Задача A. Бой с числами

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мебибайта

Звёздочка и Марко попали в мир, где их атакуют полчища чисел! К счастью, друзья знают, что делать.

Каждое число сначала получает заряд из бинарного бластера Марко, отчего предстаёт перед героями в двоичной записи. После этого Звёздочка один или несколько раз применяет суперудар: переворачивает двоичную запись числа, после чего вычитает из него единицу. Когда число становится нулём, оно рассыпается.

Только что друзья вступили в бой с числом n. Сколько раз Звёздочке придётся применить свой суперудар, чтобы число рассыпалось?

Переворачивание работает так: первая двоичная цифра записи меняется местами с последней, вторая — с предпоследней, и так далее. Если при этом появились ведущие нули, они удаляются из записи.

Формат входных данных

В первой строке задано целое число n ($1 \leqslant n \leqslant 10^9$).

Формат выходных данных

Выведите одно целое число: сколько раз нужно применить суперудар, чтобы число n стало нулём и рассыпалось.

Примеры

стандартный ввод	стандартный вывод	пояснение
4	1	$4_{10} = 100_2$ $100_2 \to 001_2 \to 0$
3	2	$3_{10} = 11_2$ $11_2 \to 11_2 \to 10_2$ $10_2 \to 01_2 \to 0$
11	3	$11_{10} = 1011_2$ $1011_2 \to 1101_2 \to 1100_2$ $1100_2 \to 0011_2 \to 10_2$ $10_2 \to 01_2 \to 0$

Задача B. Время магического числа

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мебибайта

Давным-давно в сказочном королевстве Времени жил добрый волшебник Николай. Он обожал магическое число 239, которое приносило ему удачу и радость каждый раз, когда он видел его на волшебных электронных часах королевства, расположенных на главной площади.

Эти чудесные часы показывали время, состоящее из четырёх цифр: две цифры для часов и две для минут. Если число часов или минут было меньше 10, то первой цифрой шёл 0.

Николай может заколдовать время, чтобы в каждых сутках стало H часов, пронумерованных от 0 до $H-1$, а в каждом часе — M минут, пронумерованных от 0 до $M-1$. Для этого заклинания Николай может выбрать любые H и M от 10 до 100, включительно.

Часы показывают *счастливое* время, если цифры на часах похожи на заветное сочетание 239:

- Когда часы показывают `X2:39`, где первая цифра `X` может принимать значение от `0` до `9`.
- Или `2X:39`, где цифра `X` может принимать значение `0`, `2` или `3` (так как `0` не мешает восприятию магического числа).
- Или `23:X9`, где цифра `X` может принимать значение `0`, `3` или `9`.
- Или даже `23:9X`, где цифра `X` может принимать значение от `0` до `9`.

Однажды перед Николаем встал вопрос: «Если известны H и M, сколько минут за одни полные сутки часы показывают счастливое время?» Помогите Николаю найти это число.

Формат входных данных

В первой строке через пробел записаны целые числа H и M — число часов в сутках и число минут в часе ($10 \leqslant H, M \leqslant 100$).

Формат выходных данных

Выведите одно число: сколько минут за полные сутки часы показывают счастливое время.

Пример

стандартный ввод	стандартный вывод
24 60	6

Задача C. Инфляция

Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мебибайта

В далёком волшебном королевстве каждому магу при поступлении на службу назначалось фиксированное ежемесячное жалованье.

Но вот беда: пока маг усердно колдовал, цены увеличивались в течение года из-за инфляции.

Однако мудрый король для сдерживания бурного роста цен объявлял в первый день первого месяца года итоговый годовой процент инфляции p и разрешал повышать цены лишь раз в месяц: ровно за день до того, когда выплачивалась зарплата магам. Зарплата магам выплачивается в последний день каждого месяца.

Более того, для удобства счёта он установил, что в году всего десять месяцев, а в каждом месяце тридцать дней. Цены каждый месяц должны были расти ровно на $p/10$ процентов от цены товара на **начало** года. Таким образом, стоимость любого товара за год увеличивалась ровно на p процентов.

Текущая система выплаты зарплат устроена так. В первый день первого месяца года назначается зарплата в n золотых, которую они будут получать в конце каждого месяца года. При этом каждый год зарплаты индексируются: если ежемесячная зарплата в какой-то год — n золотых, а процент инфляции — p, то ежемесячная зарплата на следующий год будет на p процентов больше, то есть составит $n + (np/100)$ золотых.

