

## Задача А. Сравнения подстрок

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Дана строка  $s$ . Ответьте на  $m$  запросов вида: равны ли подстроки  $s[a..b]$  и  $s[c..d]$ .

### Формат входных данных

В первой строке ввода записана строка  $s$  ( $1 \leq |s| \leq 10^5$ ).

Во второй строке записано целое число  $m$  — количество запросов ( $0 \leq m \leq 10^5$ ).

В следующих  $m$  строках четверки чисел  $a, b, c, d$  ( $1 \leq a \leq b \leq |s|, 1 \leq c \leq d \leq |s|$ ).

### Формат выходных данных

Выведите  $m$  строк. Выведите **Yes**, если подстроки совпадают, и **No** иначе.

### Пример

стандартный ввод	стандартный вывод
trololo	Yes
3	Yes
1 7 1 7	No
3 5 5 7	
1 1 1 5	

## Задача В. Неточное совпадение

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
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 —  $t$  ( $1 \leq |p|, |t| \leq 10^5$ ). 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.

### Пример

стандартный ввод	стандартный вывод
aaaa	4
Caaabdaaaa	1 2 6 7

## Задача С. Профили-двойники

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 1 second  
Ограничение по памяти: 256 megabytes

Вам предложили работу в компании, разрабатывающей крупную социальную сеть. Ваше первое задание связано с поиском профилей, с большой вероятностью принадлежащих одному и тому же пользователю.

В социальной сети зарегистрировано  $n$  профилей, пронумерованных от 1 до  $n$ . Некоторые пары среди них являются друзьями (отношение «быть друзьями» взаимно, то есть если  $i$  является другом  $j$ , то и  $j$  является другом  $i$ ). Будем говорить, что профили  $i$  и  $j$  ( $i \neq j$ ) являются *двойниками*, если для любого профиля  $k$  ( $k \neq i$ ,  $k \neq j$ ), верно одно из двух утверждений: либо  $k$  дружит и с  $i$ , и с  $j$ , либо  $k$  не дружит ни с одним из них. При этом  $i$  и  $j$  могут как дружить между собой, так и не дружить.

Вам нужно посчитать количество различных неупорядоченных пар  $(i, j)$ , таких что профили  $i$  и  $j$  — двойники. Обратите внимание, что пары неупорядоченные, то есть пары  $(a, b)$  и  $(b, a)$  считается одинаковыми.

### Формат входных данных

В первой строке записано два целых числа  $n$  и  $m$  ( $1 \leq n \leq 10^6$ ,  $0 \leq m \leq 10^6$ ), разделенных пробелом — количество профилей и количество пар друзей соответственно.

В следующих  $m$  строках записаны описания пар друзей в формате « $v$   $u$ », где  $v$  и  $u$  ( $1 \leq v, u \leq n$ ,  $v \neq u$ ) — номера профилей, являющихся друзьями. Гарантируется, что каждая неупорядоченная пара друзей встречается не более одного раза и никакой профиль не является другом самого себя.

### Формат выходных данных

Выведите единственное целое число — количество неупорядоченных пар профилей, являющихся двойниками.

### Примеры

<code>stdin</code>	<code>stdout</code>
3 3 1 2 2 3 1 3	3
3 0	3
4 1 1 3	2

## Задача D. Подпалиндромы

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

Строка называется палиндромом, если она читается одинаково как слева направо, так и справа налево. Например, строки "abba", "kek" являются палиндромами.

Дана строка. Ее подстрокой называется некоторая непустая последовательность подряд идущих символов. Напишите программу, которая определит, сколько подстрок данной строки является палиндромами.

### Формат входных данных

Вводится одна строка, состоящая из маленьких латинских букв. Длина строки не превышает 100 000 символов.

### Формат выходных данных

Выведите одно число – количество подстрок данной строки, являющихся палиндромами.

### Примеры

стандартный ввод	стандартный вывод
aaa	6
aba	4

## Задача Е. Хеш-код

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

Согласно документации стандартной библиотеки Java, хеш-код для строки вычисляется как:

$$s[0] \cdot 31^{n-1} + s[1] \cdot 31^{n-2} + \dots + s[n-1]$$

Где  $s[i]$  — это  $i$ -й символ строки,  $n$  длина строки. Для вычисления используются целые 32-битные числа в форме дополнения до двух.

Вы собираетесь взломать сервера одной известной компании. Чтобы вы смогли выполнить атаку, вам нужны  $k$  различных строк, которые имеют одинаковые хеш-коды. К сожалению, сервера этой компании не принимают строки запроса, содержащие буквы отличные от английских букв нижнего и верхнего регистров.

