

Задача А. Префикс-функция

Имя входного файла: `prefix-function.in`
Имя выходного файла: `prefix-function.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Дана непустая строка S , длина которой N не превышает 10^6 . Будем считать, что элементы строки нумеруются от 1 до N .

Требуется для всех i от 1 до N вычислить её префикс-функцию $\pi[i]$.

Формат входных данных

Одна строка длины N , $0 < N \leq 10^6$, состоящая из маленьких латинских букв.

Формат выходных данных

Выведите N чисел — значения префикс-функции для каждой позиции, разделённые пробелом.

Примеры

<code>prefix-function.in</code>	<code>prefix-function.out</code>
abracadabra	0 0 0 1 0 1 0 1 2 3 4

Задача В. А-функция

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

Дана строка S , состоящая из N символов. Определим функцию $A(i)$ от первых i символов этой строки следующим образом:

$A(i)$ = максимально возможному k такому, что равны следующие две строки: $S[1] + S[2] + S[3] + \dots + S[k] = S[i] + S[i-1] + S[i-2] + \dots + S[i-k+1]$, где $S[i]$ — i -ый символ строки S , а знак $+$ означает, что символы записываются в строчку непосредственно друг за другом.

Напишите программу, которая вычислит значения функции A для заданной строки для всех возможных значений i от 1 до N .

Формат входных данных

В первой строке входного файла записано одно число N . $1 \leq N \leq 200\,000$. Во второй строке записана строка длиной N символов, состоящая только из больших и/или маленьких латинских букв.

Формат выходных данных

В выходной файл выведите N чисел — значения функции $A(1), A(2), \dots, A(N)$.

Примеры

<code>afunction.in</code>	<code>afunction.out</code>
5 aabaa	1 2 0 1 5

Задача С. Циклическая строка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Строка S была записана много раз подряд, после чего из получившейся строки взяли подстроку и дали вам. Ваша задача определить минимально возможную длину исходной строки S .

Формат входных данных

На вход программы поступает строка, которая содержит только латинские буквы, длина строки не превышает 50 000 символов.

Формат выходных данных

Требуется вывести одно число — ответ на вопрос задачи.

Примеры

стандартный ввод	стандартный вывод
z	1
сac	2

Задача D. Последнее слово Джека

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

Джек недавно прочитал на заборе интересное и новое для него слово. Оно настолько понравилось Джеку, что он захотел сам придумать ещё какое-нибудь интересное слово. Но только ничего у него не вышло — все придуманные им слова состояли из префиксов исходного слова и поэтому не приносили радости. Он стал придумывать всё более и более длинные слова, но ни одно из них не было оригинальным. . .

И вот настало время Джеку сказать своё последнее слово.

Формат входных данных

Первая строка содержит интересное слово, которое было написано на заборе. Вторая строка содержит последнее слово Джека. Длины слов не превосходят 75 000, слова непустые и состоят из строчных латинских букв.

Формат выходных данных

Если Джек так ничего и не придумал своего, выведите первой строкой `No`. В этом случае покажите Джеку, как разбить его последнее слово на несколько частей, каждая из которых является исходным словом или его непустым префиксом — выведите все эти части во второй строке, разделяя их пробелом. Если же такого разбиения нет, и последнее слово было за Джеком, выведите единственной строкой `Yes`.

Примеры

<code>prefix.in</code>	<code>prefix.out</code>
<code>abracadabra</code> <code>abrabracada</code>	<code>No</code> <code>abr abracada</code>
<code>abracadabra</code> <code>arbadacarba</code>	<code>Yes</code>

Задача Е. Дендрохронология

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дендрохронология — метод, в основе которого лежит закон природы, согласно которому каждый год толщина дерева увеличивается на одно кольцо. Толщина каждого кольца зависит от погоды в год его образования. У деревьев, растущих в одном и том же месте и климате, толщины колец примерно одинаковы. По числу колец можно определить возраст дерева, а по толщине колец — погодные условия каждого года, в который это дерево росло.

Используя перекрёстную датировку, можно построить дендрохронологическую шкалу, увязав друг с другом следующие одно за другим поколения деревьев, годы жизни которых перекрываются. Так можно определить погодные условия от некоторого момента до настоящего времени, а по построенной шкале — датировать деревянные предметы, сохранившиеся с давних лет.

Однажды в древнем городе, основанном в болотистой местности, археологи обнаружили n слоёв мостовых, положенных один над другим и состоящих из отлично сохранившихся брёвен. Первый слой был положен в момент основания города, а последний — в текущем году. При этом в один год могли быть положены и несколько слоёв мостовых. Для одного бревна из каждого слоя археологи определили последовательность толщин годовых колец.

Слои перечислены в хронологическом порядке. Дерево из следующего слоя начало расти не раньше дерева из предыдущего слоя и было срублено не раньше него. Если годы жизни двух деревьев пересекались, последовательности толщин их колец в общие годы должны совпадать.

