

Задача А. Участники олимпиад

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В классе учатся n школьников.

Некоторые из них участвовали в олимпиадах:

- a школьников участвовали в олимпиаде по математике,
- b школьников — по информатике,
- c школьников — по русскому языку.

Некоторые ребята могли участвовать сразу в нескольких олимпиадах.

Помогите учителю оценить, сколько школьников могло не поучаствовать ни в одной из этих олимпиад. Найдите минимальное и максимальное возможное число учеников, которые не участвовали ни в одной олимпиаде.

Формат входных данных

В первой строке входных данных дано одно число n — число учеников в классе ($1 \leq n \leq 2 \cdot 10^5$).

Во второй, третьей и четвертой строке даны числа a , b и c соответственно ($0 \leq a, b, c \leq n$).

Формат выходных данных

В первой строке выведите минимальное возможное число школьников, которые не участвовали ни в какой олимпиаде. Во второй строке выведите максимальное возможное число таких школьников.

Примеры

стандартный ввод	стандартный вывод
10 7 5 3	0 3
7 7 7 7	0 0

Задача В. Подходящий конверт

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Семён Игоревич — руководитель кружка по олимпиадной информатике. Школьники из его кружка поехали на выездную командную олимпиаду на море в известный образовательный центр «Кассиопея». К сожалению, Семён Игоревич не смог поехать на олимпиаду, и его ученики решили передать ему привет с моря, отправив красивые бумажные открытки.

Каждый из n школьников купил открытку, каждая открытка — прямоугольник с высотой h_i и шириной w_i . Для оптимизации отправки открыток школьники решили купить один большой конверт и положить в него все открытки одной стопкой. Конверт представляет собой прямоугольник с высотой H и шириной W . Открытка кладется в конверте таким образом, чтобы стороны открытки были параллельны сторонам конверта. Открытку можно поворачивать на 90° . Открытка помещается в конверт, если стороны открытки не больше соответствующих параллельных им сторон конверта.

Школьники хотят, чтобы купленный конверт имел минимальную площадь $H \cdot W$.

Определите высоту и ширину конверта, чтобы в него можно было положить все открытки одной стопкой и площадь его была минимальной.

Формат входных данных

В первой строке входных данных дано целое число n , $1 \leq n \leq 10^5$ — число школьников. В следующих n строках дана высота и ширина каждой открытки h_i и w_i , ($1 \leq h_i, w_i \leq 10^9$).

Формат выходных данных

Выведите два целых числа H и W — высоту и ширину подходящего конверта. Если стороны не равны, то выведите сначала меньшую из сторон.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3 1 4 2	2 4
2 1 1 2 2	2 2

Задача С. Объединение амулетов

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

У Хорнет появилось очень много амулетов, а именно целых n штук! Каждый амулет уникален и обладает особой силой. Обозначим силу i -го амулета за a_i .

Так как в большом числе амулетов очень легко запутаться, она может один раз объединить непустой подотрезок амулетов в один, более мощный. Пусть она выбрала отрезок $[i, j]$. Тогда сила нового амулета будет равна $\text{lcm}(a_i, a_{i+1}, \dots, a_j)$, где lcm — наименьшее общее кратное чисел $a_i, a_{i+1}, \dots, a_j$. Таким образом, после объединения амулетов с i по j у нее будут амулеты с силами $a_1, \dots, a_{i-1}, a_{j+1}, \dots, a_n, k$, где за k обозначим силу получившегося амулета.

Так как Хорнет важна итоговая боевая сила, после объединения амулетов она хочет её пересчитать. Боевая сила набора амулетов определяется как НОД всех амулетов в наборе, где НОД или gcd обозначает наибольший общий делитель.

Хорнет интересуется сумма значений боевой силы по всем возможным способам объединить амулеты на отрезке в один более мощный амулет. Формально, обозначим за $f(i, j)$ боевую силу амулетов Хорнет, которая получится, если объединить амулеты с i -го по j -й включительно. Она будет равна $\text{gcd}(a_1, \dots, a_{i-1}, a_{j+1}, \dots, a_n, \text{lcm}(a_i, \dots, a_j))$. Тогда Хорнет хочет посчитать $\sum_{i=1}^n \sum_{j=i}^n f(i, j)$.

