Задача A. Дуэль

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Двое дуэлянтов решили выбрать в качестве места проведения поединка тёмную аллею. Вдоль этой аллеи растёт n деревьев и кустов. Расстояние между соседними объектами равно одному метру. Дуэль решили проводить по следующим правилам. Некоторое дерево выбирается в качестве стартовой точки. Затем два дерева, находящихся на одинаковом расстоянии от исходного, отмечаются как места для стрельбы. Дуэлянты начинают движение от стартовой точки в противоположных направлениях. Когда соперники достигают отмеченных деревьев, они разворачиваются и начинают стрелять друг в друга.

Дана схема расположения деревьев вдоль аллеи. Требуется определить количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

Формат входных данных

Задана одна строка, состоящая из символов «`0`» и «`1`» — схема аллеи. Деревья обозначаются символом «`1`», а кусты — символом «`0`». Длина строки не превосходит $100\,000$ символов.

Формат выходных данных

Выведите количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

Примеры

стандартный ввод	***стандартный вывод***
`101010101`	`4`
`101001`	`0`

Пояснения к примерам

В первом примере возможны следующие конфигурации дуэли (стартовое дерево и деревья для стрельбы выделены жирным шрифтом): **1**0**1**0**1**0101, 10**1**0**1**0**1**01, 1010**1**0**1**0**1** и **1**010**1**010**1**.

Задача B. Уравнение

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	10 секунд
Ограничение по памяти:	512 мебибайт

Дано уравнение вида $X^N + Y^N \equiv Z^N \pmod{M}$.

Требуется для фиксированных N и M найти количество различных решений этого уравнения. Решением назовём такую тройку натуральных чисел (X, Y, Z), что выполняется:

- $1 \leqslant X \leqslant Y < M$
- $1 \leqslant Z < M$
- $X^N + Y^N \equiv Z^N \pmod{M}$

Формат входных данных

В единственной строке записаны числа N и M ($1 \leqslant N \leqslant 7^7$, $1 \leqslant M \leqslant 7^7$).

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

стандартный ввод	**стандартный вывод**
1 3	2
2 4	5
3 5	8

Задача C. Преобразование Фурье

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Петя недавно нашёл в книжке определение *частичного преобразования Фурье* набора комплексных чисел. Теперь он хочет, чтобы вы написали ему программу для вычисления частичных преобразований Фурье, для того, чтобы получше изучить их свойства.

Вот определение частичного дискретного преобразования Фурье, найденное Петей в книжке:

> Пусть $n = 2^s$, где $s \geqslant 0$ — целое число. Определим сначала инволюцию rev_s на множестве целых чисел от 0 до $2^s - 1$ следующим образом: $\mathrm{rev}_0(0) = 0$, $\mathrm{rev}_{s+1}(2x) = \mathrm{rev}_s(x)$, $\mathrm{rev}_{s+1}(2x+1) = 2^s + \mathrm{rev}_s(x)$ при $0 \leqslant x < 2^s$.
>
> Зафиксируем теперь первообразный корень из единицы n-й степени $\zeta = e^{2\pi i/n} = \cos\frac{2\pi}{n} + i\sin\frac{2\pi}{n}$, после чего определим для любого t от 0 до s *t-е частичное преобразование Фурье* $\mathrm{Four}_s^{(t)}(\mathbf{a})$ набора $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ из n комплексных чисел следующим образом: $\mathrm{Four}_s^{(t)}(\mathbf{a}) = \mathbf{b}^{(t)} = (b_0^{(t)}, b_1^{(t)}, \ldots, b_{n-1}^{(t)})$, где
>
> $$b^{(t)}_{2^{s-t}j+k} = \sum_{j'=0}^{2^t-1} \zeta^{2^{s-t}jj'} a_{2^{s-t}j'+\mathrm{rev}_{s-t}(k)}$$
>
> $$\text{при } 0 \leqslant j < 2^t \text{ и } 0 \leqslant k < 2^{s-t}.$$
>
> Таким образом, $\mathrm{Four}_s^{(s)} = \mathrm{Four}_s$ — обычное преобразование Фурье:
>
> $$b_j^{(s)} = \sum_{j'=0}^{n-1} \zeta^{jj'} a_{j'},$$
>
> а $\mathrm{Four}_s^{(0)}$ всего лишь переставляет числа в исходном наборе: $b_k^{(0)} = a_{\mathrm{rev}_s(k)}$.

У Пети есть подозрение, что найти $\mathrm{Four}_s^{(t)}$ очень просто, если уже известно $\mathrm{Four}_s^{(t-1)}$, однако сам он вывести нужную формулу пока не сумел.

Ваша задача состоит в том, чтобы вычислить t-е частичное преобразование Фурье данного набора из 2^s комплексных чисел.

Формат входных данных

Первая строка содержит числа s и t, разделённые пробелом ($0 \leqslant t \leqslant s \leqslant 16$). Последующие 2^s строк содержат по два вещественных числа каждая — вещественную и мнимую часть соответствующего комплексного числа a_j.

Формат выходных данных

В каждой из 2^s строк следует вывести два вещественных числа с шестью точными знаками после десятичной точки — вещественную и мнимую часть числа $b_j^{(t)}$.

