

## Problem A. Анализ хода матча

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 1 секунда  
Memory limit: 512 мегабайт

В спин-оффе популярного фильма Алиса играет шахматный матч с Бобом. Матч состоит из нескольких игр и начинается со счёта 0:0. Если какой-то игрок выигрывает игру, к её/его баллам прибавляется 1, в противном случае (в случае ничьей) каждый игрок получает по 0.5 балла за эту партию.

Вам дан текущий счёт. Найдите общее количество сыгранных партий, а также наименьшее и наибольшее возможное количество ничьих в матче.

### Input

Первая строка входных данных содержит одно вещественное число  $A$  — количество баллов, набранных Алисой. Вторая строка содержит одно вещественное число  $B$  — количество баллов, набранных Бобом ( $0 \leq A, B \leq 6$ ). Числа даны без лишних знаков (то есть без лишних нулей после десятичной точки и без самой десятичной точки в случае целых чисел). Гарантируется, что существует такая последовательность результатов партий, при которой счёт  $A:B$  будет достигнут.

### Output

В первой строке выведите общее количество сыгранных партий. Во второй — наименьшее возможное количество ничьих в матче. В третьей — наибольшее возможное количество ничьих в матче.

### Examples

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 1.5              | 2                 |
| 0.5              | 1                 |
|                  | 1                 |
| 0                | 0                 |
| 0                | 0                 |
|                  | 0                 |

## Problem B. Больше разнообразия

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 1 секунда  
Memory limit: 512 мегабайт

Дано целое число  $N$ . Ваша задача — построить такую последовательность целых чисел  $a_i$  длины  $M$ , что:

- $1 \leq a_i \leq N$ .
- $a_i \neq a_j$ , если  $i \neq j$ .
- Для всех  $i \neq j$  и  $k \neq l$  верно, что  $a_i + a_j \neq a_k + a_l$ , если  $(i, j) \neq (k, l)$  и  $(i, j) \neq (l, k)$  (то есть все  $M(M-1)/2$  попарных сумм различны).
- $M \geq \sqrt{N}/2$ .

### Input

Входные данные содержат одно целое число  $N$  ( $1 \leq N \leq 5 \cdot 10^6$ ).

### Output

В первой строке выведите одно целое число  $M$  — длину последовательности  $a_i$ . Во вторую строку выведите  $M$  попарно различных целых чисел между 1 и  $N$  включительно — последовательности  $a_i$ .

Если более одной последовательности удовлетворит требованиям задачи, выведите любую из них.

### Example

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 10               | 3<br>1 3 7        |

## Problem C. Выделяем римские анаграммы

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 1 секунда  
Memory limit: 512 мегабайт

Вам дана строка  $S$ , составленная из букв I, V, X, L, C, D и M.

Ваша задача — подсчитать количество различных анаграмм этой строки, которые являются корректными римскими числами (сама строка также считается своей анаграммой).

Таблица из Википедии показывает, каким образом записываются римские числа:

| Разряды | Тысячи | Сотни | Десятки | Единицы |
|---------|--------|-------|---------|---------|
| 1       | M      | C     | X       | I       |
| 2       | MM     | CC    | XX      | II      |
| 3       | MMM    | CCC   | XXX     | III     |
| 4       |        | CD    | XL      | IV      |
| 5       |        | D     | L       | V       |
| 6       |        | DC    | LX      | VI      |
| 7       |        | DCC   | LXX     | VII     |
| 8       |        | DCCC  | LXXX    | VIII    |
| 9       |        | CM    | XC      | IX      |

Заметим, что:

- Обозначения для 4, 9, 40, 90, 400 и 900 используют так называемую *инвертированную запись*, когда первый символ вычитается из последующего (например, для 40 (XL) X (10) вычитается из L (50)). Это **единственные** случаи инвертированной записи.
- Число, содержащее несколько десятичных цифр, строится конкатенацией римской записи каждой цифры, слева направо от большей к меньшей.
- Если в каком-то разряде стоит 0, то в соответствующем месте ничего не приписывается.
- Наибольшее число, которое может быть представлено в римской записи, это 3,999 (MMMCMXCIX).

