

Задача А. Большой круглый стол

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Сегодня Маша принимает гостей. Включая Машу, в празднике принимает участие n человек, которые расселись по кругу за большим круглым столом.

Разумеется, Маша хочет пообщаться со многими гостями, но кричать через весь стол неудобно. Тогда она быстро придумала решение проблемы: иногда она просит соседа слева или справа от неё поменяться с ней местами. Гости, разумеется, любезно соглашаются на её просьбу.

Проводив гостей, Маша вспомнила, что забыла телефон на месте, на котором она сидела в конце мероприятия. Маша не помнит, на каком месте она сидела в конце, зато помнит, на каком месте она сидела в начале, а также помнит, что она ровно k раз менялась местами с одним из соседей. Теперь она хочет узнать количество мест, на которых она могла оказаться в конце вечера.

Формат входных данных

Входные данные содержат два натуральных числа n и k — количество мест за столом и число раз, которое Маша менялась местами с одним из своих соседей ($3 \leq n \leq 10^9$, $0 \leq k \leq 10^9$).

Формат выходных данных

Выведите одно число — количество мест, на которых Маша могла оказаться в конце мероприятия.

Примеры

стандартный ввод	стандартный вывод
5 2	3
3 3	3

Замечание

В первом примере, два раза поменявшись местами с одним из своих соседей, Маша могла оказаться либо на исходном месте, либо на одном из двух мест, находящихся через одно от исходного.

Во втором примере Маша могла оказаться на любом месте.

Задача В. Помеха справа

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Каждое утро жители столицы Берляндии вынуждены стоять в ужасных пробках по дороге на работу. Особенно сильно эти пробки заметны на центральной площади столицы, которая представляет собой перекресток, да еще и нерегулируемый — берляндцы стремятся сохранить нетронутым исторический облик центра города.

Решив заняться исследованием ситуации, мэр столицы поручил изучить, как именно скапливаются пробки. Ведь на перекрестке запрещены повороты, таким образом, машины могут проезжать перекресток только прямо. Установив специальные датчики, специалисты выяснили, что каждое утро перекресток пытаются проехать n машин. К перекрестку подходят улицы с четырех сторон: если посмотреть на карту, то эти улицы идут в направлении вверх «U», влево «L», вниз «D» и вправо «R». На каждой из улиц в процессе проезда перекрестка может скапливаться очередь из машин.

Каждый водитель при подъезде к перекрестку действует следующим образом: i -й водитель подъезжает к перекрестку в начале t_i -й секунды, встает в конец очереди на этой улице и анализирует ситуацию.

Если в момент анализа ситуации перед водителем в очереди есть хотя бы одна другая машина, он продолжает ждать и в следующий раз анализирует ситуацию в начале следующей секунды. Если машин перед ним в очереди нет, он пытается проехать перекресток. Если у водителя нет *помехи справа*, то он покидает очередь, проезжает перекресток за эту секунду и уезжает из трудного места. Иначе он остается в очереди и анализирует ситуацию еще раз в начале следующей секунды. Водители анализируют ситуацию одновременно, и лишь затем первый в очереди водитель может начать движение, поэтому каждую секунду в каждом направлении перекресток может проехать не более одной машины.

У водителя перед перекрестком есть помеха справа, если на перпендикулярном направлении справа в очереди есть хотя бы одна машина. Таким образом, водителям, пытающимся проехать перекресток в направлении вверх, мешают машины, стоящие в очереди в направлении влево, направлению влево мешают машины из очереди в направлении вниз, и так далее. Заметим, что если все четыре очереди непусты, то у каждого водителя есть помеха справа, и они уже никогда не проедут перекресток.

По данным о времени, когда водители подъезжали к перекрестку, помогите выяснить, когда каждый из них проедет перекресток. Обратите внимание, что некоторые водители могут так и не проехать перекресток, оставшись в очереди перед ним.

Формат входных данных

Первая строка содержит целое число n — количество машин, подъезжающих к перекрестку ($1 \leq n \leq 10^5$).

