

“Простая” задача

Входной файл – a.in

Выходной файл – a.out

Время работы на одном тесте – 1 с

Рассмотрим множество простых чисел, не превосходящих некоторого N . Некоторые из них можно представить следующим образом:

$p_1 = p_2 * p_3 + p_4 * p_5 + 1$, где p_2, p_3, p_4, p_5 также простые числа.

Требуется для данного N подсчитать количество различных разложений всех простых чисел, не превосходящих N . Разложения, отличающиеся только порядком слагаемых или множителей считаются одинаковыми.

Так, для $N = 120$ искомых разложений 4:

$$107 = 3 \times 5 + 7 \times 13 + 1;$$

$$107 = 3 \times 7 + 5 \times 17 + 1;$$

$$107 = 3 \times 17 + 5 \times 11 + 1;$$

$$113 = 5 \times 11 + 3 \times 19 + 1.$$

Входные данные

На вход программе подается значение N , не превосходящее 50000.

Выходные данные

Выведите искомое количество разложений.

a.in	a.out
120	4

Задача В. Сдача

Имя входного файла:	change.in
Имя выходного файла:	change.out
Максимальное время работы на одном тесте:	2 секунды
Максимальный объем памяти:	64 мб

Когда Миша и Маша покупали подарок, возникла интересная ситуация. У них была в распоряжении только одна большая купюра, а у продавца – некоторое количество мелочи. Дело происходило утром, поэтому продавцу нужно было экономить мелочь, и он хотел отдать сдачу минимальным количеством монет. Подумав некоторое время, они точно определили, с каким количеством монет продавцу придется расстаться.

А вы сможете решить такую задачу?

Формат входных данных

В первой строке входного файла записано число n ($1 \leq n \leq 10$) – количество различных номиналов монет, содержащихся в кассе. Можно считать, что количество монет каждого номинала достаточно.

На следующей строке содержится n целых чисел a_i ($0 < a_i \leq 2000$) – номиналы монет.

В третьей строке записано одно число k ($1 \leq k \leq 1\ 000\ 000$) – сумма, которую нужно набрать.

Формат выходных данных

В выходной файл выведите минимальное количество монет, которое придется отдать продавцу, или -1, если продавец вообще не сможет дать им сдачу.

change.in	change.out
3 1 3 5 13	3
4 5 6 7 8 9	-1

Задача С. Ну и возраст!

Ввод с клавиатуры

Ограничение времени – 1 секунда

Школьники 3 класса на уроке математики подсчитывали возраст знаменитых людей в днях. По полученному им числу и дню, когда проходил урок математики, определить дату рождения по григорианскому календарю для заданной персоны, родившейся не ранее Рождества Христова. Напомним, что годы, которые делятся на 100, но не делятся на 400, високосными не являются.

Входные данные

В первой строке потока ввода находится дата дня, когда проходил урок математики, записанная в виде DD MM YYYY, с помощью трех натуральных чисел, разделенных пробелами. Во второй строке — возраст искомой персоны в днях. Ведущие нули выводить обязательно.

Выходные данные

Ваша программа должна выводить на экран найденную дату рождения по тому же формату для даты, как и при вводе.

Пример входных данных	Выходные данные	Комментарий
01 01 0001 7	25 12 0000	В выходных данных условная дата Рождества Христова, которая используется в этой задаче

Лабиринт

Имя входного файла:

maze.in

Имя выходного файла:

maze.out

Максимальное время работы на одном тесте:

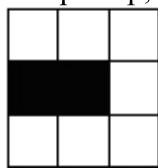
0,2 секунды

Максимальный объем используемой памяти:

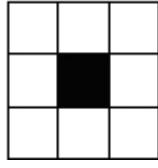
64 мегабайта

Карта местности задана квадратной таблицей A из N ($1 \leq N \leq 500$) строк и N столбцов. Причем элемент $A_{ij} = 0$, если клетка, находящаяся на пересечении i -й строки и j -го столбца свободна (проходима) и $A_{ij} = 1$ в противном случае. Робот может перемещаться только по свободным клеткам, не может выходить за границы карты и из данной клетки может пройти за один ход в одну из четырех соседних (вверх, вниз, влево, вправо, если они свободны). Назовем путем из клетки A_{ab} в клетку A_{cd} такую последовательность корректных ходов робота, выполнив которую он, начиная из клетки A_{ab} , придет в клетку A_{cd} , причем ни одну клетку не посетит два раза. Карта местности такова, что между любыми двумя свободными клетками существует хотя бы один путь. Назовем карту лабиринтом, если между любыми двумя свободными клетками существует ровно один путь.

Например,



- лабиринт, а



- нет.

По введенному числу N и квадратной таблице A размера $N \times N$ определить, является ли карта местности, определяемая таблицей A , лабиринтом.

Формат входных данных

Первая строка входного файла содержит число N . Следующие N строк содержат по N символов каждая: j -й символ в i -й строке равен "0", если $A_{ij} = 0$ и равен "1", если $A_{ij} = 1$.

Формат выходных данных

Выведите в выходной файл слово "Yes", если карта местности является лабиринтом и "No" в противном случае.

Примеры

maze.in	maze.out
3 000 110 000	Yes
3 000 010 000	No

Задача Е. Утраивающаяся последовательность

Ввод с клавиатуры

Ограничение времени – 1 секунда

Бесконечная последовательность чисел $A(1), A(2), \dots$ строится следующим образом: $A(1) = 0$.

Пусть уже построены элементы $A(1), A(2), \dots, A(3^m)$.

Тогда элементы $A(3^m + 1), A(3^m + 2), \dots, A(3^{m+1})$ принимают значения

$A(3^m) + 3^m, A(3^m - 1) + 3^m, \dots, A(1) + 3^m, A(1) + 2 \times 3^m, A(2) + 2 \times 3^m, \dots, A(3^m) + 2 \times 3^m$ соответственно.

Напишите программу, которая по заданному натуральному $N \leq 1\ 000\ 000\ 000$ находит $A(N)$.

Файл исходных данных содержит число N . Вывести в выходной файл значение $A(N)$.

Пример входных данных	Выходные данные для примера входных данных
123457	123456

Бал

Ввод данных

Выход данных

Максимальное время работы на одном тесте:

Максимальный объем используемой памяти:

с клавиатуры

на экран

1 секунда

8 мегабайт

Во время бала в зале, имеющем форму M -угольника $A_1A_2\dots A_M$, этикетом предписано размещаться N придворным дамам вдоль стен и в углах так, чтобы у всех стен стояло равное число дам. Если дама находится в углу зала, то она считается стоящей у обеих стен этого угла. Вдоль стены может размещаться любое количество дам, а в углу не больше одной.

Напишите программу, находящую требуемое расположение дам.

Формат входных данных

На вход подаются два натуральных числа M и N ($3 \leq M \leq 1000$, $1 \leq N \leq 10^9$).

Формат выходных данных

Для каждого угла требуется вывести число дам, стоящих в этом углу (0 или 1), а для каждой стены — количество дам, стоящих вдоль нее (не считая тех, что стоят в углах). Таким образом, нужно вывести $2M$ чисел в соответствии со следующим порядком: сторона A_MA_1 , угол A_1 , сторона A_1A_2 , угол A_2 , ..., сторона $A_{M-1}A_M$, угол A_M .

Если требуемым образом дамам разместиться невозможно, следует вывести единственную строку -1.

Пример

Входные данные	Выходные данные
3 10	3 1 2 1 3 0

Бал

Ввод данных

Выход данных

Максимальное время работы на одном тесте:

Максимальный объем используемой памяти:

с клавиатуры

на экран

1 секунда

8 мегабайт

Во время бала в зале, имеющем форму M -угольника $A_1A_2\dots A_M$, этикетом предписано размещаться N придворным дамам вдоль стен и в углах так, чтобы у всех стен стояло равное число дам. Если дама находится в углу зала, то она считается стоящей у обеих стен этого угла. Вдоль стены может размещаться любое количество дам, а в углу не больше одной.

Напишите программу, находящую требуемое расположение дам.

Формат входных данных

На вход подаются два натуральных числа M и N ($3 \leq M \leq 1000$, $1 \leq N \leq 10^9$).

Формат выходных данных

Для каждого угла требуется вывести число дам, стоящих в этом углу (0 или 1), а для каждой стены — количество дам, стоящих вдоль нее (не считая тех, что стоят в углах). Таким образом, нужно вывести $2M$ чисел в соответствии со следующим порядком: сторона A_MA_1 , угол A_1 , сторона A_1A_2 , угол A_2 , ..., сторона $A_{M-1}A_M$, угол A_M .

Если требуемым образом дамам разместиться невозможно, следует вывести единственную строку -1.

Пример

Входные данные	Выходные данные
3 10	3 1 2 1 3 0

Задача «Гипер-каналы»

Входной файл – H.IN

Выходной файл – H.OUT

Ограничение времени: 5 секунд на тест

«Не плюй в телепорт: вылетит — не поймаешь!»

На зараженной радиацией планете некоторые точки соединены между собой гипер-каналами. Когда человек заходит в гипер-канал в одной точке, он мгновенно оказывается в другой. Все гипер-каналы двусторонние — то есть их можно использовать для перемещения в обоих направлениях (как из первой точки во вторую, так и из второй в первую).

К сожалению, гипер-каналы платные — каждый проход через гипер-канал стоит 10 у.е.

Перемещаться по поверхности планеты из одной точки в другую, не используя гипер-каналы, чревато для здоровья (радиация, однако!).

Напишите программу, которая определит, какой минимальной суммой у.е. должен располагать путешественник, чтобы добраться из одной точки в другую, не рискуя своим здоровьем.

Входные данные

Во входном файле записаны сначала два числа — начальные координаты расположения путешественника, затем еще два числа — координаты точки, куда ему надо попасть. Затем записано число N — количество гипер-каналов на планете ($0 \leq N \leq 500$). Затем идет N описаний гипер-каналов. Каждый гипер-канал описывается четверкой чисел. Первые два задают координаты одной из соединяемых гипер-каналом точек, последние два — координаты другой. Все координаты — целые числа, не превышающие по модулю 1000000.

Выходные данные

В выходной файл запишите одно число — минимальную сумму, которой должен располагать путешественник для достижения цели. Если, не рискуя здоровьем, он не сможет добраться до конечной точки, запишите в выходной файл число 171717 (столько стоит лечение лучевой болезни на этой планете).

Примеры

Примеры файла H.IN	Примеры файла H.OUT
10 10 -10 -10 1 2 2 3 3	171717
10 10 -10 -10 2 -10 -10 1 1 10 10 1 1	20
10 10 10 10 4 1 1 2 2 10 10 20 20 2 2 1 1 10 10 20 20	0

Словарь

Имя входного файла:	stdin
Имя выходного файла:	stdout
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем памяти:	64 мб

Вам дано N слов. Ваша задача - для каждого слова определить, сколькими способами его можно представить в виде конкатенации двух других из этого же набора(можно использовать одно слово два раза).

Формат входных данных

В первой строке входного файла записано число N ($1 \leq N \leq 10^5$).

Далее в N строках записано по одному слову, состоящих из строчных латинских букв, длиной не более 50-ти символов.

Формат выходных данных

Выполните N строк - в i -й строке выведите искомую величину для i -го слова.

input.txt	output.txt
5	0
x	1
xx	2
xxx	3
xxxx	4
xxxxx	

Задача «Куда идем мы с Пятачком?»

Входной файл – a.in

Выходной файл – a.out

Ограничение времени: 1 секунд на тест

«Кто ходит в гости по утрам, тот поступает мудро...»

Пятачок и Винни-Пух каждое утро ходят пить чай в гости к Кролику. Естественно, самым коротким путем.

К сожалению, однажды Винни-Пуху пришла в голову идея вырыть ловушку для Слонопотама. Самое обидное, что они с Пятачком ее даже вырыли. Поэтому теперь каждое утро, идя в гости к Кролику, они боятся в нее провалиться.

Напишите программу, которая посчитает длину самого короткого безопасного пути от домика Винни-Пуха до домика Кролика.

Ловушка для Слонопотама представляет собой яму абсолютно круглой формы. Путь является безопасным, если он не проходит по ловушке (но может проходить по ее границе).

Входные данные

Во входном файле записаны сначала координаты домика Винни-Пуха X_B Y_B , затем — координаты домика Кролика X_K Y_K , а затем — координаты центра и радиус ловушки X_L Y_L R_L . Все координаты — целые числа из диапазона от -32000 до 32000. Радиус ловушки — натуральное число, не превышающее 32000.

