

Задача А. Высота и видимость

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Рассмотрим последовательность чисел a_1, \dots, a_n , которые представляют собой высоты. Назовём радиусом видимости элемента a_i такое максимальное d , что $a_i \geq a_k$ для всех k , таких что $1 \leq k \leq n$ и $|k - i| < d$. Радиус видимости может быть бесконечным.

Требуется для каждого элемента найти его радиус видимости.

Формат входных данных

В первой строке входного файла задано число n , а во второй — числа a_1, \dots, a_n ($1 \leq n \leq 10^6$, $0 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите n чисел — радиус видимости каждого из элементов по порядку. Если радиус видимости бесконечен, выведите вместо него 0.

Примеры

input.txt	output.txt
1 10	0
6 1 5 2 2 1 4	1 0 1 2 1 4

Задача В. Поколения

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В поход пошли два отца и два сына. Хорошо известно, что это может описывать группу из трёх или четырёх человек.

Пусть теперь в поход пошли a отцов и b сыновей. Может ли это быть группа ровно из n человек?

Человек считается отцом, если в этой группе есть его сын. Человек считается сыном, если в этой группе есть его отец. У отца может быть больше одного сына, у каждого сына есть ровно один отец. Отец не может быть потомком сына. Каждый человек в группе — либо отец, либо сын, либо и отец, и сын.

По данным a , b и n определите, возможна ли такая ситуация. Если возможна, то приведите пример.

Формат входных данных

Во входном файле заданы три числа — a , b , n ($1 \leq a, b \leq 100$, $1 \leq n \leq 200$).

Формат выходных данных

Если такая ситуация невозможна, выведите одно число 0.

В противном случае выведите в первой строке число m — количество отношений отцовства в группе. В каждой из следующих m строк выведите числа a_i и b_i , которые означают, что a_i — отец b_i . Люди нумеруются с единицы.

Примеры

input.txt	output.txt
2 2 3	2 1 2 2 3
2 2 4	2 1 2 3 4
2 2 5	0

Задача С. Элитные числа

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Назовём число элитным, если оно делится на каждую цифру в своём десятичном представлении.

Вам дано число x . Вам нужно найти минимальное элитное число, больше либо равное x .

Формат входных данных

Входной файл содержит целое число x ($1 \leq x \leq 10^{10}$).

Формат выходных данных

Выходной файл должен содержать единственное число — минимальное элитное, больше либо равное x .

Примеры

<code>input.txt</code>	<code>output.txt</code>
10	11
7777777777	7777777777
579916	593595

Задача D. Шестиугольники на экране

Имя входного файла: `hexes.in`
Имя выходного файла: `hexes.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Компания разработчик приступила к созданию новой игры, игровое поле в которой представляет собой шестиугольное клетчатое поле. Первая задача — отобразить часть поля на экране компьютера. Чтобы ускорить фазу прототипирования, было решено использовать текстовую графику, вместо обычной.

Шестиугольное клетчатое поле состоит из почти правильных шестиугольников со сторонами длиной n символов. В каждом шестиугольнике верхняя и нижняя стороны состоят из n символов `'_'` (ASCII 95), правая-верхняя и левая-нижняя стороны состоят из n символов `'\'` (ASCII 92), верхняя-левая и нижняя-правая стороны состоят из n символов `'/'` (ASCII 47). Все остальные символы на картинке равны `'.'` (ASCII 46).

Поле является бесконечным, и точка с координатами $(0;0)$ находится в верхнем-левом углу какого-то шестиугольника. На экране отображается прямоугольная часть этого изображения.

Ваша программа должна получает числа x, y — координаты верхнего-левого угла прямоугольника, соответствующего экрану, и w, h — ширина и высота экрана. Выведите изображение, отображающееся на экране.

Формат входных данных

В единственной строке содержатся числа n, x, y, w и h ($1 \leq n \leq 100$; $0 \leq x, y \leq 10^9$, $1 \leq w, h \leq 100$).

Формат выходных данных

Выведите h строк по w символов в каждой — изображение на экране.

Примеры

hexes.in	hexes.out
3 0 2 33 10\...../.....\...../.....\.....___/.....___/.....___/.....\...../.....\...../...../.....\...../.....\...../..... ___/.....___/.....___/.....\...../.....\...../.....\.....\...../.....\...../.....\.....___/.....___/.....___/.....\...../.....\...../...../.....\...../.....\...../.....

Задача Е. Исследование сжатия

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

Многие алгоритмы сжатия данных основываются на поиске часто повторяющихся подстрок во входных данных. Так как обычно невыгодно просматривать все входные данные для поиска повторов, на одном шаге просматривается только некоторый отрезок поиска. Во время разработки нового алгоритма сжатия, юный программист Вася столкнулся со следующей задачей.

