

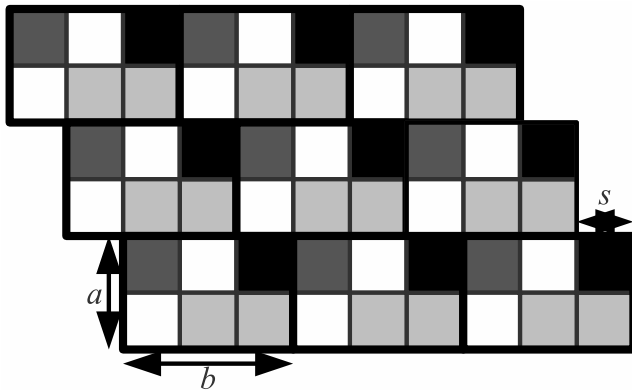
Задача А. Снова в космос

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

К 50-летию первого пилотируемого полета в космос решено создать новый тип космического корабля многоразового использования “Восторг”. Прямоугольная часть его корпуса (далее прямоугольник) должна быть облицована квадратными термозащитными плитками разных цветов одного и того же размера. Прямоугольник состоит из r рядов по c плиток в каждом. Плитки должны образовывать заданный рисунок.

Облицовка космического корабля отдельными плитками очень трудоемка, поэтому для выкладывания заданного рисунка используются одинаковые прямоугольные панели, состоящие из плиток. Панели крепятся на корпусе одна за другой, заполняя ряд за рядом сверху вниз. Каждый ряд панелей может быть сдвинут относительно предыдущего на одно и то же число плиток. При этом панели могут выходить за пределы прямоугольника. Панели должны быть одинаково ориентированы, то есть при параллельном переносе одной панели на место другой цвета образующих эти панели плиток должны совпадать.

Главный конструктор хочет выбрать такой размер панели $a \times b$ и сдвиг s , чтобы этими панелями можно было выложить заданный рисунок, и площадь панели была минимальна.



Требуется написать программу, которая по заданному расположению плиток в прямоугольнике рассчитывает размеры минимальной по площади панели, которую можно использовать при его облицовке, а также величину сдвига вправо ($0 \leq s < b$) каждого следующего ряда относительно предыдущего.

Формат входных данных

Первая строка входного файла содержит два целых числа: r и c ($r, c \leq 1961$) — размеры прямоугольника в плитках. В последующих r строках указаны цвета плиток фрагмента. Каждый из $k \leq 26$ цветов обозначен одной из первых k прописных букв латинского алфавита. Гарантируется, что для этого прямоугольника можно подобрать панель размера $a \times b$, такую, что $2a \leq r$ и $2b \leq c$.

Формат выходных данных

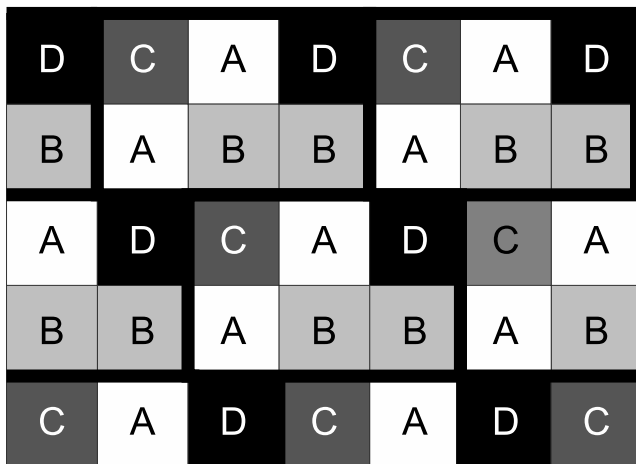
В выходной файл необходимо вывести три целых числа a , b и s , удовлетворяющих условиям задачи. Если решений несколько, разрешается вывести любое из них.

Примеры

shuttle.in	shuttle.out
2 4 ABAB ABAB	1 2 0
5 7 DCADCAD BABBAB ADCADCA BBABBA CADCAD	2 3 1

Замечание

Во втором примере облицовка прямоугольника соответствуют следующему рисунку (выступающие за границы прямоугольника части панелей не показаны):



Задача В. Множественный поиск

Имя входного файла: search4.in
Имя выходного файла: search4.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

Дано множество строк S и строка t . Требуется для каждой строки $p \in S$ определить, встречается ли она в t как подстрока.

Формат входных данных

Первая строка входного файла содержит целое число n — мощность S ($1 \leq n \leq 10^6$). Следующие n строк содержат по одной строке из S . Сумма длин всех строк из S не превосходит 10^6 . Последняя строка входного файла содержит t ($1 \leq t \leq 10^6$). Все строки состоят из прописных латинских букв.

