

Задача А. Две улитки (1st Division Only!)

Имя входного файла: A.in
Имя выходного файла: A.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Из верхнего левого угла прямоугольного поля, состоящего из M строк и N столбцов, улитка обошла по спирали по часовой стрелке. При этом все клетки она пронумеровала числами $1, 2, 3, \dots$ последовательно в порядке обхода.

Теперь на этом поле появляется другая улитка, которой необходимо попасть из клетки (i_1, j_1) в клетку (i_2, j_2) . Каждую секунду она может перемещаться в соседнюю по горизонтали или вертикали клетку, но лишь при условии, что номер этой клетки отличается от номера предыдущей не более, чем на k .

Ваша задача – узнать сколько времени потребуется второй улитке, чтобы добраться до нужной клетки.

Формат входного файла

Во входном файле записаны целые числа $M, N, k, i_1, j_1, i_2, j_2$ ($1 \leq M, N \leq 10^{18}, 1 \leq k \leq 2 \cdot 10^{18}, 1 \leq i_1, i_2 \leq M, 1 \leq j_1, j_2 \leq N$).

Формат выходного файла

В выходной файл выведите одно целое число – минимальное время, за которое вторая улитка сможет добраться до нужной клетки.

Примеры

A.in	A.out
3 5 1 1 2 3 4	6
5 5 7 3 1 3 3	4

Задача В. Треугольная комната (1st Division Only!)

Имя входного файла: B.in
Имя выходного файла: B.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Во многих книгах по занимательной математике приводится такая задача. *Расставить по периметру треугольной комнаты 3 стула так, чтобы у каждой стены стояло по 2. Ее решение — поставить по стулу в каждый из углов комнаты.*

Рассмотрим более общую задачу. Пусть комната представляет собой треугольник ABC . Дано общее количество стульев n , количество стульев n_{AB} , которое должно стоять у стены AB , количество стульев n_{BC} , которое должно стоять у стены BC , количество стульев n_{AC} , которое должно стоять у стены AC . Необходимо найти общее количество расстановок стульев, удовлетворяющих условию. Стулья можно ставить только в углы комнаты и вдоль стен, в центр комнаты стулья ставить нельзя. В любой из углов можно поставить произвольное количество стульев.

Формат входного файла

Входной файл содержит целые числа $n, n_{AB}, n_{BC}, n_{AC}$ ($0 \leq n, n_{AB}, n_{BC}, n_{AC} \leq 10^{18}$).

Формат выходного файла

В первой строке выходного файла выведите количество различных вариантов расстановки стульев. В случае, когда есть хотя бы один вариант, выведите во второй строке 6 целых неотрицательных чисел: $k_A, k_{AB}, k_B, k_{BC}, k_C, k_{AC}$ — соответственно количество стульев, которые необходимо поставить в угол A , вдоль стены AB , в угол B , вдоль стены BC , в угол C и вдоль стены AC .

Примеры

B.in	B.out
3 2 2 2	1 1 0 1 0 1 0
3 3 2 2	0

Задача С. Игра S-Грюнди (1st Division Only!)

Имя входного файла: C.in
Имя выходного файла: C.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Пусть S – некоторое множество целых неотрицательных чисел. Построим по нему последовательность $G_S(n)$ ($n > 0$) по следующей формуле

$$G_S(n) = \text{mex}\{G_S(x) \text{ xor } G_S(y) : x, y > 0, x + y = n, |x - y| \notin S\},$$

где $\text{mex } A$ – наименьшее целое неотрицательное число, не принадлежащее множеству A , а $a \text{ xor } b$ – результат побитового сложения чисел a и b по модулю 2.

Будем называть множество S хорошим, если $G_S(n + 4) = G_S(n)$ при $n > 4$, а значения $G_S(n)$ при $n = 1, 2, 3, 4, 5, 6, 7, 8$ равны $0, 0, 0, 1, 0, 2, 1, 3$ соответственно.

Дано $n \geq 0$. Требуется найти количество хороших множеств, элементы которых не превосходят n , по модулю $10^9 + 7$.

Формат входного файла

В единственной строке входного файла задано целое неотрицательное число $n \leq 10^{18}$.

