Array functions exercises - solutions

1. Sorting

Create a function which expects an array of strings as parameter then sort the array in ascending order by the length of the strings.
Example: if the input is **['apple', 'cat', 'cucumber']** then the output should be **['cat', 'apple', 'cucumber']**
Hint: sort

```
const f = (array) => {
 array.sort((s1, s2) => {
   // Ascending order by string length
   // It will be called for every item in the array
   if (s1.length < s2.length) {
     // s1 will be ahead of s2 based on our sorting
     return -1;
   } else if (s1.length === s2.length) {
     // s1 and s2 has equal length
     return 0;
   } else {
     // s2 will be ahead of s1 based on our sorting
     return 1;
   }
 });
};

// Same solution in a much more compact format
const f2 = (array) => {
 array.sort((s1, s2) => s1.length - s2.length);
};

const array = ['apple', 'cat', 'cucumber'];
f(array); // The sort function operates on reference, so it modifies the array inplace so we don't have to return with the sorted array
console.log(array);

const array2 = ['apple', 'cat', 'cucumber'];
f2(array2);
console.log(array2);
```

2. Transformation

Create a function which expects an array of numbers as parameter then assumes that these input values were temperatures in **°C** and then convert them to **°F**. At the end return the array of number in **°F**.

Example: if the input is **[20, 24]** then the output should be **[68, 75.2]**

Hint:

- **°C** - **°F** conversion: (0**°C** × 9/5) + 32 = 32**°F**
- map

```
const f = (array) => {
 // The map function does not operate inplace, it always returns a new array
 // so we have to return with this new array
 return array.map(i => i * 9/5 + 32);
}

console.log(f([0, 14, 32, 3, -5]));
```

3. Find minimum and maximum value

Create a function which expects an array of numbers and a boolean flag as parameters then returns the highest or the lowest (based on the boolean flag) value from the array.

Example: if the input is **[1, 2, 3, 3], true** then the output should be **3**

Hint:

- Math.min
- Math.max
- spread operator

```
const f = (array, isMin) => {
 // If we use the name of the function without () then we just refer to that
function and not calling it
 // We can store this reference in a variable and call it later on (so function is
a first class member in JavaScript)
 const fn = isMin ? Math.min : Math.max;
 // spread operator:
 // [1, 2, 3] -> 1, 2, 3
 // array -> comma separated values
 return fn(...array);
}

console.log(f([1, 2, 3, 3], true));
console.log(f([1, 2, 3, 3], false));
```

4. Find

Create a function which expects an array of objects as parameter then finds the object in the array where the object's age key's value is **5** and then returns this whole object.

Example: if the input is **[{name: 'John', age: 12}, {name: 'Sara', age: 5}, {name: 'Ben', age: 5}]** then the output should be **{name: 'Sara', age: 5}**

Hint:

- find

```
const find = (array) => {
 // We create a predicate function for find and it will be called for each item of the array, until we find the first matching element
 return array.find(i => i.age === 5);
}

const john = {name: 'John', age: 12};
const sara = {name: 'Sara', age: 5};
const ben = {name: 'Ben', age: 5};
console.log(find([john, sara, ben]));
console.log(find([john, ben, sara]));
```

5. Delete

Create a function which expects an array and a number (index) as parameters then delete the item at the given index.

Example: if the input is **['apple', 'cat', 'cucumber'], 1** then the output should be **['apple', 'cucumber']**

Hint:

- splice

```
const del = (array, index) => {
 // First parameter: which index should be the first to delete, second parameter: how much item do we delete
 array.splice(index, 1);
};

const array = ['apple', 'cat', 'cucumber'];
del(array, 1); // splice is operates inplace so it modifies the original array

console.log(array)
```

6. Complex exercise

Create a function which expects an array of numbers as parameter then do the following operations on this array:

- Multiply every item by 2
- Finds the lowest item in the array and prints it to the console
- Calculate the sum of the values in the array and prints it to the console

Example: if the input is **[2, 5, 8]** then the output should be:

- **[4, 10, 16]**
- **4**
- **30**

```
const multipleByTwo = (numArray) => numArray.map((n) => n * 2);

// Find the lowest number in the array
const findMin = (numArray) => Math.min(...numArray);

const sum = (numArray) => {
 return numArray.reduce((acc, l) => acc + l, 0);
};

const f = (numArray) => {
 const multipledByTwo = multipleByTwo(numArray);
 console.log(multipledByTwo);
 console.log(findMin(multipledByTwo));
 console.log(sum(multipledByTwo));
};

const array = [3, 5, 0, 6, 7, 11];
f(array);
```