Домики Винни-Пуха и Кролика не могут находиться внутри ловушки, но могут находиться на ее границе.

Выходные данные

Выведите в выходной файл одно число — длину самого короткого безопасного пути от домика Винни-Пуха до домика Кролика с тремя знаками после точки.

Примеры

a.in	a.out
0 0 0 1 10 10 1	1.000
5 0 0 5 0 0 5	7.854
-5 0 5 0 0 0 3	11.861

Задача «Буратино»

Входной файл – b.in

Выходной файл – b.out

Ограничение времени: 1 секунда на тест

«Эх, дубинушка, ухнем!»

Папа Карло сменил работу: теперь он работает в мастерской, и целый рабочий день занимается тем, что забивает гвоздики. Чтобы ему было не скучно, у него в мастерской стоит постоянно работающий телевизор. К сожалению, производительность папы Карло напрямую зависит от его настроения, а оно, в свою очередь, — от того, что в данный момент показывают по телевизору. Правда, пока папа Карло забивает гвоздик, он не обращает ни малейшего внимания на телевизор, и поэтому скорость его работы зависит только от того, что показывали по телевизору в тот момент, когда он только начал забивать этот гвоздик. Забив очередной гвоздик, он обязательно мельком смотрит в телевизор (его настроение, естественно, меняется), и после этого он может либо сразу начать забивать следующий гвоздик, либо отдохнуть несколько секунд или даже минут, смотря телевизор.

Папа Карло начинает работу ровно в 9 часов. С 13 часов у него начинается обеденный перерыв. При этом если он незадолго до обеда хочет начать вбивать гвоздик, но понимает, что до перерыва он не закончит эту работу, то он и не начинает ее. Аналогично в 14 часов он вновь приступает к работе, а в 18 уходит домой. Это значит, что в 9:00:00 (аналогично, как и в 14:00:00) он уже может начать забивать гвоздик. Если, например, в 12:59:59 (аналогично, в 17:59:59) он хочет начать вбивать гвоздик, и на это у него уйдет 1 секунда, то он успевает вбить гвоздик до обеда (до окончания работы соответственно), а если 2 — то уже нет.

Известна программа телевизионных передач и то, как они влияют на папу Карло. Требуется составить график работы и маленьких перерывчиков папы Карло так, чтобы за рабочий день он вбил максимально возможное количество гвоздей.

Входные данные

Во входном файле записано расписание телевизионных передач с 9:00:00 до 18:00:00 в следующем формате. В первой строке число N — количество телевизионных передач в этот период ($1 \leq N \leq 32400$). В каждой из последующих N строк записано описание одной передачи: сначала время ее начала в формате ЧЧ:ММ:СС (ЧЧ — две цифры, задающие часы, ММ — две цифры, задающие минуты начала, СС — две цифры, задающие секунды начала). А затем через один или несколько пробелов число T_i — время в секундах, которое папа Карло будет тратить на забивание одного гвоздика, если он перед этим увидит по телевизору эту передачу ($1 \leq T_i \leq 32400$).

Передачи записаны в хронологическом порядке. Первая передача всегда начинается в 09:00:00. Можно считать, что последняя передача заканчивается в 18:00:00.

Выходные данные

В файл требуется вывести максимальное количество гвоздиков, которое папа Карло успеет вбить за рабочий день.

Примеры

b.in	b.out	Пояснение
2 09:00:00 3600 14:00:00 3600	8	Каждый час папа Карло вбивает по одному гвоздику
4 09:00:00 1800 12:59:31 10 13:45:23 1800 15:00:00 3600	14	Первую половину дня он вбивает по гвоздику за полчаса, но в 12:30:00 он не начинает вбивать гвоздики, а ждет 12:59:31, и успевает до обеда вбить 2 гвоздика. С 14 до 15 часов вбиваются 2 гвоздя, а затем по одному гвоздю в час.

Задача. Circuit

Входной файл – c.in

Выходной файл – c.out

Время тестирования – 0,1с

Дана цепочка, состоящая из $k = 4n$ звеньев. Причем $2n$ звеньев — золотые и $2n$ звеньев — серебряные. Два разбойника желают справедливо разделить серебро и золото данной цепочки. Составьте программу, которая находит минимальное количество разрезов цепочки, так, чтобы как серебро, так и золото можно было бы разделить между двумя разбойниками.

Входные данные

Программа читает сначала количество звеньев k , а далее k чисел 0 либо 1 (0 — серебряное звено, 1 — золотое). Все числа расположены в одной строке через пробелы.

Входные данные

Программа выводит количество разрезов и номера звеньев, между которыми они были сделаны. Первый разрез должен быть как можно ближе к началу цепи, при прочих равных второй должен быть сделан как можно ближе к первому и т.д. $N \leq 10000$.

c.in	c.out
8 0 1 0 0 1 1 0 1	2 1 2 5 6

Комментарий. Первый разрез выполняется между звеньями 1 и 2, второй — между звеньями 5 и 6. Один разбойник получает цепочки 0 и 101, второй — 1001.

Такси

Имя входного файла:

`taxi.in`

Имя выходного файла:

`taxi.out`

Максимальное время работы на одном тесте:

1 секунда

Максимальный объем используемой памяти:

64 мегабайта

На плане города, в котором Петя Булочкин работает таксистом, есть N ($1 \leq N \leq 1000$) площадей, пронумерованных от 1 до N (в дальнейшем, будем отождествлять площадь с ее номером). Некоторые площади соединены дорогами. Дороги эти настолько узкие, что на них введено одностороннее движение. По своему опыту Петя знает, что, выехав с какой-нибудь площади и проехав несколько дорог, невозможно приехать опять на эту же площадь. Нет двух дорог, соединяющих одну и ту же пару площадей. Дороги пересекаются только на площадях.

Один раз на площади A Петя подобрал пассажира, который попросил довезти его до площади B , причем по дороге остановиться на площадях C_1, C_2, \dots, C_k (на них пассажира будут ждать знакомые). Порядок посещения промежуточных площадей для пассажира не важен. «Сколько же денег взять с него?» - задумался Петя. Он решил взять с пассажира столько рублей, сколько существует различных маршрутов от площади A до площади B , проходящих через C_1, C_2, \dots, C_k .

По введенной информации о дорогах в городе, а также перекресткам $A, B, C_1, C_2, \dots, C_k$ требуется определить, сколько существует маршрутов от площади A до площади B , проходящих через C_1, C_2, \dots, C_k .

Формат входных данных

Первая строка ввода содержит числа N — количество площадей и M — количество дорог. В следующие M строках находятся описания дорог. Каждая из этих строк содержит по два числа — начальную и конечную площадь для дороги. Обратите внимание, что проехать по дороге в обратном направлении (от конечной площади к начальной) нельзя: движение одностороннее. В следующей строке будут числа A — начальная площадь, B — конечная площадь и K — количество промежуточных площадей. И, наконец, следующие K строк будут содержать числа C_1, C_2, \dots, C_k по одному в строке. Среди чисел $A, B, C_1, C_2, \dots, C_k$ нет одинаковых.

Формат выходных данных

Вы должны вывести единственное число S — количество маршрутов от площади A до площади B , проходящих через C_1, C_2, \dots, C_k . Заметим, что в случае отсутствия таких маршрутов S должно быть равно 0. Входные данные будут таковы, что S не превысит 2000000000.

Примеры

<code>taxi.in</code>	<code>taxi.out</code>
4 4	
1 2	
2 3	
1 4	
4 3	
1 3 1	
2	1

Сумма цифр

Имя входного файла:

sumnum.in

Имя выходного файла:

sumnum.out

Максимальное время работы на одном тесте:

0,1 секунды

Максимальный объем используемой памяти:

64 мегабайта

Заданы 2 натуральных числа N_1 и N_2 . Необходимо вычислить сумму цифр всех чисел от N_1 до N_2 (N_1, N_2 — натуральные числа ≤ 2000000000 , $N_1 \leq N_2$).

Формат входных данных

Во входном файле через пробел заданы числа N_1 и N_2 .

Формат выходных данных

Выходной файл должен содержать целое число, равное сумме цифр всех чисел от N_1 до N_2 .

Примеры

sumnum.in	sumnum.out
21 24	18

Таблица

Имя входного файла:	table.in
Имя выходного файла:	table.out
Максимальное время работы на одном тесте:	0,2 секунды
Максимальный объем используемой памяти:	64 мегабайта

Рассмотрим множество S , состоящее из k различных неотрицательных целых чисел A_1, A_2, \dots, A_k .

Напишите программу, которая строит некоторую таблицу T , удовлетворяющую следующим условиям:

1) В каждой клетке таблицы записано одно число, равное 0 или 1.

2) Сумма чисел в каждой строке таблицы принадлежит множеству S .

3) Сумма чисел в каждом столбце таблицы принадлежит множеству S .

4) Для каждого i ($1 \leq i \leq k$) найдется хотя бы одна строка или столбец таблицы, сумма чисел в котором равна A_i .

5) Количество строк в таблице не превышает 100.

6) Количество столбцов в таблице не превышает 100.

Формат входных данных

Первая строка входного файла содержит натуральное число k ($1 \leq k \leq 101$) — количество чисел в множестве M . Следующие k строк описывают множество S : i -я из этих строк содержит число A_i ($0 \leq A_i \leq 100$).

Формат выходных данных

Первая строка вывода должна содержать два целых числа N и M — количество строк и столбцов в построенной Вами таблице. Следующие N строк должны содержать найденную Вами таблицу. i -я из этих строк должна содержать ровно M чисел, совпадающих с числами, расположенными в i -й строке построенной Вами таблицы. Числа во всех строках вывода должны быть разделены одним или несколькими пробелами. Входные данные таковы, что существует хотя бы одна таблица, удовлетворяющая всем условиям.

Примеры

table.in	table.out
1 0	3 3 0 0 0 0 0 0 0 0 0
1 3	3 3 1 1 1 1 1 1 1 1 1
3 1 2 3	3 3 1 1 1 0 1 1 0 0 1

Задача «РЛС»

Входной файл – g.in

Выходной файл – g.out

Ограничение времени: 1 секунда на тест

«Задачи противовоздушной обороны: ...борьба с десантом на всем маршруте пролета, уничтожение вертолетов огневой поддержки, действующих из засады»

Радиолокационная станция (РЛС) состоит из нескольких передатчиков (не более 5). К сожалению, их нельзя ставить рядом — они друг для друга создают помехи. Каждый передатчик состоит из квадратных модулей, которые располагаются вплотную друг к другу.

Вам дана карта района, в котором расположена РЛС. Вся карта для удобства разбита на квадраты, и для каждого квадрата известно, располагается в нем какой-то из модулей одного из передатчиков РЛС или нет.

Требуется оградить забором (или несколькими заборами) минимально возможной суммарной длины все передатчики РЛС. Забор — это произвольная ломаная (ее элементы не обязаны идти по сторонам клеток). Одним забором могут быть огорожены сразу несколько передатчиков.

Входные данные

Во входном файле записаны два числа N и M , задающие размеры района, в котором расположена РЛС ($1 \leq N \leq 20$, $1 \leq M \leq 20$). Далее идут N строк, по M чисел в каждой, задающих карту района. Каждое из этих чисел 0 или 1 — 1 означает, что в этом квадрате находится один из модулей передатчика РЛС, а 0 — что в этом квадрате ничего ценного нет.

Общее количество передатчиков РЛС не превышает 5. Каждый передатчик — это связанная группа модулей (модули называются связанными, если они располагаются в квадратах карты, у которых есть общая граница, либо связаны через какие-то другие модули).

