

Задача А. Чипсы

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

Программист Василий купил k банок с чипсами, по n чипсов в каждой. Теперь он смотрит кино и ест чипсы. После каждой минуты фильма наш герой берёт какую-то банку и съедает один чипс из неё.

В один прекрасный момент Василий взял какую-то банку, но чипсов там не оказалось. Какое минимальное и какое максимальное число минут могло пройти с начала фильма?

Формат входных данных

В единственной строке заданы два числа k и n через пробел — число банок с чипсами и их размер, соответственно ($1 \leq k, n \leq 50$).

Формат выходных данных

Выведите два числа через пробел: минимальное и максимальное число минут, которое могло пройти до того момента, как Василий взял банку без чипсов.

Примеры

стандартный ввод	стандартный вывод
3 4	5 13
49 15	16 736

Задача В. Карта Программ

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

В далёкой стране жил мудрый старец — хранитель всех компьютерных программ, которые когда-либо были созданы. Однажды правитель его страны повелел провести перепись всех программ, и старец задумал создать для этого Карту Программ, отображающую наличие или отсутствие различных приложений.

Он обратился к вам за помощью в этом важном деле. Вам предоставлен список программ, названия которых состоят из маленьких английских букв.

Ваша задача — заполнить таблицу из пяти строк и шести столбцов. Первые 26 клеток таблицы соответствуют буквам «a», «b», ..., «z» в алфавитном порядке. Последние четыре клетки пусты. В каждой клетке таблицы должна стоять соответствующая буква, если в списке есть хотя бы одна программа, название которой начинается с этой буквы. Если же программ с такой начальной буквой нет, следует поставить в ячейке точку.

Формат входных данных

В первой строке задано целое число n — число программ ($0 \leq n \leq 100$). Далее в n строках идет перечень названий программ. Название каждой программы содержит от 1 до 20 маленьких английских букв.

Формат выходных данных

Выведите Карту Программ в описанном выше формате: пять строк с буквами и точками, без пробелов между ними. В первых четырёх строках выведите по шесть символов. В последней строке выведите два символа.

Пример

стандартный ввод	стандартный вывод
10	abcdef
apple	g...kl
banana	m.....
cherry
date	..
eggplant	
fig	
grape	
kiwi	
lemon	
mango	

Задача С. Ординальное число

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

Ординальные числа — расширение множества целых неотрицательных чисел. Каждому целому неотрицательному числу x поставим в соответствие ординальное число — множество $f(x)$. Первые несколько ординальных чисел можно определить так:

- Нулю сопоставим пустое множество:
 $f(0) = \{\}$.
- Единице сопоставим множество, содержащее множество $f(0)$ как элемент:
 $f(1) = \{f(0)\} = \{\{\}\}$.
- Двойке сопоставим множество, содержащее $f(0)$ и $f(1)$ как элементы:
 $f(2) = \{f(0), f(1)\} = \{\{\}, \{\{\}\}\}$.
- И так далее: каждому положительному целому числу k сопоставим множество, состоящее из всех предыдущих ординальных чисел как элементов. Формула:
 $f(k) = \{f(0), f(1), \dots, f(k-1)\}$.

Далее можно аналогично определить ординальные числа, не соответствующие целым. Увы, в нашей задаче это не понадобится.

Дана запись ординального числа, соответствующего какому-то целому неотрицательному числу n . Найдите n .

Формат входных данных

В первой строке записано ординальное число, соответствующее целому неотрицательному числу n ($0 \leq n \leq 15$). Строка состоит из символов «{», «», «,» и «}».

В каждом множестве, встречающемся в записи, все элементы перечислены ровно по одному разу. Однако, поскольку множество не меняется от перестановки элементов в нём, порядок перечисления может быть произвольным.

Формат выходных данных

Выведите целое число n , соответствующее заданному ординальному числу.