Идеальное решение? Нет! Маги всё равно остаются недовольны: цены растут ежемесячно, а зарплата повышается лишь раз в год. Маги предлагают следующую альтернативу, называемую *прогрессивной системой выплаты зарплат*: в последний день каждого месяца в качестве зарплаты выплачивается сумма денег, имеющая в этот момент такую же покупательную способность, какую зарплата в n золотых имела в первый день первого месяца года. Иными словами, за эту сумму можно купить ровно такое же количество любого товара, какое можно было купить за n золотых в первый день первого месяца.

Для оценки этой несправедливости король поручил вам выяснить, исходя из ежемесячной заработной платы мага в начале года, какую сумму он недополучает за год при текущей системе выплаты зарплат по сравнению с прогрессивной системой выплаты зарплат.

Формат входных данных

В первой строке через пробел записаны целые числа n и p — ежемесячное жалованье мага в золотых и годовой процент инфляции ($10^3 \leqslant n \leqslant 10^5$; $0 \leqslant p \leqslant 100$).

Формат выходных данных

Выведите сумму, которую маг недополучит за этот год по сравнению с прогрессивной системой выплаты зарплат. Помните, что в волшебном королевстве в году десять месяцев. Ответ считается верным, если абсолютная или относительная погрешность не превосходит 10^{-4}.

Пример

стандартный ввод	стандартный вывод
4346 1	239.03

Задача D. Девять из десяти

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мебибайта

Безумный учёный провёл n независимых друг от друга однотипных экспериментов и заявил, что x из них завершились успешно. Общеизвестно, что безумный учёный ошибается ровно в 90% случаев при определении успешности отдельно взятого эксперимента. Ваша задача — написать программу, вычисляющую для всех x от 0 до n минимальное и максимальное возможное значение числа успешных экспериментов. Гарантируется, что общее число экспериментов всегда делится на 10.

Формат входных данных

В первой строке задано целое число n, кратное десяти ($10 \leqslant n \leqslant 10\,000$).

Формат выходных данных

Выведите $n + 1$ строку. В i-й строке выведите два целых числа, разделённых пробелом — минимальное и максимальное значение числа успешных экспериментов для $x = i - 1$.

Пример

стандартный ввод	стандартный вывод
10	9 9 8 10 7 9 6 8 5 7 4 6 3 5 2 4 1 3 0 2 1 1

Задача E. Грядка

Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мебибайта

В одном чудесном краю, где солнце светит ярче, а цветы растут пышнее, жил-был добрый садовод Николай. С нетерпением он ждал, когда же наступит время открыть дачный сезон и заняться своим любимым делом: возделыванием грядок.

Однажды, собравшись с силами, Николай привёз с собой волшебные пары досок. Доски в каждой паре были одинаковой длины, а доски из разных пар могли иметь разную длину. Николай мечтал создать *красивую* прямоугольную грядку, где бы росли самые прекрасные овощи и цветы. Для этого ему нужно было использовать все свои доски, не распиливая их, ставя парные доски на противоположные стороны прямоугольника.

И вот, стоя перед своим садом, Николай задумался: «Какая площадь может получиться у красивой грядки?» Помогите ему выяснить, какую минимальную и максимальную положительную площадь он может получить.

Формат входных данных

В первой строке записано целое число n — число пар досок ($2 \leqslant n \leqslant 7$).

Во второй строке заданы n целых чисел — длины досок в первой, второй, …, n-й паре. Эти числа лежат в пределах от 1 до 10^8.

Формат выходных данных

Выведите два целых числа: минимальную и максимальную площадь красивой грядки.

Примеры

стандартный ввод	стандартный вывод
2 10 239	2390 2390
3 1 2 2	4 6

Задача F. Наибольшая площадь

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мебибайта

Дан прямоугольник на плоскости с вершинами в целых точках. Вычислите площадь наибольшего эллипса, полностью помещающегося в этот прямоугольник. Напомним, что *прямоугольник* — это выпуклый четырёхугольник, у которого все углы прямые, а *эллипс* — фигура на плоскости, которая задаётся двумя точками-фокусами F_1 и F_2 (необязательно с целыми координатами; возможно, $F_1 = F_2$) и вещественным числом $d > |F_1F_2|$ и состоит из таких точек P на плоскости, для которых $|F_1P| + |F_2P| \leqslant d$.