Напишите программу, которая генерирует такие строки.

### Формат входных данных

Первая строка содержит целое число  $k$  — количество необходимых строк запроса для генерации ( $2 \leq k \leq 1000$ ).

### Формат выходных данных

Необходимо вывести  $k$  различных непустых строк, каждая из которых имеет длину не более 1000 символов. Все строки должны состоять только из английских букв верхнего или нижнего регистров и иметь равный хеш-код.

### Пример

стандартный ввод	стандартный вывод
4	edHs mENAGeS fEHs edIT

## Задача F. С днём рождения!

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

В этой задаче требуется найти коллизии при полиномиальном хешировании строк, состоящих из маленьких букв английского алфавита.

*Полиномиальный хеш* строки имеет два параметра: множитель  $p$  и модуль  $q$ . Для пустой строки  $\varepsilon$  значение хеш-функции  $h(\varepsilon) = 0$ , а для любой строки  $S$  и любого символа  $c$  хеш-функция рекуррентно определяется как  $h(S + c) = (h(S) \cdot p + \text{code}(c)) \bmod q$ . Здесь  $\text{code}(c)$  — это ASCII-код символа  $c$ . Как известно, коды маленьких букв английского алфавита идут подряд:  $\text{code}(\text{'a'}) = 97$ ,  $\text{code}(\text{'b'}) = 98$ , ...,  $\text{code}(\text{'z'}) = 122$ . Можно выписать и нерекуррентную формулу: если строка  $S = s_1 s_2 \dots s_n$ , то  $h(S) = (\text{code}(s_1) \cdot p^{n-1} + \text{code}(s_2) \cdot p^{n-2} + \dots + \text{code}(s_n) \cdot p^0) \bmod q$ .

По заданным числам  $p$  и  $q$  найдите две различные непустые строки  $A$  и  $B$  такие, что  $h(A) = h(B)$ .

### Формат входных данных

Первая строка ввода содержит два целых числа  $p$  и  $q$ , разделённых пробелом — параметры функции хеширования ( $0 < p < q \leq 2 \cdot 10^9 + 9$ ).

### Формат выходных данных

В первых двух строках выведите две различные непустые строки  $A$  и  $B$ , для которых  $h(A) = h(B)$ . Строки должны состоять исключительно из маленьких букв английского алфавита (ASCII-коды 97–122) и иметь длину от 1 до 100 000 символов. Заметим, что длины строк не обязательно должны совпадать. Если возможных ответов несколько, разрешается вывести любой из них.

### Примеры

стандартный ввод	стандартный вывод
31 47	aa bq
2 1000000007	rp nx
179 1000000009	weeoutf hronndauw

## Задача G. 4 элемента с суммой 0

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.6 секунд  
Ограничение по памяти: 256 мегабайт

Дано четыре списка целых чисел  $A$ ,  $B$ ,  $C$  и  $D$  одинаковой длины  $n$ . Посчитайте количество четверок целых чисел  $(x, y, z, t)$  ( $1 \leq x, y, z, t \leq n$ ) таких, что  $A_x + B_y + C_z + D_t = 0$ .

### Формат входных данных

Первая строка содержит целое число  $n$  — количество чисел в списках ( $1 \leq n \leq 2500$ ). Следующие  $n$  строк содержат четверки целых чисел  $A_i, B_i, C_i, D_i$  —  $i$ -е элементы каждого из четырех списков. Все элементы по абсолютному значению не превосходят  $2^{28}$ .

### Формат выходных данных

В единственной строке выведите ответ на данную задачу.

### Пример

стандартный ввод	стандартный вывод
6 -45 22 42 -16 -41 -27 56 30 -36 53 -37 77 -36 30 -75 -46 26 -38 -10 62 -32 -54 -6 45	5

### Замечание

Четверки элементов, образующие в примере сумму 0, следующие:  $(-45; -27; 42; 30)$ ,  $(26; 30; -10; -46)$ ,  $(-32; 22; 56; -46)$ ,  $(-32; 30; -75; 77)$ ,  $(-32; -54; 56; 30)$

## Задача Н. Изоморфизм корневых деревьев

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

Даны два корневых дерева, состоящие из  $n$  вершин. Корень первого дерева имеет номер  $r_1$ , а корень второго дерева — номер  $r_2$ . Требуется проверить, изоморфны ли эти корневые деревья друг другу.

### Формат входных данных

Первая строка содержит целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество вершин в каждом из деревьев.