Примеры

стандартный ввод	стандартный вывод
{}	0
{\{\}}	1
{\{\}, \{\{\}\}}	2
{\{\{\}\}, \{\{\{\}\}\}, \{\{\}\}, \{\{\}\}}	3

Задача D. RACI

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

Николай на лекциях по менеджменту дал студентам задание составить для проекта матрицу RACI. Это матрица распределения ответственности, в которой перечислены все заинтересованные стороны проекта и уровни их вовлечённости в отдельные задачи. Уровни обозначаются буквами «R», «A», «C» и «I». При отсутствии вовлечённости используется «-». Уровни вовлечённости имеют следующее значение:

- R (*Responsible*): выполняет задачу (если таких нет, то *Accountable* выполняет задачу самостоятельно)
- A (*Accountable*): принимает задачу у *Responsible*; для каждой задачи должен быть ровно один экземпляр этого уровня вовлечённости — в отличие от остальных уровней, которых может быть сколько угодно
- C (*Consulted*): консультирует в процессе выполнения задачи
- I (*Informed*): получает информацию о ходе выполнения задачи

Помогите студентам проверить корректность заполнения матрицы.

Формат входных данных

Первая строка содержит два целых числа n и m — число строк и столбцов матрицы RACI ($1 \leq n, m \leq 100$).

Далее следуют n строк, каждая из которых содержит по m элементов, разделённых пробелами. Каждая строка соответствует задаче, а каждый столбец — заинтересованной стороне.

Каждый элемент матрицы — либо заглавная английская буква «R», «A», «C» или «I», либо минус «-», означающий, что данная заинтересованная сторона по этой задаче не имеет никакого уровня вовлечённости.

Формат выходных данных

Выведите «Yes», если матрица корректна, или «No» иначе. Регистр букв не важен.

Примеры

стандартный ввод	стандартный вывод
3 5 C C A - I A R - C I A R I C -	Yes
3 3 A - C R C I - A I	No

Задача E. Треугольник на оси

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

На плоскости дан набор точек с целочисленными координатами. Найдите треугольник наибольшей площади с вершинами из этого множества точек, одна из сторон которого лежит на оси Ox .

Формат входных данных

В первой строке дано целое число n — число точек ($1 \leq n \leq 1000$). Каждая из следующих n строк содержит два целых числа x и y — координаты точек. Все координаты не превосходят 1000 по абсолютной величине.

Формат выходных данных

Выведите одно вещественное число — максимальную площадь треугольника, удовлетворяющего условию задачи. Если такого треугольника не существует или он вырожденный, выведите 0.

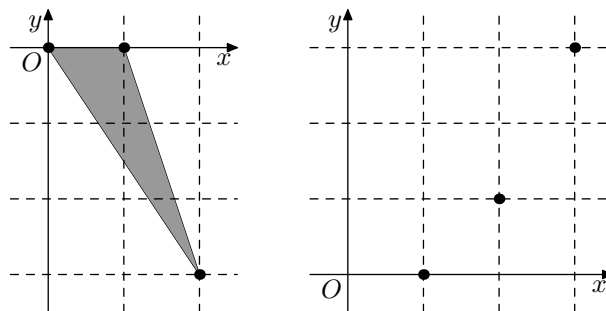
Ваш ответ будет считаться верным, если он отличается от правильного не более чем на 10^{-9} .

Примеры

стандартный ввод	стандартный вывод
3 0 0 1 0 2 -3	1.5
3 1 0 2 1 3 3	0

Замечание

Иллюстрации к примерам:



