Session 5 & 6

Notes:

1- Make any needed validations.
2- Use "===" not "==" in conditions.
3- Don't use "var" for declaring variables.
4- Use the appropriate methods for the needed task.
5- Beware of corner cases.
6- Follow the same input and output format.
7- Read the following tasks carefully in order not to lose marks.
8- All the following functions should be in only one JavaScript file.

Task 1:

For a number of test cases **n** which is read from the user, in each test case you will read an **array of strings** from the user. Capitalize the first character from each word and make the rest of the characters lower case.

Required:

Create a function called **Task1** which takes no arguments, does the specified above, and returns an array of arrays of strings, each array of strings represents a test case.

Sample Test cases:

Inputs	Outputs
Enter the number of test cases: **2** 1: Enter your words: **apples BANANAS** 2: Enter your words: **hELlo worLD**	[["Apples", "Bananas"], ["Hello", "World"]]
Enter the number of test cases: **1** 1: Enter your words: **a**	[["A"]]

Task 2:

For a number of test cases **n** which is read from the user. In each test case you will read an **array of numbers** from the user. Be sure to **convert** the numbers from strings to real numbers. **Sort** the array ascendingly, then calculate the **sum** of the numbers in the array. Then **create** another array which contain the numbers from the original array, but only the numbers which are **greater than** the half of the length of the original array. After all of this you will end up with three variables, the original array, the sum, and the created array. With those three, create an object which contain them.

Required:

Create a function called **Task2** which takes no arguments, does the specified above, and returns an array of objects, each object represents a test case.

Sample Test cases:

Inputs	Outputs

Enter the number of test cases: **2**
Enter the numbers of array 1: **12 3 9 4 10 6**
Enter the numbers of array 2: **0.5**
	[
	  {
	    createdArray: [4, 6, 9, 10, 12],
	    originalArray: [3, 4, 6, 9, 10, 12],
	    sum: 44
	  },
	  {
	    createdArray: [],
	    originalArray: [0.5],
	    sum: 0.5
	  }
	]

Task 3:

With the array returned from **Task2** function, loop through its objects and calculate the product of the numbers in the createdArray property **using eval function**.

Required:

Create a function called **Task3** which takes the returned array from **Task2** function as an argument, does the specified above, and returns an array with results, each result represents a product of a createdArray.

Sample Test cases:

Considering the array returned from the test case in Task 2:

Inputs	Outputs

[
  {
    createdArray: [4, 6, 9, 10, 12],
    originalArray: [3, 4, 6, 9, 10, 12],
    sum: 44
  },
  {
    createdArray: [],
    originalArray: [0.5],
    sum: 0.5
  }
]
	[25920]

In the previous test case, the output was an array with only one number in it, although there are two createdArrays, but the second createdArray is empty, therefore there is no product.

Task 4:

Given an array of numbers, calculate their average.

Required:

Create an asynchronous function called **Task4** which takes an array as an argument, the function should return a promise which either resolve the average of the array or reject with an error message, there should be a time out of at least 1 second.

Call the function, and if there are no errors, print the product of the average and the arrays length in the console, else throw an error.

Sample Test cases:

Inputs	Outputs
[1, 2, 3, 4]	Result: 10
[]	Error

Deliverables:

Only deliver the JavaScript file with the 4 functions and their calls inside it.

Don't submit anything else.

Don't submit .ZIP.