

Задача 1А. [С] Время в зазеркалье

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

В офисе, где работает Анастасия, установлены стильные дизайнерские часы. Их циферблат имеет стандартную разметку: на окружности расположены 60 делений, соответствующих минутам, 12 из которых (начиная с расположенного вверх от центра окружности, далее равномерно с шагом в пять делений) сделаны крупнее остальных, то есть соответствуют часам. Стильными эти часы делает тот факт, что циферблат не содержит никаких цифр, так как предполагается, что всем хорошо известно, какое деление соответствует какому значению текущего времени.

Сегодня над рабочим местом Анастасии повесили такие часы. Периодически поглядывая на них, она сначала заметила некоторую странность в направлении движения стрелок. Приглядевшись внимательнее, Анастасия обнаружила, что на самом деле над её рабочим местом находится зеркало, а часы расположены на стене за её спиной. Это означает, что Анастасия видит циферблат отражённым относительно вертикальной оси, проходящей через его центр. Теперь она хочет научиться быстро определять настоящее текущее время, зная время, которое показывается на отражённом циферблате.

Часы устроены таким образом, что обе стрелки движутся **дискретно**, то есть что часовая стрелка всегда указывает на одно из 12 крупных делений, соответствующих текущему количеству часов, а минутная стрелка на одно из 60 делений, соответствующее текущему количеству минут.

Формат входных данных

В единственной строке входных данных записаны два целых числа h и m ($0 \leq h \leq 11$, $0 \leq m \leq 59$) — положение часовой стрелки и положение минутной стрелки в отражённом циферблате соответственно. $h = 0$ означает, что часовая стрелка указывает вертикально вверх, $h = 3$ соответствуют стрелке, направленной строго направо, $h = 6$ — стрелка смотрит вертикально вниз, $h = 9$ — строго налево. Аналогичные указания верны для минутной стрелки для значений $m = 0$, $m = 15$, $m = 30$ и $m = 45$.

Формат выходных данных

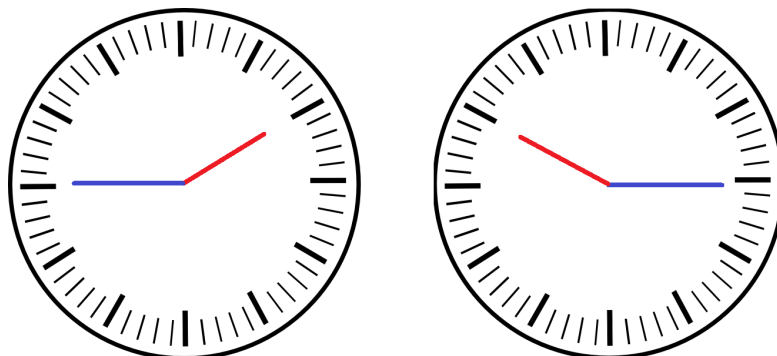
Выведите два целых числа x и y ($0 \leq x \leq 11$, $0 \leq y \leq 59$) — реальное значение текущего времени на часах.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 2 45 | 10 15 |
| 6 0 | 6 0 |

Замечание

На картинке изображён первый тестовый пример. Левый циферблат соответствует тому, что видит Анастасия, а правый — реальному положению стрелок. Часовая стрелка изображена красным цветом, а минутная синим.



Задача 1В. [С] Матрица

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

Иннокентий недавно начал изучать линейную алгебру. Он узнал, что матрица - это прямоугольная таблица, заполненная числами. А квадратная матрица - это матрица, у которой число строк совпадает с числом столбцов и это число называется порядком матрицы. А еще у квадратной матрицы есть 2 диагонали - главная (из верхнего левого угла в правый нижний) и побочная (из нижнего левого в правый верхний).

Теперь Иннокентий делает упражнения и закрепляет полученные знания на практике. Но иногда он ошибается и поэтому просит Вас написать программу, которая будет заполнять квадратную матрицу по следующим правилам: все элементы выше побочной диагонали - нули, на ней - единицы, ниже нее - двойки.

Формат входных данных

Одно натуральное число N ($N \leq 1000$) — порядок матрицы.

Формат выходных данных

Ваша программа должна вывести на печать квадратную матрицу порядка N , у которой все элементы выше побочной диагонали - нули, на побочной диагонали - единицы, ниже нее - двойки.

Пример

| стандартный ввод | стандартный вывод |
|------------------|---|
| 5 | 00001 00012 00122 01222 12222 |

Задача 1С. [C] Непрерывные сегменты

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Прямоугольник $m \times n$ составлен из единичных квадратов. Некоторые из этих квадратов пусты, на остальных проведён один отрезок: любая из двух диагоналей или любой из двух отрезков, соединяющих середины противоположных сторон.

Назовём сегментом набор единичных квадратов, изображения на которых образуют один отрезок, который нельзя расширить до другого набора единичных квадратов, изображения на которых также образуют один отрезок.

Ваша задача — по заданным рисункам на единичных квадратах найти количество сегментов.

Формат входных данных

Первая строка входного файла содержит два целых числа n и m ($1 \leq n, m \leq 100$) — высота и ширина прямоугольника соответственно.

Следующие n строк содержат по m символов «/», «\», «-», «|» или «.», задающих прямоугольник. Первые четыре символа обозначают отрезок под соответствующим углом, а последний — пустую клетку.

Формат выходных данных

Выходной файл должен содержать одно целое число — количество сегментов.

Примеры

| stdin | stdout |
|---|--------|
| 4 5 .. /./ ./ /. /. | 4 |

Задача 1D. [C] Воздухоплавательный парк Байтландии

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

Среди отраслей техники, получивших бурное развитие в начале XX века, было и воздухоплавание. Система управления первых дирижаблей была весьма примитивна. Так, функции управления первым байтландским дирижаблем «Byteland» можно описать следующим набором команд:

- «Fwd» — начать движение вперёд на скорости 6 км/ч;
- «Back» — начать движение назад на скорости 6 км/ч;
- «More» — при движении вперёд увеличить скорость на 6 км/ч, при этом максимальная скорость не превышает 30 км/ч;
- «Less» — при движении вперёд уменьшить скорость на 6 км/ч;
- «Stop» — прекратить движение;
- «Rght» — повернуться направо;
- «Left» — повернуться налево;
- «Pass» — не менять параметров.

В начале каждой минуты отдаётся одна из вышеперечисленных команд. При этом команды «Fwd», «Back», «Rght» и «Left» действуют только в ситуации неподвижно висящего дирижабля, в противном случае они эквивалентны команде «Pass». То же самое касается и команд «More» и «Less» — в случае, когда дирижабль двигается назад или висит неподвижно, эти команды игнорируются.

У Вас есть список из n команд, которые были отданы в первые n минут испытательного полёта дирижабля. Команды отдавались каждую минуту. Изначально дирижабль находился в начале координат, точки $(0, 1)$ и $(1, 0)$ находятся на расстоянии 1 метра от начала координат, ось x направлена с запада на восток, ось y — с юга на север, сам дирижабль повернут в направлении на север.

Требуется выяснить координаты дирижабля после первых n минут полёта.

Формат входных данных

В первой строке входного файла задано одно целое число n ($1 \leq n \leq 1000$) — количество минут, прошедших с начала полёта. i -я из последующих n строк содержит команду, отданную в i -ю минуту полёта дирижабля.

