

FENWITER: Fenwick Iterations

题目描述

大厨刚刚学会了一个新的数据结构：**树状数组**。这个数据结构可以维护序列的信息，并支持下面两种操作：

- 修改序列中第 i 个元素的值；
- 计算序列的某个前缀中元素之和。

对于长度为 N 的序列，两种操作的复杂度都是 $O(\log N)$ 。这个数据结构的内存占用是出了名的少。具体来说，它所需要的存储空间和序列本身是相同的。

对于给定的序列 A ，我们将这一数据结构构造在另一个数组 T 中。 T_i 的值为 $A_{start}, A_{start+1}, \dots, A_i$ 之和。下标 $start$ 是按照公式 $start = F_{down}(i) = (i \& (i + 1))$ 计算的，式中 $\&$ 代表**按位与**操作。

那么，为了求出前缀 A_1, A_2, \dots, A_L 之和，我们从下标 L 开始，然后计算 $T_L + T_{F_{down}(L)-1} + T_{F_{down}(F_{down}(L)-1)-1} + \dots + T_{F_{down}(F_{down}(\dots(F_{down}(L)-1)\dots)-1)-1}$ 之和。通常我们使用一个循环来计算，每次用函数 F_{down} 来更新 L 的值，直到 L 变为 0 为止。大厨想要验证这一算法的复杂度确实为 $O(\log N)$ 。为此他想要知道，为了计算这一前缀和，一共需要访问几次数组 T 。请你帮他求解这一问题。

由于大厨需要处理的问题规模特别大， L 的值也非常大，因此他会将 L 的二进制表示提供给你。具体地， L 表示为 L_1 串接重复 N 遍的 L_2 ，再串接 L_3 ，其中 L_1 、 L_2 和 L_3 都是二进制串。

输入格式

输入的第一行包含一个整数 T ，代表测试数据的组数。接下来是 T 组数据。

每组数据仅有一行，包含三个非空字符串 L_1 、 L_2 和 L_3 ，以及一个整数 N 。字符串均只包含 0 和 1 两种字符。为获得 L 的字符串表示，你需要将 L_1 串接重复 N 遍的 L_2 ，再串接 L_3 。保证得到的下标为正数。

输出格式

对于每组数据，输出一行，包含一个整数，代表为求出前缀和所需访问 T 数组的次数。

数据范围与子任务

- $1 \leq T \leq 300$
- $1 \leq |L_i| \leq 1000$
- $1 \leq N \leq 10^6$

子任务 1 (20 分)：

- $|L_1| + |L_2| \times N + |L_3| \leq 60$

子任务 2 (30 分)：

- $1 \leq T \leq 30$
- $1 \leq N \leq 100$

子任务 3 (50 分)：

- 无附加限制

样例数据

输入	输出
4	6
001 100 011 4	12
1000 1101 100 3	8
1010 001 101 4	10
010 101 000 4	

时间限制

1 秒