

Задача А. Плохая подстрока

Имя входного файла: badsubs.in
Имя выходного файла: badsubs.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Найдите, сколько существует строк заданной длины n , состоящих только из символов «а», «b» и «с», но не содержащих подстроки «ab».

Формат входных данных

В первой строке задано целое число n ($0 \leq n \leq 22$).

Формат выходных данных

Выведите количество таких строк.

Примеры

badsubs.in	badsubs.out
0	1
3	21
11	46368

Задача В. Ядра

Имя входного файла: balls.in
Имя выходного файла: balls.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Капитан Вася всегда держит на своем корабле запас пушечных ядер для борьбы с пиратами. Так как он привык во всем поддерживать порядок, он хранит ядра в виде пирамид. Каждый из слоёв одной пирамиды является заполненным ядрами равносоставленным треугольником, сторона которого содержит ровно k ядер. Сторона основания пирамиды состоит из n ядер, в следующем слое сторона состоит из $n-1$ ядра, и так далее, пока на вершину не будет положено одно ядро (которое является равносоставленным треугольником со стороной 1).

Например, пирамида размера 3 состоит из трех уровней, выглядящих так (сверху вниз):

```
x
x
x x
x
x x
x x x
```

Ясно, что каждый из треугольников может содержать только 1, 3, 6, 10, ... ядер. Таким образом, пирамида может содержать только 1, 4, 10, 20, ... ядер.

Вася отправляется в плавание и берет с собой ровно m ядер. Какое минимальное число пирамид требуется ему сложить из них на своем корабле?

Формат входных данных

В первой строке ввода записано количество тестов $1 \leq T \leq 20$. В последующих T строках задается количество ядер в i -м тесте $1 \leq m_i \leq 300\,000$.

Формат выходных данных

Для каждого из T тестов выведите в отдельной строке минимальное количество пирамид.

Пример

balls.in	balls.out
5	1
1	2
5	3
9	3
15	2
91	

Задача С. Кирпичи

Имя входного файла: bricks.in
Имя выходного файла: bricks.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дано бесконечное количество кирпичей двух цветов — красного и синего. Красные кирпичи имеют длину 2 дециметра, а синие — 3 дециметра. Сколько различных способов выложить ряд из кирпичей длины n дециметров? Способы считаются различными, если на каком-то одинаковом расстоянии от начала ряда в них лежат кирпичи разного цвета.

Формат входных данных

В первой строке задано целое число n ($0 \leq n \leq 80$).

Формат выходных данных

В первой строке выведите одно число — количество способов выложить ряд из кирпичей длины n дециметров.

Примеры

bricks.in	bricks.out
2	1
6	2

Задача D. Мне это больше не нужно!

Имя входного файла: dontneed.in
Имя выходного файла: dontneed.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Один отважный путешественник отправился облетать шар земной на шаре воздушном. К сожалению, его воздушный шар подхватило бурей. Шар начал падать, и путешественник приготовился было умирать, но, к счастью, на горизонте показался остров. Остров был маленький, однако стало ясно, что если выкинуть из корзины вещи, то можно до него долететь, и даже перелететь. Известно, что если путешественник выкинет вещи суммарным весом больше, чем w , то он перелетит за край острова и упадёт в бушующее море, где его ждёт неминуемая смерть. Естественно, что его не устраивает такой исход, и потому он хочет выкинуть вещей, с одной стороны, как можно большего суммарного веса (чтобы гарантированно оказаться на острове), с другой стороны, этот вес ни в коем случае не должен превосходить w . Нужно написать программу, которая бы решила бы, какие вещи нужно выкидывать. Помогите отважному путешественнику!

Формат входных данных

В первой строке заданы два целых числа: n (число вещей в корзине воздушного шара, $1 \leq n \leq 20$) и w (предельный вес, $1 \leq w \leq 10^9$). Во второй строке записаны n целых чисел, разделённые пробелами, каждое из которых описывает вес некоторой вещи, находящейся в корзине воздушного шара. Каждое из этих чисел лежит в пределах от 1 до 10^6 .