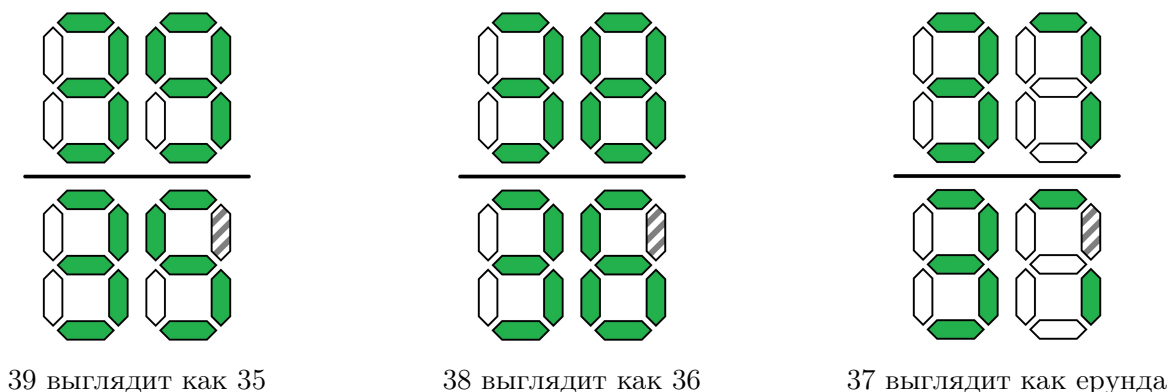
Задача F. Неисправный светофор

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

Недалеко от дома Влада стоит светофор для пешеходов, который показывает оставшееся на переход время. Каждую секунду показываемое светофором число уменьшается на 1. Но однажды Влад подошёл к пешеходному переходу и не поверил своим глазам: показания светофора переключились с 35 на 36!

Влад сильно удивился, но его удивление прошло уже на следующей секунде: вместо 37 или какого-либо другого числа светофор показал что-то, не являющееся числом вообще. Оказалось, что один из индикаторов светофора был сломан и никогда не горел. Из-за этого число 39 выглядело как 35, а 38 — как 36 (см. рисунки 1–3). После этого Влад задумался: насколько долго может продолжаться такая иллюзия того, что светофор «тикает вверх на 1» каждую секунду?

Рисунки 1–3. Поведение светофора с момента, когда он переключился на 39 до момента, когда он переключился на 37: ожидаемое — сверху, реальное — снизу. Зелёным (выглядит серым при чёрно-белой печати) выделены горящие индикаторы, белым — исправные, но не горящие в данный момент, заштрихованным — неисправный индикатор.



Например, в описанном примере иллюзия продолжалась 1 секунду: от момента, когда светофор «переключился с 35 на 36» (на самом деле с 39 на 38) до момента, когда светофор начал показывать ерунду. Строже говоря, светофор «тикал вверх на 1» в течение k секунд, если он последовательно показал $x, x + 1, \dots, x + k$.

Перед вами стоит следующая задача. Светофор показывает две цифры. Обе цифры «рисуются» стандартным образом с помощью семи индикаторов-палочек: четырёх вертикальных и трёх горизонтальных (см. рисунок 4).

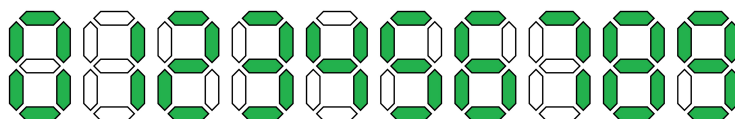


Рисунок 4. Как выглядят цифры на исправном светофоре

Каждый индикатор либо исправен (работает как надо), либо сломан (никогда не горит). Переключение цвета светофора всегда работает исправно и всегда разрушает иллюзию «тикания вверх». На переход улицы даётся ровно 99 секунд, то есть исправный светофор последовательно показывает 99, 98, 97, \dots , 2 и 1. Он **не** показывает 0, а вместо этого сразу переключается на другой цвет.

Важное замечание: светофор **не показывает лидирующие нули**, то есть на исправном светофоре число 6 отображается как одна шестёрка во втором разряде, а не как «06». На рисунках 5–6 можно увидеть, как числа 6 и 37 отображаются на исправном светофоре.

