

Задача А. Линейные уравнения

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Система линейных уравнений, как всем известно, есть множество уравнений

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &= b_1 \\ &\dots \\ a_{n1}x_1 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

Ваша задача — решить её.

Формат входных данных

В первой строке входного файла записано целое число n ($1 \leq n \leq 20$). В следующих n строках записано по $n + 1$ целых чисел: $a_{i1}, \dots, a_{in}, b_i$. Все эти числа не превышают 100 по абсолютному значению.

Формат выходных данных

Первая строка выходного файла должна содержать одно из следующих сообщений:

- `impossible` — решений нет
- `infinity` — бесконечно много решений
- `single` — единственное решение. В этом случае вторая строка должна содержать n чисел x_1, \dots, x_n , разделенных пробелами. Решение должно быть выведено с точностью не менее трех знаков после десятичной точки.

Примеры

стандартный ввод	стандартный вывод
2 1 1 1 2 2 2	infinity
2 1 2 0 1 2 1	impossible
2 1 2 1 2 1 0	single -0.333333333 0.666666667

Задача В. Возведение матрицы в степень

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Дана матрица A размера $N \times N$ и неотрицательное целое число K . Выведите матрицу A^K .
Все вычисления делайте по модулю 998244353.

Формат входных данных

Первая строка содержит два целых числа N и K ($1 \leq N \leq 200$, $0 \leq K \leq 10^{18}$).

Следующие N строк содержат описание матрицы A . i -я строка содержит N целых чисел $a_{i,1}, a_{i,2}, \dots, a_{i,n}$ ($0 \leq a_{i,j} < 998244353$) — элементы в i -й строке матрицы.

Формат выходных данных

Выведите N строк, i -я из которых содержит N целых чисел — элементы в i -й строке матрицы A^K по модулю 998244353.

Примеры

стандартный ввод	стандартный вывод
2 7 0 1 1 1	8 13 13 21
3 0 0 0 0 0 0 0 0 0 0	1 0 0 0 1 0 0 0 1

Задача С. Автоматический ответатель

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Жюри олимпиады надоело отвечать на вопросы участников, поэтому они собрали автоматический ответатель. У ответателя есть $n + 2$ состояния. Изначально он находится в состоянии 1. Состояния $n + 1$ и $n + 2$ являются терминальными. Если ответатель оказывается в состоянии $n + 1$, он отвечает «Без комментариев», если он оказывается в состоянии $n + 2$ — то «Смотри условие». Состояния от 1 до n являются промежуточными. Если ответатель находится в состоянии i , то на следующем шаге он перейдет в состояние j с вероятностью $p_{i,j}$.

Вам дано описание ответателя. Определите, с какой вероятностью он отвечает «Без комментариев».

Формат входных данных

Первая строка содержит число n ($1 \leq n \leq 50$). Следующие n строк содержат по $n + 2$ числа — вероятности $p_{i,j}$ в процентах ($1 \leq p_{i,j} \leq 99$).

Формат выходных данных

Выведите одно число — вероятность того, что ответатель ответит «Без комментариев». Относительная или абсолютная погрешность не должна превышать 10^{-6} .

Пример

стандартный ввод	стандартный вывод
1 50 20 30	0.4

Задача D. Двоичный Гаусс

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан набор из n битовых векторов размера n и еще один вектор. Получить этот вектор как хог исходных.

Формат входных данных

В первой строке записано число n ($1 \leq n \leq 300$). В следующих n строках записаны исходные вектора, в последней строке записан вектор, который нужно получить.

Формат выходных данных

Если решений нет, выведите **No solution**. Если решений несколько, выведите **Multiple solutions**. Если решение единственное, выведите номера векторов (вектора нумеруются с 0), которые нужно сложить, чтобы получить данный вектор. Номера выведите в порядке возрастания.

Примеры

стандартный ввод	стандартный вывод
3 100 111 101 010	1 2
3 100 111 011 010	No solution
3 111 010 101 000	Multiple solutions

Задача Е. Добавление векторов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В изначально пустое множество один за другим добавляются m битовых векторов размера n . После добавления каждого скажите, можно ли его представить как хог векторов, добавленных до него.

Формат входных данных

В первой строке записаны числа n и m ($1 \leq n \leq 50$, $1 \leq m \leq 10000$). В следующих m строках записаны вектора.