Формат выходных данных

В первой строке необходимо выдать максимально возможное значение суммарного веса w_{max} . Во второй строке необходимо вывести список номеров вещей, взяв которые, можно достигнуть такого веса. Номера должны выводиться через пробел. Если оптимальных решений несколько, разрешается выводить любое из них.

Пример

dontneed.in	dontneed.out
5 13	12
3 5 6 9 9	1 5

Задача E. Редакционное расстояние

Имя входного файла: editdist.in
Имя выходного файла: editdist.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В информатике *редакционным расстоянием* между двумя строками называется минимальное количество добавлений, удалений и замен символов, при помощи которых

можно из одной строки получить другую. К примеру, редакционное расстояние между строками «ab» и «ab» равно нулю, так как строки равны между собой безо всяких изменений; расстояние между строками «short» и «ports» равно трём: в слове «short» нужно удалить из начала букву «s», заменить «h» на «p» и добавить в конец букву «s». Редакционное расстояние также называют *расстоянием Левенштейна*.

Найдите редакционное расстояние между двумя заданными строками.

Формат входных данных

В первой строчке задана одна строка, во второй — другая. Длины обеих строк — от 1 до 100 символов, включительно.

Формат выходных данных

Выведите единственное число — редакционное расстояние между двумя заданными строками.

Примеры

editdist.in	editdist.out
ab ab	0
short ports	3

Задача F. Поле

Имя входного файла: field.in
Имя выходного файла: field.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Отряду нужно пересечь прямоугольное поле размера $m \times n$ квадратов, двигаясь из левого верхнего угла в правый нижний и перемещаясь между соседними квадратами только в двух направлениях — вправо и вниз. Поле не очень ровное, но у отряда есть карта, на которой отмечена высота каждого квадрата. Опасность перехода с квадрата высоты h_1 на соседний квадрат высоты h_2 оценивается числом $|h_2 - h_1|$; опасность всех переходов в пути суммируется. Выясните, какова минимальная опасность пути из квадрата $(1, 1)$ в квадрат (m, n) .

Формат входных данных

В первой строке заданы два числа m и n через пробел ($1 \leq m, n \leq 100$). В следующих n строках записано по m чисел в каждой; i -е число j -й из этих строк соответствует высоте квадрата (i, j) . Все высоты — целые числа в диапазоне от 1 до 100, включительно.

Формат выходных данных

Выведите одно число — минимальную опасность пути из квадрата $(1, 1)$ в квадрат (m, n) .

Примеры

field.in	field.out
2 2 1 1 1 1	0
4 2 1 2 3 5 3 8 4 7	6
2 3 1 2 2 3 3 1	4

Задача G. Лестница

Имя входного файла: ladder.in
Имя выходного файла: ladder.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

У лестницы n ступенек, пронумерованных числами $1, 2, \dots, n$ снизу вверх. На каждой ступеньке написано число. Начиная с подножия лестницы (его можно считать ступенькой с номером 0), требуется взобраться на самый верх (ступеньку с номером n). За один шаг можно подниматься на одну или на две ступеньки. После подъёма числа, записанные на посещённых ступеньках, складываются. Нужно подняться по лестнице так, чтобы сумма этих чисел была как можно больше.

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 100$). Во второй строке заданы целые числа a_1, a_2, \dots, a_n через пробел ($-10\,000 \leq a_i \leq 10\,000$) — это числа, записанные на ступеньках.

Формат выходных данных

В первой строке выведите одно число — максимальную сумму, которую можно получить, поднявшись по данной лестнице.

Примеры

ladder.in	ladder.out
2 1 2	3
2 2 -1	1
3 -1 2 1	3

Задача H. Наибольшая общая подпоследовательность

Имя входного файла: lcs.in
Имя выходного файла: lcs.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны две последовательности. Найдите длину их наибольшей общей подпоследовательности (подпоследовательность — это то, что можно получить из данной последовательности вычёркиванием некоторых элементов).

