

## Задача А. Игры уголовников

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 256 мегабайт

Случилось ужасное! Товарищ майор узнал, что вы репостнули мем про Иисуса, написали комментарий, оскорбляющий власть, лайкнули очередной видосик Навального, да и ещё и сходили на нанесанкционированный митинг! Теперь вы официально экстремист, а за это вам грозит тюремный срок на 100 лет!

В тюрьме вы сразу повстречали заядлого уголовника Антона, который предложил вам сыграть в особую тюремную игру уголовников. Игра заключается в следующем: есть 2 игрока, один — чётный, а другой — нечётный. В каждом раунде игры оба игрока показывают один или два пальца. Если  $a$  — сумма показанных пальцев, то если  $a$  чётно, то нечётный игрок платит чётному  $a$  монет, а если  $a$  нечётно, то чётный игрок платит нечётному  $a$  монет. Условия одинаковы, поэтому Антон предложил вам самим выбрать, чётным вы будете или нечётным. После этого каждый день своего десятилетнего заключения (всего  $365 * 100 = 36500$  дней) вы сыграете с Антоном по партии. Ваша цель — не остаться в минусе и в конце своего заключения не потерять свои деньги после всех 36500 партий. Вам лениво каждый раз приходиться к Антону и играть с ним в эту игру, поэтому вы решили написать программу, играющую за вас.

### Протокол взаимодействия

Это интерактивная задача.

В первой строке выведете 0, если вы хотите играть за чётного игрока, или 1, если вы хотите играть за нечётного.

После этого следующие 36500 ходов пройдут следующим образом. Выведите одно число 1 или 2 — количество пальцев, которое вы показываете в этом раунде. В ответ на это программа жюри, имитирующая Антона, выведет вам одно число 1 или 2 — количество пальцев, которое показывает Антон. Гарантируется, что в  $i$ -й ход выведенное интерактором число никак не зависит от того, какое число вы вывели в  $i$ -й ход, т.е. программа Антон играет честно.

После 36500 ходов программа жюри выведет ваш баланс — разницу полученных и отданных вами монет. В случае, если баланс неотрицательный, ваша программа будет считаться верной.

Все строки завершайте символом перевода строки и сбросом буфера ввода. Все числа программы жюри выводятся в новой строке. Для удобства в примере показана игра, состоящая из 4 партий.

### Пример

стандартный ввод	стандартный вывод
	0
1	1
2	1
1	2
2	2
0	

## Задача В. Эскалатор

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Илья устал от олимпиадного программирования, ушёл из университета и устроился на работу в мажоранале метрополитен. Перед ним поставили задачу определения нагрузки на эскалатор.

Пусть  $n$  человек стоят в очереди на эскалатор. В каждую секунду происходит одно из двух: либо первый человек в очереди с вероятностью  $p$  заходит на эскалатор, либо первый человек в очереди с вероятностью  $1 - p$  остаётся стоять на месте, не в силах совладать с боязнью эскалаторов, задерживая при этом всю очередь за ним.

$i$ -й в очереди не сможет зайти на эскалатор, пока на него не зайдут люди с номерами от 1 до  $i - 1$  включительно. За одну секунду может зайти только один человек. Так как эскалатор бесконечный, то, единожды зайдя на него, человек никогда с него не сойдёт, т. е. будет ехать на нем в эту и в любую последующую секунды. Илье нужно посчитать математическое ожидание количества людей, которые будут находиться на эскалаторе после  $t$  секунд.

Вам необходимо помочь ему в решении этой непростой задачи.

### Формат входных данных

В единственной строке входных данных заданы три числа  $n, p, t$  ( $1 \leq n, t \leq 2000, 0 \leq p \leq 1$ ). Числа  $n$  и  $t$  — целые, число  $p$  — вещественное, заданное ровно с двумя знаками после запятой.

### Формат выходных данных

Выведите одно вещественное число — математическое ожидание количества людей, которые будут на эскалаторе через  $t$  секунд. Абсолютная или относительная погрешность не должна превышать  $10^{-6}$ .

