1. Magical String (Encrypted Sequence)

Description: Convert a list of integers into a binary-like string where each element becomes '1' if double its value exists, else '0'.

Sample Input: [2, 4, 5, 8, 3] **Sample Output:** "11000"

Easy Approach: - Use a Set for O(1) lookup. - Map each number to '1' or '0' depending if double exists.

JavaScript Solution:

```
function magicalString(nums) {
    const numSet = new Set(nums);
    return nums.map(x => numSet.has(2 * x) ? '1' : '0').join('');
}
console.log(magicalString([2, 4, 5, 8, 3])); // 11000
```

2. Find Consecutively Repeated Words (Chat Repetition Detector)

Description: Detect consecutive repeated words (case-insensitive) and return each repeated word once in order.

Sample Input: "Please please help me me quickly Quickly" **Sample Output:** ['please','me','quickly']

Easy Approach: - Convert text to lowercase. - Split into words and compare each word with the previous one.

JavaScript Solution:

```
function findRepeatedWords(text) {
    const words = text.toLowerCase().split(' ');
    const result = [];
    for (let i = 1; i < words.length; i++) {
        if (words[i] === words[i-1] && !result.includes(words[i])) {
            result.push(words[i]);
        }
    }
    return result;
}
console.log(findRepeatedWords("Please please help me me quickly Quickly"));
```

3. Sort Strings by Last Character (Username Sorting)

Description: Sort usernames based on the last character; preserve order if last characters match.

Sample Input: ["alex", "ryan", "mona", "chris"] **Sample Output:** ["mona", "alex", "ryan", "chris"]

Easy Approach: - Use array sort with custom comparator using last character.

JavaScript Solution:

```
function sortByLastChar(arr) {
    return arr.slice().sort((a, b) => a.slice(-1).localeCompare(b.slice(-1)));
}
console.log(sortByLastChar(["alex", "ryan", "mona", "chris"]));
```

4. Replace Every K-th Word with Its Length (Caption Shortener)

Description: Replace every k-th word in a sentence with its length.

Sample Input: k=2, "Coding challenges are amazing to solve" **Sample Output:** "Coding 10 are 7 to 5"

Easy Approach: - Split sentence into words. - Loop every k-th word and replace with length.

JavaScript Solution:

```
function replaceKthWordWithLength(k, sentence) {
    const words = sentence.split(' ');
    for (let i = k-1; i < words.length; i += k) {
        words[i] = words[i].length.toString();
    }
    return words.join(' ');
}
console.log(replaceKthWordWithLength(2, "Coding challenges are amazing to
solve"));
```

5. Remove Words with ≥ 3 Repeated Letters (Spam Cleaner)

Description: Remove words containing three or more consecutive repeated letters.

Sample Input: "Sooo excited for this cooool event yesss absolutely" **Sample Output:** "excited for this event absolutely"

Easy Approach (No Regex): - Loop through characters in each word. - Check if any character repeats three times consecutively. - Filter out words that match.

JavaScript Solution:

```
function hasTripleRepeat(word) {
    for (let i = 0; i < word.length - 2; i++) {
        if (word[i] === word[i+1] && word[i] === word[i+2]) {
            return true;
        }
    }
    return false;
}

const sentence = "Sooo excited for this cooool event yesss absolutely";
const clean = sentence.split(' ').filter(word => !hasTripleRepeat(word)).join('
');
console.log(clean); // "excited for this event absolutely"
```