Формат выходных данных

Два целых числа — x и y — координаты дирижабля после n минут полёта.

Примеры

| стандартный ввод | стандартный вывод |
|--|-------------------|
| 6 Fwd Rgt Stop Rgt Fwd More | 300 200 |
| 3 More More More | 0 0 |
| 7 Fwd More More More More More Pass | 0 2500 |

Задача 1E. [C] Фактор палиндромности

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

Аркадий — большой фанат использования машинного обучения в любой задаче. Он верит в безграничную силу волшебства этой популярной молодой науки. Именно поэтому Аркадий постоянно придумывает всё новые и новые факторы, которые можно вычислить для различных объектов.

Напомним, *палиндромом* называется строка, которая одинаково читается от начала к концу и от конца к началу. Для каждой строки в своей базе данных Аркадий хочет найти самую короткую её **подстроку**, состоящую хотя бы из двух символов и являющуюся палиндромом. Если таких подстрок несколько, Аркадий хочет выбрать лексикографически минимальную.

Формат входных данных

В единственной строке входных данных записана одна строка из базы Аркадия — непустая последовательность строчных букв английского алфавита. Длина строки составляет не менее 2 и не превосходит 200 000 символов.

Формат выходных данных

Выведите минимальную по длине подстроку строки из входных данных, состоящую хотя бы из двух символов и являющуюся палиндромом. Напомним, что среди всех таких строк Аркадий хочет найти лексикографически минимальную.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| abac | aba |
| yandex | -1 |

Замечание

Говорят, что строка $a = a_1a_2 \dots a_n$ лексикографически меньше строки $b = b_1b_2 \dots b_m$, если верно одно из двух условий:

- либо $n < m$ и $a_1 = b_1, a_2 = b_2, \dots, a_n = b_n$, то есть первая строка является префиксом второй;
- либо есть такая позиция $1 \leq i \leq \min(n, m)$, что $a_1 = b_1, a_2 = b_2, \dots, a_{i-1} = b_{i-1}$ и $a_i < b_i$, то есть в первой позиции, в которой строки различаются, в первой строке стоит меньшая буква.

Задача 1F. [C] Юбилей

Имя входного файла: *стандартный ввод* или `input.txt`
Имя выходного файла: *стандартный вывод* или `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Ромка недавно вернулся с юбилея своего друга Сашки. Праздник отмечался с размахом и Ромка подумал: а что, если бы юбилеи были чаще? Можно же не ограничиваться десятичной системой счисления.

Ромка ввёл понятие юбилейности числа, равное максимальному количеству нулей в конце записи этого числа в какой-то системе счисления с основанием B , где B — целое число, большее единицы.

Например, юбилейность числа 256 равна 8, так как в двоичной системе счисления оно оканчивается на 8 нулей.

Ромка хочет узнать, когда его ближайший следующий значимый юбилей, если сейчас ему X лет? Значимым юбилеем он считает количество лет, которое обладает юбилейностью как минимум L .

Формат входных данных

Единственная строка ввода содержит два целых числа X и L ($1 \leq X \leq 10^{12}$; $1 \leq L \leq 50$).

Формат выходных данных

Выведите единственное число — ответ на задачу.

Примеры

| <i>стандартный ввод</i> или <code>input.txt</code> | <i>стандартный вывод</i> или <code>output.txt</code> |
|--|--|
| 9990 4 | 10000 |
| 100 8 | 256 |
| 100 2 | 104 |

Задача 2А. [С-В'] Отрицание

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Дано целое число x . Выведите $-x$.

Формат входных данных

Входной файл содержит одно целое число x ($|x| \leq 2^{64}$).

Формат выходных данных

Выходной файл должен содержать одно целое число $-x$.

Примеры

| <code>stdin</code> | <code>stdout</code> |
|--------------------|---------------------|
| -17 | 17 |

Задача 2В. [С-В'] Отрицляндия

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

Миша хочет вернуться к себе на родину в Отрицляндию, но за все нужно платить, а за вход в Отрицляндию особенно.

Платой за вход необычна, как и сама страна: необходимо перевести заданное стражником число в систему счисления с основанием -2 .

Миша не умеет переводить числа в системы счисления с отрицательным основанием, поэтому он просит вас о помощи.

Формат входных данных

В первой строке дано целое число N — это число загаданное стражником в десятичной системе счисления. Известно, что N по абсолютной величине не превосходит 10^9 .

Формат выходных данных

На единственной строке выведите без ведущих нулей представление числа N в системе счисления с основанием -2 .

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 | 111 |
| -12 | 110100 |
| 12 | 11100 |

Задача 2С. [С-В'] Бизоны и цветы

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

Бизон Виталий на лесной барахолке приобрел три разноцветных цветочных горшка: красного, зеленого и синего цвета. Придя домой, он поставил их на подоконник в следующем, если смотреть слева направо, порядке: сначала красный, затем зеленый и за ним синий.

Чтобы каждый новый день был не похож на другой, бизон решил менять горшки местами по следующим двум правилам:

1. В нечетный день Виталий будет менять местами левый и центральный горшки.
2. В четный день Виталий поменяет местами правый и центральный горшки.

Виталию стало интересно: в каком порядке будут стоять горшки по прошествии N дней. Но для него это слишком сложная задача, поэтому он просит вас помочь ему.

Формат входных данных

В первой строке дано натуральное число N ($N \leq 10^{18}$) — число дней, по прошествии которых нужно рассчитать положение горшков.

Формат выходных данных

Выведите одну строку, которая состоит из трех символов: положение горшков с цветами (слева направо) по прошествии N дней. Цвет горшка обозначается большой латинской буквой: красный горшок обозначается буквой R , зеленый обозначается буквой G , а синий — буквой B .

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 1 | GRB |
| 2 | GBR |

Задача 2D. [C-B'] Кто подставил кроликов Роджеру

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

Байтландский биолог и естествоиспытатель Роджер Эпикфейл установил, что под влиянием радио, паровых машин и других техногенных факторов скорость размножения кроликов изменилась со времён Фибоначчи: в x -й с начала наблюдения день рождается x хог $(x - 1)$ кроликов.

Для проверки своей гипотезы Роджер хочет посчитать, какое суммарное количество кроликов родится в интервал с P -го по Q -й включительно дни наблюдения.

Формат входных данных

Первая строка входа содержит два целых числа P и Q ($1 \leq P, Q \leq 10^9$).

Формат выходных данных

Выведите одно целое число — суммарное количество кроликов, родившихся в дни с P -го по Q -й.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 2 7 | 16 |
| 3 10 | 32 |

Задача 2E. [C-B'] Block Shuffling

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Есть тождественная перестановка из 2^p элементов $(1, 2, 3, \dots, 2^p)$. Требуется произвести с ней последовательность операций следующего вида. По заданному числу k , $0 \leq k < p$, нужно разбить перестановку на непрерывные непересекающиеся блоки длиной 2^k . То есть, элементы на позициях $(1, \dots, 2^k)$ формируют первый блок, элементы на позициях $(2^k + 1, \dots, 2 \cdot 2^k)$ — второй блок, элементы $(2 \cdot 2^k + 1, \dots, 3 \cdot 2^k)$ — третий блок, и так далее. После этого соседние блоки поменять местами. То есть, первый блок меняется со вторым, третий с четвертым и так далее.

