

Содержание

Фурье и многочлены	2
Задача A0. Бесплатный плюсик [0.3 sec, 256 mb]	2
Задача A1. FFT и манная кашка [1 sec, 256 mb]	3
Задача A2. Деление многочленов [1 sec, 256 mb]	4
Автоматы и суффиксы	5
Задача B1. Суффиксный пулемёт [0.5 sec, 256 mb]	5
Задача B2. Суффиксный автомат [1 sec, 256 mb]	7
Задача B3. Помогите, спасите! [1 sec, 256 mb]	8
Задача B4. LZSS encoding [2 sec, 256 mb]	9
Задача B5. Подстроки-4 [1.5 sec, 256 mb]	10
Задача B6. Рефрен [1 sec, 256 mb]	11
Паросочетания	12
Задача C1. Рабочее расписание [0.25 sec, 256 mb]	12
Линейное программирование	13
Задача D1. Простая задача [0.2 sec, 256 mb]	13
Задача D2. Простая задача [0.2 sec, 256 mb]	14
Задача D3. Простая задача с хорошими тестами [0.2 sec, 256 mb]	15
Задача D4. Быстрый симплекс [0.2 sec, 256 mb]	16
Задача D5. Road times [0.4 sec, 256 mb]	17
Факторизация	18
Задача E1. Крайчик и факторизация [3 sec, 256 mb]	18
Задача E2. Крайчик и факторизация (+) [3 sec, 256 mb]	18
Алгоритмы на графах	19
Задача F1. Два китайца и размер дерева [1 sec, 256 mb]	19
Задача F2. Два китайца и дерево [1 sec, 256 mb]	20

Задача F3. Disconnected Graph [0.4 sec, 256 mb]	21
Планарные графы	22
Задача G1. Грани планарного графа [1 sec, 256 mb]	22
Задача G2. Дай шесть. На плоскости. [1 sec, 256 mb]	23
Задача G3. Планарность [1 sec, 256 mb]	24

Фурье и многочлены

Задача A0. Бесплатный плюстик [0.3 сек, 256 mb]

Даны многочлены $P(x)$ и $Q(x)$, найдите такие $A(x)$ и $R(x)$, что $P(x) = A(x)Q(x) + R(x)$ и $\deg R < \deg Q$. $Q(x) = x^k - 1$. У $P(x)$ все коэффициенты — случайные целые числа от 0 до $m - 1$. Все вычисления происходят по модулю m , m — простое от 2 до $10^9 + 7$.

Формат входных данных

Первая строка содержит целые числа n, k, m .

Следующая строка содержит n чисел p_0, p_1, \dots, p_{n-1} , $P(x) = \sum p_i x^i$.

Формат выходных данных

На первой строке выведите $t = \max(n - k, 1)$ целых чисел от 0 до $m - 1$: a_0, a_1, \dots, a_{t-1} . На второй строке k целых чисел от 0 до $m - 1$: r_0, r_1, \dots, r_{k-1} . Все эти числа должны обладать свойством, что $P(x) = Q(x)(\sum a_i x^i) + \sum r_i x^i \pmod m$.

Ограничения: $1 \leq n, k \leq 10^5$.

Примеры

stdin	stdout
3 1 7 1 5 1	6 1 0
3 1 7 2 4 1	5 1 0
3 1 7 0 4 1	5 1 5

Задача A1. FFT и манная кашка [1 сек, 256 mb]

Даны два многочлена

$$A(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$$

$$B(x) = b_m x^m + b_{m-1} x^{m-1} + \dots + b_0$$

Найти $C(x) = A(x)B(x)$

Формат входных данных

$$n, a_n, a_{n-1}, \dots, a_0$$

$$m, b_m, b_{m-1}, \dots, b_0$$

$$0 \leq n, m < 2^{16}, |a_i|, |b_j| \leq 9$$

$$a_n \neq 0, b_m \neq 0$$

Формат выходных данных

Выведите коэффициенты C в том же формате.

Примеры

stdin		stdout	
2	1 0 2	3	2 3 4 6
1	2 3		

Задача A2. Деление многочленов [1 сек, 256 mb]

Даны два многочлена с коэффициентами из $\mathbb{Z}/7\mathbb{Z}$. Старший коэффициент обоих **не равен нулю**. Нужно поделить их с остатком.