Помогите ей посчитать данное значение.

Формат входных данных

В первой строке дано целое число n — изначальное количество амулетов ($1 \leq n \leq 2 \cdot 10^5$).

Во второй строке даны n целых чисел a_i — изначальные силы амулетов, которые есть у Хорнет ($1 \leq a_i \leq 10^7$).

Формат выходных данных

Выведите единственное число — ответ на задачу. Так как ответ может быть большим, выведите его по модулю 998 244 353.

Примеры

стандартный ввод	стандартный вывод
5 2 6 9 3 6	44
6 1 2 3 4 5 6	85

Задача D. Постройка сцены

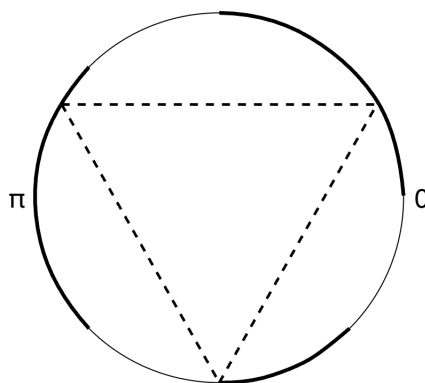
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Для награждения победителей олимпиады решено построить новую сцену. Награждение будет проходить в цирке, поэтому сцена должна представлять собой треугольник, который лежит внутри арены — круга радиуса r . Основаниями сцены являются вершины данного треугольника.

На окружности — границе круга — выделены три непересекающиеся дуги. Было решено, что на каждой дуге должно лежать ровно одно основание сцены.

Организаторы хотят сделать сцену как можно большей площади, вы должны помочь им и сказать, какую максимальную площадь можно получить.

Ниже приведен пример сцены для тестов из примера:



На рисунке жирными линиями показаны заданные дуги, пунктирными — оптимальная сцена.

Формат входных данных

В первой строке дано одно вещественное число r — радиус круга ($1 \leq r \leq 100$).

Вторая строка описывает дуги. Зафиксируем произвольный радиус-вектор заданного круга, будем считать его направление нулевым. Каждая дуга описывается двумя вещественными числами: углами в радианах, на которые нужно повернуть заданный радиус-вектор против часовой стрелки, чтобы его конец указывал на концы дуги.

Таким образом задано шесть вещественных чисел $a_1, a_2, b_1, b_2, c_1, c_2$ — углы, задающие концы трех заданных дуг ($0 \leq a_1 < a_2 < b_1 < b_2 < c_1 < c_2 \leq 2 \cdot \pi$).

Формат выходных данных

Вам необходимо вывести одно вещественное число — максимальную площадь сцены.

Ответ необходимо вывести с абсолютной или относительной погрешностью не менее 10^{-6} . Иначе говоря, если верный ответ a , а выведенный ответ b , должно выполняться условие $\frac{|a-b|}{\max(|a|, 1)} \leq 10^{-6}$.

Пример

стандартный ввод
5 0.0 1.570796 2.356194 3.926991 4.712389 5.497787
стандартный вывод
32.4759526419

Задача Е. Кубические кусты

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Мальчик Петя часто гуляет с мамой в парке «Декартгоф». «Декартгоф» представляет собой классический парк, известный своими строгими формами и прямыми линиями.

Одна из достопримечательностей «Декартгофа» — аллея Кубических кустов. Кубический куст представляет собой растение из нескольких одинаковых кубических сегментов, стоящих друг на друге. Всего в аллее Кубических кустов n кустов, высаженных в ряд. Петя увлекается геометрией, поэтому особенно любит гулять по этой аллее. В первое посещение аллеи Петя записал высоту каждого из кустов. Высота кустов — количество кубических сегментов в нём.

