Задача A. Адские круги

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.5 секунды
Ограничение по памяти:	512 мебибайт

I'm on the highway to hell...

AC/DC

Когда-то давно у Дьявола в Аду был $k+1$ круг. На каждом из них томились грешники, причём некоторые в очень удобных котлах. Не так давно Невероятно Важный Совет Демонов решил, что передвигаться между кругами очень неудобно, поэтому была построена магистраль в виде прямой, проходящая через все круги (которые, на самом деле, являются окружностями, но демоны не сильно разбираются в геометрии).

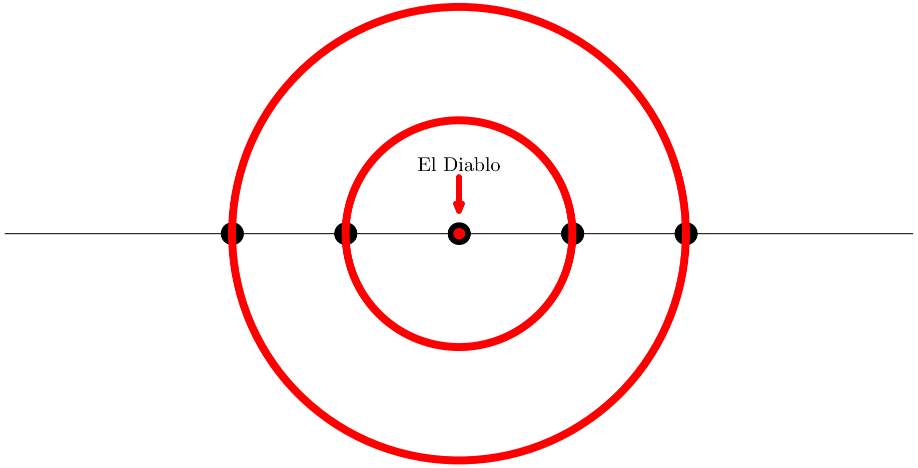
На открытие магистрали собрались все грешники (в Аду не часто происходит что-то новое). Они сразу же разбрелись по ней и стали изучать, как устроена магистраль. Один из дозорных ангелов сразу же заметил, что в аду как-то пусто, и немедленно доложил об этом Богу. Он не растерялся и тут же отдал приказ уничтожить пустые круги Ада, который тотчас же был исполнен.

Дьяволу это крайне не понравилось, он решил встать в какое-то место на магистрали и создать новые круги Ада, с блекджеком и демонессами. Для оптимизации работ все грешники должны сразу оказаться на каких-то кругах, чтобы их потом не пришлось гонять туда-сюда. Конечно, создание каждого круга требует много сил, поэтому Дьявол хочет создать минимальное число кругов (мы всё ещё помним, что они являются окружностями), причём центры кругов будут ровно в той точке, куда встанет Дьявол (он хочет всегда находиться в центре внимания). Вам, как главному программисту Ада, поручено найти место, куда нужно встать Дьяволу.

Формат входных данных

Первая строка содержит n — число грешников ($1 \leqslant n \leqslant 200\,000$).

На следующей строке заданы n целых чисел — координаты грешников на магистрали, по модулю не превосходящие $60\,000$.



Формат выходных данных

В первой строке выведите вещественную координату — куда надо встать Дьяволу. Во второй строке выведите минимальное количество кругов, необходимое для возвращения всех грешников на круги своя. В третьей строке выведите разделённые пробелами вещественные радиусы кругов. Заметим, что круги могут иметь нулевой радиус.

Если правильных ответов несколько, вы можете вывести любой из них. Для всех точек минимальное расстояние до найденных окружностей не должно превышать 10^{-6}.

Примеры

стандартный ввод	стандартный вывод
4 1 1 2 2	1.5 1 0.5
5 1 2 3 4 5	3.0 3 0.0 1.0 2.0

Задача B. Дуэль

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.5 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 C. Уравнение

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	2 секунды
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 D. Преобразование Фурье

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.5 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
>
> $$b^{(t)}_{2^{s-t}j+k} = \sum_{j'=0}^{2^t-1} \zeta^{2^{s-t}jj'} a_{2^{s-t}j'+\mathrm{rev}_{s-t}(k)}$$
>
> при $0 \leqslant j < 2^t$ 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. Заданные числа не превосходят 100 по абсолютной величине и содержат не более 25 знаков в десятичной записи.

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 E. Произведение

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.5 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 F. Ук

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.5 секунды
Ограничение по памяти:	512 мебибайт

У библиотекаря есть талон, на котором напечатана строка, состоящая из букв «o» и «k». А также выданный во фруктовой лавке шаблон. На шаблоне напечатана строка, состоящая из символов «o», «k» и «?».

Библиотекарь может выполнять следующие операции любое количество раз в любом порядке:

1. Приложить шаблон к талону и вырезать соответствующий кусок талона: количество букв в куске должно быть равно количеству символов в шаблоне. Части талона, оставшиеся после вырезания, можно использовать для дальнейшего вырезания, но только по отдельности (части не склеиваются). Одна или даже обе оставшиеся части могут не содержать больше букв (в таком случае приложить к ним шаблон, конечно, не удастся).
2. Обменять в лавке отрезанный первой операцией кусок талона на бананы.

При обмене куска талона на бананы продавец в лавке действует следующим образом. Сначала он собирает в кучу бананы. Изначально куча пуста. Продавец просматривает кусок талона слева направо, и за каждую напечатанную на нём букву «o» добавляет в кучу o бананов, а за каждую букву «k» — k бананов.

