

Задача А. Ральф и арифметика

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Ральф — второстепенный персонаж компьютерной игры, и ему надоело находиться в тени главного героя. Ральф заметил кое-что общее между его компьютерной игрой и арифметикой.

Ральф считает, что в арифметике некоторые цифры встречаются чаще других, делая все остальные цифры второстепенными. Чтобы проверить свою гипотезу, Ральф выписал все второстепенные цифры и теперь хочет узнать количество чисел от 1 до n , которые не содержат второстепенных цифр в своей десятичной записи. Помогите ему это сделать.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^{18}$).

Вторая строка содержит целое число k — количество цифр, которые Ральф считает второстепенными ($1 \leq k \leq 9$).

В третьей строке через пробел записаны сами второстепенные цифры d_1, \dots, d_k ($0 \leq d_1 < d_2 < \dots < d_k \leq 9$).

Формат выходных данных

В единственной строке выходного файла выведите одно число — количество чисел от 1 до n , в десятичной записи которых не встречаются второстепенные цифры.

Примеры

стандартный ввод	стандартный вывод
9 2 3 4	7
1000 9 0 2 3 4 5 6 7 8 9	3
100000 8 0 1 2 5 6 7 8 9	62

Замечание

В первом тестовом примере подходят все числа от 1 до 9, кроме 3 и 4.

Во втором тестовом примере подходят только числа 1, 11 и 111.

В третьем тестовом примере подходят все числа длиной от 1 до 5, состоящие только из 3 и 4.

Задача В. Битовый автомат

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Ральф нашёл новое оружие — битовый автомат! Этот автомат стреляет очередями бит, причем первый выстреленный бит наносит цели урон 1, а каждый следующий бит наносит цели урон в два раза больший, чем предыдущий.

Этот автомат пришёлся как нельзя кстати, когда на Ральфа внезапно обрушилось полчище кибер-жуков. Стреляя в жука, Ральф замечает, какие биты попали в цель, а какие нет. К несчастью, за годы тяжелой работы наш герой изрядно испортил себе зрение, и поэтому может распознать траекторию движения одного из бит неверно. В связи с этим реальный и замеченный Ральфом урон могут различаться. Зная количество выстреленных бит и предполагаемый урон помогите Ральфу узнать, какой минимальный и максимальный реальный урон мог быть нанесен кибер-жуку.

Формат входных данных

В первой строке находится два числа n и a — количество бит в коде и предполагаемый урон соответственно ($1 \leq n \leq 63$, $0 \leq a \leq 2^n - 1$).

Формат выходных данных

Выведите два числа: минимальный и максимальный возможный реальный урон, если учесть, что Ральф мог допустить ошибку в распознавании результата выстрела в не более чем одном бите. $0 \leq a_{min} \leq a_{max} \leq 2^n - 1$.

Примеры

стандартный ввод	стандартный вывод
3 5	1 7
5 0	0 16

Задача С. Монетки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Когда в автомат с его игрой никто не играет и Ральфу становится скучно, он выходит прогуляться и пособрать монетки. За все время он собрал их уже целых n^2 штук. Несмотря на свой внешний вид, он также любит аккуратность, поэтому уложил их все в квадрат $n \times n$, по одной монетке в ячейку, так, что свободного места в квадрате не осталось.

Однако, неожиданно к Ральфу в гости пришел Феликс и принес еще одну монетку. Наш герой был безумно рад такому вниманию и сюрпризу, но абсолютно не имел понятия, куда ее теперь положить. Поэтому он решил поменять место для хранения монеток и положить все $n^2 + 1$ монетку в другой прямоугольник. Однако, не все так просто, ведь Ральф не только аккуратен, но и придирчив. А именно, он хочет, чтобы для нового прямоугольника $x \times y$ — места хранения его монеток — выполнялись следующие условия:

- Прямоугольник вмещает в себя все монетки и не содержит пустых мест, то есть $x \cdot y = n^2 + 1$;
- Периметр прямоугольника максимально возможный;
- Каждая сторона прямоугольника должна иметь длину хотя бы 2.

По данному n Ральф хочет найти заветные числа x и y , и как можно быстрее — изготовление прямоугольника нужно начинать уже сейчас. Помогите ему!

Формат входных данных

В первой строке содержится число q — количество тестов ($1 \leq q \leq 10^6$).

