Содержание

В некоторых задачах большой ввод и вывод. Имеет смысл пользоваться супер быстрым вводом-выводом: `http://acm.math.spbu.ru/~sk1/algo/input-output/`

Задача A. Рюкзак [2 sec, 256 mb]

У нас есть несколько предметов и рюкзак, который выдерживает вес C. Предмет с номером i имеет вес x_i. Определите число различных наборов предметов, которые можно унести в рюкзаке. Два набора считаются различными, если существует хотя бы один предмет, который включён в один из наборов и не включён в другой.

Формат входных данных

В первой строке ввода записано целое число n — количество предметов ($1 \leqslant n \leqslant 30$). Во второй строке записано n целых чисел x_i ($1 \leqslant x_i \leqslant 10^9$). В третье строке записано целое число C — вместимость рюкзака ($0 \leqslant C \leqslant 10^9$).

Формат выходных данных

Выведите единственное целое число — искомое число способов.

Примеры

knapsack.in	knapsack.out
1 1 1	2
1 1 2	2
2 2 2 1	1
2 1 1 2	4
2 1 1 1	3
30 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 30	1073741824

Задача B. Удаление скобок — 2 [1 sec, 256 mb]

Дана строка, составленная из круглых, квадратных и фигурных скобок. Определите, какое наименьшее количество символов необходимо удалить из этой строки, чтобы оставшиеся символы образовывали правильную скобочную последовательность.

Формат входных данных

Во входном файле записана строка из круглых, квадратных и фигурных скобок. Длина строки не превосходит 100 символов.

Формат выходных данных

Выведите строку максимальной длины, являющуюся правильной скобочной последовательностью, которую можно получить из исходной строки удалением некоторых символов. Если возможных ответов несколько, выведите любой из них.

Примеры

erase.in	erase.out
([)]	[]
{([(]{)})]	([]{})
]{)[	

Задача C. Симпатичные узоры 3 [1 sec, 16 mb]

Компания *BrokenTiles* планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер 1×1 метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника $n \times m$ метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть симпатичным.

Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата 2×2 метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

Формат входных данных

На первой строке входного файла находятся три натуральных числа n, m, mod. ($1 \leqslant n \leqslant 10^{100}, 1 \leqslant m \leqslant 6, 1 \leqslant mod \leqslant 10\,000$.

n, m — размеры двора. mod — модуль, по которому нужно посчитать ответ.

Формат выходных данных

Выведите в выходной файл единственное число — количество различных симпатичных узоров, которые можно выложить во дворе размера $n \times m$ по модулю mod. Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.

Пример

nice3.in	nice3.out
2 2 5	4
3 3 23	0

Задача D. Мега-инверсии [2 sec, 256 mb]

Инверсией в перестановке $p_1, p_2, ..., p_N$ называется пара (i, j) такая, что $i < j$ и $p_i > p_j$. Назовем мега-инверсией в перестановке $p_1, p_2, ..., p_N$ тройку (i, j, k) такую, что $i < j < k$ и $p_i > p_j > p_k$. Придумайте алгоритм для быстрого подсчета количества мега-инверсий в перестановке.

Формат входных данных

Первая строка входного файла содержит целое число N ($1 \leqslant N \leqslant 100\,000$). Следующие N чисел описывают перестановку: $p_1, p_2, ..., p_N$ ($1 \leqslant p_i \leqslant N$), все p_i попарно различны. Числа разделяются пробелами и/или переводами строк.

Формат выходных данных

Единственная строка выходного файла должна содержать одно число, равное количеству мега-инверсий в перестановке $p_1, p_2, ..., p_N$.

Примеры

mega.in	mega.out
4 4 3 2 1	4

Задача E. Знакочередование [2 sec, 256 mb]

Реализуйте структуру данных из n элементов $a_1, a_2 \dots a_n$, поддерживающую следующие операции:

- присвоить элементу a_i значение j;
- найти знакочередующуюся сумму на отрезке от l до r включительно $(a_l - a_{l+1} + a_{l+2} - \dots \pm a_r)$.

