Problem A. RPG

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

В RPG "Crazy Elves" кроме обычных *skill point*, начисляемых с каждым уровнем, полученным персонажем, присутствуют ещё и *специальные умения*.

Система skill points работает как обычно: первоначально значения всех skills у каждого персонажа равен нулю. Каждый из полученных skill points можно прибавить к значению какого-то из skill-ов (при этом разбивать skill points на части нельзя — значения skills являются целыми числами). Уровень персонажа (и, соответственно, количество заработанных skill points) в игре может быть сколь угодно большим.

Специальные умения характеризуются набором ограничений на значения определённых skill-ов. Ограничения могут быть как сверху (тогда значение skill-а не должно превосходить параметра специального умения), так и снизу (тогда значение skill-а не должно быть меньше этого параметра). В момент, когда все ограничения, соответствующие данному специальному умению, впервые выполняются, персонаж получает соответствующее специальное умение.

Если персонаж уже получил специальное умение, то оно остаётся с ним вне зависимости от того, как в дальнейшем изменяются skill points. Переназначать уже назначенные skill points нельзя.

Требуется выяснить, возможно ли выбирать skills для улучшения таким образом, чтобы в какой-то момент персонаж получил все специальные умения.

Input

В первой строке входного файла заданы два целых числа (M, N), где M — это количество специальных умений ($1 \le M \le 100$), а N — количество skills у персонажа ($1 \le N \le 100$). Как специальные умения, так и skills занумерованы с единицы.

Далее задаётся M групп условий получения специального умения.

Первая строка группы условий содержит целое число K_i ($0 \le K_i \le 100$), где K_i — количество «примитивных» условий, которые должны выполняться для того, чтобы получить i-е специальное умение.

Последующие K_i строк группы условий задают ограничения на значения skill-ов. Каждое ограничение состоит из $s_{i,j}$ — номера задействованного в ограничении скилла, условия $cond_{i,j}$ и значения $t_{i,j}$. Если условие равно "`<=`", то значение скилла $s_{i,j}$ должно быть меньше или равно, чем $t_{i,j}$, а если равно "`>=`" —- то значение скилла $s_{i,j}$ должно быть больше или равно $t_{i,j}$.

Output

Выведите "`Yes`", если персонаж может получить все специальные умения, и "`No`" в противном случае.

Examples

Standard input	Standard output
2 2 2 1 >= 3 2 <= 5 2 1 >= 4 2 >= 3	Yes
2 2 2 1 >= 5 2 >= 5 2 1 <= 4 2 <= 3	Yes
2 2 2 1 >= 3 2 <= 3 2 1 <= 2 2 >= 5	No
1 2 2 1 <= 10 1 >= 15	No
5 5 3 2 <= 1 3 <= 1 4 <= 1 4 2 >= 2 3 <= 1 4 <= 1 5 <= 1 3 3 >= 2 4 <= 1 5 <= 1 2 4 >= 2 5 <= 1 1 5 >= 2	Yes

Problem B. Integer in integer

Input file:	Standard input
Output file:	Standard output
Time limit:	3 seconds
Memory limit:	256 mebibytes

По заданному интервалу $[A, B]$ и целому числу C определите, сколько раз десятичная запись числа C встречается в десятичной записи чисел интервала в качестве строки.

Например, строка 33 встречается в строке 333 дважды, а в строке 334 один раз. Таким образом, 33 среди целых чисел интервала $[333, 334]$ встречается 3 раза.

Input

Входной файл состоит из трёх целых чисел A, B, C ($0 \le A \le B \le 10^{10000}$, $0 \le C \le 10^{500}$).

Output

Выведите количество вхождений C как строки в числа интервала по модулю $10^9 + 7$.

Examples

Standard input	Standard output
1 3 2	1
333 334 33	3
0 10 0	2

Problem C. Card Game

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Трус, Балбес и Бывалый играют в карточную игру.

Карты в этой игре делятся на три вида: «гильотина», 12 «картинок» и 6 карт действия. На лицевой стороне каждой «картинки» и каждой карты действия написано некоторое целое число.

Перед началом игры, участники выкладывают все «картинки» лицом вверх в одну линию на стол. Справа от них кладётся «гильотина». После чего каждый из участников получает по две карты действия. При этом карты действия, полученные участником, кладутся перед ним лицом вверх, так что у всех участников есть полная информация.