Формат входных данных

Каждая из двух строк задаёт описание многочлена. Многочлен $a_k x^k + \dots + a_2 x^2 + a_1 x + a_0$ описывается числом k ($0 \leq k \leq 50\,000$) и $k + 1$ числами от 0 до 6: $a_k, \dots, a_2, a_1, a_0$.

Формат выходных данных

На первой строке многочлен-частное. На второй строке многочлен-остаток. Выводите многочлены в том же формате. Если многочлен – тождественный ноль, для него $k = 0$.

Примеры

stdin	stdout
3 1 1 1 1 1 1 1	2 1 0 1 0 0
3 1 1 3 1 2 1 1 1	1 1 0 1 2 1
8 2 1 2 1 2 1 2 1 2 4 1 2 3 4 5	4 2 4 2 5 2 3 3 1 3 6

Автоматы и суффиксы

Задача В1. Суффиксный пулемёт [0.5 sec, 256 mb]

Или зачёт, или автомат.

Ганнибал Ректор

Теоретическая подготовка новобранцев армии Поссилтума включала в себя не только занятия по военному праву, но и начала криптографии. Лекции читал майор Мега Байт, не чуждый солдатского юмора. Гвидо и Нунцио, в чьё задание входил развал армии Поссилтума изнутри, решили на этом сыграть, внеся путаницу в терминологию. В начале очередной лекции Нунцио поднял руку и спросил:

— Вот вы на прошлой лекции рассказывали про конечные автоматы. А про конечные пулемёты расскажете?

Мега Байт не растерялся.

— Суффиксный пулемёт — это конечный автомат, принимающий все суффиксы данной строки (от нулевого до L -го включительно, где L — длина строки), и только их. Сержант Гвидо!

— Я, господин майор!

— Вы сможете отличить автомат от пулемёта?

— Так точно, господин майор!

— Вам дан конечный автомат. Требуется проверить, является ли он суффиксным пулемётом данной строки.

К сожалению, написание программ такого типа не входило в обязанности Гвидо и Нунцио как в Синдикате, так и в корпорации М. И. Ф. Так что соответствующую программу придётся писать Вам.

Это легенда по мотивам произведений Роберта Асприна,

Задача. Вам дан автомат и строка s . Проверьте, является ли данный автомат суффиксным **пулемётом** строки s , то есть, **детерминированным конечным автоматом**, принимающим ровно суффиксы строки s .

Формат входных данных

Во входном файле задан один или несколько тестовых наборов. В первой строке каждого набора заданы количество состояний автомата N , количество переходов M , а также количество принимающих состояний T ($1 \leq T \leq N \leq 50\,000$, $1 \leq M \leq 100\,000$). Во второй строке через пробел заданы T различных чисел в пределах от 1 до N — принимающие состояния автомата, в возрастающем порядке. В последующих M строках заданы переходы в виде $a_i b_i c_i$, где $1 \leq a_i, b_i \leq n$, а c_i — маленькая буква латинского алфавита. Переход производится из состояния a_i в состояние b_i по букве c_i . Из каждого состояния a_i есть не более одного перехода по символу c_i . Последняя строка описания набора — это строка S , для которой автомат должен являться пулемётом. Она состоит только из маленьких латинских букв, и ее длина лежит в пределах от 1 до 50 000 включительно. Кроме того, сумма всех N и суммарная длина всех строк, для которых необходимо произвести проверку, не превосходит 50 000, а сумма всех M не превосходит 100 000.

Файл заканчивается фиктивным набором, в котором $N = M = T = 0$.

Начальным состоянием автомата является первое. Если при интерпретации какой-то строки в автомате отсутствует соответствующий переход, то автомат вываливается по ошибке и строку не принимает. Таким образом, строка принимается, только если при её интерпретации были найдены все переходы, и по их завершении автомат оказался в принимающем состоянии (при этом неважно, были по пути принимающие состояния, или нет).

Формат выходных данных

Выведите в выходной файл, является ли данный автомат пулемётом, следуя формату примера.