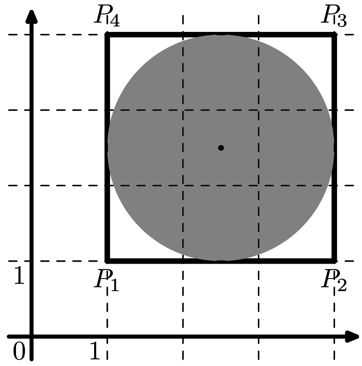
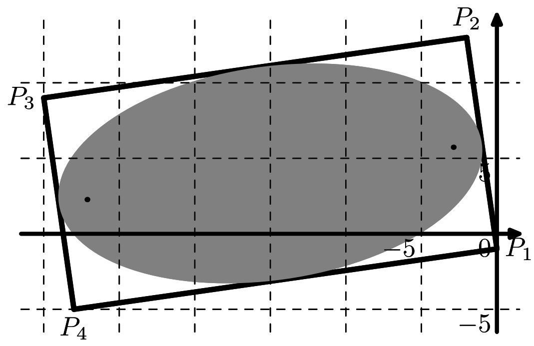
Формат входных данных

В единственной строке входных данных находится восемь целых чисел $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$ — координаты четырёх точек ($-1000 \leqslant x_i, y_i \leqslant 1000$). Точки (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) являются вершинами прямоугольника положительной площади, перечисленными против часовой стрелки.

Формат выходных данных

Выведите одно вещественное число — наибольшую площадь эллипса, который можно начертить внутри прямоугольника, заданного во входных данных. Ответ считается верным, если его относительная или абсолютная погрешность не превосходит 10^{-6}.

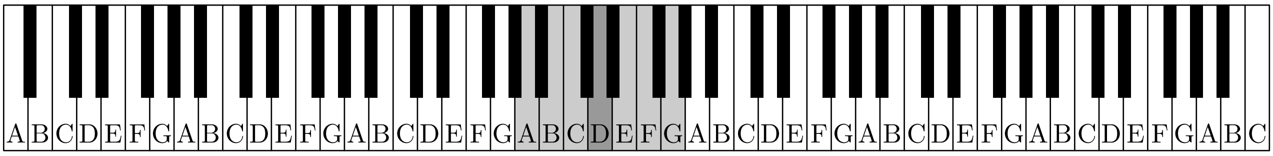
Примеры

стандартный ввод	стандартный вывод	пояснение
1 1 4 1 4 4 1 4	7.06858347	
0 -1 -2 13 -30 9 -28 -5	314.15926536	

Задача G. За границы пианино

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мебибайта

Ваш друг Гассассини Пианини учится играть на пианино. У него дома стоит пианино со стандартной раскладкой в 88 клавиш:



Нас в этой задаче будут интересовать только 52 белые клавиши. Они называются подряд буквами английского алфавита A–G с циклом длины семь (называемым *октавой*), как показано на рисунке.

Гассассини Пианини раздобыл партитуры нескольких музыкальных произведений скрипача Паганини и хочет переложить их на пианино. Для каждого из них он уже расшифровал нотную запись и получил строку из букв A–G, составляющую мелодию этого произведения. Теперь он хочет сыграть эти мелодии на пианино, при этом ему неважно, из какой октавы взята та или иная нота: скажем, если в строке написано, что очередная нота — A, его устроит любая из восьми нот A, имеющихся на пианино.

Проблема в том, что Гассассини Пианини пока неопытен и не может слишком широко расставлять пальцы. Первую ноту он может выбрать произвольно (из имеющих нужную букву); а каждую следующую он сможет сыграть лишь такую, какая отстоит от предыдущей максимум на три клавиши в какую-то сторону. Например, предположим, что последней была сыграна тёмно-серая D на иллюстрации выше; тогда следующей может быть сыграна либо ровно эта же D, либо одна из шести светло-серых нот в её окрестности.

Вам даны строковые расшифровки нескольких музыкальных произведений, интересных Гассассини Пианини; определите, сможет ли он сыграть их, или ему надо подучиться расставлять пальцы.

Формат входных данных

В первой строке находится целое число t — число музыкальных произведений ($1 \leqslant t \leqslant 10^4$).