Далее игра начинается. Игроки делают ходы по очереди. Бывалый ходит первым, Балбес — вторым, Трус — третьим и далее по циклу. Ход делается на две части — действие и снос (причём в рамках одного хода действие всегда предваряет снос). Изначальное количество очков у каждого из игроков равно нулю.

В первой части, если у участника есть какие-то карты действия, он может сыграть одной из них. Пусть на этой карте написано число y. Если на столе менее y картинок, ничего не происходит. В противном случае отсчитывается y-я картинка, начиная от гильотины, и переносится на позицию непосредственно перед гильотиной (при этом картинки, находившиеся впереди неё, сдвигаются на одну позицию назад). При этом использовать карту действия не обязательно. Сыгранная карта действия сбрасывается с руки и в дальнейшем в игре не участвует.

Во второй части участник убирает картинку, расположенную непосредственно перед гильотиной, и прибавляет себе количество очков, изображённое на лицевой стороне этой картинки. Убранная картинка в дальнейшем в игре не участвует. Эта часть хода является обязательной.

Игра заканчивается, когда все картинки убраны со стола.

Игроки действуют согласно следующей стратегии:

- Каждый игрок предполагает, что цель игры остальных — максимизировать сумму набранных очков.
- Бывалый и Трус играют с целью максимизировать сумму набранных очков. Балбес же играет с целью минимизировать сумму очков, набранных Бывалым. При этом Балбес знает, что Бывалый и Трус действуют, ошибочно считая, что Балбес тоже хочет максимизировать сумму набранных очков.
- Если вышеописанные приоритеты оставляют выбор, то игроки предпочитают иметь по окончании каждого хода как можно больше карт действия на руке. Если выбор всё равно остаётся, то игроки предпочитают максимизировать суммарное значение оставшихся на руке карт действия.

Ваша задача — вычислить итоговый результат для каждого игрока.

Input

В первой строке входного файла заданы 12 целых чисел $x_{12}, x_{11}, \ldots, x_1$ ($-2 \le x_i \le 5$). x_i — число, записанное на i-й картинке, считая от гильотины. Следующие 3 строки задают числа $y_1, \ldots, y_6$

($1 \le y_j \le 4$), записанные на картах действия. В первой из них заданы карты Бывалого y_1, y_2, во второй — карты Балбеса y_3, y_4 и в третьей — карты Труса y_5 и y_6.

Output

Выведите три целых числа, разделённых пробелами — результат Бывалого, Балбеса и Труса соответственно.

Example

Standard input	Standard output
3 2 1 3 2 1 3 2 1 3 2 1 1 1 1 1 1 1	4 8 12
4 4 4 3 3 3 2 2 2 1 1 1 1 2 2 3 3 4	8 10 12
0 0 0 0 0 0 0 -2 0 -2 5 0 1 1 4 1 1 1	-2 -2 5

Problem D. Epidemy (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	3 seconds
Memory limit:	256 mebibytes

Правительство объявило чрезвычайную ситуацию в связи с угрозой эпидемии в государстве, состоящем из n островов, пронумерованных последовательными числами от 1 до n. Между некоторыми парами островов курсируют океанские лайнеры. Правительство предложило открыть карантинные станции на некоторых островах, чтобы препятствовать распространению эпидемии. При этом для каждого лайнера либо на острове отправления, либо на острове прибытия должна быть карантинная станция.

Однако из-за бюджетных ограничений правительство может построить не более K станций.

Ваша задача — выяснить, сможет ли правительство реализовать свой план, и, если сможет, определить наименьшее количество карантинных станций, которые должны быть построены.

Input

Входной файл начинается строкой, содержащей три целых числа N ($2 \le N \le 3{,}000$), M ($1 \le M \le 30{,}000$) и K ($1 \le K \le 32$).

Каждая из последующих M строк содержит по два целых числа a_i ($1 \le a_i \le N$) и b_i ($1 \le b_i \le N$) — номера островов, между которыми курсирует i-й лайнер. Гарантируется, что $a_i \ne b_i$ и что не существует двух различных лайнеров, курсирующих между одной и той же парой островов.

Output

Если план правительства реализовать не удастся, выведите "`Impossible`". Иначе выведите минимальное количество карантинных станций, которое требуется построить для реализации плана правительства.