Формат входных данных

В первой строке записано число N — длина первой последовательности ($1 \leq N \leq 1000$).

Во второй строке записаны члены первой последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю.

В третьей строке записано число M — длина второй последовательности ($1 \leq M \leq 1000$).

В четвёртой строке записаны члены второй последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю.

Формат выходных данных

Выведите единственное целое число: длину наибольшей общей подпоследовательности, или же число 0, если такой не существует.

Пример

lcs.in	lcs.out
3 1 2 3 4 2 1 3 5	2

Задача I. Учим языки

Имя входного файла: linguae.in
Имя выходного файла: linguae.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В нашем университете открылись новые курсы французского языка. Каждый, кто записывается на эти курсы, в зависимости от своих знаний получает после тестирования уровень k — некоторое целое число в пределах от 0 до n . После этого он может выбирать один из трёх курсов — лёгкий, средний или сложный. В зависимости от выбранного курса после его прослушивания уровень повышается на 1, 2 или 3 соответственно. При этом учащийся может посещать и курсы, предназначенные для уровня l , если выполнено условие $l < k$, в таком случае его уровень после прослушивания будет равен $\max\{l + i, k\}$, где $i = 1, 2, 3$ в зависимости от интенсивности курса. Также уровень не может стать больше, чем n .

Петя хочет выучить французский язык, не зная его (то есть, имея уровень 0), при этом по возможности затратив как можно меньше денег на обучение. Напишите программу, которая вычисляет минимальное число денег, которое ему придется потратить для достижения уровня n .

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 10\,000$) — количество различных уровней на курсах. Далее следуют n строк, в каждой из которых записаны по три целых числа — стоимости легкого, среднего и сложного курсов для соответствующего уровня (от 0 до $n - 1$). Стоимости лежат в пределах от 1 до 10 000.

Формат выходных данных

Выведите единственное число — минимальное количество денег, которое требуется Пете, чтобы выучить французский язык.

Пример

linguae.in	linguae.out
3	2
1 2 3	
3 2 1	
2 1 3	

Задача J. Лекция

Имя входного файла: lecture.in
Имя выходного файла: lecture.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Образование — это то, что остаётся, когда забываешь всё, чему учился в школе.

Эйнштейн

Константин Игоревич читает завтра лекцию по алгебре. Сегодня же ему надо решить, какие из теорем, которые он будет рассказывать, следует доказать. С одной стороны, лекция не может длиться более T минут, так что на доказательство вообще всех теорем времени, скорее всего, не хватит. С другой стороны, хочется, чтобы лекция оказалась для студентов как можно полезнее.

Полезность лекции определяется как сумма полезностей всех доказанных теорем, в то время как *полезность теоремы* — какое-то заранее известное для каждой теоремы число. Теоремы делятся на три класса по сложности, и их доказательство занимает одну, две или три минуты в зависимости от класса.

Конечно же, прилежные студенты заранее выучили все определения и формулировки, так что всё время лекции можно потратить на доказательство теорем.

Формат входных данных

В первой строке ввода заданы через пробел два числа N и T ($1 \leq N \leq 100\,000$, $1 \leq T \leq 300\,000$) — количество теорем и продолжительность лекции (в минутах). Далее идут N строк, в каждой два числа t_i и a_i ($1 \leq t_i \leq 3$, $0 \leq a_i \leq 10^9$), обозначающих класс сложности i -й теоремы (то есть, время в минутах, требуемое на её доказательство) и её полезность.

Формат выходных данных

Выведите одно число — максимальную полезность завтрашней лекции Константина Игоревича.

Примеры

lecture.in	lecture.out
3 3 1 3 2 10 2 14	17
3 4 2 5 2 10 3 14	15

Задача K. Зайчик

Имя входного файла: lepus.in
Имя выходного файла: lepus.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Зайчик прыгает по прямой просеке, для удобства разделённой на n клеток. Клетки пронумерованы по порядку натуральными числами от 1 до n . Некоторые клетки заболочены: если зайчик прыгнет на такую клетку, ему несдобровать. Некоторые другие клетки просеки поросли вкусной зелёной травой: прыгнув на такую клетку, зайчик сможет отдохнуть и подкрепиться.