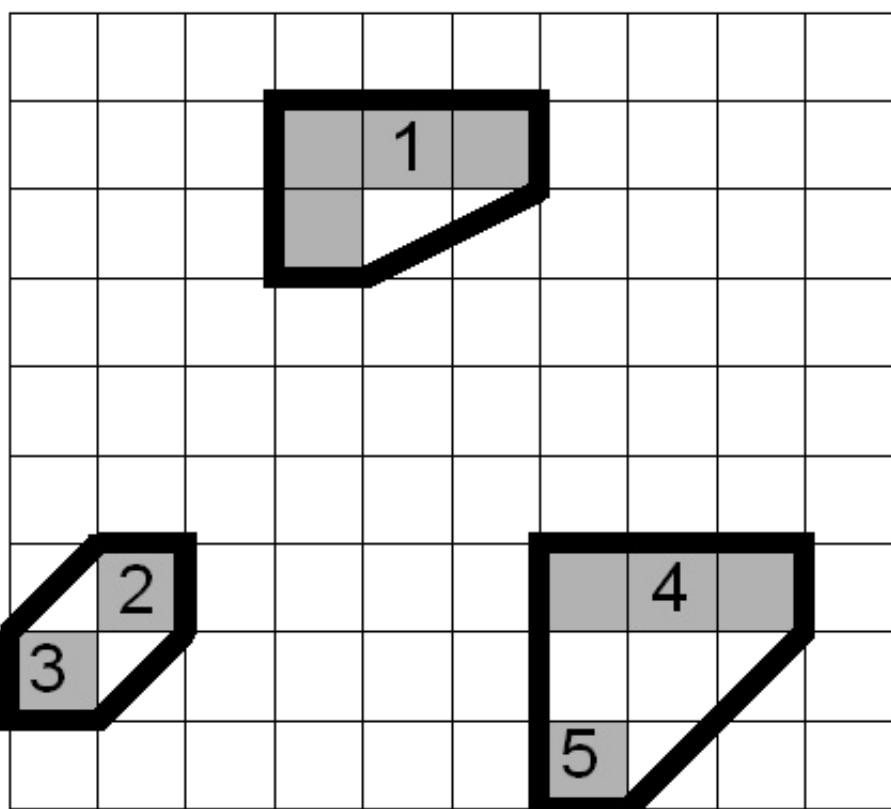
Ограничения на число модулей нет.

Выходные данные

В выходной файл выведите одно число — минимально возможную длину забора с тремя значащими цифрами после точки.

Примеры

В примере на рисунке РЛС состоит из 5 передатчиков. Длина забора вокруг передатчика 1 равна 9.236. Передатчики 2 и 3 окружены общим забором длины 6.828, и передатчики 4 и 5 окружены общим забором длины 10.828.



g.in	g.out
<pre>9 10 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0</pre>	26.893
<pre>3 3 0 0 0 0 1 0 0 0 0</pre>	4.000

Задача телепортация

Имя входного файла:

teleport.in

Имя выходного файла:

teleport.out

Максимальное время работы на одном тесте:

1 секунда

Максимальный объем используемой памяти:

64 мегабайта

Рассмотрим прямоугольную декартову систему координат O_{XY} (будем считать ее моделью Вселенной). Предположим, что в точке (x_1, y_1) находится космический корабль. За одну секунду из точки (x, y) он может телепортироваться в точки $(x + C, y + C)$, $(x + C, y - C)$, $(x - C, y + C)$, $(x - C, y - C)$, где C — произвольное натуральное число. Какое минимальное время понадобится кораблю для того, чтобы достичь точки (x_2, y_2) ?

Формат входных данных

Во входном файле находятся координаты пункта отправления и пункта назначения x_1, y_1, x_2, y_2 — целые числа, по модулю не превосходящие 10^9 .

Формат выходных данных

Если возможно достичь точку (x_2, y_2) из точки (x_1, y_1) , то выведите минимальное время в секундах, необходимое для этого. В противном случае выведите число 0.

Примеры

teleport.in	teleport.out
0 0	
0 2	2

Интересная игра

Имя входного файла:	game.in
Имя выходного файла:	game.out
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	64 мегабайта

Рассмотрим следующую интересную игру для двух игроков. Для этой игры необходима таблица из 2-х строк и N столбцов, в клетках которой записаны натуральные числа, следующего вида:

A_1	A_2	A_N
B_1	B_2	B_N

Игроки делают ходы по очереди. Начинает игру 1-й игрок.

За один ход 1-й игрок выполняет следующие два действия:

1) Выбирает произвольный столбец (к примеру, j -й), который еще ни разу не был выбран одним из игроков на предыдущих ходах.

2) Прибавляет к своим очкам число A_j .

За один ход 2-й игрок выполняет следующие два действия:

3) Выбирает произвольный столбец (к примеру, j -й), который еще ни разу не был выбран одним из игроков на предыдущих ходах.

4) Прибавляет к своим очкам число B_j .

Игра заканчивается, когда какой-либо из игроков не сможет сделать ход (по той причине, что все столбцы уже были выбраны). Изначально, у каждого из игроков есть 0 очков.

После того, как игра закончилась, происходит взаиморасчет между игроками. К примеру, 1-й игрок набрал S_1 очков, а 2-й игрок — S_2 очков. В случае, когда $S_1 > S_2$, 2-й игрок отдает 1-му игроку $S_1 - S_2$ УДЕ (условных денежных единиц). В противном случае, 1-й игрок отдает 2-му игроку $S_2 - S_1$ УДЕ. С этих позиций, целью 1-го игрока является максимизация величины $S_1 - S_2$, а целью 2-го игрока — максимизация $S_2 - S_1$.

Назовем стоимостью игры величину $S_1 - S_2$ при оптимальной игре обоих игроков. Напишите программу, которая определяет стоимость игры.

Формат входных данных

Первая строка ввода содержит натуральное число N ($1 \leq N \leq 300000$) — количество столбцов в таблице. Следующие N строк описывают числа в столбцах таблицы. i -я из этих строк содержит два натуральных числа A_i и B_i ($1 \leq A_i, B_i \leq 3000$), разделенные одним пробелом.

Формат выходных данных

Единственная строка вывода должна содержать одно целое число — стоимость игры.

Примеры

game.in	game.out
1 1 1	1
2 1 1 1 1	0
3 1 2 3 4 5 6	2

Задача Метро

Имя входного файла:

j.in

Имя выходного файла:

j.out

Максимальное время работы на одном тесте:

1 секунда

Метрополитен состоит из нескольких линий метро. Все станции метро в городе пронумерованы натуральными числами от 1 до N . На каждой линии расположены несколько станций. Если одна и та же станция расположена сразу на нескольких линиях, то она является станцией пересадки и на этой станции можно пересесть с любой линии, которая через нее проходит, на любую другую (опять же проходящую через нее).

Напишите программу, которая по данному вам описанию метрополитена определит, с каким минимальным числом пересадок можно добраться со станции А на станцию В. Если данный метрополитен не соединяет все линии в одну систему, то может так получиться, что со станции А на станцию В добраться невозможно, в этом случае ваша программа должна это определить.

Формат входных данных

Во входном файле записано сначала число N — количество станций метро в городе ($2 \leq N \leq 200$). Далее записано число M — количество линий метро ($1 \leq M \leq 20$). Далее идет описание M линий. Описание каждой линии состоит из числа P_i — количества станций на этой линии ($2 \leq P_i \leq 50$) и P_i чисел, задающих номера станций, через которые проходит линия (ни через какую станцию линия не проходит дважды).

В конце файла записаны два различных числа А — номер начальной станции, и В — номер станции, на которую нам нужно попасть. При этом если через станцию А проходит несколько линий, то мы можем спуститься на любую из них. Так же если через станцию В проходит несколько линий, то нам не важно, по какой линии мы приедем.

Формат выходных данных

В выходной файл выведите минимальное количество пересадок, которое нам понадобится. Если добраться со станции А на станцию В невозможно, выведите в выходной файл одно число -1 (минус один).

j.in	j.out
5 2 4 1 2 3 4 2 5 3 3 1	0
5 5 2 1 2 2 1 3 2 2 3 2 3 4 2 4 5 1 5	2
10 2 6 1 3 5 7 4 9 6 2 4 6 8 10 7 3 8	1
4 2 2 1 2 2 3 4 1 3	-1

Задача A. Clear World and Brothers

Имя входного файла:	clear.in
Имя выходного файла:	clear.out
Ограничение по времени:	4 секунды
Ограничение по памяти:	64 мегабайта

Наконец в деревнях Виллорибо и Виллобаджо закончились праздники. Перемыта вся посуда! Этот процесс прошел так быстро и непринуждённо, что братьями Карлионе было решено открыть сеть агентств «Clear World and Brothers», специализирующихся на профессиональном мытье посуды. В области Новая Берляндия, где и находятся знаменитые деревни, всего N деревень. Система координат введена так, что Виллорибо имеет координаты $(x_1, 0)$, а Виллобаджо — $(x_2, 0)$. Координаты всех деревень целые числа не превосходящие по модулю 10^6 . Вы работаете на мистера Берлионе старшего и ваша задача найти оптимальное расположение для регионального отделения «Clear World and Brothers», то есть сумма расстояний от агентства до всех деревень должна быть наименьшей и агентство обязательно должно располагаться на прямолинейном шоссе Виллорибо-Виллобаджо (возможно расположение не только внутри, но и на границе отрезка).

Формат входного файла

В первой строке входного файла записано натуральное число N ($2 \leq N \leq 15000$). Далее в N строках записаны пары координат всех вершин Виллорибо и Виллобаджо первая и вторая деревня соответственно. Возможно, что сколько-то деревень расположены так близко, что их координаты совпадают.

Формат выходного файла

Выведите абсциссу оптимального расположения агентства. Разрешается абсолютная или относительная погрешность не более 10^{-7} .

Пример

clear.in	clear.out
4 -10 0 10 0 3 1 1 -1	2.0000

Задача В. Годовой баланс

Имя входного файла: **balance.in**
Имя выходного файла: **balance.out**
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В конторе «Рога и Копыта» подходит время подведения годового баланса. В бухгалтерию поступили сведения о том, что, согласно документам, суммарный расход составил a рублей, а суммарный приход — b рублей. Поскольку с реальным положением дел эти цифры все равно не имеют ничего общего, бухгалтер решил реализовать следующую свою идею. Как известно, при наборе чисел на компьютере люди часто вводят цифры в неправильном порядке. Поэтому бухгалтер хочет найти такой способ переставить цифры в числах a и b , чтобы в результате разность $a - b$ (и, соответственно, количество денег, которые он положит к себе в карман), была максимальна, а в случае чего можно будет сослаться на ошибку секретаря. При этом нельзя забывать о знаке чисел и о том, что ноль не может быть первой цифрой числа. Напишите программу, которая поможет бухгалтеру.

Формат входного файла

В входном файле задаются два целых числа — a и b на отдельных строках. По модулю числа не превосходят $2 \cdot 10^9$.

Формат выходного файла

В выходной файл выведите одно целое число — наибольшую разность чисел, первое из которых может быть получено перестановкой цифр a , а второе — перестановкой цифр b .

Примеры

balance.in	balance.out
18	71
10	
1	33
-23	

Разложи число

Максимальное время работы: 100 мс.
Максимальный объем памяти: 256 МВ

Условия

Дано число N . Найдите все разложения этого числа в виде $a^x + b^y = N$. $2 \leq b \leq a$, если $a = b$, то $2 \leq y \leq x$.

Входные данные

Программе подается одно число N ($1 \leq N \leq 2 * 10^9$).

Выходные данные

В первой строке запишите число K - число разложений. Далее выведите K строк - число a, x, b, y разложения. Все разложения должны быть отсортированы сначала по a , при равенстве a по b , при равенстве b по x .

Примеры

stdin	stdout
200	2 10 2 10 2 14 2 2 2

Задача D. Автобусные маршруты

Входной файл: newbus.in

Выходной файл: newbus.out

Максимальное время работы на одном тесте: 1 секунда

Для решения транспортной проблемы в некотором городе до недавнего времени использовались N ($N \leq 100000$) автобусных маршрутов. Каждый маршрут начинался на одной из M ($M \leq 100000$) площадей и там же заканчивался. В процессе проезда по маршруту автобус мог несколько раз проезжать одну и ту же площадь, и даже мог проезжать более одного раза по одной и той же улице в одном и том же направлении.

В определенный момент местные власти решили сократить количество автобусных маршрутов в городе до одного. По их мнению, должен был остаться лишь один маршрут, который проходил бы по всем улицам, по которым раньше проходили автобусные маршруты, причем в том же направлении (но не обязательно в том же порядке). Если по каким-либо улицам автобусы ездили в обоих направлениях, то и новый маршрут должен проходить по этим улицам в обоих направлениях. По тем улицам и в тех направлениях, по которым раньше автобусы не ездили, новый маршрут проходить не должен. Однако так как контролеров увольнять нельзя, власти решили, что по каждой улице в каждом направлении новый маршрут должен проходить столько раз, сколько по ней проходили все старые маршруты, вместе взятые.

Требуется написать программу, которая для заданных исходных данных определяет требуемый местным властям автобусный маршрут.