Пример

stdin	
2 1 2	
1 2	
1 2 a	
a	
2 2 2	
1 2	
1 1 a	
1 2 b	
ab	
0 0 0	
stdout	
Automaton 1 is a machinegun.	
Automaton 2 is not a machinegun.	

Задача В2. Суффиксный автомат [1 сек, 256 mb]

Или зачёт, или автомат.

Ганнибал Ректор

Дана строка. Постройте её суффиксный автомат.

Формат входных данных

Строка длины от 1 до 100 000, состоящая из маленьких латинских букв.

Формат выходных данных

На первой строке число состояний автомата и число рёбер. Следующие строки содержат рёбра в формате “откуда” “куда” “символ на ребре”. Далее число терминальных состояний и строка, содержащая все терминальные состояния в произвольном порядке. Начальным состоянием автомата должно быть состояние номер один.

Примеры

stdin	stdout
ababb	7 9 1 2 a 1 7 b 2 3 b 3 4 a 3 6 b 4 5 b 5 6 b 7 4 a 7 6 b 3 6 7 1

Задача В3. Помогите, спасите! [1 sec, 256 mb]

Дана строка. Найдите для каждого её префикса количество различных подстрок в нём.

Формат входных данных

В единственной строке входных данных содержится непустая строка S , состоящая из N ($1 \leq N \leq 2 \cdot 10^5$) маленьких букв английского алфавита.

Формат выходных данных

Выведите N строк, в i -й строке должно содержаться количество различных подстрок в i -м префиксе строки S .

Примеры

stdin	stdout
aabab	1 2 5 8 11
atari	1 3 5 9 14

Задача В4. LZSS encoding [2 sec, 256 mb]

Алиса хочет отправить сообщение Бобу. Она хочет зашифровать сообщение, используя оригинальный метод шифрования. Сообщение – строка S , состоящая из N строчных английских букв.

$S[a \dots b]$ означает подстроку S от $S[a]$ до $S[b]$ ($0 \leq a \leq b < N$). Если первые i букв уже зашифрованы, Алиса найдёт такие (j, k) : $s[j..j+k] = s[i..i+k]$, $k \geq 0$, $0 \leq j < i$, $k = \max$. Если несколько j дают максимальное k , Алиса выберет минимальное j . Если $k > 0$ Алиса добавит пару $\langle j, k \rangle$ в шифр и увеличит i на k , иначе Алиса добавит -1 и ASCII код буквы $S[i]$ в шифр и увеличит i на 1 .

Очевидно шифр начнёт с -1 , далее будет ASCII код символа $S[0]$. Помогите Алисе реализовать её метод шифрования.

Формат входных данных

Первая строка ввода содержит количество тестов T ($1 \leq T \leq 50$). Следующие T строк содержат сообщения для шифровки, каждое длины от 1 до 10^5 , состоящие из строчных английских букв. Гарантируется, что суммарная длина всех сообщений не превосходит $2 \cdot 10^6$.

Формат выходных данных

Для каждого теста на отдельной строке выведите “Case #X:”, где X – номер теста, нумерация с 1 . Далее выведите шифр, в каждой строке по два целых числа через пробел.

Примеры

stdin	stdout
2	Case #1:
aaaaaa	-1 97
aaaaabbbbbbaaabbc	5 0
	Case #2:
	-1 97
	4 0
	-1 98
	4 5
	5 2
	-1 99

Задача В5. Подстроки-4 [1.5 sec, 256 mb]

Даны K строк из маленьких латинских букв. Найдите их наибольшую общую подстроку.

Формат входных данных

В первой строке число K ($1 \leq K \leq 10$). Далее K строк длины от 1 до 200 000.

Формат выходных данных

Наибольшая общая подстрока.

Примеры

stdin	stdout
3 abacaba myscabarchive acabistrue	cab

Задача В6. Рефрен [1 sec, 256 mb]

Рассмотрим последовательность n целых чисел от 1 до m . Подпоследовательность подряд идущих чисел называется *рефреном*, если произведение ее длины на количество вхождений в последовательность максимально.

По заданной последовательности требуется найти ее рефрен.

Формат входных данных

Два целых числа: n и m ($1 \leq n \leq 150\,000$, $1 \leq m \leq 10$).

Вторая строка содержит n целых чисел от 1 до m .