В каждой из следующих t строк находится непустая строка s_i из заглавных английских букв A–G — расшифровка музыкального произведения. Общее число букв во всех расшифровках не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждой из t строк выведите «`Yes`», если у Гассассини Пианини получится сыграть произведение на пианино, и «`No`», если это невозможно. Выводите каждый из ответов в отдельной строке. Вы можете произвольно выбирать регистр выводимых вами букв; другими словами, проверяющей программе неважно, какие из ваших букв будут прописными, а какие — строчными.

Пример

стандартный ввод	стандартный вывод
5 A ABCDEFGABCDEFGABCDE ADGCFBEADGCFBEADGCF ABEADGCFBEADGCFBEAD CCGGAAGFFEEDDCGGFFEEBGGFFEEB	Yes YES NO no YES

Задача H. Результаты соревнования

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мебибайта

В гонке участвуют n бегунов. У каждого из них есть уникальный номер от 1 до n. Они прибыли к финишной черте в определённом порядке, все в строго разное время. Скажем, что бегун i *неожиданно победил* бегуна j, если i обогнал j и $i < j$.

Для каждого i от 1 до n известно, что бегун i неожиданно победил ровно a_i других бегунов. Ваша задача — восстановить результаты соревнования: номер бегуна, занявшего первое место, номер бегуна, занявшего второе место, $\ldots$, номер бегуна, занявшего n-е место. Можно показать, что ответ всегда уникален — при условии, что он существует.

Формат входных данных

Первая строка входных данных содержит целое число n от 1 до 1000 — число бегунов.

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$, разделённых пробелами, где a_i — это число бегунов, неожиданно побеждённых бегуном i.

Предоставленные данные согласуются с некоторыми возможными результатами соревнования: для всех i верно $a_i \leqslant n - i$. В частности, $a_n = 0$.

Формат выходных данных

Выведите n целых чисел: номера бегунов, занявших первое, второе, $\ldots$, n-е место.

Примеры

стандартный ввод	стандартный вывод
5 3 0 2 1 0	3 1 4 5 2
1 0	1
2 0 0	2 1

Замечание

Проверим, что ответ на первый пример согласуется с данными a_i.

1. Бегун 1 неожиданно победил в точности трёх бегунов 2, 4 и 5.
2. Бегун 2 выступил хуже всех. Следовательно, бегун 2 не мог никого неожиданно победить.
3. Бегун с номером 3 занял первое место и, следовательно, неожиданно победил обоих бегунов с бо́льшими номерами.
4. Бегун с номером 4 неожиданно победил ровно одного другого бегуна: бегуна 5.
5. Нет бегунов с номерами больше 5. Следовательно, бегун 5 не мог никого неожиданно победить.

Задача I. Кто я?

Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мебибайта

Это задача с двойным запуском.

Команда из n игроков сидит за большим круглым столом в библиотеке и играет в кооперативную игру. У каждого из игроков на лбу приклеена карточка с целым числом от 1 до n. Игроки видят карточки друг друга, но не видят свои карточки, числа на которых они пытаются угадать. Поскольку игра происходит в библиотеке, где нужно вести себя тихо, игроки делают свои догадки письменно и одновременно, не общаясь друг с другом. Команда выигрывает, если хотя бы один из игроков угадывает своё число.

Ваша задача заключается в построении выигрышной стратегии для команды. Проверка стратегии будет устроена в два запуска.

При первом запуске вам будут даны числа на карточках всех игроков за столом в порядке следования по часовой стрелке. После этого вам нужно будет выбрать номер игрока, который правильно угадает свою карточку.

При втором запуске вам будут известны номер выбранного игрока, а также числа только на тех карточках, которые видит этот игрок со своего места, в порядке следования слева направо. В качестве ответа при втором запуске вам нужно вывести число на карточке выбранного игрока.

Формат входных данных

При обоих запусках первая строка содержит одно целое число: 0, если это первый запуск, и номер игрока, выбранного на первом запуске, если это второй запуск.

Во второй строке задано число n — число игроков, либо число карточек, которые видны игроку ($n \leqslant 2 \cdot 10^5$).

В третьей строке записано n чисел — числа на карточках.

За столом сидит хотя бы один игрок.

Формат выходных данных

При первом запуске выведите одно целое число: номер игрока, который угадает свою карточку.