Формат входных данных:

Входной файл состоит из следующей последовательности строк. Первая строка содержит число N – количество автобусных маршрутов, M – количество площадей. Каждая из последующих N строк служит для описания соответствующего автобусного маршрута и содержит сначала число k ($k \leq 100000$), определяющее количество элементов маршрута, а затем $k+1$ чисел, задающих номера площадей, которые последовательно проезжает автобус на этом маршруте. Общая длина маршрутов не более 100000 улиц. При описании маршрута всегда задаются номера первой и последней площади маршрута, причем они всегда совпадают.

Формат выходных данных:

Выходной файл должен содержать либо описание нового маршрута в том же формате, что используется во входном файле, либо одно число 0, если организовать требуемый маршрут не удастся.

Пример файла входных данных:

2 5

6 1 2 3 4 3 2 1

2 4 5 4

Пример файла выходных данных (для приведенного выше входного файла):

9 1 2 3 4 5 4 3 2 1

Задача Е. Отгадай число

Имя входного файла: **guess.in**
Имя выходного файла: **guess.out**
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Двое играют в игру. Первый игрок (ведущий) загадал число от 1 до N . Второй хочет отгадать это число. Для достижения цели он задает вопросы. Каждый вопрос имеет вид: «Содержится ли загаданное число в наборе $S?$ », где S – произвольный набор чисел от 1 до N . В случае положительного ответа второй игрок платит две конфеты ведущему, в противном случае – одну.

Какое наименьшее количество конфет должен иметь второй игрок, чтобы наверняка суметь отгадать число?

Формат входного файла

В первой строке файлы записано целое число N ($1 \leq N \leq 10^5$).

Формат выходного файла

Выведите искомое количество конфет.

Пример

guess.in	guess.out
6	5
2	2

Задача F. Треугольник

Имя входного файла: triangle.in
Имя выходного файла: triangle.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

После окончания многолетней войны короли-победители решили разделить между собой захваченную территорию. Для того, чтобы избежать лишних споров, короли решили делить территорию следующим образом.

Каждому королю был предоставлен набор, состоящий из n отрезков l_1, \dots, l_n . Королям разрешается присоединять к их королевствам территории, имеющую форму треугольника, составленного из имеющихся отрезков.

Помогите королю вашего королевства максимизировать площадь треугольника.

Формат входного файла

На первой строке задано число n ($3 \leq n \leq 100000$) — количество отрезков в наборе. В следующих n строках заданы длины отрезков в следующем формате.

Сначала указывается целое положительное число l длина отрезка, а затем единица измерения (m, km, mile, uin, kairi, zhang, sen).

Напомним, что:

$$\begin{aligned}1\text{mile} &= 1609\text{m} \\1\text{km} &= 1000\text{m} \\1\text{uin} &= 33\text{m} \\1\text{kairi} &= 1852\text{m} \\1\text{zhang} &= 3\text{m} \\1\text{sen} &= 38\text{m}\end{aligned}$$

Длины всех отрезков не превосходят десяти километров. Гарантируется, что из заданных отрезков можно выбрать три отрезка так, что из них можно составить треугольник.

Формат выходного файла

Выведите в выходной файл максимальную площадь (в таланг вахах) с точностью 10^{-6} и номера трех отрезков в любом порядке (если ответов несколько, выведите любой). Помните, что один таланг вах равен четырем квадратным метрам.

Примеры

triangle.in	triangle.out
7 1mile 1km 100m 20uin 1kairi 300zhang 40sen	291590.0233624191 7 1 5

Задача G. Спираль

Имя входного файла:	<code>spiral.in</code>
Имя выходного файла:	<code>spiral.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В одной из предыдущих задач уже описывался язык «Лого» для рисования простейших фигур. Теперь вам предлагается написать на этом языке программу, изображающую контур спирали.

Спираль — это множество клеток, которое строится по следующим правилам. Рассмотрим прямоугольник, состоящий из $m \times n$ единичных клеток. Закрасим некоторые из его клеток в следующей последовательности.

Начнем красить левый столбец снизу вверх. Когда продвижение невозможно, то есть либо следующая клетка лежит вне прямоугольника, либо соприкасается с закрашенной ранее, направление движения поворачивается по часовой стрелке на 90 градусов. Если и после этого поворота дальнейшее движение невозможно, процесс заканчивается.

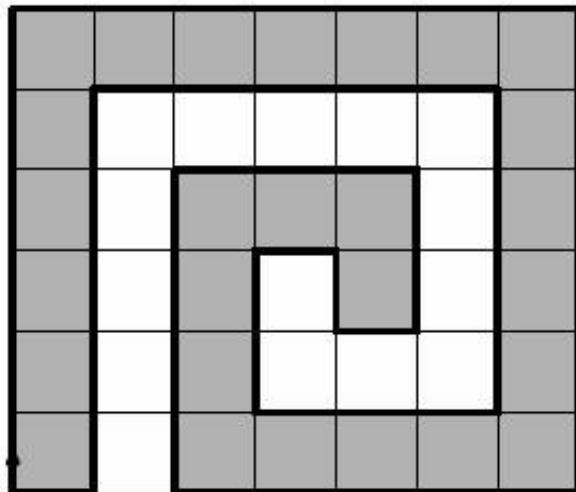
Множество закрашенных таким образом клеток и образует спираль.

Формат входного файла

Во входном файле записаны два целых числа m и n ($1 \leq m, n \leq 20\,000$).

Формат выходного файла

В выходной файл выведите последовательность команд, в результате выполнения которой будет нарисован контур ломаной. Напомним формат, в котором следует выводить результат. Первая строка должна содержать k — количество команд. Последующие k строк задают либо длину отрезка d , который надо провести, в формате `f d`, либо направление поворота — `l` для поворота налево и `r` для поворота направо. Команды рисования отрезка и поворота должны чередоваться, начинаясь и заканчиваясь командой рисования отрезка. Считайте, что начало первого отрезка совпадает с левым нижним углом прямоугольника. При этом контур должен быть выведен в направлении обхода по часовой стрелке, и никакая его часть ненулевой длины не должна быть нарисована дважды.



Примеры

spiral.in	spiral.out
6 7	31 f 6 r f 7 r f 6 r f 5 r f 4 r f 3 r f 2 r f 1 r f 1 l f 1 l f 2 l f 3 l f 4 l f 5 l f 5 r f 1

Задача Н. Башни

Имя входного файла:	towers.in
Имя выходного файла:	towers.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Для того, чтобы защититься от некоторых соседей, король решил построить стену, имеющую форму отрезка. С некоторыми соседями король находится в хороших отношениях, а некоторым готовится объявить войну. Король решил не загораживаться от друзей очень высокой стеной. Однако, стена, отделяющая его от врагов, должна быть достаточно высокой. Было решено, что для наблюдения за прилежащей территорией нужно построить башни. При этом, на участках между башнями высота стен должна изменяться равномерно.

После того, как стена и башни были построены, король заметил, что башни могут быть использованы для наблюдения за состоянием других башен. Однако, некоторые башни оказались очень высокими и загородили другие.

Для каждой башни король попросил вас выяснить, сколько других башен из нее видно.

Формат входного файла

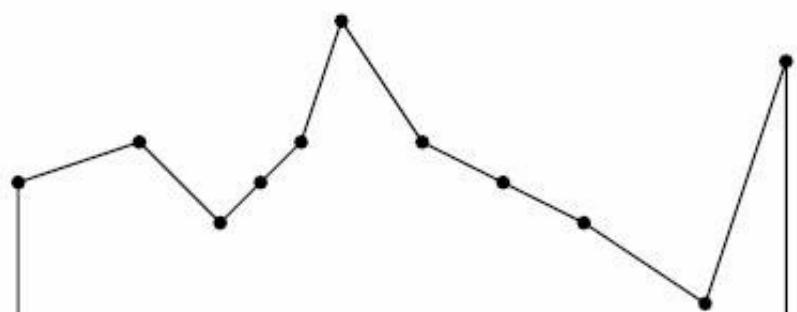
В первой строке входного файла находится n ($2 \leq n \leq 2000$) — количество башен стены. В следующих n строках находятся натуральные числа x_i и h_i ($0 \leq x_i \leq 100000$, $1 \leq h_i \leq 10000$) — координата и высота i -ой башни. Все x_i различны.

Формат выходного файла

Выходной файл должен содержать n строк. В i -ой строке выведите количество башен, которые видно из башни номер i .

Примеры

towers.in	towers.out
11	2
0 4	5
3 5	4
5 3	4
6 4	4
7 5	10
8 8	4
10 5	4
12 4	5
14 3	3
17 1	5
19 7	



Задача I. Чашечные весы

Имя входного файла: scales.in
Имя выходного файла: scales.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Имеются N гирь разной массы и чашечные весы. За ход игрок докладывает 1 гирю на любую чаша весов. После хода перевешивает либо правая, либо левая чаша (ситуаций равенства весов возникать не должно). Вы знаете, какая чаша перевешивает после каждого хода. Ваша задача указать для каждого хода, какую гирю надо класть на какую чашу, что бы результат после каждого хода соответствовал исходным данным.

Формат входного файла

В первой строке находятся числа N и K — количество гирь и ходов соответственно ($K \leq N$). Во второй строке содержатся N чисел — массы гирь, а в третьей — строка из K символов 'L' или 'R', где 'L' значит, что перевесила левая чаша, а 'R' — правая. Например, строка 'LRR' означает, что в первый ход перевесила левая чаша, а во второй и третий — правая. Все числа во входном файле натуральные и не превосходят 1000.

Формат выходного файла

Если задача имеет решение, то выходной файл содержит $K + 1$ строку. В первой необходимо вывести Solution:, а в следующих K строках записано сначала число — номер выкладываемой гири на соответствующем ходу, а затем через пробел символ 'L' или 'R' (на какую чашу эта гиря кладётся). Необходимо вывести любое из решений. Если решения нет, вывести No solution.

Пример

scales.in	scales.out
4 3	Solution:
10 2 7 1	3 L
LRR	1 R
	4 R

Задача J. Слова

Имя входного файла:	<code>words.in</code>
Имя выходного файла:	<code>words.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Для шифрования слов с ними можно производить множество различных операций. Например, интересна такая операция: первые несколько букв заданного слова приписываются к его концу в обратном порядке, после чего удаляются из начала слова. При этом слово $a_1a_2\dots a_k a_{k+1}\dots a_n$ переходит в слово $a_{k+1}\dots a_n a_k a_{k-1}\dots a_1$ (число k выбирается в диапазоне от 0 до n).

Для двух заданных слов требуется определить, можно ли применением описанной операции преобразовать первое слово во второе.

Формат входного файла

Входной файл состоит из двух строк — на первой из них записано исходное слово, а во второй — предполагаемый результат.

Формат выходного файла

В первой строке выходного файла выведите “Yes”, если преобразование возможно и “No”, если нет. В случае положительного ответа во второй строке выведите k — длину перемещаемой части исходного слова k (из всех таких k выберите минимальный).

Примеры

<code>words.in</code>	<code>words.out</code>
<code>wpwdwprw</code>	<code>Yes</code>
<code>wdwprwprw</code>	<code>2</code>
<code>dW0ddWd</code>	<code>No</code>
<code>d0dWdWd</code>	

Задача А. Ближайшее число - 1

Входной файл: input.txt

Выходной файл: output.txt

Ограничение времени на тест: 0,1 сек

Ограничение памяти на тест: 200 Мб

Дан массив A , состоящий из N неотрицательных целых чисел.

Назовём правым (левым) соседом нулевого элемента ближайший к нему справа (слева) ненулевой элемент.

Требуется построить массив B , который получается из массива A заменой каждого нулевого элемента на его ближайшего соседа в массиве A . Если оба соседа отсутствуют, либо расстояния до них равны, замена не производится (элемент остаётся нулевым).

Формат входного файла

Входной файл содержит число N , за которым следуют N целых чисел — элементы массива A .

Формат выходного файла

Выходной файл должен содержать N целых чисел — элементы массива B .

Ограничения

$1 \leq N \leq 10000$, $0 \leq A_i \leq 10000$

Примеры тестов

№ Входной файл Выходной файл

1 5
0 0 1 0 2 1 1 1 0 2

2 4
8 0 0 6 8 8 6 6

В. Конкатенация

Входной файл: input.txt

Ограничение времени 2 сек на тест:

Выходной файл: output.txt

Конкатенация строк — это операция, приписывающая одну строку к другой. Например, из строк "foo" и "bar" получается строка "foobar".

Нам дано N строк a_1, \dots, a_N и M операций конкатенации, описанных парами $(p_1, q_1), \dots, (p_M, q_M)$. Каждая пара (p_j, q_j) означает, что новая строка a_{N+j} — есть конкатенация строк a_{pj} и a_{qj} , где $1 \leq p_j, q_j < N+j$. То есть вновь образованные строки добавляются к уже имеющимся.

