You are given a string s consisting of digits and an integer k.

A **round** can be completed if the length of s is greater than k. In one round, do the following:

1. **Divide** s into **consecutive groups** of size k such that the first k characters are in the first group, the next k characters are in the second group, and so on. **Note** that the size of the last group can be smaller than k.
2. **Replace** each group of s with a string representing the sum of all its digits. For example, "346" is replaced with "13" because 3 + 4 + 6 = 13.
3. **Merge** consecutive groups together to form a new string. If the length of the string is greater than k, repeat from step 1.

Return s *after all rounds have been completed*.

**Example 1:**

**Input:** s = "11111222223", k = 3

**Output:** "135"

**Explanation:**

- For the first round, we divide s into groups of size 3: "111", "112", "222", and "23".

​​​​​Then we calculate the digit sum of each group: 1 + 1 + 1 = 3, 1 + 1 + 2 = 4, 2 + 2 + 2 = 6, and 2 + 3 = 5.

  So, s becomes "3" + "4" + "6" + "5" = "3465" after the first round.

- For the second round, we divide s into "346" and "5".

  Then we calculate the digit sum of each group: 3 + 4 + 6 = 13, 5 = 5.

  So, s becomes "13" + "5" = "135" after second round.

Now, s.length <= k, so we return "135" as the answer.

**Example 2:**

**Input:** s = "00000000", k = 3

**Output:** "000"

**Explanation:**

We divide s into "000", "000", and "00".

Then we calculate the digit sum of each group: 0 + 0 + 0 = 0, 0 + 0 + 0 = 0, and 0 + 0 = 0.

s becomes "0" + "0" + "0" = "000", whose length is equal to k, so we return "000".

**Constraints:**

* 1 <= s.length <= 100
* 2 <= k <= 100
* s consists of digits only.