Задача A. Перестановка по номеру

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Выведите перестановку по её номеру.

Формат входных данных

В первой строке входного файла записано число N $(1 \leqslant N \leqslant 12)$ — количество элементов в перестановке. Во второй строке записано число K $(0 \leqslant K < N!)$ — номер перестановки в нумерации с нуля.

Формат выходных данных

В выходной файл выведите N чисел через пробел — искомую перестановку.

Пример

стандартный ввод	стандартный вывод
3 0	1 2 3

Задача B. ПСП по номеру

Имя входного файла:	`parens.in`
Имя выходного файла:	`parens.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Определим по индукции множество $\mathcal{R}$ *правильных скобочных последовательностей*:

- $\varepsilon \in \mathcal{R}$ (пустая строка)
- $A \in \mathcal{R} \Rightarrow \texttt{(}A\texttt{)} \in \mathcal{R}$
- $A \in \mathcal{R}$, $B \in \mathcal{R} \Rightarrow AB \in \mathcal{R}$

Пусть теперь $\mathcal{R}_n$ — это множество правильных скобочных последовательностей из $2n$ символов — n открывающих и n закрывающих скобок.

Упорядочим элементы множества $\mathcal{R}_n$ лексикографически с порядком символов '$\texttt{(}' < '\texttt{)}$'.

По данным числам n и p найдите p-ый в этом порядке элемент множества $\mathcal{R}_n$.

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа n и p ($0 \leqslant n \leqslant 20$, $0 \leqslant p \leqslant 2 \cdot 10^9$). Скобочные последовательности нумеруются с нуля.

Формат выходных данных

В первой строке выходного файла выведите $2n$ символов без пробелов — p-ю правильную скобочную последовательность длины $2n$.

Если для данного n не существует p-я правильная скобочная последовательность, выведите в первой строке "`N/A`".

Примеры

parens.in	parens.out
3 0	((()))
4 2000000000	N/A
3 4	()()()

Задача C. Новогодняя гирлянда

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

256 мегабайт

Дети в детском саду как-то раз решили повесить к Новому году гирлянду. Но это оказалось для них очень трудной задачей. На помощь пришёл Дед Мороз, который теперь каждый Новый год приносит с собой гирлянду и помогает её повесить.

Гирлянда представляет собой ломаную в плоскости, состоящую из n звеньев. Гирлянда начинается в точке $(0, 0)$, возле электророзетки и должна заканчиваться в точке $(n, 0)$. Число n назыветя длиной гирлянды. Каждое звено может располагаться либо горизонтально, либо под углом 45° к оси OX. Длина горизонтальной проекции любого звена равна 1. При этом не должно быть вершины ломаной с отрицательной координатой y, а также двух последовательных вершин с нулевой координатой y. Поднимающимся (опускающимся) назовём звено ломаной, у которого координата y правого конца больше (соответственно, меньше) координаты y левого конца. Звено, у которого координаты y концов совпадают, назовём горизонтальным. Обозначим поднимающееся звено буквой `u`, опускающееся — буквой `d`, а горизонтальное — буквой `h`. Тогда гирлянда кодируется строкой из n символов. У Деда Мороза есть волшебная книга, в которой перечислены все гирлянды длины n в виде строк. Хотя книга и волшебная, строки в ней располагаются в обычном лексикографическом порядке, по возрастанию. Дед Мороз отметил на полях книги галочкой гирлянду, которую повесил в прошлый раз. В этот Новый год он хочет повесить следующую в книге гирлянду. Найдите эту гирлянду без использования волшебной книги.

Формат входных данных

В первой строке вводится целое число n ($2 \leqslant n \leqslant 100\,000$). Во второй — строчка из n букв (все буквы: `u`, `d`, либо `h`) — прошлогодняя гирлянда.

Формат выходных данных

Выведите в виде строки гирлянду, которую Дед Мороз Павлович должен прихватить с собой в этот Новый год, либо `No solution`, если такой гирлянды не существует.

Пример

стандартный ввод	стандартный вывод
6 uhduhd	uhhdud

Задача D. Диофантово уравнение

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

Даны натуральные числа a, b и c. Решите в целых числах уравнение $ax+by=c$. Среди множества решений следует выбрать такое, где x имеет наименьшее неотрицательное значение.

Формат входных данных

Входной файл содержит три целых числа a и b и c $(1 \leqslant a, b, c \leqslant 10^4)$.