Дано множество позиций; требуется сказать, какие элементы перестановки находятся на этих позициях после выполнения всех операций.

Формат входных данных

Первая строка содержит три числа N , K и Q ($N = 2^p$, $0 \leq p \leq 21$, $1 \leq K, Q \leq 100\,000$) — количество элементов перестановки, количество операций над перестановкой и количество интересующих позиций.

Далее следуют K строк по одному числу k_i в каждой ($0 \leq k_i < p$), означающих, что нужно менять соседние блоки по 2^{k_i} элементов.

Далее следуют Q строк по одному числу q_i в каждой ($1 \leq q_i \leq N$) — номера интересующих позиций.

Формат выходных данных

Вывести Q строк, в i -ой из которых номер элемента перестановки на позиции q_i .

Примеры

| stdin | stdout |
|-------|--------|
| 8 3 3 | 8 |
| 0 | 6 |
| 1 | 2 |
| 2 | |
| 1 | |
| 3 | |
| 7 | |

Задача 2F. [С-В'] Перенесем всё!

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Анализируя выходные и праздники в районе Нового года и 1 мая в России, президент Флатландии пришел к выводу, что отдых своих граждан можно оптимизировать кардинально. Главное — это чтобы им, гражданам, не пришлось работать больше 6 дней подряд в одном календарном году. Он поручил Министерству труда разработать такой график переноса выходных дней (суббот и воскресений), чтобы с учетом имеющихся во Флатландии праздников граждане смогли отдыхать как можно больше дней подряд.

Напомним, что если государственный праздник Флатландии приходится на выходной день (субботу или воскресенье), то этот выходной автоматически переносится на первый после праздника рабочий день. Но по указу президента любой выходной, как совпавший с праздничным днем, так и не совпавший, может быть перенесен на любой рабочий день. Праздники при этом никогда не переносятся.

Напишите программу, которая поможет Министерству труда составить требуемый график переноса выходных на очередной год. Праздники и выходные как предыдущего, так и следующего года при этом учитывать не нужно. Максимизировать требуется число дней отдыха подряд именно в одном году.

Формат входных данных

В первой строке входных данных находятся два числа — номер года Y ($2012 \leq Y \leq 2050$) и номер дня недели для 1 января этого года W ($1 \leq W \leq 7$) от понедельника до воскресенья соответственно. В этом диапазоне лет номера високосных годов делятся на 4.

Во второй строке дано число ежегодных праздников N ($0 \leq N \leq 366$) Флатландии. В каждой из следующих N строк записана дата очередного праздника в формате DD.MM. Праздничные дни перечислены в хронологическом порядке. Все даты различны и корректны относительно рассматриваемого года.

Формат выходных данных

Выдайте одно число — максимально возможное число дней непрерывного отдыха жителей Флатландии в указанном году, если выходные перенести так, что число подряд идущих рабочих дней в этом году не будет превышать 6.

Примеры

| <code>stdin</code> | <code>stdout</code> |
|----------------------|---------------------|
| 2012 7 1 01.01 | 63 |

Задача 3А. [В'-В] Автобус не опаздывает, автобус задерживается

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

В городе Байтограде 500 автобусных остановок, пронумерованных последовательными целыми числами от 1 до 500.

Каждый из N автобусных маршрутов города содержит не менее двух остановок (начальную и конечную); например, для i -го маршрута задан набор остановок $s_{i,1}, s_{i,2}, \dots, s_{i,N_i}$ со временами прибытия $t_{i,1}, t_{i,2}, \dots, t_{i,N_i}$. Также задан интервал движения P_i секунд, т.е. автобус прибывает на остановку $s_{i,j}$ в моменты времени $t_{i,j} + k \cdot P_i$ для каждого целого k (отметим, что k может быть и отрицательным). Когда автобус прибывает на конечную остановку, все пассажиры обязаны выйти.

В момент времени 0 Вы находитесь на автобусной остановке с номером K . Ваша задача — запланировать маршрут, который позволит Вам добраться до остановки 1 за наименьшее время. Вы можете выйти на автобусной остановке и ждать нужного Вам автобуса. Чтобы учесть отклонения от расписания, Ваш план должен предусмотреть, что Вы находитесь на каждой остановке (включая начальную) не менее D секунд.

Формат входных данных

Вход состоит не более, чем из 20 тестовых примеров. Каждый тестовый пример состоит из $3R+1$ строк, где R — количество автобусных маршрутов. Первая строка входного файла содержит целые числа H , D и R ($1 < H \leq 500, 0 \leq D \leq 8.64 \cdot 10^4, 1 \leq R \leq 200$) — номер стартовой остановки, количество секунд, которые Вам необходимо провести на каждой остановке, и количество маршрутов в городе.

Строки, начиная с $3i-1$ до $3i+1$ задают i -й автобусный маршрут. Первая из этих строк содержит два целых числа N_i и P_i ($2 \leq N_i \leq 200, 1 \leq P_i \leq 8.64 \cdot 10^4$) — количество остановок и интервал движения. Вторая строка содержит последовательность целых чисел $s_{i,1}, s_{i,2}, \dots, s_{i,N_i}$ ($1 \leq s_{i,j} \leq 500$) — номера автобусных остановок в порядке их следования. Третья строка содержит возрастающую последовательность целых чисел $0 \leq t_{i,1}, t_{i,2}, \dots, t_{i,N_i} \leq 8.64 \cdot 10^4$ — расписание движения автобуса. Гарантируется, что для каждого тестового примера решение существует. Входной файл завершается тестовым примером с $H = D = R = 0$, обрабатывать который не требуется.

Формат выходных данных

Для каждого тестового примера выведите отдельную строку, содержащую наиболее раннее время (в секундах), когда Вы прибудете на остановку с номером 1.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 42 3 1 | 30 |
| 5 10 | 10 |
| 42 2 3 2 1 | |
| 0 4 20 36 40 | |
| 42 0 1 | |
| 5 10 | |
| 42 2 3 2 1 | |
| 0 4 20 36 40 | |
| 0 0 0 | |

Замечание

Тест из условия содержит два тестовых примера, которые отличаются только временем, которое нужно провести на остановке - в первом примере это 3 секунды, во втором - 0. При этом, оба примера содержат лишь один автобусный маршрут - от остановки 42 до остановки 1, и может показаться,

что оптимально будет просто проехать по полному маршруту. Однако это не так! Дело в том, что автобус дважды проезжает через остановку номер 2, а интервал между автобусами этого маршрута оказывается равным 10 секундам. Вследствие этого, оптимальным оказывается:

- сесть на остановке 42 на первый доступный из автобусов, которые приезжают в моменты времени ..., -30, -20, -10, 0, 10, 20, ...; в первом примере нас интересует автобус, отправляющийся в момент времени 10, ибо мы должны провести на стартовой остановке хотя бы 3 секунды; во втором примере мы ждать не должны, поэтому садимся в автобус с временем отправления 0;
- доезжаем до остановки 2 в момент времени 14 в первом примере и в момент времени 4 во втором; в первом примере "висим" еще 3 секунды;
- ждем ближайшего автобуса того же маршрута, но который проезжает остановку 2 уже второй раз. Это происходит в моменты времени ..., 6, 16, 26, 36, 46, В первом примере садимся в автобус в момент времени 26, во втором – в 6;
- едем 4 минуты до остановки 1, завершая поездку в моменты времени 30 и 10, соответственно.

