# 题目

给你一个由若干数字（0 - 9）组成的字符串s，和一个整数。

如果s的长度大于k，则可以执行一轮操作。在一轮操作中，需要完成以下工作：

1、将s拆分成长度为k的若干连续数字组，使得前k个字符都分在第一组，接下来的k个字符都分在第二组，依此类推。注意，最后一个数字组的长度可以小于k。

2、用表示每个数字组中所有数字之和的字符串来 替换 对应的数字组。例如，"346"会替换为"13"，因为3 + 4 + 6 = 13。

3、合并所有组以形成一个新字符串。如果新字符串的长度大于 k 则重复第一步。

返回在完成所有轮操作后的s。

示例 1：

输入：s = "11111222223", k = 3

输出："135"

解释：

- 第一轮，将s分成："111"、"112"、"222"和"23"。

接着，计算每一组的数字和：1 + 1 + 1 = 3、1 + 1 + 2 = 4、2 + 2 + 2 = 6和2 + 3 = 5。

这样，s在第一轮之后变成"3" + "4" + "6" + "5" = "3465"。

- 第二轮，将s分成："346"和"5"。

接着，计算每一组的数字和：3 + 4 + 6 = 13、5 = 5。

这样，s在第二轮之后变成"13" + "5" = "135"。

现在，s.length <= k，所以返回"135"作为答案。

示例 2：

输入：s = "00000000", k = 3

输出："000"

解释：

将"000", "000", and "00".

接着，计算每一组的数字和：0 + 0 + 0 = 0、0 + 0 + 0 = 0和0 + 0 = 0。

s变为"0" + "0" + "0" = "000"，其长度等于k，所以返回"000"。

提示：

1 <= s.length <= 100

2 <= k <= 100

s 仅由数字（0 - 9）组成。

# 分析

## 方法一：模拟

思路：

代码：

class Solution {

public:

string digitSum(string s, int k) {

while (s.size() > k) {

string tmp; // 每次操作结束的字符串

int n = s.size();

for (int i = 0; i < n; i += k) {

int val = 0;

for (int j = i; j < min(i + k, n); ++j) {

val += s[j] - '0';

}

tmp.append(to\_string(val));

}

s = tmp;

}

return s;

}

};

或：

class Solution {

public:

string digitSum(string s, int k) {

while (s.length() > k) {

string new\_s = "";

int i = 0;

while (i < s.length()) {

int sum = 0;

for (int j = 0; j < k && i < s.length(); ++j) {

sum += s[i++] - '0';

}

new\_s += to\_string(sum);

}

s = new\_s;

}

return s;

}

};