Examples

Standard input	Standard output
3 3 2 1 2 2 3 3 1	2
3 3 1 1 2 2 3 3 1	Impossible
7 6 5 1 3 2 4 3 5 4 6 5 7 6 2	4
10 10 10 1 2 1 3 1 4 1 5 2 3 2 4 2 5 3 4 3 5 4 5	4

Problem E. MST (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	5 seconds
Memory limit:	256 mebibytes

Задан неориентированный граф G с n вершинами и m рёбрами, занумерованными натуральными числами от 1 до m. Каждому ребру присвоено целое неотрицательное значение — его цена.

Пусть G_i — граф, который строится удалением i-го ребра из графа G. Требуется найти суммарную стоимость минимального остовного дерева графа G_i для каждого i.

Input

Первая строка входного файла содержит два целых числа n ($2 \le n \le 10^5$) и m ($1 \le m \le 2 \cdot 10^5$). n — количество вершин графа и m — количество рёбер.

Далее следуют m строк, каждая из которых содержит три целых числа a_i ($1 \le a_i \le n$), b_i ($1 \le b_i \le n$) и w_i ($0 \le w_i \le 10^6$) — номера вершин, соединяемых ребром и цена ребра, соответственно.

Гарантируется, что граф не содержит петель и кратных рёбер.

Output

Для каждого i в i-й строке выведите стоимость минимального остовного дерева. Если в графе G_i нет остовных деревьев, для соответствующего i выводите -1.

Examples

Standard input	Standard output
4 6 1 2 2 1 3 6 1 4 3 2 3 1 2 4 4 3 4 5	8 6 7 10 6 6
4 4 1 2 1 1 3 10 2 3 100 3 4 1000	1110 1101 1011 -1
7 10 1 2 1 1 3 2 2 3 3 2 4 4 2 5 5 3 6 6 3 7 7 4 5 8 5 6 9 6 7 10	27 26 25 29 28 28 28 25 25 25
3 1 1 3 999	-1

Problem F. Molecules (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	8 seconds
Memory limit:	256 mebibytes

В рамках проекта по исследованию движения молекул белка требуется вычислить средние расстояния между всеми парами молекул в некоторый момент времени и построить гистограмму расстояний.

Для облегчения подсчётов координаты молекул аппроксимируются узлами квадратной сетки $N \times N$.

Input

Первая строка входного файла содержит длину стороны сетки N ($1 \le N \le 1024$). Каждая из последующих N строк содержит по N целых чисел C_{xy} ($0 \le C_{xy} \le 9$) — количество молекул, аппроксимированных узлом с координатами (x, y). Гарантируется, что сумма всех C_{ij} больше единицы.

Output

В первой строке выведите среднее расстояние между двумя молекулами с абсолютной или относительной погрешностью не более 10^{-8}. Далее выведите данные для построения гистограммы Каждая строка содержит два целых числа — квадрат расстояния и количество пар молекул, расположенных на этом расстоянии. Строки должны быть упорядочены по возрастанию квадрата расстояния. Если различных расстояний более 10^4, выведите первые 10^4 расстояний. Если на данном расстоянии молекул нет, соответствующие этому расстоянию данные выводить не надо.

Example

Standard input	Standard output
2 1 0 0 1	1.4142135624 2 1
3 1 1 1 1 1 1 1 1 1	1.6349751825 1 12 2 8 4 6 5 8 8 2
5 0 1 2 3 4 5 6 7 8 9 1 2 3 2 1 0 0 0 0 0 0 0 0 0 1	1.8589994382 0 125 1 379 2 232 4 186 5 200 8 27 9 111 10 98 13 21 16 50 17 37 18 6 20 7 25 6

Problem G. Carrier Pigeon (Division 1 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	2 seconds
Memory limit:	256 mebibytes

В компанию «ЭкспрессТурманДоставка», занимающуюся рассылкой голубиной почты, поступил заказ на переправку f писем из города s в город t.

Всего в компании M голубей. i-й из них может нести груз из u писем из города s_i в город t_i (но не в обратном направлении!), при этом стоимость транспортировки равна ua_i, если u не превосходит d_i, иначе она равна $d_i a_i + (u - d_i)b_i$.

Грузоподъёмность каждого голубя не ограничена. Если голубь перевозит почту в несколько приёмов, то стоимость всё равно определяется, исходя из суммарного количества перевезённой почты. Всего в стране N городов, занумерованных целыми числами от 0 до $N - 1$.