### Примеры

стандартный ввод	стандартный вывод
1 0.50 1	0.5000000000
1 0.50 4	0.9375000000
4 0.20 2	0.4000000000

## Задача С. Линейные уравнения

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Система линейных уравнений, как всем известно, есть множество уравнений

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &= b_1 \\ &\dots \\ a_{n1}x_1 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

Ваша задача — решить её.

### Формат входных данных

В первой строке входного файла записано целое число  $n$  ( $1 \leq n \leq 20$ ). В следующих  $n$  строках записано по  $n + 1$  целых чисел:  $a_{i1}, \dots, a_{in}, b_i$ . Все эти числа не превышают 100 по абсолютному значению.

### Формат выходных данных

Первая строка выходного файла должна содержать одно из следующих сообщений:

- `impossible` — решений нет
- `infinity` — бесконечно много решений
- `single` — единственное решение. В этом случае вторая строка должна содержать  $n$  чисел  $x_1, \dots, x_n$ , разделенных пробелами. Решение должно быть выведено с точностью не менее трех знаков после десятичной точки.

### Примеры

стандартный ввод	стандартный вывод
2 1 1 1 2 2 2	infinity
2 1 2 0 1 2 1	impossible
2 1 2 1 2 1 0	single -0.333333333 0.666666667

## Задача D. Обобщенные числа фибоначчи

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Мы чуть-чуть обобщили для вас последовательность Фибоначчи, теперь:

$$f_1 = f_2 = 1$$

$$f_i = a \cdot f_{i-1} + b \cdot f_{i-2} + c \cdot 2^i + d \cdot i + e, \text{ для } i > 2$$

Дано  $n$ , найдите значение  $f_n$ , взятое по модулю  $10^9 + 7$ .

### Формат входных данных

Неотрицательные целые числа:  $a, b, c, d, e, n$ . ( $0 \leq a, b, c, d, e \leq 10^9$ ;  $1 \leq n \leq 10^{18}$ )

### Формат выходных данных

Выведите  $f_n$ , взятое по модулю  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
1 1 0 0 0 8	21
1 2 3 4 5 6	775

## Задача Е. Гладкие числа

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
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

стандартный ввод	стандартный вывод
3	3
11	12

## Задача F. Полные квадраты

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Множество целых положительных чисел будем называть *полноквадратным*, если произведение его элементов является полным квадратом (равно 1, 4, 9, 16, 25, 36, ...).

Задано множество  $A$ . Определите, сколько непустых подмножеств  $B$  множества  $A$  являются полноквадратными.

### Формат входных данных

В первой строке записано целое число  $N$  — количество элементов множества ( $1 \leq N \leq 100$ ). Во второй строке записаны  $N$  попарно различных чисел  $a_i$  — элементы множества ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите одно число — количество полноквадратных подмножеств по модулю 1 000 000 007.

### Пример

стандартный ввод	стандартный вывод
4 49 20 500 7	3

## Задача G. Шоппинг

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

Сегодня долгожданный для всех школьников — первый день каникул нового учебного года. Наша главная героиня — Дени — учится в 10 классе. Она хорошо подготовилась к сегодняшнему дню и выяснила, что в центре города находятся  $N$  магазинов. Теперь Дени планирует вместе со своими друзьями посетить некоторые из них. В городе есть  $M$  пар магазинов  $(x_i, y_i)$ , соединенных двусторонними дорогами. Для каждой дороги известно время, которое требуется для перемещения по ней, оно одно и то же для перемещения в обоих направлениях. Никакой магазин не соединен дорогой сам с собой, никакая пара магазинов не соединена более чем одной дорогой.

