

# Problem of the Week – First Recurring Character in a String

**Company:** Google

## Scenario:

While building an autocomplete feature, you're analyzing the patterns of user-typed strings. You're interested in identifying the **first recurring character** — the one that appears **more than once** and **whose second occurrence happens earliest**.

This will help prioritize autocomplete suggestions based on early recurring patterns.

## Problem Statement:

Given a string `s`, return the **first character** that appears **more than once**, with the **earliest second appearance**.

If there is **no such character**, return `null`.

## Input Format:

- A **single string** `s` consisting of lowercase and/or uppercase English letters.

## Output Format:

- Return the first recurring character as a **single character string**.
- Return `null` if no character repeats.

## Examples:

### Example 1:

Input: "acbbac"

Output: "b"

- 'a' appears at index 0 and again at index 4
- 'b' appears at index 2 and again at index 3 — **earlier second occurrence**  
→ Output is "b"

### Example 2:

Input: "abcdef"

Output: null

- All characters are unique

### Example 3:

Input: "abca"  
Output: "a"

---

### Constraints:

- $1 \leq s.length \leq 10^5$
  - All characters are ASCII printable characters
- 

### Approach Hints:

- Use a **set** or **hashmap** to keep track of characters seen so far
  - Traverse string from left to right
  - As soon as a character is seen again, return it
  - If loop finishes, return `null`
- 

### Practice Links:

- [LeetCode Discuss: First Recurring Character](#)
  - [GeeksforGeeks \(Similar\): Find repeated character present first in a string](#)
- 

### Video Explanations:

- [Kevin Naughton Jr. – First Recurring Character](#)
- [FireShip Shorts – Interview Challenge \(First Recurring\)](#)