Требуется минимизировать стоимость доставки заказа.

Input

Первая строка входного файла содержит пять целых чисел: N ($2 \le N \le 100$), M ($1 \le M \le 1{,}000$), s ($0 \le s \le N - 1$), t ($0 \le t \le N - 1$) и f ($1 \le f \le 200$). Гарантируется, что $s \ne t$.

i-я из последующих M строк содержит описание параметров i-го голубя — пять целых чисел s_i ($0 \le s_i \le N - 1$), t_i ($0 \le t_i \le N - 1$), a_i ($0 \le a_i \le 1{,}000$), b_i ($0 \le b_i \le 1{,}000$) и d_i ($1 \le d_i \le 200$). Из всех пар a_i и b_i неравенство $a_i < b_i$ выполняется не более, чем для одной пары, для остальных же $a_i > b_i$.

Output

Выведите минимальную стоимость доставки f писем из города s в город t. Если доставить заказ невозможно, выведите "Impossible".

Examples

Standard input	Standard output
2 2 0 1 5 0 1 3 0 3 0 1 2 1 6	9
4 4 0 3 5 0 1 3 0 3 1 3 3 0 3 0 2 2 1 6 2 3 2 1 6	18
2 1 0 1 1 1 0 1 0 1	Impossible
2 2 0 1 2 0 1 5 1 2 0 1 6 3 1	9
3 3 0 2 4 0 2 3 4 2 0 1 4 1 3 1 2 3 1 1	14

Problem H. Circles

Input file:	Standard input
Output file:	Standard output
Time limit:	2 seconds
Memory limit:	256 mebibytes

В трёхмерном пространстве задны две окружности радиуса 1. Проверьте, зацеплены ли они.

Input

Первая строка входного файла содержит три вещественных числа ($-3 \le x_i,\ y_i,\ z_i \le 3$) — координаты центра первой окружности. Вторая строка содержит шесть вещественных чисел ($-1 \le X_{i,j},\ Y_{i,j},\ Z_{i,j} \le 1$). Единичный вектор ($X_{1,1}$, $Y_{1,1}$, $Z_{1,1}$) направлен из центра к окружности. Второй единичный вектор ($X_{1,2}$, $Y_{1,2}$, $Z_{1,2}$) также направлен из центра к окружности и ортогонален первому.

Третья и четвёртая строка задают вторую окружность в аналогичном формате.

Гарантируется, что ни в одном из тестовых примеров окружности не касаются.

Output

Если окружности зацеплены, выведите "YES". Иначе выведите "NO".

Example

Standard input	Standard output
0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0 0.0 1.0 0.0 0.5 1.0 0.0 0.0 0.0 0.0 1.0	YES
0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0 0.0 0.0 3.0 0.0 0.0 1.0 0.0 -1.0 0.0 0.0	NO
1.2 2.3 -0.5 1.0 0.0 0.0 0.0 1.0 0.0 1.1 2.3 -0.4 1.0 0.0 0.0 0.0 0.70710678 0.70710678	YES
1.2 2.3 -0.5 1.0 0.0 0.0 0.0 1.0 0.0 1.1 2.7 -0.1 1.0 0.0 0.0 0.0 0.70710678 0.70710678	NO

Problem I. Queries

Input file:	Standard input
Output file:	Standard output
Time limit:	4 seconds
Memory limit:	256 mebibytes

Изначально Вам заданы N пустых массивов $t_1, \ldots, t_n$. Сначала выполняются M запросов на заполнение:

- Добавить объект, равный v, в массивы t_i $(a \le i \le b)$;

Далее выполняются Q запросов на вывод.

- Вывести j-е число в отсортированном по возрастанию массиве, составленном из элементов массивов t_i $(x \le i \le y)$.

Ваша задача — промоделировать данный процесс.

Input

В первой строке входного файла заданы три целых числа N $(1 \le N \le 10^9)$, M $(1 \le M \le 10^5)$ и Q $(1 \le Q \le 10^5)$. Каждая из последующих M строк содержит по три целых числа a_i, b_i и v_i $(1 \le a_i \le b_i \le N, 1 \le v_i \le 10^9)$ — параметры запросов на заполнение.

Далее заданы запросы на вывод — каждая из последующих Q строк содержит список запросов на вывод. Каждый запрос описывается тремя целыми числами x_i, y_i и j_i $(1 \le x_i \le y_i \le N,\ 1 \le j_i \le \sum_{x_i \le k \le y_i} |t_k|)$.