В каждой из следующих n строк содержится число t_i и символ d_i — номер секунды, в начале которой i -я машина подъезжает к перекрестку, и направление, в котором она пытается его проехать ($0 \leq t_i \leq 10^9$, d_i равно «U», если машина едет вверх по карте, «L», если машина едет влево, «D», если вниз, и «R», если вправо). Машины во вводе заданы в порядке неубывания t_i .

Гарантируется, что в каждый момент времени с каждой стороны подъезжает не более одной новой машины.

Формат выходных данных

Для каждой машины в порядке их описания во вводе выведите в отдельной строке номер секунды, когда она проедет перекресток. Если машина останется стоять на перекрестке, выведите в соответствующей строке число -1 .

Примеры

стандартный ввод	стандартный вывод
4 0 R 0 U 0 L 5 D	2 1 0 5
7 0 U 0 D 1 L 1 D 2 D 2 R 2 U	0 0 -1 1 -1 -1 -1

Задача С. Древнее заклинание

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Гидеон нашел древний свиток, позволяющий активировать мощнейшее заклинание, которое помогает решить любую задачу. Осталось провести ритуал.

На свитке изображена прямоугольная таблица, содержащая n строк и m столбцов, в каждой ячейке которой написана латинская буква. Заклинание представляет собой строку, также состоящую из латинских букв. Чтобы активировать заклинание, необходимо бесконечное число раз произнести эту строку, перемещая при этом конец волшебной палочки по таблице на свитке вдоль подходящей циклической последовательности ячеек.

Назовем циклической последовательностью ячеек такую последовательность ячеек $(x_1, y_1), (x_2, y_2) \dots (x_k, y_k)$, что каждая пара соседних в последовательности ячеек, а также первая и последняя ячейки в последовательности, не совпадают и имеют общую сторону. Таким образом, при перемещении конца волшебной палочки вдоль последовательности, каждый раз необходимо переместиться в соседнюю ячейку вверх, влево, вправо или вниз, а в конце — вернуться в начальную ячейку. При этом одна и та же ячейка может встречаться в последовательности несколько раз.

Для активации заклинания требуется найти такую циклическую последовательность ячеек, что если одновременно делать следующее: произносить бесконечное число раз строку заклинания, по одному символу в секунду, и, начав с первой ячейки, каждую секунду до бесконечности перемещать конец волшебной палочки к следующей ячейке в последовательности (после последней ячейки следует снова перейти к первой ячейке), то буквы заклинания и буквы в ячейках таблицы всегда будут совпадать.

Помогите Гидеону найти подходящую циклическую последовательность, либо выясните, что такой последовательности не существует.

Формат входных данных

Первая строка содержит целые числа n , m и l — высоту и ширину таблицы, а также длину заклинания, соответственно ($2 \leq n, m \leq 200$, $1 \leq l \leq 200$).

В следующих n строках содержится по m строчных латинских букв — содержимое таблицы.

В последней строке содержится строка из l строчных латинских букв — заклинание.

Формат выходных данных

Если искомой последовательности не существует, в единственной строке выведите «NO».

Иначе, в первой строке выведите «YES». Во второй строке выведите k — длину последовательности, k не должно превышать 10^7 . Гарантируется, что если ответ существует, то существует ответ с k , не превышающим 10^7 .

В третьей строке выведите координаты первой ячейки последовательности: сначала номер строки, а потом номер столбца, на пересечении которых находится эта ячейка. Наконец, в последней строке выведите строку длины k , состоящую из символов «U», «L», «D», «R», которая описывает последовательность перемещения конца волшебной палочки по ячейкам в искомой циклической последовательности последовательности.

Символ «U» соответствует переходу из текущей ячейки в соседнюю сверху, символ «L» — в соседнюю слева, «D» — в соседнюю снизу, а «R» — в соседнюю справа.

После переходов по всем символам необходимо вернуться в исходную ячейку. Выходить за границы поля не разрешается.

Если ответов несколько, выведите любой.

Примеры

стандартный ввод	стандартный вывод
3 2 4 aa bb cc abcb	YES 4 1 1 DDUU
3 3 5 abc dee ghi eeeeee	YES 2 2 2 RL
3 3 5 abc def ghi eeeeee	NO
3 6 16 abegme ckwass ichrdq egassemssdrawkcab	YES 16 1 3 RDRRULDDLULLLURR

Замечание

Пояснение к третьему тесту. По определению, последовательность из одной ячейки не является циклической, поэтому решения не существует.