Формат входных данных

В первой строке входного файла содержится натуральное число n ($1 \leqslant n \leqslant 10^5$) — длина массива. Во второй строке записаны начальные значения элементов (неотрицательные целые числа, не превосходящие 10^4).

В третьей строке находится натуральное число m ($1 \leqslant m \leqslant 10^5$) — количество операций. В последующих m строках записаны операции:

- операция первого типа задается тремя числами `0 i j` ($1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant 10^4$).
- операция второго типа задается тремя числами `1 l r` ($1 \leqslant l \leqslant r \leqslant n$).

Формат выходных данных

Для каждой операции второго типа выведите на отдельной строке соответствующую знакочередующуюся сумму.

Пример

signchange.in	signchange.out
3 1 2 3 5 1 1 2 1 1 3 1 2 3 0 2 1 1 1 3	-1 2 -1 3

Задача F. Цветные волшебники [2 sec, 256 mb]

Сказочная страна представляет собой множество городов, соединенных дорогами с двухсторонним движением. Причем из любого города страны можно добраться в любой другой город либо непосредственно, либо через другие города. Известно, что в сказочной стране не существует дорог, соединяющих город сам с собой и между любыми двумя разными городами, существует не более одной дороги.

В сказочной стране живут желтый и синий волшебники. Желтый волшебник, пройдя по дороге, перекрашивает ее в желтый цвет, синий — в синий. Как известно, при наложении желтой краски на синюю, либо синей краски на желтую, краски смешиваются и превращаются в краску зеленого цвета, который является самым нелюбимым цветом обоих волшебников.

В этом году в столице страны (городе f) проводится конференция волшебников. Поэтому желтый и синий волшебники хотят узнать, какое минимальное количество дорог им придется перекрасить в зеленый цвет, чтобы добраться в столицу. Изначально все дороги не покрашены.

Начальное положение желтого и синего волшебников заранее не известно. Поэтому необходимо решить данную задачу для k возможных случаев их начальных расположений.

Формат входных данных

Первая строка входного файла содержит целые числа: n ($1 \leqslant n \leqslant 100\,000$) и m ($1 \leqslant m \leqslant 500\,000$) — количество городов и дорог в волшебной стране соответственно. Третья строка содержит одно целое число f ($1 \leqslant f \leqslant n$) — номер города, являющегося столицей сказочной страны. В следующих m строках, находится описание дорог страны. В этих m строк записано по два целых числа a_i и b_i, означающих, что существует дорога, соединяющая города a_i и b_i. Следующая строка содержит целое число k ($1 \leqslant k \leqslant 100\,000$) — количество возможных начальных расположений волшебников. Далее следуют k строк, каждая из которых содержит два целых числа — номера городов, в которых изначально находится желтый и синий волшебники соответственно.

Формат выходных данных

Для каждого из k случаев, ваша программа должна вывести в выходной минимальное количество дорог, которое придется покрасить в зеленый цвет волшебникам для того, чтобы добраться в столицу.

Пример

magic.in	magic.out
6 6 1 1 2 2 3 3 4 4 2 4 5 3 6 2 5 6 6 6	1 2

Задача G. Игровой автомат [2 sec, 256 mb]

Создатели игровых автоматов выпустили на рынок математическую игру. Игра состоит в следующем: в каждом автомате создателями задана уникальная комбинация из $M+1$ числа в P-ичной системе счисления. Каждое число состоит из N разрядов. Первые M чисел во время игры не меняют своего значения, обозначим их a_1, a_2, ..., a_M. Во время игры автомат несколько раз случайным образом выбирает из первых M чисел одно число и поразрядно прибавляет его к $M+1$-му числу (будем называть $M+1$-е число счетчиком) по модулю P, тем самым, изменяя его (т. е. счетчик поразрядно накапливает сумму всех сложений по модулю P). Игрок получает выигрыш, если в результате игры счетчик обнулится.

Поразрядное сложение по модулю P выполняется следующим образом: если в каком-либо разряде числа получено значение, большее $P-1$, то оно уменьшается на P, например, при $P=5$ и $N=3$ результат сложения чисел 123 и 144 равен 212.

Вам прислали на инспекцию несколько таких автоматов, удостоверьтесь в том, что выигрыш принципиально возможен.