Формат выходного файла

В единственную строку выходного файла выведите ответ на задачу.

Примеры

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

Задача D. Получить одинаковые (1st Division Only!)

Имя входного файла: D.in
Имя выходного файла: D.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

На столе лежат карточки, на каждой из которых написано некоторое число. Кроме того, имеется довольно большая стопка чистых карточек. За один ход разрешается убрать со стола две карточки с одинаковыми числами, затем взять из стопки две чистые, написать на каждой из них по одному числу и положить на стол.

Требуется узнать как за минимальное количество ходов сделать так, чтобы на столе были карточки, на которых написаны одинаковые числа.

Формат входного файла

В первой строке задается количество лежащих на карточек N ($1 \leq N \leq 30000$), а во второй – N чисел, написанных на них. Все числа натуральные и не превышают 10^9 .

Формат выходного файла

В первой строке выведите минимальное количество операций L , необходимых для того, чтобы получить на столе все одинаковые карточки. В последующих L строках выведите по четыре числа, определяющие, что нужно делать в соответствующий ход: первое и второе – числа, записанные на убираемых карточках, третье и четвертое – числа, которые нужно написать на новых карточках. Если невозможно сделать так, чтобы на столе оказались все одинаковые карточки, выведите одно число -1 .

Примеры

D.in	D.out
3 1 2 2	1 2 2 1 1
6 3 3 3 4 4 4	2 3 3 4 3 3 3 4 4

Задача Е. Прыжки на полосе (1st Division Only!)

Имя входного файла: E.in
Имя выходного файла: E.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Горизонтальная полоса $1 \times N$ разбита на клетки размера 1×1 . В клетке с номером s стоит пешка. В первый ход она должна переместиться на одну клетку (влево или вправо). Второй ход должен быть сделан уже на 2 клетки и т.д. Каждый следующий ход выполняется с шагом, на единицу больше, чем предыдущий. Так продолжается до тех пор, пока у пешки не останется ходов, не выводящих ее за пределы полосы. Определите какое максимальное и минимальное число ходов может сделать пешка.

Формат входного файла

В первой строке входного файла задано количество тестов T ($1 \leq T \leq 20$). В каждой из последующих T строк заданы по два целых числа N и s ($1 \leq N \leq 10^{18}$, $1 \leq s \leq N$), определяющих данные для соответствующего теста.

Формат выходного файла

Выходной файл должен состоять из T строк, в каждой из которых содержится по два целых числа — максимальное и минимальное количество ходов пешки для соответствующего теста.

Пример

E.in	E.out
2	2 2
3 2	5 3
6 2	

Задача F. Произведение взаимно простых (1st Division Only!)

Имя входного файла: F.in
Имя выходного файла: F.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Вам дано натуральное число m . Требуется найти произведение всех натуральных чисел не превосходящих m и взаимно простых с m , и выдать его остаток при делении на m .

Формат входного файла

В единственной строке входного файла задано натуральное число $m \leq 10^{18}$.

Формат выходного файла

В единственную строку выходного файла выведите ответ на задачу.

Примеры

F.in	F.out
1	0
2	1
3	2
4	3
5	4
6	5
7	6

Задача G. Делимость биномиальных коэффициентов (1st Division Only!)

Имя входного файла: G.in
Имя выходного файла: G.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Обозначим $C_n^i = \frac{n!}{i!(n-i)!}$, где $0 \leq i \leq n$ и n, i — целые числа. Вам даны натуральное число n и простое число p . Обозначим через k наибольшее целое неотрицательное число, для которого $p^k \leq n$. Далее обозначим для $j \geq 0$ через a_j количество чисел $i \in \{0, 1, \dots, n\}$, для которых C_n^i делится на p^j , но не делится на p^{j+1} . Легко проверить, что $a_j = 0$ при $j > k$. Поэтому от вас требуется найти числа a_0, a_1, \dots, a_k .

Формат входного файла

В единственной строке входного файла заданы натуральное число $n \leq 10^{18}$ и простое число $p < 10^{18}$.