Формат выходных данных

Для каждой строки из S выведите «YES», если она встречается в t и «NO» в противном случае. Строки нумеруются в порядке появления во входном файле.

Примеры

search4.in	search4.out
3	YES
abc	NO
abcdr	YES
abcde	
xabcdef	

Задача С. 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>
<code>aaaa</code>	<code>4</code>
<code>Caabdaaaa</code>	<code>1 2 6 7</code>

Задача D. Диаграммы Ахо

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Это ИНТЕРАКТИВНАЯ задача!!!11!!

Саша и Рома играют в игру с диаграммами Ахо. *Диаграмма Ахо* — это строка, состоящая из строчных латинских букв.

Её правила таковы:

- Первый игрок пишет диаграмму Ахо S фиксированной длины N . Также у первого игрока есть диаграмма T , изначально пустая.
- Второй игрок не видит эти диаграммы в течение всей игры. Однако ему разрешено спрашивать про любые две позиции (в обеих диаграммах), правда ли, что символы в них равны. Например, вопрос может выглядеть так “Равны ли второй символ диаграммы S и пятый символ диаграммы T ?” Обратите внимание, что можно спрашивать про два символа одной диаграммы.
- Игра состоит из M раундов. В начале кадого раунда первый игрок добавляет один символ в конец диаграммы T .
- После добавления символа второй игрок может задать не более пяти вопросов. После этого он должен сказать, какое число поддиаграмм диаграммы T равно диаграмме S .

Саша быстро заметила, что Рома всегда преуспевает в роли второго игрока. Она подозревает наличие стратегии, позволяющей второму игроку выигрывать независимо от S и T .

Хотите это проверить?

Формат входных данных

При запуске ваша программа должна считывать два целых числа N и M ($1 \leq N, M \leq 20\,000$) из стандартного потока ввода.

Далее следуют M раундов игры. В i -м раунде вы можете задать **не более пяти** вопросов в формате “<позиция1> <позиция2>”. Описание любой позиции выглядит как “s x ” где $1 \leq x \leq N$ (если это x -ый символ диаграммы S) или как “t y ” где $1 \leq y \leq i$ (если это y -ый символ диаграммы T). Ответ программы жюри будет “Yes”, если символы на этих позициях равны и “No” иначе.

Формат выходных данных

В конце любого раунда вы должны вывести ответ в формате “\$ k ”, где k равно числу вхождений диаграммы S в диаграмму T . После этого в диаграмму T будет автоматически добавлен новый символ (если это не последний раунд).

Не забудьте делать flush.

После окончания игры ваша программа должна автоматически завершиться, иначе ваш вердикт будет каким угодно, кроме ОКа.

Примеры

стандартный ввод	стандартный вывод
aba	
abacaba	

Задача Е. Подпалиндромы

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

Дано слово и запросы двух типов:

- заменить i -ю букву в слове на букву c ;
- проверить, является ли подстрока $s_j \dots s_k$ палиндромом.

Формат входных данных

В первой строке записано слово из n строчных латинских букв. Во второй строке записано целое число m — количество запросов ($5 \leq n, m \leq 10^5$). Следующие m строк содержат запросы. Каждый запрос имеет вид «change i a » или «palindrome? j k », где i, j, k — целые числа ($1 \leq i \leq n; 1 \leq j \leq k \leq n$), а символ c — строчная латинская буква.

Формат выходных данных

На все запросы второго типа выведите «Yes», если подслово $s_j \dots s_k$ является палиндромом, и «No» в противном случае.

Примеры

<code>substring-palindromes.in</code>	<code>substring-palindromes.out</code>
<code>abcda</code>	<code>No</code>
<code>5</code>	<code>Yes</code>
<code>palindrome? 1 5</code>	<code>Yes</code>
<code>palindrome? 1 1</code>	<code>Yes</code>
<code>change 4 b</code>	
<code>palindrome? 1 5</code>	
<code>palindrome? 2 4</code>	

Задача F. Цензура

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

Посчитайте, сколько строк над алфавитом из n символов длины m не содержат ни одной подстроки из заданного множества “запрещенных” строк.

Формат входных данных

В первой строке написаны целые числа n ($1 \leq n \leq 100$) — количество символов в алфавите, m ($1 \leq m \leq 100$) — длина искомых строк и p ($0 \leq p \leq 10$) — количество “запрещенных” подстрок. Следующая строка содержит n символов с кодами больше 32 — буквы алфавита. Далее идет p “запрещенных” строк, длины которых не превосходят $\min(m, 10)$ символов. Строки целиком состоят из символов алфавита.

Формат выходных данных

В первой строке выведите ответ на задачу.

Примеры

<code>censored.in</code>	<code>censored.out</code>
2 3 1 ab bb	5