

Задача А. Имя для питона

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Питоны очень внимательно относятся к выбору своих имён. Сначала для каждого детёныша его мать придумывает последовательность маленьких латинских букв по известному только питонам правилу. Однако в результате часто получается труднопроизносимое слово. Поэтому в полученной последовательности труднопроизносимые буквы надо заменить на любимую питонами букву *A*. Но это не всегда возможно, так как древний питоний закон гласит, что нельзя менять буквы, стоящие на позициях $C[i]$ (по древней традиции, позиции нумеруются, начиная с нуля).

Формат входных данных

В первой строке записано слово S ($1 \leq \text{len}(S) \leq 100$), придуманное матерью питона и состоящее из маленьких латинских букв. Во второй строке через пробел перечислены буквы $B[i]$ ($1 \leq i \leq 25$), труднопроизносимые для питонов. В третьей строке через пробел перечислены позиции $C[i]$ ($0 \leq C[i] < \text{len}(S)$, $0 \leq i \leq \text{len}(S) \leq 100$), изменения в которых запрещены древним законом. В последней строке находится любимая питонами буква A ($A \neq B[i]$ при любом i).

Формат выходных данных

Выведите одну строку — имя питона после преобразований.

Примеры

<code>stdin</code>	<code>stdout</code>
<pre>james m r t e s</pre>	<pre>jasss</pre>
<pre>niishka k i 1 2 h</pre>	<pre>niishha</pre>

Задача В. Реверс удавов

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На каждом удаве из стаи написано его имя. Имя удава написано маленькими латинскими буквами от головы к хвосту. Все удавы из стаи ползут друг за другом, ведь так легче ползти. Иногда вожак стаи дает команду «Реверс». В этом случае каждый удав стаи разворачивается, и стая начинает ползти в противоположном направлении. Название стаи можно прочитать, если читать от головы удава, ползущего первым, к хвосту последнего. При этом название может измениться после команды «Реверс». Имена же удавов не меняются.

Формат входных данных

Первая строка содержит одно число $1 \leq N \leq 100\,000$ — количество удавов. В следующих N строках написаны имена удавов в том порядке, в котором они ползут. Имя удава — строка, содержащая не более 10 маленьких латинских букв.

Формат выходных данных

Выведите единственную строку — название стаи после команды «Реверс».

Примеры

<code>stdin</code>	<code>stdout</code>
3 abc def ghi	ghidefabс

Задача С. Шифр Юлия

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Юлий Цезарь использовал свой способ шифрования текста. Каждая буква заменялась на следующую по алфавиту через K позиций по кругу. Необходимо по заданной шифровке определить исходный текст.

Формат входных данных

В первой строке дана шифровка, состоящая из заглавных латинских букв и не превышающая по длине 255 символов. Во второй строке задано число K ($1 \leq K \leq 10$).

Формат выходных данных

Требуется вывести результат расшифровки.

Примеры

<code>stdin</code>	<code>stdout</code>
XPSE 1	WORD

Задача D. Палиндром

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Палиндром — это строка, которая одинаково читается слева направо и справа налево. Составьте программу, которая проверяет, является ли заданный текст палиндромом. Не забудьте, что при чтении пробел никак не произносится.

Формат входных данных

Дана строка S ($|S| \leq 255$), состоящая из строчных латинских букв и пробелов. Под $|S|$ подразумевается длина строки.

Формат выходных данных

Требуется вывести YES, если текст является палиндромом, NO если не является.

Примеры

<code>stdin</code>	<code>stdout</code>
<code>palindrom</code>	<code>NO</code>
<code>a roza upala na lapu azora</code>	<code>YES</code>

Задача E. Архивация

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 8 секунды
Ограничение по памяти: 256 мегабайт

Алгоритм сжатия RLE устроен по следующему принципу. Файл рассматривается как последовательность бит — нулей и единиц. Результатом его работы являются числа, обозначающие длины последовательных непрерывных цепочек нулей и единиц. Максимальная длина цепочки ограничена числом K . Поэтому, если имеется непрерывная цепочка одинаковых бит длины, большей K , то она разбивается на несколько цепочек, каждая из которых, кроме, возможно, последней, имеет длину K , и между длинами частей цепочки вставляется ноль. Для определенности считается, что последовательность бит всегда начинается с цепочки нулей (возможно, нулевого размера). То есть, если последовательность начинается с единиц, то в начале сжатых данных будет стоять ноль, обозначающий пустую последовательность нулей.