Output

Выведите результат выполнения запросов на вывод в порядке их задания во входном файле. Ответ на каждый запрос выводить в новой строке.

Examples

Standard input	Standard output
5 4 1 1 5 1 1 1 3 4 5 1 3 4 2 1 3 4	2
10 4 4 1 4 11 2 3 22 6 9 33 8 9 44 1 1 1 4 5 1 4 6 2 1 10 12	11 11 33 44

Note

В первом примере после выполнения M-го запроса на заполнение, массивы t_i выглядят так:

$[1, 3]$, $[1]$, $[1, 2]$, $[1, 1, 2]$, $[1, 1]$

Последовательность отсортированных значений в массивах t_1, t_2 и t_3 — $[1, 1, 1, 2, 3]$. 4-й её элемент равен 2.

Problem J. Restoring the Flows

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Заданы несколько ориентированных графов. Для каждого ребра $e(u, v)$ определено значение потока из u в v. Пример графа с потоками приведён на следующей иллюстрации:

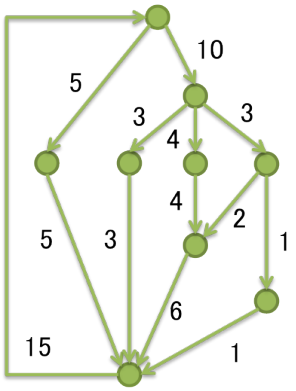
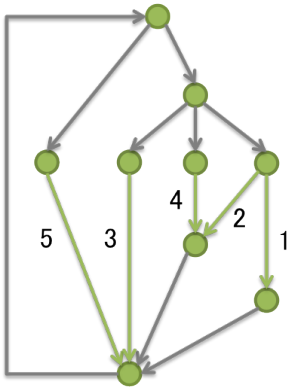


Рисунок соответствует первому примеру из условия задачи.

Можно выбрать некоторое количество рёбер так, что, зная значения потока для этих рёбер, можно однозначно восстановить значение потоков на всех остальных рёбрах графа. Например, значения потоков из первого примера могут быть восстановлены по пяти рёбрам (см. рисунок ниже).



Требуется вычислить наименьшее количество рёбер, значение потока через которые надо запомнить, чтобы восстановить значения потока для всех рёбер графа. Значения потоков обладают следующими свойствами:

- Для каждой вершины суммарный входной поток равен суммарнимо выходному потоку.
- Все значения потоков неотрицательны.

Input

Первая строка входного файла содержит два целых числа N ($1 \le N \le 500$) и M ($0 \le M \le 3000$) — количество вершин и рёбер в графе.

Каждая из последующих M строк содержит по два целых числа s_i и t_i, что обозначает наличие ребра e (s_i, t_i).

Гарантируется, что выполняются следующие условия:

- $s_i \neq t_i$.
- Если $i \neq j$, то $(s_i \neq s_j)$ or $(t_i \neq t_j)$.
- Входной файл состоит из нескольких графов, каждый из которых является сильно связным.

Output

Выведите одно целое число — ответ к задаче.

Examples

Standard input	Standard output
9 13 1 2 1 3 2 9 3 4 3 5 3 6 4 9 5 7 6 7 6 8 7 9 8 9 9 1	5
7 9 1 2 1 3 2 4 3 4 4 5 4 6 5 7 6 7 7 1	3
4 4 1 2 2 1 3 4 4 3	2

Problem K. Genome (Division 2 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	2 seconds
Memory limit:	256 mebibytes

На некоторой планете все существа имеют геном, ДНК которого состоит только из двух типов нуклеотидов: 'A' и 'B'. Также цепочка ДНК должна удовлетворять следующему ограничению: в ней не должно быть двух нуклеотидов A подряд. Например, "ABBAB" — допустимая цепочка, а "BAABA" — нет. Цепочки являются ориентированными, то есть, к примеру, цепочки "ABBAB" и "BABBA" являются различными

Вычислите количество всех допустимых цепочка ДНК длины n по модулю m.

Input

Входной файл состоит из нескольких (не более 20) тестовых примеров. Каждый тестовый пример состоит из двух целых чисел, заданных в отдельной строке — длины цепочки n ($1 \le n \le 8 \cdot 10^{18}$) и модуля m ($1 \le m \le 2 \cdot 10^9$). Входной файл завершается строкой, состоящей из двух нулей, обрабатывать которую не требуется.