Зайчик начинает свой путь из клетки с номером 1 и хочет попасть в клетку с номером n , по пути ни разу не провалившись в болото и скушав как можно больше вкусной зелёной травы. Конструктивные особенности зайчика таковы, что из клетки с номером k он может прыгнуть лишь в клетки с номерами $k + 1$, $k + 3$ и $k + 5$.

Выясните, какое максимальное количество клеток с травой сможет посетить зайчик на своём пути.

Формат входных данных

В первой строке задано число n — количество клеток ($2 \leq n \leq 1000$). Вторая строка состоит из n символов; i -й символ соответствует i -й клетке просеки. Символ «w» обозначает болото, символ «.» — зелёную траву, а символ «.» соответствует клетке без

каких-либо особенностей. Гарантируется, что первая и последняя клетки не содержат болот и травы.

Формат выходных данных

В первой строке выведите одно число — максимальное количество клеток с травой, которые зайчик сможет посетить на своём пути. Если зайчику не удастся оказаться в клетке с номером n , выведите -1 .

Примеры

lepus.in	lepus.out
4 ."".	2
5 .w".	0
9 .www.www.	-1

Задача L. Максимальная сумма

Имя входного файла: maxsum.in
Имя выходного файла: maxsum.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дана последовательность чисел a_1, a_2, \dots, a_n . Разрешается выбрать из неё подпоследовательность $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ ($1 \leq i_1 < i_2 < \dots < i_k, 0 \leq k \leq n$) такую, чтобы любые два индекса i_r и i_s отличались больше, чем на 1 (таким образом, в подпоследовательности не могут попасть два числа, стоящие рядом в исходной последовательности). Как осуществить этот выбор так, чтобы сумма чисел в подпоследовательности была максимально возможной?

Формат входных данных

В первой строке ввода задано целое число n — длина последовательности ($1 \leq n \leq 10\,000$). Во второй строке заданы n целых чисел a_1, a_2, \dots, a_n через пробел — сама последовательность ($-10\,000 \leq a_i \leq 10\,000$).

Формат выходных данных

В единственной строке выведите максимально возможную сумму чисел в подпоследовательности.

Примеры

maxsum.in	maxsum.out
2 1 2	2
1 -1	0
4 -10 10 10 -10	10

Задача M. Деньги

Имя входного файла: money.in
Имя выходного файла: money.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

У вас имеется неограниченное количество монеток N заданных достоинств. Вам требуется определить, можно ли с их помощью разменять заданные K сумм денег.

Формат входных данных

В первой строке число N , далее во второй строке N чисел, задающих достоинства монеток. В третьей строке — число K , далее в четвёртой K чисел, определяющих размеры сумм. Все заданные числа натуральные и не превосходят 1000.

Формат выходных данных

В единственной строке выведите K чисел — для каждой суммы 0, если её разменять нельзя, и 1, если можно.

Пример

money.in	money.out
2 3 5	1 1 0 1 1
5 3 6 7 11 12	

Задача N. Грибы

Имя входного файла: mushroom.in
Имя выходного файла: mushroom.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Маша решила навестить свою бабушку. Она взяла с собой две корзинки — одну с пирожками, а другую — пустую, для грибов, которые она хочет собрать по пути.

Для того, чтобы попасть к бабушке, Маше необходимо пройти через лес, который представляет собой прямоугольник размером $m \times n$, в некоторых клетках которого

растут деревья, а в некоторых — грибы. Маша выходит из клетки $(1, 1)$ и идёт к бабушке в деревню, расположенную в клетке (m, n) . Каждым своим ходом Маша может пойти вправо или вниз (то есть увеличить одну и только одну из своих координат на 1), если в клетке, в которой она после этого окажется, не стоит дерево. Если в обеих клетках и справа, и снизу, находятся деревья, то Маша считается заблудившейся.