Формат входных данных

На первой строке дано число K ($1 \leq K \leq 1000$). На второй строке дана последовательность нулей и единиц, в конце которой находится число 2. Длина последовательности — натуральное число, не превышающее 10^6 .

Формат выходных данных

Сжатая последовательность (см. примеры).

Примеры

<code>stdin</code>	<code>stdout</code>
3 0 0 0 0 0 0 0 0 0 0 0 0 2	3 0 3 0 3 0 3 0 1
3 0 0 0 1 1 1 0 0 2	3 3 2

Задача F. Игра в футбол

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На чемпионате ЛКШ по футболу теперь применяется новый способ приветствия команд. Для этого команды выстраиваются в одну линию. Капитаны команд встают рядом, а команды выстраиваются по сторонам от своих капитанов. Капитаны делают шаг навстречу друг другу, здороваются и отходят в концы своих команд. На некоторых матчах между капитанами может стоять судья, но на процедуру приветствия это не влияет. Напишите программу, которая выведет расположение людей на поле после приветствия капитанов.

Формат входных данных

Каждый человек, находящийся на поле, задаётся номером, написанном на его майке. Для удобства будем считать, что у судьи тоже есть номер на майке. В единственной строке вводится последовательность чисел-номеров на майках людей (номера на майках — натуральные числа ≤ 1000 , все числа разделены пробелами).

Формат выходных данных

Выведите последовательность чисел — номеров на майках людей, находящихся на поле, после приветствия капитанов

Примеры

<code>stdin</code>	<code>stdout</code>
5 3 2 9 2	3 5 2 2 9

Замечание

Сначала вводятся номера игроков первой команды, заканчивая капитаном (5 3), затем номер судьи (2), затем игроки второй команды, начиная с капитана (9 2). Судьи может не быть, кол-во игроков в командах одинаковое.

Задача G. Юбилей Винни-Пуха

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Вот и наступил долгожданный Юбилей Винни-Пуха. В волшебный лес на праздник собралось множество гостей. В том числе Винни-Пух пригласил к себе друзей из других галактик. К сожалению, когда он посылал приглашения, он совсем забыл, что на планете, где живут его друзья инопланетяне, все читают не слева направо, а справа налево. Винни-Пух понимает, что к Юбилею они уже не прилетят, но медвежонок не унывает. Он хочет проверить, правда ли, что дата его Юбилея, прочитанная справа налево, тоже существует, и инопланетяне прилетят в другой день. Помогите Винни-Пуху определить, ждать ли ему в гости инопланетных друзей.

Формат входных данных

Входной файл содержит дату Юбилея Винни-Пуха в формате `dd.mm.gggg`. Гарантируется, что дата корректна.

Формат выходных данных

В выходной файл нужно вывести `YES`, если дата, читающаяся справа налево корректна, и `NO` в противном случае.

Примеры

<code>stdin</code>	<code>stdout</code>
<code>23.02.2002</code>	<code>YES</code>
<code>20.02.2023</code>	<code>NO</code>

Замечание

В первом примере при чтении справа налево получается число `20.02.2032`, что является корректной датой, а во втором примере получается дата `32.02.2002`, которая не является корректной, так как ни в одном месяце года нет 32-ого дня.

Задача Н. Удаление лишних пробелов

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дана строка. Напишите программу, которая удалит из этой строки все лишние пробелы. Пробел будем считать лишним, если:

- он находится в самом начале строки, до самого первого слова;
- он находится в конце строки, после самого последнего слова;
- несколько пробелов расположены между двумя словами (проще говоря, если слова разделены более чем одним пробелом, тогда все пробелы кроме одного — лишние)

Формат входных данных

Дана строка S ($0 \leq |S| \leq 255$). Строка содержит только строчные латинские буквы и пробелы.

Формат выходных данных

Требуется вывести строку без лишних пробелов.

Примеры

<code>stdin</code>	<code>stdout</code>
Alexandr Sergeevich Pushkin	Alexandr Sergeevich Pushkin