Формат выходных данных

В выходной файл выведите искомые x и y через пробел. Если решения не существует, выведите одну строку «`Impossible`».

Пример

стандартный ввод	стандартный вывод
`1 2 3`	`1 1`

Задача E. Система линейных сравнений

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

256 мегабайт

Дана система из двух линейных сравнений:

$$\begin{cases} x \equiv a \pmod{n}, \\ x \equiv b \pmod{m}; \end{cases}$$

где числа n и m не обязательно взаимно простые. Решите эту систему или определите, что она не имеет решений.

Формат входных данных

В первой строке входного файла записано единственное число $1 \leqslant t \leqslant 100\,000$. В следующих t строках содержатся по четыре целых числа a, b, n, m, задающих одну систему сравнений. Все числа не превосходят по модулю 10^4, $n > 1$, $m > 1$.

Формат выходных данных

Программа должна вывести t строк, по одной на каждую систему.

В случае, если система не имеет решений, выведите строку “`NO`”.

В случае, если решение есть, то необходимо вывести слово “`YES`” и два таких числа x_0 и p, $0 \leqslant x < p$, такие, что множество чисел $x = x_0 + kp$, где k — произвольное целое число является решением данной системы.

Пример

стандартный ввод	стандартный вывод
3 3 2 5 9 1 1 5 9 7 13 20 24	YES 38 45 YES 1 45 NO

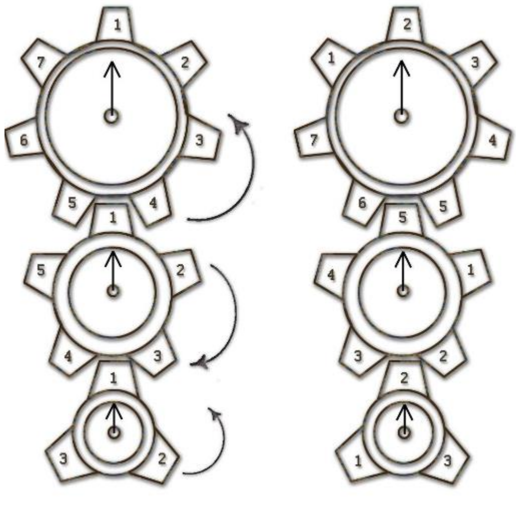
Задача F. Засекреченная переписка

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На каждой из трех осей установлено по одному вращающемуся диску и неподвижному указателю (стрелке). Диски соединены последовательно. На первом диске n зубцов, на втором — m, на третьем — k. На каждом диске первого, второго и третьего диска по часовой стрелке написаны в порядке возрастания числа от 1 до n, от 1 до m и от 1 до k, соответственно. Неподвижные указатели зафиксировали таким образом, что когда указатель первой оси указывает на число, указатели двух других осей также указывают на числа. Вася записывает три числа (a_1, a_2, a_3), на которые показывают указатели. После этого он поворачивает первое колесо на угол $\frac{360°}{n}$ против часовой стрелки, чтобы напротив указателя на первой оси оказался следующий (по часовой стрелке) зубец. При этом второе колесо поворачивается на угол $\frac{360°}{m}$ по часовой стрелке (размеры зубцов у вращающихся колесиков одинаковые, поэтому размеры самих колесиков разные, чтобы на границе колесиков равномерно уложилось разное число одинаковых по размеру зубцов), а третье колесо поворачивается на угол $\frac{360°}{k}$ против часовой стрелки. Вася снова записывает три числа, на которые указывают указатели.

Поступая и далее таким образом, Вася заметил, что после некоторого количества таких действий указатели показывают на три первоначальных числа.

Чтобы понять, как рассекречивать переписку, основанную на считывании данных с колесиков, Васе необходимо понять, как по двум данным тройкам чисел определить, принадлежат ли они к одной последовательности. Иначе говоря, можно ли целым количеством поворотов перейти от первой тройки ко второй. Вы, конечно, хотели бы помочь Васе и готовы написать программу, которая поможет ему получить ответ.



Формат входных данных

В первой строке содержится число T ($1 \leqslant T \leqslant 10$) — количество пар троек, которые хочет проверить Вася.

Во второй строке содержатся три числа n, m и k ($1 \leqslant n, m, k \leqslant 10^{18}$) — количества зубцов, соответственно, на первом, втором и третьем колесе.