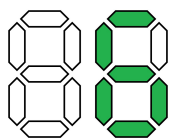


Рисунок 5. Число 6 на исправном светофоре

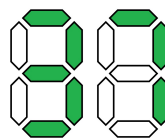


Рисунок 6. Число 37 на исправном светофоре

Если такая иллюзия вообще не может создаваться (в частности, в том случае, когда светофор вообще всегда показывает ерунду), то ответом считается число 0.

Итого, можно формализовать задачу следующим образом. Найти наибольшее такое неотрицательное k , что существует хотя бы одно такое y , удовлетворяющее условиям $k + 1 \leq y \leq 99$, что числа $y, y - 1, \dots, y - k$ выглядят из-за сломанных индикаторов как числа $x, x + 1, \dots, x + k$. Если такого k не существует, ответом считается 0.

Формат входных данных

В первой строке дана строка из семи символов, каждый из которых — «0» или «1». Они описывают состояния индикаторов в первом разряде показываемого числа в соответствии с порядком на рисунке 7: первый символ соответствует верхнему горизонтальному индикатору, второй — левому верхнему вертикальному, третий — правому верхнему вертикальному, четвёртый — среднему горизонтальному, пятый — левому нижнему вертикальному, шестой — правому нижнему вертикальному, седьмой — нижнему горизонтальному. Символ «1» означает, что соответствующий индикатор исправен, а «0» — что сломан.

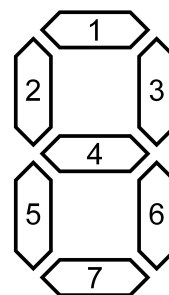


Рисунок 7. Порядок индикаторов

Во второй строке аналогично задано состояние индикаторов во втором разряде показываемого числа.

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
1111111 1101111	1
1011011 1101111	1
1101111 0010010	0
0000000 0000000	0

Замечание

Первый пример — рассмотренная в условии ситуация. В условии показано, как иллюзия может сохраняться в течение секунды. Можно показать, что дольше она сохраняться не может.

Во втором примере дополнительно сломаны второй и пятый (левые вертикальные) индикаторы в первом разряде. Ответ 1 всё ещё достигается, причём тем же самым способом (39 выглядит как 35, а 38 — как 36).

В третьем примере ответ 0. Обратите внимание, что 90 на таком светофоре выглядит как 51, а 89 — как 61, что больше 51. Тем не менее, эти два последовательных показания светофора **не** образуют иллюзию длины 1, так как числа 51 и 61 не являются последовательными.

В четвёртом примере все индикаторы сломаны, и поэтому светофор никогда не показывает корректно записанное число. В этом случае, в соответствии с договорённостью, нужно вывести 0.

Задача G. Программисты и камни

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

Программисты Алиса и Дмитрий придумали новую игру. В этой игре на столе есть n кучек камней. Игроки делают ходы по очереди, начинает Алиса. На своём ходу игрок может выбрать любое непустое множество непустых кучек, после чего убрать по одному камню из каждой из них. Проигрывает тот, кто не может сделать ход. Кто выигрывает при правильной игре?

Формат входных данных

В первой строке задано целое число n ($1 \leq n \leq 100\,000$).

Во второй строке заданы n чисел a_1, a_2, \dots, a_n — изначальные размеры кучек камней ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите «Alice» или «Dmitry», в зависимости от того, кто выигрывает при правильной игре. Регистр букв в именах **важен**.

Примеры

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

Задача Н. Минимизация обменами

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

Маша изучает большие числа. Она положила в ряд n карточек. На каждой карточке написана цифра от 1 до 9. Вместе они образуют n -значное целое число s .

За одну операцию Маша может взять две **соседние** карточки и поменять их местами (поворачивать карточки, превращая одни цифры в другие, нельзя). Маша может произвести не более k операций. Какое минимальное n -значное число может получиться в итоге?

Формат входных данных

В первой строке задано целое число t — число наборов входных данных ($1 \leq t \leq 100\,000$). Далее следуют сами наборы.

