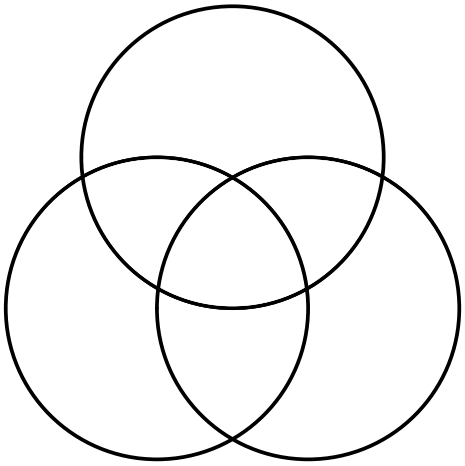
Задача A. Окружности-2

Имя входного файла:	`circles2.in`
Имя выходного файла:	`circles2.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

У Антона есть n колец. Он настолько силен, что может сжимать и растягивать кольца так, чтобы они становились любого радиуса. Каждое кольцо имеет пренебрежимо малую толщину, поэтому его можно рассматривать как окружность.

Теперь Антону интересно, на какое максимальное количество частей он сможет разбить плоскость, если он разложит на ней n колец.

На рисунке изображено разбиение плоскости на восемь частей с помощью трех окружностей.



Формат входного файла

Во входном файле находится одно целое неотрицательное число $n \leq 10^8$.

Формат выходного файла

В выходной файл выведите одно число — максимальное возможное число частей.

Примеры

circles2.in	circles2.out
0	1
1	2
2	4
3	8
4	14

Задача B. Цепная дробь

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

Цепная дробь — это выражение следующего вида.

$$a_0 + \cfrac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \ldots}}}$$

В этом выражении a_0 является целым числом, а остальные a_n — положительными целыми числами. Цепные дроби интересны тем, что с их помощью может быть записано любое вещественное число. При этом для рациональных чисел дробь будет конечной, а для иррациональных — бесконечной.

Например, для числа $\frac{9}{4}$ представление в виде цепной дроби таково: $\frac{9}{4} = 2 + \frac{1}{3+\frac{1}{1}}$.

Ваша задача состоит в том, чтобы найти представление в виде цепной дроби для заданного рационального числа $\frac{p}{q}$.

Формат входного файла

Первая строка входного файла содержит два целых числа: p и q ($1 \le p, q \le 10^3$).

Формат выходного файла

В первой строке выходного файла выведите число n элементов цепной дроби, которая равна $\frac{p}{q}$. Во второй строке выведите числа a_0, a_1, …, a_n. Если вариантов представления числа $\frac{p}{q}$ несколько, то выведите любой из них.

Примеры

frac.in	frac.out
9 4	3 2 3 1

Задача C. Привет

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

Антон учится в классе, в котором n мальчиков и m девочек. В очередной раз придя раньше всех на занятия и сидя в классе, заметил, что каждый мальчик входя в класс жмет руку каждому мальчику уже там находившемуся, а каждой девочке говорит «Привет», при этом девочка тоже говорит ему «Привет». С девочками все гораздо проще. Каждая девочка, заходя в класс говорит каждому в классе «Привет» и каждый ей отвечает «Привет».

Антону стало интересно сколько происходит рукопожатий и сколько раз звучит слово «Привет», если все ученики последовательно заходят в класс.

Формат входного файла

Во входном файле два целых числа n и m ($1 \le n \le 10^8$; $0 \le m \le 10^8$).

Формат выходного файла

В выходной файл выведите через пробел два числа — количество рукопожатий и приветов соответственно.

Примеры

hello.in	hello.out
1 2	0 6
5 0	10 0

Задача D. Метро

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

В городе ЭмСити возникла необходимость в постройке метро. Для это был подготовлен специальный секретный план. Единственное, что известно обычным жителям — будет построено n линий метро, и с каждой на каждую будет хотя бы одна пересадка. При этом не будет существовать станции с пересадками более, чем с одной линии на другую.

Васе в руки случайно попал план первой линии этого метрополитена. Из него он узнал сколько на этой линии будет станций и на какую линию будут пересадки с каждой из станций этой линии (некоторые станции на этой линии могут не быть пересадочными). Теперь ему интересно, какое минимальное количество станций может быть построено, чтобы были соблюдены все условия.

Формат входного файла

Первая строка входного файла содержит два целых числа n и k — количество линий и количество станций на первой линии ($1 \le k \le 100$, $1 \le n \le k+1$). Вторая строка содержит k целых чисел a_i — номер линии, на которую ведет пересадка с i-й станции первой линии. При этом, если $a_i = 0$, то эта станция не является пересадочной, иначе $2 \le a_i \le n$.

Формат выходного файла

В выходной файл выведите ответ на задачу — минимальное количество станций в метрополитене ЭмСити.

