# 题目

一个整数num的k美丽值定义为num中符合以下条件的子字符串数目：

子字符串长度为k。

子字符串能整除num。

给你整数num和k，请你返回num的k美丽值。

注意：

允许有前缀0。

0不能整除任何值。

一个子字符串是一个字符串里的连续一段字符序列。

示例 1：

输入：num = 240, k = 2

输出：2

解释：以下是 num 里长度为 k 的子字符串：

- "240" 中的 "24" ：24 能整除 240 。

- "240" 中的 "40" ：40 能整除 240 。

所以，k 美丽值为 2 。

示例 2：

输入：num = 430043, k = 2

输出：2

解释：以下是 num 里长度为 k 的子字符串：

- "430043" 中的 "43" ：43 能整除 430043 。

- "430043" 中的 "30" ：30 不能整除 430043 。

- "430043" 中的 "00" ：0 不能整除 430043 。

- "430043" 中的 "04" ：4 不能整除 430043 。

- "430043" 中的 "43" ：43 能整除 430043 。

所以，k美丽值为2。

提示：

1 <= num <= 109

1 <= k <= num.length（将num视为字符串）

# 分析

## 方法一：枚举

思路；

为了方便起见，我们用s表示num对应十进制表示的字符串。我们可以从左至右枚举s中长度为k的字符串，并判断该子串对应的整数能否被num整除。与此同时，我们用res统计能被num整除的子串数量，如果某个子串能被num整除，则我们将res加上1。最终，res即为num的k美丽值，我们返回res作为答案。

代码：

class Solution {

public:

int divisorSubstrings(int num, int k) {

string s = to\_string(num); // num 十进制表示字符串

int n = s.size();

int res = 0;

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

// 枚举所有长度为 k 的子串

int tmp = stoi(s.substr(i, k));

if (tmp && num % tmp == 0) {

++res;

}

}

return res;

}

};

复杂度分析：

时间复杂度：O(nk)，其中n为num的位数, k为子串的长度。我们总共需要枚举O(n)个子串，其中判断每个子串都需要O(k)的时间复杂度。

空间复杂度：O(n)，即为辅助字符串的空间开销。