Каждый набор входных данных — это строка, в которой записаны через пробел целые числа s и k . Число s положительное и состоит из цифр от 1 до 9. Кроме того, $0 \leq k \leq 10^{18}$.

Суммарное число цифр во всех числах s не превосходит 100 000.

Формат выходных данных

Для каждого набора входных данных выведите строку с ответом: минимальное n -значное число, которое может получиться из s , если не более k раз поменять местами две соседние цифры.

Пример

стандартный ввод	стандартный вывод
4	321
321 0	9
9 1	11122247
21241127 10	629
692 1	

Задача I. Такси по модулю

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

Лонгляндия — очень длинная страна. Все её n городов расположены вдоль отрезка прямой. Если их пронумеровать в порядке от начала до конца отрезка, то в i -м городе a_i жителей.

Вам надо добраться от города s до города f . Для этого в Лонгляндии работает бесконечное число марок такси, называемых Kaban-2, Kaban-3, Kaban-4, Kaban-5, Такси под названием Kaban- m может довезти вас от города i до города j , если во всех городах от i до j включительно количества жителей сравнимы между собой по модулю m . Формально, для любого целого k , для которого $\min\{i, j\} \leq k \leq \max\{i, j\}$, должно выполняться $a_k \equiv a_i \pmod{m}$.

Найдите наименьшее число Q вызовов такси, достаточное, чтобы добраться от города s до города f , и выведите Q строк с описанием маршрута. Если добраться при помощи такси не получится, выведите «Impossible».

Формат входных данных

В первой строке находится целое число n — число городов в Лонгляндии ($2 \leq n \leq 2 \cdot 10^5$).

Во второй строке даны n целых чисел a_1, a_2, \dots, a_n — население каждого из городов ($1 \leq a_i \leq 10^9$).

В третьей строке записаны два целых числа s и f — номера стартового и финишного городов ($1 \leq s, f \leq n; s \neq f$).

Формат выходных данных

Пусть Q — наименьшее число вызовов такси, достаточное, чтобы добраться от города s до города f . Выведите Q строк вида «Kaban- m_i s_i f_i », означающих, что i -ю поездку осуществит такси Kaban- m_i и довезёт вас от города s_i до города f_i ($1 \leq s_i, f_i \leq n; 2 \leq m_i \leq 10^9$). Должны выполняться равенства: $s_1 = s; f_Q = f; s_{i+1} = f_i$. Ну и, конечно, такси Kaban- m_i должно иметь возможность доставить вас от города s_i до города f_i .

Если ни за какое число вызовов такси невозможно добраться от s до f , выведите одно слово «Impossible».

Регистр букв не важен, то есть вы можете выводить, например, «kaBAN» и «IMPOSSIBLE».

Примеры

стандартный ввод	стандартный вывод
6 1 2 3 4 5 6 5 3	Impossible
8 1 16 20 20 20 23 7 8 1 7	Kaban-5 1 2 Kaban-4 2 5 Kaban-3 5 6 Kaban-8 6 7
11 55 55 55 55 55 55 55 55 55 55 7 2	kaBAN-239239239 7 2

Задача J. Электрик

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

В деревне J есть n домов с доступом в интернет. В одном из домов сидит злодей и делает свои злодейские дела: пишет нам без остановки гневные комментарии в соцсетях с фейкового аккаунта. Мы не знаем, в каком доме он живёт, но хотим это выяснить.

На электроподстанции есть n рубильников: по одному для каждого дома. Когда рубильник включён, свет в доме есть, а когда выключен — нет, а значит, нет и доступа в интернет. Изначально все рубильники включены. В начале каждого часа мы можем произвольно менять состояние рубильников: какие-то включать, какие-то выключать. После чего в течение следующего часа мы следим за соцсетями: если в доме злодея есть свет, комментарии обязательно появятся, а если света нет, то и комментариев не будет. После наших исследований мы должны вернуть всё в исходное состояние: включить свет во всех домах.