Если толщины колец последних k лет жизни одного дерева совпадают с первыми k кольцами жизни второго дерева, то деревья могли жить одновременно эти k лет. Точное расположение шкал по годам неизвестно. Среди всех допустимых вариантов археологов интересует тот, при котором город существует наименьшее возможное число лет. Иными словами, для каждой пары соседних слоёв следует выбрать максимально возможный общий период жизни. Возраст города считается, как число лет между укладкой первого и последнего слоя мостовых.

Если совместить шкалы с перекрытием длины хотя бы 1 невозможно, их следует расположить вплотную: первое кольцо следующего дерева относится к году, следующему сразу после последнего года жизни предыдущего дерева.

Помогите археологам определить наименьший возможный возраст города.

Формат входных данных

Первая строка содержит натуральное число n ($1 \leq n \leq 10^6$) — количество слоёв мостовых.

Следующие n строк описывают в хронологическом порядке дендрохронологическую шкалу для каждого бревна, взятого из отдельного слоя мостовой. Каждая строка состоит из цифр от 1 до 9, соответствующих толщинам годовых колец в миллиметрах от первого до последнего года жизни дерева.

Общее число годовых колец не превосходит 10^6 .

Формат выходных данных

Выведите одно целое число — наименьший возможный возраст города.

Примеры

стандартный ввод	стандартный вывод
4 123123123 3123111321 13212212 212331	13
5 1 2 1 2 1	4
2 12121 121	0

Замечание

В первом примере строки можно совместить следующим образом:

```
123123123
.....3123111321
.....13212212
.....212331
```

На основании этих данных можно построить дендрохронологическую шкалу для последних 22 лет. Между укладкой первого и последнего слоёв прошло 13 лет.

Во втором примере расположение будет таким.

```
1
.2
..1
...2
....1
```

В третьем примере расположение будет таким.

```
12121
..121
```

Задача F. Шифр Цезаря

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Шифр Цезаря — простой исторический шифр, разновидность шифра подстановки. В этом шифре последовательность символов алфавита циклически сдвигается на фиксированное значение k . Например, при $k = 3$ буква А заменяется на букву Г, буква Б — на букву Д, а буква Я — на букву В.

Иными словами, если в алфавите n символов и они обозначены числами от 0 до $n - 1$, то буква i при шифровании заменяется на букву $(i + k) \bmod n$. Например, при $n = 5$ и $k = 2$, применяя шифр к сообщению «3 2 1 4 0», получится сообщение «0 4 3 1 2».

Вы перехватили q секретных сообщений, которые были зашифрованы шифром Цезаря. Каждое сообщение было зашифровано независимо, поэтому величина сдвига k может быть различной. Вам нужно расшифровать их все. Задача нелёгкая, но ваш резидент смог для каждого сообщения узнать какую-то подстроку, которая содержалась в исходном сообщении.

Формат входных данных

В первой строке входных данных записаны число перехваченных сообщений q ($1 \leq q \leq 1000$) и размер алфавита n ($1 \leq n \leq 10^8$).

В следующих $3q$ строках содержатся описания перехваченных сообщений, каждое из которых состоит из трёх строк.

Первая строка описания сообщения содержит числа x_i и y_i — длину i -го сообщения и длину известной подстроки его расшифровки ($1 \leq y_i \leq x_i \leq 10^6$).

Вторая строка описания сообщения содержит x_i чисел через пробел — символы зашифрованного сообщения. Символы принимают значения от 0 до $n - 1$.

Третья строка описания сообщения содержит y_i чисел через пробел — известная подстрока расшифровки i -го сообщения.

Сумма всех значений x_i не превышает 10^6 .

Формат выходных данных

В строке номер i выведите расшифрованный вариант сообщения с номером i . Если подходящих расшифровок несколько, выведите любую из них. Гарантируется, что существует хотя бы один вариант расшифровки каждого сообщения.

Примеры

стандартный ввод	стандартный вывод
1 5 5 3 4 3 2 0 1 1 0 3	2 1 0 3 4
1 11 10 4 10 6 0 0 1 5 6 0 0 7 1 6 6 7	5 1 6 6 7 0 1 6 6 2

Замечание

Во втором примере зашифровано исходное сообщение с $k = 5$.

Задача G. Палиндромы

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

Строка называется палиндромом, если она одинаково читается как слева направо, так и справа налево. Например, `abba` — палиндром, а `омах` — нет. Для строки α будем обозначать $\alpha[i..j]$ ее подстроку длины $j - i + 1$ с i -й по j -ю позицию включительно (позиции нумеруются с единицу). Для заданной строки α длины N ($1 \leq N \leq 100\,000$) требуется подсчитать число q пар (i, j) , $1 \leq i < j \leq n$, таких что $\alpha[i..j]$ является палиндромом.

Формат входных данных

Входной файл содержит одну строку α длины N , состоящую из маленьких латинских букв.

Формат выходных данных

В выходной файл выведите искомое число q .