В аллее Кубических кустов Пете особенно нравится симметрия, поэтому он ищет последовательности из кустов, высоты которых образуют ступенчатый палиндром. Ступенчатый палиндром — последовательность нечетной длины из кустов, в первой половине которой высота каждого следующего куста больше высоты предыдущего на один, в центре последовательности находится один самый высокий куст, затем высоты кустов уменьшаются на один до конца последовательности. Высоты кустов на одинаковом расстоянии от центра одинаковые. Например, $[1, 2, 3, 2, 1]$ является ступенчатым палиндромом, а $[1, 2]$, $[1, 5, 2]$ — нет. Из всех ступенчатых палиндромов Петя хочет найти самый длинный.

Петя приходит в парк каждый месяц. Садовники в «Декартгофе» стараются подрезать кубические кусты так, чтобы аллея всегда выглядела одинаково, но Петя заметил, что в каждое посещение парка один из кустов меняет свою высоту, поэтому длина самого длинного ступенчатого палиндрома может измениться.

Петя еще маленький и не может решить задачу, поэтому помогите ему найти максимальную длину ступенчатого палиндрома в его первое посещение парка и во все последующие.

Формат входных данных

В первой строке дано одно число n — количество кустов в аллее ($1 \leq n \leq 10^5$).

Во второй строке дано n чисел a_i — изначальные высоты кустов ($1 \leq a_i \leq 10^7$).

В третьей строке дано одно число q — количество недель, когда Петя приходил в парк ($1 \leq q \leq 10^5$).

В последующих q строках дано по два числа i x — высота куста i стала равна x ($1 \leq i \leq n$, $1 \leq x \leq 10^7$).

Формат выходных данных

Выведите $q + 1$ строку — максимальную длину среди ступенчатых палиндромов в первое посещение парка Петей и во все последующие.

Пример

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

Задача F. Бардак никому не нужен

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Однажды, разбирая завал в своей комнате, Петя и Варя нашли n ниток и ножницы. Естественно, чтобы от них избавиться, они придумали игру. Первым ходит Петя, и после этого игроки ходят по очереди.

Ход состоит из одного действия: игрок выбирает нитку длины x . После этого он должен разрезать её на две нитки с натуральными длинами a и b , таким образом, что $\gcd(a, b) > 1$, и вернуть эти нитки обратно в игру. Здесь $\gcd(x, y)$ обозначает наибольший общий делитель чисел x и y .

Заметим, что после каждого хода количество ниток в игре увеличивается на один. Проигрывает тот, кто не может сделать ход.

Петя — ваш хороший друг, и он хочет во что бы то ни стало выиграть у Вари в этой игре. Подскажите ему, может ли он выиграть вне зависимости от ходов соперника.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t — количество наборов входных данных ($1 \leq t \leq 5000$). Далее следует описание наборов входных данных.

Первая строка каждого набора содержит целое число n — изначальное количество ниток в игре ($1 \leq n \leq 10^5$).

Вторая строка каждого набора содержит n целых чисел s_i — длины ниток ($2 \leq s_i \leq 10^{12}$).

Гарантируется, что сумма n по всем наборам входных данных не превосходит 10^5 .

Формат выходных данных

Для каждого набора входных данных в отдельной строке выведите «Yes», если Петя сможет обыграть Варю, и «No» в противном случае.

Пример

стандартный ввод	стандартный вывод
6	No
1	Yes
7	No
1	Yes
38	Yes
6	No
34 11 17 34 17 11	
5	
2 3 4 5 6	
5	
7 8 9 10 11	
7	
12 15 16 21 25 27 49	

Задача G. Волшебный чемодан

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Волшебница Софья получила в подарок на день рождения новый чемодан. Чемодан Софьи снабжен супер-современным электрическим кодовым замком. Замок на чемодане представляет собой квадрат со стороной n , в каждой ячейке замка написано одно число от 1 до n^2 , все числа в ячейках различны.

Софья придумывает свой код — расстановку чисел в ячейках замка, который будет знать только она. После чего она переставит числа в другом порядке, и злоумышленник, который не знает её код, не сможет взломать замок. Софья хочет, чтобы код был очень надежным.

Софья знает, что маги и волшебники будут пытаться подобрать код для взлома её чемодана, поэтому хочет, чтобы код для замка представлял собой *антимагический квадрат* — такую расстановку чисел, что все $2n + 2$ числа: суммы чисел в строках, в столбцах и на двух больших диагоналях, — различны.

