# 题目

给你一个由小写字母组成的字符串 s ，以及一个整数 k 。

首先，用字母在字母表中的位置替换该字母，将 s 转化 为一个整数（也就是，'a' 用 1 替换，'b' 用 2 替换，... 'z' 用 26 替换）。接着，将整数 转换 为其 各位数字之和 。共重复 转换 操作 k 次 。

例如，如果 s = "zbax" 且 k = 2 ，那么执行下述步骤后得到的结果是整数 8 ：

转化："zbax" ➝ "(26)(2)(1)(24)" ➝ "262124" ➝ 262124

转换 #1：262124 ➝ 2 + 6 + 2 + 1 + 2 + 4 ➝ 17

转换 #2：17 ➝ 1 + 7 ➝ 8

返回执行上述操作后得到的结果整数。

示例 1：

输入：s = "iiii", k = 1

输出：36

解释：操作如下：

- 转化："iiii" ➝ "(9)(9)(9)(9)" ➝ "9999" ➝ 9999

- 转换 #1：9999 ➝ 9 + 9 + 9 + 9 ➝ 36

因此，结果整数为 36 。

示例 2：

输入：s = "leetcode", k = 2

输出：6

解释：操作如下：

- 转化："leetcode" ➝ "(12)(5)(5)(20)(3)(15)(4)(5)" ➝ "12552031545" ➝ 12552031545

- 转换 #1：12552031545 ➝ 1 + 2 + 5 + 5 + 2 + 0 + 3 + 1 + 5 + 4 + 5 ➝ 33

- 转换 #2：33 ➝ 3 + 3 ➝ 6

因此，结果整数为 6 。

提示：

1 <= s.length <= 100

1 <= k <= 10

s 由小写英文字母组成

# 分析

## 方法一：模拟

思路：

在第一次操作前，我们根据给定的字符串s，得到对应的数字串digits。

随后我们进行k次操作，每次操作将digits的每个数位进行累加，得到新的数字串。

注意到当digits的长度为1时，后续的操作都不会改变结果。此时可以提前退出循环并返回答案。

代码：

class Solution {

public:

int getLucky(string s, int k) {

string digits;

for (char ch: s) {

digits += to\_string(ch - 'a' + 1);

}

for (int i = 1; i <= k && digits.size() > 1; ++i) {

int sum = 0;

for (char ch: digits) {

sum += ch - '0';

}

digits = to\_string(sum);

}

return stoi(digits);

}

};