Задача 3В. [В'-В] После финала

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

После финала Чемпионата Мира по программированию, на котором предлагалось n задач (и, соответственно, раздавались шарики n различных цветов), осталось некоторое количество шариков. n из этих шариков (не обязательно попарно различных цветов) привязали к одной точке и выровняли таким образом, чтобы шарики были расположены последовательно.

В случае, если какой-то шарик лопнет, то соседние с ним шарики сдвинутся, и вся конструкция всё равно останется непрерывной. Один из участников, проходя мимо шариков, решил несколько «подправить» расцветку шариков и достал k дротиков. Каждый дротик может пробить ровно один шарик. Участник хочет максимизировать количество шариков одного цвета, идущих подряд (при этом тратить все k дротиков необязательно).

Формат входных данных

В первой строке входа заданы два целых числа n и k ($1 \leq n \leq 1\,000\,000$, $1 \leq k \leq 1\,000\,000$) — общее количество шариков и количество дротиков у участника. В следующей строке заданы n целых чисел c_i ($1 \leq c_i \leq n$) — цвета соответствующих шариков.

Формат выходных данных

Выведите одно число — максимальную длину последовательности одноцветных шариков, которую можно получить, бросая дротики.

Пример

| стандартный ввод | стандартный вывод |
|------------------------|-------------------|
| 8 2 2 4 4 2 1 4 2 4 | 3 |

Задача 3С. [В'-В] Лампочки

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 megabytes

Есть n лампочек и k переключателей. Для каждого переключателя известно, состояния каких лампочек он меняет. Определите на какие переключатели необходимо нажать, чтобы включить все лампочки. Первоначально все лампочки выключены. На каждый выключатель можно нажимать не более одного раза.

Формат входных данных

Первая строка входного файла содержит два целых числа n и k ($1 \leq n, k \leq 15$). Далее следуют k строк, описывающие переключатели. В каждой строке сначала идёт целое положительное число t_i , не превосходящее n — количество лампочек, у которых данный переключатель изменяет состояния. Далее в этой же строке заданы t_i целых чисел — номера этих лампочек. Для одного переключателя номера не повторяются. Лампочки нумеруются с единицы.

Формат выходных данных

Выходной файл должен содержать -1 , если решения не существует. Иначе в первой строке должно быть записано количество переключателей, на которые надо нажать, а в следующей строке их номера. Переключатели нумеруются с единицы в том порядке, в котором описываются во входном файле. Все номера переключателей должны быть попарно различны. Если решений несколько, выведите любое из них.

Примеры

| stdin | stdout |
|--|------------|
| 4 5 3 1 2 3 3 1 3 4 2 3 4 2 2 3 1 3 | 3 2 4 5 |
| 3 2 1 1 1 2 | -1 |

Задача 3D. [В'-В] Оффлайн-судейство

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

Жюри Берляндской областной олимпиады по информатике не доверяет разным там автоматическим тестирующим системам, поэтому проверка решений на берляндской областной олимпиаде по информатике проводится следующим способом. Всего в жюри n судей, для каждого судьи определён авторитет a_i . Когда в жюри приносят дискету с очередным решением, каждый член жюри читает исходный код и голосует "OK", если он считает, что решение верно, и "WA", если он считает, что решение неверно. Если после того, как все члены жюри проголосовали, сумма авторитетов судей, считающих, что решение верно, больше или равна p , решение получает вердикт "Accepted", иначе оно получает вердикт "Rejected".

Некоторые члены жюри считают, что система оценки слишком сложна для них, так что они предложили присвоить судьям более «красивые» значения авторитетов b_i и изменить лимит на q . По их мнению, система оценок останется эквивалентной, а считать баллы будет проще.

Министерство образования Берляндской области поручило Вам проверить, действительно ли новая и старая системы эквивалентны, то есть верно ли, что для каждого возможного распределения голосов итоговый вердикт не поменяется. Если системы не эквивалентны, приведите пример распределения голосов, который приводит к разным вердиктам в новой и старой системах.

Формат входных данных

В первой строке дано число n ($1 \leq n \leq 100$).

Во второй строке даны числа p, a_1, a_2, \dots, a_n ($1 \leq p, a_i \leq 10^6$).

В третьей строке даны числа q, b_1, b_2, \dots, b_n ($1 \leq q, b_i \leq 10^6$).

Формат выходных данных

Выведите «YES», если замена возможна. Иначе в первой строке выведите «NO», а во второй — описание контрпримера в виде битовой строки длины n . i -ый бит этой строки должен быть равен 1, если i -ый судья считает, что решение верно, и 0, если он считает, что решение неверно.

Если контрпримеров несколько, выведите любой из них.

Примеры

| стандартный ввод | стандартный вывод |
|-------------------------|-------------------|
| 3 8 4 5 6 2 1 1 1 | YES |
| 3 6 4 5 6 2 1 1 1 | NO 001 |

Задача 3Е. [В'-В] Кубик

| | |
|-------------------------|--------------|
| Имя входного файла: | stdin |
| Имя выходного файла: | stdout |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Имеется сетка, узлы которой в декартовой системе координат имеют целые координаты. Шаг сетки и по x и по y равен 1. Координатами единичных квадратов (клеток), на которые сетка разбивает всю плоскость, будем считать координаты их левых нижних вершин. n клеток сетки поместили числами от 1 до n . На помеченной числом 1 клетке лежит кубик с ребром, равным единице, так, что его нижняя грань совпадает с этой клеткой. На каждой из шести граней этого куба написаны по два натуральных числа a_i и b_i , a_i — четное, b_i — нечетное, где $1 \leq i \leq 6$. Кроме того, имеется табло, на которое можно выводить два натуральных числа a и b .

«Водитель» кубика управляет его перемещением с помощью последовательности команд. Каждая команда имеет либо формат вида kPQ , где k — натуральное число, P — один из символов A или B (A — означает число a , которое в данный момент находится на табло, а B , соответственно, число b , которое в данный момент находится на табло), а Q — один из символов L, U, R, D (L — перемещение влево, R — вправо, U — вверх, D — вниз). Если формат команды имеет вид PQ , то считаем, что k равно 1. Кроме того, существует еще команда S (stop).

Подав команду в формате kPQ , водитель кубика перекачивает его по сетке (при этом за один шаг кубик попадает на соседнюю клетку и внизу оказывается соответствующая грань, соседняя с той, на которой он стоял) в направлении, указанным символом Q , на a (или на b , зависит от P) единиц. Эта операция по перекачиванию кубика в направлении Q на a (или на b) единиц повторяется k раз.

Команда S может быть подана, только если кубик стоит на помеченной клетке. При выполнении этой команды (и только в этом случае) отмеченный квадрат считается посещенным, а на табло помещаются новые значения a и b , равные значениям a_i и b_i , которые написаны на грани, на которой стоит кубик. Задача водителя состоит в том, чтобы посетить как можно больше отмеченных квадратов.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 10\,000$) — количество помеченных клеток. Следующие n строк содержат по паре целых чисел x_i, y_i ($|x_i|, |y_i| \leq 10\,000$) — координаты этих клеток. Ось Ox направлена слева направо, ось Oy — снизу вверх.

