

Задача А. Головоломка

Имя входного файла: puzzle.in
Имя выходного файла: puzzle.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вася очень любит играть в квесты. Он любит кликать тут и там мышкой и наслаждается шутками, которых много в квестах. Но там есть еще и много разных головоломок, и Вася проводит много времени, решая их. Но однажды он столкнулся с головоломкой которую никак не мог решить. К счастью, Вася отличный программист, и он смог написать программу, которая решила головоломку и помогла ему закончить квест.

Васина головоломка представляет собой матрицу 3×3 , каждая клетка которой окрашена в черный или белый цвет. Если кликнуть на клетку, то она и ее соседи меняют свой цвет на противоположный. Цель — сделать все клетки одного цвета.

Ваша задача чуть более сложная. Пусть есть N клеток, занумерованных от 1 до N . Каждая клетка имеет множество клеток, связанных с ней. Когда игрок кликает на клетку, все клетки из множества, связанного с ней, меняют свой цвет. По данным связанным множествам и начальной раскраске выведите последовательность клеток, на которые нужно кликать, чтобы все клетки приобрели один цвет. Если есть несколько различных решений, выведите любое.

Формат входных данных

Первая строка содержит единственное целое число N ($1 \leq N \leq 200$). В $i+1$ -й строке ($1 \leq i \leq N$) находится описание множества, связанного с i -й клеткой. Каждое описание начинается с целого числа k — количество клеток в множестве, а затем идут k различных целых чисел (номера клеток). Последняя строка содержит N нулей и единиц — начальная раскраска клеток.

Формат выходных данных

Если невозможно окрасить все клетки в один цвет, выведите единственное число -1 . Иначе выведите целое число L — количество кликов, необходимых для решения головоломки, а затем L чисел — номера клеток, на которые необходимо кликать.

Примеры

puzzle.in	puzzle.out
9	9
3 1 2 4	1 2 3 4 5 6 7 8 9
4 1 2 3 5	
3 2 3 6	
4 1 4 5 7	
5 2 4 5 6 8	
4 3 5 6 9	
3 4 7 8	
4 5 7 8 9	
3 6 8 9	
1 0 1 0 1 0 1 0 1	

Задача В. Квадратное уравнение

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

Детей в школе учат решать квадратные уравнения, т.е. уравнения вида

$$ax^2 + bx + c = 0,$$

где a , b и c некоторые заданные действительные числа, а x — действительное число, которое необходимо найти.

В этой задаче вам потребуется решить квадратное уравнение для многочленов с коэффициентами из нулей или единиц, и все операции производятся по модулю 2.

Даны многочлены $a(t)$, $b(t)$ и $c(t)$, найдите такой полином $x(t)$ что

$$a(t)x^2(t) + b(t)x(t) + c(t) = 0,$$

где равенство понимается как равенство многочленов. Напомним, что многочлены равны тогда и только тогда, когда равны их коэффициенты при соответствующих степенях t .

Формат входных данных

Входной файл содержит многочлены $a(t)$, $b(t)$ и $c(t)$, которые задаются их степенями, за которыми следуют коэффициенты, начиная со старшего. Нулевые многочлены в данной задаче имеют степень -1 . Степени всех многочленов не превосходят 127. Между старшим коэффициентом и степенью находится два пробела. После многочлена степени -1 также находится один пробел.

Формат выходных данных

Если есть хотя бы одно решение уравнения, выведите любое из них в таком же формате. Старший коэффициент найденного многочлена не должен быть нулевым. Степень полинома не должна превышать 512.

В противном случае напечатайте `no solution`.

Примеры

quadratic.in	quadratic.out
0 1	1 1 0
2 1 1 0	
3 1 0 0 0	

Задача С. Гладкие числа

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

Назовем число гладким, если его соседние цифры по модулю различаются не более чем на 1. Для данного N определите количество имеющих длину N .

Формат входных данных

На вход программы поступает количество цифр $N \leq 10^9$.

Формат выходных данных

Выведите ответ по модулю $10^9 + 7$.

Примеры

<code>smooth.in</code>	<code>smooth.out</code>
1	9

Задача D. Транзисторы над Пекином возвращаются

Имя входного файла: transistor2.in
Имя выходного файла: transistor2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Всемирно известный профессор В.В. Адимов продолжает свои разнообразные исследование устойчивости транзисторов. Теперь в голову ему пришла следующая задача: в доме N этажей, профессор хочет выяснить номер максимального этажа, падение с которого оставляет транзистор целым. Поскольку профессор исследует сферические транзисторы в вакууме, то можете считать что разбившись при падении с этажа f транзистор обязательно разобьется при падении с этажа $f + 1$. Дополнительно поставлено условие, что разрешено проведение не более чем K испытаний.

Эта задача была поручена именно вам, как самому успешному аспиранту профессора Адимова. Поскольку транзисторы нынче в цене, но наука все-таки дороже, то необходимо выяснить, какое минимальное количество транзисторов необходимо закупить, чтобы успешно провести эксперимент даже если вам будет катастрофически не везти.

Формат входных данных

В первой и единственной строке входного файла содержатся два целых числа N и K ($1 \leq N \leq 10^{18}$, $0 \leq K \leq 10^{18}$).

Формат выходных данных

Выведите единственное число — ответ на поставленную задачу. Если для данных N и K возможна ситуация, при которой мы не сможем получить ответ на вопрос даже имея неограниченный запас бесплатных транзисторов выведите -1 .

Примеры

transistor2.in	transistor2.out
4 2	-1
4 3	2

Задача Е. Симпатичные узоры наносят ответный удар

Имя входного файла: nice3.in
Имя выходного файла: nice3.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

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 два натуральных числа n и m . $1 \leq n \leq 10^{100}$, $1 \leq m \leq 5$, $1 \leq p \leq 10000$.

Формат выходных данных

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

Примеры

nice3.in	nice3.out
2 2 20	14
3 3 7	0

Задача F. Гладкие числа

Имя входного файла: `smooth.in`
Имя выходного файла: `smooth.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Назовем число гладким, если его цифры, начиная со старшего разряда, образуют неубывающую последовательность. Упорядочим все такие числа в возрастающем порядке и присвоим каждому номер. Вам требуется по номеру N вывести N -ое гладкое число.

Формат входных данных

На вход программы поступает номер N ($1 \leq N \leq 2147483647$).

Формат выходных данных

Выведите соответствующее номеру N гладкое число.

Примеры

<code>smooth.in</code>	<code>smooth.out</code>
3	3
11	12