К сожалению, если свет в доме злодея пропадёт более k раз, то он заподозрит неладное, а мы хотели бы этого избежать. При этом, когда свет просто долго отсутствует, это не вызывает подозрения — учитывается именно число отключений. За какое наименьшее число часов мы можем с уверенностью сказать, в каком доме живёт злодей, и включить всем свет?

Формат входных данных

В первой строке задано целое число t — число наборов входных данных ($1 \leq t \leq 2 \cdot 10^5$). Далее следуют t наборов.

Каждый набор входных данных задан на отдельной строке, содержащей два целых числа n и k — число домов и максимальное возможное число отключений света в доме злодея ($1 \leq n, k \leq 10^{18}$).

Формат выходных данных

Для каждого набора входных данных выведите строку с одним целым числом: наименьшим числом часов, которые потребуются, чтобы точно определить дом злодея и включить всем свет.

Пример

стандартный ввод	стандартный вывод
5	2
3 1	2
4 1	1
2 2	5
17 3	7
73 6	

Замечание

В первом наборе входных данных $n = 3$ и $k = 1$. Мы можем в начале первого часа отключить свет в домах с номерами 1 и 2, затем спустя час вернуть свет в доме номер 1, а спустя ещё час — вернуть свет в доме номер 2. К началу третьего часа мы будем точно знать, где живёт злодей, и вернём свет во все дома.

Задача К. Деревянная матрица

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

Рассмотрим квадратную матрицу размера $n \times n$, состоящую из неотрицательных целых чисел. Матрица симметрична относительно главной диагонали, которая, в свою очередь, содержит лишь нули. Такая матрица называется *деревянной*, если существует такое неориентированное дерево T на n вершинах с рёбрами положительных длин, что в каждой ячейке матрицы (i, j) записано расстояние между вершинами i и j в этом дереве.

Дана матрица. Проверьте, является ли она деревянной.

Формат входных данных

В первой строке дано целое число n — размер матрицы ($1 \leq n \leq 1000$). В каждой из следующих n строк задано по n целых чисел $d_{i,j}$ — элементы матрицы ($0 \leq d_{i,j} \leq 10^9$). Матрица симметрична относительно главной диагонали. На главной диагонали стоят нули, а вне её — строго положительные целые числа.

Формат выходных данных

Выведите «Yes» или «No» в зависимости от того, является ли матрица деревянной. Регистр букв не важен.

Примеры

стандартный ввод	стандартный вывод
3 0 1 3 1 0 2 3 2 0	Yes
3 0 1 3 1 0 1 3 1 0	No

Задача L. Число операций

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

На занятии кружка Аня написала программу, которая для всех чисел от 0 до $n - 1$ проделывает работу:

```
for (int i = 0; i < n; i++) {
    work (i);
}
```

Преподаватель Петя посмотрел на программу и сказал, что она будет работать слишком долго, ведь максимальное значение n равно 10^9 .

Тогда Аня решила перебирать не все числа, а только некоторые. Её новая программа выглядит так:

```
for (int i = 0; i < n; i++) {
    if (i == x_1) i = y_1;
    if (i == x_2) i = y_2;
    ...
    if (i == x_k) i = y_k;
    work (i);
}
```

Сколько операций теперь делает эта программа для заданного числа n , и завершает ли она вообще работу когда-нибудь? Операциями следует считать присваивания ($i = 0; i++; i = y_1; \dots$), сравнения ($i < n; i == x_1; \dots$) и работу ($work (i)$).

Формат входных данных

Во вводе задана программа, которую написала Аня. Программа отформатирована ровно так, как показано в примерах: в частности, открывающая фигурная скобка находится на строке с `for`, перед открывающей круглой скобкой всегда стоит пробел, а отступы состоят из четырёх пробелов. Во вводе нет пробелов в конце строк.

Число строк с условиями: $0 \leq k \leq 10^5$. Ограничения на числа: $0 \leq x_i, y_i < n \leq 10^9$, в остальном числа могут быть любыми. Строка с `work` следует один раз после всех строк с `if`.