Формат выходных данных

Для каждого вектора выведите 1, если его можно представить как хог предыдущих и 0 если нельзя.

Примеры

стандартный ввод	стандартный вывод
2 2 10 11	0 0
3 4 100 111 011 010	0 0 1 0
3 4 000 111 111 111	1 0 1 1

Задача F. Максимальный XOR

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан набор из n чисел. Найдите максимальное число, которое можно получить как битовый XOR подмножества данных чисел.

Формат входных данных

В первой строке записано число n ($1 \leq n \leq 1000$). В следующих n строках записаны исходные числа a_i ($0 \leq a_i \leq 2^{63} - 1$).

Формат выходных данных

Выведите максимальное число, которое можно получить.

Примеры

стандартный ввод	стандартный вывод
2 1 2	3
3 6 2 8	14
3 15 3 8	15

Задача G. Число возможных векторов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан набор из m битовых векторов размера n ($1 \leq n, m \leq 50$). Сколько различных векторов можно получить как хог этих векторов.

Формат входных данных

В первой строке записаны числа n и m ($1 \leq n, m \leq 50$). В следующих m строках записаны вектора.

Формат выходных данных

Выведите число векторов, которые можно получить

Примеры

стандартный ввод	стандартный вывод
2 2 10 11	4
3 4 100 111 011 010	8
3 4 111 111 111 000	2

Задача Н. Фибоначчи

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Последовательность чисел Фибоначчи задается следующим образом: $f_0 = 0$, $f_1 = 1$, $f_i = f_{i-1} + f_{i-2}$ для любого $i > 1$.

Ваша задача — вычислить n -е число Фибоначчи. Так как ответ может быть достаточно большим, выведите остаток от деления ответа на число 998 244 353.

Формат входных данных

В единственной строке задано целое число n ($0 \leq n \leq 10^{18}$).

Формат выходных данных

Выведите одно целое число — остаток от деления n -го числа Фибоначчи на 998 244 353.

Система оценки

Ваше решение получит некоторое количество баллов, в зависимости от оптимальности алгоритма.

Примеры

стандартный ввод	стандартный вывод
0	0
1	1
4	3

Задача I. Мульти Гаусс

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Решите несколько систем линейных уравнений $n \times n$ вида $\sum_{j=1}^n a_{ij}x_j = b_i$ с одинаковыми a_{ij} , но разными b_i . Гарантируется, что решения существует и единственны.

Формат входных данных

В первой строке записаны числа n и m ($1 \leq n, m \leq 300$). В следующих n строках записаны по n чисел — коэффициенты при переменных a_{ij} , в следующих m строках записаны по n чисел — правые части b_i для каждой системы.

Формат выходных данных

Для каждой системы выведите значения переменных с точностью не менее 10^{-4} .

Пример

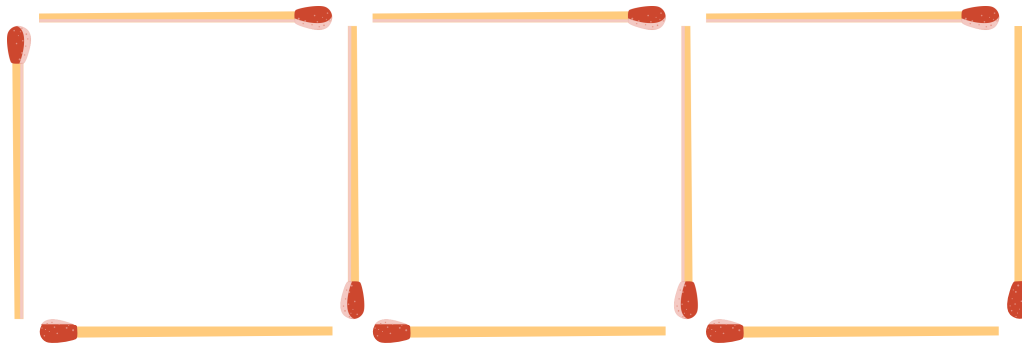
стандартный ввод	стандартный вывод
2 3	5.0 3.0
2.0 3.0	2.0 1.0
4.0 1.0	0.0 -0.0
19.0 23.0	
7.0 9.0	
0.0 0.0	