Потом продавец посимвольно слева направо сравнивает строки, напечатанные на куске талона и шаблоне. При сравнении символ «?» из шаблона считается равным любой букве. Если продавец находит несовпадение в какой-то позиции, то он начинает очень сердиться, и от этого у него усиливается чувство голода. В результате он делит кучу на две части так, чтобы разница в количестве бананов в двух новых кучах была не больше одного, а потом съедает все бананы из той части, которая не меньше другой. Утолив голод, продавец продолжает процесс сравнения, пока либо он не сравнит последнюю пару символов, либо в куче не закончатся бананы.

Все бананы, которые остались в куче после сравнения, выдаются библиотекарю в обмен на кусок талона.

Найдите максимальное число бананов, которое может получить библиотекарь.

Формат входных данных

В первой строке заданы два целых числа o и k ($0 \leqslant o, k \leqslant 5000$). Во второй строке задана строка S из букв «o» и «k», напечатанная на талоне. В третьей строке задана строка P из символов «o», «k» и «?», напечатанная на шаблоне. Гарантируется, что $1 \leqslant |P| \leqslant |S| \leqslant 250\,000$.

Формат выходных данных

Выведите максимальное число бананов, которое может получить библиотекарь.

Примеры

стандартный ввод	*стандартный вывод*
2 1 ookookook koo	10
1 3 koooooook ?	13
1000 0 kookoo ook	2000
21 1 ooo kkk	7

Задача G. Произведение многочленов

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.5 секунды
Ограничение по памяти:	512 мебибайт

Заданы два многочлена от переменной `x`. Найдите их произведение.

Формат входных данных

В первой строке записан многочлен P, а во второй — многочлен Q. Степень каждого из них строго меньше $65\,536$. Коэффициенты многочленов целые и не превосходят $100\,000$ по абсолютной величине.

Далее описан формат записи многочлена.

Одночлены, из которых состоит многочлен, отделены друг от друга знаком арифметической операции «+» или «-», вокруг которого стоят одиночные пробелы. Одночлены перечислены в порядке от старших к младшим. Многочлен не может содержать двух одночленов, соответствующих одной и той же степени `x`. Степени `x` с нулевым коэффициентом опускаются.

Одночлен записывается в общем случае в виде «c`x^`p». Но, если $p = 1$, запись имеет вид «c`x`», а если $p = 0$, запись имеет вид «c». Если $|c| = 1$, а $p \neq 0$, символ «1» опускается. Коэффициент c записывается как строго положительный, кроме, возможно, коэффициента у самого старшего одночлена: у всех остальных знак учитывается в арифметической операции перед ними.

Если многочлен тождественно равен нулю, он записывается как «`0`».

Для лучшего понимания формата посмотрите на примеры.

Формат выходных данных

Выведите многочлен $P \cdot Q$ в том же формате, что и заданные многочлены. Соблюдайте формат как можно более точно! Проверка производится посимвольно.

Примеры

стандартный ввод	**стандартный вывод**
`x^2 - 2x + 1` `-5x^3 - x`	`-5x^5 + 10x^4 - 6x^3 + 2x^2 - x`
`9999` `9999`	`99980001`

Задача H. Раздвоение

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 0.5 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 I. ДНК роботов

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	0.5 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`

Задача J. Произведение корней

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Боб любит играть с многочленами. Вчера он придумал две последовательности неотрицательных целых чисел: $a_1, \ldots, a_n$ и $b_1, \ldots, b_m$, где $1 \leqslant n, m \leqslant 100\,000$.

После этого он построил многочлены

$$f(x) = \prod_{i=1}^{n}(1 + a_i x),$$

$$g(x) = \prod_{j=1}^{m}(1 + b_j x),$$

$$h(x) = \prod_{i=1}^{n}\prod_{j=1}^{m}(1 + a_i b_j x).$$

У многочлена $h(x)$ он посчитал только первые k коэффициентов $(1 \leqslant k \leqslant \min(100\,000, nm+1))$. Поскольку Боб любит не только многочлены, но и преобразование Фурье, он посчитал коэффициенты всех трёх многочленов по модулю $998\,244\,353$.

Однако сегодня Боб потерял и последовательности a_i и b_j, и многочлен $h(x)$. Остались только многочлены $f(x)$ и $g(x)$. Боб расстроился и теперь просит Алису помочь ему восстановить многочлен $h(x)$, зная только многочлены $f(x)$ и $g(x)$. У Алисы нет времени это делать, поэтому она попросила вас помочь ей.

Формат входных данных

В первой строке даны целые числа n, m и k $(1 \leqslant n, m, k \leqslant 100\,000$, $k \leqslant nm + 1)$.

Во второй строке даны целые числа $f_0, \ldots, f_n$, где $f(x) = f_0 + f_1 x + \ldots + f_n x^n$.

В третьей строке даны целые числа $g_0, \ldots, g_m$, где $g(x) = g_0 + g_1 x + \ldots + g_m x^m$.

Коэффициенты многочленов даны по модулю $998\,244\,353$.

Формат выходных данных

Выведите ровно k целых чисел $h_0, \ldots, h_{k-1}$, где $h(x) = h_0 + h_1 x + \ldots + h_{nm} x^{nm}$.

Выводить коэффициенты тоже нужно по модулю $998\,244\,353$.

Пример

стандартный ввод	*стандартный вывод*

`2 2 5`
`1 2 1`
`1 2 1`
	`1 4 6 4 1`