Дана строка из n битов, назовем отрезком поиска подстроку длины m исходной строки. Внутри отрезка поиска требуется найти максимальное число вхождений некоторой строки длины l в этот отрезок ($l \leq m$). Требуется вычислить это значение для всех отрезков поиска.

В первом примере длина отрезка поиска равна длине строки, поэтому ответ требуется найти только для одного отрезка поиска. Одна из наиболее часто встречающихся в этом отрезке строк длины 2 это строка 01, которая встречается 5 раз.

Формат входных данных

В первой строке даны два числа l и m . Во второй строке дана строка длины n , состоящая из 0 и 1. $1 \leq l \leq m \leq n \leq 200\,000$, $l \leq 100$.

Формат выходных данных

Выведите $n - m + 1$ целое число — максимальное количество раз, которая строка может встретиться в отрезках поиска, в порядке слева направо.

Примеры

<code>compression.in</code>	<code>compression.out</code>
2 10 0101010101	5
1 3 1110000	3 2 2 3 3

Задача F. Парк

Имя входного файла: `fence-relicts.in`
Имя выходного файла: `fence-relicts.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В некотором регионе есть большой лес, где растут реликтовые деревья. Местное правительство решило создать парк-резервацию с площадью между 0 и S квадратных метров. Парк должен иметь форму прямоугольника со сторонами, параллельными осям координат. Экологические активисты обошли лес и установили, что в нем N реликтовых деревьев, i -ое из которых расположено по координатам (x_i, y_i) (измерения в метрах). Найдите такое расположение и размеры парка, что количество реликтовых деревьев внутри его или на границе максимально возможное.

Формат входных данных

Входной файл содержит два целых числа N и S , за которыми следуют N пар целых чисел x_i, y_i .

Формат выходных данных

Выходной файл должен содержать целые числа x_a, y_a, x_b, y_b — координаты двух противоположных углов парка. Если существует более одного правильного решения, выведите любое.

Примеры

<code>fence-relicts.in</code>	<code>fence-relicts.out</code>
5 100 0 0 10 0 10 10 0 10 15 10	0 0 10 10
3 2 0 0 10 0 20 0	0 0 20 0

Замечание

$1 \leq N \leq 500$, $-10^4 \leq x_i, y_i \leq 10^4$, $1 \leq S \leq 10^9$

Задача G. Дверь и обои

Имя входного файла: `wallpaper.in`
Имя выходного файла: `wallpaper.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Юный строитель Вася должен обклеить стену обоями. Стена имеет форму прямоугольника с шириной w и высотой h метров. В стене есть дверь, которая имеет форму прямоугольника с шириной w_d и высотой h_d метров, расположенная у левого края стены. Обои упакованы в рулоны шириной 1 метр и длиной d метров.

Вася придерживается высоких стандартов качества, поэтому он должен:

- Покрыть обоями всю стену, без зазоров или наложений.
- Приклеивать полосы обоев вертикально, длины $h - h_d$ над дверью и h не над дверью.
- Нарезать все рулоны одинаковым образом, так, чтобы последовательности длин полосок, получившихся из разных рулонов, совпадали.

Ваша программа должна найти минимальное возможное количество рулонов обоев, необходимое, чтобы выполнить задачу.

В первом примере оптимальным является нарезка рулона на полосы с длинами 3, 2 и 1. Но если бы рулоны можно было нарезать разными способами, хватило бы двух рулонов, один из которых нужно было бы нарезать на две полосы длины 3, а другой — на три полосы длины 2.

Формат входных данных

В единственной строке содержатся 5 целых числа w , h , w_d , h_d и d ($1 \leq w_d \leq w \leq 10^9$, $1 \leq h_d \leq h \leq 10^9$, $1 \leq d \leq 2 \cdot 10^9$). Гарантируется, что для таких размеров существует способ поклейки.

Формат выходных данных

Выведите одно число — минимальное необходимое количество рулонов.

Примеры

wallpaper.in	wallpaper.out
5 3 3 1 6	3

Задача Н. Рекламный слоган

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На стадионе с гоночной трассой проходят гонки. Зрители наблюдают за ними с трибуны, которая состоит из h рядов сидений по w в каждом. Некоторые сидения заняты, некоторые свободны.

Маркетинговые исследования показывают, что когда один зритель громко выкрикивает какой-нибудь слоган, его соседи слева, справа, спереди и сзади подхватят его, затем — их соседи, и слоган будет распространяться по трибуне.

Рекламная компания придумала новый слоган и хочет, чтобы его выкрикнуло как можно больше зрителей. Чтобы симулировать «спонтанное» выкрикивание слогана, они наняли агента, который займёт свободное место и прокричит оттуда слоган.

Определите, куда следует посадить его, чтобы как можно больше людей прокричали слоган.