Формат выходного файла

В единственную строку выходного файла выведите через пробел числа a_0, a_1, \dots, a_k .

Примеры

G.in	G.out
4 2	2 1 2
8 3	9 0
4 5	5

Задача Н. Факториал и 4-я степень (1st Division Only!)

Имя входного файла: Н.in
Имя выходного файла: Н.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Для натурального m и простого p обозначим через $\deg_p(m)$ показатель вхождения числа p в каноническое разложение числа m на простые множители. Вам даны натуральное число n и простое число p . Требуется вычислить остаток числа $n!/p^{\deg_p(n!)}$ при делении на p^4 . Другими словами $n!$ делят нацело на p пока это возможно, и полученное число берут по модулю p^4 . Число n будет задано в p -ичной записи, то есть $n = d_{L-1}p^{L-1} + d_{L-2}p^{L-2} + \dots + d_1p + d_0$, где $d_{L-1}, d_{L-2}, \dots, d_1, d_0$ – некоторые целые неотрицательные числа, меньшие p (цифры числа n в p -ичной записи).

Формат входного файла

В первой строке входного файла задано простое число p ($3 < p < 55000$) и натуральное число $L \leq 500000$ – длина p -ичной записи числа n . Во второй строке записаны через пробел числа $d_{L-1}, d_{L-2}, \dots, d_1, d_0$, при этом $d_{L-1} > 0$.

Формат выходного файла

В единственную строку выходного файла выведите ответ на задачу.

Примеры

Н.in	Н.out
5 5 1 2 1 3 4	607
11 10 2 0 3 6 9 0 6 2 6 10	8891

Задача I. Сумма двух квадратов (1st Division Only!)

Имя входного файла: I.in
Имя выходного файла: I.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Как известно, любое простое число p вида $4k+1$ представимо в виде суммы двух квадратов натуральных чисел, причем единственным способом. В данной задаче вам предлагается найти такое представление. Чтобы облегчить задачу, будут рассматриваться только простые числа вида $8k+5$.

Формат входного файла

В первой строке входного файла задано натуральное число $T \leq 1000$, количество простых чисел вида $8k+5$, которые вам надо представить в виде суммы двух квадратов натуральных чисел. В последующих T строках заданы сами эти числа. Гарантируется, что каждое из них является простым числом, дает остаток 5 при делении на 8 и не превосходит 10^{18} .

Формат выходного файла

Для каждого простого числа p из входного файла выведите в отдельной строке через пробел пару натуральных чисел x и y такую, что $x < y$ и $x^2 + y^2 = p$.

Примеры

I.in	I.out
4	1 2
5	2 3
13	2 5
29	260483990 965478167
999999999999999989	

Задача J. Сказки Шахразады

Имя входного файла: J.in
Имя выходного файла: J.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

...И Шахразаду застигло утро, и она прекратила дозволенные речи...

«Тысяча и одна ночь»

Когда царь Шахрияр узнал об измене своей жены, то вошел он во дворец и отрубил голову своей жене, и рабам, и невольницам. И с тех пор стал он каждый день брать невинную девушку в жены, а потом на утро убивал ее, и так продолжалось в течение трех лет.

И возопили жители Багдада и бежали со своими дочерьми, и в городе не осталось ни одной девушки.

И вот потом царь приказал своему везирию привести ему, по обычаю, девушку, и везирь вышел и стал искать, но не нашел девушки и отправился в свое жилище, угнетенный и подавленный, боясь для себя зла от царя. А у царского везирия было две дочери: старшая – по имени Шахразада, и младшая – по имени Дуньязада.

Узнав от отца-везирия от начала до конца про все, что у него случилось с царем, Шахразада тот час же стала упрашивать отца, чтобы выдал ее за Шахрияра. И не смог везирь отговорить свою дочь и тогда снарядил ее и отвел к царю Шахрияру. А Шахразада подучила сестру свою, чтобы, когда они будут прощаться, та попросила рассказать какую-либо историю.

И так все случилось, и царь, мучившийся бессонницей, обрадовался, и позволил дочери везирия начать свой рассказ. Но когда настало утро, Шахразада прекратила дозволенные речи на самом интересном месте, и Шахрияру ничего не оставалось, кроме как оставить ее в живых, чтобы услышать окончание истории.