Формат выходных данных

Выведите, сколько операций делает заданная программа. Если программа никогда не завершит работу, выведите `-1`.

Примеры

стандартный ввод	стандартный вывод	операции
<pre>for (int i = 0; i < 3; i++) { work (i); }</pre>	11	<pre>i = 0; 0 < 3; work (0); i++; 1 < 3; work (1); i++; 2 < 3; work (2); i++; 3 < 3.</pre>
<pre>for (int i = 0; i < 5; i++) { if (i == 3) i = 1; if (i == 1) i = 3; if (i == 3) i = 4; work (i); }</pre>	16	<pre>i = 0; 0 < 5; 0 == 3; 0 == 1; 0 == 3; work (0); i++; 1 < 5; 1 == 3; 1 == 1; i = 3; 3 == 3; i = 4; work (4); i++; 5 < 5.</pre>

Задача M. Шары трёх цветов

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

Имеется r красных, g зелёных и b синих шаров. Сколько существует способов положить все эти шары в ряд так, чтобы любые два соседних шара имели разные цвета? Так как это число способов может быть очень большим, выведите его остаток при делении на простое число 998 244 353.

Формат входных данных

Даны три целых числа через пробел: r , g и b . Каждое из чисел — от 1 до 10^5 включительно.

Формат выходных данных

Выведите одно число: искомое число способов по модулю 998 244 353.

Примеры

стандартный ввод	стандартный вывод
1 1 1	6
4 1 1	0
1 1 2	6

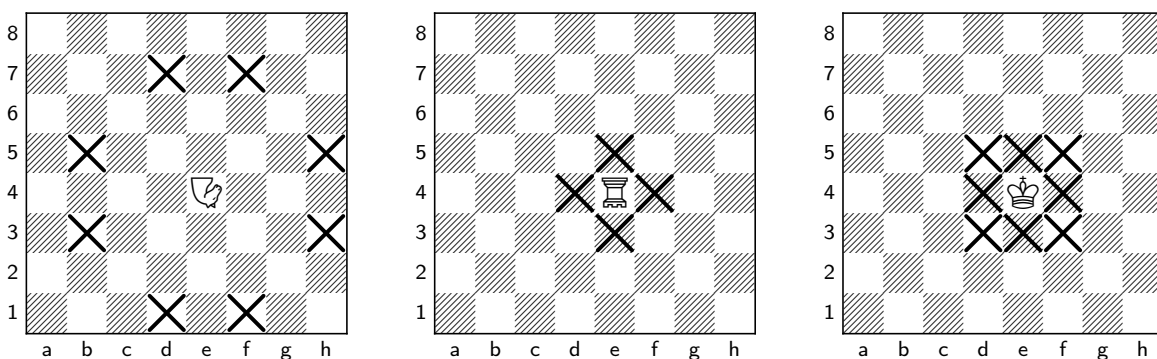
Задача N. Совершенно точно не шахматы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	15 секунд
Ограничение по памяти:	1024 мегабайта

Ваша задача — поставить мат чёрному королю при помощи белого короля, верблюда и визиря.

Верблюд (обозначается перевернутой фигурой коня) — это шахматная фигура, которая ходит удлинённым ходом коня: на три клетки по одному направлению (на одну больше, чем конь) и на одну по другому. Визирь (обозначается перевернутой ладьей) всегда ходит на соседнюю по стороне клетку, то есть ладейными ходами длины 1. Король ходит как в обычных шахматах: на любую клетку, соседнюю по горизонтали, вертикали или диагонали.

Ниже показаны возможные ходы всех фигур с поля e4. Поля, достижимые за один ход, помечены крестиком.



Правило ничьей после 50 ходов не применяется.

Формат входных данных

В первой строке задано целое число t ($1 \leq t \leq 2 \cdot 10^6$) — число наборов входных данных. Далее следуют t наборов.