В i -й из следующих q строк содержится число n_i — размер изначального прямоугольника с монетками ($1 \leq n_i \leq 10^6$).

Формат выходных данных

Выведите q строк, в i -й из которой должны находиться два числа x_i и y_i — размеры нового прямоугольника $x_i \times y_i$ или -1 , если прямоугольника, удовлетворяющего условиям задачи, не существует.

Пример

стандартный ввод	стандартный вывод
6	-1
1	-1
2	2 5
3	-1
4	2 13
5	5 65
18	

Замечание

В тестовом примере числа $1^2 + 1 = 2$, $2^2 + 1 = 5$ и $4^2 + 1 = 17$ — простые, и такое количество монеток нельзя уложить в прямоугольник, удовлетворяющий условиям задачи.

$3^2 + 1 = 10$ и $5^2 + 1 = 26$ монеток уложить в прямоугольник единственным способом, а $18^2 + 1 = 325$ монеток можно уложить двумя способами:

- $5 \cdot 65$, периметр 70;
- $13 \cdot 25$, периметр 38.

В первом случае периметр больше, поэтому это и будет ответом.

Задача D. Уничтожение дронов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

После того, как Ральф сбежал из своей игры, его начали искать — за ним было послано n специально обученных дронов. Однако, Ральф не так прост и занял оборонительную позицию с турелью в руках.

Внимательно оценив ситуацию, Ральф понял, что если рассмотреть плоскость, где он находится в начале координат — точке $(0, 0)$, то получится, что i -й из дронов находится в точке с координатами (x_i, y_i) . Однако, пока Ральф разведывал ситуацию, дроны его заметили, а значит пора действовать. За одну секунду Ральф может поразить из турели любого дрона, а все уцелевшие дроны после этого могут передвинуться в любую из 8 соседних для них по горизонтали, вертикали или диагонали точек (при этом некоторые дроны могут оказаться в точках с одинаковыми координатами).

Задача дронов — добраться до Ральфа, то есть до точки $(0, 0)$, а задача Ральфа — поразить всех дронов, пока они до него не добрались. Со своей стороны Ральф гарантирует вам, что ни разу не промахнется и каждым выстрелом будет поражать ровно одного дрона. Вас же он просит сказать ему, в каком порядке их поражать. Помогите ему — скажите, в каком порядке поражать дронов, чтобы они не добрались до точки $(0, 0)$, или скажите, что сделать этого не получится, и Ральфу лучше спасаться бегством.

Формат входных данных

В первой строке содержится число n — количество дронов ($1 \leq n \leq 10^5$).

В i -й из следующих n строк содержатся два числа x_i и y_i — координаты i -го дрона ($|x_i|, |y_i| \leq 10^5$). Гарантируется, что в точке $(0, 0)$ нет дронов.

Формат выходных данных

В единственной строке через пробел выведите n чисел от 1 до n — номера дронов в порядке, в котором Ральфу в них нужно стрелять. Если же какой-то дрон в любом случае доберется до точки $(0, 0)$, в единственной строке выведите «-1».

Если существует несколько решений, разрешается вывести любое из них.

Примеры

стандартный ввод	стандартный вывод
3 0 1 -2 3 2 2	1 3 2
3 0 1 -2 2 2 2	-1

Задача Е. Конфета в лабиринте

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Для постройки машины Ральфу и Ванилопе понадобилось найти длинный леденец. Когда они нашли его, они поняли, что вернуться можно, только пройдя через лабиринт. И разумеется, леденец придется взять с собой.

Лабиринт представляет собой поле n на m клеток. Каждая клетка либо пуста, либо в ней находится стена лабиринта. В том месте, которое нашли Ральф и Ванилопа, леденцы бывают разной длины, но все они занимают k подряд идущих клеток в одной линии для некоторого k . Разумеется, леденец не может находиться в клетке, в которой находится стена лабиринта. Ральф очень сильный, поэтому он может перенести леденец любой длины.

Изначально, Ральф может зайти в лабиринт в любой клетке левого столбца, но он должен держать леденец параллельно левой границе лабиринта. Что бы выйти из лабиринта, он должен оказаться в какой-нибудь клетке правого столбца лабиринта, держа леденец параллельно этой границе.