### Input

Первая строка входных данных содержит одно целое число  $T$  — количество тестовых примеров ( $1 \leq T \leq 77777$ ).

Каждая из последующих  $T$  строк содержит один тестовый пример — слово  $s$ , состоящее из букв I, V, X, L, C, D и M ( $1 \leq |s| \leq 15$ ).

### Output

Для каждого тестового примера выведите одно целое число — количество анаграмм заданной строки, являющихся корректными римскими числами.

### Example

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 2                | 1                 |
| IC               | 1                 |
| XV               |                   |

## Problem D. Графопостроитель

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 2 секунды  
Memory limit: 512 мегабайт

Графопостроитель строит чертёж на бумаге.

Графопостроитель работает следующим образом:

- За один шаг рисуется один отрезок.
- Если новый отрезок пересекается с каким-то из существующих отрезков, то он пересекается ровно по одной точке и точка пересечения не может быть внутренней точкой ни для какого из отрезков.

Два отрезка считаются принадлежащими одной *фигуре*, если отрезки соединены прямо или опосредованно, то есть существует последовательность отрезков с одним из этих двух отрезков как первым элементом, вторым — как последним, обладающая следующим свойством: любые два отрезка, соседние в этой последовательности, имеют общую вершину.

После каждого шага, тем самым, на бумаге нарисовано несколько фигур.

Назовём фигуру *забавной*, если она может быть нарисована за один росчерк, то есть можно нарисовать фигуру целиком без отрыва карандаша от бумаги и без рисования какого-либо участка ненулевой длины дважды. При этом **не имеет значения**, сколько раз были посещены концы отрезков.

Ваша задача — вывести количество забавных фигур на бумаге после каждого шага графопостроителя.

### Input

Первая строка входных данных содержит одно целое число  $N$  — количество шагов графопостроителя ( $1 \leq N \leq 2 \cdot 10^5$ ).

$i$ -я из последующих  $N$  строк задаёт  $i$ -й шаг и содержит четыре целых числа  $x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2}$  — координаты концов отрезка, начерченного графопостроителем на  $i$ -м шаге ( $0 \leq x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2} \leq 10^9$ , длина отрезка строго больше нуля).

Гарантируется, что для каждой пары отрезков их пересечение или пусто, или состоит из одной точки, которая является концом для обоих отрезков.

### Output

Выведите  $N$  строк. В  $i$ -й строке выведите одно целое число — количество забавных фигур на бумаге после  $i$ -го шага.

### Example

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

## Problem E. Деление на провинции

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 1 секунда  
Memory limit: 512 мегабайт

Недавно в Байтландской библиотеке была найдена старинная книга с историями и легендами о полумифической мудрой королеве Байтике и трёх её детях. Одна из историй рассказывает, что королевство Байтики имело форму трапеции ненулевой площади со сторонами  $a$ ,  $b$ ,  $c$  и  $d$ . Когда дети выросли, Байтике решила разбить королевство на три треугольные провинции, поставить детей губернаторами и посмотреть, как те справляются со своими обязанностями, чтобы потом выбрать преемницу или преемника среди них.

Чтобы избежать конфликтов между детьми относительно размера и формы провинций, мудрая королева решила, что все части должны быть **равны** как треугольники, провела два отрезка на карте — и дети получили свои провинции.

Вам интересно, правдива ли эта легенда. Конечно, у вас не получится поехать в археологическую экспедицию прямо во время конкурса, но вы можете проверить, возможна ли такая история с заданными значениями  $a$ ,  $b$ ,  $c$  и  $d$ .

### Input

Вход содержит четыре целых числа  $a$ ,  $b$ ,  $c$  и  $d$ , каждое число на новой строке ( $1 \leq a, b, c, d \leq 1000$ ).