В следующих $2 \cdot T$ строках записаны по три натуральных числа a_1, a_2, a_3 (первая тройка на одной строке), b_1, b_2, b_3 (вторая тройка на другой строке).

Гарантируется, что $1 \leqslant a_1, b_1 \leqslant n$, $1 \leqslant a_2, b_2 \leqslant m$, $1 \leqslant a_3, b_3 \leqslant k$.

Формат выходных данных

Для каждой пары троек выведите `YES`, если обе тройки принадлежат одной последовательности, и `NO` иначе.

Каждое слово должно быть в отдельной строке, в порядке, соответствующем входным данным.

Примеры

стандартный ввод	стандартный вывод
3 11 13 15 5 5 5 6 4 6 11 13 15 1 12 1 2 13 2 1 1 1	YES YES YES
2 2 2 2 1 1 1 1 1 2 1 1 1 2 2 2	NO YES
1 7 5 3 1 1 1 2 1 1	YES

Замечание

В первом примере в 1-й и 2-й парах вторая тройка получается из первой за один поворот первого колеса против часовой стрелки. В третьем случае из второй тройки можно получить первую одним поворотом первого колеса против часовой стрелки. Отсюда следует, что тогда из первой можно каким-то образом получить вторую.

Во втором примере в первой паре тройки нельзя перевести друг в друга. Во второй тройки переходят друг в друга при одном повороте.

Задача G. Sigma-функция на отрезке

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

Нужно научиться считать $\sum\limits_{i=L}^{R} \sigma(i)$. Где $\sigma(n)$ — сумма натуральных делителей числа n.

Формат входных данных

Последовательность из не более чем 10^5 запросов. Каждый запрос записан на отдельной строке. Формат запроса прост: числа L, R $(1 \leqslant L \leqslant R \leqslant 5 \cdot 10^6)$.

Формат выходных данных

Для каждого запроса нужно вывести одно число — $\sum\limits_{i=L}^{R} \sigma(i)$.

Пример

стандартный ввод	стандартный вывод
3 10 3 3 10 10	83 4 18

Задача H. Функция Эйлера

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

Красить забор — не очень. Вернёмся к математике.

Формат входных данных

Дано число n ($1 \leqslant n \leqslant 10^8$).

Формат выходных данных

Для каждого числа от 1 до n требуется посчитать функцию Эйлера от него. Так как чисел очень много, сначала выведите сумму функций Эйлера для первых 100 чисел, потом для вторых 100 чисел, потом для третьих 100 чисел и так далее. Если n не делится на 100, последнее из выведенных вами чисел будет состоять из суммы меньше, чем 100 слагаемых.

Примеры

стандартный ввод	стандартный вывод
10	32
200	3044 9188

Замечание

Для чисел от 1 до 10 функция Эйлера будет равна соответственно $1, 1, 2, 2, 4, 2, 6, 4, 6, 4$, что в сумме даёт 32.

Задача I. Хорошие массивы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	1024 мегабайта

Совсем недавно Вася узнал, что числа можно делить друг на друга нацело. Невероятно воодушевленный этим знанием, он стал изучать массивы, в которых одни числа делятся на другие. Вася называет массив из n целых положительных чисел $a_1,\ a_2,\ a_3,\ \ldots,\ a_n$ *хорошим*, если для любого i от 1 до $n-1$ число a_i делится нацело на число a_{i+1}. Вася очень любит изучать хорошие массивы, а поэтому ему интересно, сколько всего существует хороших массивов размера n, все числа в которых не превосходят c.

Формат входных данных

В единственной строке даны два целых числа n и c ($1 \leqslant n, c \leqslant 5 \cdot 10^7$) — количество чисел в массиве и максимальное значение чисел в массиве.

Формат выходных данных

Выведите единственное число — количество хороших массивов из n целых положительных чисел, не превосходяших c. Так как искомое количество массивов может быть слишком большим, выведите его по модулю 998 244 353.

Примеры

стандартный ввод	стандартный вывод
`3 3`	`7`
`2 6`	`14`

Замечание

В первом примере подходят следующие массивы: $(1, 1, 1)$, $(2, 1, 1)$, $(3, 1, 1)$, $(2, 2, 1)$, $(3, 3, 1)$, $(2, 2, 2)$, $(3, 3, 3)$.

Во втором примере удовлетворят условиям 14 массивов: $(1, 1)$, $(2, 1)$, $(3, 1)$, $(4, 1)$, $(5, 1)$, $(6, 1)$, $(2, 2)$, $(4, 2)$, $(6, 2)$ $(3, 3)$, $(6, 3)$, $(4, 4)$, $(5, 5)$, $(6, 6)$.