Софья сейчас очень занята заклинаниями и зельями, поэтому помогите Софье придумать такой код, чтобы он представлял собой антимагический квадрат, или скажите, что такой код придумать не получится.

Формат входных данных

В первой строке ввода дано единственное целое число n , $1 \leq n \leq 1000$ — сторона квадрата.

Формат выходных данных

Если код существует, то выведите в первой строке вывода «Yes». Затем выведите n строк по n чисел в каждой — искомый код. Если подходит несколько кодов, можно вывести любой из них.

Если такой код придумать нельзя, то выведите «No».

Примеры

стандартный ввод	стандартный вывод
3	Yes 9 7 3 4 2 5 8 1 6
1	No

Задача Н. Нечетное число подмасок

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Андрей подарил Кириллу массив целых чисел $a_1, a_2, \dots, a_n$, где $0 \leq a_i < 2^{30}$.

Кирилл изучает битовые операции и пытается понять, как с их помощью можно проверять числа на чётность. В процессе изучения он заметил, что для некоторых чисел результат побитовой операции «И» оказывается *особенным* — например, когда при применении этой операции к числу с другим числом оно не изменяется.

Андрей рассказал Кириллу, что число y называется *подмаской* числа x , если

$$y \& x = y,$$

где $\&$ обозначает побитовое «И». В двоичной системе счисления это означает, что при дополнении чисел до одной длины ведущими нулями, во всех позициях, где в числе y стоит единица, в числе x тоже стоит единица. Иными словами, y можно получить из x заменой некоторых единичных битов x на нули (при дописывании ведущих нулей к двоичному представлению более короткого числа, чтобы два числа были равной длины). Например, 1 является подмаской 3, так $1 = 2^0$, $3 = 2^0 + 2^1$. А 2 не является подмаской 12, так как $2 = 2^1$, а $12 = 2^3 + 2^2$.

Кириллу стало интересно: существует ли такое целое число x ($0 \leq x < 2^{30}$), что количество элементов массива, являющихся подмасками x , оказывается нечётным? Число x не обязано само быть элементом массива.

Если такое число существует, он хочет найти *любое* из них. Если же такого числа не существует, нужно сообщить, что решений нет.

Если одно и то же число встречается в массиве несколько раз, оно учитывается столько раз, сколько встречается.

Формат входных данных

Каждый тест содержит один или несколько тестовых примеров. Первая строка содержит одно целое число t — количество тестовых примеров ($1 \leq t \leq 10^4$).

Далее следуют описания тестовых примеров.

Каждый тестовый пример начинается строкой с одним числом n ($1 \leq n \leq 10^5$) — количеством элементов в массиве.

Следующая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i < 2^{30}$).

Гарантируется, что сумма значений n по всем тестовым примерам одного теста не превышает 10^5 .

Формат выходных данных

Для каждого тестового примера, выведите -1 , если не существует такого целого числа x ($0 \leq x < 2^{30}$), что массив a содержит нечетное число его подмасок.

В противном случае, выведите любое такое x .

Пример

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

Задача I. Плейлист

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Катя слушает музыку. В очереди плеера уже есть несколько песен. Про каждую из них известны название и длина.

Плеер включает песню, которая находится первой в очереди, и сразу убирает её оттуда. Иногда Кате попадает новая композиция, которую она хочет послушать следующей, в этот момент она нажимает кнопку «Играть следующей». Эта песня становится первой в очереди плеера, но текущая композиция доигрывается до конца. Если это произошло в момент, когда текущая песня закончилась, то следующей играть будет новая композиция.

Ваша задача — определить, в каком порядке Катя послушает песни и в какой момент начнётся каждая из них.

Формат входных данных

Первая строка содержит целое число n — количество песен в плеере в момент, когда Катя начала слушать музыку ($1 \leq n \leq 100\,000$).

Далее следуют n строк, каждая из которых содержит два значения: $name_i$, len_i , где $name_i$ — названия песни, а len_i — длина песни в секундах ($1 \leq len_i \leq 10^9$).