Пример

стандартный ввод	стандартный вывод
2 1 2.00 0.00 7.00 0.00 3.00 0.00 -3.00 0.00	5.00000000 0.00000000 4.00000000 0.00000000 -1.00000000 0.00000000 10.00000000 0.00000000

Задача D. Произведение

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мебибайт

Требуется найти произведение двух целых чисел.

Формат входных данных

В каждой из двух строк входных данных записано целое число, состоящее не более чем из 239 000 цифр.

Формат выходных данных

Выведите одно целое число — произведение этих чисел.

Примеры

стандартный ввод	***стандартный вывод***
`2` `2`	`4`
`-1` `1`	`-1`

Задача E. Раздвоение

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Обозначим две последовательности действительных чисел $x(k)$ и $y(k)$. Определим последовательность комплексных чисел $z(k)$: $z(k) = x(k) + iy(k)$.

Пусть $\mathrm{FFT}_N(k, z) = \sum\limits_{n=0}^{N-1} z_n e^{2\pi ikn/N}$. Аналогичным образом определяются $\mathrm{FFT}_N(k, x)$ и $\mathrm{FFT}_N(k, y)$.

Требуется по вычисленным значениям $\mathrm{FFT}_N(k, z)$ восстановить значения $\mathrm{FFT}_N(k, x)$ и $\mathrm{FFT}_N(k, y)$.

Формат входных данных

В первой строке записано целое число N ($1 \leqslant N \leqslant 2^{30}$, N является степенью двойки). Далее следуют целые неотрицательные числа A, B, C, D, E, F, не превосходящие 1000. Для экономии времени ввода значения $\mathrm{FFT}_N(k, z)$ нужно будет вычислять по следующим формулам:

$\mathrm{FFT}_N(k, z).\mathrm{real} = ((A + B \cdot k) \text{ xor } (C \cdot k)) \cdot 10^{-3}$,

$\mathrm{FFT}_N(k, z).\mathrm{imag} = ((D + E \cdot k) \text{ xor } (F \cdot k)) \cdot 10^{-3}$,

где $\mathrm{FFT}_N(k, z).\mathrm{real}$ и $\mathrm{FFT}_N(k, z).\mathrm{imag}$ — действительная и мнимая части соответственно.

Затем дано число M — количество запросов ($1 \leqslant M \leqslant 10^5$). Далее следуют M целых чисел q_j ($0 \leqslant q_j < N$).

Формат выходных данных

Выведите M строк. В j-й строке — значения $\mathrm{FFT}_N(q_j, x)$ и $\mathrm{FFT}_N(q_j, y)$. Значения должны отличаться от правильных не более чем на 10^{-4}.

Примеры

стандартный ввод	**стандартный вывод**
2 1000 0 0 0 0 0 2 0 1	1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0
4 0 100 300 500 100 200 4 0 1 2 3	0.000 0.000 0.500 0.000 0.504 0.140 0.516 0.176 0.656 0.000 0.812 0.000 0.504 -0.140 0.516 -0.176
1048576 999 998 997 996 995 994 3 17 239239 2011	540.737 -1587.741 1589.778 539.689 2404.809 531.421 1359.578 1569.751 3678.277 -523.243 526.382 3664.887

Задача F. ДНК роботов

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мебибайт

Новые технологии построения ДНК позволили провести большой эксперимент по построению биороботов. Экспериментом занимались учёные из НИИ Данных Строк (НИИДС).

ДНК построенных роботов записываются в виде строк из $M = 2^n$ символов, где $n \geqslant 2$ — так проще всего было их моделировать. Кроме того, по техническим причинам ДНК робота — это не обычная строка, а циклическая, то есть её можно начинать читать с любого места.

Через некоторое время после начала эксперимента учёные обнаружили несколько типов роботов, и заинтересовались, какие мутации к этому привели. Чтобы восстановить дерево мутаций, нужно научиться решать следующую задачу: считать *коэффициент сходства* для ДНК двух роботов. Коэффициент сходства — это количество совпадающих букв при наилучшем совмещении двух заданных ДНК: чем больше совпадений, тем больше сходство.

Напишите программу, которая будет вычислять коэффициент сходства двух заданных ДНК роботов, а также находить их наилучшее совмещение.

Формат входных данных

Первая строка содержит целое число M ($4 \leqslant M \leqslant 131\,072$, гарантируется, что M — степень двойки). В следующих двух строках записаны ДНК двух роботов. Каждая из них содержит ровно M символов и состоит только из «A», «C», «G» и «T».

Формат выходных данных

Выведите два числа: коэффициент сходства и оптимальный сдвиг. Коэффициент — это количество совпадающих букв при совмещении двух ДНК после сдвига. Сдвиг же — количество букв в конце второй строки (число от 0 до M включительно), которые нужно перенести в её начало. При совмещении первой строки со сдвинутой второй количество совпадающих букв должно быть максимально возможным. Если оптимальных ответов несколько, выведите любой из них.

Пример

стандартный ввод	*стандартный вывод*
16 ACGTACGTACGTACGT CGTACGTACGTACGTC	15 1