Так продолжалось много ночей: Шахразада продолжала недорассказанную сказку прошлой ночи с того места, где остановилась на рассвете, и как только среди ночи заканчивалась одна сказка, Шахразада тут же начинала другую, увлекая царя новым сюжетом. Единственное за чем ей нужно было следить – чтобы сказки не повторялись и (быть может, кроме последней) не заканчивались под утро. Тогда уже либо царь убьет Шахразаду, либо к тому времени настолько полюбит ее, что рука не поднимется. Естественно, чем больше она историй расскажет, тем больше будет любовь Шахрияра, и тем больше шансов остаться в живых.

Формат входного файла

В первой строке задается целое число N ($1 \leq N \leq 10000$) – количество сказок, которые знает Шахразада, а в следующей строке – N натуральных чисел, не превышающих 100 – продолжительности по времени сказок в часах. Считается, что каждая ночь длится 8 часов.

Формат выходного файла

В первую строку выведите наибольшее количество сказок, которое сможет рассказать Шахразада, а во второй – длительности этих сказок, в том порядке, в котором она должна их рассказывать.

Примеры

J.in	J.out
6 1 2 3 4 5 6	6 1 2 3 4 5 6
2 8 9	2 9 8
4 4 4 8 12	3 4 8 12

Задача К. Попарно различные расстояния

Имя входного файла: K.in
Имя выходного файла: K.out
Ограничение по времени: 4 seconds
Ограничение по памяти: 256 мегабайт

Для данного натурального N требуется построить на плоскости множество из N точек с целыми координатами, все попарные расстояния между которыми попарно различны.

Формат входного файла

В единственной строке входного файла задано натуральное число $N \leq 200$ — количество точек в множестве.

Формат выходного файла

В выходной файл выведите координаты точек построенного множества (каждую точку в отдельной строке, координаты через пробел). Координаты точек не должны превышать по модулю 800. Гарантируется, что такое множество существует. Если таких множеств несколько, можно выдать любое.

Примеры

K.in	K.out
1	800 -800
2	0 0 0 1
4	0 0 0 1 0 3 0 7

Задача L. Две улитки (2nd Division Only!)

Имя входного файла: L.in
Имя выходного файла: L.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Из верхнего левого угла прямоугольного поля, состоящего из M строк и N столбцов, улитка обошла по спирали по часовой стрелке. При этом все клетки она пронумеровала числами $1, 2, 3, \dots$ последовательно в порядке обхода.

Теперь на этом поле появляется другая улитка, которой необходимо попасть из клетки (i_1, j_1) в клетку (i_2, j_2) . Каждую секунду она может перемещаться в соседнюю по горизонтали или вертикали клетку, но лишь при условии, что номер этой клетки отличается от номера предыдущей не более, чем на k .

Ваша задача – узнать сколько времени потребуется второй улитке, чтобы добраться до нужной клетки.

Формат входного файла

Во входном файле записаны целые числа $M, N, k, i_1, j_1, i_2, j_2$ ($1 \leq M, N \leq 1000, 1 \leq k \leq 20000, 1 \leq i_1, i_2 \leq M, 1 \leq j_1, j_2 \leq N$).

Формат выходного файла

В выходной файл выведите одно целое число – минимальное время, за которое вторая улитка сможет добраться до нужной клетки.

Примеры

L.in	L.out
3 5 1 1 2 3 4	6
5 5 7 3 1 3 3	4

Задача М. Треугольная комната (2nd Division Only!)

Имя входного файла: M.in
Имя выходного файла: M.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Во многих книгах по занимательной математике приводится такая задача. *Расставить по периметру треугольной комнаты 3 стула так, чтобы у каждой стены стояло по 2. Ее решение — поставить по стулу в каждый из углов комнаты.*

Рассмотрим более общую задачу. Пусть комната представляет собой треугольник ABC . Дано общее количество стульев n , количество стульев n_{AB} , которое должно стоять у стены AB , количество стульев n_{BC} , которое должно стоять у стены BC , количество стульев n_{AC} , которое должно стоять у стены AC . Необходимо найти общее количество расстановок стульев, удовлетворяющих условию. Стулья можно ставить только в углы комнаты и вдоль стен, в центр комнаты стулья ставить нельзя. В любой из углов можно поставить произвольное количество стульев.