Формат входных данных

Первая строка файла содержит натуральное число — количество автоматов, присланных на инспекцию. В следующих строках описываются сами игровые автоматы. Каждый автомат описывается отдельно в следующем формате. Первая строка содержит числа P, N, M ($1 \leqslant N, M \leqslant 100$, $2 \leqslant P \leqslant 255$). Следующие $M+1$ строк содержат по N чисел в P–ичной системе счисления, каждое из которых — значение одного разряда P–ичного числа из уникальной комбинации чисел описываемого автомата. Значения разрядов P–ичного числа задаются как числа в десятичной записи через пробел. Суммарный размер входного файла не превосходит 40 000 байт.

Формат выходных данных

Ответ по каждому автомату должен содержаться в отдельной строке. Ответ — это число 0, если игроку вообще не удастся выиграть. Если же выигрыш возможен, то ответ — это число 1 и далее — M чисел через пробел в этой же строке файла: k_1, k_2, ..., k_M, где значение k_i ($k_i \leqslant P$) указывает, сколько раз нужно прибавить к счетчику число a_i, чтобы в результате всех сложений счетчик обнулился. Если решение не единственно, то выведите любое из них.

Пример

arcade.in	arcade.out
3 4 2 2 2 2 2 2 3 3 3 2 4 1 0 2 0 0 0 0 1 2 1 14 2 2 12 12 10 10 3 3	0 1 1 0 0 2 0

Задача H. Простые числа [1 sec, 256 mb]

Нужно уметь отвечать на запрос вида "k-е по величине простое число".

Формат входных данных

На первой строке целое число N не более 10^4 — количество запросов. Далее собственно запросы. Каждый запрос — целое число от 1 до 10^5.

Формат выходных данных

Для каждого запроса выведите соответствующее простое число.

Пример

primes.in	primes.out
4 1 2 3 100000	2 3 5 1299709

Задача I. Discrete Logging [3 sec, 256 mb]

Given a prime P, $2 \leqslant P < 2^{31}$, an integer B, $2 \leqslant B < P$, and an integer N, $2 \leqslant N < P$, compute the discrete logarithm of N, base B, modulo P. That is, find an integer L such that $B^L \equiv N (\mod P)$.

Формат входных данных

Read several lines of input, each containing P,B,N separated by a space.

Формат выходных данных

For each line print the logarithm on a separate line. If there are several, print the smallest; if there is none, print "`no solution`".

Пример

logging.in	logging.out
5 2 1	0
5 2 2	1
5 2 3	3
5 2 4	2
5 3 1	0
5 3 2	3
5 3 3	1
5 3 4	2
5 4 1	0
5 4 2	no solution
5 4 3	no solution
5 4 4	1
12345701 2 1111111	9584351
1111111121 65537 1111111111	462803587

Задача J. incrementator [2 sec, 256 mb]

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

Формат входных данных

Входной файл состоит из одной или нескольких строк, описывающих операции. 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 мегабайта.

Формат выходных данных

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

Примеры

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

Задача K. Дуэль [3 sec, 256 mb]

Двое дуэлянтов решили выбрать в качестве места проведения поединка тёмную аллею. Вдоль этой аллеи растёт n деревьев и кустов. Расстояние между соседними объектами равно одному метру. Дуэль решили проводить по следующим правилам. Некоторое дерево выбирается в качестве стартовой точки. Затем два дерева, находящихся на одинаковом расстоянии от исходного, отмечаются как места для стрельбы. Дуэлянты начинают движение от стартовой точки в противоположных направлениях. Когда соперники достигают отмеченных деревьев, они разворачиваются и начинают стрелять друг в друга.

Дана схема расположения деревьев вдоль аллеи. Требуется определить количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

Формат входных данных

Во входном файле содержится одна строка, состоящая из символов '0' и '1' — схема аллеи. Деревья обозначаются символом '1', кусты — символом '0'. Длина строки не превосходит 100000 символов.

Формат выходных данных

Выведите количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

Примеры

duel.in	duel.out
101010101	4
101001	0

В первом примере возможны следующие конфигурации дуэли (стартовое дерево и деревья для стрельбы выделены жирным шрифтом): **1**0**1**0**1**0101, 10**1**0**1**0**1**01, 1010**1**0**1**0**1** и **1**010**1**010**1**.