Когда Ральф держит леденец параллельно одной из границ лабиринта, он может перенести его на одну клетку вдоль этой границы, или же он может взять леденец за один из его концов, поднять его в этом месте в воздух, и опустить его параллельно другой стороне лабиринта. Разумеется, он может сделать эти действия, только если после этих действий леденец будет находиться на пустых клетках.

Теперь Ральфу и Ванилопе интересно, какой максимальной длины леденец можно перенести через лабиринт.

Формат входных данных

В первой строке входных данных находятся два целых числа n и m — количество строк и столбцов в лабиринте соответственно ($1 \leq n, m \leq 300$). В следующих n строках находятся m символов. j -й символ i -й строки равняется «#», если в j -й клетке i -й строки находится стена, иначе он равен «.».

Формат выходных данных

В единственной строке выведете одно число — максимальное количество клеток, которое может занимать леденец, такой что Ральф может перенести его через лабиринт. Если леденец никакой длины нельзя пронести через лабиринт, выведите число 0.

Пример

стандартный ввод	стандартный вывод
3 5 ...## .#.#. ##...	2

Задача F. Женитьба

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Исследуя автоматы в игровом зале, Ральф, нашел, кажется, самую скучную из когда-либо созданных игр. Она носит гордое название «Давай потанцуем». Цель игры заключается в том, чтобы составить удачные пары для танца из данных игроку мальчиков и девочек.

В игре есть n мальчиков, пронумерованных для удобства от 1 до n , и n девочек, также пронумерованных от 1 до n . Все дети расположены вдоль координатной прямой, причем i -й мальчик расположен в точке с координатой b_i , а j -я девочка расположена в точке с координатой g_j . Игра сделана не очень реалистично, поэтому может быть такое, что несколько детей располагаются в одной точке.

Разумеется, у детей есть свои предпочтения, которые, правда, устроены довольно просто: назовем *симпатией* между i -м мальчиком и j -й девочкой величину, обратную расстоянию между ними на прямой, которое в свою очередь вычисляется по формуле $|b_i - g_j|$. Иными словами, чем ближе располагаются мальчик и девочка, тем больше они нравятся друг другу. Обратите внимание, что одному мальчику могут быть одинаково симпатичны сразу несколько девочек, и наоборот.

Игрок должен составить n пар, в каждой из которых должен быть один мальчик и одна девочка, при этом каждый ребенок должен ровно один раз присутствовать в одной из пар.

Однако не любое разбиение на пары подойдет. Пусть мальчик A находится в паре с девочкой a , а мальчик B находится в паре с девочкой b . Будем говорить, что между мальчиком A и девочкой b , возникает *соблазн*, если симпатия между A и b строго больше, чем симпатия между A и a , а также симпатия между B и b . Иными словами, между мальчиком и девочкой возникает соблазн, если симпатия между ними больше, чем симпатия внутри их пар.

Цель игры — разбить всех мальчиков и девочек на пары так, чтобы при этом разбиении не возникало соблазнов. Игра оказалась на удивление затягивающей, однако Ральф никак не может справиться с очередным ее уровнем. Поэтому он попросил вас написать программу, которая будет выигрывать в эту игру либо определять, что это сделать невозможно.

Формат входных данных

Первая строка входных данных содержит единственное целое число n — количество мальчиков и девочек ($1 \leq n \leq 10^5$).

Вторая строка содержит n целых чисел b_i — координаты мальчиков на прямой ($1 \leq b_i \leq 10^9$).

Третья строка содержит n целых чисел g_i — координаты девочек на прямой ($1 \leq g_i \leq 10^9$).

Формат выходных данных

Если невозможно разбить детей на пары так, чтобы соблазнов не возникало, выведите единственное число -1 .

В противном случае выведите n строк, указывающих подходящее разбиение на пары. Каждая строка должна содержать два целых числа от 1 до n — номер мальчика и девочки в очередной паре соответственно.

Если подходящих разбиений на пары несколько, выведите любое из них.

Пример

стандартный ввод	стандартный вывод
3	3 3
6 12 18	1 2
3 9 16	2 1

Задача G. Игра с деревом

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Путешествуя по вселенной игр, Ральф и Ванилопа обнаружили удивительную игру. В этой игре необходимо очень быстро считать, что очень им понравилось. Действие игры происходит вокруг волшебной яблони.

Изначально яблоня состоит только из одного яблока — корня. Номер этого яблока равен 1. После этого Ральф добавляет новое яблоко, которое он связывает с уже существующим с помощью ветки. На каждой ветке Ральф записывает букву латинского алфавита от a до z .