Например, строки "a" и "bc" и пары $(1, 2), (3, 3)$ означают, что будут получены строки "abc" и "abcaabc".

Ваша программа должна выдать подстроку строки a_{N+M} с позиции L до позиции R , позиции пронумерованы с 1.

Формат входного файла

В первой строке входного файла находятся 4 натуральных числа $N M L R$. $1 \leq N \leq 1000$, $1 \leq M \leq 1000$, $R - L + 1 \leq 1000$.

Следующие N строк содержат исходные строки a_i — по одной в строке. Каждая из следующих M строк содержит пару натуральных чисел $p_j q_j$. $1 \leq \text{length}(a_i) \leq 1000$ for $i = 1, N$, $1 \leq \text{length}(a_{N+j}) \leq 2 \times 10^9$ для $j = 1, M$, $1 \leq L \leq R \leq \text{length}(a_{N+M})$,

Формат выходного файла

Выходной файл должен содержать $R - L + 1$ искомых символов.

Примеры

No.	Входной файл	Выходной файл
	1 2 3 6	

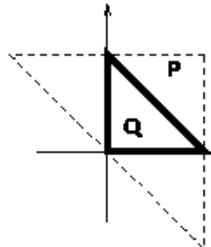
1	a bc 1 2 3 3	cabc
2	1 5 1 32 z 1 1 2 2 3 3 4 4 5 5	zzzzzzzzzzzzzzzzzzzzzzzzzzzzzz

Задача С. Нечётный N-угольник

Исходный файл: INPUT.TXT

Выходной файл: OUTPUT.TXT

Время тестирования: 1 секунда на тест



Выпуклый N -угольник **P** преобразуется в N -угольник **Q** путём замены середин сторон исходного многоугольника **P** на вершины многоугольника **Q**. Требуется по выпуклому N -угольнику **Q**, заданному координатами вершин, восстановить координаты вершин исходного N -угольника **P**.

Входные и выходные данные

Входной файл содержит нечётное число вершин N ($3 \leq N \leq 999$), за которым следуют целочисленные координаты x, y ; вершин многоугольника **Q**, перечисленные в порядке обхода по часовой стрелке. Значения координат находятся в диапазоне от -20000 до 20000 .

Все числа во входном файле целые и разделены произвольным количеством пробелов и/или символов перевода строки.

В выходном файле должны содержаться координаты вершин N -угольника **P**, перечисленные в порядке обхода по часовой стрелке. При этом первая и вторая вершина должны образовывать сторону, на которой лежит первая вершина N -угольника **Q**.

Примеры

INPUT.TXT	OUTPUT.TXT
3 0 0 0 1 1 0	1 -1 -1 1 1 1

Задача D. Билеты

Входной файл: input.txt

Ограничение времени на тест: 2 сек

Выходной файл: output.txt

Ограничение памяти на тест: 200 Мб

Петя любит интересные автобусные билеты.

Билет с номером длины $2N$ называется интересным, если произведение первых N цифр равна произведение последних N цифр.

Петя нашел в кармане использованный билет. К сожалению он оказался пробит компостером, и некоторые цифры невозможно распознать. Ему интересно, мог ли этот билет быть интересным 😊.

Более того, он хотел бы знать, сколько интересных и неинтересных билетов могли быть пробиты.

Формат входного файла

Первая строка входного файла содержит число N ($1 \leq N \leq 18$).

В следующей строке содержится строка, представляющая пробитый номер.

Пробитые цифры заменены на знаки ?.

Формат выходного файла

В первой строке выведите количество интересных билетов. Во второй строке – количество неинтересных.

Примеры

INPUT.TXT	OUTPUT.TXT
2	4
2??3	96
3	46
2?6??3	954

Задача E. OLE

Входной файл: input.txt

Ограничение времени на тест: 2 сек

Выходной файл: output.txt

Ограничение памяти на тест: 200 Мб

Текстовый редактор OLE (One-Line Editor) работает с текстом, состоящим ровно из одной строки строчных латинских букв. Редактор поддерживает следующие команды, длиной в один символ каждая:

- **a ... z** — вставить справа от текущей позиции курсора указанный символ, переместить курсор на один символ вправо
- **L** — переместить курсор на 1 символ влево
- **R** — переместить курсор на 1 символ вправо
- **X** — удалить символ справа от позиции курсора

Команды, пытающиеся переместить курсор за пределы строки или удалить символ справа от последнего символа строки, игнорируются редактором.

Требуется по данному начальному состоянию строки, начальной позиции курсора и последовательности команд определить результат работы редактора.

Формат входного файла

Входной файл состоит из 3 строк. В первой строке содержится позиция курсора p , (0 - курсор перед первым символом, 1 - после первого перед вторым, и т.д.) во второй строке — начальное состояние строки редактора, в третьей — последовательность команд.

Формат выходного файла

Выходной файл должен содержать строку, полученную в результате выполнения команд.

Ограничения

Длина исходной строки находится в диапазоне от 1 до 1000000 символов. Длина строки команд находится в диапазоне от 1 до 100000 символов.

$0 \leq p \leq$ длина исходной строки.

Примеры тестов

№ Входной файл Выходной файл

1	1	
1	abc	adc
	deLxx	
0		
2	aa	xbbaa
	bbLLx	

Задача F. Дружба и кефир

Входной файл: input.txt
Выходной файл: output.txt

Ограничение времени на тест: 10 сек
Ограничение памяти на тест: 2 Мб

Любимым напитком большинства учеников К-й средней школы является кефир. Однако, в последнее время в результате активной рекламной кампании всё больше учеников переходят на новый газированный напиток "UnhealthyCola". Этот факт беспокоит школьную администрацию. В результате социологического опроса учащихся выяснилось, что у каждого школьника в классе есть ближайшие друзья, с мнением которых он считается. Если больше половины ближайших друзей школьника уже пьют UnhealthyCola, то школьник поддаётся их дурному влиянию и, в свою очередь, влияет на оставшихся друзей. К сожалению, никакое количество друзей не способно переубедить школьника перейти обратно с UnhealthyCola на кефир.

На педагогическом совете было решено, что распространение UnhealthyCola можно было бы сдержать, если бы больше школьников дружили между собой.

Требуется подружить двух учеников школы таким образом, чтобы максимизировать количество школьников, которые станут верны кефиру.

Формат входного файла

Входной файл содержит число школьников N , за которым идёт N чисел c_i , где $c_i=0$, если i -й ученик пьёт кефир, и $c_i=1$, если он пьёт UnhealthyCola. Далее число пар друзей M , за которым идут M пар чисел $a_i b_i$, означающих, что школьники a_i и b_i — друзья.

Формат выходного файла

В выходной файл должны быть выведены два числа — ученики, которых следует подружить. Если решений несколько, вывести любое из них. Если никакая пара новых друзей не улучшит результат, следует вывести число -1 .

Ограничения

$2 \leq N \leq 500$, $0 \leq M \leq 1000$, $1 \leq a_i, b_i \leq N$

Примеры тестов

№	Входной файл	Выходной файл
1	3 1 0 0 2 1 2 1 3	2 3
2	9 1 0 0 0 0 1 1 1 0 7 1 2 2 3 3 4 4 5 6 2 7 3 8 4	9 2

Задача G. Король и склеп

Входной файл: input.txt

Ограничение времени на тест: 2 сек

Выходной файл: output.txt

Ограничение памяти на тест: 200 Мб

Король решил, что он хочет иметь новый дизайн семейного склепа. Он должен состоять из нескольких секций — каждая представлять собой квадрат. Все секции должны иметь разное число могил. Длины секций должна быть последовательностью последовательных положительных чисел. Секция длиной s содержит s^2 могил. Требуется для общего числа могил N определить — сколько различных дизайнов склепа можно сделать.

Формат входного файла

Входной файл содержит число n — количество могил в склепе ($1 \leq n \leq 10^{14}$).

Формат выходного файла

Первая строка выходного файла должна содержать число k — число возможных дизайнов склепа. Следующие k строк должны содержать описание соответствующих склепов. Каждое описание должно начинаться с количества секций в склепе L . Затем должны идти L последовательных чисел — длины секций.

Примеры

INPUT .TXT	OUTPUT .TXT
29	1

	3 2 3 4
2030	2
	4 21 22 23 24
	3 25 26 27

Задача N. Highway

Входной файл:	input.txt	Ограничение времени на тест:	2 сек
Выходной файл:	output.txt	Ограничение памяти на тест:	200 Мб

В некоторой стране N городов и все они расположены вдоль хайвея. Хайвей – это прямая линия, начинающаяся в первом городе и идущая через второй, третий и т.д. города, заканчивающаяся в N -м городе. i -й город расположен на расстоянии X_i миль от первого. Проблема одна – хайвей является односторонней дорогой. Поэтому возвращаться жители должны по сельским дорогам.

Новый президент решил построить два новых хайвея, опять же односторонних, но направленных в другую сторону, чтобы стало возможным проехать из любого города в любой другой по хайвеям. Каждый из них должен соединять два различных города и не проходить через другие города. Пары городов тоже должны быть различными. Стоимость дорог, пропорциональна расстоянию между ними. Стоимость строительства должна быть минимально возможной.

Формат входного файла

Первая строка входного файла содержит число N ($2 \leq N \leq 50000$) – количество городов. Следующая строка содержит $N-1$ чисел – расстояния X_2, X_3, \dots, X_n до первого города. ($1 \leq X_2 < X_3 < \dots < X_N \leq 10^9$)

Формат выходного файла

Если удовлетворяющие всем требованиям дороги построить невозможно, то выведите 0, в противном случае сначала выведите суммарную минимально возможную длину дорог, затем номера двух пар городов, которые надо соединить.

Примеры

INPUT .TXT	OUTPUT .TXT
4	12
3 5 10	3 1 4 2

Задача I. Невозможное возможно

Входной файл:	input.txt	Ограничение времени на тест:	2 сек
Выходной файл:	output.txt	Ограничение памяти на тест:	200 Мб

Дана прямоугольная таблица размером $n \times 3$. Известна сумма чисел в каждой строке и в каждом столбце. Вычислите количество таких таблиц по модулю 10^{17} .

Формат входного файла

Первая строка входного файла содержит числа N ($0 \leq N \leq 125$), $c1, c2$ и $c3$ – рядов и суммы чисел в каждом из трех столбцов. Следующие N строк содержат суммы в каждой из строк. Все числа не отрицательные и не превосходят 125.

Формат выходного файла

Выполните требуемое число по модулю 10^{17} .

Примеры

INPUT .TXT	OUTPUT .TXT
3 1 2 3	0
2	
~	

4

2 1 1 1

1

2

3

Задача В. Bottom (Дно моря)

Входной файл: input.txt

Ограничение времени на тест: 1 сек

Выходной файл: output.txt

Ограничение памяти на тест: 256 МБ

Институт «Нирсия», занимающийся подводными исследованиями, решил сделать обзор морского дна.

Область, на которой происходит обзор, представлена как прямоугольная сетка. Ось X проходит с запада на восток, а ось Y с юга на север.

Робот – подводный аппарат, который используется на выполнения данного обзора. У робота есть программа, которая позволяет ему покрыть прямоугольную часть морского дна, определенную координатами своих юго-западных и северо-восточных углов.

Первоначально планировалось, чтобы робот погрузился один раз и обследовал одну прямоугольную область с координатами $(a_x, a_y) - (b_x, b_y)$.

Но вдруг, появился представитель Министерства обороны и заявил, что прямоугольная область с координатами $(c_x, c_y) - (d_x, d_y)$ содержит секретные военные объекты, и не должна быть посещена роботом.

Поэтому было принято решение разделить площадь съемки на меньшие прямоугольники и выполнять отдельное погружение для каждого из них. Прямоугольники не должны пересекаться друг с другом, а так же должны обхватывать всю зону исследования, кроме секретной зоны.

Вашей программе следует найти такое разделение прямоугольника, чтобы количество погружений роботом было минимально.

Формат входного файла

Входной файл содержит 8 целых чисел $a_x \ a_y \ b_x \ b_y \ c_x \ c_y \ d_x \ d_y$.

Формат выходного файла

Выходной файл должен содержать минимальное количество погружений робота M в 1 строке, в каждой следующей из M строк должно следовать 4 целых числа $x_i \ y_i \ u_i \ v_i$ – координаты юго-восточного и северо-западного углов, которые должны быть охвачены i-ым погружением. Если существует несколько оптимальных решений, можно вывести любое из них.