Задача D. Эскалатор

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В Санкт-Петербурге открывают новую станцию метро, и для нее требуется произвести эскалатор. Эскалатор состоит из n ступенек, пронумерованных целыми числами от 1 до n . Традиционно на ступеньках с номерами, кратными десяти, а также на первой и последней ступеньке, пишут их номера. При записи номера на каждую записанную цифру уходит одно и то же количество краски.

Чтобы рассчитать необходимое количество краски, требуется узнать, сколько цифр будет написано. Напишите программу, которая определяет, сколько всего цифр будет использовано в номерах подписанных ступенек.

Формат входных данных

Во входном файле задано одно целое число n — количество ступеней эскалатора ($1 \leq n \leq 10^{12}$).

Формат выходных данных

Выведите суммарное количество цифр в номерах подписанных ступенек.

Примеры

стандартный ввод	стандартный вывод
20	5
23	7

Замечание

В первом примере номера будут написаны на ступеньках 1, 10, 20; во втором — 1, 10, 20, 23.

Задача Е. Сериал

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Джон — фанат сериала «Во все престолы». Совсем скоро выходит новый сезон, и Джон хочет его посмотреть.

Сериалы будут выходить по одной в день. Джону не хочется скачивать их каждый раз вручную, поэтому он собирается написать программу, которая будет делать это за него. Каждая серия — это отдельный файл, размер файла, содержащего i -ю серию, равен s_i байт.

Программа Джона будет действовать следующим образом. Она один за другим будет отправлять на сервер запросы, где j -й запрос представляет собой «загрузить очередные x_j байт». В ответ на такой запрос сервер возвращает пакет данных, содержащий очередные x_j байт файла, а также заголовок, содержащий k байт различной служебной информации. Таким образом, размер пакета равен $x_j + k$ байт, при этом значение k одно и то же для всех запросов.

Когда в результате некоторого запроса скачивается последний байт файла, программа завершает свою работу и не делает дальнейших запросов к серверу. Однако протокол устроен таким образом, что размер пакета равен $x_j + k$, даже если был достигнут конец файла и в действительности было загружено меньше x_j байт полезной информации.

Джон ничего не знает в программировании, поэтому он может написать только простую программу, которая для загрузки каждой серии будет обращаться к серверу с одинаковой последовательностью запросов. Интернет у него медленный, поэтому он хочет, чтобы суммарный размер всех загруженных пакетов был как можно меньше.

Из-за утечки информации от создателей сериала Джону известен размер каждой серии. Помогите ему узнать, какой минимальный суммарный размер пакетов придётся загрузить, чтобы скачать все серии.

Формат входных данных

В первой строке входного файла заданы целые числа n и k — количество серий и размер заголовка пакета ($1 \leq n \leq 10000$; $0 \leq k \leq 10^9$).

Во второй строке задано n целых чисел s_i — размеры серий ($1 \leq s_i \leq 10^9$).

Формат выходных данных

Выведите одно число — минимальный суммарный размер пакетов, которые придётся загрузить.

Примеры

стандартный ввод	стандартный вывод
4 1000 100 200 200 800	6400
4 0 100 200 800 200	1300

Замечание

В первом примере можно сначала загрузить 200 байт, а затем 600. Тогда первые три серии будут скачаны после первого запроса и на каждую из них будет потрачено по $200 + 1000 = 1200$ байт. Последняя серия будет скачана за два запроса, на нее будет потрачено $(200 + 1000) + (600 + 1000) = 2800$ байт. Итого $1200 + 1200 + 1200 + 2800 = 6400$ байт.

Во втором примере заголовка нет, поэтому можно не бояться делать много запросов. Например, можно загружать блоками по 100 байт.