Формат входного файла

Входной файл содержит целые числа $n, n_{AB}, n_{BC}, n_{AC}$ ($0 \leq n, n_{AB}, n_{BC}, n_{AC} \leq 1000$).

Формат выходного файла

В первой строке выходного файла выведите количество различных вариантов расстановки стульев. В случае, когда есть хотя бы один вариант, выведите во второй строке 6 целых неотрицательных чисел: $k_A, k_{AB}, k_B, k_{BC}, k_C, k_{AC}$ — соответственно количество стульев, которые необходимо поставить в угол A , вдоль стены AB , в угол B , вдоль стены BC , в угол C и вдоль стены AC .

Примеры

M.in	M.out
3 2 2 2	1 1 0 1 0 1 0
3 3 2 2	0

Задача N. Получить одинаковые (2nd Division Only!)

Имя входного файла: N.in
Имя выходного файла: N.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

На столе лежат карточки, на каждой из которых написано некоторое число. Кроме того, имеется довольно большая стопка чистых карточек. За один ход разрешается убрать со стола две карточки с одинаковыми числами, затем взять из стопки две чистые, написать на каждой из них по одному числу и положить на стол.

Требуется узнать как сделать так, чтобы на столе были карточки, на которых написаны одинаковые числа.

Формат входного файла

В первой строке задается количество лежащих на карточек N ($1 \leq N \leq 30000$), а во второй – N чисел, написанных на них. Все числа натуральные и не превышают 10^9 .

Формат выходного файла

В первой строке выведите количество операций L , необходимых для того, чтобы получить на столе все одинаковые карточки. Значение L необязательно должно быть минимально возможным, но не должно превосходить N . В последующих L строках выведите по четыре числа, определяющие, что нужно делать в соответствующий ход: первое и второе – числа, записанные на убираемых карточках, третье и четвертое – числа, которые нужно написать на новых карточках. Если невозможно сделать так, чтобы на столе оказались все одинаковые карточки, выведите одно число -1 .

Примеры

N.in	N.out
3 1 2 2	1 2 2 1 1
6 3 3 3 4 4 4	2 3 3 4 3 3 3 4 4

Задача О. Прыжки на полосе (2nd Division Only!)

Имя входного файла: 0.in
Имя выходного файла: 0.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Горизонтальная полоса $1 \times N$ разбита на клетки размера 1×1 . В клетке с номером s стоит пешка. В первый ход она должна переместиться на одну клетку (влево или вправо). Второй ход должен быть сделан уже на 2 клетки и т.д. Каждый следующий ход выполняется с шагом, на единицу больше, чем предыдущий. Так продолжается до тех пор, пока у пешки не останется ходов, не выводящих ее за пределы полосы. Определите какое максимальное и минимальное число ходов может сделать пешка.

Формат входного файла

В первой строке входного файла задано количество тестов T ($1 \leq T \leq 20$). В каждой из последующих T строк заданы по два целых числа N и s ($1 \leq N \leq 1000$, $1 \leq s \leq N$), определяющих данные для соответствующего теста.

Формат выходного файла

Выходной файл должен состоять из T строк, в каждой из которых содержится по два целых числа — максимальное и минимальное количество ходов пешки для соответствующего теста.

Пример

0.in	0.out
2	2 2
3 2	5 3
6 2	

Задача Р. Произведение взаимно простых (2nd Division Only!)

Имя входного файла: P.in
Имя выходного файла: P.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Вам дано натуральное число m . Требуется найти произведение всех натуральных чисел не превосходящих m и взаимно простых с m , и выдать его остаток при делении на m .

Формат входного файла

В единственной строке входного файла задано натуральное число $m \leq 10^{12}$.

Формат выходного файла

В единственную строку выходного файла выведите ответ на задачу.

