

Задача А. Бесноватики

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

В лаборатории химической биологии одного из наших ВУЗов выведена новая, чрезвычайно полезная в народном хозяйстве форма жизни: *бесноватики*.

Бесноватики — это аморфные амёбоподобные существа, которые живут колониями на веревках. Химики-биологи обычно выращивают колонию бесноватиков на веревке, натянутой между двумя столбиками. Сразу после рождения бесноватики начинают беспорядочно двигаться вдоль веревки. Если два бесноватика с массами m_1 и m_2 сталкиваются, то они умирают, а на том же месте образуется новый бесноватик с массой $\sqrt{m_1^2 + m_2^2}$.

Химики-биологов чрезвычайно интересует вопрос, до каких пределов может уменьшиться суммарная масса данной колонии бесноватиков.

Формат входного файла

В первой строке входного файла записано число N ($1 \leq N \leq 100$) — количество бесноватиков в колонии. Во второй строке входного файла записано N чисел m_1, \dots, m_N ($0 \leq m_i \leq 10000$) — массы бесноватиков в колонии. Бесноватики расположены на веревке в том же порядке, в каком их массы записаны во входном файле.

Формат выходного файла

В выходном файле должна быть записана округленная до трех цифр после запятой минимальная величина, до которой может уменьшиться суммарная масса колонии бесноватиков.

Примеры

<code>besiki.in</code>	<code>besiki.out</code>
2	2.236
1 2	
2	5.000
3 4	

Задача В. Грузовики

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

Необходимо погрузить несколько ящиков в грузовики. Для этого планируется разделить имеющиеся ящики на две одинаковые группы (в случае нечетного количества получается две группы, в одной из которых на один ящик больше), потом разделить каждую из этих групп аналогичным образом, и так далее, пока мы не получим группы, которые влезают в грузовик. Как только какая-то из получившихся групп влезает в грузовик, производится загрузка ящиков этой группы, и грузовик уезжает. Требуется определить, сколько грузовиков потребуется, чтобы увезти все ящики.

Формат входного файла

Во входном файле заданы два числа n и k ($2 \leq n \leq 10\,000$, $1 \leq k \leq n - 1$) — количество ящиков и емкость грузовика.

Формат выходного файла

Выведите требуемое количество грузовиков.

Пример

<code>trucks.in</code>	<code>trucks.out</code>
14 3	6
15 1	15
1024 5	256

Задача С. День рождения Короля

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

Король Артур издал королевский приказ о выдаче подарков парам, которые сыграют свадьбу в день его рождения. Он выбрал n подарков и приказал своему дворецкому распределить подарки среди этих пар.

Дворецкий приехал во дворец бракосочетания и начал распределять подарки. В каждой паре он дал k_i подарков жениху и l_i подарков невесте, причем известно, что k_i никогда не было меньше l_i , потому что дворецкий ненавидел женщин.

Люди узнали про этот приказ и очередь в дворец бракосочетания выглядела бесконечной. После каждой пары дворецкий думал: “Как много пар! А осталось совсем немного подарков...”, — и всегда давал следующему жениху подарков не больше, чем невесте предыдущей пары, так что выполнялось условие $l_i \geq k_{i+1}$.

Подарки закончились, а очередь выглядела такой же длинной. Дворецкий вышел из дворца и повесился, так как не смог выполнить желание своего Короля. Такая грустная история!

Известно что каждая невеста получила подарок, и что все подарки были распределены. Найдите количество способов, которыми дворецкий мог распределить подарки Короля. Порядок, в котором подарки были распределены, не важен.

Формат входного файла

Во входном файле записано одно число — количество подарков, которые Король Артур решил распределить. Гарантируется, что n всегда меньше 100.

Формат выходного файла

В выходной файл выведите искомое количество способов.

Примеры

<code>king.in</code>	<code>king.out</code>
3	1
4	3
6	6

Задача D. Билетик

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

Витя нашел половину счастливого билетика. Он хочет узнать, сколько счастливых билетиков могло так начинаться. Счастливым считается билетик, у которого сумма первой половины чисел совпадает с суммой второй половины (длина билетика четна).

Формат входного файла

В первой строке входного файла записана первая половина номера счастливого билета. Длина этой половины не превосходит 10 цифр.

Формат выходного файла

В выходной файл выведите одно число — количество возможных счастливых билетов с заданной первой половиной.