Задача F. Парадокс с дробями

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Никита очень любит математические парадоксы. Недавно он заметил, что

$$\frac{2}{3} < \frac{1}{1}; \quad \frac{1}{2} < \frac{6}{11},$$

но при этом если у меньших дробей сложить числители и знаменатели и то же сделать с большими дробями, то получатся дроби

$$\frac{2+1}{3+2} = \frac{3}{5} \quad \text{и} \quad \frac{1+6}{1+11} = \frac{7}{12},$$

причем

$$\frac{3}{5} > \frac{7}{12}.$$

Тогда Никита выписал в ряд k дробей и хочет выбрать среди них четыре дроби, чтобы выполнялись неравенства

$$\frac{m_1}{n_1} \leq \frac{m_2}{n_2}; \quad \frac{m_3}{n_3} \leq \frac{m_4}{n_4},$$

а величина

$$\frac{m_1 + m_3}{n_1 + n_3} - \frac{m_2 + m_4}{n_2 + n_4}$$

была максимальной. Каждую из записанных дробей можно взять только в качестве одной из выбранных четырех. Помогите Никите решить эту сложную задачу.

Формат входных данных

Первая строка ввода содержит число k — количество дробей, выписанных Никитой ($4 \leq k \leq 2000$).

Следующие k строк содержат по два положительных целых числа: для каждой дроби задан ее числитель и знаменатель. Все заданные дроби являются несократимыми. Числитель и знаменатель каждой дроби не превышают 10 000.

Формат выходных данных

Выведите четыре различных целых числа: номера дробей, которые следует выбрать в качестве $\frac{m_1}{n_1}$, $\frac{m_2}{n_2}$, $\frac{m_3}{n_3}$ и $\frac{m_4}{n_4}$, соответственно. Дроби пронумерованы от 1 до n в том порядке, в котором они заданы во вводе. Если возможных оптимальных решений несколько, разрешается выдать любое из них.

Пример

стандартный ввод	стандартный вывод
4 1 1 1 2 2 3 6 11	2 4 3 1

Задача G. Головоломка

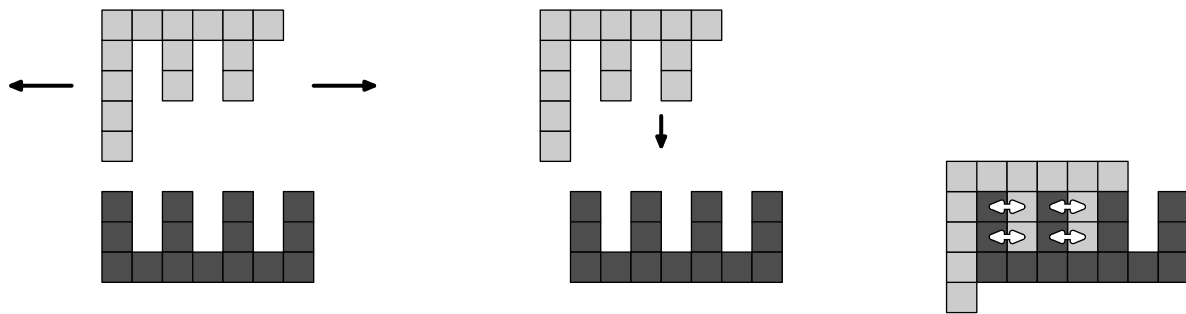
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Дима играет в логическую компьютерную игру. Очередной уровень устроен следующим образом: у Димы есть клетчатая фигура, составленная из единичных квадратов. Ее необходимо правильным образом уронить на препятствие, также составленное из единичных квадратов. Фигура и препятствие являются связными: от любого квадрата фигуры можно добраться до любого другого, переходя между квадратами по стороне, аналогичное свойство выполнено для препятствия.

Исходно фигура находится сверху над препятствием так, что любой квадрат фигуры выше любого квадрата препятствия. Дима может произвольным образом переместить фигуру по горизонтали, а затем нажать на специальную кнопку, и фигура опустится вниз до упора. После этого за каждую пару соседних по горизонтали единичных квадратов, таких что левый квадрат принадлежит препятствию, а правый — фигуре, Дима получает одно очко. Если Дима расположит фигуру так, что при падении вниз она не упрётся в препятствие, то Дима не получит очков.

Помогите Диме решить головоломку оптимальным образом.