Формат выходных данных

Первая строка выходного файла должна содержать произведение длины рефрена на количество ее вхождений. Вторая строка должна содержать длину рефрена. Третья строка должна содержать последовательность которая является рефреном.

Пример

stdin	stdout
9 3	9
1 2 1 2 1 3 1 2 1	3
	1 2 1

Замечание

Эту задачу обязательно сдавать суффмассивом.

Даже если больше вам по душе деревья и автоматы.

Паросочетания

Задача C1. Рабочее расписание [0.25 сек, 256 mb]

Дан неориентированный граф из N вершин и нескольких рёбер.
Найти максимальное паросочетание.

Здесь можно прочесть оригинальную легенду.

Формат входных данных

На первой строке число вершин N ($1 \leq N \leq 222$).

Далее строки, содержащие рёбра.

Формат выходных данных

На первой строке число вершин, покрытых максимальным паросочетанием.

На следующих строках рёбра паросочетания.

Если есть несколько максимальных паросочетаний, выведите любое.

Примеры

stdin	stdout
3	2
1 2	1 2
2 3	
1 3	

Подсказка по решению

Алгоритм Эдмондса сжатия соцветий. Реализация Габова.

Линейное программирование

Задача D1. Простая задача [0.2 сек, 256 mb]

Найдите оптимальное решение задачи линейного программирования.

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n \leq b_2 \\ \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m \\ x_1 \geq 0 \\ \dots \\ x_n \geq 0 \\ c_1x_1 + \dots + c_nx_n \rightarrow \max \end{cases}$$

Формат входных данных

Первая строка входного файла содержит два целых числа: n и m — количество переменных и количество уравнений ($1 \leq n, m \leq 5$). Следующие m строк содержат по $n + 1$ целому числу: $a_{i1}, \dots, a_{in}, b_i$. Следующая строка содержит n целых чисел: c_1, \dots, c_n . Все числа во входном файле не превышают 100 по модулю.

Формат выходных данных

Выведите одно число: максимальное значение $c_1x_1 + \dots + c_nx_n$. Гарантируется, что решение существует и максимальное значение достигается.

Пример

stdin	stdout
2 2 1 2 3 2 1 3 1 1	2.0
1 1 1 2 -2	0.0
1 1 4 5 3	3.75

Задача D2. Простая задача [0.2 сек, 256 mb]

Найдите оптимальное решение задачи линейного программирования.

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n \leq b_2 \\ \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m \\ x_1 \geq 0 \\ \dots \\ x_n \geq 0 \\ c_1x_1 + \dots + c_nx_n \rightarrow \max \end{cases}$$

Формат входных данных

Первая строка входного файла содержит два целых числа: n и m — количество переменных и количество уравнений ($1 \leq n, m \leq 5$). Следующие m строк содержат по $n + 1$ целому числу: $a_{i1}, \dots, a_{in}, b_i$. Следующая строка содержит n целых чисел: c_1, \dots, c_n . Все числа во входном файле не превышают 100 по модулю.

Формат выходных данных

Выведите одно число: максимальное значение $c_1x_1 + \dots + c_nx_n$. Если решения нет, выведите “No solution”. Если можно получить сколь угодно большое значение, выведите “Unbounded”.

Пример

stdin	stdout
2 2 1 2 3 2 1 3 1 1	2.0
2 1 -1 -1 0 1 1	Unbounded
2 1 1 1 -1 1 1	No solution

Задача D3. Простая задача с хорошими тестами [0.2 сек, 256 mb]

Найдите оптимальное решение задачи линейного программирования.

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n \leq b_2 \\ \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n \leq b_m \\ x_1 \geq 0 \\ \dots \\ x_n \geq 0 \\ c_1x_1 + \dots + c_nx_n \rightarrow \max \end{cases}$$

Формат входных данных

Входной файл содержит один или несколько тестов. На первой строке число тестов, далее сами тесты. Каждый тест описывается следующим образом. Первая строка теста содержит два целых числа: n и m — количество переменных и количество неравенств ($1 \leq n, m \leq 5$). Следующие m строк содержат по $n + 1$ целому числу: $a_{i1}, \dots, a_{in}, b_i$. Следующая строка содержит n целых чисел: c_1, \dots, c_n . Все числа во входном файле целые и не превышают 100 по модулю.