Вам необходимо по данному лесу выяснить, может ли Маша дойти до бабушки, не заблудившись, и если может, то посчитать максимальное количество грибов, которое она может при этом собрать.

Формат входных данных

В первой строке находятся четыре числа m, n, g, t ($2 \leq m, n \leq 100$, $0 \leq g, t \leq g + t \leq m \cdot n - 2$). В следующих g строках расположены по два числа в каждой — x и y -координаты i -го гриба. За ними следуют t строк с описаниями деревьев в аналогичном формате. Ни в какой клетке не может расти больше одного гриба, гриб и дерево одновременно, или больше одного дерева. Кроме того, в клетках $(1, 1)$ и (m, n) ничего не растёт.

Формат выходных данных

Если Маша может дойти до бабушки, то в первой строке необходимо выдать максимальное количество грибов, которое она сможет при этом собрать, а в последующих $m + n - 1$ строках нужно выдать координаты клеток, последовательно посещаемых Машей, в формате $x_i y_i$, для пути, на котором достигается максимальное количество грибов. Если таких путей несколько, то разрешается выдавать любой из этих путей.

В противном случае нужно вывести единственное число -1 .

Примеры

mushroom.in	mushroom.out
4 4 3 2 1 4 2 3 4 3 2 2 3 4	2 1 1 1 2 1 3 2 3 3 3 4 3 4 4
2 2 0 2 1 2 2 1	-1

Задача О. Гвоздики

Имя входного файла: nails.in
Имя выходного файла: nails.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На прямой дощечке вбиты гвоздики. Любые два гвоздика можно соединить ниточкой. Требуется соединить какие-то пары гвоздиков ниточками так, чтобы к каждому гвоздику была привязана хотя бы одна ниточка, а суммарная длина всех ниточек была минимальна.

Формат входных данных

В первой строке записано число N — количество гвоздиков ($2 \leq N \leq 100$). В следующей строке записано N чисел — координаты всех гвоздиков (неотрицательные целые числа, не превосходящие 10 000).

Формат выходных данных

Выведите единственное число — минимальную суммарную длину всех ниточек.

Пример

nails.in	nails.out
5 4 10 0 12 2	6

Задача Р. Наилучшее приближение

Имя входного файла: nearest.in
Имя выходного файла: nearest.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам даны N целых чисел. Ваша задача — вставить ровно по одному знаку «+» или «-» между каждой парой соседних таким образом, чтобы сделать значение получившегося выражения максимально близким к заданному числу A .

Формат входных данных

Первая строка содержит два целых числа: N ($1 \leq N \leq 10\,000$) и A , которое по модулю не превосходит 10 000. Далее следуют N строк, в каждой из которых содержится ровно одно целое число X_i , не превосходящее по модулю 10 000. Кроме того, гарантируется, что сумма абсолютных величин всех N чисел также не превосходит 10 000.

Формат выходных данных

В первой строке необходимо вывести значение получившегося выражения (которое должно быть настолько близко к A , насколько это возможно). Во второй строке необходимо вывести само выражение, дающее такое значение, в форме $X_1[+|-]X_2[+|-]\dots X_{N-1}[+|-]X_N$. Если оптимальных решений несколько, то разрешается выводить любое из них.

Пример

nearest.in	nearest.out
3 0	0
3	3+-2-1
-2	
1	

Задача Q. Три единицы

Имя входного файла: ones.in
Имя выходного файла: ones.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Требуется найти количество последовательностей заданной длины из нулей и единиц, в которых не встречается трёх единиц подряд.

Формат входных данных

В первой строке задано одно натуральное число n — длина последовательностей ($1 \leq n \leq 10^5$).

Формат выходных данных

Выведите количество последовательностей по модулю 12345.

Примеры

ones.in	ones.out
1	2
4	13

Задача R. Шаблоны

Имя входного файла: patterns.in
Имя выходного файла: patterns.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Многие операционные системы используют шаблоны для ссылки на группы объектов: файлов, пользователей и прочего. Ваша задача — реализовать простейший алгоритм проверки шаблонов для имён файлов.