Следующие 6 строк содержат по паре чисел a_i, b_i — числа на нижней, передней, верхней, задней, левой и правой стороне кубика соответственно ($1 \leq a_i, b_i \leq 50$, a_i — чётное число, b_i — нечётное). Передняя сторона смотрит в направлении уменьшения координаты y , левая — вдоль уменьшения x .

Формат выходных данных

Выходной файл должен содержать одну строку не длиннее 3×10^6 символов — набор команд для кубика, при выполнении которого он посетит как можно больше отмеченных клеток. Последовательность команд должна начинаться с команды S . В процессе выполнения команд каждая координата кубика не должна превышать 10^9 по абсолютной величине.

Примеры

| stdin | stdout |
|---|---------------|
| 3 0 0 2 1 30 15 2 1 6 3 10 5 14 7 12 9 20 11 | S1ARBUS2ARBUS |

Задача 3F. [B'-B] Начало славного пути

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

И снова армия Байтландии проводит масштабные учения. По приказу Главнокомандующего прибыло несколько элитных подразделений, в их числе — 91-й пехотный полк байтландской армии. Только что принявший полк командир построил личный состав в одну шеренгу на плацу по росту. Рядовые байтландской армии носят фиолетовые парадные мундиры, унтер-офицеры - белые. При этом среди личного состава полка один унтер-офицер приходится ровно на k солдат.

Для распределения солдат по взводам командир использовал следующий метод: каждую минуту он выбирал k рядовых и одного унтер-офицера, которые на начало минуты находились в одном «связном» участке строя (то есть между любыми двумя выбираемыми в одну и ту же минуту военнослужащими не должно быть пустого места в строю) и приказывал им выйти из шеренги, сформировав очередной взвод. Если военнослужащий выходит из шеренги, то его место остаётся пустым.

Требуется помочь командиру и найти такую последовательность выбора, чтобы в результате полк был полностью поделен на взводы.

Формат входных данных

Первая строка входа содержит два целых числа n и k ($2 \leq n \leq 1\,000\,000$, $1 \leq k \leq n - 1$) — численность полка в байтландской армии и количество солдат, приходящихся на одного унтер-офицера. Во второй строке заданы n символов «v» и «w», задающих цвета мундиров; i -й из этих символов равен «v», если на i -м месте стоит рядовой, и «w», если там стоит унтер-офицер. Гарантируется, что n делится на $k + 1$ и для каждого теста существует решающая задачу последовательность выбора.

Формат выходных данных

Выведите $n/(k + 1)$ строк, задающих выбор командира в начале каждой минуты. i -я строка должна содержать $k + 1$ номеров военнослужащих, выбираемых в соответствующую минуту. Места в шеренге пронумерованы с единицы. Принимается любая последовательность, удовлетворяющая условию задачи.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 12 2 | 10 11 12 |
| wwwvvvvvvvvwv | 1 8 9 |
| | 2 6 7 |
| | 3 4 5 |

Задача 4А. [В-Х] Матрица

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

Нео наконец понял, как устроена Матрица.

Матрица — это квадратная таблица размера $n \times n$, в каждой из ячеек которой находится 0 или

1. При этом должны выполняться следующие ограничения:

- в каждом столбце имеется хотя бы одна единица; все единицы в столбце идут подряд (то есть между любыми двумя единицами в одном столбце нет нулей);
- в каждой строке количество единиц не меньше a и не больше b .

Матрица последовательно принимает все свои допустимые значения (то есть значения, при которых ограничения выполняются) в следующем порядке. Определим состояние Матрицы как последовательность из n^2 элементов, полученную последовательным выписыванием всех строк Матрицы друг за другом $(a_{1,1}, \dots, a_{1,n}, a_{2,1}, \dots, a_{n,n})$. Все состояния Матрицы, соответствующие допустимым значениям, упорядочиваются лексикографически, и в момент времени t Матрица принимает значение, соответствующее t -му в получившемся списке состоянию.

Нео научился восстанавливать Матрицу в любой момент времени. А вы так сможете?

Формат входных данных

В первой строке находятся четыре целых числа n , a , b и q ($1 \leq n \leq 10$, $1 \leq a \leq b \leq n$, $1 \leq q \leq 1000$). В i -ой из последующих q строк находится число t_i — интересующий момент времени ($1 \leq t_i \leq 10^{18}$).

Формат выходных данных

Для i -го запроса из входных данных выведите n строк по n символов в каждой — значение Матрицы в t_i -й момент времени. Если допустимых состояний Матрицы меньше, чем t_i , то выведите «No such matrix.» без кавычек. Ответы на запросы разделяйте переводом строки. Смотрите пример для более точного понимания.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 2 3 5 | 011 |
| 1 | 011 |
| 2 | 101 |
| 16 | |
| 34 | 011 |
| 35 | 011 |
| | 110 |
| | 101 |
| | 111 |
| | 101 |
| | 111 |
| | 111 |
| | 111 |
| | No such matrix. |

Задача 4В. [B-X] Mortal Points

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На прямой есть N точек. Точки бывают двух цветов: черные и белые. С точками происходит такой процесс: каждую итерацию любая точка, которая видит хотя бы одну точку противоположного цвета, умирает. Точка A видит точку B , если отрезок AB не содержит других живых точек множества. В течение одной итерации все необходимые точки умирают одновременно. Если не осталось живых точек какого-нибудь цвета, процесс заканчивается.

Определить, сколько итераций будет продолжаться процесс, и точки какого цвета выживут в результате.

Формат входных данных

В первой строке число N ($1 \leq N \leq 100\,000$) — количество точек на прямой. Далее N строк по два целых числа в каждой: x_i и $color_i$ ($0 \leq x_i \leq 1\,000\,000$, $color_i \in \{0, 1\}$) — координата и цвет i -ой точки. 0 означает белый, 1 — черный. Никакие две точки не имеют одинаковые координаты.

Формат выходных данных

Слово “Draw”, если в результате умерли все точки, “Black”, если черные точки выжили, или “White”, если выжили белые. В той же строке через пробел вывести количество итераций процесса.

Примеры

| stdin | stdout |
|------------------------|---------|
| 3 0 0 1 1 2 1 | Black 1 |

Задача 4С. [В-Х] Доверяй, но проверяй

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 4 секунды |
| Ограничение по памяти: | 256 мегабайт |

Как-то днём командир одной из рот 91-го байтландского заметил в небе аэроплан без опознавательных знаков. Он решил пронаблюдать за траекторией аэроплана и нанёс её на листок бумаги. Оказалось, что пилот совершал полёт по траектории, лежащей в горизонтальной плоскости и представляющей собой замкнутую ломаную. Донесение вместе со схемой наблюдения было отправлено в штаб полка. При этом, в соответствии с инструкцией, на схеме в качестве вершин ломаной были отмечены не только точки поворотов, но и точки, в которых аэроплан «собирался поворачивать» (то есть качал крыльями), но так и не повернул; соседние в порядке облёта вершины ломаной были соединены стрелкой, указывающей направление движения (если по какому-то участку полёт проходил многократно, использовался карандаш другого цвета).