Примеры

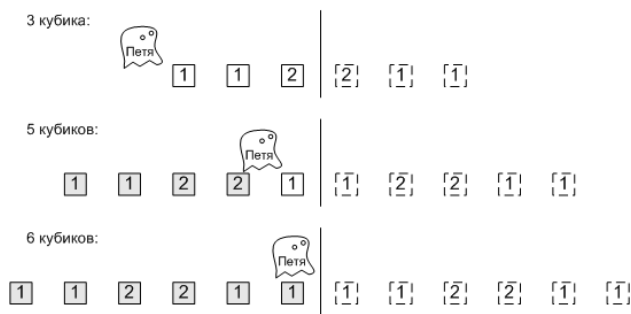
<code>palindrome.in</code>	<code>palindrome.out</code>
<code>aaa</code>	3
<code>abba</code>	2
<code>омах</code>	0

Задача Н. Кубики

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

Привидение Петя любит играть со своими кубиками. Он любит выкладывать их в ряд и разглядывать своё творение. Однако недавно друзья решили подшутить над Петей и поставили в его игровой комнате зеркало. Ведь всем известно, что привидения не отражаются в зеркале! А кубики отражаются.

Теперь Петя видит перед собой N цветных кубиков, но не знает, какие из этих кубиков настоящие, а какие — всего лишь отражение в зеркале.



Помогите Пете! Выясните, сколько у него может быть кубиков. Петя видит отражение всех кубиков в зеркале и часть кубиков, которая находится перед ним. Часть кубиков может быть позади Пети, их он не видит.

Формат входных данных

Первая строка входного файла содержит два целых числа: N ($1 \leq N \leq 100\,000$) и количество различных цветов, в которые могут быть раскрашены кубики, — M ($1 \leq M \leq 100\,000$). Следующая строка содержит N целых чисел от 1 до M — цвета кубиков.

Формат выходных данных

В выходной файл выведите в порядке возрастания все такие K , что у Пети может быть K кубиков.

Примеры

<code>cubes.in</code>	<code>cubes.out</code>
6 2 1 1 2 2 1 1	3 5 6

Задача I. Неточное совпадение

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

Даны строки p и t . Требуется найти все вхождения строки p в строку t в качестве подстроки с точностью до возможного несовпадения одного символа.

Формат входных данных

Первая строка входного файла содержит p , вторая — t ($1 \leq |p|, |t| \leq 10^6$). Строки состоят из букв латинского алфавита.

Формат выходных данных

В первой строке выведите количество вхождений строки p в строку t . Во второй строке выведите в возрастающем порядке номера символов строки t , с которых начинаются вхождения p . Символы нумеруются с единицы.

Примеры

<code>inexact-matching.in</code>	<code>inexact-matching.out</code>
aaaa	4
Caabdaaaa	1 2 6 7

Задача J. Движущаяся строка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Поликарп нашёл строку s и перестановку p . Их длины оказались одинаковы и равны n .

Перестановка из n элементов — это массив длины n , в котором каждое целое число от 1 до n встречается ровно по одному разу. Например, $[1, 2, 3]$ и $[4, 3, 5, 1, 2]$ — это перестановки, но $[1, 2, 4]$, $[4, 3, 2, 1, 2]$ и $[0, 1, 2]$ — это не перестановки.

За одну операцию он может умножить s на p , то есть заменить строку s на строку new , в которой для каждого i от 1 до n верно, что $new_i = s_{p_i}$. Например, при $s = wmbc$ и $p = [3, 1, 4, 2]$, после применения операции строка превратится в $s = s_3s_1s_4s_2 = bwcm$.

Поликарпу стало интересно, через сколько операций строка впервые вернётся к своему первоначальному виду. Так как это может занять слишком много времени, он просит вашей помощи в этом вопросе.

Можно доказать, что искомое количество операций всегда существует. Оно может оказаться очень большим, используйте 64-битный целочисленный тип.

Формат входных данных

В первой строке входных данных записано целое число t ($1 \leq t \leq 5000$) — количество наборов входных данных в тесте.

Первая строка каждого набора содержит целое число n ($1 \leq n \leq 200$) — длину строки и перестановки.

Вторая строка каждого набора содержит строку s длины n , состоящую из строчных латинских букв.

Третья строка каждого набора содержит n целых чисел — перестановку p ($1 \leq p_i \leq n$), все p_i различны.

Формат выходных данных

Выведите t строк, каждая из которых содержит ответ на соответствующий набор входных данных. В качестве ответа выведите единственное число — минимальное количество операций, после которого строка s станет такой же, какой была до их применения.

Пример

стандартный ввод	стандартный вывод
3	1
5	6
ababa	12
3 4 5 2 1	
5	
ababa	
2 1 4 5 3	
10	
codeforces	
8 6 1 7 5 2 9 3 10 4	

Замечание

В первом наборе входных данных применение операции не изменяет строку, поэтому она станет равной самой себе после 1 операции.

Во втором наборе входных данных строка будет меняться следующим образом:

- $s = babaa$
- $s = abaab$

- $s = \text{baaba}$
- $s = \text{abbaa}$
- $s = \text{baaab}$
- $s = \text{ababa}$