Например, если у Димы есть фигура и препятствие, показанные на рисунке ниже, выгоднее всего уронить фигуру следующим образом:



В этом случае количество очков, набранных Димой, будет равно 4.

Формат входных данных

В первой строке ввода содержатся два числа n_f и m_f — размеры фигуры ($1 \leq n_f, m_f \leq 300$). В каждой из следующих n_f строк содержатся по m_f символов — описание фигуры, «*» означает квадрат, принадлежащий фигуре, «.» — пустой квадрат.

В следующей строке содержатся два числа n_o и m_o — размеры препятствия ($1 \leq n_o, m_o \leq 300$). В каждой из следующих n_o строк содержатся по m_o символов — описание препятствия, «#» означает квадрат, принадлежащий препятствию, «.» — пустой квадрат.

Гарантируется, что фигура и препятствие представляют собой непустые связные множества единичных квадратов.

Формат выходных данных

Выведите максимальное количество очков, которое Дима может получить.

Пример

стандартный ввод	стандартный вывод
5 6 ***** *.*.*. *.*.*. *..... *..... 3 8 #.#.#.#. #.#.#.#. #####.	4

Замечание

В тестовом примере оптимальный способ уронить фигуру показан на рисунке в условии.

Задача Н. «Кто хочет стать миллионером?»

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В Бейтландии решили провести шоу «Кто хочет стать миллионером?». Участнику шоу по очереди задаются n вопросов, если он ответил на i -й вопрос, его приз становится равным a_i . После любого вопроса участник шоу может забрать свой приз и покинуть шоу.

Организаторы шоу решили, что число вопросов будет равно n , но не могут определиться с призами. Первый вопрос обычно очень простой и за него решено было установить приз равный $a_1 = 100$ битов. Каждый следующий вопрос сложнее, поэтому очередной приз должен быть хотя бы вдвое больше предыдущего. Наконец, призы должны быть достаточно круглыми.

Организаторы называют сумму достаточно круглой, если нули в конце этой суммы составляют хотя бы половину цифр в записи этой суммы. Они решили, что в качестве приза a_i для всех $i > 1$ они выберут минимальное достаточно круглое число, хотя бы в 2 раза большее a_{i-1} . Помогите организаторам понять, чему будут равны призы.

Формат входных данных

На ввод подается одно число n ($1 \leq n \leq 25$).

Формат выходных данных

Выведите n чисел по одному на строке — призы, которые будут установлены организаторами шоу.

Пример

стандартный ввод	стандартный вывод
10	100 200 400 800 1600 3200 6400 13000 26000 52000

Задача I. Операция «Перестановка»

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Больше всего Генерал Петров любит строевую подготовку. Однажды в построении участвовало n солдат, для простоты пронумерованных от 1 до n . Солдаты выстроились в ряд, при этом на месте i стоял солдат с номером a_i . После этого адъютант генерала прошёл вдоль строя и проверил, что перестановка в точности совпадает с придуманной генералом.

Некоторое время спустя генерал захотел снова расставить солдат в том же порядке. Однако он забыл, как именно он расставил солдат в тот раз. К счастью, у адъютанта генерала хорошая память — про m пар позиций x_i, y_i он помнит, что номер солдата, стоявшего на месте x_i , меньше номера солдата, стоявшего на месте y_i .

Адъютант стал по очереди сообщать генералу пары x_i, y_i . Но генерал хочет побыстрее начать строить солдат. Помогите ему определить такое минимальное k , что как только адъютант сообщит ему первые k пар позиций, генерал сможет однозначно определить искомую перестановку.

Формат входных данных

В первой строке находятся два целых числа n и m — количество солдат, участвовавших в операции, и количество пар позиций, которые запомнил адъютант ($2 \leq n \leq 10^5$; $1 \leq m \leq 10^5$).

В следующих m строках даны пары, которые помнит адъютант, в том порядке, в котором сообщает их генералу. Каждая строка содержит по два числа x_i и y_i , которые означают, что номер солдата на позиции x_i был меньше номера солдата на позиции y_i ($1 \leq x_i, y_i \leq n$; $x_i \neq y_i$).