При втором запуске также выведите одно число: догадку этого игрока.

Примеры

стандартный ввод	стандартный вывод
0 6 1 2 3 3 3 6	2
2 5 3 3 3 6 1	2

Задача J. Заповедник

Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мебибайта

В волшебном королевстве на плоскости живут единороги. Известны координаты n точек на карте — мест, где их когда-либо видели.

Недавно король согласился объявить территорию обитания единорогов заповедником. Границами заповедника должны стать две параллельные прямые, между которыми (или на которых) находятся все n мест, где видели единорогов. Но земля в королевстве дорогая, поэтому король не хочет отдавать под заповедник лишнего: нужно, чтобы каждая граница проходила хотя бы по одной из n точек.

Вы — главный эколог королевства. Ваша задача — выбрать границы заповедника так, чтобы воля короля была выполнена, но заповедник получился как можно больше. Размером заповедника считается расстояние между границами.

Формат входных данных

В первой строке задано целое число n — число мест, где видели единорогов ($2 \leqslant n \leqslant 1000$).

Каждая из следующих n строк содержит два целых числа x и y — координаты очередной точки на карте, где видели единорогов ($0 \leqslant x, y \leqslant 10^4$). Все заданные точки различны.

Формат выходных данных

Будем задавать каждую прямую тремя коэффициентами: a, b и c. Такая прямая содержит точки (x, y), удовлетворяющие уравнению $ax + by + c = 0$.

Выведите две строки, задающие границы заповедника. В каждой строке выведите три целых числа — коэффициенты уравнения прямой a, b и c ($|a|, |b|, |c| \leqslant 10^9$).

Расстояние между прямыми должно быть максимально возможным. Можно показать, что существует оптимальный ответ, удовлетворяющий ограничениям выше. Если ответов с максимальным расстоянием несколько, выведите любой из них.

Пример

стандартный ввод	стандартный вывод
4 0 0 3 4 8 1 1 1	8 1 0 8 1 -65

Задача K. Операции с гексом

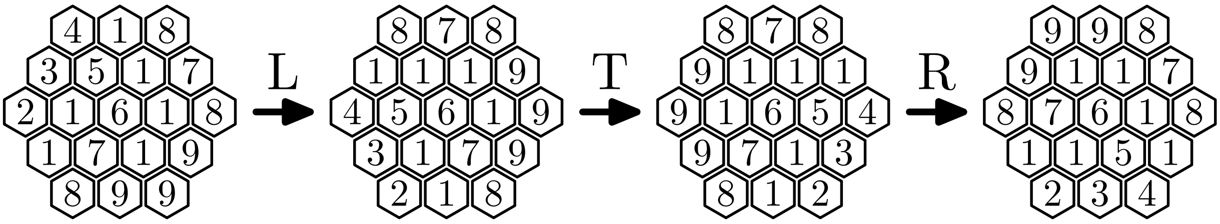
Ограничение по времени: 2 секунды
Ограничение по памяти: 1024 мебибайта

В этой задаче *гекс* — это шестиугольная таблица, состоящая из *клеток* — маленьких правильных шестиугольников. У гекса размера n на каждой стороне ровно n клеток. В каждой клетке записано целое число.

Определим следующие операции с гексом:

- «T»: отразить гекс относительно вертикальной оси симметрии,
- «R»: повернуть гекс на 60 градусов по часовой стрелке,
- «L»: повернуть гекс на 60 градусов против часовой стрелки.

Вот пример гекса размера $n = 3$ и операций с ним:



Даны гекс и последовательность операций. Выполните все операции по порядку и выведите итоговый гекс.

Формат входных данных

В первой строке дано целое число n — размер гекса ($2 \leqslant n \leqslant 500$). В следующих $2n - 1$ строках заданы целые числа от 1 до 99, изначально стоящие в клетках гекса. В каждой строке записано несколько чисел через пробел — столько, сколько клеток в соответствующей строке гекса. Кроме того, для удобства чтения в этих строках могут быть дополнительные пробелы до всех чисел, между соседними числами и после всех чисел.

Последняя строка ввода — последовательность операций. Она имеет длину от 1 до 250 000 символов и состоит из букв «T», «R» и «L».

Общий размер входных данных, включая пробелы и переводы строк, не превосходит 2^{23} байт.

