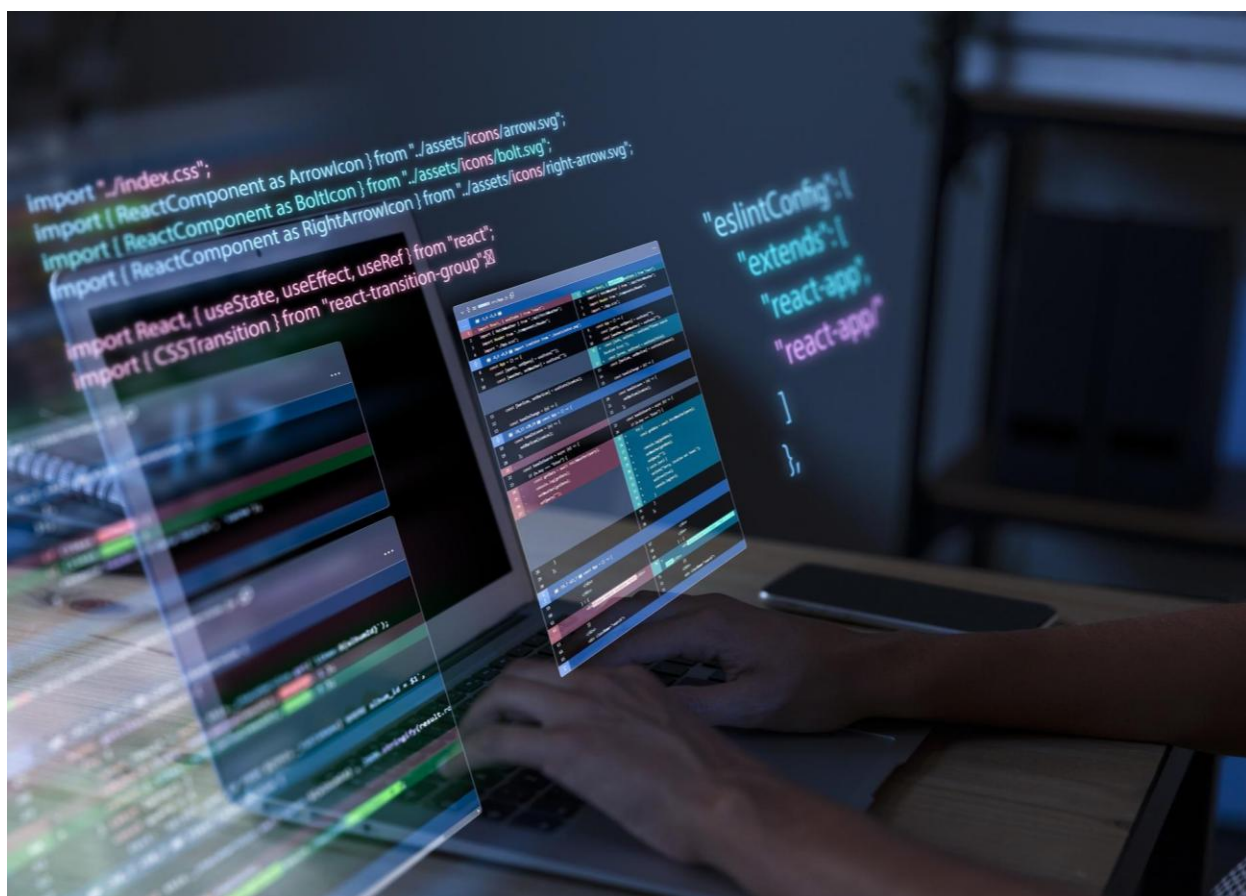


Ejercicios

Arrays (I)