Ограничения

$$-10^9 \leq a_x < b_x \leq 10^9, -10^9 \leq a_y < b_y \leq 10^9, -10^9 \leq c_x < d_x \leq 10^9, -10^9 \leq c_y < d_y \leq 10^9$$

Примеры тестов

№	Входной файл	Выходной файл
1	-10 -10 10 10 0 0 20 20	2 -10 -10 10 0 -10 0 0 10
2	1 2 3 4 10 11 12 13	1 1 2 3 4

3	-1 -1 1 1 -2 -2 2 2	0
---	------------------------	---

Задача С. Cthulhu (Ктулху)

Входной файл: input.txt

Выходной файл: output.txt

Ограничение времени на тест: 1 сек

Ограничение памяти на тест: 256 МБ

Ученые института «Нирсия» подводной мифологии исследуют мифическое существо под названием Ктулху.



Это существо славится своим производством большого потомства, известным как “потомство Ктулху”. У каждого потомка маленький круглый корпус с N радиально выступающими придатками. Каждый придаток - это щупальца или коготь.

Два потомка называются идентичными, если одно из них может быть повернуто таким образом, что последовательность щупалец и когтей будет совпадать с этой последовательностью другого потомка.

Например, если мы обозначим, щупальцу как “Т”, а коготь как “С”, потомки "TCTCCT" и "CCTTCT" идентичны, а потомки "TTTCCT" и "CCTTTC" не идентичны.

Ваша программа должна для данного N посчитать общее кол-во различных (то есть не идентичных) потомков с N придатками.

Формат входного файла

Входной файл содержит одно целое число N .

Формат выходного файла

Выходной файл должен содержать одно целое число – количество различных потомков.

Ограничения

$1 \leq N \leq 58$

Примеры тестов

№	Входной файл	Выходной файл
1	1	2
2	4	6

Задача Elementary arithmetic (Элементарная арифметика)

Входной файл: input.txt

Ограничение времени на тест: 1 сек

Выходной файл: output.txt

Ограничение памяти на тест: 256 МБ

Подводная арифметика это очень просто. Есть только две операции : “WITH” и “WITHOUT”, которые означают сложение и вычитание соответственно.

Подводные математики еще не изобрели скобки, так что все выражения вычисляются справа налево. Например, выражение «3 WITH 5 WITHOUT 4 WITHOUT 7» вычисляется как $(3 + (5 - (4 - 7)))$.

Ученым института «Нирсия» подводной арифметики требуется программа, которая бы могла вычислять значение такого выражения.

Формат входного файла

Первая строка входного файла содержит целое число N . В следующих $2 \times N + 1$ строках. Эти $2 \times N + 1$ строк описывают выражение. Нечетные строки содержат целые операнды, а четные содержат операции.

Формат выходного файла

Выходной файл должен содержать одно целое число – результат вычисления.

Ограничения

$$1 \leq N \leq 10^3$$

Все операнды находятся в диапазоне от 0 до 1000.

Примеры тестов

№	Входной файл	Выходной файл
1	1 2 WITH 3	5
2	3 10 WITHOUT 5 WITH 3 WITH 15	-13

Задача Force of thrust (Сила тяги)

Входной файл: input.txt

Ограничение времени на тест: 1 сек

Выходной файл: output.txt

Ограничение памяти на тест: 256 МБ

Инженеры лаборатории «Нирсия», занимающаяся вопросами плавания, разработала новый подводный аппарат. Чтобы получить дополнительное финансирование, им посоветовали назвать его «nano-автомобиль».

После некоторых размышлений о том, как оправдать такое название, молодой инженер Вася предложил начать измерения силы двигателей в нано-ニュтонах.

Это предложение было весело принято, и вам сказали написать программу контроля для этого аппарата.

Автомобиль имеет четыре двигателя, расположенных на его левой, правой, верхней и нижней сторонах. Значения этих осевых сил обозначаются целыми числами f_L , f_R , f_U и f_D соответственно. Действительные значения этих сил – целые числа в диапазоне от -10^8 до 10^8 нано-ニュтонов. Положительные значения означают тягу вперед, отрицательные значения – в обратном направлении тяги.

Не смотря на то что все двигатели параллельны друг другу, их можно использовать для того, чтобы повернуть автомобиль. Например, если $f_L = -10^8$, $f_R = 10^8$, $f_U = f_D = 0$, то автомобиль поворачивает налево. Если $f_L = f_R = f_U = f_D = 10$, то автомобиль идет вперед на полной скорости.

Для человека, вместо того чтобы указывать значения f_L , f_R , f_U и f_D , легче указать такие значения : общая сила - T , H - горизонтально крутящий момент, V – вертикально крутящий момент, которые определяются следующим образом :

- $T = f_L + f_R + f_U + f_D$
- $H = f_R - f_L$
- $V = f_U - f_D$

Ваша программа должна, учитывая значения T , H и V , рассчитать допустимые значения f_L , f_R , f_U и f_D .

Формат входного файла

Входной файл содержит 3 целых числа T H V .

Формат выходного файла

Выходной файл должен содержать 4 целых числа f_L , f_R , f_U и f_D . Если существует несколько решений, разрешается вывести любой. Утверждается, что хотя бы одно решение существует.

Ограничения

$-4 \cdot 10^8 \leq T \leq 4 \cdot 10^8$;

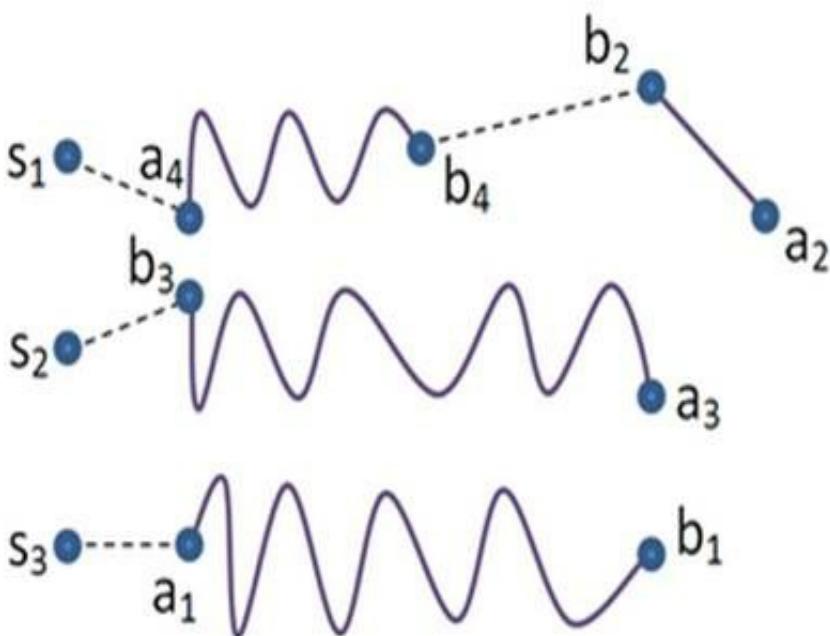
$-2 \cdot 10^8 \leq H, V \leq 2 \cdot 10^8$;

Примеры тестов

№	Входной файл	Выходной файл
1	0 10 0	-5 5 0 0
2	10 0 2	5 5 1 -1

Задача Group performance (Производительность группы)

Входной файл: input.txt Ограничение времени на тест: 1 сек
Выходной файл: output.txt Ограничение памяти на тест: 256 МБ
Сотрудничество транспортного средства является горячей темой в современной подводной робототехники. Часто группа транспортных средств может закончить миссию быстрее, чем только один. Но вы должны реализовать сложные методы для управления и извлечь выгоду из сотрудничества.



следовать аппарат. Задание с номером i имеет номера точек a_i и b_i , и длиной d_i ($d_i \geq |a_i - b_i|$).

Аппарату разрешено следовать по треку в любом направлении - a_i к b_i или наоборот. В обоих случаях длина пути будет d_i .

План представляет собой последовательность задач для каждого аппарата в порядке исполнения. Каждая задача следует в прямом порядке или в обратном, каждая задача должна появиться в плане ровно один раз.

Является горячей темой в современной подводной робототехнике. Часто группа транспортных средств может закончить миссию быстрее, чем только один. Но вы должны реализовать сложные методы для управления и извлечь выгоду из сотрудничества.

Номер задачи определяется двумя крайними точками трека, по которому должен пройти транспортное средство. Длина трека определяется суммой расстояний между всеми точками, по которым транспортное средство должно пройти. Длина трека – максимальная длина пути среди всех транспортных средств в плане.

Ваша программа должна, учитывая транспортные средства и задачи, найти план минимальной длины.

Можно считать, что все действие происходит на плоскости постоянной глубины, а размером транспортного средства можно пренебречь.

Формат входного файла

Каждая точка во входном файле описывается двумя целыми числами – x и y координатами. В первой строке входного файла даны два целых числа N M . В следующих N строках описываются отправные точки s_i . Следующие M строк

описывают задачи, каждое описание задачи состоит из точек a_i , b_i и длиной пути d_i .

Формат выходного файла

Выходной файл должен описать план минимальной длины. Если таких планов существует несколько, разрешается вывести любой из них. Описание плана состоит из N блоков, каждый из которых описывает последовательность задач для одного транспортного средства. J -ый блок начинается с целого неотрицательно числа nim_j , а затем nim_j пар (t_{jk}, dir_{jk}) . Здесь t_{jk} обозначает ряд k -ой задачи для J -ого транспортного средства, где это целое число в диапазоне $1 .. M$. Dir_{jk} может принимать два значения : 0 или 1. 0 обозначает направление от А к Б, 1 обозначает направление от Б к А.

Ограничения

$$1 \leq N, M \leq 10; 0 \leq d_i \leq 10^4;$$

Все координаты целые числа в диапазоне $0 .. 1000$.

Примеры тестов

№	Входной файл	Выходной файл
1	3 4 2 8 2 5 2 2 5 2 12 2 14 14 7 12 9 3 12 4 5 6 14 5 7 10 8 10	2 4 0 2 1 1 3 1 1 1 0

Задача Hall of water life (Зал водной жизни)

Входной файл: input.txt

Ограничение времени на тест: 1 сек

Выходной файл: output.txt

Ограничение памяти на тест: 256 Мб

Главный зал института «Нирсия» подводных исследований имеет форму длинного коридора. В коридоре расположены N аквариумов, которые демонстрируют различные морские существа. Аквариумы расположены на расстояниях x_1, \dots, x_N от входа в зал ($x_i < x_{i+1}$).

В институте сменился директор, который решил, что обеспечение всех аквариумов достаточно дорогое дело, поэтому он издал приказ об удалении M ($0 \leq M \leq N - 2$) аквариумов.

Чтобы свести к минимуму разногласия, было решено :

1) первый и последний аквариум должны остаться на своих местах;

2) максимальное расстояние между соседними аквариумами должно быть как можно меньше.

Ваша программа должна выбрать аквариумы для удаления таким образом, чтобы выполнялось условие выше.

Формат входного файла

Входной файл содержит два целых числа $N M$, на следующей строке следует N целых чисел x_i , которые являются искомыми расстояниями.

Формат выходного файла

Выходной файл должен содержать одно целое число – наименьшее возможное максимальное расстояние.

Ограничения

$2 \leq N \leq 400$, $1 \leq x_i \leq 10^9$,

Примеры тестов

№	Входной файл	Выходной файл
1	5 2 1 2 3 4 5	2
2	4 1 10 21 30 40	19

Задача Ichthyology (Ихтиология)

Входной файл: input.txt

Ограничение времени на тест: 1 сек

Выходной файл: output.txt

Ограничение памяти на тест: 256 МБ

Ученые института «Нирсия» лингвистической ихтиологии стали изучать язык рыбы. Они нашли несколько конкретных звуков, которые некоторые рыбы могут издавать, и решили назначить буквы латинского алфавита для них. Тогда они взяли прибор, который может записывать звуки под водой, и программное обеспечение, переводящее их в строку букв.

Они полагают, что подстрока из букв может иметь особое значение в языке рыбы, поэтому они хотят вычислить, как часто подстроки используются во время записи.

Ваша программа должна, имея две строки T и W , найти минимальное и максимальное непересекающихся вхождений W в T .

Например, если $W="abab"$ и $T=="ababbabababab"$, может интерпретироваться как "(abab)bb(abab)(abab)" (давая 3 случая) или как "(abab)bbab(abab)ab" (дающая 2 случая). Таким образом, минимальное число равно 2, а максимум 3.