Формат выходных данных

Выведите гекс после выполнения всех операций: $2n - 1$ строк, в которых записаны целые числа. В каждой строке должно быть записано несколько чисел через пробел — столько, сколько клеток в соответствующей строке гекса. Кроме того, для удобства чтения в этих строках могут быть дополнительные пробелы до всех чисел, между соседними числами и после всех чисел.

Общий размер вывода, включая пробелы и переводы строк, должен быть не больше 2^{23} байт.

Пример

стандартный ввод	стандартный вывод

3
  4 1 8
 3 5 1 7
2 1 6 1 8
 1 7 1 9
  8 9 9
LTR
	9 9 8
	 9 1 1 7
	8 7 6 1 8
	 1 1 5 1
	  2 3 4

Задача L. Гипотеза Коллатца и случайные увеличения

Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мебибайта

Это интерактивная задача.

Определим функцию Коллатца $\text{collatz}(x)$, действующую на целых числах, так: если x чётное, то $\text{collatz}(x) = \frac{x}{2}$, а иначе $\text{collatz}(x) = 3x + 1$. Знаменитая гипотеза Коллатца гласит, что, если начать с любого целого положительного x_0 и построить последовательность $x_1 = \text{collatz}(x_0)$, $\ldots$, $x_{i+1} = \text{collatz}(x_i)$, то в этой последовательности обязательно встретится единица.

Невероятная сложность, из-за которой гипотеза до сих пор ни доказана, ни опровергнута, заключается в очень хаотичном поведении последовательности $\{x_i\}$ до момента, когда она достигает единицы. Даже для очень небольших чисел единица может достигаться довольно поздно: так, если стартовать с $x_0 = 9$, получим

$$9 \to 28 \to 14 \to 7 \to 22 \to 11 \to 34 \to 17 \to 52 \to 26 \to 13 \to 40 \to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1,$$

а если начать с $x_0 = 27$, то получим единицу лишь после 111 применений $\text{collatz}(x)$!

Вы сидите перед автоматом, у которого есть экран и две кнопки: красная и синяя. На экране отображается целое положительное число (гарантируется, что это случайное число от 2 до 10^7 включительно), и его вам надо превратить в единицу. Красная кнопка называется `collatz`, она заменяет число x на экране на $\text{collatz}(x)$. Синяя кнопка называется `random`, она заменяет x на случайное целое число от $3x + 1$ до $6x$ включительно. Нажатия на кнопки не бесплатны: после каждого нажатия кнопки, когда на экране отображается число x_{i+1}, требуется внести в автомат столько жетонов, сколько цифр в десятичной записи числа x_{i+1}. Например, в вышеприведённом процессе, стартующем с девятки, нужно заплатить за все цифры чисел $28, 14, 7, \ldots, 2, 1$; на это уйдёт 32 жетона.

Если нажимать только на красную кнопку, вы достигнете единицы, потратив в среднем 707 жетонов. Несмотря на то, что синяя кнопка всегда увеличивает число на экране (причём значительно), можно ускорить процесс получения единицы, если в правильные моменты нажимать синюю кнопку! Ваша задача — написать программу, которая тратит на одно число в среднем не более 600 жетонов. Чтобы проверка меньше зависела от случайности, мы в каждом тесте дадим программе $t \leqslant 50$ случайных стартовых чисел, и вам надо будет превратить их все в единицы за $600 \cdot 50 = 30\,000$ жетонов.

Протокол взаимодействия

Для начала прочитайте строку, содержащую целое число t — число стартовых чисел ($1 \leqslant t \leqslant 50$). У жюри есть t стартовых целых чисел x_0, выбранных случайно, независимо и равномерно распределённых на отрезке $\left[2; 10^7\right]$, и вам требуется по очереди превратить их все в единицу.

В начале i-го раунда прочитайте строку, содержащую одно число x_0 — очередное стартовое число ($2 \leqslant x_0 \leqslant 10^7$). Далее вам требуется преобразовывать число. После очередного преобразования, если на экране число x_j, выведите в отдельной строке либо слово «`collatz`» (буквы могут быть в любом регистре), если вы хотите заменить число на $x_{j+1} = \text{collatz}(x_j)$, либо слово «`random`» (буквы могут быть в любом регистре), если вы хотите заменить число на x_{j+1}, равное случайному целому числу, равномерно распределённому на отрезке $[3x_j + 1; 6x_j]$. После вывода каждой строки не забывайте очищать буфер вывода — иначе, скорее всего, вы получите ошибку `Idleness Limit Exceeded`:

- `std::cout.flush()` в C++;
- `stdout.flush()` в Python;
- `System.out.flush()` в Java.