Каждый набор входных данных задан на отдельной строке в следующем формате: поля в алгебраической нотации, на которых стоят белый король, чёрный король, белый верблюд и белый визирь, а за ними буква «w» или «b», обозначающая ход белых либо чёрных, соответственно. Алгебраическая нотация для полей на шахматной доске состоит из двух символов: номера вертикали, закодированного буквой от «a» до «h», и номера горизонтали, закодированного цифрой от «1» до «8». Горизонтали нумеруются снизу вверх, а вертикали слева направо, как показано на картинках.

В каждом наборе входных данных:

- Все фигуры стоят на различных полях.
- Игрок, не имеющий права хода, не находится под шахом.
- Партия ещё не закончилась, то есть на доске не может быть пат или мат.

Формат выходных данных

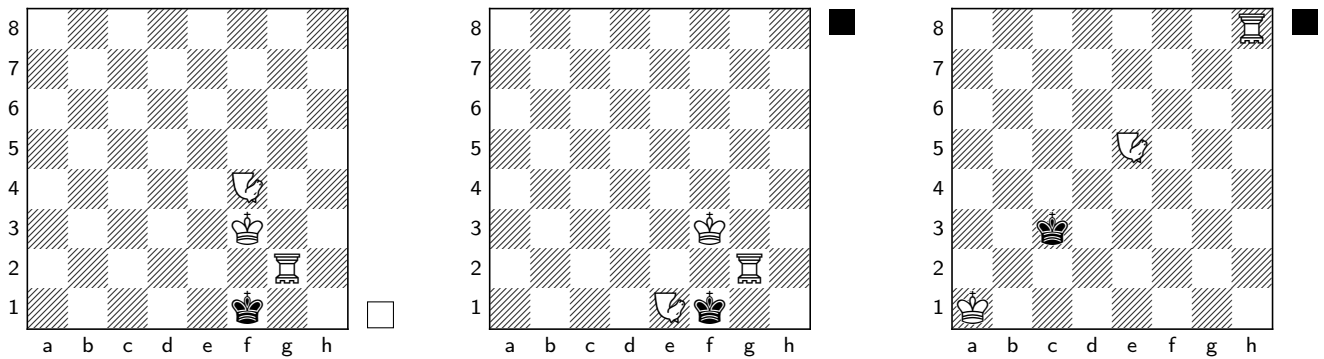
Для каждого набора входных данных выведите оценку позиции в отдельной строке. Оценка позиции должна быть либо числом 0, если позиция ничейная, либо символом «#», за которым сразу следует минимальное число ходов, необходимых, чтобы гарантированно поставить мат.

Пример

стандартный ввод	стандартный вывод
3	#1
f3 f1 f4 g2 w	0
f3 f1 e1 g2 b	#66
a1 c3 e5 h8 b	

Замечание

Позиции в примере выглядят так:



В этой задаче используются следующие стандартные шахматные правила и понятия:

- Два игрока ходят по очереди: один чёрными фигурами, другой белыми.
- Во время своего хода игрок обязан выбрать одну фигуру своего цвета и сделать ею ход.
- Запрещено ходить на поля, занятые фигурами своего цвета.
- Ход на поле с фигурой противоположного цвета называется *взятием*. Взятая фигура удаляется из игры.
- *Шах* — это состояние, когда одна из фигур может пойти на поле с королём.
- Игроку запрещено ходить так, чтобы его король оказался под шахом в конце хода.
- Состояние, когда игрок с правом хода не имеет корректных ходов и его король **не** находится под шахом, называется *патом* и приводит к ничьей.
- Состояние, когда игрок с правом хода находится под шахом и не имеет корректных ходов, называется *матом* и объявляется победой его оппонента. В этой задаче победы могут достичь только белые.
- При подсчёте числа ходов до мата учитываются только ходы матующей стороны.
- Позиции, из которых нельзя достичь мата при оптимальной игре, в этой задаче считаются ничейными.