Формат входных данных

В первой строке записаны два числа w и h , а затем — h строк по w символов в каждой. «.» означает свободное место, «#» — занятое. Хотя бы одно место свободно.

$2 \leq w, h \leq 2000$.

Формат выходных данных

Выведите номер строки и столбца того места, куда следует сесть агенту рекламной компании. Если ответов несколько, выведите любой.

Примеры

<code>input.txt</code>	<code>output.txt</code>
<pre>2 2 # .#</pre>	<pre>2 1</pre>
<pre>5 5 ..#.# ..### #..## #.#.# #....</pre>	<pre>2 4</pre>
<pre>6 4 ###### ..###.# ...#.#</pre>	<pre>3 2</pre>

Задача I. Загадочный сад

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

У эксцентричного миллионера Василия Вороткина Третьего есть сад.

Сад — это прямоугольная область размером w метров с запада на восток и h метров с севера на юг. В каждой клетке 1×1 метр находится либо проходимая трава, либо непроходимое препятствие.

Загадочность сада — это число компонент связности из проходимых клеток в нём (клетки считаются соседними, если имеют общую сторону). Василий очень гордится загадочностью своего сада и хочет увеличить её. Для этого он хочет построить дополнительную стену препятствий шириной в 1 метр, которая пересечёт весь сад с севера на юг (т.е. заменить все свободные клетки в некотором столбце на препятствия).

Определите, где нужно построить стену, чтобы загадочность стала как можно больше.

Формат входных данных

В первой строке заданы два числа w и h ($1 \leq w, h \leq 1000$).

Следующие h строк содержат по w символов. Свободные клетки обозначаются символом «.», препятствия — «#».

Формат выходных данных

Выведите два числа: максимально возможная загадочность и номер столбца, который нужно заменить на препятствия, чтобы её достигнуть. Если таких столбцов несколько, выведите наименьший из них. Столбцы нумеруются с единицы слева направо.

Примеры

<code>input.txt</code>	<code>output.txt</code>
5 1	2 2
6 5# #.#.# .#...# ###.##	5 4
1 1 #	0 1

Задача J. Микропроцессор

Имя входного файла: `jump-or-increment.in`
Имя выходного файла: `jump-or-increment.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Молодой программист Вася решил изучить устройство микропроцессора. На первом шаге, он разработал процессор с одним целочисленным регистром с начальным значением 0, и двумя инструкциями:

1. Увеличить значение в регистре на 1 (обозначается буквой *i*);
2. Перейти в начало программы (обозначается буквой *j*).

Вася быстро заметил, что такой набор инструкций не очень полезен, потому что любая программа с хотя бы одной инструкцией *j* будет ходить по циклу бесконечно. Васин друг Петя предложил поменять устройство так, чтобы после выполнения инструкции *j* процессор заменял бы ее специальной *no-op* инструкцией, которая ничего не делает. Теперь, Васин микропроцессор может сделать хоть что-то интересное. Вася хочет исследовать низкоуровневые оптимизации для своего нового микропроцессора. Ваша программа должна вывести кратчайшую непустую программу для Васиного микропроцессора, которая поместит заданное значение в регистр.

Формат входных данных

Входной файл содержит единственное целое число N — желаемое значение в регистре.

Формат выходных данных

Вывод должен содержать единственную строку из букв *i* и *j* — кратчайшую программу. Если таких несколько, выведите любую из них.

Примеры

<code>jump-or-increment.in</code>	<code>jump-or-increment.out</code>
0	<code>j</code>
1	<code>i</code>
17	<code>iiijijjjj</code>

Замечание

$0 \leq N \leq 10^9$.

Задача К. Астротурфинг

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Хорошо известная интернет-компания «Далстолб» создала онлайн-торговую площадку. Купив товар на этой площадке, покупатель может оценить продавца целым числом от 1 до 10. У каждого продавца отображается число оценок n и средняя оценка a .

Если продавец недоволен своим рейтингом, он может либо повысить качество своих товаров, либо поставить себе высокие оценки с фейковых аккаунтов. «Далстолб» знает о таком способе поднять рейтинг, поэтому были введены следующие ограничения: покупатель может поставить только одну оценку одному продавцу, а если его оценка строго больше чем вдвое превышает текущий средний рейтинг, то она проверяется администрацией сайта.

По данным n и a посчитайте минимальное число фейковых аккаунтов, которое потребуется, чтобы поднять среднюю оценку хотя бы до b и не привлечь внимания администрации.

Формат входных данных

Во входном файле заданы целые числа n , a и b ($1 \leq n \leq 10^8$, $1 \leq a \leq b \leq 9$).