После этого прочитайте строку, содержащую одно целое число x. Возможны следующие варианты:

- если $x = 0$, то у вас кончились жетоны;
- если $x = 1$, то $x_{j+1} = 1$ — вы успешно завершили раунд, переходите к следующему раунду (для чего начните со считывания следующего стартового числа x_0);
- если $x > 1$, то $x_{j+1} = x$ — число на экране изменилось после нажатия кнопки, продолжайте выводить строки «`collatz`» и «`random`». Обратите внимание: в процессе взаимодействия число x может превысить и 2^{64}, и 2^{128}.

Также жюри передаст в вашу программу число 0, если вы нарушите формат вывода и напечатаете любую строку, кроме «`collatz`» и «`random`». Прочитав 0, ваша программа должна сразу завершить работу, чтобы получить вердикт `Wrong Answer`. В противном случае вердикт может быть каким угодно (кроме `Accepted`).

Считав в t-й раз единицу, завершите работу программы, чтобы получить вердикт `Accepted`.

Пример

стандартный ввод	стандартный вывод
2	
4	
	Collatz
2	
	cOLLaTZ
1	
3	
	RANDOM
16	
	collatz
8	
	COLlatz
4	
	cOLLATz
2	
	CoLlAtZ
1	

Замечание

Пример — единственный тест, где стартовые числа x_0 выбраны не случайно, а вручную.

Задача M. Суммы двух

Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мебибайта

Фокусница Лина утверждает, что обычный современный компьютер легко может производить сотню миллиардов операций в секунду! Чтобы доказать это, она предлагает проделать следующие вычисления.

Пусть V — множество целых чисел, изначально пустое. Дано начальное значение числа s. Сделаем n шагов следующего вида:

- $s \leftarrow (s \cdot 618\,023 + 1) \bmod 999\,983$;
- найдём число различных пар чисел из V, которые в сумме дают s;
- если это число пар чётно, добавим число s в множество V.

Сколько элементов будет в множестве V после n шагов?

Формально: на каждом шаге мы считаем число пар (a, b), в которых $a \in V$, $b \in V$, $a \leqslant b$ и $a + b = s$.

Формат входных данных

В первой строке заданы целые числа n и s ($1 \leqslant n \leqslant 200\,000$; $0 \leqslant s < 999\,983$; $s \neq 742\,681$).

Формат выходных данных

Выведите одно целое число: размер множества V после n шагов.

Пример

стандартный ввод	стандартный вывод
4 179629	3

Замечание

В примере значения s на четырёх шагах равны $740\,740$, $139\,655$, $469\,353$ и $880\,395$.

Задача N. Разыскивается: второй носок

Ограничение по времени:	1 секунда
Ограничение по памяти:	1024 мебибайта

Никита собирается на соревнование. Самое сложное в этом деле — найти парные носки. В его ящике p различных пар носков, а также m одиночных носков, чьи пары Никита давно потерял. Никита берёт носки один за другим, пока среди вытащенных им носков не найдутся два парных. Чему равно математическое ожидание числа вытащенных Никитой носков?

Посчитайте это матожидание по модулю простого числа $10^9 + 7$. А именно, если ответ равен рациональному числу r/q, то необходимо вывести такое число a, для которого $0 \leqslant a < 10^9 + 7$ и $a \cdot q = r \pmod{10^9 + 7}$.

Формат входных данных

В первой строке записано целое число t — число наборов входных данных ($1 \leqslant t \leqslant 10^4$). Далее описаны сами наборы.

Каждый набор входных данных задан в отдельной строке, которая содержит два целых числа p и m — число пар носков и число одиночных носков в ящике ($1 \leqslant p \leqslant 10^6$; $0 \leqslant m \leqslant 10^6$).

Формат выходных данных

Для каждого набора входных данных выведите строку, содержащую одно целое число: ответ на задачу.

Пример

стандартный ввод	стандартный вывод
1 1 1	666666674

Замечание

В первом наборе входных данных математическое ожидание равно 8/3, что по модулю $10^9 + 7$ равняется 666 666 674.