Задача J. Глина или не глина?

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

Вам дано положительное целое число k. Найдите количество троек положительных целых чисел (n, p, m), таких что $n^2 - k \cdot p^m = 1$, где p — простое число, либо сообщите, что существует бесконечное количество таких троек чисел.

Формат входных данных

В первой строке записано число t ($1 \leqslant t \leqslant 100$) — количество наборов входных данных.

Далее следует описание наборов входных данных.

Единственная строка описания набора входных данных содержит целое число k ($1 \leqslant k \leqslant 10^9$).

Формат выходных данных

Для каждого набора входных данных выведите количество троек положительных чисел (n, p, m), таких что $n^2 - k \cdot p^m = 1$ и p — простое число, либо -1, если существует бесконечное количество таких троек чисел.

Пример

стандартный ввод	стандартный вывод
2 5 22	3 0

Замечание

В первом наборе для $k = 5$ существуют три подходящие тройки чисел: $(4, 3, 1)$, $(6, 7, 1)$ и $(9, 2, 4)$.

Во втором наборе для $k = 22$ не существует ни одной подходящей тройки чисел.

Задача K. Очередная задача про теорию чисел

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Даны n **простых** чисед $1 < p_1 < p_2 < \ldots < p_n < 10^{18}$, где $p_1 \leqslant 100$. Назовем число x *хорошим*, если x делится хотя бы на одно p_i.

Рассмотрим все *хорошие* числа $a_1, a_2, \ldots, a_m$ в промежутке $[0, p_1 \cdot p_2 \cdot \ldots \cdot p_n]$ и отсортируем их по возрастанию ($a_1 < a_2 < \ldots < a_m$). Ваша задача — вычислить следующую величину: $\sum\limits_{i=1}^{m-1}(a_{i+1} - a_i)^2$. Так как ответ может быть достаточно большим, выведите остаток от деления ответа на число 998 244 353.

Формат входных данных

Первая строка содержит целое число n ($1 \leqslant n \leqslant 10^5$).

Вторая строка содержит n целых чисел $p_1, p_2, \ldots, p_n$ ($1 < p_1 < p_2 < \ldots < p_n < 10^{18}$). Гарантируется, что $2 \leqslant p_1 \leqslant 100$ и все p_i являются простыми.

Формат выходных данных

Выведите одно целое число — остаток от деления ответа на число 998 244 353.

Примеры

стандартный ввод	стандартный вывод
2 2 5	18
3 5 7 233	31275

Задача L. Маткульт-привет!

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

> Маткульт-привет!
>
> Алексей Савватеев

Сегодня на очередном занятии в математическом кружке, посвященном теории чисел, Сережа узнал много новых для него интересных функций. В частности, ему очень понравилась функция $\varphi(n)$, которая определяется следующим образом: $\varphi(n)$ равно количеству натуральных чисел, не превосходящих n, *взаимно-простых* с n. Эта функция показалась Сереже очень красивой, так как на занятии он узнал несколько ее замечательных свойств. Например, для любых *взаимно-простых* чисел a и b верно, что $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$.

Напомним, что натуральные числа a и b называются *взаимно-простыми*, если их наибольший общий делитель равен единице. Например, числа 5 и 8 являются взаимно-простыми, а числа 12 и 9 — нет (их наибольший общий делитель равен 3).

Приведем некоторые примеры значений функции $\varphi(n)$:

- $\varphi(5) = 4$ (натуральные числа, не превосходящие 5, взаимно-простые с 5: $1, 2, 3, 4$),
- $\varphi(1) = 1$ (существует всего одно натуральное число, не превосходящее 1 — само число 1),
- $\varphi(6) = 2$ (натуральные числа, не превосходящие 6, взаимно-простые с 6: $1, 5$).

Сережа очень любит натуральные числа из промежутка $[l, r]$, то есть числа $l, l+1, \ldots, r$. Начинающему математику тут же захотелось исследовать поведение функции $\varphi(n)$ на промежутке $[l, r]$.

Сережа хочет найти такое натуральное число x, что $l \leqslant x \leqslant r$, а также $\varphi(x) \geqslant \varphi(y)$ для любого натурального числа $l \leqslant y \leqslant r$. Так как Сережа является начинающим математиком, он не справился с этой задачей, поэтому решить ее придется вам.