В этой задаче алфавит состоит из маленьких букв английского алфавита и точки («.»). Шаблоны могут содержать произвольные символы алфавита, а также два специальных символа: «?» и «*». Знак вопроса («?») соответствует ровно одному произвольному символу. Звёздочка «*» соответствует подстроке произвольной длины (возможно, нулевой). Символы алфавита, встречающиеся в шаблоне, отображаются на ровно один такой же символ в проверяемой строке. Строка считается подходящей под шаблон, если символы шаблона можно последовательно отобразить на символы строки таким

образом, как описано выше. Например, строки «ab», «aab» и «beda.» подходят под шаблон «*a?», а строки «bebe», «a» и «ba» — нет.

Формат входных данных

Первая заданная строка определяет шаблон P . Вторая строка S состоит только из символов алфавита. Её необходимо проверить на соответствие шаблону. Длины обеих строк не превосходят 10 000. Строки могут быть пустыми — будьте внимательны!

Формат выходных данных

Если данная строка подходит под шаблон, выведите «YES». Иначе выведите «NO».

Примеры

patterns.in	patterns.out
k?t*n kitten	YES
k?t?n kitten	NO

Задача S. Два шаблона

Имя входного файла: patterns2.in
Имя выходного файла: patterns2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Многие операционные системы используют шаблоны для ссылки на группы объектов: файлов, пользователей и прочего. Ваша задача — найти строку минимально возможной длины, которая подходит под два заданных шаблона.

Алфавит в этой задаче состоит из маленьких букв латинского алфавита и точки («.»). Шаблоны могут содержать любые символы алфавита, а также специальные символы «?» и «*». Под «?» подходит любой символ алфавита, а под «*» — любая строка символов алфавита (возможно, пустая). Под символы алфавита, встречающиеся в шаблоне, подходят только такие же символы алфавита. Строка считается подходящей под шаблон, если символы шаблона можно последовательно отобразить в строку вышеуказанным способом. Например, строки «ab», «aab» и «beda.» подходят под шаблон «*a?», а строки «bebe», «a» и «ba» — нет.

Формат входных данных

Входные данные состоят из одного или нескольких тестовых случаев. В первой строке записано одно целое число — количество тестовых случаев.

Каждый тестовый случай состоит из двух строк, содержащих шаблоны P_1 и P_2 . Длина любого из шаблонов не превосходит 100 символов.

Формат выходных данных

Для каждого из тестовых случаев ответ задаётся одной строкой:

- Если строка, подходящая под оба шаблона, существует, выведите такую строку минимально возможной длины (если таких несколько, разрешается выводить любую).
- В противном случае выведите строку «NO».

Пример

<code>patterns2.in</code>	<code>patterns2.out</code>
2 *k*tt*n* *i*e* haha hihi	kitten NO

Задача Т. Мирные множества

Имя входного файла: `peacefulsets.in`
Имя выходного файла: `peacefulsets.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Группа математиков проводит бои между натуральными числами. Результаты боя между двумя натуральными числами, вообще говоря, случайны, однако подчиняются следующему правилу: если одно из чисел не менее чем в два раза превосходит другое, то большее число всегда побеждает; в противном случае победить может как одно, так и другое число.

Бой называется *неинтересным*, если его результат предопределён. Множество натуральных чисел называется *мирным*, если бой любой пары различных чисел из этого множества неинтересен. *Силой* множества называется сумма чисел в нём. Сколько существует мирных множеств натуральных чисел силы n ?

Формат входных данных

В первой строке задано число n ($1 \leq n \leq 2000$).

Формат выходных данных

В первой строке выведите одно число — количество мирных множеств натуральных чисел силы n .

