

Problem Statement

Find the First Non-Repeating Character Write a program to find the first non-repeating character in a string. For input "swiss", the output should be "w". You cannot use any built-in string or character frequency counting functions.

type scripting program

```
function firstNonRepeatingCharacter(str: string): string {

    for (let i = 0; i < str.length; i++) {
        let isUnique = true;

        for (let j = 0; j < str.length; j++) {
            if (i !== j && str[i] === str[j]) {
                isUnique = false;
                break;
            }
        }

        if (isUnique) {
            return str[i];
        }
    }

    return "";
}

console.log(firstNonRepeatingCharacter("swiss")); // Output: "w"
```

Explanation

? **Function Definition:** firstNonRepeatingCharacter takes a string str to find the first non-repeating character.

? **Outer Loop:** Iterates through each character of str.

? **Check Uniqueness:** Initializes isUnique to true for each character.

? **Inner Loop:** Compares the current character with all others. If a match is found, isUnique becomes false, and the loop exits early.

? **Return Non-Repeating Character:** If isUnique is still true, return the current character.

? **Return Empty String:** If no non-repeating character is found, return an empty string.

? **Time Complexity:** $O(n^2)$ due to nested loops.

? **Space Complexity:** $O(1)$ because only a few variables are used.

? **Example:** For "swiss", the function returns "w" as it's the first non-repeating character.

Sample Inputs and Outputs

Sample 1:

Input: swiss

Output: w

The screenshot shows the myCompiler web interface. The code editor contains the following TypeScript code:

```
1 function firstNonRepeatingCharacter(s: string): string | null {
2   const maxChar = 128;
3   const counts: number[] = new Array(maxChar).fill(0);
4   for (let i = 0; i < s.length; i++) {
5     const charCode = s.charCodeAt(i);
6     counts[charCode]++;
7   }
8   for (let i = 0; i < s.length; i++) {
9     const charCode = s.charCodeAt(i);
10    if (counts[charCode] === 1) {
11      return s[i];
12    }
13  }
14  return null;
15 }
16 const inputString = "swiss";
17 const result = firstNonRepeatingCharacter(inputString);
18 console.log(`The first non-repeating character in "${inputString}" is "${result}"`);
19
```

The output panel on the right shows the result of the execution:

```
Program input
The first non-repeating character in "swiss" is "w"
[Execution complete with exit code 0]
```

Sample 2:

Input: hello

Output: h

The screenshot shows the myCompiler web interface. The code editor contains the following TypeScript code:

```
1 function firstNonRepeatingCharacter(s: string): string | null {
2   const maxChar = 128;
3   const counts: number[] = new Array(maxChar).fill(0);
4   for (let i = 0; i < s.length; i++) {
5     const charCode = s.charCodeAt(i);
6     counts[charCode]++;
7   }
8   for (let i = 0; i < s.length; i++) {
9     const charCode = s.charCodeAt(i);
10    if (counts[charCode] === 1) {
11      return s[i];
12    }
13  }
14  return null;
15 }
16 const inputString = "hello";
17 const result = firstNonRepeatingCharacter(inputString);
18 console.log(`The first non-repeating character in "${inputString}" is "${result}"`);
19
```

The output panel on the right shows the result of the execution:

```
Program input
The first non-repeating character in "hello" is "h"
[Execution complete with exit code 0]
```

Sample 3:

Input: concatenation

Output: e

The screenshot shows a web browser window with the URL `mycompiler.io/new/typescript`. The page is titled "myCompiler" and has a navigation bar with "English", "Recent", "Login", and "Sign up" links. Below the navigation bar is a "Enter a title..." input field and a "Ctrl+Enter" button. The main area is divided into two sections: "TypeScript" and "Output".

The "TypeScript" section contains the following code:

```
1 function firstNonRepeatingCharacter(s: string): string | null {
2   const maxChar = 128;
3   const counts: number[] = new Array(maxChar).fill(0);
4   for (let i = 0; i < s.length; i++) {
5     const charCode = s.charCodeAt(i);
6     counts[charCode]++;
7   }
8   for (let i = 0; i < s.length; i++) {
9     const charCode = s.charCodeAt(i);
10    if (counts[charCode] === 1) {
11      return s[i];
12    }
13  }
14  return null;
15 }
16 const inputString = "concatenation";
17 const result = firstNonRepeatingCharacter(inputString);
18 console.log("The first non-repeating character in '${inputString}' is '${result}'");
19
```

The "Output" section shows the result of the program execution:

```
Program input
Output
The first non-repeating character in 'concatenation' is 'e'
[Execution complete with exit code 0]
```

At the bottom of the page, there is a footer with the "Auth0 by Okta" logo and a message: "All the connections, zero limits. Unlimited Okta and social connections on our Free Plan. Sign up ->".