Вторая строка содержит два целых числа  $r_1$  и  $r_2$  ( $1 \leq r_1, r_2 \leq n$ ) — номера корней первого и второго дерева, соответственно.

Каждая из следующих  $n - 1$  строк содержит два целых числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n$ ) — описание  $i$ -го ребра первого дерева. Гарантируется, что данная последовательность ребер задает дерево.

Каждая из следующих  $n - 1$  строк содержит два целых числа  $c_i$  и  $d_i$  ( $1 \leq c_i, d_i \leq n$ ) — описание  $i$ -го ребра второго дерева. Гарантируется, что данная последовательность ребер задает дерево.

### Формат выходных данных

Выведите слово «YES», если корневые деревья изоморфны друг другу. В противном случае выведите слово «NO».

### Примеры

стандартный ввод	стандартный вывод
6 3 2 1 3 2 3 3 4 4 5 4 6 1 2 1 5 1 6 2 4 2 3	YES
4 1 4 1 2 1 3 1 4 1 2 2 3 3 4	NO

## Задача I. Изоморфизм деревьев

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

Даны два дерева, состоящие из  $n$  вершин. Требуется проверить, изоморфны ли эти деревья друг другу.

### Формат входных данных

Первая строка содержит целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество вершин в каждом из деревьев.

Каждая из следующих  $n - 1$  строк содержит два целых числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n$ ) — описание  $i$ -го ребра первого дерева. Гарантируется, что данная последовательность ребер задает дерево.

Каждая из следующих  $n - 1$  строк содержит два целых числа  $c_i$  и  $d_i$  ( $1 \leq c_i, d_i \leq n$ ) — описание  $i$ -го ребра второго дерева. Гарантируется, что данная последовательность ребер задает дерево.

### Формат выходных данных

Выведите слово «YES», если деревья изоморфны друг другу. В противном случае выведите слово «NO».

### Примеры

стандартный ввод	стандартный вывод
6 1 3 2 3 3 4 4 5 4 6 1 2 1 5 1 6 2 4 2 3	YES
4 1 2 1 3 1 4 1 2 2 3 3 4	NO

## Задача J. Одинаковые квадраты

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

На разборе задач одного из контестов петрозаводских сборов Вова и Саша поспорили, кто из них сможет найти за 300 минут в матрице размера  $N \times M$ , состоящей из строчных латинских букв, пару одинаковых квадратов наибольшего размера. Квадраты могут накладываться друг на друга, но не могут совпадать. Кто нашёл пару большего размера, тот и выиграл. Мимо проходил Петя, посмотрел на матрицу, сказал, что оптимальная пара квадратов имеет сторону  $K$ , и пошёл дальше. Вова и Саша до сих пор пытаются найти этот ответ. Может быть, вы скажете, какую пару квадратов имел в виду Петя?

### Формат входных данных

В первой строке через пробел даны два целых числа  $N$  и  $M$ .  $1 \leq N, M \leq 500$ . В следующих  $N$  строках по  $M$  символов в каждой строке приведена матрица из строчных латинских букв.

### Формат выходных данных

В первой строке выведите целое число  $K$ , которое сказал Петя. В следующих двух строках выведите координаты левой верхней клетки каждого из квадратов. Если существует более одной пары одинаковых квадратов наибольшего размера, то можно вывести любую из них. Вы можете считать, что левая верхняя клетка матрицы имеет координаты  $(1, 1)$ , правая нижняя — координаты  $(N, M)$ . Если Петя сказал, что в матрице не существует пары одинаковых квадратов, выведите единственное число 0.

### Примеры

стандартный ввод	стандартный вывод
5 10 ljkfghdfas isdfjksiye pgljkijlgp eyisdafdsi lnpglkfkjl	3 1 1 3 3

## Задача К. Забывчивый Антипа

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Антипа уже давно путешествует и странствует по разным городам и странам, так что нет ничего удивительного в том, что он начинает постепенно забывать, какие места он уже успел посетить, а какие нет.

К счастью, у него есть удобный способ помнить все свои маршруты — каждое путешествие длиной в  $n$  дней он запоминает в виде строки из маленьких латинских букв длины  $n$ , каждая из которых означает направление его движения в соответствующий день.

Проблема заключается в том, что все его маршруты циклически и возвращают его в исходную точку, а также в том, что у Антипы нет компаса, а потому направления, задаваемые буквами, могут отсчитываться от разных ориентиров. Таким образом, один и тот же маршрут может описываться разными строками в зависимости от стартовой точки и «нулевого» направления.