Пример

ticket.in	ticket.out
123	28

Задача Е. Инкрементатор

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

Ваша задача — написать программу, моделирующую простое устройство, которое умеет добавлять целые значения к целочисленным переменным.

Формат входного файла

Входной файл состоит из одной или нескольких строк, описывающих операцию. Строка состоит из названия переменной и числа, которое к этой переменной надо добавить. Все числа не превосходят 100 по абсолютной величине. Изначально все переменные равны нулю. Названия переменных состоят из не более чем 100 000 маленьких латинских букв. Размер входного файла не превосходит 2 мегабайт.

Формат выходного файла

Для каждой операции выведите значение соответствующей переменной после выполнения операции на отдельной строке.

Пример

incrementator.in	incrementator.out
a 2	2
b 3	3
a -1	1
c 4	4
b 17	20
xuz 23	23

Задача F. Подстрока

Имя входного файла: substr.in
Имя выходного файла: substr.out
Ограничение по времени: 0.5 секунды
Ограничение по памяти: 64 мегабайта

Ваша задача — найти все вхождения строки в текст.

Формат входного файла

Входной файл начинается со строки, которую нужно искать. Все символы входного файла с кодами меньше 33 или больше 126 необходимо полностью игнорировать. Символы с допустимыми кодами от начала файла до символа «звёздочка» (*) — подстрока, которую нужно найти. Символы от первой звёздочки до конца файла — текст, в котором нужно искать.

Все символы с кодами 33...126 (кроме '*') могут встречаться как в подстроке, так и в тексте. Все пробелы и переводы строк должны быть проигнорированы.

Длина подстроки не превосходит 10 000, длина текста — 200 000 символов.

Формат выходного файла

Выведите в возрастающем порядке позиции всех вхождений данной подстроки. Позиция — это число символов от начала текста до первого символа вхождения, увеличенное на 1. Если вхождение начинается сразу после звёздочки, его позиция равна 1.

Пример

substr.in	substr.out
aa* bb	3
aaa	4

Задача G. Ретрострока

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

Строкой S называется последовательность символов S_1, \dots, S_n , где $|S| = n$ — это длина строки S .

Для любого k ($1 \leq k \leq |S|$) k -м префиксом строки S называется строка S_1, \dots, S_k длины k . Если $k < |S|$, то префикс называется *собственным*.

Аналогично для любого k ($1 \leq k \leq |S|$) k -м суффиксом строки S называется строка $S_{|S|-k+1}, \dots, S_{|S|}$ длины k . Если $k < |S|$, то суффикс также называется *собственным*.

Назовём *числом повторяемости* строки S количество её различных собственных суффиксов, каждый из которых совпадает с префиксом той же длины, что и этот суффикс.

Назовём строку *ретрострокой*, если её число повторяемости строго больше чисел повторяемости всех её собственных префиксов.

Дана строка S . Нужно найти её префикс максимальной длины (не обязательно собственный), являющийся ретрострокой.

Формат входного файла

В первой строке входного файла записана строка S , $1 \leq |S| \leq 1000000$. Строка содержит лишь символы с ASCII-кодом от 33 до 126.

Формат выходного файла

В первой строке выходного файла должен быть выведен префикс S максимальной длины, являющийся ретрострокой.

Пример

retro.in	retro.out
z	z
aabaabaabaabaaba	aabaabaabaaba

Задача Н. Преобразование строчковых функций

Имя входного файла: trans.in
Имя выходного файла: trans.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Для строки S определим Z -функцию следующим

образом: $Z[i] = lcp(S, S[i..|S|])$, где $lcp(S_1, S_2)$ равно длине наибольшего общего префикса строк S_1 и S_2 . Например, для $S = abacabaa$ Z -функция равна $[8, 0, 1, 0, 3, 0, 1, 1]$.

Для строки S определим ее префикс-функцию: $\pi[i] = \max\{k | 0 \leq k < i, S[1..k] = S[i-k+1..i]\}$. Например, для $S = abacabaa$ ее префикс-функция имеет вид: $[0, 0, 1, 0, 1, 2, 3, 1]$.

Для некоторой строки S была посчитана ее Z -функция, а строка S была утеряна. Ваша задача получить ее префикс-функцию по заданной Z -функции.

Формат входного файла

В первой строке входного файла содержится натуральное число N ($1 \leq N \leq 200\,000$), где N — длина S . Во второй строке записана Z -функция строки S .

Формат выходного файла

Выведите N чисел — искомую префикс-функцию.