Формат входного файла

Первая строка входного файла содержит строку W , Вторая строка входного файла содержит строку T .

Формат выходного файла

Выходной файл должен содержать два целых числа – минимальное и максимальное количество непересекающихся вхождений W в T .

Ограничения

$1 \leq \text{length}(W) \leq 100$;

$1 \leq \text{length}(T) \leq 1000$;

W и T должны содержать только строчные латинские буквы.

Примеры тестов

№	Входной файл	Выходной файл
1	a b	0 0
2	qq qqaqqqaqqq	3 4

Задача Jewel trade (Торговля)

Входной файл: input.txt

Ограничение времени на тест: 1 сек

Выходной файл: output.txt

Ограничение памяти на тест: 256 МБ

Недавно исследователи обнаружили новый вид гигантских разумных существ, живущих на дне глубокого озера.

Эти существа обычно не заинтересованы в делах с людьми. Тем не менее, они очень признательны за различные драгоценности, особенно бриллианты, которые трудно найти под водой. В обмен, эти существа предлагают широкий выбор высококачественного жемчуга.

Так была создана торговля. Каждый день, N различных видов драгоценностей продаются, некоторые из поверхности в воду, некоторые из воды на поверхность.

Государственная комиссия по торговле установила следующие правила :

1) Каждый драгоценный камень должен быть заключен в отдельный контейнер.

2) Ювелирные изделия того же происхождения должны иметь одинаковые контейнеры.

3) Эта торговля не должна изменить уровень воды в озере, поэтому общий объем всех контейнеров, перемещающихся из поверхности в озеро, должен быть равен общему объему всех контейнеров, перемещающихся из озера.

Ваша программа должна найти объем контейнера для каждого вида драгоценного камня, чтобы удовлетворить требованиям Комиссии.

Формат входного файла

Входной файл содержит целое число N .

Далее следует N целых чисел a_i , где если оно положительно, означает, что a_i драгоценностей i -ого происхождения перемещается с поверхности к озеру, а если отрицательное, то что a_i драгоценностей i -ого происхождения перемещается из озера.

Формат выходного файла

Выходной файл должен содержать N целых чисел b_i ($1 \leq b_i \leq 10^{12}$), что указывает на объем контейнеров для каждого вида драгоценного камня. Если существует несколько решений, разрешается вывести любое из них.

Ограничения

$$2 \leq N \leq 10^5$$

$$1 \leq |a_i| \leq 10^5$$

$2 \times \min(pos, neg) \geq \max(pos, neg)$, где pos – число положительных значений среди a_i , и neg – число отрицательных значений.

Примеры тестов

№	Входной файл	Выходной файл
1	3 1 2 -3	1 1 1
2	5 20 -27 48 -17 6	5 13 8 11 9

Задача Упрощение плана

Входной файл: input.txt

Выходной файл: output.txt

Ограничение времени на тест: 1 сек

Ограничение памяти на тест: 256 Мб

Условие

Стройка, приуроченная к саммиту АТЭС — очень большая задача. Как и все большие задачи, её разбили на подзадачи. Между подзадачами есть зависимости: есть такие пары подзадач (i, v) , что к подзадаче v нельзя приступать, не завершив предварительно подзадачу i .

При этом некоторые зависимости могут присутствовать и неявно. Например, если имеются зависимости (i, v) и (v, w) , то подзадачу w нельзя выполнять, пока не будет выполнена подзадача i .

Был составлен план стройки с подзадачами и зависимостями между ними. Однако выяснилось, что зависимостей в плане обозначено слишком много — некоторые из них можно удалить и всё равно они будут присутствовать неявно. Вам поручили написать программу, удаляющую как можно больше лишних зависимостей.

Множество зависимостей можно удалить, если после его удаления все неявные зависимости сохраняются.

Так, в примере 1 могут быть удалены зависимости $(1, 3)$ и $(2, 4)$. В том же примере зависимость $(2, 3)$ удалена быть не может, поскольку остальные четыре не создают её неявно.

Формат входного файла

Во первой строке входного файла находятся числа $N M$ — количество подзадач и количество зависимостей. Далее следует M пар чисел, обозначающих зависимости. Пара $i v$ означает, что подзадача v не может быть выполнена, пока не будет выполнена i .

Подзадачи пронумерованы первыми N натуральными числами.

Гарантируется, что ни одна зависимость не упомянута дважды.

Гарантируется, что никакая задача не зависит от самой себя и зависимости не образуют циклов.

Формат выходного файла

Выходной файл должен содержать целое число — максимальное количество зависимостей, которые можно удалить. Затем должна следовать последовательность номеров удаляемых зависимостей, перечисленных в порядке возрастания.

Ограничения

$1 \leq N \leq 50$

Примеры тестов

№ Входной файл Выходной файл

4 5 1 2 1 1 3 1 2 4	2 2 3
------------------------------	----------

3 4
2 3

Problem “Advertisement”

Input file: advert.in
Output file: advert.out
Time Limit: 1 second
Memory Limit: 64 megabytes

Администрация парка отдыха решила повысить рентабельность, продав место под рекламу вдоль популярных маршрутов прогулок горожан. Для этого было построено несколько щитков (места для рекламы). Все щитки пронумерованы соответственно их порядку на тропинках. На каждом щите может быть размещено не более одного рекламного объявления.

Рекламодатели согласны купить некоторые места под рекламу, но им необходимы гарантии того, что каждый прохожий увидит их рекламу не менее K раз, в течение прогулки по парку. Однако, у каждого человека свои собственные маршруты.

Исследования показали, что каждый посетитель выбрал для себя некоторый путь и гуляет по нему каждый день. Поскольку рекламодателей интересует только количество рекламных щитков, которые видят прохожие, то каждый маршрут задается номерами рекламных плакатов, встречающихся на пути. А поскольку номера щитков заданы в порядке их следования по прямой, то для каждого посетителя достаточно указать только индекс первого и последнего плаката, которые он видит.

К сожалению, некоторые посетители не гуляют настолько долго, чтобы видеть K плакатов, а некоторые находятся в такой плохой форме, что проходят только один плакат (в этом случае номера первого и последнего плакатов совпадают). Из-за этого рекламодатели решили, что реклама должна быть размещена на всех щитках вдоль таких маршрутов, так как это лучшее, что они могли предпринять, исходя из своего правила.

Ваша задача — написать программу, минимизирующую количество щитков, за которые придется платить, и в то же время выполняющую все указанные требования.

Входные данные.

Первая строка входного файла содержит два натуральных числа K и N ($1 \leq K, N \leq 1000$), разделенных пробелами. K — количество плакатов, которые каждый посетитель должен увидеть, N — количество посетителей.

Следующие N строк описывают путь каждого посетителя. В каждой строке находятся два числа. Первое — номер первого плаката, второе — последнего, виденного очередным посетителем. Числа разделены пробелом и не превосходят 10000 по модулю. Во время прогулки прохожий видит плакаты с указанными номерами, а также все плакаты, расположенные между ними.

Выходные данные.

В выходной файл выведите сначала целое A — минимальное количество щитков, на которые надо разместить рекламу, а потом A строчек. В каждой строчке номер (в возрастающем порядке) щитка, который необходимо купить для рекламы.

Пример входных и выходных данных

advert.in	advert.out
5 10	19
1 10	-5
20 27	-4
0 -3	-3
15 15	-2
8 2	-1
7 30	0
-1 -10	4
27 20	5
2 9	6
14 21	7
	8
	15
	18
	19

	20
	21
	25
	26
	27

Задача С Призы

Имя входного файла:

c.in

Имя выходного файла:

c.out

Максимальное время работы на одном тесте:

1 секунда

Максимальный объем используемой памяти:

64 мегабайта

Для N участников соревнований ($0 < N \leq 500$) спонсоры предоставили N призов стоимостью от 1 до N , причем стоимость всех призов разная. Спонсоры оговорили предоставление призов M условиями ($0 \leq M \leq 1000$). Каждое условие имеет вид: i -й участник должен получить более ценный приз, чем j -й. Необходимо предоставить организаторам возможный вариант распределения призов, учитывающий все условия спонсоров, если же это невозможно, то вывести 0.

В первой строке входного файла записано количество участников N , во второй – количество условий M , в последующих M строках – условия, записанные в виде $i\ j$, где i и j – номера участников, записанные через пробел. Такая запись означает, что i -й участник должен получить более ценный приз, чем j -й.

В выходном файле на i -ой строке необходимо записать стоимость приза, предоставляемого i -му участнику.

C.IN	C.OUT
4	4
3	3
1 2	2
2 3	1
1 4	

Задача А. Циклы.

Входной файл: `cycles.in`

Выходной файл: `cycles.out`

Максимальное время работы на одном тесте: 2 секунды

Максимальный объем используемой памяти: 128 мегабайт

Задан простой неориентированный граф. Выведите число простых циклов в нем. Цикл называется простым, если не содержит повторяющихся вершин и ребер (т.е. каждая вершина и ребро содержатся в нем не более 1 раза).

Формат входных данных

В первой строке входных данных записаны 2 числа N и M ($1 \leq N \leq 19$, $M \leq N*(N-1)/2$). В следующих M строках записаны пары чисел x и y ($1 \leq x, y \leq N$, $x \neq y$). Между каждой парой вершин не более одного ребра.

Формат выходных данных

Выведите одно число — число простых циклов.

Примеры

<code>cycles.in</code>	<code>cycles.out</code>
4 6	7
1 2	
1 3	
1 4	
2 3	
2 4	
3 4	

“Дефрагментировать”

Имя входного файла:

defrag.in

Имя выходного файла:

defrag.out

Максимальное время работы на одном тесте:

1 секунда

Вы принимаете участие в разработке операционной системы “Новое поколение” и файловой системы НГ. В этой файловой системе все дисковое пространство разделено на N ячеек равного размера, занумерованных числами от 1 до N. Каждый файл занимает одну или больше ячеек на произвольных местах на диске. Все ячейки, пока не заняты файлами считаются свободными. Файлы считаются быстрее, если все ячейки находятся последовательно на диске в естественном порядке.

Вращение диска с постоянной скоростью подразумевает, что разное время необходимо для доступа к ячейкам.

Таким образом, чтение ячеек, расположенных ближе к началу диска выполняется быстрее, чем чтение ячеек расположенных в конце. Все файлы заранее пронумерованы числами от 1 до K по убыванию частоты.

После оптимальной расстановки файлов на диске, файл 1 занимает ячейки: 1,2 .. S₁, файл 2: S₁+1, S₁+2, ... S₁+S₂ и.т.д. (здесь S_i это номер последней ячейки, которой занимает i файл)

Для того чтобы разместить файлы на диске оптимальным образом используются операции движения.

Одна операция движения включает в себя чтение одной занятой ячейки с диска и запись его в другую (свободную) ячейку на диске. После этого первая объявляется свободной, а вторая занятой.

Input

В первой строке находятся числа N и K разделенные пробелами. Дальше идет K строк, в каждой из которых описывается 1 файл. Описание i файла начинается с числа S_i – количество ячеек памяти для i файла, после идут S_i целых чисел – номера ячеек в естественном порядке. Все номера ячеек различны и есть хотя бы одна свободная ячейка на диске.

Output

Ваша программа должна вывести минимальное количество операций движений, для того чтобы разместить файлы на диске оптимальным способом. Если файлы на диске уже размещены оптимальным способом, то ваша программа должна вывести ноль.

Пример 1

```
20 3
4 2 3 11 12
1 7
3 18 5 10
```

Вывод 1

9

Пример 2

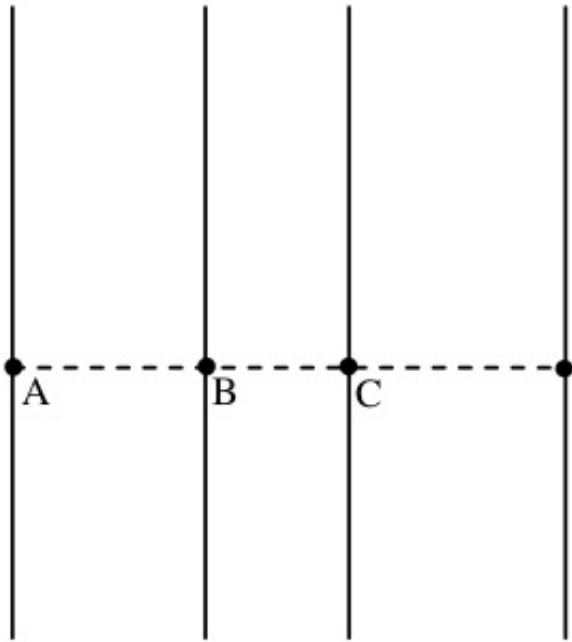
```
30 4
2 1 2
3 3 4 5
2 6 7
8 8 9 10 11 12 13 14 15
```

Вывод 2

0

Задача abcd

Имя входного файла:	stdin
Имя выходного файла:	stdout
Максимальное время работы на одном тесте:	1 с.
Максимальный объем памяти:	256 мб



У вас есть четыре котёнка. Назовём их, скажем, так: котёнок Беня, котёнок Веня, котёнок Сеня и котёнок Деня. Котёнок Беня находится в точке А, котёнок Веня находится в точке В, котёнок Веня - в точке С и котёнок Деня - в точке D.