Гарантируется, что каждая пара x_i, y_i встречается во входном файле не более одного раза. Гарантируется, что входные данные корректные, то есть существует хотя бы одна перестановка, для которой выполняются все условия, которые помнит адъютант.

Формат выходных данных

Выведите одно число k — минимальный номер пары позиций, после произнесения которой можно однозначно восстановить перестановку. Если входные данные таковы, что перестановку однозначно восстановить не удастся, выведите -1 .

Примеры

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

Замечание

В первом примере солдаты стояли в порядке (5, 2, 1, 3, 4). Уже после четвёртой пары чисел, запомненной адъютантом, можно восстановить этот порядок.

Во втором примере существует четыре варианта расстановки солдат, удовлетворяющих входным данным: (1, 3, 4, 2), (2, 3, 4, 1), (3, 2, 4, 1) и (4, 2, 3, 1).

Задача J. Трудности переписки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Этим летом Джек ездил в летнюю школу в России. Там он завел много новых друзей, а также встретил красивую девушку. По возвращении домой родители подарили Джеку новый ноутбук, и теперь он всегда может быть на связи со своими новыми друзьями. Естественно, получив подарок, Джек сразу стал переписываться со своей подругой Ирой.

Отправив несколько сообщений, Джек заметил, что ноутбук, а точнее его клавиатура, работает не так, как он ожидал. В процессе ввода сообщения у ноутбука иногда внезапно срабатывает клавиша «Home», в результате чего курсор ввода перемещается в начало строки. Так, например, если у Джека в процессе ввода строки «irailikeyou» клавиша «Home» сработала после ввода букв «a» и «y», то получится строка «ouilikeyira».

Джек планировал набрать строку s , нажимая по очереди на соответствующие клавиши. Закончив набор, он посмотрел на экран и увидел строку t . Теперь он хочет понять, может ли она быть результатом его ввода, если единственная неисправность его ноутбука — лишние срабатывания клавиши «Home», либо у его ноутбука есть еще проблемы. Помогите Джеку.

Формат входных данных

В первой строке задано число n — длина строк s и t ($1 \leq n \leq 5000$).

Во второй строке задана последовательность маленьких латинских букв длины n — строка s .

В третьей строке задана последовательность маленьких латинских букв длины n — строка t .

Формат выходных данных

Выведите «Yes», если из строки s могла получиться строка t , иначе выведите «No».

Примеры

стандартный ввод	стандартный вывод
11 irailikeyou ouilikeyira	Yes
3 abc bac	No

Замечание

При наборе строки «abc» могут получиться следующие строки: «abc», «bca», «cab», «cba».

Задача К. Таня, мячи и «исключающее или»

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

У Тани было n мячей, она пронумеровала их от 1 до n . Но, к сожалению, Таня уронила все мячи в реку и сильно расстроилась.

Чтобы утешить ее, старший брат Сережа предложил Тане забавное математическое развлечение: посчитать сумму попарных *исключающих или* от номеров ее мячей.

Исключающее или двух чисел обозначается как \oplus и соответствует операции «xor» в паскале или «^» в других языках. Чтобы вычислить $x \oplus y$ для двух целых чисел, необходимо сделать следующее: представить каждое из чисел в двоичной системе счисления и сделать i -й разряд результата единицей, если он равен единице ровно в одном из чисел x и y . Например, $3 \oplus 2 = 11_2 \oplus 10_2 = 1_2 = 1$, $17 \oplus 5 = 10001_2 \oplus 101_2 = 10100_2 = 20$.

Помогите Тане! Посчитайте сумму по всем парам ее мячей исключающих или их номеров. Таня не любит большие числа, так что ответ необходимо вывести по модулю $10^9 + 7$.

Например, если у Тани было 3 мяча, то искомое значение равно $(1 \oplus 2) + (1 \oplus 3) + (2 \oplus 3) = 3 + 2 + 1 = 6$.

Формат входных данных

В первой строке задано число n — количество мячей у Тани ($1 \leq n \leq 10^9$).

Формат выходных данных

Выведите сумму по всем парам ее мячей исключающих или их номеров, взятую по модулю $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
3	6