

最长公共后缀

题目描述：

编写一个函数来查找字符串数组中的最长公共后缀；

如果不存在公共后缀，返回固定字符串： `@Zero`。

补充说明：

- 1、字符串长度范围：[2, 1000]；
- 2、字符串中字符取值范围为[1, 126]。

示例 1

输入：

`["abc", "bbc", "c"]`

输出：

`"c"`

说明：

返回公共后缀：`c`

示例 2

输入：

`["aa", "bb", "cc"]`

输出：

`"@Zero"`

说明：

不存在公共后缀，返回固定结果：`@Zero`

```
/**
 * 求最长公共后缀
 * @param strs string 字符串一维数组 字符串数组
 * @return string 字符串
 */
function longestCommonPostfix(strs) {
    // write code here

    let s = strs;
    //   .toString()
    // .replace(", ", " ")
    // .replace("[", "")
    // .replace("]", "")
```

```

    // .replace(/\\"/g, "")
    // .split(",");
    // return s;
    let ans = s[0];
    let len = s.length;
    for (let i = 0; i < len; ++i) {
        let k = 1;
        let n = ans.length;
        let m = s[i].length;
        while (n - k >= 0 && m - k >= 0 && ans.charAt(n - k) == s[i].charAt(m - k)) {
            k++;
        }
        if (k == 1) {
            return "@Zero";
        }
        ans = ans.substring(n - k + 1);
    }
    return ans;
}
module.exports = {
    longestCommonPostfix: longestCommonPostfix,
};

```

```

// const readline = require("readline");
// const rl = readline.createInterface({
//     input: process.stdin,
//     output: process.stdout,
// });
// rl.on("line", (line) => {
//     let s = line.replace("[", "").replace("]", "").replace(/\\"/g, "");
//     let ans = s[0];
//     let len = s.length;
//     for (let i = 0; i < len; ++i) {
//         let k = 1;
//         let n = ans.length;
//         let m = s[i].length;
//         // 'a'.charAt(m - k)
//         while (n - k >= 0 && m - k >= 0 && ans.charAt(n - k) == s[i].charAt(m - k)) {
//             k++;
//         }
//         if (k == 1) {
//             console.log("@Zero");
//             return;
//         }
//     }
// }

```

```
//      // 'a'.substring
//      ans = ans.substring(n - k + 1);
//  }
//  console.log(ans);
//  });
```