Autor: Marcela Martín

Fecha: noviembre 2025

1. Crea 3 arrays de 5 elementos: el primero de enteros, el segundo de doubles, y el tercero de booleanos. Muestra las referencias en las que se encuentra cada una de los arrays anteriores.
Realiza dos versiones del programa, una que utilice el operador new para crear el array, y otra que no lo utilice.
2. Construye un array de 10 enteros. Declara 3 variables que referencien el array creado. Comprueba, imprimiendo por pantalla, que todas las variables contienen la misma referencia.
Realiza dos versiones del programa, una que utilice el operador new para crear el array, y otra que no lo utilice.
3. Crea un array de longitud 10 que se inicializará con números aleatorios comprendidos entre 1 y 100. Muestra la suma de todos los números aleatorios que se guardan en el array.
4. Diseña un programa que solicite al usuario 5 números decimales y los almacene en un array. A continuación, se muestran los números en el mismo orden que se han introducido.
5. Diseña un programa que pregunte al usuario cuántos números enteros quiere introducir. A continuación, se solicita esa cantidad de números enteros y por último se imprimen en orden inverso al introducido.
6. Diseña un programa que solicite al usuario 4 números enteros. A continuación, se muestran el máximo de los números introducidos.
Utiliza una función que reciba como parámetro el array que contiene los 4 números y devuelva un entero, correspondiente al valor máximo almacenado en el array.
7. Escribe un programa que solicite un número entero al usuario (n). A continuación, solicita al usuario que introduzca n números. El programa realiza

la media de los números positivos, la media de los negativos, y cuenta el número de ceros introducido.

8. Escribe un programa que utilice una función con el siguiente prototipo
 - `int[] rellenaPares(int longitud, int fin)`
 - La función `rellenaPares` crea y devuelve un array ordenado de la longitud especificada, cuyos elementos son números pares aleatorios comprendidos en el rango desde 2 hasta 'fin' (inclusive).
9. Escribe un programa que utilice una función con el siguiente prototipo
 - `int buscar(int t[], int clave)`
 - La función `buscar` busca de forma secuencial en el array `t` el valor `clave`. En caso de encontrarlo devuelve la posición en la que se encuentra , y en caso contrario, devuelve -1
10. Realiza un programa que utilizando un array de 12 posiciones, solicite al usuario que introduzca un mes del año, y devuelva el número de días de dicho mes. El programa debe tratar el literal del mes como no sensible a mayúsculas.
11. Realiza un programa que rellene un array de 5 posiciones con nombres de ciudades. A continuación, solicita al usuario nombres de ciudades mientras el usuario no introduzca 'salir' (no sensible a mayúsculas). El programa devuelve 'Ciudad encontrada' si se ha localizado en el array, o 'Ciudad no encontrada' en caso contrario para cada uno de los datos de entrada.
12. Modifica el programa del ejercicio11 de forma que en caso de que se encuentre la ciudad introducida por el usuario también imprima : cuántas veces aparece, y cuál es la primera y última posición en la que se ha encontrado.
13. Realiza los cambios necesarios en el programa implementado en el ejercicio 8, para no introducir valores duplicados en el array final. Considera la restricción en el dato de entrada 'fin' que debe tener un valor mayor o igual a `longitud*2`.

14. Realiza un programa que implemente una función que reciba como parámetro un array de enteros con valores duplicados y devuelva array del tamaño adecuado que contenga los valores del array recibido pero sin duplicados. Puedes reutilizar funciones implementadas en ejercicios anteriores.
15. Realiza un programa que solicite al usuario un número entero (n). A continuación, se solicita al usuario que introduzca n números que se almacenarán en un array y a partir de este se crearán dos nuevos arrays para almacenar los números pares en uno y los impares en el otro. Las nuevas tablas deben mostrarse ordenadas.
16. Realiza un programa que cree un array de enteros de 10 posiciones con valores aleatorios. A continuación, muestra el contenido del array ordenado y pide al usuario un valor para insertar.
Se utiliza una función que inserta el nuevo valor en la posición que le corresponde.
17. Escribe una función que utilice una función con el siguiente prototipo
- `int[] eliminaMayores(int t[], int valor);`
 - La función `eliminaMayores` devuelve una copia del array `t`, donde se han eliminado todos los elementos que son mayores que 'valor'
18. Realiza un programa que cree un array de enteros de 10 posiciones con valores aleatorios. A continuación, muestra el contenido del array ordenado y pide al usuario qué valor eliminar.
Se utiliza una función que elimina el valor y mantiene el array ordenado.