Задача L. Ёжики [2 sec, 256 mb]

Дан неориентированный граф без петель и кратных ребер, требуется найти в нем:

1. Размер минимального контролирующего множества вершин.
2. Размер максимального независимого множества вершин.
3. Размер минимального глобального разреза.

Def: *контролирующее множество* — множество вершин, такое, что у каждого ребра хотя бы один из двух концов лежит в множестве.

Def: *независимое множество* — множество вершин, такое, что никакие две вершины множества не соединены ребром.

Def: *глобальный разрез* — разбиение множества всех вершин V на непустые S и T ($S \cup T = V, S \cap T = \varnothing$). Величиной разреза называется количество ребер между S и T.

Формат входных данных

В первой строке файла записаны два числа — n, m (количество вершин и количество ребер).

В следующих m строках записаны пары чисел. Пара чисел xy означает, что есть ребро между вершинами x и y.

Ограничения: $2 \leqslant n \leqslant 15$.

Формат выходных данных

Выведите 3 числа — ответы для трех задач.

Пример

hedgehog.in	hedgehog.out
4 4 1 2 2 3 3 1 3 4	2 2 1

Задача M. Разрез [2 sec, 256 mb]

Дан неориентированный граф. Найдите минимальный разрез между вершинами 1 и n.

Формат входных данных

На первой строке входного файла содержится n ($1 \leqslant n \leqslant 100$) — число вершин в графе и m ($0 \leqslant m \leqslant 400$) — количество ребер. На следующих m строках входного файла содержится описание ребер. Ребро описывается номерами вершин, которые оно соединяет, и его пропускной способностью (положительное целое число, не превосходящее 10 000 000), при этом никакие две вершины не соединяются более чем одним ребром.

Формат выходных данных

На первой строке выходного файла должны содержаться количество ребер в минимальном разрезе и их суммарная пропускная способность. На следующей строке выведите возрастающую последовательность номеров ребер (ребра нумеруются в том порядке, в каком они были заданы во входном файле).

Пример

cut.in	cut.out
3 3 1 2 3 1 3 5 3 2 7	2 8 1 2

Задача N. Просто поток [5 sec, 256 mb]

Дана система из узлов и труб, по которым может течь вода. Для каждой трубы известна наибольшая скорость, с которой вода может протекать через нее. Известно, что вода течет по трубам таким образом, что за единицу времени в каждый узел (за исключением двух — источника и стока) втекает ровно столько воды, сколько из него вытекает.

Ваша задача — найти наибольшее количество воды, которое за единицу времени может протекать между источником и стоком, а также скорость течения воды по каждой из труб.

Трубы являются двусторонними, то есть вода в них может течь в любом направлении. Между любой парой узлов может быть более одной трубы.

Формат входных данных

В первой строке записано натуральное число N — количество узлов в системе ($2 \leqslant N \leqslant 100$). Известно, что источник имеет номер 1, а сток номер N. Во второй строке записано натуральное M ($1 \leqslant M \leqslant 5000$) — количество труб в системе. Далее в M строках идет описание труб. Каждая труба задается тройкой целых чисел A_i, B_i, C_i, где A_i, B_i — номера узлов, которые соединяет данная труба ($A_i \neq B_i$), а C_i ($0 \leqslant C_i \leqslant 10^4$) — наибольшая допустимая скорость течения воды через данную трубу.

Формат выходных данных

В первой строке выведите наибольшее количество воды, которое протекает между источником и стоком за единицу времени. Далее выведите M строк, в каждой из которых выведите скорость течения воды по соответствующей трубе. Если направление не совпадает с порядком узлов, заданным во входных данных, то выводите скорость со знаком минус. Числа выводите с точностью 10^{-3}.