### Output

Если трапеция с заданными сторонами существует и королева может разделить королевство на три равных треугольника, выведите 1. Иначе выведите 0.

### Examples

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 6<br>2<br>3<br>4 | 1                 |
| 1<br>2<br>4<br>8 | 0                 |

## Problem F. Если лодку украли...

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 1 секунда  
Memory limit: 512 мегабайт

У знаменитого изобретателя Леонардо украли лодку. Но ему надо перевезти на другой берег свою библиотеку. Леонардо решил склотить вместо лодки плавучий ящик в форме прямоугольного параллелепипеда. Быстро написав трактат «О прямоугольных лодках», он определил, что длина, ширина и высота, измеренные в сантиметрах, должны быть целыми числами, ширина лодки должна быть в точности равна  $1/3$  от длины, а высота должна быть в точности равна  $1/2$  от ширины.

Сам ящик (вне зависимости от размеров — всё же не зря Леонардо считают гением) весит 50 килограммов. Библиотека Леонардо весит  $W$  килограммов, и он хочет перевезти её за один раз. Напоминаем, что лодка с грузом плавает, если суммарный вес лодки и груза в точности равен весу воды, вытесненной лодкой. Верхняя сторона ящика и поверхность воды параллельны. Леонардо хочет, чтобы расстояние от верхней стороны ящика до ватерлинии (линии контакта с поверхностью воды) была не менее 30 сантиметров.

Напишите программу, которая найдёт минимальное значение длины, ширины и высоты ящика, который должен сделать Леонардо. Напоминаем, что 1000 кубических сантиметров воды весят 1 килограмм.

### Input

Входные данные состоят из одного целого числа  $W$  ( $1 \leq W \leq 1000$ ) — суммарный вес библиотеки Леонардо в килограммах.

### Output

Выведите три целых числа в одной строке — минимальную длину, ширину и высоту ящика, в соответствующем порядке.

### Examples

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 42               | 216 72 36         |
| 128              | 240 80 40         |

## Problem G. Ё-софт и планы апгрейда

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 2 секунды  
Memory limit: 512 мегабайт

Компания «Ё-софт» планирует массовый апгрейд. Руководство компании рассматривает варианты покупки новых серверов прямо с выставки.

На выставке Server Expo представлены  $N$  серверов. Каждый сервер стоит  $p_i$  килофранклинов и имеет экономический эффект  $V_i$ . На выставке есть только один сервер каждого типа.

СЕО компании может потратить на апгрейд не более  $K$  килофранклинов. Чтобы определить точную сумму финансирования, он хочет для каждого целого положительного числа  $i \leq K$  определить максимальный суммарный экономический эффект, который «Ё-софт» может получить, затратив на покупку серверов с Server Expo не более  $i$  килофранклинов.

### Input

Первая строка входных данных содержит два целых числа  $N$  и  $K$  — количество серверов на выставке и максимальное финансирование в килофранклинах, соответственно ( $1 \leq N \leq 10^6$ ,  $1 \leq K \leq 5 \cdot 10^4$ ). Каждая из последующих  $N$  строк содержит по два целых числа  $p_i$  и  $V_i$  — цену очередного сервера и его экономический эффект ( $0 \leq p_i \leq 300$ ,  $0 \leq V_i \leq 10^9$ ).

### Output

Выведите строку, содержащую  $K$  целых чисел.  $i$ -е из этих чисел должно задавать максимальный суммарный экономический эффект, который «Ё-софт» может получить, выделив не более  $i$  килофранклинов на покупку серверов.

### Example

| стандартный ввод                            | стандартный вывод                     |
|---|---------------------------------------|
| 5 10<br>4 2<br>1 49<br>4 25<br>2 73<br>3 60 | 49 73 122 122 133 182 182 182 182 207 |

## Problem H. «Железные» секреты

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 4 секунды  
Memory limit: 512 мегабайт

Компания Thinkbox планирует построить новую фабрику в джунглях, которая будет выпускать новый продукт компании — беспроводные клавиатуры с миниджойстиком, подсветкой клавиатуры и полугодовым временем работы между зарядками. Успех этого продукта гарантирует компании Thinkbox длительное доминирование на рынке.