Через некоторое время в штаб пришло другое донесение — полёт аэроплана независимо наблюдал один из сержантов. Он тоже заметил, что траектория лежала в горизонтальной плоскости, и при изображении траектории на листе бумаги в соответствии с инструкцией получил замкнутую ломаную с тем же числом вершин.

Командиру полка надо выяснить, могли ли эти два донесения описывать полёт одного и того же аэроплана. Две схемы описывают полёт одного и того же аэроплана, если одна схема переводится в другую с помощью масштабирования всей схемы, сдвигов и поворотов вокруг некоторой точки.

Формат входных данных

В первой строке ввода задано число N ($2 \leq N \leq 5 \cdot 10^5$) — количество вершин ломаных. В каждой из последующих N строк заданы по два целых числа x_i и y_i ; i -я строка содержит координаты i -й в порядке обхода точки ($0 \leq x_i, y_i \leq 3 \cdot 10^4$).

Звенья ломаной проведены между i -й и $i+1$ -й точками для $1 \leq i \leq N-1$, а также между N -й и первой. При этом гарантируется, что ни одно из звеньев не имеет нулевой длины. В последующих N строках в аналогичном формате и с такими же ограничениями заданы координаты вершин второй ломаной a_j и b_j .

Формат выходных данных

Если полученные командиром полка донесения не могут описывать полёт одного и того же аэроплана, выведите 0. Иначе выведите одно целое число — наименьший номер вершины второй ломаной, соответствующей вершине (x_1, y_1) .

Примеры

| стандартный ввод | стандартный вывод |
|---|-------------------|
| 3 1 2 2 1 1 1 2 1 1 2 1 1 | 0 |
| 4 0 0 0 1 1 1 0 1 -1 -1 -1 0 0 0 -1 0 | 1 |
| 3 2 2 3 2 2 3 9 9 49 9 29 29 | 3 |

Задача 4D. [B-X] Забор

| | |
|-------------------------|--------------|
| Имя входного файла: | stdin |
| Имя выходного файла: | stdout |
| Ограничение по времени: | 2 секунды |
| Ограничение по памяти: | 256 мегабайт |

Необходимо с помощью забора огородить выпуклый участок максимальной площади. Правила построения забора следующие. Разрешается выстраивать непрерывный замкнутый забор из отрезков длины 1 и отрезков длины $\sqrt{2}$, соединенных концами между собой. Необходимо использовать ровно n_1 единичных отрезков, параллельных оси x , ровно n_2 единичных отрезков параллельных оси y , ровно n_3 отрезков длины $\sqrt{2}$ образующих угол в 45 градусов с положительной полуосью Ox , ровно n_4 отрезков длины $\sqrt{2}$ образующих угол в 135 градусов с положительной полуосью Ox . Использование отрезков под углами, отличными от вышеперечисленных, не допускается.

Формат входных данных

Входной файл содержит четыре целых числа n_1, n_2, n_3 и n_4 ($1 \leq n_i \leq 800$).

Формат выходных данных

Если забора, удовлетворяющего условиям задачи, нет, то выведите одну строку «impossible».

В противном случае выведите ответ в виде строки из четырех цифр (1 — горизонтальный отрезок, 2 — вертикальный, 3 — диагональный под углом 45 градусов, 4 — диагональный под углом 135 градусов), соответствующих обходу забора по часовой стрелке, начиная с любого отрезка, из которых составлен забор. Если ответов несколько — выведите любой.

Примеры

| stdin | stdout |
|---------|-------------------|
| 5 7 1 4 | 11144222311442222 |
| 5 7 1 5 | impossible |

Задача 4Е. [В-Х] Вентиляция

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

Норман заблудился в вентиляции и уже четвёртую неделю ищет свою квартиру.

Вентиляция состоит из n узлов, соединённых $n - 1$ переходами таким образом, что между любыми двумя узлами существует ровно один путь.

Иногда Норман задаётся вопросом: в каком направлении идти, чтобы попасть в некоторый узел. Норман — всего лишь морская свинка, поэтому он не может запомнить все узлы и переходы между ними. Помогите ему узнать, куда идти.

Формат входных данных

В первой строке входного файла задано число n — количество узлов в вентиляции ($2 \leq n \leq 200\,000$).

В следующих $n - 1$ строках описаны переходы — по одному в строке. Каждый переход задаётся номерами узлов, которые он соединяет: a_i и b_i ($1 \leq a_i, b_i \leq n$; $a_i \neq b_i$). Гарантируется, что между любыми двумя узлами существует единственный путь по переходам.

В следующей строке задано число m — количество вопросов Нормана ($1 \leq m \leq 100\,000$).

В следующих m строках описаны вопросы — по одному в строке. Каждый вопрос задаётся номером узла, в котором находится Норман (s_i) и номером узла, куда он хочет попасть (t_i) ($1 \leq s_i, t_i \leq N$; $s_i \neq t_i$).

Узлы нумеруются с 1.

Формат выходных данных

Для каждого вопроса выведите номер узла, в который нужно идти из s_i напрямую, чтобы попасть в t_i . Обратите внимание, что ответ единственный, так как между любыми двумя вершинами существует ровно один путь.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 5 | 3 |
| 1 2 | 4 |
| 1 3 | 1 |
| 1 4 | |
| 3 5 | |
| 3 | |
| 5 2 | |
| 1 4 | |
| 4 3 | |

Задача 4F. [B-X] Армия покемонов

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

Пикачу — милый и дружелюбный покемон, живущий в стае диких пикачу.

Однако недавно стало известно, что команда R хочет украсть всех этих покемонов! Тренер покемонов Андрей решил помочь Пикачу собрать армию для борьбы с командой R.

В первую очередь Андрей посчитал всех покемонов: их оказалось ровно n штук. Затем он установил силу каждого покемона и так получилось, что i -й покемон имеет силу, равную a_i .

В качестве армии Андрей может выбрать любую непустую подпоследовательность покемонов. Иными словами, Андрей выбирает какой-то массив b из k индексов таких, что $1 \leq b_1 < b_2 < \dots < b_k \leq n$, и его армия будет состоять из покемонов с силами $a_{b_1}, a_{b_2}, \dots, a_{b_k}$.

Сила армии вычисляется как знакопеременная сумма элементов подпоследовательности, то есть $a_{b_1} - a_{b_2} + a_{b_3} - a_{b_4} + \dots$.

Андрей экспериментирует с построением покемонов. Он q раз меняет двух покемонов местами, а именно, в i -й раз он менял местами покемонов с номерами l_i и r_i .

Андрею надо знать: какую максимальную силу армии он мог получить при начальной расстановке покемонов, а также после каждого изменения строя?

Помогите Андрею и покемонам, иначе команде R удастся воплотить в жизнь свой коварный план!

Формат входных данных

В первой строке каждого теста находятся два целых числа n и q ($1 \leq n \leq 3 \cdot 10^5, 0 \leq q \leq 3 \cdot 10^5$) — количество покемонов и количество обменов соответственно.

Во второй строке находятся n целых положительных чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — силы покемонов.

i -я из следующих q строк содержит два целых положительных числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$) — номера обмениваемых покемонов в i -й операции.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $3 \cdot 10^5$, а также сумма q по всем тестовым случаям не превосходит $3 \cdot 10^5$.