Формат входных данных

Единственная строка содержит два натуральных числа l и r ($1 \leqslant l \leqslant r \leqslant 10^{12}$).

Формат выходных данных

Выведите одно натуральное число x, для которого верно, что $l \leqslant x \leqslant r$, а также $\varphi(x) \geqslant \varphi(y)$ для любого натурального числа $l \leqslant y \leqslant r$.

Если существует несколько подходящих чисел x, выведите любое из них.

Примеры

стандартный ввод	стандартный вывод
1 6	5
10 10	10
14 16	16

Замечание

В первом примере значения функции $\varphi(n)$ для всех натуральных чисел из промежутка $[1, 6]$ равны: $\varphi(1) = 1, \varphi(2) = 1, \varphi(3) = 2, \varphi(4) = 2, \varphi(5) = 4, \varphi(6) = 2$.

Во втором примере 10 — единственное натуральное число из промежутка $[10, 10]$.

В третьем примере можно вывести в качестве ответа числа 15 или 16, так как $\varphi(14) = 6$, а $\varphi(15) = \varphi(16) = 8$.

Задача M. Полифемовы тройки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Циклоп Полифем, некогда ослепленный хитроумным Одиссеем, ныне бросил овцеводство и занимается математикой. За прошедшее время обида на коварного грека несколько улеглась, Полифем проанализировал ситуацию и всецело поглощен работой над ошибками. Корни своего поражения слепой Полифем видит в незнании квадратных корней; им и только им посвящены его изыскания.

В настоящий момент циклопа занимают тройки целых неотрицательных чисел, обладающие следующим свойством: сумма корней из первых двух элементов равна корню из третьего (из уважения к ученому мы будем называть такие тройки *полифемовыми*). Так, например, $\sqrt{7857} + \sqrt{24832} = \sqrt{60625}$ — полифемова тройка.

В наибольшей степени циклопа заинтересовал тот факт, что некоторые числа могут принадлежать более, чем одной полифемовой тройке. Для всякого числа C Полифем обозначил $z(C)$ количество пар целых неотрицательных чисел $A \leqslant B$, для которых $\sqrt{A} + \sqrt{B} = \sqrt{C}$. Циклоп нашел поистине превосходный алгоритм вычисления $z(C)$ с помощью циркуля и линейки, но увы: использовать его на практике Полифему мешает собственная слепота! Помогите циклопу найти значение функции $z(C)$.

Формат входных данных

В единственной строке находится одно целое число C, $0 \leqslant C \leqslant 10^{18}$.

Формат выходных данных

Выведите ровно одно целое число — $z(C)$.

Примеры

стандартный ввод	стандартный вывод
9	2
3	1

Задача N. Чиселки

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Даны два числа n и k.

Определим q_i. Изначально есть число i. Вы можете изменять его двумя способами:

1. Умножить текущее число на какое-то простое $p \leqslant n$.
2. Разделить текущее число на какое-то простое $p \leqslant n$ (если делится).

q_i — количество различных чисел, которые можно получить, если вы можете выполнить эти операции в сумме не более k раз.

Найдите $\sum\limits_{i=1}^{n} i \cdot q_i$ по модулю $10^9 + 7$.

Формат входных данных

Первая строка содержит два целых числа n, k ($1 \leqslant n, k \leqslant 10^6$).

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
3 1	23
4 2	82

Задача O. Гиперпрефиксные суммы

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Дан массив s_0, состоящий из n элементов. После этого по массиву s_0 строится массив s_1 следующим образом:

$$s_1[i] = \sum_{j=1}^{i} s_0[j] \pmod{998244353}$$

Затем по аналогичной формуле по массиву s_1 строится массив s_2, и так далее. От вас требуется вывести элементы массива s_k.

Формат входных данных

В первой строке через пробел записаны два числа n и k ($1 \leqslant n \leqslant 2000, 0 \leqslant k \leqslant 10^9$).

Во второй строке через пробел записаны n целых чисел — элементы массива s_0 ($0 \leqslant s_0[i] < 998244353$).

Формат выходных данных

Выведите через пробел n целых чисел — элементы массива s_k.

Примеры

стандартный ввод	стандартный вывод
`4 1` `3 20 3 4`	`3 23 26 30`
`1 1` `3`	`3`
`5 0` `3 14 19 92 6`	`3 14 19 92 6`