Чтобы избежать утечек информации к конкурентам, фабрика будет расположена *под лесом* и будет иметь форму квадрата. Лес на данной территории может быть представлен как матрица  $R \times C$ , для каждой клетки которой определён вид деревьев, доминирующих в данной клетке. Виды деревьев заданы строчными латинскими буквами. Территория фабрики должна быть квадратом  $F \times F$ , где  $1 \leq F \leq \min(R, C)$ , стороны которого проходят точно по границам между клетками леса.

Подлесное расположение фабрики обозначает, что фабрика должна быть максимально безопасно расположена. Сейчас конкуренты могут легко сделать фотографию участка леса, под которым расположена компания, используя дроны. Поэтому СЕО компании хочет выбрать конфигурацию типов деревьев над фабрикой таким образом, что в лесу будет как минимум  $K$  квадратов с точно такой же стороной и одинаковым расположением доминирующих деревьев (иначе говоря, для каждого из  $K$  квадратов существует параллельный перенос, который при совмещении этих квадратов даёт полное совпадение всех обозначающих типы букв). Квадраты при этом могут перекрываться.

Подлесное расположение фабрики обозначает также, что фабрика должна быть построена «на халяву», но если такое невозможно — то с минимумом дополнительных затрат. Налог на постройку фабрики не зависит от длины стороны квадрата, то есть большая длина  $F$  обозначает меньшие относительные затраты на квадратный метр.

Вам дана карта леса и значение коэффициента секретности  $K$ . Найдите максимальное значение стороны квадрата  $F$  для постройки фабрики, или  $-1$ , если требование секретности выполнить не удастся.

### Input

Первая строка входных данных содержит три целых числа  $R$ ,  $C$  и  $K$  ( $1 \leq R, C \leq 600$ ,  $2 \leq K \leq R \cdot C$ ) — количество строк и столбцов в матрице, задающей лес, и коэффициент секретности (число одинаково выглядящих сверху квадратов).

Далее следуют  $R$  строк, каждая из которых содержит  $C$  строчных латинских букв — карту леса. Клетки с одним и тем же доминирующим типом растительности обозначены одинаковыми буквами, клетки с разными доминирующими типами — разными.

### Output

Выведите одно целое число — максимальную длину квадрата  $F$ . Если требование секретности выполнить не удастся и  $K$  одинаково выглядящих квадратов не существует, выведите  $-1$ .

### Examples

| стандартный ввод                           | стандартный вывод |
|--|-------------------|
| 4 5 2<br>baacd<br>bbaab<br>bbbaa<br>bbbbbb | 3                 |
| 3 3 5<br>cbb<br>dxb<br>xxx                 | -1                |

## Note

В первом тестовом примере квадраты  $3 \times 3$  с верхними левыми углами  $(0,0)$  и  $(1,1)$  имеют один и тот же вид сверху:

baa

bba

bbb

## Problem I. Зачёт по механике

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 1 секунда  
Memory limit: 512 мегабайт

Для зачёта по механике для студентов первого курса МФТИ была предложена следующая задача:

Дан идеальный резиновый шарик, который катается без трения по жёлобу. Жёлоб представляет собой бесконечную прямую, мяч стартует в точке 0 и движется направо (в направлении положительных координат) со скоростью 1, то есть через одну секунду он будет в позиции 1, через две — в позиции 2 и так далее.

Посередине жёлоба установлены  $N$  стенок.  $i$ -я из этих стенок установлена в точке с координатой  $X_i$  и обладает прочностью  $T_i$ . Когда шарик удаляется об стенку, происходит идеально упругий удар и шарик начинает двигаться в противоположном направлении с той же самой скоростью 1. После соударения прочность стены уменьшается на 1. Как только прочность стены станет равной 0, стена исчезает.