Формат выходных данных

Для каждого теста выведите одну строку.

Выведите максимальное значение $c_1x_1 + \dots + c_nx_n$, пробел, двоеточие, пробел, сами числа $x_1x_2 \dots x_n$. Все числа выводите с максимальной точностью. Если решения нет, выведите “No solution”. Если можно получить сколь угодно большое значение, выведите “Unbounded”.

Пример

stdin	stdout
4	2.0 : 1.0 1.0
2 2	Unbounded
1 2 3	No solution
2 1 3	1.5 : 0 1.5
1 1	
2 1	
-1 -1 0	
1 1	
2 1	
1 1 -1	
1 1	
2 2	
1 2 3	
2 1 3	
-10 1	

Задача D4. Быстрый симплекс [0.2 сек, 256 mb]

Дана системы равенств и неравенств:

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{m1}x_1 + \dots + a_{mn}x_n = b_m \end{cases} \quad \begin{cases} l_1 \leq x_1 \leq u_1 \\ \dots \\ l_n \leq x_n \leq u_n \end{cases}$$

Также вам даны вектора $(c_{11}, c_{12}, \dots, c_{1n}), \dots, (c_{k1}, c_{k2}, \dots, c_{kn})$.
 $\forall i = 1..k$ найдите вектор $x: x_1c_{i1} + \dots + x_nc_{in} \rightarrow \max$.

Формат входных данных

Входной файл содержит один или несколько тестов.

На первой строке число тестов, далее сами тесты. Каждый тест описывается следующим образом. Первая строка теста содержит три целых числа: n, m, k — количество переменных, количество уравнений, количество целевых функций ($1 \leq n, m, k \leq 30$).

Следующие m строк содержат по $n + 1$ целому числу: $a_{i1}, \dots, a_{in}, b_i$.

Следующие n строк содержат пары чисел l_i, r_i ($l_i \leq r_i$).

Следующие k строк содержат по n целых чисел: c_{i1}, \dots, c_{in} .

Все числа во входном файле целые и не превышают 100 по модулю.

Формат выходных данных

Для каждого теста выведите k строк, i -я строка – результат максимизации выражения $x_1c_{i1} + \dots + x_nc_{in}$. Выведите максимальное значение $c_1x_1 + \dots + c_nx_n$, пробел, двоеточие, пробел, сами числа $x_1x_2 \dots x_n$. Все числа выводите с максимальной точностью. Гарантируется, что в каждом тесте решение существует. Ответы на тесты разделяйте пустой строкой.

Пример

stdin	stdout

Задача D5. Road times [0.4 sec, 256 mb]

Дорожная сеть страны – ориентированный граф. У каждого ребра есть длина, целое число от 1 до 1000, и ограничение на максимальную скорость, вещественное число от 30 до 60, в километрах в час. Длины вам известны, а ограничения – нет. Известно, что когда водитель такси берётся доставить пассажира из вершины a в вершину b , он осуществляет перевозку строго по кратчайшему пути между вершинами и едет с максимальной допустимой скоростью. Длина пути – сумма длин рёбер. Также известно, что $\forall a, b \exists$ *единственный* кратчайший путь из a в b .

Ваша задача – по длинам рёбер и уже сделанным поездкам $a_i b_i time_i$ оценить минимальное и максимальное время в пути между вершинами $c_j d_j$

Формат входных данных

На первой строке число вершин n ($1 \leq n \leq 30$).

Вершины нумеруются числами от 0 до $n - 1$.

Следующие n строк содержат матрицу $n \times n$ длин дорог. Дороги односторонние.

Отсутствие дорог обозначено числом -1 , длины дорог целые от 1 до 1000.

Всего не более 100 дорог.

Далее идёт число r ($1 \leq r \leq 100$) и r уже известных маршрутов $a_i b_i time_i$, $time_i \in \mathbb{R}$.

Времена в минутах (не в часах).

Затем число запросов q ($1 \leq q \leq 100$) и q строк $c_j d_j$.

Формат выходных данных

Для каждого запроса выведите четыре числа – откуда, куда, min время, max время.