Дени очень суеверна и одно из её суеверий заключается в том, что она верит, что время, потраченное на перемещения между магазинами, должно нацело делиться на  $D$ . При этом Дени с друзьями не может перемещаться между магазинами слишком долго, её путь должен занимать суммарно не больше  $K$ . Как и все девушки, Дени очень любопытна. Она хочет выяснить, сколько существует различных способов начать свой путь в некотором магазине, перемещаться по дорогам между магазинами, и закончить путь в некотором магазине (возможно посещая по пути некоторые магазины и/или дороги более одного раза). Дени помнит, что у нее есть друг-программист — вы — и она просит написать программу, которая вычислит количество корректных способов перемещаться между магазинами. Дени считает способ корректным, если её время в пути не превышает  $K$  и делится на  $D$ . Вы немедленно указываете Дени, что количество путей может быть слишком большим, поэтому Дени просит вывести остаток от деления количества путей на число 1 000 000 007.

### Формат входных данных

На первой строке ввода находятся четыре целых числа  $N$ ,  $M$ ,  $D$  и  $K$  ( $2 \leq N \leq 30$ ,  $2 \leq M \leq 435$ ,  $2 \leq D \leq K \leq 10^9$ ).

На каждой из следующих строк находятся по три целых числа  $x_i$ ,  $y_i$  и  $t_i$  ( $1 \leq x_i, y_i \leq N$ ,  $1 \leq t_i \leq 10^9$ ) — они задают двустороннюю дорогу между магазинами  $x_i$  и  $y_i$ , перемещение по которой занимает  $t_i$  ( $1 \leq i \leq M$ ).

### Формат выходных данных

Выведите остаток от деления количества искомых путей на 1 000 000 007.

## Примеры

стандартный ввод	стандартный вывод
3 3 2 2 1 2 1 2 3 2 3 1 1	8
5 7 5 10 1 3 8 2 5 7 3 4 3 1 4 2 2 3 1 1 5 4 4 5 4	58
5 9 2 20 1 2 1 2 3 2 3 1 1 3 4 1 4 5 2 5 3 1 1 5 1 2 4 1 2 5 1	989802661
5 7 500000 500000 1 3 8 2 5 7 3 4 3 1 4 2 2 3 1 1 5 4 4 5 4	598634781



## Задача Н. Транзисторы над Пекином возвращаются

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
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

стандартный ввод	стандартный вывод
4 2	-1
4 3	2

## Задача I. Круги ожидания

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Фишку поставили на поле с системой координат в точку  $(0, 0)$ .

Каждую секунду фишка перемещается случайным образом. Пусть в некоторый момент времени фишка находится на позиции  $(x, y)$ . Тогда через секунду она с вероятностью  $p_1$  окажется на позиции  $(x - 1, y)$ , с вероятностью  $p_2$  окажется на позиции  $(x, y - 1)$ , с вероятностью  $p_3$  окажется на позиции  $(x + 1, y)$ , а с вероятностью  $p_4$  — на позиции  $(x, y + 1)$ . Гарантируется, что  $p_1 + p_2 + p_3 + p_4 = 1$ .

Требуется посчитать математическое ожидание времени, через которое фишка впервые отдалится от начала координат на расстояние, большее  $R$  (то есть будет выполнено  $x^2 + y^2 > R^2$ ).

### Формат входных данных

В первой строке вводятся пять целых чисел  $R, a_1, a_2, a_3$  и  $a_4$  ( $0 \leq R \leq 50, 1 \leq a_1, a_2, a_3, a_4 \leq 1000$ ). Вероятности  $p_i$  вычисляются по формуле  $p_i = \frac{a_i}{a_1 + a_2 + a_3 + a_4}$ .

### Формат выходных данных

Можно показать, что ответом на задачу всегда является рациональное число вида  $\frac{P}{Q}$ , где  $Q \not\equiv 0 \pmod{10^9 + 7}$ .

В качестве ответа выведите  $P \cdot Q^{-1}$  по модулю  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
0 1 1 1 1	1
1 1 1 1 1	666666674
1 1 2 1 2	538461545

### Замечание

В первом тестовом примере фишка изначально находится на расстоянии 0 от начала координат. Через секунду фишка сдвинется на 1 в какую-то сторону, поэтому расстояние от нее до начала координат станет равным 1.

Ответы на второй и третий тесты:  $\frac{8}{3}$  и  $\frac{36}{13}$ .