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

<code>input.txt</code>	<code>output.txt</code>
4 2 4	6
182736 1 9	8040384
2 8 9	2

Задача L. H₂O и другие формулы

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Когда молодой программист Вася был ещё ещё моложе, он нашёл книгу по органической химии. Он ничего там не понимал, но очень любил рисунки структурных формул химических соединений. Они ему настолько нравились, что он начал рисовать свои собственные.

Годы спустя, прибираясь в комнате, Вася нашёл свои старые рисунки и заинтересовался, на каких из них изображены корректные формулы. Так как рисунков было много, он решил написать программу для решения этой задачи.

Структурная формула химического соединения — это графическое изображение молекулярной структуры, демонстрирующее, каким образом соединены атомы. Атомы обозначаются буквами, а химические связи между ними — отрезками.

Формула представлена во входном файле в виде двумерного массива символов. Допустимые символы: «.» — пустота, «С» — атом углерода, «Н» — атом водорода, «О» — атом кислорода, «|», «\», «-», «/» — химические связи. Связи в корректной формуле изображены при помощи прямых вертикальных, горизонтальных или диагональных линий без пересечений. Атомы в соседних клетках считаются не связанными.

Корректная формула должна содержать хотя бы один атом и должна быть связной (должен существовать путь от каждого атома до каждого, проходящий по связям).

Дополнительно, Вася хочет проверить, что число связей для каждого атома равно валентности соответствующего элемента. Для простоты, он считает, что валентность углерода всегда равна 4, кислорода — 2, а водорода — 1.

Формат входных данных

В первой строке заданы целые числа N и M ($1 \leq N, M \leq 50$). Следующие N строк содержат по M символов каждая и задают графическое изображение формулы.

Формат выходных данных

Выходной файл должен содержать единственную строку: «GOOD», если изображённая формула корректна, «VALENCY», если в формуле корректно всё, кроме валентностей. В противном случае, выведите «BAD».

Примеры

<code>input.txt</code>	<code>output.txt</code>
<pre>2 3 -C .. </pre>	BAD
<pre>3 6 3 6 ..HH.. ./..\ 0---0</pre>	GOOD
<pre>3 6 H--C..\H</pre>	VALENCY

Задача М. Перемежающиеся строки

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Назовём строку из цифр $d_1d_2 \dots d_k$ перемежающейся, если цифры на чётных и нечётных позициях образуют два непересекающихся множества.

Например, 121212 и 01234567 — перемежающиеся, а 11 и 5435 — нет.

Ваша программа должна по строке из цифр найти самую длинную перемежающуюся подстроку. В случае, если решений несколько, выведите самое левое из них.

Формат входных данных

Входной файл содержит строку из цифр длиной от 1 до 30 000 символов.

Формат выходных данных

Выходной файл должен содержать два целых числа P и L — позиция и длина перемежающейся подстроки.

Примеры

<code>input.txt</code>	<code>output.txt</code>
7777	1 1
01234567802345	2 13
9	1 1

Задача N. Иерархия

Имя входного файла: `hierarchical-layout.in`
Имя выходного файла: `hierarchical-layout.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Молодой программист Вася был принят на работы в компанию "WorkMountain". Его первое задание было следующим: реализовать алгоритм, который располагает иерархию контроллеров на форме GUI согласно ограничениями, заданным дизайнером. Несмотря на то, что Васе было дано задание расположить контроллеры в двумерном пространстве, он решил сперва решить задачу для одномерного случая. Контроллер номер i представляется как отрезок на прямой длины минимум A_i , максимум B_i , и имеет C_i контроллеров-детей. Контроллеры могут иметь произвольную вложенность. Ограничения простые:

1. реальная длина контроллера должна быть между минимальной и максимальной включительно;
2. длина каждого контроллера с хотя бы одним потомком должна быть равна сумме длин потомков.

Ваша программа должна для контроллеров (возможно, имеющих детей) и ограничений, определить длину каждого контроллера так, чтобы все ограничения были выполнены.

Формат входных данных

Входной файл содержит целое число N — количество контроллеров, затем идет описание контроллеров. Описание каждого контроллера содержит три целых числа $A_i B_i C_i$, за которыми следуют описания C_i потомков i -го контроллера.

Формат выходных данных

Выведите N целых чисел L_i — длины контроллеров ($A_i \leq L_i \leq B_i$). Если есть несколько решений, выведите любое. Если решений нет, выведите единственно число -1 .

Примеры

<code>hierarchical-layout.in</code>	<code>hierarchical-layout.out</code>
1 10 20 0	10
3 100 100 2 40 50 0 55 55 0	100 45 55
2 5 6 1 7 8 0	-1

Замечание

$$1 \leq N \leq 1000; 1 \leq A_i \leq B_i \leq 10^6; 0 \leq C_i < N; C_1 + C_2 + \dots + C_N = N - 1$$