Задача A. Движение с ускорением

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

В начальный момент времени автомобиль двигался со скоростью V_0. После этого он t_1 секунд двигался с ускорением a_1, а затем — t_2 секунд с ускорением a_2. Необходимо найти максимальную скорость, с которой двигался автомобиль.

Если автомобиль перед началом движения с ускорением a имел скорость V, то через t секунд он будет иметь скорость $V + a \cdot t$. Положительное ускорение означает, что скорость увеличивается, отрицательное — уменьшается.

Формат входного файла

Входной файл содержит пять целых чисел: V_0, t_1, a_1, t_2, a_2 ($0 \le V_0 \le 1000$, $0 < t_1, t_2 \le 10$, $|a_1| \le 10$, $|a_2| \le 10$). Гарантируется, что ни в какой момент времени скорость автомобиля не становится отрицательной.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

accel.in	accel.out
1 2 3 4 5	27
1 2 3 4 -1	7

Задача B. Дигитация

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

В некоторых задачах встречается такая функция от целого неотрицательного числа, как сумма цифр. Например, установить, делится ли число на 9, достаточно просто: нужно лишь определить, делится ли сумма цифр числа на 9.

Рассмотрим обобщение суммы цифр числа — *дигитацию*. А именно, если сумма цифр числа $s(x)$ состоит из одной цифры, то дигитация $d(x)$ равна $s(x)$. Иначе, применяем $s(x)$ к сумме цифр, пока результат не будет состоять из одной цифры. Это можно выразить и короче: $d(x) = d(s(x))$.

Задано число n. Необходимо вычислить дигитацию факториала этого числа: $d(n!) = d(1 \cdot 2 \cdot \ldots \cdot n)$.

Формат входного файла

В первой строке входного файла содержится единственное число n ($1 \le n \le 10^9$).

Формат выходного файла

В выходной файл выведите дигитацию $n!$.

Примеры

digits.in	digits.out
2	2
3	6

Задача C. Зайцы в клетках

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

Всем известен, так называемый, принцип Дирихле, который формулируется следующим образом.

Предположим, что некоторое число кроликов рассажены в клетках. Если число кроликов больше, чем число клеток, то хотя бы в одной из клеток будет больше одного кролика.

В данной задаче мы рассмотрим более общий случай этого классического математического факта. Пусть есть N клеток и M зайцев, которых рассадили по этим клеткам. Вам требуется рассчитать максимальное количество зайцев, которое гарантированно окажется в одной клетке.

Формат входного файла

В первой строке входного файла записаны два натуральных числа N и M $(1 \le N, M \le 10^9)$.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

hares.in	hares.out
2 3	2

Задача D. Ленивый студент

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

Вася — гениальный программист. По крайней мере, он так считает. Поэтому все, что рассказывают на уроках информатики, ему давно известно. Поэтому вместо того, чтобы слушать преподавателя, Вася придумал следующую игру: он берет с собой газету и вычеркивает в тексте все буквы, содержащие «полости». Например, он вычеркивает буквы o и a, но пропускает w и c. Вася считает, что таким образом он тренирует внимательность. Однако ему как-то надо проверять, что он нигде не ошибся. Он считает, что для этого достаточно проверить, что количество вычеркнутых букв верно. Поэтому он просит вас написать программу, определяющую, сколько букв должно быть вычеркнуто в данном тексте.

Формат входного файла

В первой строке входного файла задана строка из строчных латинских букв и пробелов — текст, который Вася брал с собой на урок. Длина строки не превышает 100000 символов. Газета, из которой Вася взял этот текст использует те же шрифты, что и данное условие.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

`lazy.in`	`lazy.out`
`a sample sentence`	`7`
`jing jang walla walla bing bang`	`12`
`abcdefghijklmnopqrstuvwxyz`	`8`

Задача E. Точка и отрезок

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

На плоскости задана точка с координатами (x, y) и отрезок, параллельный одной из координатных осей. Необходимо найти так называемое *манхэттенское расстояние* от точки до этого отрезка.

Это расстояние определяется следующим образом. Рассмотрим две точки A с координатами (x_A, y_A) и B с координатами (x_B, y_B). *Манхэттенское расстояние* между ними равно $\rho_M(A, B) = |x_A - x_B| + |y_A - y_B|$. Определим теперь манхэттенское расстояние от точки A до отрезка BC следующим образом: $\rho_M(A, BC) = \min\limits_{D \in BC} \rho_M(A, D)$ — расстояние до отрезка есть расстояние до ближайшей точки этого отрезка.

