

1. **Suma de elementos:** Escribe una función que reciba un array de números y devuelva la suma de todos los elementos.
2. **Conteo de vocales:** Escribe una función que tome una cadena de texto como entrada y devuelva el número de vocales que contiene.
3. **Buscar el número mayor:** Escribe una función que reciba un array de números y devuelva el número más grande.
4. **Revertir una cadena:** Escribe una función que tome una cadena de texto como entrada y devuelva la misma cadena pero en orden inverso.
5. **Pares e impares:** Escribe una función que reciba un array de números y devuelva dos arrays separados, uno con los números pares y otro con los números impares.
6. **Eliminar duplicados:** Escribe una función que tome un array como entrada y devuelva un nuevo array sin ningún elemento duplicado.
7. **Palíndromo:** Escribe una función que tome una cadena de texto como entrada y devuelva true si es un palíndromo (se lee igual de adelante hacia atrás que de atrás hacia adelante), y false si no lo es.
8. **Generador de tablas de multiplicar:** Escribe una función que tome un número como entrada y devuelva una tabla de multiplicar para ese número.
9. **Calcular el factorial:** Escribe una función que tome un número como entrada y devuelva su factorial.

MANIPULAR TEXTO

10. **Contar palabras:** Escribe una función que tome una cadena de texto como entrada y devuelva el número de palabras que contiene.
11. **Contar letras:** Escribe una función que tome una cadena de texto como entrada y devuelva un objeto que contenga el recuento de cada letra en la cadena.
12. **Eliminar caracteres especiales:** Escribe una función que tome una cadena de texto como entrada y devuelva una nueva cadena que contenga solo letras y números, eliminando cualquier carácter especial.
13. **Capitalizar palabras:** Escribe una función que tome una cadena de texto como entrada y devuelva una nueva cadena en la que la primera letra de cada palabra esté en mayúscula y el resto en minúscula.
14. **Invertir palabras:** Escribe una función que tome una cadena de texto como entrada y devuelva una nueva cadena en la que el orden de las palabras esté invertido.
15. **Eliminar espacios en blanco:** Escribe una función que tome una cadena de texto como entrada y devuelva una nueva cadena en la que se hayan eliminado todos los espacios en blanco.
16. **Contar ocurrencias de una palabra:** Escribe una función que tome una cadena de texto y una palabra como entrada, y devuelva el número de veces que aparece la palabra en la cadena.
17. **Reemplazar palabras:** Escribe una función que tome una cadena de texto, una palabra a buscar y una palabra de reemplazo, y devuelva una nueva cadena en la que todas las ocurrencias de la palabra a buscar se reemplacen por la palabra de reemplazo.

18. **Eliminar duplicados en una cadena:** Escribe una función que tome una cadena de texto como entrada y devuelva una nueva cadena en la que se hayan eliminado todas las palabras duplicadas.
19. **Contar letras mayúsculas y minúsculas:** Escribe una función que tome una cadena de texto como entrada y devuelva un objeto que contenga el recuento de letras mayúsculas y minúsculas en la cadena.
20. **Convertir a camel case:** Escribe una función que tome una cadena de texto en formato "snake_case" o "kebab-case" y la convierta en "camelCase".

ARRAYS

21. **Ordenar números:** Escribe una función que tome un array de números como entrada y devuelva un nuevo array con los números ordenados de menor a mayor.
22. **Buscar elementos:** Escribe una función que tome un array y un elemento como entrada, y devuelva true si el elemento está presente en el array, y false si no lo está.
23. **Eliminar duplicados:** Escribe una función que tome un array como entrada y devuelva un nuevo array que contenga solo los elementos únicos del array original, eliminando cualquier duplicado.
24. **Suma de matrices:** Escribe una función que tome dos matrices como entrada y devuelva una nueva matriz que sea la suma de las dos matrices originales.
25. **Calcular promedio:** Escribe una función que tome un array de números como entrada y devuelva el promedio de todos los números en el array.
26. **Filtrar números pares:** Escribe una función que tome un array de números como entrada y devuelva un nuevo array que contenga solo los números pares del array original.
27. **Buscar el índice de un elemento:** Escribe una función que tome un array y un elemento como entrada, y devuelva el índice del primer elemento encontrado en el array. Si el elemento no está presente, devuelve -1.
28. **Unir arrays:** Escribe una función que tome dos arrays como entrada y devuelva un nuevo array que sea la concatenación de los dos arrays originales.
29. **Eliminar elementos repetidos:** Escribe una función que tome un array como entrada y devuelva un nuevo array en el que se eliminen todas las ocurrencias repetidas de cada elemento.
30. **Buscar el número más frecuente:** Escribe una función que tome un array de números como entrada y devuelva el número que aparece con más frecuencia en el array.