Пример

trans.in	trans.out
8	0 0 1 0 1 2 3 1
8 0 1 0 3 0 1 1	

Задача I. Обратная префикс-функция

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

Для строки S определим ее префикс-функцию: $\pi[i] = \max\{k | 0 \leq k < i, S[1..k] = S[i-k+1..i]\}$ для всех $1 \leq i \leq N$, где N — длина строки. Например, для $S = abacabaa$ ее префикс-функция имеет вид: $[0, 0, 1, 0, 1, 2, 3, 1]$.

Ваша задача — по заданной префикс-функции восстановить строку. В качестве символов строки разрешается использование M первых строчных букв латинского алфавита.

Формат входного файла

Входной файл состоит из одного или более наборов входных данных.

В первой строке каждого набора записаны два целых числа N, M ($N \geq 1, 1 \leq M \leq 26$). Во второй строке записана последовательность целых чисел $\pi[1], \pi[2], \dots, \pi[N]$. Все числа в последовательности целые неотрицательные, не превосходящие 10^6 .

Сумма значений N по всем наборам не превосходит 10^6 , количество наборов входных данных не превосходит 10^5 .

Формат выходного файла

Выведите в первую строку выходного файла YES, если существует искомое слово, и NO в противном случае. В случае положительного ответа выведите во вторую строку выходного файла выведите искомое слово. Если решений несколько, выведите любое.

Пример

inverse.in	inverse.out
8 3	YES
0 0 1 0 1 2 3 1	abacabaa
1 1	NO
10	

Задача J. Преобразование строковых функций: обратная задача

Имя входного файла: `invtrans.in`
Имя выходного файла: `invtrans.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Для строки S определим Z -функцию следующим образом: $Z[i] = lcp(S, S[i..|S|])$, где $lcp(S_1, S_2)$ равно длине наибольшего общего префикса строк S_1 и S_2 . Например, для $S = abacabaa$ Z -функция равна $[8, 0, 1, 0, 3, 0, 1, 1]$.

Для строки S определим ее префикс-функцию: $\pi[i] = \max\{k | 0 \leq k < i, S[1..k] = S[i-k+1..i]\}$. Например, для $S = abacabaa$ ее префикс-функция имеет вид: $[0, 0, 1, 0, 1, 2, 3, 1]$.

Для некоторой строки S была посчитана ее префикс-функция, а строка S была утеряна. Ваша задача получить ее Z -функцию по заданной префикс-функции.

Формат входного файла

В первой строке входного файла содержится натуральное число N ($1 \leq N \leq 200\,000$), где N — длина S . Во второй строке записана префикс-функция строки S .

Формат выходного файла

Выведите N чисел — искомую Z -функцию.

Пример

invtrans.in	invtrans.out
8	8 0 1 0 3 0 1 1
0 0 1 0 1 2 3 1	

Задача K. Поисковая система

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

По заданию руководства НИИ Исследований Данных Строк Вася участвует в разработке новой интеллектуальной поисковой системы “Abort”, которая по замыслу авторов должна уметь обрабатывать не только обычные поисковые запросы, но также и запросы, в которых пользователь допустил одну опечатку. А именно, пользователь мог случайно поменять местами два соседних символа, удалить один из символов, вставить какой-то один лишний символ, случайно набрать вместо одного символа какой-то другой, а также он мог и не сделать ошибки в запросе. Например, вместо `baba` пользователь мог набрать `abba`, `aba`, `babca`, `cabab` или `baba`. Если же пользователь набрал в запросе `abab`, `saab` или `bababa`, то он сделал более одной опечатки, и система не должна считать этот запрос ошибочно набранным запросом `baba`.

Вася поручил вам написать программу, которая находила бы количество мест в тексте, в которых встречается данное слово, если при наборе его, возможно, была допущена такая опечатка.

Формат входного файла

В первой строке входного файла задается количество запросов, которые необходимо обработать. Каждый запрос состоит из двух непустых строк, первая из которых — текст, в котором нужно искать ответ на запрос, а вторая — собственно набранный пользователем запрос. Все строки

состоят из маленьких латинских букв. Суммарная длина всех строк не превосходит 500 000.

Формат выходного файла

Для каждого из запросов вывести количество позиций в тексте, в которых могло встречаться слово из запроса, возможно, набранное с какой-то из вышеописанных ошибок.

Пример

search.in	search.out
4	10
abacabadabacaba	3
baba	0
abababbababababbabba	2
abaca	
botva	
poisk	
yabeda	
baed	