Требуется найти, сколько времени пройдёт с момента запуска шарика до последнего столкновения со стенкой.

Преподаватель поручил Вам сгенерировать варианты ответов. По заданным координатам стенок и их начальной толщине найдите ответ к задаче. Так как ответ может быть очень большим, от вас требуется вывести остаток от его деления на 998 244 353.

### Input

Первая строка входных данных содержит одно целое число  $N$  ( $1 \leq N \leq 50\,000$ ) — количество стенок.

Каждая из последующих  $N$  строк содержит по два целых числа  $X_i$  и  $T_i$  ( $-10^9 \leq X_i \leq 10^9$ ,  $1 \leq T_i \leq 10^9$ ) — положение  $i$ -й стенки и начальное значение её прочности, соответственно.

Гарантируется, что все  $X_i$  попарно различны, что в точке с координатой 0 стенки нет и что шарик гарантированно ударится как минимум в одну стенку.

### Output

Выведите одно целое число — остаток от деления времени в секундах, прошедшего от момента начала движения до последнего удара шарика об стенку, на 998 244 353.

### Examples

| стандартный ввод                             | стандартный вывод |
|--|-------------------|
| 3<br>2 7<br>-2 1<br>-4 1                     | 22                |
| 2<br>10000000 10836006<br>-10000000 87654321 | 535438694         |

## Problem J. Играем в угадайку?

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 1 секунда  
Memory limit: 512 мегабайт

*Это интерактивная задача*

Программа жюри загадала целое число  $N$  от 1 до  $10^6$ , и ваша задача — отгадать его. Для этого вы можете работать со специальным регистром  $X$ . В начале игры  $X = N$ .

Вы можете задавать запросы вида  $? d$ , где  $d$  — целое число между 0 и  $10^6$ . Программа жюри **заменяет** текущее значение  $X$  значением  $X + d$ . Если  $X + d > 2^{20}$ , то вы проиграли. В противном случае программа жюри отвечает 1, если  $X + d$  — квадрат целого числа, и 0 в противном случае. Вы можете задать не более 2023 запросов.

Если вы считаете, что у вас хватает информации, чтобы назвать  $N$ , вы сообщаете ответ программе жюри в формате  $! N$ . Если вы угадали, решение зачтено. Это действие **не** считается запросом.

Гарантируется, что интерактор не является адаптивным, то есть значение  $N$  определяется перед началом взаимодействия и не меняется в процессе.

### Interaction Protocol

Взаимодействие начинается ваша программа, выводя запрос. Запрос имеет вид  $? d$  ( $0 \leq d \leq 10^6$ ). В ответ на это программа жюри выведет 0 или 1 в зависимости от нового значения регистра  $X$  (1, если для некоторого целого  $q$   $q \cdot q = X$ , и 0 в противном случае). После чего вы задаёте следующий вопрос и так далее.

Если вы хотите вывести ответ, выводите его в виде  $! N$ .

Не забывайте сбрасывать буфер вывода после каждого запроса или вывода ответа, иначе ваша программа может получить вердикт Idleness Limit.

### Example

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 1                | ? 2               |
| 0                | ? 8               |
| 1                | ? 8               |
| 1                | ? 75              |
|                  | ! 7               |

## Problem K. Йо-Йо

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 1 секунда  
Memory limit: 512 мегабайт

Игрушечный магнат Билл Геймер решил стать монополистом по производству игрушек йо-йо.

Всего есть  $N$  различных типов фабрик.

Количество выпущенных йо-йо  $f_0$  увеличивается каждую секунду сначала в  $C$  раз, затем к результату прибавляется количество йо-йо, выпущенных фабриками первого (базового) типа. Фабрика первого типа производит  $K_1$  йо-йо каждую секунду, так что если в момент времени  $t$  у Билла есть  $f_0$  йо-йо и  $f_1$  фабрик первого типа, в момент времени  $t + 1$  у него будет  $C \cdot f_0 + K_1 \cdot f_1$  йо-йо.