Задача J. Хорошие раскраски — 2024

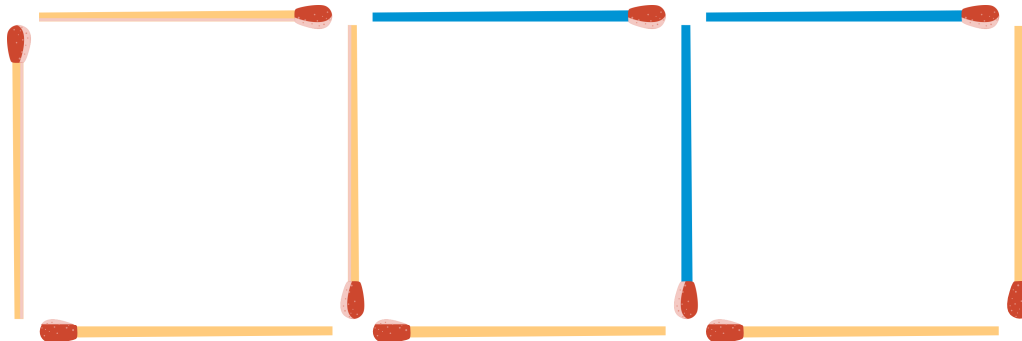
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Саша очень любит комбинаторные задачи, а также он является главным фанатом задачи «Хорошие раскраски», которая появилась на региональном этапе ВсОШ по информатике несколько лет назад. Более того, Саша очень любит играть со спичками. Поэтому мальчик решил придумать свою задачу о хороших раскрасках и порадовать вас своим творением.

Саша выложил в ряд на столе n квадратов, состоящих из спичек. На рисунке ниже приведен пример получившейся фигуры для $n = 3$. Можно заметить, что фигура, состоящая из n квадратов, содержит в себе $3n + 1$ спичек.



Назовем две спички *соседними*, если они касаются в углу некоторого квадрата. Например, на рисунке ниже спички, окрашенные синим цветом, являются попарно соседними.



У Саши в распоряжении есть краски k различных цветов, и, разумеется, он хочет окрасить каждую спичку выложенной фигуры в некоторый цвет. Саша считает раскраску спичек *хорошей*, если никакие две соседние спички не окрашены в один цвет.

Будучи любителем комбинаторных задач, Саша задумался, сколько существует различных хороших раскрасок спичек. Помогите Саше решить данную тривиальную задачу. Так как ответ может быть достаточно большим, вам необходимо вычислить остаток от деления количества хороших раскрасок на число 998 244 353.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 10^{18}$) — количество квадратов из спичек, которые выложил на столе Саша.

Вторая строка содержит одно целое число k ($3 \leq k < 998\,244\,353$) — количество различных цветов, которые может использовать Саша.

Обратите внимание, что входные данные в этой задаче могут превышать возможное значение 32-битной целочисленной переменной, поэтому необходимо использовать 64-битные целочисленные типы данных (тип `int64` в языке Pascal, тип `long long` в C++, тип `long` в Java и C#).

Формат выходных данных

Выведите одно целое число — остаток от деления количества различных хороших раскрасок на число 998 244 353.

Примеры

стандартный ввод	стандартный вывод
1 5	260
3 5	223380
10000000000000000000 900000000	591253139

Замечание

Баллы за подзадачи 1, 2, 4 – 8 начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Баллы за каждый тест в подзадаче 3 начисляются независимо, если все тесты необходимых подзадач успешно пройдены.

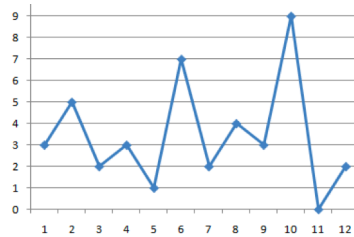
Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
0	0	Тесты из условия		полная
1	5	$n \leq 4, k = 3$		первая ошибка
2	10	$n \leq 20, k = 3$	1	первая ошибка
3	10	$n \leq 30, k = 3$	1, 2	полная
4	5	$n = 1$		первая ошибка
5	15	$n \leq 100, k \leq 10$	1, 2, 3	первая ошибка
6	20	$n \leq 10^6$	1 – 5	первая ошибка
7	15	$n \leq 10^9$	1 – 6	первая ошибка
8	20	нет	1 – 7	первая ошибка

Задача К. Разрушенные дороги

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

После каждого сражения Мстителей с кем бы то ни было город, в котором проходило сражение, превращается в руины. Рушатся здания, уничтожаются памятники, покрываются ямами от взрывов дороги. Именно о дорогах и пойдет речь.

