

- 1) Explicar qué es lo que hace el siguiente código. También determine el tamaño de el array, y el valor de cada una de sus elementos:

```
let array = new Array(5);
array[3] = 5;
array[2] = array[3] + 3;
array[1] = array[3] + array[2];
array[4] = null;
```

- 2) Dado el siguiente código:

```
const largo = 5;
const array = new Array(largo);
valor = 2;
for (let i = 1; i < 4; i++) {
  array[i] = valor;
  valor *= valor;
}
```

Indicar el tamaño del array, y el valor de cada uno de sus elementos.

- 3) Programar el método **modifyElement**(array, index, newValue) que reciba un array, un índice, y un nuevo valor. La función se encarga de modificar el valor del array, en el índice dado, por el valor dado.
- 4) Programar una función **printArray**(array) que recibe un arreglo e imprima en alertas cada uno de sus elementos, indicando el índice y su valor.
- 5) Programar la función **diezEnDiez**(array) , que recibe un array y lo carga de la siguiente forma: en la celda cero, carga cero, en la celda 1 carga 10, en la celda 2 carga 20, ... , en la celda 10 carga 100, ..., en la celda 100 carga 1000
- 6) Programar el método **printReverse**(array) que recibe un array e imprime los elementos desde el último al primero.
- 7) Programar el método **arrayAlCuadrado**(array) que recibe un array de numeros, y retorna un nuevo array, surgido de elevar al cuadrado cada elemento del array anterior. Nota: No se debe modificar el array original

- 8) Programar el método **changeStatus**(array) que reciba un array únicamente de booleanos (si hay uno de sus elementos que es un booleano, no se deberá hacer nada). Este método se encargará de modificar el array recibido. Modificará todos los elementos que sean false, a true. En caso de haber elementos que ya tienen valor true, muestra un mensaje por consola indicando que el índice x ya es true. Al final, deberá retornar el array modificado.
- 9) Modificar el método de la función anterior, para que permita enteros (únicamente ceros o unos) o strings ("0" o "1"), donde el cero simbolizará un false, y el 1 un true. En caso que el valor sea 0 o 1 (ya sea un string o number), se modificará a true (boolean).
- 10) Programar la función **isOrdered**(array) que recibe un array de numbers. Esta función retornará 1 si el array está ordenado de menor a mayor, -1 si está ordenado de mayor a menor, y 0 si está desordenado.
- 11) Programar la función **convertirArrayConjunto**(array) que recibe un array de números, y retorna un nuevo array a partir del array anterior, pero con la condición de que no puede haber ningún valor repetido.
- 12) Programar la función **concat**(array1, array2) que se encarga de unir dos arrays, pero de forma inversa. Es decir, deberá retornar un array que primero tenga los elementos del array2, y luego los del array1.
- 13) Programar la función **operationsWithArray**(array) que recibe un array de enteros, y muestre por consola lo siguiente:
 - a) La suma de todos sus elementos.
 - b) El promedio de todos sus elementos.
 - c) El valor máximo y el valor mínimo.
- 14) Programar una función **printPar**(array) que reciba un array, el cual únicamente pueda tener un tamaño par (en caso de no ser así, retornar false y salir de la función). Esta función se encargará de mostrar por consola únicamente los elementos con índice par del array, empezando desde el 2.