BARAJA ESPAÑOLA: ARRAYS MULTIDIMENSIONALES Y OBJETOS

La baraja española consta de 40 cartas divididas en cuatro palos: copas, oros, espadas y bastos. Cada palo contiene 10 cartas numeradas del 1 al 7, y tres cartas con figuras: sota, caballo y rey. Se te pide crear un programa que genere una baraja española representada como un array multidimensional en JavaScript.

Vamos a implementar la baraja.

Instrucciones:

1. Define un array multidimensional llamado ``baraja`` que represente la baraja española. Cada palo debe ser un array independiente dentro de ``baraja``, y cada carta debe ser un objeto con propiedades para el valor y el nombre de la carta.
2. Utiliza un bucle para generar las cartas de cada palo. Para las cartas numeradas, asigna el valor numérico y el nombre de la carta (por ejemplo, `{ valor: 1, nombre: 'As' }`). Para las cartas con figuras, asigna el valor correspondiente y el nombre de la carta (por ejemplo, `{ valor: 10, nombre: 'Sota' }`).
3. Una vez generada la baraja, imprime en la consola el array ``baraja`` completo para verificar que se ha generado correctamente.

Pistas:

```
[  
  [ { valor: 1, nombre: 'As' }, { valor: 2, nombre: 'Dos' }, ... ],  
  [ { valor: 1, nombre: 'As' }, { valor: 2, nombre: 'Dos' }, ... ],  
  [ { valor: 1, nombre: 'As' }, { valor: 2, nombre: 'Dos' }, ... ],  
  [ { valor: 1, nombre: 'As' }, { valor: 2, nombre: 'Dos' }, ... ]  
]
```

```
console.log(baraja);
```

Ahora vamos a jugar a las cartas.

Utilizando la baraja española generada en el ejercicio anterior, vamos a implementar un juego simple de cartas en el que dos jugadores compiten para ver quién tiene la carta de mayor valor.

Instrucciones:

1. Implementa una función llamada ``jugar`` que reciba como entrada dos jugadores y la baraja de cartas.
2. La función deberá repartir 5 cartas aleatorias a cada jugador desde la baraja.
3. Luego, la función deberá comparar las cartas de cada jugador y determinar cuál es la carta de mayor valor en cada ronda.

4. La función deberá imprimir en la consola el resultado de cada ronda, indicando qué jugador ganó la ronda y cuál fue la carta ganadora.

5. Al finalizar las 5 rondas, la función deberá imprimir en la consola el resultado final del juego, indicando qué jugador ganó la mayoría de las rondas y cuántas rondas ganó en total.

Pistas:

```
// Definir la baraja española y la función jugar
```

```
// Jugadores
```

```
const jugador1 = "Jugador 1";
```

```
const jugador2 = "Jugador 2";
```

```
// Jugar una partida
```

```
jugar(jugador1, jugador2, baraja);
```

```
//Resultado esperado (ejemplo parcial):
```

Ronda 1: Jugador 1 gana con la carta As de Copas

Ronda 2: Jugador 2 gana con la carta Caballo de Bastos

```
//Resultado final del juego.
```

Resultado final: Jugador 2 gana 3 de 5 rondas

Este enunciado ampliado introduce la idea de un juego de cartas simple utilizando la baraja española generada previamente. Los alumnos tendrán la oportunidad de aplicar sus conocimientos de arrays y objetos para implementar la lógica del juego y practicar conceptos como generación aleatoria y comparación de valores.