Задача A. Кротовья нора

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

Кротовья нора состоит из пещерок, соединённых коридорами, причём из каждой пещерки выходит ровно три коридора. Коридорами и пещерками пользуется не только крот — в одной из пещерок обосновался жук-носорог, а в другую только что свалился жук-олень. Жук-олень чует своего соперника и спешит встретиться, чтобы выяснить, кто из них будет главным в кротовьей норе!

Выясните, каким минимальным количеством коридоров потребуется пройти жуку-оленю, чтобы попасть в пещерку, где его поджидает жук-носорог.

Формат входного файла

В первой строке входного файла заданы три числа: N ($4 \leqslant N \leqslant 1000$) — количество пещерок в кротовьей норе, X ($1 \leqslant X \leqslant N$) — номер пещерки жука-носорога и Y ($1 \leqslant Y \leqslant N$) — номер пещерки жука-оленя. В последующих N строках стоит по три числа в каждой: в $(i+1)$-ой строке стоят номера пещерок, в которые из i-ой пещерки ведут коридоры. Каждый коридор упоминается в строках обеих вершин, которые он соединяет. Никакой коридор не соединяет пещерку с самой собой, и никакие две пещерки не соединены более чем одним коридором. Гарантируется, что $X \neq Y$. Все числа во входном файле целые.

Формат выходного файла

Выведите в выходной файл одно целое число — минимальное количество коридоров, которыми требуется пройти, чтобы попасть из пещерки жука-оленя в пещерку жука-носорога. Если это оказалось невозможно, выведите число -1.

Примеры

burrow.in	burrow.out
4 1 4 2 3 4 3 4 1 4 1 2 1 2 3	1
6 2 3 4 5 6 4 5 6 4 5 6 1 2 3 1 2 3 1 2 3	2
8 5 4 2 3 4 1 3 4 1 2 4 1 2 3 6 7 8 5 7 8 5 6 8 5 6 7	-1

Задача B. Золотое сечение

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

Вы, наверное, уже знаете, что цепная дробь — это последовательность вида

$$[a_0; a_1, a_2, \ldots, a_n],$$

которой ставится в соответствие число

$$x = a_0 + \frac{1}{a_1 + \frac{1}{\ldots + \frac{1}{a_n}}}.$$

Здесь $n \geqslant 0$, a_i — целые числа, $a_i > 0$ при $i > 0$ и $a_n > 1$ при $n > 0$; такие условия нужны для того, чтобы между числами и цепными дробями существовало взаимно однозначное соответствие.

Заметим, что если цепная дробь числа x конечна, то, последовательно приводя дроби к общему знаменателю снизу вверх, её можно свести к виду $\frac{p}{q}$, где p и q — целые числа. Так, например, для $x = [1; 2, 1, 3]$ получаем

$$x = 1 + \frac{1}{2 + \frac{1}{1+\frac{1}{3}}} = 1 + \frac{1}{2 + \frac{3}{4}} = 1 + \frac{4}{11} = \frac{15}{11}.$$

Цепные дроби полезны прежде всего тем, что, если рассматривать лишь первые несколько членов дроби, то соответствующее число будет достаточно хорошим приближением для исходной цепной дроби. Так, в приведённом примере число $y = [1; 2, 1] = [1; 3]$ (вспомним, что в каноническом виде последнее число цепной дроби при $n > 0$ должно быть больше единицы) отличается от x всего на $|\frac{4}{3} - \frac{15}{11}| = \frac{1}{33}$.

В этой задаче мы будем искать хорошие приближения для числа $\phi = \frac{\sqrt{5}+1}{2} \equiv 1.6180339887\ldots$ — отношения золотого сечения. Это число известно как отношение частей отрезка, при котором весь отрезок относится к большей части так же, как бо́льшая часть к меньшей.

Известно, что цепная дробь ϕ — это бесконечная последовательность $[1; 1, 1, 1, 1, 1, \ldots]$. Поиск приближения будет состоять в том, чтобы взять первые n членов этой бесконечной последовательности и найти число, соответствующее полученной цепной дроби, в виде обыкновенной дроби $\frac{p}{q}$. Кроме того, требуется, чтобы эта дробь была несократима, то есть наибольший общий делитель p и q был равен единице.

Формат входного файла

Входной файл содержит одно целое число n ($1 \leqslant n \leqslant 1000$).

Формат выходного файла

Выведите в выходной файл два целых числа: в первой строке выведите p, а во второй — q. Числа не должны начинаться с нуля.