Ванилопа в свою очередь иногда срывает некоторое яблоко. Вместе с яблоком исчезает и ветка, с помощью которой оно было связано. Гарантируется, что никакое другое яблоко не подвешено к яблоку, которое сорвет Ванилопа.

Назовем словом последовательность букв на ветках, идущих от корня к яблоку. Подсловом назовем непустое количество подряд идущих букв в слове.

Вам необходимо после каждого действия посчитать количество различных подслов. Подслова считаются различными, если существуют позиции, в которых стоят различные буквы.

Формат входных данных

В первой строке содержатся два целых числа q — количество действий ($1 \leq q \leq 100\,000$). Каждая из последующих q строк содержит описание действий в следующем формате: $1\ p\ c$ — означает, что Ральф добавляет яблоко с минимальным положительным номером, который еще не был использован ($1 \leq p \leq n$). Предком нового яблока является яблоко v , а на ветке написана латинская буква c . $2\ v$ — Ванилопа срывает яблоко с номером v . Гарантируется, что корневое яблоко не будет сорвано, а также никакое яблоко не будет сорвано дважды.

Формат выходных данных

После каждого действия выведите количество различных подслов.

Пример

стандартный ввод	стандартный вывод
5	1
1 1 a	2
1 1 b	2
1 1 a	4
1 4 c	3
2 3	

Замечание

После первой операции есть слово «a». Различные подслова: «a».

После второй операции слова «a», «b». Различные подслова: «a», «b».

После третьей операции слова «a», «b», «a». Различные подслова: «a», «b».

После четвертой операции слова «a», «b», «ac». Различные подслова: «a», «b», «ac», «c».

После пятой операции слова «a», «ac». Различные подслова: «a», «ac», «c».

Задача Н. Пароли

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Ральф хочет зарегистрироваться на трех сайтах. Для каждого сайта Ральф хочет выбрать свой пароль, причем все три пароля должны быть различны. У Ральфа есть любимая строка s . Для удобства запоминания паролей, Ральф решил разбить s на три части: a , b и c . Будем обозначать последовательное записывание двух строк операцией $+$. Тогда $s = a + b + c$. В качестве паролей Ральф будет использовать $a + b$, $b + c$ и $a + c$.

Помогите Ральфу посчитать количество различных способов разбить строку s на a , b и c , чтобы получившиеся пароли были различны. Два способа являются разными, если в них отличается хотя бы одна из строк a , b или c .

Формат входных данных

В единственной строке дана строка s , состоящая из строчных латинских букв ($1 \leq |s| \leq 500\,000$).

Формат выходных данных

В единственной строке выведите одно целое число — искомое количество разбиений.

Примеры

стандартный ввод	стандартный вывод
aabc	2
ababcb	9

Задача I. Случайное дерево

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Рассмотрим следующий процесс построения дерева T .

Изначально дерево состоит из одной вершины, которая имеет номер 1.

Дальше в дерево добавляются вершины с номерами $2 \dots n$.

На i -м шаге в дерево добавляется вершина с номером $i + 1$, также в дерево добавляется ребро из нее в некоторую уже добавленную вершину p ($1 \leq p \leq i$), которая выбирается среди них случайно равномерно.

Пусть V — множество уже добавленных в T вершин.

Тогда пусть $f(A)$ — количество вершин T , что они лежат или в A , или на пути между какими-либо двумя вершинами из A (возможно, и там, и там).

Ваша задача после добавления каждой вершины вывести сумму $f(A)$ по всем множествам A , которые являются подмножествами множества уже добавленных в T вершин ($\sum f(A)$ по всем $A \subseteq V$). Так как ответы могут быть очень большим, выводите лишь остатки от деления ответа на 998244353.

Формат входных данных

Первая строка входного файла содержит одно число n — количество вершин в дереве после последнего шага ($2 \leq n \leq 200000$).

В следующей строке расположено $n - 1$ целое число p_1, p_2, \dots, p_n ($1 \leq p_i \leq i$), обозначающих добавления в граф вершины $i + 1$ и ребра между p_i и $i + 1$ соответственно. Гарантируется, что p_i выбрано случайно равномерно среди чисел от 1 до i .