Следующая строка содержит целое число m — количество событий добавления песни в начало очереди ($0 \leq m \leq 100\,000$).

Далее следуют m строк, каждая из которых содержит три значения: t_j , $name_j$, len_j , где t_j — момент времени добавления песни в секундах ($0 \leq t_j \leq 10^9$), $name_j$ — имя добавляемой песни, а len_j — её длина в секундах ($1 \leq len_j \leq 10^9$).

Гарантируется, что время добавления указано в порядке неубывания. Название песен состоит только из латинских букв и их длина не превосходит 20.

Формат выходных данных

Выведите для каждой проигранной песни по одной строке, содержащей два значения: $name_k$ и $start_k$, где $name_k$ — имя песни, а $start_k$ — момент времени, когда эта песня начала воспроизводиться. Песни должны быть указаны в порядке, в котором их слушала Катя.

Пример

стандартный ввод	стандартный вывод
3 Abracadabra 223 Pedro 145 Believer 220	Abracadabra 0 STAY 223 Abracadabra 343 Pedro 566
3 223 Abracadabra 223 223 STAY 120 1024 Friday 1234	Believer 711 Friday 1024

Задача J. Квантовые коты

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Алёна — знаток квантовой физики. Однажды она решила обобщить знаменитый опыт с котом и коробкой на n котов и n коробок. Для этого Алёне требуется провести n экспериментов. Для удобства она пронумеровала всех котов, все коробки и все эксперименты числами от 1 до n .

В каждом эксперименте в каждой коробке должен сидеть ровно один кот. Чтобы избежать любых неточностей, связанных с предвзятостью некоторых котов к некоторым коробкам, каждый кот должен за все n экспериментов ровно по одному разу побывать в каждой коробке. Более того, для дополнительного контроля Алёна собирается во время каждого эксперимента следить за одной из коробок и соответствующим котом в ней. В i -м эксперименте она собирается наблюдать за i -й коробкой и хочет, чтобы в ней в этот момент сидел i -й кот, чтобы гарантировать, что за все n экспериментов Алёна ровно по одному разу будет наблюдать за каждой коробкой и каждым котом.

Помогите Алёне найти возможную расстановку котов по коробкам для каждого эксперимента, если это возможно.

Формат входных данных

Вход содержит одно натуральное число n ($1 \leq n \leq 300$) — число котов, коробок и экспериментов.

Формат выходных данных

Если расстановка существует, выведите таблицу размера n на n , где значение в i -й строке и j -м столбце будет содержать номер кота, который должен сидеть в j -й коробке во время i -го эксперимента.

Если расстановки не существует, выведите целое число -1 .

Примеры

стандартный ввод	стандартный вывод
1	1
2	-1
3	1 3 2 3 2 1 2 1 3

Замечание

Во время экспериментов ни один котик не пострадал.

Задача К. Восстановление весов

Ограничение по времени: 3 секунды
Ограничение по памяти: 512 мегабайт

В деревне графов жили два исследователя, Лена и Миша. Они нашли связный неориентированный граф, состоящий из n вершин и m рёбер.

Некоторые рёбра имели известные положительные веса, а другие — неизвестные. Лена и Миша решили узнать, сколько существует способов присвоить рёбрам с неизвестными весами целые значения от 1 до l включительно так, чтобы для каждой вершины v кратчайший путь от вершины 1 до вершины v имел длину ровно d_v .

Так как число возможных способов может быть очень большим, необходимо вывести остаток от деления искомого количества на $10^9 + 7$.

Формат входных данных

Каждый тест содержит один или несколько тестовых примеров. Первая строка содержит одно целое число t ($1 \leq t \leq 50$) — количество тестовых примеров.

Далее следуют описания тестовых примеров.

Первая строка тестового примера содержит три целых числа n , m и l ($2 \leq n \leq 50$; $n - 1 \leq m \leq \frac{n(n-1)}{2}$; $1 \leq l \leq 10^9$) — число вершин, число рёбер и максимальный возможный вес неизвестного ребра.