Примеры

stdin	stdout
3	0 1 50.0 80.0
0 50 -1	1 2 40.0 70.0
55 0 40	1 0 55.0 110.0
-1 40 0	
1	
0 2 120	
3	
0 1	
1 2	
1 0	

Факторизация

Задача E1. Крайчик и факторизация [3 sec, 256 mb]

Дано натуральное число n , которое является произведением двух случайных простых. Вам нужно найти эти два простых.

Формат входных данных

Во входном файле записано единственное число n ($6 \leq n \leq 2.9 \cdot 10^{30}$)

Формат выходных данных

Выведите любой один простой делитель n .

Примеры

stdin	stdout
6	3
187	17
25295820419	197203
1966979996793858097	1014008657
1934189026183600580721953	1057785117097

Задача E2. Крайчик и факторизация (+) [3 sec, 256 mb]

Дано натуральное число n , которое является произведением двух случайных простых. Вам нужно найти эти два простых.

Формат входных данных

Во входном файле записано единственное число n ($6 \leq n \leq 2.1 \cdot 10^{36}$)

Формат выходных данных

Выведите любой один простой делитель n .

Примеры

stdin	stdout
6	3
187	17
25295820419	197203
1966979996793858097	1014008657
1934189026183600580721953	1057785117097

Алгоритмы на графах

Задача F1. Два китайца и размер дерева [1 сек, 256 mb]

Дан ориентированный взвешенный граф. Выбрать множество рёбер минимально возможного размера и при равенстве размера минимального суммарного веса, чтобы все вершины были достижимы из **первой**. Гарантируется, что такое множество существует.

Формат входных данных

Входные данные содержат описание одного или более тестов.

Каждый тест описывается следующим образом. На первой строке количества вершин и рёбер в графе n ($1 \leq n \leq 100\,000$) и m ($1 \leq m \leq 300\,000$). На следующих m строках рёбра в формате « $a_i b_i w_i$ » ($1 \leq a_i, b_i \leq n$, $-10^5 < w_i < 10^5$), что обозначает ребро из вершины a_i в вершину b_i веса w_i . В графе могут быть и петли, и кратные рёбра.

Сумма n по всем тестам не более 100 000. Сумма m по всем тестам не более 300 000.

Формат выходных данных

Для каждого теста выведите одно число — суммарный вес выбранных рёбер.

Примеры

stdin	stdout
3 3	-4
1 2 -3	21
2 3 -1	3
3 1 -10	
4 5	
1 2 10	
2 4 2	
1 3 10	
3 4 1	
1 4 10	
5 6	
1 2 1	
2 3 1	
3 4 1	
4 5 1	
1 5 2	
5 3 -1	

Задача F2. Два китайца и дерево [1 сек, 256 mb]

Дан ориентированный взвешенный граф. Выбрать множество рёбер минимально возможного размера и при равенстве размера минимального суммарного веса, чтобы все вершины были достижимы из первой. Гарантируется, что такое множество существует.

Формат входных данных

Входные данные содержат описание одного или более тестов.

На первой строке количества вершин и рёбер в графе n ($1 \leq n \leq 100\,000$) и m ($1 \leq m \leq 300\,000$). На следующих m строках рёбра в формате " $a_i b_i w_i$ " ($1 \leq a_i, b_i \leq n$, $-10^5 < w_i < 10^5$), что обозначает ребро из a_i в b_i веса w_i . В графе могут быть и петли, и кратные рёбра.

Сумма n по всем тестам не более 100 000. Сумма m по всем тестам не более 300 000.

Формат выходных данных

Для каждого теста выведите две строки ответа. На первой строке выведите суммарный вес выбранных рёбер. Во второй строке выведите номера выбранных рёбер, числа от 1 до m в произвольном порядке.

Примеры

stdin	stdout
3 3	-4
1 2 -3	1 2
2 3 -1	21
3 1 -10	1 3 4
4 5	3
1 2 10	1 3 5 6
2 4 2	
1 3 10	
3 4 1	
1 4 10	
5 6	
1 2 1	
2 3 1	
3 4 1	
4 5 1	
1 5 2	
5 3 -1	

Задача F3. Disconnected Graph [0.4 sec, 256 mb]

Дан связный неориентированный граф и несколько *маленьких* множеств рёбер. Для каждого множества определите, останется ли граф связным, если удалить все рёбра множества.