Пример

flow.in	flow.out
2 2 1 2 1 2 1 3	4.0000000 1.0000000 -3.0000000

Задача O. Molecule. Химия!!! [1 sec, 256 mb]

Вася и Сережа играют в следующую игру. В некоторых клетках клетчатого листка Сережа рисует один из символов ‘H’, ‘O’, ‘N’ или ‘C’, после чего Вася должен провести между некоторыми находящимися в соседних клетках символами линии так, чтобы получилось корректное изображение химической молекулы. К сожалению, Сережа любит рисовать много символов, и Вася не может сразу определить, возможно ли вообще нарисовать линии нужным способом. Помогите ему написать программу, которая даст ответ на этот вопрос.

В этой задаче проведенные между символами химических элементов линии будем считать корректным изображением молекулы, если они удовлетворяют следующим условиям:

- каждая линия соединяет символы, нарисованные в соседних (по стороне) клетках,
- между каждой парой символов проведено не более одной линии,
- от каждого элемента отходит ровно столько линий, какова валентность этого элемента (1 для H, 2 для O, 3 для N, 4 для C),
- пустые клетки ни с чем не соединены, и
- **хотя бы в одной клетке** нарисован какой-то символ.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m ($1 \leqslant n, m \leqslant 50$) — размеры листочка, на котором рисует Сережа. Далее следуют n строк по m символов в каждой, задающих конфигурацию химических элементов, которую нарисовал Сережа; пустые клетки задаются символом ‘.’.

Формат выходных данных

В выходной файл выведите одно слово: ‘Valid’, если линии провести требуемым образом можно, и ‘Invalid’, если нельзя.

Пример

molecule.in	molecule.out
3 4 HOH. NCOH OO..	Valid
3 4 HOH. NCOH OONH	Invalid

Задача P. Кубики [2 sec, 256 mb]

Привидение Петя любит играть со своими кубиками. Он любит выкладывать их в ряд и разглядывать свое 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 нестоящие, а какие — всего лишь отражение в зеркале. Помогите Пете! Выясните, сколько кубиков может быть у Пети. 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 число N ($1 \leqslant N \leqslant 100\,000$) и количество различных цветов, в которые могут быть раскрашены кубики — M ($1 \leqslant M \leqslant 100\,000$). Следующая строка содержит N целых чисел от 1 до M — цвета кубиков.