Следующие m строк тестового примера описывают рёбра. Каждое ребро задаётся тремя целыми числами u_i, v_i, w_i ($1 \leq u_i, v_i \leq n$; $u_i \neq v_i$; $-1 \leq w_i \leq l$; $w_i \neq 0$) — концы ребра и его вес. Если $w_i = -1$, то вес этого ребра неизвестен и может быть выбран от 1 до l . Если $w_i \neq -1$, то это известный вес ребра.

Последняя строка тестового примера содержит n целых чисел $d_1, d_2, \dots, d_n$ ($0 \leq d_v \leq 10^{12}$), где d_v — требуемая длина кратчайшего пути от вершины 1 до вершины v .

Гарантируется, что граф связный, в нём нет петель и кратных рёбер.

Формат выходных данных

Для каждого тестового примера выведите одно целое число — количество способов назначить веса рёбрам с неизвестными весами от 1 до l так, чтобы сумма весов рёбер на кратчайшем пути от вершины 1 до вершины v была равна d_v для всех вершин v , взятое по модулю $10^9 + 7$.

Пример

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

Задача L. Долгий прыжок домой

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Мало кто знает, но на самом деле зайцы высоко прыгают не сразу после рождения! За всё надо платить — поэтому способности к прыжкам зайцы покупают в заячьем магазине. Вот и заяц Ксюша пришла в этот магазин, который находится в точке 0 на числовой прямой.

В магазине есть n способностей, i -я из них стоит c_i морковок и обучает зайца прыгать на расстояние x_i вдоль числовой прямой. То есть, после покупки i -й способности Ксюша сможет в любой момент времени прыгать на расстояние x_i в любую сторону.

Сразу после покупок в магазине Ксюша хочет допрыгать до точки L ($L \neq 0$), в которой находится её дом. Помогите ей выяснить, какое минимальное количество морковок придётся потратить, чтобы у неё была возможность с помощью полученных прыжков добраться до точки L , или скажите ей, что это невозможно.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t — количество наборов входных данных ($1 \leq t \leq 1\,000$). Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа n и L — количество способностей в магазине и координаты дома Ксюши ($1 \leq n \leq 3\,000$; $|L| \leq 3\,000$; $L \neq 0$).

Вторая строка каждого набора входных данных содержит n чисел x_i — длины прыжков, способности к которым можно приобрести ($1 \leq x_i \leq 3\,000$).

Третья строка каждого набора входных данных содержит n чисел c_i — стоимости способностей ($1 \leq c_i \leq 10^9$).

Гарантируется, что сумма n по всем наборам входных данных не превосходит 3 000.

Формат выходных данных

Для каждого теста в единственной строке выведите минимальное количество морковок, которое придётся потратить Ксюше, чтобы допрыгать до дома, или -1 , если это невозможно.

Пример

стандартный ввод	стандартный вывод
5	-1
3 5	7
2 4 6	8
3 1 2	9
2 555	1
5 9	
7 9	
3 12	
6 8 14	
100 3 5	
4 13	
6 10 15 13	
4 3 2 100	
4 -21	
7 7 9 4	
5 1 10 10	

Задача М. Остаток от суммы остатков

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дано целое n . Требуется вычислить сумму $i \bmod j$ по всем парам (i, j) , таким что $1 \leq i, j \leq n$.
Иными словами, требуется вычислить:

$$S = \sum_{i=1}^n \sum_{j=1}^n (i \bmod j)$$

Операция $a \bmod b$ означает остаток от деления a на b .

Так как сумма может быть большой, выведите остаток от деления искомой суммы на число 998 244 353.

Формат входных данных

В первой строке входных данных задается число n ($1 \leq n \leq 10^{12}$).

Формат выходных данных

Выведите единственное число — ответ на задачу, взятый по модулю 998 244 353.

Пример

стандартный ввод	стандартный вывод
5	26

Замечание

Рассмотрим пример.

Построим таблицу остатков $i \bmod j$ для $i, j = 1 \dots 5$:

$i \backslash j$	1	2	3	4	5
1	0	1	1	1	1
2	0	0	2	2	2
3	0	1	0	3	3
4	0	0	1	0	4
5	0	1	2	1	0

Просуммируем числа в таблице и получим 26.