Формат входных данных

В первой строке число вершин n ($1 \leq n \leq 10\,000$) и число рёбер m ($1 \leq m \leq 100\,000$). Вершины нумеруются числами от 1 до n . Следующие m строк содержат описание рёбер графа. В графе нет петель и кратных рёбер. Рёбра нумеруются числами от 1 до m в том порядке, что даны вам.

Далее следует число множества k ($1 \leq k \leq 100\,000$). Следующие k строк содержат описание множеств. Множество задаётся размером c ($1 \leq c \leq 4$) и c различными числами – номерами рёбер в множестве.

Формат выходных данных

Выведите k строк. Для каждого множества «Connected» или «Disconnected» в зависимости от того, что произойдёт, если удалить из графа все рёбра соответствующего множества.

Примеры

stdin	stdout
4 5	Connected
1 2	Disconnected
2 3	Connected
3 4	
4 1	
2 4	
3	
1 5	
2 2 3	
2 1 2	

Подсказка по решению

Вы можете написать `offline-dynamic-connectivity`.

А можете вспомнить про XOR-ы и решить задачу `online`, отвечая на запрос за $\mathcal{O}(1)$.

Планарные графы

Задача G1. Грани планарного графа [1 сек, 256 mb]

Посчитать число граней в планарном графе.

Формат входных данных

Первая строка содержит два числа, N и M ($N \leq 100$) — число точек на плоскости и число отрезков.

Далее N строк содержат пары целых чисел x, y ($|x|, |y| \leq 10^4$) — координаты точек. Далее M строк содержат пары целых чисел от 1 до N — номера точек, которые соединяет очередной отрезок.

В графе нет петель и кратных ребер. Отрезки не пересекаются (т.е. получившийся граф действительно планарен).

Формат выходных данных

Выведите единственное число G — число граней в данном вам планарном графе.

Пример

stdin	stdout
3 3 0 0 1 0 0 1 1 2 2 3 3 1	2

Задача G2. Дай шесть. На плоскости. [1 sec, 256 mb]

Проверьте для данного графа с не более чем 6 вершинами, планарный ли он.

Формат входных данных

В первой строке содержится число тестов t ($1 \leq t \leq 500$).

В каждой из следующей строк описан граф. Если в графе n вершин, описание состоит из $\frac{n(n-1)}{2}$ символов, означающих наличие (1) или отсутствие (0) следующих ребер: между вершинами 1 и 2, 1 и 3, 2 и 3, 1 и 4, 2 и 4, 3 и 4 и так далее (то есть нижний треугольник матрицы смежности, выписанный слитно построчно).

Формат выходных данных

Для каждого графа выведите «YES», если он планарный, и «NO» иначе.

Пример

stdin	stdout
3	NO
1111111111	NO
000111111011100	YES
111111	

Замечание

В примере заданы графы K_5 , K_{33} и K_4 .

Задача G3. Планарность [1 сек, 256 mb]

Задан неориентированный граф и гамильтонов цикл в нём. Требуется расположить граф на плоскости без самопересечений таким образом, чтобы рёбра изображались отрезками и дугами окружностей.

Формат входных данных

В первой строке входного файла содержится число вершин графа n ($1 \leq n \leq 100$) и число рёбер m . В последующих m строках заданы рёбра, каждое номерами двух вершин. Далее дана последовательность вершин $p_1 \dots p_n$ в порядке их обхода по гамильтонову циклу.

Формат выходных данных

Если расположить граф требуемым образом невозможно, выдать в выходной файл слово «NO».

Иначе в первой строке выходного файла выдать слово «YES», во второй строке — координаты вершин графа, а в последующих m — координаты середины каждого из рёбер. Вершины и рёбра должны идти в той же последовательности, что и во входном файле

Пример

stdin	stdout
3 3	YES
1 2	0 0 1 0 2 0
2 3	0.5 0
1 3	1.5 0
1 2 3	1 11

Подсказка по решению

Это простая задача. Воспользуйтесь вашими знаниями из алгоритма Демукрона.