🔷 Problem Summary (Simple Words)

You are given:

- A string s
- An integer k

👉 For **every block of 2k characters**:

- **Reverse the first k characters**
- Leave the next k characters **as they are**
- If remaining characters are:
    - < k → reverse all
    - k ≤ remaining < 2k → reverse first k

🔷 Key Observation (Very Important)

We **only reverse characters in ranges**:

[start, start + k - 1]

where:

start = 0, 2k, 4k, 6k, ...

So we move in steps of **2k**.

🔷 Why Two Pointers?

To reverse a portion of the string:

- One pointer from **left**
- One pointer from **right**
- Swap characters until they cross

This is:

- Efficient
- In-place
- No extra space

🔷 Algorithm (Step-by-Step)

1. Convert string into a **mutable array**
2. Loop i from 0 to n-1 with step 2k

3. Set:
    - left = i
    - right = min(i + k - 1, n - 1)
4. Reverse characters using two pointers
5. Continue to next 2k block
6. Convert array back to string

🔷 Visual Example

Input

s = "abcdefg", k = 2

Process

i = 0 → reverse [0,1] → "bacdefg"

i = 4 → reverse [4,5] → "bacdfeg"

Output

"bacdfeg"

🔷 Time & Space Complexity

- **Time:** O(n) → each character touched once
- **Space:** O(n) (Java/Python due to mutable array)

✅ JAVA SOLUTION

```
class Solution {
   public String reverseStr(String s, int k) {
      char[] arr = s.toCharArray();
      int n = arr.length;

      for (int i = 0; i < n; i += 2 * k) {
         int left = i;
         int right = Math.min(i + k - 1, n - 1);

         while (left < right) {
```

```
            char temp = arr[left];
            arr[left] = arr[right];
            arr[right] = temp;
            left++;
            right--;
        }
    }
    return new String(arr);
  }
}
```

✅ **C++ SOLUTION**

```
class Solution {
public:
  string reverseStr(string s, int k) {
    int n = s.size();

    for (int i = 0; i < n; i += 2 * k) {
      int left = i;
      int right = min(i + k - 1, n - 1);

      while (left < right) {
        swap(s[left], s[right]);
        left++;
        right--;
      }
    }
    return s;
  }
};
```

✅ **PYTHON SOLUTION**

```
class Solution:
    def reverseStr(self, s: str, k: int) -> str:
        arr = list(s)
        n = len(arr)

        for i in range(0, n, 2 * k):
            left = i
            right = min(i + k - 1, n - 1)

            while left < right:
                arr[left], arr[right] = arr[right], arr[left]
                left += 1
                right -= 1

        return "".join(arr)
```

🔷 **Common Interview Mistakes ❌**

- Forgetting min(i + k - 1, n - 1)
- Reversing every k instead of every 2k
- Using extra stacks unnecessarily

🔷 **Final Takeaway □**

This problem is NOT about strings — it is about index control and block-wise reversal.

If you master:

- for (i += 2k)
- two pointers
- min boundary

👉 You will **never fail this question**.