Котята любят играть в игры. Сейчас они придумали такую игру: они как угодно перемещаются по четырём параллельным прямым (каждый по своей, проходящей через его исходную позицию) так, чтобы в итоге позиции, где они окажутся, образовали прямоугольник. При этом их исходные позиции А, В, С и D находятся на одном отрезке, перпендикулярном четырём прямым перемещения (см. рис. для пояснения).

Зная расстояния AB, BC и CD (т. е. расстояния между прямыми перемещения), помогите котятам узнать, получится ли им образовать прямоугольник, и если да, то какую минимальную площадь он может иметь.

Формат входных данных

В единственной строке входного файла заданы три действительных числа AB, BC и CD ($0 < AB, BC, CD \leq 10000$, числа заданы с не более чем двумя знаками после десятичной точки).

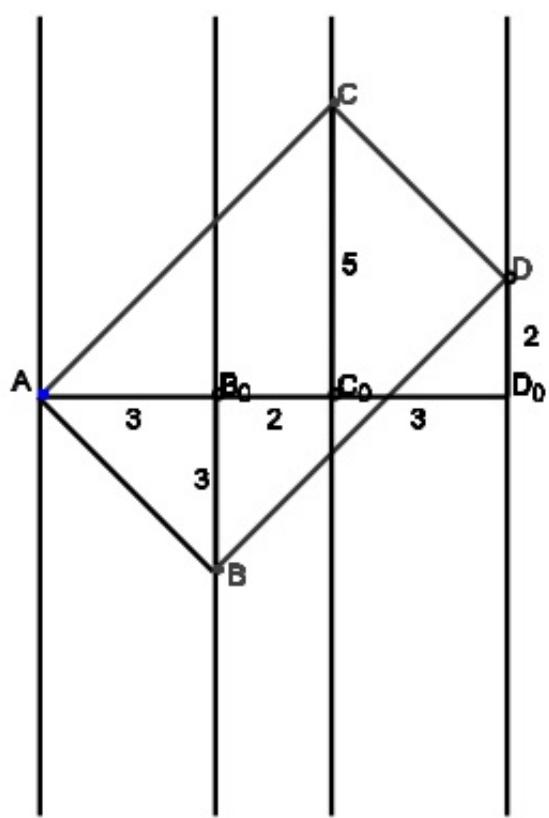
Формат выходных данных

Если образовать прямоугольник невозможно, выведите грустный смайлик: ":-(". Иначе выведите минимальную площадь подобного прямоугольника **с не менее, чем четырьмя знаками после запятой**.

Примечание

Первые два теста - тесты из условия. Каждый тест оценивается в отдельности.

Пояснение к первому примеру:



Пример

stdin	stdout
3 2.0 3.00	30.0000
0.01 10 10000	:-)

Задача Королевская прогулка

Имя входного файла:	kingwalk.in
Имя выходного файла:	kingwalk.out
Максимальное время работы на одном тесте:	1 с.
Максимальный объем памяти:	256 мб

Шахматный король хочет совершить прогулку по Весеннему полю. Весенне поле представляет собой прямоугольник, каждая клетка которого содержит одну букву латинского алфавита. Король начнет прогулку в произвольной клетке поля и сделает $n - 1$ ходов; каждый ход происходит по правилам хода шахматного короля (в клетку, соседнюю с текущей по стороне или по вершине). Нельзя вместо хода оставаться на той же клетке. Во время прогулки разрешается посещать одну и ту же клетку более одного раза. Таким образом, король посетит n клеток. Если рассмотреть буквы из этих клеток в порядке их обхода, то получится строка s , состоящая из n букв. Вечером король сравнит эту строку с девизом его династии, в котором тоже ровно n букв. Если в i -й позиции в строке s и в i -й позиции в девизе стоит одна и та же буква, то король повысит жалованье i -й пешке. Помогите пешкам подсказать королю такой маршрут, при котором как можно больше пешек получат повышение.

Формат входных данных

Первая строка входного файла содержит два целых числа h и w ($2 \leq h, w \leq 20$) — размеры Весеннего поля. Далее следуют h строк, каждая из которых содержит w строчных латинских букв без пробелов, — это описание Весеннего поля. Затем следует строка, содержащая целое число n ($1 \leq n \leq 50$). В последней строке содержится девиз королевской династии — строка из n строчных букв латинского алфавита.

Формат выходных данных

В первой строке выведите m — максимальное возможное число пешек, получающих повышение. В следующих n строках выведите координаты клеток в том порядке, в котором их должен обходить король, чтобы m пешек получили повышение. Координаты клетки — это номер строки (строки нумеруются сверху вниз, начиная с 1) и номер столбца (столбцы нумеруются слева направо, начиная с 1). Если оптимальных маршрутов несколько, выведите любой из них.

Пример

kingwalk.in	kingwalk.out
3 9	4
zhautykov	3 1
olympiadk	2 2
azakhstan	1 3
6	1 4
almaty	1 5
	1 6

3 9	6
zhautykov	2 7
olympiadk	3 6
azakhstan	3 7
6	3 8
astana	3 9
	3 8

Задача Group performance (Производительность группы)

Входной файл: input.txt

Ограничение времени на тест: 1 сек

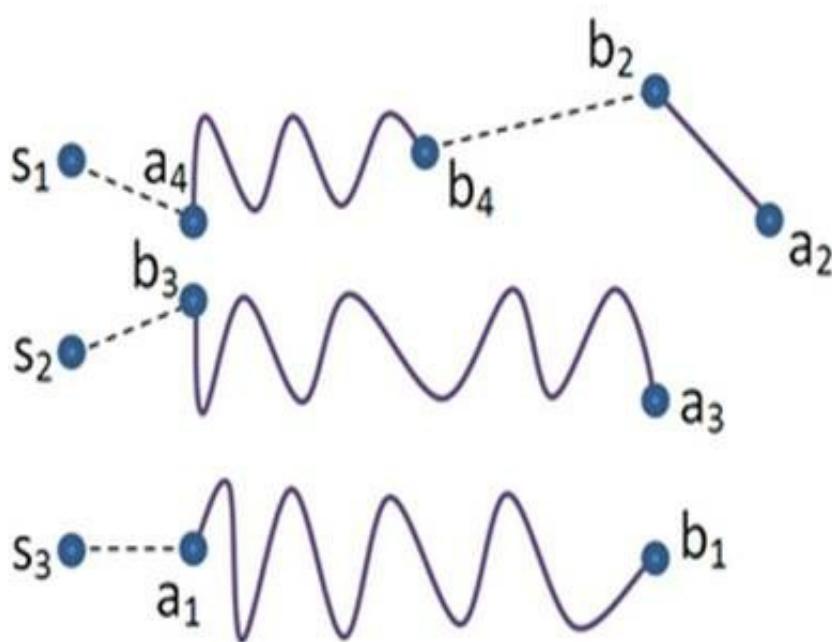
Выходной файл: output.txt

Ограничение памяти на тест: 256 МБ

Сотрудничество транспортного средства

является горячей темой в современной подводной робототехники. Часто группа транспортных средств может закончить миссию быстрее, чем только один. Но вы должны реализовать сложные методы для управления и извлечь выгоду из сотрудничества.

Рассмотрим N аппаратов и M задач, которые они должны выполнить. Каждая задача должна быть выполнена ровно одним автомобилем. Каждая задача представляет из себя трек по которому должен определяться двумя краевыми



следовать аппарат. Задание с номером i имеет точками a_i и b_i , и длиной d_i ($d_i \geq |a_i - b_i|$).

Аппарату разрешено следовать по треку в любом направлении - a_i к b_i или наоборот. В обоих случаях длина пути будет d_i .

План представляет собой последовательность задач для каждого аппарата в порядке исполнения. Каждая задача следует в прямом порядке или в обратном, каждая задача должна появиться в плане ровно один раз.

Д-ый аппарат начинается движение в точке s_j . Он выполняет свои задачи в соответствии с планом. Когда аппарат выполняет задачу, он идет к начальной точке следующего, двигаясь по прямой линии. Когда все задачи, поставленные транспорту, закончатся, он останавливается. Длиной пути аппарата является общая длина, пройденная транспортным средством во время выполнения плана. Длина плана – максимальная длина пути среди всех транспортных средств в плане.

Ваша программа должна, учитывая транспортные средства и задачи, найти план минимальной длины.

Можно считать, что все действие происходит на плоскости постоянной глубины, а размером транспортного средства можно пренебречь.

Формат входного файла

Каждая точка во входном файле описывается двумя целыми числами – x и y координатами. В первой строке входного файла даны два целых числа N M . В следующих N строках описываются отправные точки s_i . Следующие M строк

описывают задачи, каждое описание задачи состоит из точек a_i , b_i и длиной пути d_i .

Формат выходного файла

Выходной файл должен описать план минимальной длины. Если таких планов существует несколько, разрешается вывести любой из них. Описание плана состоит из N блоков, каждый из которых описывает последовательность задач для одного транспортного средства. J -ый блок начинается с целого неотрицательно числа nim_j , а затем nim_j пар (t_{jk}, dir_{jk}) . Здесь t_{jk} обозначает ряд k -ой задачи для J -ого транспортного средства, где это целое число в диапазоне $1 .. M$. Dir_{jk} может принимать два значения : 0 или 1. 0 обозначает направление от А к Б, 1 обозначает направление от Б к А.

Ограничения

$$1 \leq N, M \leq 10; 0 \leq d_i \leq 10^4;$$

Все координаты целые числа в диапазоне $0 .. 1000$.

Примеры тестов

№	Входной файл	Выходной файл
1	3 4 2 8 2 5 2 2 5 2 12 2 14 14 7 12 9 3 12 4 5 6 14 5 7 10 8 10	2 4 0 2 1 1 3 1 1 1 0

Задача L. Число возрастающих подпоследовательностей

Имя входного файла:	subseq.in
Имя выходного файла:	subseq.out
Максимальное время работы на одном тесте:	0.5 секунды
Максимальный объем памяти:	64 мб

Задана последовательность из n чисел a_1, a_2, \dots, a_n . Подпоследовательностью длины k этой последовательности называется набор индексов i_1, i_2, \dots, i_k , удовлетворяющий неравенствам $1 \leq i_1 < i_2 < \dots < i_k \leq n$. Подпоследовательность называется возрастающей, если выполняются неравенства $a_{i_1} < a_{i_2} < \dots < a_{i_k}$.

Необходимо найти число возрастающих подпоследовательностей наибольшей длины заданной последовательности a_1, \dots, a_n . Так как это число может быть достаточно большим, необходимо найти остаток от его деления на $10^9 + 7$.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$). Вторая строка входного файла содержит n целых чисел: a_1, a_2, \dots, a_n . Все a_i не превосходят 10^9 по абсолютной величине.

Формат выходных данных

В выходной файл выведите ответ на задачу.

subseq.in	subseq.out
5 1 2 3 4 5	1
6 1 1 2 2 3 3	8

Задача А. Обобщенные числа Фибоначчи

Имя входного файла: fib.in
Имя выходного файла: fib.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Обобщенными числами Фибоначчи $F_n^{(k)}$ называют следующую последовательность:
 $F_1^{(k)} = F_2^{(k)} = \dots = F_k^{(k)} = 1, F_i^{(k)} = \sum_{j=1}^k F_{i-j}^{(k)}, i > k.$

Ваша задача — вычислить остаток от деления $F_n^{(k)}$ на p .

Формат входного файла

Входной файл содержит три целых числа: n, k и p ($1 \leq n, k \leq 10^6, 2 \leq p \leq 10^9$).

Формат выходного файла

В выходной файл выведите $F_n^{(k)} \bmod p$.

Примеры

fib.in	fib.out
3 2 10	2
10 2 10	5