Формат входного файла

Первая строка входного файла содержит два целых числа x и y — координаты заданной точки. Вторая строка содержит четыре целых числа: x_1, y_1, x_2 и y_2 — координаты концов заданного отрезка. Все числа во входном файле не превосходят 10^9 по абсолютной величине.

Формат выходного файла

В выходной файл выведите целое число — ответ на задачу. Отметим, что при указанных ограничениях искомое расстояние будет целым числом.

Примеры

point.in	point.out
0 0 0 1 1 1	1
1 1 0 0 0 -10	2

Задача F. Строчечки

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

Строка $A = a_1a_2\dots a_n$ является *подстрокой* строки $B = b_1b_2\dots b_m$, если существует такое число k $(1 \le k \le m - n + 1)$, что $a_1 = b_k$, $a_2 = b_{k+1}$, $\dots$, $a_n = b_{k+n-1}$.

Строка B называется *надстрокой* A, если A является подстрокой B.

Строка $A = a_1a_2\dots a_n$ *лексикографически меньше* строки $B = b_1b_2\dots b_m$, если для некоторого k и для всех $1 \le t \le k$ верно $a_t = b_t$ и либо $a_{k+1} < b_{k+1}$, либо длина A равна k, а длина B больше k.

Даны S и T. Необходимо найти строку, являющуюся одновременно подстрокой S и надстрокой T. Если таких строк несколько, выведите самую длинную из них. Если строк наибольшей длины несколько, выведите лексикографически наименьшую.

Формат входного файла

В первой строке находится строка S $(1 \le |S| \le 4000)$. Во второй строке находится строка T $(1 \le |T| \le 4000)$. Обе строки состоят из строчных латинских букв. Здесь как $|S|$ и $|T|$ обозначены соответственно длины строк S и T.

Формат выходного файла

В выходной файл выведите искомую строку или «`NO SOLUTION`» (без кавычек), если такой строки не существует.

Примеры

strings.in	strings.out
bee be	bee
string nothing	NO SOLUTION

Задача G. Счастливый билетик — 2

Имя входного файла:	`ticket2.in`
Имя выходного файла:	`ticket2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Назовем *билетиком* последовательность цифр длины n. Билетик A называется *счастливым*, если существует число k ($1 \le k < n$) такое, что $\sum\limits_{i=1}^{k} a_i = \sum\limits_{i=k+1}^{n} a_i$. Число k при этом называется *границей счастья*.

Ваша задача — написать программу, определяющую для заданного билетика его наименьшую границу счастья, если она существует.

Формат входного файла

Первая строка входного файла содержит число n ($2 \le n \le 10^6$) — длина билетика A. Во второй строке содержатся цифры a_1, a_2, …, a_n ($0 \le a_i \le 9$), разделенные пробелами.

Формат выходного файла

Если билетик является счастливым, выведите его наименьшую границу счастья, в противном случае выведите «-1».

Примеры

ticket2.in	ticket2.out
4 3 2 1 6	3
4 1 2 3 4	-1

Задача H. Транспортные узлы

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

В стране N-мерике n городов. Некоторые из них соединены двухсторонними дорогами — всего в стране m дорог. Из некоторых городов выходит одна дорога, а некоторые являются настоящими транспортными узлами — из них выходит достаточно много дорог. В этой задаче будем называть город *транспортным узлом*, если из него выходит хотя бы k дорог.

Задано описание дорожной сети N-мерики. Необходимо найти все ее транспортные узлы.

Формат входного файла

Первая строка входного файла содержит число n городов ($1 \le n \le 10000$) и число m дорог ($0 \le m \le 100000$). Каждая из последующих m строк описывает одну дорогу и содержит два числа u и v ($1 \le u, v \le n$, $u \ne v$) — номера городов, соединенных дорогами. Последняя строка входного файла содержит целое число k ($1 \le k \le 10000$).

Каждая дорога упоминается во входном файле не более одного раза.

Формат выходного файла

В первой строке выведите число c транспортных узлов. Во второй строке выведите их номера в порядке возрастания.

Примеры

trans.in	trans.out
2 1 1 2 1	2 1 2
4 3 1 2 1 3 1 4 3	1 1