Фабрики типа  $i > 1$  не производят йо-йо напрямую, они производят другие фабрики.

Количество фабрик типа  $i$  также увеличивается каждую секунду сначала в  $C$  раз, затем к результату прибавляется количество фабрик типа  $i$ , произведённых фабриками типа  $i + 1$ .

Фабрика типа  $i + 1$  (где  $i > 0$ ) каждую секунду производит  $(i + 1) \cdot K_{i+1}$  фабрик типа  $i$ , так что если в момент времени  $t$  у Билла есть  $f_i$  фабрик типа  $i$  и  $f_{i+1}$  фабрик типа  $i + 1$ , то в момент времени  $t + 1$  у него будет  $C \cdot f_i + (i + 1) \cdot K_{i+1} \cdot f_{i+1}$  фабрик типа  $i$ .

Вам дано количество йо-йо ( $f_0$ ) и фабрик каждого типа у Билла на момент времени  $t = 0$ , а также значения  $C$  и  $K_i$ . Определите остаток от деления количества йо-йо в момент времени  $t = T$  на 998 244 353.

### Input

Первая строка входных данных содержит три целых числа  $N$  ( $1 \leq N \leq 2 \cdot 10^5$ ) — количество типов продукции (йо-йо и  $N - 1$  тип фабрик), момент времени  $T$ , на который надо подсчитать ответ ( $1 \leq T \leq 10^9$ ) и константу  $C$  ( $1 \leq C \leq 10^9$ ).

Вторая строка содержит  $N$  целых чисел — начальные значения  $f_0, f_1, \dots, f_{N-1}$ , заданные в порядке возрастания индекса ( $0 \leq f_i \leq 10^9$ ). Третья строка содержит  $N$  целых чисел  $K_0, K_1, \dots, K_{N-1}$ , ( $0 \leq K_i \leq 10^9$ ).

### Output

Выведите одно целое число — остаток от деления количества йо-йо после  $T$ -й секунды на 998 244 353.

### Example

| стандартный ввод        | стандартный вывод |
|-------------------------|-------------------|
| 3 3 2<br>1 1 1<br>1 1 1 | 32                |

## Problem L. Крутые номера

Input file: стандартный ввод  
Output file: стандартный вывод  
Time limit: 1 секунда  
Memory limit: 512 мегабайт

Автомобильные номерные знаки в Берляндии имеют следующий формат:

letter letter digit digit digit letter letter digit digit

Здесь letter обозначает строчную латинскую букву, а digit — десятичную цифру. Допустима любая комбинация букв и цифр в рамках этих ограничений.

«Крутыми» номерами в Берляндии считаются номера, в которых и цифровая часть, и буквенная являются палиндромами. Например, номер **ab123ba21** — крутой, так как и **abba**, и **12321** — палиндромы, а номер **aa111aa22** — нет, так как число **11122** палиндромом не является.

Дорожная инспекция Байтландии засекла нарушителя на крутом автомобиле с крутым номером. Номер записался на камеру, однако есть проблема: из-за высокой скорости распознать символ на некоторых местах не получается. Соответствующие буквы (или цифры) в строке номера заменены звёздочкой (\*), одна звёздочка соответствует одной цифре или букве в зависимости от номера позиции.

Вам дана распознанная строка. Найдите количество возможных крутых номеров, которые могли быть на фотографии.

### Input

Входные данные состоят из одной строки длины 9. Символы на позициях 0,1,5,6 — или строчные латинские буквы, или '\*'; символы на остальных позициях — или цифры, или '\*'.

### Output

Выведите одно целое число — количество крутых номеров, которые соответствуют данным с фотографии.

### Examples

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| a*****b**        | 0                 |
| *****            | 676000            |
| aa777aa77        | 1                 |