Представим дорогу в качестве прямой и измерим глубину ям через каждый метр. Для удобства нарисуем диаграмму глубин ям, где по горизонтальной оси отметим координату ямы, а по вертикальной оси — ее глубину. Ниже вы можете увидеть пример такой диаграммы.



По данной диаграмме видно, что для каждой точки, кроме первой и последней, глубина ямы либо строго меньше, либо строго больше, чем глубины соседних слева и справа ям. Например, яма в точке с координатой 6 имеет глубину 7, а соседние ямы с координатами 5 и 7 имеют глубины 1 и 2. Дороги, обладающие таким свойством, назовем разрушенными.

Невероятно большой практической значимостью обладает задача подсчета количества разрушенных дорог по модулю 998 244 353 — именно это от вас и требуется.

По заданной длине дороги N (это число совпадает с числом точек на диаграмме, соответствующей дороге) посчитайте количество разрушенных дорог длины N по модулю 998 244 353. Учтите, что глубина ямы — неотрицательное целое число, не превышающее 10. В каком-то месте дороги ямы может не быть, тогда ее глубина равна 0.

Формат входных данных

На вход подается одно число N ($3 \leq N \leq 10^{18}$) — длина дороги.

Формат выходных данных

Программа должна вывести одно число — количество разрушенных дорог длины N по модулю 998 244 353. Не забудьте, что глубины ям являются целыми числами и лежат в промежутке от 0 до 10 включительно.

Пример

стандартный ввод	стандартный вывод
3	770

Задача L. Запутывание следов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Поиски книг о космическом анализе привлекли много внимания, и двое патрульных попытались задержать Рика и Мирлина Теренса. К счастью, на помощь к ним пришла Валлона, и, оглушив нейрохлыстом патрульных, наши герои сбежали обратно в Нижний город. Незамедлительно за ними был выслан целый отряд патрульных, но в Нижнем городе нашелся человек, который был готов спрятать беглецов от патрульных. Транторианский агент Матт Хоров укрыл Рика и Валлону в своей пекарне, а Теренс тем временем отправился домой — отсутствие резидента в течение дня могли заметить, а привлекать лишнее внимание Теренс не хотел.

На следующее утро Матт решил отправить Рика и Валлону с Флорины, где им угрожала опасность, на какую-нибудь другую планету. Хоров выдал им поддельные паспорта и билеты на корабль до Вотекса, но оставалась нерешенная проблема — огромное количество патрульных в Верхнем городе. Но Матт Хоров мог использовать других агентов, чтобы доставить Рика и Валлону в космопорт.

Чтобы не привлекать излишнего внимания, Матт решил, что поднимутся они не у самого космопорта, а в каком-либо из 8 кварталов Верхнего города, затем проедут по некоторым из этих кварталов, вероятно, не по одному разу, а затем из последнего посещенного квартала отправятся прямо в космопорт. Сопровождать Рика и Валлону будут некоторые из n агентов Хорова. Для каждого агента известно, из каких кварталов в какие он может перемещаться. Путешествие проходит следующим образом: Рик с Валлоной поднимаются в одном из кварталов, где их встречает первый агент. За один доступный ему переезд он добирается до некоторого следующего квартала, где его ждет второй агент, который везет их до следующего квартала, и так далее до последнего квартала перед космопортом.

Чтобы сбежать от возможной погони, Матт хочет рассмотреть разные маршруты. Он уже придумал m планов, каждый состоит из стартового квартала s_i , конечного квартала t_i , а также чисел l_i и r_i — Рика и Валлону будут сопровождать агенты с l_i -го по r_i -го включительно, в порядке следования их номеров. Таким образом, в квартале s_i их встречает l_i агент и перемещается с ними в некоторый следующий квартал, где их поджидает l_{i+1} агент, который помогает им переместиться в другой квартал, и так далее. Наконец r_i агент должен доставить Рика и Валлону в квартал t_i . При этом Матт выбирал агентов обдуманно, по этому никакой из отрезков $[l_i; r_i]$ не вложен в другой. Формально, не существует таких i и j , таких что $l_i < l_j \leq r_j < r_i$.

