Exercise 1

You need to create a getSumOfSequence function that takes one number parameter. This function should create an array of numbers from 0 to number. That is, if a number equal to 5 were passed, then the array should look like this: [1, 2, 3, 4, 5]. Use a for loop to populate an array with elements.

The getSumOfSequence function must return the sum of the first and last elements of the resulting array.

Task 2

Imagine that we are developing a queuing system for an establishment. For example, for mail. To do this, we created the peopleWaiting array, which displays the current status of the queue.

const peopleWaiting = ['Kristina', 'Oleg', 'Kirill', 'Maria', 'Svetlana', 'Artem', 'Gleb'];

Logically, the next package will be received by Kristina, after her, Oleg will be the first in line.

Now you need to implement the following logic in code step by step:

Kristina and Oleg received the parcels and left the queue. You need to remove them from the array.

Now it's Kirill's turn. And suddenly the postal worker says that they will soon have a lunch break and she will have time to serve only Kirill. Therefore, all the other people behind Kirill decided not to wait until lunch was over and simply left the post office. You must first remove Kirill from the peopleWaiting array, and then remove the people who did not have time to receive the packages.

When a person receives a package, it is necessary to display in the modal window the message “name received the package. There are length people left in the queue.” (Replace name with the name of the person who received the package, and length with the number of people left in the queue.)

If the person did not receive the package and left the queue, then display the message “name did not receive the package and left the queue” in the modal window via alert.

It is recommended to create 2 functions: giveParcel - for issuing a parcel and removing a client from the beginning of the array, leaveQueueWithoutParcel - for removing a client that has not received a parcel from the end of the list.

Task 3

You are given an array of numbers.

const numbers = [10, 4, 100, -5, 54, 2];

You need to calculate the sum of the cubes of all the numbers in the array. For example, for the array [1, 5, 9], the result should be:

1^3 + 5^3 + 9^3 = 1 + 125 + 729 = 855

Solve this problem in 4 ways:

Through the for loop

Through the for of loop

Through the forEach method

Through the reduce method

Given the numbers array, the sum of the cubes should be 1158411.

Task 4

Imagine that you have a friend, Alexei, who owns a coffee shop. He wants to make a website for this coffee shop so that people can order coffee delivered to their homes. The most important feature that Alexey wants to see is the search for coffee by name. Your friend asks you to help him develop a website.

Initially, you have an array of coffees, which stores all types of coffee that Alexey has in his coffee shop. Drinks are distributed in order of popularity among customers. Those. “Latte” coffee is the most popular.

const coffees = ['Latte', 'Cappuccino', 'Americano'];

With the prompt function, you need to ask for the name of the coffee. Pass the message “Search coffee by name:” to prompt. Store the value that the user entered into the coffeeName variable.

If the user-entered coffee name exists, then display a message with alert “Keep your favorite coffee coffee. It is number-th most popular in our coffee shop.”, where coffee is the name of the found coffee, number is the number of the found coffee in the coffees array. If no coffee was found, then display the message “Unfortunately, this type of coffee is not available”.

Conditions:

The coffee name must be case-insensitive. Those. if the user types "lATte" it should show the result with coffee "Latte"

You need to use findIndex.

Task 5

You have developed the main functionality that Alexey wanted to see on his site. He says thank you very much, because thanks to the site, the coffee shop has increased the number of sales. Now Alexei wanted to raise the price of coffee to make even more profit.

You initially have 2 arrays: coffees (stores coffee names) and prices (stores coffee prices). Indexes for coffee names and prices are interconnected. Those. “Latte” costs 1.5 euros, “Cappuccino” costs 1 euro, etc.

const coffees = ['Latte', 'Cappuccino', 'Americano'];

const prices = [1.5, 1, 2];

You need to create a new updatedPrices array that will contain the same numbers from the prices array, but only increased by 0.5.

After the price increase, you need to notify the customers of the coffee shop, so display a message for each type of coffee via alert “Coffee name is now worth price euro”, where name is the name of the coffee and price is the updated price of the coffee.

Conditions:

You must use the map and forEach array methods

Task 6

The last thing Alexei wants to see on his website is a rating system for the coffee shop. The owner needs to know the opinion of customers in order to correct shortcomings in his work.

First, you need to create an empty clientsEstimations array that will hold the coffee shop's customer ratings. Also create an askClientToGiveEstimation function that should display the message “How do you rate our coffee shop from 1 to 10?” via prompt. The end result that the user enters must be a number data type. If a number from 1 to 10 was entered, then add this estimate to the clientsEstimations array, otherwise do nothing.

To add estimates, call the askClientToGiveEstimation function 5 times. It is recommended to do this through a for loop.

After the ratings are added, you need to count the positive and negative ratings. A positive estimate is a number greater than 5, a negative estimate is a number less than 5 or equal to 5. Display the alert message “Total positive estimates: goodEstimations; Total negative ratings: notGoodEstimations”, where goodEstimations is the number of positive ratings and notGoodEstimations is the number of negative ratings.

Conditions:

The solution should use the filter array method.

Task 7

Imagine that your friend is the owner of an amateur football club. Team players score a lot of goals and club employees are tired of counting all the statistics every time after the match. Therefore, they decided to ask you as a web developer to help solve this problem. You need to optimize the task of counting statistics using the JavaScript programming language.

You have an array of goals, where each index represents the match played, and the number below the index represents the number of goals scored in that particular match.

const goals = [8, 1, 1, 3, 2, -1, 5];

That is, in the 1st match, the team scored 8 goals, in the 2nd - one, etc. If there is a negative number in the array, it means that the team was given an automatic defeat.

Using this goals array, you need to display the following statistics for the entire season.

Most scoring game in terms of goals. Display in the modal window via alert the message “The most scoring match was at number number. There were numberOfGoals scored in it.” (replace number with the number of the match, and numberOfGoals with the number of goals in the highest scoring match).

Worst games ever. There are several of them in the array, so you need to display the message “The most unsuccessful matches were under the numbers numbers. In each of them, numberOfGoals ball(s) were scored.” (replace numbers with serial numbers of matches and display them separated by commas, and numberOfGoals - with the number of goals in the most unscoring match). Do not take into account games with automatic defeat.

Total number of goals for the season. Do not take into account games with automatic defeat. Display a message via alert “The total number of goals for the season is equal to numberOfGoals” (replace numberOfGoals with the number of the total number of goals for the season).

Were there automatic defeats. If there were, then display a message via alert “There were automatic defeats: yes”, otherwise “There were automatic defeats: no”.

Average number of goals per match. Display a message via alert “The average number of goals per match is equal to numberOfGoals” (replace numberOfGoals with the average number of goals per match).

Sort the goals in ascending order and display all results separated by commas in the alert modal. The goals array must not be modified.