Output

Для каждого тестового примера в отдельной строке выведите одно целое число — ответ к задаче.

Example

Standard input	Standard output
1 10 2 10 5 10 6 8 0 0	2 3 3 5

Problem L. Encryption (Division 2 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	2 seconds
Memory limit:	256 mebibytes

Для шифровки данных Алиса и Боб используют следующую схему. Каждый байт n (8-битное беззнаковое целое) заменяется суммой первых $n+1$ простых чисел.

Ваша задача — зашифровать заданную последовательность чисел.

Input

В первой строке входного файла содержится одно целое число T ($1 \le T \le 100$) — количество тестовых примеров. Каждая из последующих T строк содержит число n_i ($0 \le n \le 255$), которое необходимо зашифровать.

Output

Для каждого целого числа в порядке появления чисел во входном файле выведите результат зашифровки.

Example

Standard input	Standard output
3 0 5 7	2 41 77

Problem M. Points in triangle (Division 2 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	2 seconds
Memory limit:	256 mebibytes

Определим целочисленную точку как упорядоченную пару (x, y), где x и y — целые числа. По заданным координатам вершин треугольника (являющихся целочисленными точками) требуется найти количество целочисленных точек, лежащих внутри (не на стороне и не в вершине) данного треугольника.

Input

Входной файл состоит из не более, чем 200 тестовых примеров. Каждый тестовый пример состоит из шести целых чисел x_1, y_1, x_2, y_2, x_3, y_3, где (x_1, y_1), (x_2, y_2), (x_3, y_3) — координаты вершин треугольника. Гарантируется, что все треугольники во входном файле имеют ненулевую площадь, и что $-15000 \le x_1, y_1, x_2, y_2, x_3, y_3 \le 15000$. Файл заканчивается тестовым примером, состоящим из шести нулей, обрабатывать который не требуется.

Output

Для каждого тестового примера в отдельной строке выведите одно число — количество целочисленных точек, лежащих строго внутри соответствующего треугольника.

Example

Standard input	Standard output
0 0 1 0 0 1 0 0 5 0 0 5 0 0 0 0 0 0	0 6

Problem N. Word Stats (Division 2 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	2 seconds
Memory limit:	256 mebibytes

В этой задаче предлагается подсчитать статистику для слов некоторого текста. Требуется выделить слова из текста, записать их строчными латинскими буквами (убрав "лишние" символы), отсортировать слова в лексикографическом порядке, после чего вывести слово (или два слова в случае чётного количества слов), стоящее в середине отсортированного списка, а также слово, встречающееся в тексте максимальное число раз (если таких слов несколько, выведите все).

Input

Входной файл состоит из не менее, чем одного, и не более, чем 500 слов. Слово определяется как последовательность символов, среди которых присутствует заглавная или строчная латинская буква, за которыми следует пробел, табуляция или перевод строки. Перед обработкой из слова выбрасываются все символы, кроме заглавных или строчных латинских букв, а заглавные буквы заменяются на соответствующие им строчные (например, слова "`w'ord`" и "`@!w12ORD`" после преобразования будут выглядеть как "`word`"). Гарантируется, что длина «необработанного» слова не превосходит 100.

Output

В первой строке выходного файла выведите слово (или два слова, в случае чётного количества слов в тексте), стоящее в середине отсортированного списка в формате, указанном в примере из условия. Например, "`My median=[easter,egg]`" или "`My median=[of]`". В случае, если количество слов в тексте чётно, слова при выводе ответа сортируются в лексикографическом порядке.

Во второй строке выходного файла выведите слово, встречающееся в тексте максимальное число раз, а также непосредственно за ним в скобках — число, показывающее, сколько раз встречается данное слово. Формат указан в примере из условия. В случае, если слов с максимальной частотой несколько, выводятся все такие слова, отсортированные в лексикографическом порядке. Например, "`My mode=[an(23),the(23)]`".

Example

Standard input	Standard output
When April with his sweet showers has pierced the drought of March to the root, and bathed every vein in such moisture as has power to bring forth the flower	My median=[moisture,of] My mode=[the(3)]
a test test1 2test2 3te3st3 4tests 123456	My median=[test,test] My mode=[test(4)]
a test test1 2test2 3te3st3 123456	My median=[test] My mode=[test(4)]