Примеры

<code>peacefulsets.in</code>	<code>peacefulsets.out</code>
2	1
5	2

Задача У. Психотренинг

Имя входного файла: `psyche.in`
Имя выходного файла: `psyche.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На очередном психологическом тренинге n участников сборов по информатике играют в занимательную игру. Участники игры рассаживаются по кругу и получают номера от 1 до n против часовой стрелки. После этого главный психолог отсчитывает против часовой стрелки k -го участника игры, начиная с первого. Этот участник выходит из круга и может идти на ужин. А остальные продолжают участие в тренинге. Главный психолог отсчитывает ещё k участников, начиная со следующего после выбывшего. Участник, который оказался k -ым, тоже покидает тренинг, и так далее.

Участники сборов решили сесть в круг таким образом, чтобы один вредный тип пошёл ужинать последним. Для этого они хотят установить, какой номер он должен для этого получить. Помогите им.

Формат входных данных

Входные данные содержат два целых числа: n и k ($1 \leq n, k \leq 1\,000\,000$).

Формат выходных данных

Выведите одно число — номер участника, который пойдёт на ужин последним.

Пример

<code>psyche.in</code>	<code>psyche.out</code>
5 3	4

Задача V. Кролик

Имя входного файла: `rabbit.in`
Имя выходного файла: `rabbit.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Как и зайчик, кролик прыгает по прямой просеке, для удобства разделённой на n клеток. Клетки пронумерованы по порядку натуральными числами от 1 до n . Некоторые клетки заболочены: прыгать в них кролик боится. Некоторые другие клетки просеки поросли вкусной зелёной травой: прыгнув на такую клетку, кролик прихватит с собой пучок травы.

Кролик начинает свой путь из клетки с номером 1 и хочет попасть в клетку с номером n , по пути ни разу не провалившись в болото. Конструктивные особенности кролика таковы, что из клетки с номером k он может прыгнуть лишь в клетки с номерами $k + 2$, $k + 3$ и $k + 6$.

Поскольку в клетке с номером n кролика ждёт его крольчиха, главное для него — оказаться там как можно быстрее, то есть сделав как можно меньше прыжков. Если маршрутов с минимальным количеством прыжков несколько, кролик предпочтёт тот

из них, на котором он посетит как можно больше клеток с травой. Если и таких маршрутов несколько, кролика устроит любой из них.

По данной карте просеки найдите оптимальный маршрут для кролика.

Формат входных данных

В первой строке задано число n — количество клеток ($2 \leq n \leq 1000$). Вторая строка состоит из n символов; i -й символ соответствует i -й клетке просеки. Символ «w» обозначает болото, символ «.» — зелёную траву, а символ «.» соответствует клетке без каких-либо особенностей. Гарантируется, что первая и последняя клетки не содержат болот и травы.

Формат выходных данных

В первой строке выведите через пробел два числа — минимальное количество прыжков k , за которое кролик может попасть из первой клетки просеки в последнюю, и максимальное количество клеток с травой t , которые он при этом сможет посетить. Во второй строке выведите через пробел $k+1$ число — номера клеток, которые должен посетить кролик, в порядке их посещения.

Если кролику не удастся оказаться в клетке с номером n , выведите одно число -1 .

Примеры

rabbit.in	rabbit.out
4 ."."	1 0 1 4
5 .w"..	2 1 1 3 5
9 .www.www.	-1

Задача W. Робот

Имя входного файла: robot.in
 Имя выходного файла: robot.out
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 256 мегабайт

Роботу дали задание пройти по лабиринту. Этот лабиринт представляет из себя прямоугольное поле размера $m \times n$ клеток, в котором каждая клетка либо занята сплошной стеной, либо свободна. Каждая клетка описывается парой координат (x, y) , где $1 \leq x \leq m$ и $1 \leq y \leq n$. Особенности конструкции робота таковы, что он из клетки (x, y) может попасть только в клетки $(x-1, y)$, $(x+1, y)$ и $(x, y+1)$, конечно, если они свободны, но не может уменьшить свою координату по y и перейти на клетку $(x, y-1)$.