Формат выходных данных

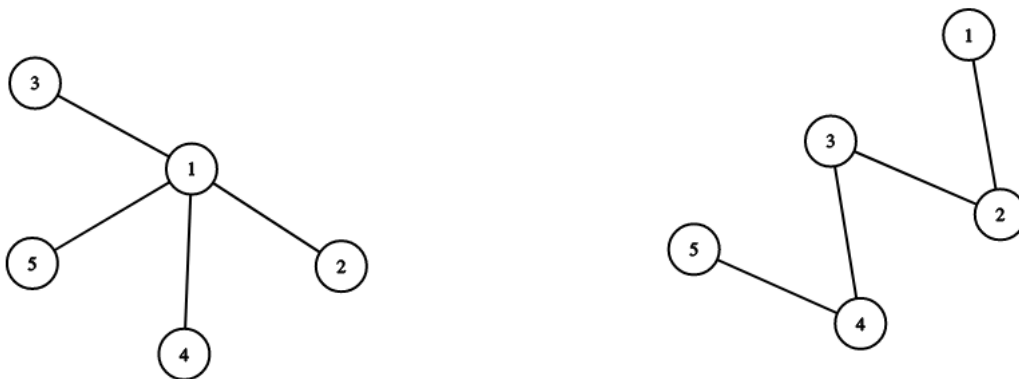
Выведите $n - 1$ целое число, остатки от деления ($\sum f(A)$ по всем $A \subseteq V$) на 998244353 после добавления каждой вершины.

Примеры

стандартный ввод	стандартный вывод
5 1 1 1 1	4 13 36 91
7 1 2 3 4 5 6	4 13 38 103 264 649

Замечание

Итоговые деревья из примеров:



Задача J. Праздничные вычисления по сахарному модулю

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Как вы уже знаете, Ральф мечтает стать лучшим в какой-нибудь игре, завоевать в ней золотую медаль и стать настоящим героем! На этот раз Ральф решил всем показать что он не только сильный и храбрый, но еще и очень умный, именно поэтому он отправился в игру «Праздничные вычисления по сахарному модулю».

Цель игры проста: игроку даются два числа. Используя калькулятор, требуется найти $x \oplus y$. Выражение $x \oplus y$ обозначает применение операции побитового исключающего или (побитового сложения по модулю 2) к числам x и y . Данная операция есть во всех современных языках программирования, например, в языке C++ и Java она обозначается `^`, в Pascal — `xor`.

Калькулятор хранит в памяти все числа, которые были получены игроком во время игры, а так же умеет складывать числа, вычитать их, умножать или целочисленно делить число на 2. При этом память калькулятора, конечно же, ограничена: он не может хранить более 1000 целых чисел. Кроме того, все числа, хранящиеся в калькуляторе должны лежать в диапазоне от 0 до $2^{31} - 1$ включительно. Изначально в памяти калькулятора лежат числа x и y . Игрок может использовать только числа, хранящиеся в памяти калькулятора.

Ральф — очень умный парень, вот только с Ванилопой снова стряслась беда, он спешит к ней на помощь. Поэтому стать лучшим в игре «Праздничные вычисления по сахарному модулю» придется именно вам!

Формат входных данных

В первой строке даны два целых числа x и y — числа, которые изначально лежат в памяти калькулятора ($1 \leq x, y \leq 10^9$).

Формат выходных данных

В первой строке выведите n — количество действий, которое нужно совершить игроку для победы в игре ($1 \leq n \leq 1000$).

В каждой из последующих n строк выведите сначала тип действия, который вы хотите совершить на текущем шаге:

- 1 — сложить два числа
- 2 — вычесть из первого числа второе
- 3 — умножить число на 2
- 4 — целочисленно поделить число на 2.

Если тип операции 1 или 2, далее через пробел выведите два числа — номера итераций, на которых каждое из используемых чисел было получено. Если тип операции 3 или 4, выведите одно число — номер итерации, на которой используемое число было получено.

Например, чтобы вычесть из числа, полученного на итерации 3 число, полученное на итерации 4, следует вывести «2 3 4».

Будем считать, что числа x и y из входных данных были получены на 1-й и 2-й итерациях соответственно.

Обратите внимание, что число $x \oplus y$ должно быть получено **на последней** итерации. От вас **не требуется** найти минимальный по количеству действий ответ.

Примеры

стандартный ввод	стандартный вывод
1 2	3 3 1 4 2 1 3 4
15 4	1 2 1 2