Формат выходных данных

Выведите $q + 1$ число — максимально возможную силу армии до изменений и после каждого изменения.

Примеры

| стандартный ввод | стандартный вывод |
|---|--------------------------------|
| 3 1 1 3 2 1 2 | 3 4 |
| 2 2 1 2 1 2 1 2 | 2 2 2 |
| 7 5 1 2 5 4 3 6 7 1 2 6 7 3 4 1 2 2 3 | 9 10 10 10 9 11 |

Задача 5А. [X] Первые спецэффекты

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

После успеха первых фильмов известный байтландский режиссёр решил обратиться к классике. Для экранизации был выбран «Макбет»... Для того, чтобы добиться наибольшего эффекта в сцене с движущимся Бирнамским лесом, режиссёр предложил следующий вариант

Муляжи деревьев расставлены на квадратной съёмочной площадке. Снимается очередной кадр, после чего окружающие площадку ассистенты выбегают и переносят муляжи деревьев так, чтобы деревья были расставлены в требуемом для следующего кадра порядке, после чего убегают назад за пределы площадки, чтобы ненароком не попасть в кадр. Во время съёмки ассистенты сменяются вдоль границ площадки таким образом, что перед очередным кадромдвигающий некоторое дерево ассистент находится в ближайшей к данному дереву точке на границе площадки. По команде режиссёра каждый из n ассистентов добегает до «своего» дерева со скоростью r (одинаковой для всех ассистентов), переносит дерево в нужную точку со скоростью $r/2$, затем по кратчайшему пути со скоростью r возвращается за пределы площадки. При этом на обход возможного препятствия (другого ассистента и/или дерева) на пути дополнительное время не тратится и ни одно дерево не ставится на место, где перед этим было другое дерево (в том числе и не остаётся на месте само).

Требуется найти минимально возможную скорость движения ассистентов, при которой возможен переход от одной заданной расстановки деревьев к другой за время t , если внешне деревья между собой неразличимы и деревья для перемещения ассистенты распределяют оптимально.

Формат входных данных

В первой строке входного файла заданы два числа — целое число n , задающее количество деревьев, и вещественное число t , заданное не более, чем с двумя знаками после десятичной точки — интервал времени между съёмкой двух кадров ($0 \leq n \leq 100$, $1 \leq t \leq 200$). В последующих n строках задана исходная расстановка деревьев. Каждая строка содержит координаты некоторого дерева — два вещественных числа x и y ($0 < x, y < 1$). Последние n строк задают в том же формате требуемую расстановку деревьев. При этом любое дерево из первой расстановки может оказаться на месте любого дерева из второй расстановки, кроме того, несколько деревьев в любой из расстановок может оказаться в одной точке, но координаты любых двух деревьев в разных расстановках всегда различны (то есть ни одно дерево на месте не остаётся). Координаты заданы не более, чем с двумя знаками после десятичной точки. Съёмочная площадка представляет собой квадрат со сторонами, параллельными осям координат, и противоположными вершинами $(0, 0)$ и $(1, 1)$.

Формат выходных данных

Выведите одно вещественное число с точностью не хуже 0.01 — минимальное значение скорости, с которой должны перемещаться ассистенты, чтобы успеть переместить все декорации и уйти из кадра.

Примеры

| стандартный ввод | стандартный вывод |
|---|------------------------|
| 1 1.1 0.5 0.5 0.25 0.25 | 1.32464252764718626310 |
| 5 4.2 0.21 0.61 0.83 0.5 0.77 0.19 0.45 0.40 0.32 0.38 0.05 0.07 0.91 0.88 0.39 0.06 0.27 0.59 0.81 0.4 | 0.28385528415712463213 |

Задача 5B. [X] Interactive Casino

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

Это интерактивная задача.

Игра «Битовая рулетка» проходит по следующим правилам.

- У игрока изначально есть 160 фишек.
- За одну ставку игрок может поставить любое целое положительное количество k фишек, не большее, чем количество фишек у игрока в настоящий момент.
- Генерируется целое число от 0 до $2^{20} - 1$. Если сумма его бит нечётна, то игрок выигрывает, т.е. возвращает ставку и получает ещё k фишек. В противном случае игрок теряет ставку. При этом само число игроку не показывается.
- Если фишек у игрока не осталось, он считается проигравшим.
- Если игрок сделал более 200 ставок, он считается проигравшим.
- Если у игрока в какой-то момент времени оказалось 200 или более фишек, он считается выигравшим

На сайте Algoleaks Вы прочитали, что каждое последующее число на барабане вычисляется по формуле $x_i = (487237 * x_{i-1} + 1011807) \bmod 2^{20}$. Откуда берётся первое число, на сайте не написано. Сможете ли Вы гарантированно выиграть?

Формат входных данных

Программа жюри выдаёт Вашей программе на вход одно целое число — количество фишек, которое у Вас есть в наличии перед очередной ставкой, или -1 в случае, если Ваша программа должна немедленно завершиться.

Формат выходных данных

Если полученное на вход число равно -1 , немедленно завершите работу программы (в противном случае корректный код ошибки не гарантируется). Если же на вход получено число $T > 0$, выведите свою ставку — целое число в интервале от 1 до T . Не забудьте очищать буфер вывода командой `flush`.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 160 | 5 |
| 155 | 10 |
| 165 | 15 |
| 180 | 20 |
| -1 | |

Задача 5С. [X] Экстремальные перестановки

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

Перестановкой длины n называется последовательность из n чисел, в которой каждое из чисел от 1 до n встречается ровно 1 раз. Например, $(3, 4, 5, 1, 2)$ и $(1, 2)$ — перестановки, а $(1, 4, 3)$ и $(2, 1, 3, 2)$ — нет.

Назовем перестановку *экстремальной*, если для любых двух соседних чисел в перестановке разница между ними больше или равна минимальному из них. Например, перестановка

$$(3, 1, 2, 4)$$

является экстремальной, поскольку $|3 - 1| \geq \min(3, 1)$, $|1 - 2| \geq \min(1, 2)$ и $|2 - 4| \geq \min(2, 4)$.

Ваша задача — посчитать количество экстремальных перестановок для некоторого **нечетного** n , в которых некоторые элементы зафиксированы.

Формат входных данных

В первой строке дано нечетное целое число n ($1 \leq n \leq 27$).

Во второй строке даны n чисел p_1, p_2, \dots, p_n ($0 \leq p_i \leq n$). Если p_i равно 0 — то на i -ой позиции может быть любое число, иначе i -ое число в рассматриваемых перестановках должно быть равно p_i . Гарантируется, что если $p_i > 0$ и $p_j > 0$ для $1 \leq i, j \leq n$, $i \neq j$, то $p_i \neq p_j$.

Формат выходных данных

Выведите одно число — количество искомым перестановок.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 5 0 0 0 0 0 | 4 |
| 5 0 1 0 0 5 | 1 |

Задача 5D. [X] Смерть

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 5 секунд |
| Ограничение по памяти: | 256 мегабайт |

И когда Он снял четвертую печать, я слышал голос четвертого животного, говорящий: иди и смотри.

И я взглянул, и вот, конь бледный, и на нем всадник, которому имя «смерть»;
и ад следовал за ним;

и дана ему власть над четвертою частью земли — умерщвлять мечом и голодом, и мором и зверями земными.

