JS Practice Problems

References for array : https://www.geeksforgeeks.org/array-data-structure/
References for string : https://www.geeksforgeeks.org/string-data-structure/
Problem solving session:

Mandatory !!! — Please watch the below two sessions - it's mandatory to watch them as you proceed further because the video covers all the basics that are required for problem solving.

Session 1:
https://drive.google.com/file/d/1qMjz6miH_KfcK2i1TFkBrQrbr4HqMtaR/view?usp=share_link

Topics covered in Session 1:

1. How to approach an interview
2. Basics in JS
3. Loops
4. conditional statements
5. Optional chaining
6. Time complexity intro
7. Conditional and loops using debugger

Problems discussed:

1. Array sorting

easy,medium level started

Session 2:
https://drive.google.com/file/d/1QFpJRuGgamzIX2bHaiUiRFmW05_xPLDK/view?usp=share_link

Topics covered in Session 2:

1. How to apporch medium level ,hard problems
2. How to handle pressure situations in interview
3. Time complexity
4. Solving some medium and hard level problems

Rules: Solve the problem without methods.

Please donot use methods. You need to do it with loops.

Make sure you try each problem in 2-3 different ways and also try to understand time complexity(time taken to execute) as well.

Difficulty Level : Easy

1. Create a function that takes two numbers as arguments and returns their sum.
2. Write a function that takes an integer `minutes` and converts it to seconds.
3. Create a function that takes a number as an argument, increments the number by +1 and returns the result.
4. Create a function that takes the age in years and returns the age in days.
5. sniCreate a function that takes `voltage` and `current` and returns the calculated **power**.
6. Write a function that returns the string `"something"` joined with a space `" "` and the given argument `a`.
7. Create a function that takes two arguments. Both arguments are integers, `a` and `b`. Return `true` if one of them is `10` or if their sum is `10`.
8. Create a function that takes two strings as arguments and returns either `true` or `false` depending on whether the total number of characters in the first string is equal to the total number of characters in the second string.
9. Create a function that takes a name and returns a greeting in the form of a string. Don't use a normal function, use an **arrow function**.
10. Create a function that takes an array of 10 numbers (between 0 and 9) and returns a string of those numbers formatted as a phone number (e.g. **(555) 555-5555**).
11. Create a function that returns an array of strings sorted by length in **ascending** order.

    Example:

    sortByLength(["a", "ccc", "dddd", "bb"]) ➞ ["a", "bb", "ccc", "dddd"]

12. Create a function that takes an array of arrays with numbers. Return a new (single) array with the largest numbers of each.

    Example:

    1. findLargestNums([[4, 2, 7, 1], [20, 70, 40, 90], [1, 2, 0]]) ➞ [7, 90, 2]

13. Create a function that takes an array of numbers and returns the **second largest** number.

    Example:

    secondLargest([10, 40, 30, 20, 50]) ➞ 40

14. Create a function that takes an array of items, removes all duplicate items and returns a new array in the same sequential order as the old array (minus duplicates).

    Example:

    removeDups([1, 0, 1, 0]) ➞ [1, 0]

    removeDups(["The", "big", "cat"]) ➞ ["The", "big", "cat"]

15. Create a function that takes an array of integers as an argument and returns a unique number from that array. All numbers except unique ones have *the same number of occurrences* in the array.

    Example:

    findSingleNumber([2, 2, 2, 3, 4, 4, 4]) ➞ 3

16. Create a function that takes two strings as arguments and returns the number of times the first string (the single character) is found in the second string.

    Example:

    charCount("c", "Chamber of secrets") ➞ 1

17. Create a function that takes a string and returns the number (count) of vowels contained within it.

    Example:

    countVowels("Celebration") ➞ 5

18. Given a string, create a function to reverse the case. All lower-cased letters should be upper-cased, and vice versa.

    Example:
    reverseCase("Happy Birthday") ➞ "hAPPY bIRTHDAY"

19. Take one integer n, loop till n and pass each value to a function, create a function that takes one integer parameter, and multiply with 2 in every integer.

    Input: n=5
    Output: 2 4 6 8 10

Explanation: Loop start with 1 go till 5 bcoz n=5
1 x 2 =2, 2 x 2=4, 3 x 2=6 …..etc

20. Create Function that will take one parameter and return type of the data.

Input: 500
Output: Integer

Input: Coding
Output: String

21. Program to find greatest of three numbers(using ternery operator).

Input: 4 8 2
Output: 8 is gretest

22 . C Program to find factorial of number.