Примеры

P.in	P.out
1	0
2	1
3	2
4	3
5	4
6	5
7	6

Задача Q. Неделимость биномиальных коэффициентов (2nd Division Only!)

Имя входного файла: Q.in
Имя выходного файла: Q.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Обозначим $C_n^i = \frac{n!}{i!(n-i)!}$, где $0 \leq i \leq n$ и n, i — целые числа. Вам даны натуральное число n и простое число p . Требуется найти количество чисел $i \in \{0, 1, \dots, n\}$, для которых C_n^i не делится на p .

Формат входного файла

В единственной строке входного файла заданы натуральное число $n \leq 10^{18}$ и простое число $p < 10^{18}$.

Формат выходного файла

В единственную строку выходного файла выведите ответ на задачу.

Примеры

Q.in	Q.out
4 2	2
8 3	9
4 5	5

Задача R. Иррациональные попарные расстояния (2nd Division Only!)

Имя входного файла: R.in
Имя выходного файла: R.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Для данного натурального N требуется построить на плоскости множество из N точек, удовлетворяющее следующим условиям

- координаты точек являются целыми числами;
- расстояние между любыми двумя точками множества является иррациональным числом;
- никакие три точки множества не лежат на одной прямой.

Формат входного файла

В единственной строке входного файла задано натуральное число $N \leq 1000$ – количество точек в множестве.

Формат выходного файла

В выходной файл выведите координаты точек построенного множества (каждую точку в отдельной строке, координаты через пробел). Координаты точек не должны превышать по модулю 1000000. Гарантируется, что такое множество существует. Если таких множеств несколько, можно выдать любое.

Примеры

R.in	R.out
1	1000000 -1000000
2	0 0 1 1
3	0 0 1 1 -1 2

Задача S. Минимаксное паросочетание (2nd Division Only!)

Имя входного файла: S.in
Имя выходного файла: S.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Задан полный двудольный граф, каждая доля которого состоит из N вершин. Каждой из вершин приписан вес – целое число. Вес ребра определяется как произведение весов вершин, которые оно соединяет.

Как известно, паросочетанием в графе называется множество ребер, попарно не имеющих общих вершин. Паросочетание называют совершенным, если оно покрывает все вершины графа, то есть каждая вершина графа инцидентна какому-либо ребру паросочетания.

Ваша задача – найти совершенное минимаксное паросочетание, то есть такое, что максимальный из весов ребер паросочетания будет минимально возможным.

Формат входного файла

В первой строке входного файла задано целое число N ($1 \leq N \leq 10^5$). Во второй строке – N целых чисел, не превосходящих по абсолютной величине 10^9 . i -ое число в строке обозначает вес i -ой вершины первой доли графа. В третьей строке аналогичным образом представлены веса вершин второй доли. Вершины каждой доли имеют номера от 1 до N .

Формат выходного файла

В первой строке выходного файла выведите одно целое число – вес совершенного минимаксного паросочетания, т.е. максимальный из весов его ребер. А во второй строке описание этого паросочетания – N целых чисел, i -ое из которых обозначает номер вершины во второй доле, с которой соединена ребром паросочетания i -ая вершина первой доли.

Пример

S.in	S.out
3	27
1 2 3	2 3 1
9 10 11	

Задача Т. Количество квадратичных вычетов (2nd Division Only!)

Имя входного файла: T.in
Имя выходного файла: T.out
Ограничение по времени: 2 seconds
Ограничение по памяти: 256 мегабайт

Пусть m некоторое натуральное число. Число $a \in \{0, 1, \dots, m - 1\}$ называется квадратичным вычетом по модулю m , если существует такое целое число x , что $x^2 - a$ делится на m . Вам дано m и требуется найти количество квадратичных вычетов по модулю m .

Формат входного файла

В первой строке входного файла задано натуральное число $T \leq 100$, количество натуральных чисел m в файле. В последующих T строках заданы сами эти числа. Гарантируется, что каждое из них не превосходит 10^{12} .

Формат выходного файла

Для каждого натурального числа m из входного файла выведите в отдельной строке количество квадратичных вычетов по модулю m .

Примеры

T.in	T.out
5	1
1	2
2	2
3	2
4	4
12	