Примеры

section.in	section.out
2	2 1
3	3 2

В первом примере цепная дробь $[1; 1]$ приводится к каноническому виду $[2]$, и соответствующее приближение оказывается равным двум. Во втором примере цепная дробь $[1; 1, 1]$ преобразуется к каноническому виду $[1, 2]$, и ответ равен $\frac{3}{2}$.

Задача C. Покраска забора

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

Фермер Пётр огородил свое новое пастбище отличным забором, состоящим из n досок. Забор имеет форму замкнутой ломаной.

К сожалению, некоторые из этих досок покрашены в белый цвет, а некоторые — нет. Разумеется, фермер хочет, чтобы его пастбище было огорожено красивым забором, поэтому он решил покрасить все доски в белый цвет.

Для покраски он решил нанять компанию «Вася и Маляры», расположенную в соседнем посёлке. Компания принимает заказы на покраску участков забора. Каждый заказ должен быть выбран из множества предложений компании. Каждое предложение характеризуется количеством раскрашиваемых досок a_i и стоимостью c_i и заключается в том, что за сумму c_i рублей маляры покрасят любые a_i идущих подряд досок в белый цвет.

Требуется определить, за какую минимальную сумму фермер Пётр может добиться того, что весь забор будет покрашен в белый цвет. Доску разрешается красить в белый цвет сколько угодно раз, при этом уже окрашенные доски заново красить не требуется (хотя разрешается), а ещё не окрашенные надо обязательно покрасить.

Формат входного файла

Формат входных данных Первая строка входного файла содержит n — количество досок в заборе ($1 \leqslant n \leqslant 2000$) и m — количество предложений компании ($1 \leqslant m \leqslant 40$). Вторая строка содержит n символов, описывающих забор; символ “+” (плюс) означает уже окрашенную в белый цвет доску, а символ “0” (ноль) — неокрашенную.

Следующие m строк содержат предложения компании. Каждое предложение описывается двумя натуральными числами: a_i и c_i ($1 \leqslant a_i \leqslant n$, $1 \leqslant c_i \leqslant 10^6$).

Формат выходного файла

Выведите в выходной файл одно целое число — минимальную стоимость покраски забора.

Примеры

fence.in	fence.out
15 2 0+000+++0+0+++0 2 3 3 5	13
1 1 0 1 10	10
1 1 + 1 10	0

В первом примере надо воспользоваться первым предложением для раскраски первой и последней доски (забор замкнутый, поэтому они идут подряд), и дважды вторым предложением: чтобы раскрасить доски с 3 по 5 и с 9 по 11 (10-я доска уже покрашена, но её можно покрасить ещё раз).

Задача D. Алгоритм Евклида

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

Дима недавно начал изучать информатику. Один из первых алгоритмов, который он узнал — алгоритм Евклида нахождения наибольшего общего делителя (НОД) двух чисел. Напомним, что наибольшим общим делителем чисел a и b называется наибольшее натуральное число x, такое что и a, и b делятся на него без остатка.

Алгоритм Евклида состоит в следующем:

1. Пусть a, b — числа, НОД которых надо найти.
2. Если $b = 0$, a — искомый НОД.
3. Если $b > a$, поменять a и b местами.
4. Присвоить a значение $a - b$.
5. Вернуться к шагу 2.

Дима уже попрактиковался в применении этого алгоритма и вычислил с его помощью не одну сотню наибольших общих делителей. Однако просто вычислять ему скоро надоело. Теперь он размышляет над новой задачей. Пусть заданы числа a, b, c и d. Требуется узнать, будет ли в процессе применения алгоритма Евклида к паре чисел $(a,\ b)$ такой момент, что перед шагом 2 алгоритма a будет равно c, а b — равно d.

Формат входного файла

Первая строка входного файла содержит количество наборов входных данных k $(1 \leqslant k \leqslant 100)$. Далее идут описания этих наборов. Каждое описание состоит из двух строк. Первая из них содержит два целых числа a и b $(1 \leqslant a, b \leqslant 10^{18})$. Вторая — два целых числа c и d $(1 \leqslant c, d \leqslant 10^{18})$.

Формат выходного файла

Для каждого набора входных данных выведите в выходной файл слово “`YES`”, если в процессе применения алгоритма Евклида к паре чисел $(a,\ b)$ в какой-то момент получается пара $(c,\ d)$. В противном случае выведите слово “`NO`”.

Примеры

euclid.in	euclid.out
2 20 10 10 10 10 7 2 4	YES NO