Input: n=5
Output: 120

Explanation: 5 x 4 x 3 x 2 x 1 = 120

23. C Program to arrange numbers in ascending order

Input: [2,3,1,5,4]
Output: [1,2,3,4,5]

Sort the Array using loop only(you can not use predefined function).

24. Print Patter using loop.

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

25. C Program to Calculate the Power of a Number(using loop only).

Input: n=5, p=3
Output: 5 ^ 3 = 125
Explanation: 5 x 5 x 5 = 125

26. Program to Check Whether a Number is Prime or Not

Input: 9
Output: 9 is not a prime no

Input: 7
Output : 7 is a prime no

27. Program to find LCM of two numbers using while loop

Input: 15 50
Output: Lcm of 15 and 50 is 150.

28. Program to Display Characters from A to Z Using Loop with count.

Output: A1 B2 C3 D4 E5 F6 ……. Z26

29. Program to find a missing number

Input: n=5(length of array), arr= [5,3,1,4]
Output: 2 is missing

Using loop only(you can not use predefined function)

30. Program to count vowels and consonants in a given String.

Input: i am ram

Output: 3 vowels 3 consonants.

31. program to insert the elements of an array for specific index.

Input: [1,2,3,4,5,7,8,9,10] , index=5
Output: [1,2,3,4,5,6,78,9,10]

32. Reverse a number using while Loop

Input: 123
Output: 321

33. Count occurrence of number:

Input: [1,6,3,1,5,9,7,2,1,9,3,7,8,9,10] , no find=7
Output: 7 present 2 times.

Difficulty Level : Medium

1. Write a function that converts an object into an array, where each element represents a key-value pair in the form of an array.

`Examples :`

toArray({ a: 1, b: 2 }) ⟶ [["a", 1], ["b", 2]]

toArray({ shrimp: 15, tots: 12 }) ➞ [["shrimp", 15], ["tots",12]]

toArray({}) ➞ []

2. Create a function that takes two numbers as arguments (`num`, `length`) and returns an array of multiples of `num` until the array length reaches `length`.
`Examples :`

arrayOfMultiples(7, 5) ➞ [7, 14, 21, 28, 35]

arrayOfMultiples(12, 10) ➞ [12, 24, 36, 48, 60, 72, 84, 96, 108, 120]

arrayOfMultiples(17, 6) ➞ [17, 34, 51, 68, 85, 102]

3. Create the function that takes an array with objects and returns the sum of people's budgets.

`Examples :`

```
getBudgets([
  { name: "John", age: 21, budget: 23000 },
  { name: "Steve",  age: 32, budget: 40000 },
  { name: "Martin",  age: 16, budget: 2700 }
]) ➞ 65700

getBudgets([
  { name: "John",  age: 21, budget: 29000 },
  { name: "Steve",  age: 32, budget: 32000 },
  { name: "Martin",  age: 16, budget: 1600 }
]) ➞ 62600
```

4. Create a function that takes an array of objects like `{ name: "John", notes: [3, 5, 4]}` and returns an array of objects like `{ name: "John", avgNote: 4 }`. If a student has no notes (an empty array) then let's assume `avgNote: 0`.

`Example :`

```
[
  { name: "John", notes: [3, 5, 4]}
] ➞ [
  { name: "John", avgNote: 4 }
]
```

5. Create a function that moves all capital letters to the front of a word.

```
Examples :
```

capToFront("hApPy") → "APhpy"

capToFront("moveMENT") → "MENTmove"

capToFront("shOrtCAKE") → "OCAKEshrt"

6. Count each occurrence of number(can not use predefined function).

Input: [1,6,3,1,5,9,7,2,1,9,3,7,8,9,10] , no find=7
Output: 1 present 2 times.
2 present 1 times.
3 present 2 times.
5 present 1 times ……. Etc

7. Write a function that accepts an array of strings. Return the longest string(can not use predefined function).

Input: [‘nik’, ’mikhil’, ’Cow’,’Elephant’]
Output: Elephant

8. Most Commonly Used two Character in String(can not use predefined function)

Input: ‘Hii i am ram’
Output; i, a

9. Write Program to remove duplicate elements in an array and sort it in descending order(can not use predefined function).

Input: [5,3,5,2,1,1,7,3,5,6]
Output: [7,6,5,32,1]

10. Write a Program to Remove brackets from an algebraic expression(can not use predefined function).

    Input: a + b-(9+c)=3
    Output: a + b- 9+c=3

11. Write Program to remove duplicate elements in an array and sort it in Accending order(can not use predefined function).