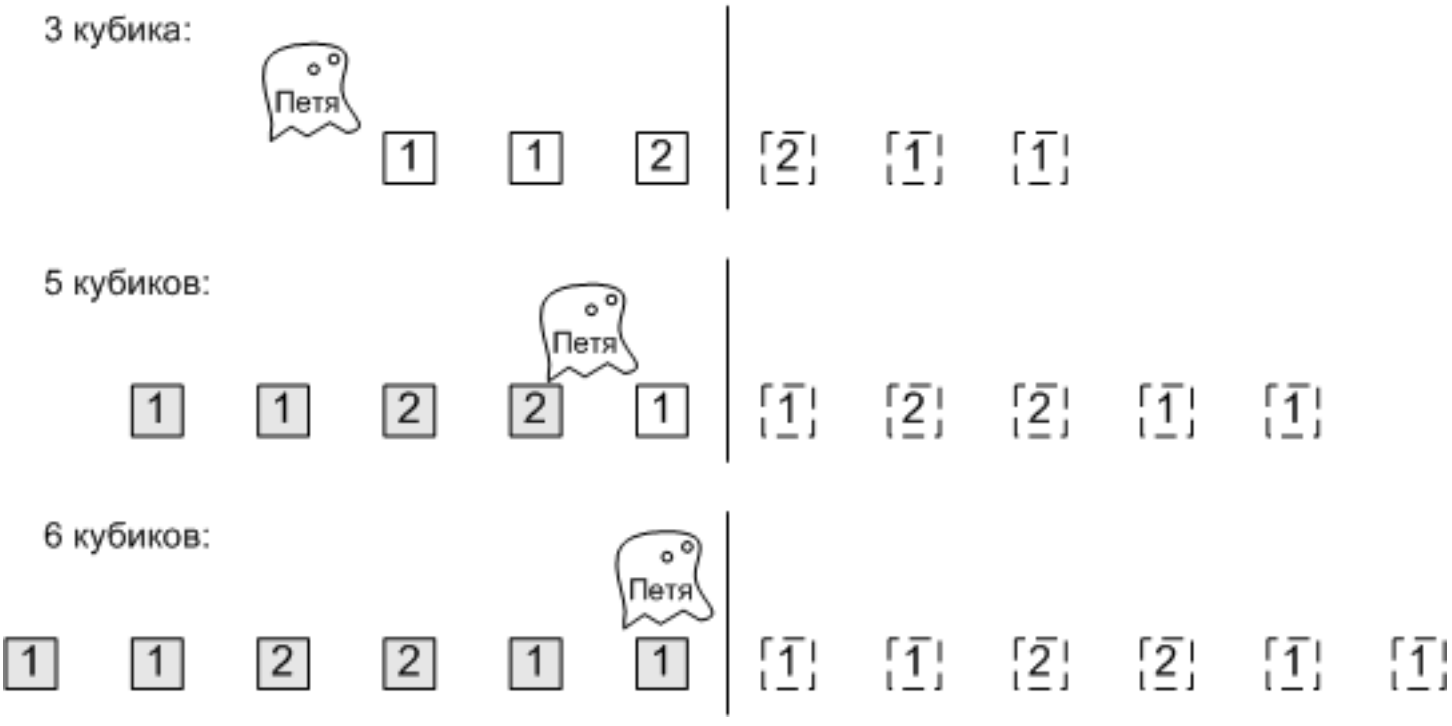
Формат выходных данных

Выведите в выходной файл все такие K, что у Пети может быть K кубиков.

Пример

cubes.in	cubes.out
6 2 1 1 2 2 1 1	3 5 6

В приведенном примере взаимные расположения Пети, кубиков и зеркала приведены на рисунке. Петя смотрит вправо, затененные на рисунке кубики находятся позади Пети и поэтому он их не видит.



Задача Q. Ретростока [1 sec, 256 mb]

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

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

Аналогично для любого k ($1 \le k \le |S|$) *k-м суффиксом* строки S называется строка $S_{|S|-k+1}, ..., S_{|S|}$ 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 *ретростокой*, 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 ретростокой.

Формат входных данных

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

Формат выходных данных

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

Пример

*

retro.in	retro.out
z	z
aabaabaaabaabaaabaaba	aabaabaaabaabaa

Задача R. Подстроки [0.4 sec, 16 mb]

Дана строка s. Вам требуется подсчитать количество её различных подстрок. Пустую строку учитывать не следует.

Формат входных данных

В единственной строке входного файла содержится данная строка s, состоящая из строчных латинских букв. Длина строки не превосходит 20 000 символов.

Формат выходных данных

В единственной строке выходного файла выведите единственное число — количество различных подстрок s.

substr.in	substr.out
aaaa	4
abacaba	21

Задача S. Игра «Жизнь» [7 sec, 256 mb]

Что наша игра? Жизнь!

Дана клетчатая доска $W \times H$. Каждая клетка исходно является либо чёрной, либо белой. Клетки называются соседними, если у них есть общая сторона.

Повторяется следующая процедура:

- Находится белая клетка, у которой не менее двух чёрных соседей. Если такой клетки нет, процесс завершается.
- Найденная клетка перекрашивается в чёрный цвет.

Напишите программу, которая вычислит количество различных раскрасок доски, которые могут получиться по завершении процесса.

Поскольку ответ может быть большим, достаточно найти его остаток по модулю 10^{18}.

Формат входных данных

В первой строке ввода записаны размеры доски: два целых числа W и H ($1 \leqslant W, H \leqslant 13$). В следующих H строках записано по W символов, описывающих саму доску:

- Символ «`B`» обозначает, что клетка в начальной раскраске была чёрной.
- Символ «`.`» обозначает, что исходный цвет клетки неизвестен, то есть он может быть как чёрным, так и белым.

Формат выходных данных

Выведите одно целое число: остаток количества различных возможных конечных раскрасок по модулю 10^{18}.

Пример

life.in	life.out
3 4 B.B .B.	3

Пояснение к примеру

В данном примере возможны следующие конечные раскраски (здесь символом «`W`» обозначаются белые клетки):

```
BBB   BBB   BBB
BBB   BBB   BBB
WWW   BBB   BBB
WWW   WWW   BBB
```