Теперь Антипе интересно, могут ли строки  $t$  и  $s$  описывать один и тот же маршрут. Формально, это возможно, если  $t$  можно получить из  $s$  следующим преобразованием:

1. сначала можно изменить стартовую точку маршрута, то есть сделать циклический сдвиг строки  $s$  на любую величину  $k$  влево (например, из «abcd» можно получить «bcda», «cdab» и «dabc»);
2. затем можно повернуть все направления на одну и ту же величину, то есть применить шифр Цезаря: выбрать целое число  $d$  и заменить каждую букву на предшествующую ей на  $d$  позиций в алфавите (например, для строки «abxu» и  $d = 2$  получится «yzvw»).

Помогите Антипе выяснить, могли ли два маршрута с описаниями  $t$  и  $s$  совпадать или нет.

### Формат входных данных

В первой строке дано одно целое число  $n$  — длина маршрутов ( $1 \leq n \leq 200\,000$ ).

Во второй строке дана строка  $t$ , состоящая из  $n$  строчных латинских символов — описание первого маршрута.

В третьей строке содержится строка  $s$ , состоящая из  $n$  строчных латинских символов — описание второго маршрута.

### Формат выходных данных

Если не существует описанного преобразования, которое переведёт строку  $s$  в строку  $t$ , выведите «Impossible» (без кавычек).

Иначе, выведите «Success», а во второй строке выведите два числа  $k$  и  $d$ , описывающие преобразование, переводящее строку  $s$  в строку  $t$  ( $0 \leq k < n$ ,  $-26 < d < 26$ ). Если различных подходящих ответов несколько, вы можете вывести любой из них.

## Примеры

стандартный ввод	стандартный вывод
3 abc fde	Success 1 3
3 abc aba	Impossible
1 z a	Success 0 -25
5 abcde cdeab	Success 3 0

## Задача L. Симметричное дерево

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

Дано дерево, состоящее из  $n$  вершин. От вас требуется сопоставить  $i$ -й вершине дерева точку на плоскости  $p_i = (x_i, y_i)$ . Каждому ребру дерева  $(u, v)$  необходимо сопоставить отрезок прямой, соединяющий точки  $p_u$  и  $p_v$ . Также должны быть выполнены следующие условия:

1. Все точки на плоскости, в которых расположены вершины дерева, попарно различны.
2. Никакие два отрезка не пересекаются в точках, отличных от концов отрезков.
3. Существует прямая, такая что множество точек, соответствующих вершинам дерева, а также множество отрезков, соответствующих ребрам дерева, симметричны относительно данной прямой.

### Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит одно целое число  $T$  ( $1 \leq T \leq 10^3$ ) — количество наборов входных данных. Далее следует их описание.

Первая строка описания набора входных данных содержит одно целое число  $n$  ( $1 \leq n \leq 10^3$ ) — количество вершин в дереве.

Каждая из следующих  $n - 1$  строк содержит два целых числа  $u$  и  $v$  ( $1 \leq u, v \leq n, u \neq v$ ) — описание ребра, соединяющего вершины  $u$  и  $v$ .

Обратите внимание, что в данной задаче **нет** ограничения на сумму  $n$  по всем наборам входных данных.

### Формат выходных данных

Выведите ответ для каждого набора входных данных в следующем формате.

Если построить требуемое множество точек невозможно, выведите в единственной строке слово «NO».

В противном случае в первой строке выведите слово «YES».

В каждой из следующих  $n$  строк выведите два целых числа  $x_i$  и  $y_i$  ( $0 \leq |x_i|, |y_i| \leq n$ ) — координаты точки на плоскости, соответствующей  $i$ -й вершине дерева.

В последней строке выведите три целых числа  $a, b$  и  $c$  ( $0 \leq |a|, |b|, |c| \leq n$ ) — коэффициенты, задающие прямую, относительно которой множества точек и отрезков симметричны (уравнение имеет вид  $ax + by + c = 0$ ).

Если существует несколько корректных ответов, выведите любой.

**Пример**

стандартный ввод	стандартный вывод
5	YES
4	1 0
3 2	-2 0
1 3	-1 0
4 1	2 0
4	1 0 0
2 4	YES
1 4	1 0
3 4	0 1
9	-1 0
9 7	0 0
4 9	1 0 0
8 4	YES
4 6	0 3
1 8	-2 0
2 6	0 0
5 1	0 1
3 4	0 4
10	-1 0
5 3	2 0
4 5	0 2
6 4	1 0
2 5	1 0 0
5 8	NO
4 9	NO
7 8	
1 2	
10 6	
7	
2 7	
7 4	
7 5	
6 2	
4 3	
2 1	