    Input: [Z, A, P, C, A, Z , K, N, C]
    Output: [A, C, K,N, P, Z]

12. If subseq's array sequence is present in the array, returns true or else returns false.
    Let arr = [5, 7, 3, 2, 2, 7,-1, 5, -3, 13, 4]
    Example:
    Input : Subseq1 = [7, -1, 5, -3] Output: true
    Subseq2 = [7, -1, 4, -3] : false
    Subseq3 = [-1] : true
    Subseq4 = [13, -3, 4, 1] : false

13. Find sum of the Unique numbers:
    Example : Let arr = [1, 2, 2, 1, 3, 5, 1];
    The unique numbers are 1,2, 3, 5 so the sum should be 11.

Difficulty Level : Hard

1. Create a function that converts dash/underscore delimited words into camel casing. The first word within the output should be capitalized only if the original word was capitalized.

   Examples :

   toCamelCase("A-B-C") → "ABC"

   toCamelCase("the-stealth-warrior") → "theStealthWarrior"

   toCamelCase("The_Stealth_Warrior") → "TheStealthWarrior"

2. Create a function that takes an array of strings and returns an array with only the strings that have numbers in them. If there are no strings containing numbers, return an empty array.

```
Examples :
```

numInStr(["1a", "a", "2b", "b"]) → ["1a", "2b"]

numInStr(["abc", "abc10"]) → ["abc10"]

numInStr(["abc", "ab10c", "a10bc", "bcd"]) → ["ab10c", "a10bc"]

numInStr(["this is a test", "test1"]) → ["test1"]

3. Write a function that takes a list of hours and returns the total weekly salary.
    - The input list hours is listed sequentially, ordered from Monday to Sunday.
    - A worker earns $10 an hour for the first 8 hours.
    - For every overtime hour, he earns $15.
    - On weekends, the employer pays double the usual rate, regardless how many hours were worked previously that week. For instance, 10 hours worked on a weekday would pay 80+30 = $110, but on a weekend it would pay 160+60 = $220.

```
Examples :
```

weeklySalary([8, 8, 8, 8, 8, 0, 0]) → 400

weeklySalary([10, 10, 10, 0, 8, 0, 0]) → 410

weeklySalary([0, 0, 0, 0, 0, 12, 0]) → 280

4. Create a function which takes in an encoded string and returns an object according to the following example:

```
Examples :
```

parseCode("John000Doe000123") ➞ {

firstName: "John",

lastName: "Doe",

id: "123"

}

parseCode("michael0smith004331") ➞ {

firstName: "michael",

lastName: "smith",

id: "4331"

}

parseCode("Thomas00LEE0000043") ➞ {

firstName: "Thomas",

lastName: "LEE",

id: "43"

}

5. Create a function that takes the current day (e.g. "2022-09-30"), an array of date objects and returns the "current streak" (i.e. number of consecutive days in a row).

Examples :

currentStreak("2022-09-23", [

{"date": "2022-09-18"},

{"date": "2022-09-19"},

{"date": "2022-09-21"},

{"date": "2022-09-22"},

{"date": "2022-09-23"}]) ➞ 3

```
currentStreak("2022-09-25", [
{"date": "2022-09-16"},
{"date": "2022-09-17"},
{"date": "2022-09-21"},
{"date": "2022-09-22"},
```

{"date": "2022-09-23"}]) ➞ 0

6. Given a String(Note:- String Will Contain all later from A-Z except 1 letter, that letter you need to find out) :-

Input string="6 E @ f w 3 x y g N 1 o p Q A b c h i j # K l d m R T U V Z"

Output = ''S is missing from the String"

Note:- (
Time Complexity:- O(n) means only 1 loop you can use.
without using any predefined function.
)

7. Given a string s, return *the longest Palindromic Substring*

Input: s = "babad"
Output: "bab"
Explanation: "aba" is also a valid answer.

Input: s = "cbbd"
Output: "bb"

8. Given an unsorted array of integers nums, return the length of the longest consecutive elements sequence.
You must write an algorithm that runs in O(n) time.

Input: nums = [100,4,200,1,3,2]

Output: 4

Explanation: The longest consecutive elements sequence is [1, 2, 3, 4]. Therefore its length is 4.

Input: nums = [0,3,7,2,5,8,4,6,0,1]

Output: 9

9. Take an array of strings and create a dropdown using array values the values of dropdown should be taken dynamically and if we select a value in dropdown and the length of dropdown value is even it should show 1 in console or else if it is odd it should 0.