Робот хочет провести в лабиринте как можно больше времени, так как после прохождения лабиринта его наверняка заставят заниматься чем-то более тяжёлым. Однако, если он остановится хоть на секунду или просто замедлит свое движение, могут

возникнуть подозрения, что он сломался, и тогда ему грозит попадание на свалку металлолома. Поэтому робот хочет выбрать путь, который бы имел наибольшую длину, и пройти по нему с постоянной скоростью.

Задача робота усложняется тем, что ему нельзя возвращаться на клетку, в которой он уже побывал — иначе баг в программе, которая записывает его путь, заставит его крутиться на одном месте, а о последствиях такой ошибки страшно даже подумать!

Помогите роботу найти самый длинный путь по лабиринту, не имеющий самопересечений и такой, что в нём не уменьшается координата y . Путь может начинаться в любой свободной клетке с $y = 1$ и заканчиваться в любой свободной клетке с $y = n$; можно считать, что весь лабиринт огорожен сплошной стеной, и выходить за его пределы нельзя.

Формат входных данных

В первой строке ввода заданы целые числа m и n через пробел ($1 \leq m, n \leq 100$). В последующих n строках содержится по m символов в каждой; j -й символ i -й из этих строк равен «X» (икс большое), если соответствующая клетка занята стеной, и «.» (точка), если она свободна.

Формат выходных данных

Если пути с указанными свойствами не существует, выведите число -1 . В противном случае в первую строку выведите число посещённых роботом клеток k , а в последующие k строк по паре целых чисел x_i и y_i через пробел — координаты клеток пути в порядке их посещения. Число k должно быть максимально; если оптимальных ответов несколько, разрешается вывести любой из них.

Примеры

robot.in	robot.out
2 2	4 1 1 2 1 2 2 1 2
3 4 .X. ... X.X ..X	6 3 1 3 2 2 2 2 3 2 4 1 4
1 1 X	-1

Задача X. Плавные числа

Имя входного файла: smooth.in
Имя выходного файла: smooth.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Назовем натуральное число плавным, если разность любых двух его соседних цифр не превосходит по модулю единицы. Вам необходимо определить количество N -значных плавных чисел.

Формат входных данных

В единственной строке задано одно число N ($1 \leq N \leq 20$).

Формат выходных данных

Выведите одно число — искомое количество.

Пример

smooth.in	smooth.out
1	9

Задача Y. Невозрастающая подпоследовательность

Имя входного файла: subseq.in
Имя выходного файла: subseq.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам требуется написать программу, которая по заданной последовательности находит максимальную невозрастающую её подпоследовательность (то есть такую последовательность чисел $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ ($i_1 < i_2 < \dots < i_k$), что $a_{i_1} \geq a_{i_2} \geq \dots \geq a_{i_k}$ и не существует последовательности с теми же свойствами длины $k + 1$).

Формат входных данных

В первой строке задано число n — количество элементов последовательности ($1 \leq n \leq 239\,017$). В последующих строках идут сами числа последовательности a_i , отделённые друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю $2^{31} - 2$).

Формат выходных данных

Вам необходимо выдать в первой строке число k — длину максимальной невозрастающей подпоследовательности. В последующих строках должны быть выведены (по одному числу в каждой строке) все номера элементов исходной последовательности i_j , образующих искомую подпоследовательность. Номера выводятся в порядке возрастания. Если оптимальных решений несколько, разрешается вывести любое.

Пример

subseq.in	subseq.out
5	3
5 8	2
10 4 1	4
	5

Задача Z. Три последовательности

Имя входного файла: threeseq.in
Имя выходного файла: threeseq.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Даны три последовательности целых чисел. Ваша задача — найти их наибольшую общую подпоследовательность.

Формат входных данных

Входные данные содержат описание трёх последовательностей. Каждая последовательность задаётся в двух строках. Первая из них содержит длину последовательности n ($1 \leq n \leq 100$), а вторая — её элементы (32-битные целые числа).

Формат выходных данных

Первая строка должна содержать длину максимальной общей подпоследовательности. Саму подпоследовательность необходимо вывести во второй строке. Если таких подпоследовательностей несколько, можно вывести любую из них.

Примеры

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