(Откровение Иоанна Богослова)

Смерть — Четвертый всадник Апокалипсиса, и за этим всадником следует ад. Однако, даже этот всадник готов пощадить некоторые города и оставить их жителей в живых.

Карта страны, которую изучает Смерть, представляет собой клетчатый прямоугольник размера $n \times m$. Каждая клетка — город, и в каждом городе живут люди, подчиняющиеся одному определенному лорду. Смерть хочет пощадить несколько городов так, чтобы выполнялись два правила:

- из любого выжившего города можно добраться в любой другой, перемещаясь только по выжившим городам, соседним по стороне
- в выживших городах проживают люди, подчиняющиеся не более, чем двум различным лордам

Теперь Смерть заинтересовало максимальное количество городов, которые он может пощадить.

Формат входных данных

В первой строке входного файла задано два целых числа n и m ($1 \leq n, m \leq 10^3$) — размеры страны. Следующие n строк содержат по m чисел каждая — номера лордов, которым подчиняются люди в соответствующих городах. Номера лордов — натуральные числа, не превышающие 10^6 .

Формат выходных данных

Выведите в первой строке одно натуральное число — максимальное количество городов, которые смогут выжить после набега смерти. Во второй — два числа: номера лордов, которым подчиняются люди в выживших городах. В случае, если выживут только люди, подчиняющиеся одному лорду, номер этого лорда должен быть выведен два раза.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 5 5 | 17 |
| 1 1 2 2 1 | 1 2 |
| 1 2 2 1 1 | |
| 1 2 3 3 1 | |
| 1 3 3 2 4 | |
| 1 2 2 4 1 | |

Задача 5Е. [X] МЕХ на пути

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

Дано дерево, на каждом ребре которого написано неотрицательное целое число. Вам необходимо ответить на несколько запросов вида «для данных вершин u , v назовите наименьшее неотрицательное целое число, которое **не** встречается среди чисел, написанных на ребрах на пути от u до v ».

Формат входных данных

Первая строка входных данных содержит два числа n и q ($2 \leq n \leq 10^5$, $1 \leq q \leq 10^5$), количество вершин и количество запросов.

Следующие $n - 1$ строк содержат по три числа u_i, v_i, x_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $0 \leq x_i \leq 10^9$), которые описывают ребро дерева (u_i, v_i) , на котором написано число x_i .

Следующие q строк содержат по паре чисел a_j, b_j ($1 \leq a_j, b_j \leq n$), которая обозначает запрос на пути от a_j до b_j .

Формат выходных данных

Для каждого запроса выведите единственное число — минимальное число, которое не встречается на пути.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 7 6 | 0 |
| 2 1 1 | 1 |
| 3 1 2 | 2 |
| 1 4 0 | 2 |
| 4 5 1 | 3 |
| 5 6 3 | 3 |
| 5 7 4 | |
| 1 3 | |
| 4 1 | |
| 2 4 | |
| 2 5 | |
| 3 5 | |
| 3 7 | |

Задача 5F. [X] В бухгалтерии опять всё перепутали

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

Луца и Пуца пошли получать зарплату. Но в бухгалтерии опять всё перепутали. Луца получил зарплату за Пуцу, а Пуца . . .

Пуца не хочет получать за Луцу и хочет доказать бухгалтерии, что она не права.

Пуца работает в крупной компании «MST Inc.», занимающейся информационным сопровождением «Всеберляндской олимпиады школьников по информатике». В компании «MST Inc.» работает n сотрудников, причём у каждого из них, кроме самой «MST», есть ровно один непосредственный начальник и несколько (возможно ноль) непосредственных подчинённых.

Всеми начальниками сотрудника компании «MST Inc.» называется множество, состоящее из его непосредственного начальника и множества начальников его непосредственного начальника. Известно, что у каждого сотрудника кроме самой «MST», «MST» входит в множество начальников этого сотрудника.

Множеством подчинённых у сотрудника называется множество, состоящее из него самого и множеств подчинённых у всех непосредственных подчинённых данного сотрудника. В частности, все сотрудники входят в множество подчинённых у «MST».

Каждый месяц каждому сотруднику начисляется зарплата, причём немаленькая, ведь иначе ни один сотрудник не согласился бы работать с «MST». Известно, что в нулевой месяц работы организации, каждому сотруднику заплатили по c_i бурлей. В качестве поощрения сотрудников «MST» придумала следующее правило: В каждый из следующих m месяцев берётся сотрудник с номером a_i и берётся число s_i — сумма зарплат всех сотрудников во множестве его начальников и подчинённых (включая его самого). Если это число оказывалось слишком большим, s_i берётся по модулю $10^9 + 7$. После этого берётся сотрудник с номером b_i , и к зарплате всех сотрудников, входящих во множество его начальников и подчинённых (включая его самого) прибавляется число s_i . С учётом этого изменения платится зарплата в i -й месяц и пересчитывается зарплата в следующие месяцы.

Вернёмся к Пуце. Пуца хочет показать бухгалтерии компании «MST Inc.» что она всё перепутала, а для этого ему надо узнать, сколько же ему должны были заплатить в каждый из месяцев с нулевого по m -й. К сожалению, в гениальной системе поощрения, разработанной «MST», не может разобраться никто. Поэтому эту задачу поручили вам.

Формат входных данных

В первой строке входных данных даны 2 числа n и m ($1 \leq n, m \leq 10^5$) — число сотрудников компании «MST Inc.» и последний день, когда выплачивалась зарплата Пуце.

Во второй строке записано $n - 1$ число. i -е из них — номер непосредственного начальника сотрудника номер i (i принимает значения от 1 до $n - 1$). При этом «MST» имеет номер 0 и не имеет непосредственного начальника. Пуца имеет номер $n - 1$.

В третьей строке записано n чисел c_i ($1 \leq c_i \leq 10^9$) — зарплата i -го сотрудника в нулевой день.

В каждой из следующих m строк записано по 2 числа a_i и b_i ($0 \leq a_i, b_i \leq n - 1$) — номер человека, на основе которого происходит поощрение и номер человека, к подчинённым и начальникам которого поощрение применяется (более подробно описано в условии).

Формат выходных данных

В единственной строке выведите $m + 1$ число — зарплату Пуцы в каждый из дней с 0-го по m -й. Напоминаем, что Пуца имеет номер $n - 1$. Обратите внимание, что зарплата **не считается** по модулю $10^9 + 7$.

Примеры

| стандартный ввод | стандартный вывод |
|--|-------------------|
| 3 3 0 0 1 1 1 0 0 2 1 1 2 | 1 4 4 28 |
| 4 3 0 1 1 0 1 0 0 0 1 1 3 2 3 | 0 1 6 20 |

Замечание

Пояснение к первому примеру:

В первый день к зарплате каждого сотрудника прибавилось 3 бурля и зарплаты стали соответственно 4, 4, 4.

Во второй день к зарплате сотрудников с номерами 0, 1 прибавилось по 8 бурлей и зарплаты стали соответственно 12, 12, 4.

Во третий день к зарплате сотрудников с номерами 0, 2 прибавилось по 24 бурля и зарплаты стали соответственно 36, 12, 28.