Для каждого из планов Матт хочет узнать число различных путей, которыми данные агенты могут доставить из стартового квартала в конечный, по модулю 998244353. Пути считаются различными, если существует номер $l \leq k \leq r$, такой что после поездки с k -м агентом по этим путям Рик с Валлоной добрались бы до разных кварталов. Обратите внимание, что для запутывания следов они могут посещать один квартал несколько раз, в том числе подряд.

Формат входных данных

В первой строке содержатся целые числа n и m — число агентов и число запросов соответственно ($1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5$).

В следующих n строках заданы целые числа от 0 до $2^{64} - 1$, задающие бинарные матрицы 8×8 перемещения агентов. Матрица получается следующим образом: число записывается в двоичной системе счисления, дополняется ведущими нулями до 64 знаков, после чего его цифры слева направо записываются как элементы матрицы. Заполнение идет по строкам слева направо, начиная с верхней и заканчивая нижней. В полученной матрице элемент i -й строки j -го столбца $a_{ij} = 1$, если и только если соответствующий агент может проехать от i -го квартала до j -го.

Далее идут m строк, по четыре числа в каждой: l_i, r_i - номера первого и последнего агентов, затем s_i, t_i - номера стартового и конечного кварталов ($1 \leq l_i \leq r_i \leq n; 1 \leq s_i, t_i \leq 8$).

Формат выходных данных

Выведите m строк, в i -й строке — ответ на i -й запрос по модулю 998244353.

Пример

стандартный ввод	стандартный вывод
3 3	0
9241386504218214000	1
4692768438333080000	1
4620710844295152000	
1 2 3 4	
1 3 1 3	
3 3 1 2	

Замечание

Переведем данные числа в двоичную систему счисления, дополнив ведущими нулями при необходимости. Получим следующий результат:

$$9241386504218214000_{10} = 100000000100000000000000001000000001000000001000000001000000001_2,$$

$$4692768438333080000_{10} = 0100000100100000000100000000100000000100000000100000000110000000_2,$$

$$4620710844295152000_{10} = 01000000000100000000100000000100000000100000000100000000110000000_2.$$

Теперь составим из полученных чисел матрицы 8×8 :

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

В первом запросе необходимо попасть из третьего квартала в четвертый, используя помощь первого и второго агентов. Но первый агент не может переместиться никуда из третьего квартала. Таким образом осуществить побег не удастся.

Во втором запросе нужно попасть из первого квартала в третий, используя помощь первого, второго и третьего агентов. Возможный маршрут лишь один: $1 \rightarrow 2 \rightarrow 3$.

В третьем запросе требуется попасть из первого квартала во второй с помощью третьего агента. Это можно сделать ровно одним способом.

Задача М. Максимизировать сумму XOR

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

Будем обозначать как \oplus операцию *побитового «исключающего или»* для целых чисел. В языках программирования C++ и Java она обозначается символом « \wedge », в паскале и Python — ключевым словом «xor». Например, $9 \oplus 3 = 1001_2 \oplus 11_2 = 1010_2 = 10$.

Даны два массива A и B длины n . Обозначим как $X(A)$ для массива A результат вычисления побитового «исключающего или» от всех элементов массива: $X(A) = A_1 \oplus A_2 \oplus \dots \oplus A_n$. Аналогично, введем обозначение $X(B) = B_1 \oplus B_2 \oplus \dots \oplus B_n$.

Для каждого i от 1 до n разрешается поменять местами элементы A_i и B_i . Необходимо определить, какие из этих обменов надо сделать, чтобы максимизировать сумму $X(A) + X(B)$.

Формат входных данных

В первой строке входных данных находится число n — количество элементов ($1 \leq n \leq 10^5$). В следующей строке находится n элементов массива A ($0 \leq A_i \leq 10^{18}$). В следующей строке в таком же формате дан массив B .

Формат выходных данных

В первой строке выведите максимальную возможную сумму и число k — количество необходимых обменов. В следующей строке выведите k различных чисел от 1 до n — индексы элементов, которые надо поменять.

Пример

стандартный ввод	стандартный вывод
2	6 1
1 1	1
2 2	

Замечание

В примере после обмена массивы равны $A = [2, 1]$ и $B = [1, 2]$, соответственно.
 $X(A) = 2 \oplus 1 = 10_2 \oplus 1_2 = 11_2 = 3$, $X(B) = 3$, $X(A) + X(B) = 6$.