Exercise 1

Using a function declaration, write a function called `analyzeArray` that only takes one parameter: **an array of numbers**. This function should loop through the array and return an object containing two pieces of information:

- `count`: Teh total number of numbers in the array
- `sum`:The total sum of numbers in the array

Important Note: Only increment the count if the element if of type `number`. You can use the `typeof` operator to do this. eg.

```
const isDavidAGirl = false;
const name = 'Adanna';
console.log(typeof bool); // boolean
console.log(typeof name); // string
```

Call your function with the following arguements:

```
analyzeArray([10, 20, 30, 40]); // Should return: {count: 4, sum: 100}
analyzeArray([45, 60, 20, 15, 35]); // Should return {count: 5, sum: 175}
analyzeArray([]); // Should return {count: 0, sum: 0}
analyzeArray([2, 40, 'watermelon', 5]); // Should return {count: 3, sum:
47}
```

Exercise 2

Write an arrow function called `updateInventory` that takes two parameters:

- An object representing an inventory where keys are item names (string), and values are quantities (number)
- An array of objects where each object represent a new item to add or update in the inventory. Each of these objects will have `item` (string) and `quantity` (number) properties.

The function should loop through the array of new items. For each item, if it already exists in the inventory object, add the new quantity to the existing quantity. If the item doesn't exist in the inventory, add it to the inventory object with its quantity. The function should return the updated inventory object.

Use the arguement in the example below to call your function:

```
let myInventory = { apples: 5, bananas: 10 };
let newItems = [
      { item: 'apples', quantity: 3 },
      { item: 'oranges', quantity: 7 },
];
```

```
updateInventory(myInventory, newItems); // Should return: { "apples": 8,
"bananas": 10, "oranges": 7 }
```

Exercise 3

Write a function (with any style of your chice) called `filterEvenNumbers` that takes one parameter: an array of numbers. The function should loop through the array and return a new array containing only the even numbers from the original array.

```
filterEvenNumbers([1, 2, 3, 4, 5, 6]); // Should return: [2, 4, 6]
filterEvenNumbers([7, 9, 11]); // Should return: []
filterEvenNumbers([2, 6, 8, 11]); // Should return [11]
```