Примеры

metro.in	metro.out
3 3 2 3 0	4
2 1 2	1

Задача E. Двоичное число

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

> Планета Шелезяка... Полезных ископаемых на планете нет. Воды тоже нет. Атмосферы нет. Ничего на планете нет. Если что и было, роботы все истратили и живут в бедности
>
> Кир Булычев «Путешествие Алисы»

Как известно, планета Шелезяка расположена в отдаленной части нашей Галактики, в стороне от космических трасс. Воду роботы (единственные обитатели планеты во второй половине 21-го века) ликвидировали специально, чтобы не ржаветь (они опасались ливней и наводнений). В связи с этим планета оказалась решительно никому кроме роботов не нужна.

Однажды на пустынную планету прилетели пираты. То были настоящие матерые злодеи. Их сердцу была неведома жалость. Они захватили почти всех роботов и увезли их в далекую-далекую галактику и заставили работать на шахтах на планете Бендомир.

Прошло немало времени, прежде чем про это стало известно на Земле. Ричард Темпест, бесстрашный сотрудник института Времени, отправился в прошлое с целью помешать дьявольским планам пиратов. Как оказалось, пользуясь гостеприимством роботов, пираты предлагали роботам решить задачку на скорость. Однако вычислительные возможности роботов были не велики, а оптимизировать алгоритмы их работы было некому. Поэтому все, что оставалось злодеям — загрузить в трюм корабля бесчувственные тела роботов, поглощенных вычислениями.

Ричарду удалось узнать, над чем безуспешно бились электронные мозги роботов. Пираты предлагали им целое неотрицательное число n. После этого они просили узнать, что произойдет с этим числом, если к нему n раз применить операцию: взять старший бит в двоичной записи числа и поставить его в конец. Например, если один раз применить эту операцию к числу $14_{10} = 1110_2$, получится $13_{10} = 1101_2$. После повторного применения получится $11_{10} = 1011_2$.

Ричард присвистнул. Если ему удастся раздобыть программу, которая быстро решает эту задачу, он сможет отправиться во времена до прилета пиратов и загрузить в роботов эту программу, чтобы они смогли противостоять злобным пиратам.

Вы — сотрудник института Времени, которому поручено в срочном порядке написать эту программу. Помогите спасти роботов.

Формат входного файла

В первой строке входного файла содержится целое число n ($0 \le n \le 10^{18}$).

Формат выходного файла

В выходной файл выведите ответ на задачу, поставленную пиратами перед роботами.

Примеры

number.in	number.out
0	0
14	7

Задача F. Период

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

При решении некоторых задач по информатике бывает полезно использовать такое свойство последовательности последних цифр степеней чисел как *периодичность*. Рассмотрим, например, последовательность степеней числа 2: $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $\ldots$. Последние цифры этих чисел образуют такую последовательность: 1, 2, 4, 8, 6, 2, 4, 8, $\ldots$. Видно, что эта последовательность периодична с периодом 4 — второй ее элемент равен шестому, десятому, $\ldots$, третий — седьмому, одиннадцатому, $\ldots$, четвертый — восьмому, двенадцатому, $\ldots$, пятый — девятому, тринадцатому, $\ldots$.

Ваша задача состоит в том, чтобы найти период последовательности последних цифр степеней числа n в системе счисления с основанием k.

Формат входного файла

Первая строка входного файла содержит целое число n ($2 \le n \le 10^9$). Вторая строка входного файла содержит целое число k ($2 \le k \le 36$). Оба числа заданы в десятичной системе счисления.

Формат выходного файла

В выходной файл выведите длину периода рассматриваемой последовательности или `-1`, если эта последовательность не является периодичной.

Примеры

period.in	period.out
2 10	4

Задача G. Точная степень

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

Натуральное число x называется *точной степенью*, если существуют такие натуральные числа a и b ($b > 1$), что $x = a^b$. Заметим, однако, что указанные числа a и b могут определяться неоднозначно. Например, $16 = 2^4 = 4^2$, то есть для $x = 16$ существуют две такие пары (a, b).

Ваша задача состоит в том, чтобы найти все такие пары (a_i, b_i), что $x = a_i^{b_i}$.

Формат входного файла

Входной файл содержит целое число x ($2 \le x \le 10^9$).

Формат выходного файла

В первой строке выходного файла выведите число k искомых пар (a, b). В каждой из последующих k строк выведите два числа: a_i и b_i.

Примеры

power.in	power.out
3	0
16	2 2 4 4 2

Задача H. Времечко

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

У Антона есть электронные часы. В любой момент времени на их табло горят четыре цифры — первые две соответствуют текущему часу дня, последние две соответствуют текущей минуте часа. Таким образом, текущее время можно можно интерпретировать как строку, состоящую из четырех символов. Например, в течении дня на табло часов можно наблюдать: «`0000`» (ноль часов ноль минут), «`0015`» (ноль часов пятнадцать минут), «`0204`» (два часа четыре минуты), «`1239`» (двенадцать часов тридцать девять минут).

Антон загадал число n. В течении всего дня (с «`0000`» до «`2359`») Антон смотрит на табло часов, и как только наступает следующая минута, он съедает ирисок ровно столько, сколько раз десятичная запись числа n встречается как подстрока на табло его часов. Ваша задача — определить, сколько ирисок за день съел Антон.

Формат входного файла

Единственная строка входного файла содержит число n ($1 \le n \le 5000$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

time.in	time.out
1	1164
123	13
2359	1