

Problem A. Фристайл

Input file:	Standard input
Output file:	Standard output
Time limit:	1 секунда
Memory limit:	256 мегабайт

Сноуборд-фристайл - один из самых зрелищных зимних видов спорта, в котором сноубордисты соревнуются, выполняя трюки различной сложности.

Фристайл является творческим и достаточно молодым видом спорта, в котором ещё не сформировалось устоявшихся подходов к тренировкам. У многих профессиональных спортсменов-фристайлеров вообще нет тренера — они обучаются новым трюкам, собираясь в сноу-парках в компании своих друзей и экспериментируя в своё удовольствие.

Двое сноубордистов разработали собственную методику совместных тренировок. Перед началом тренировки они некоторым образом формируют последовательность из N трюков, каждый из которых должен быть выполнен по одному разу в рамках любого заезда. Каждому трюку соответствует уникальный идентификатор A_i , соответствующий его уровню сложности, от самого простого — 1, до самого сложного — N .

Сноубордисты выполняют заезды по очереди и каждый следующий заезд должен быть сложнее предыдущего. Для этого сноубордист, выполняющий заезд, должен поменять порядок трюков, выполненных оппонентом в предыдущем заезде, таким образом, чтобы последовательность стала лексикографически больше — то есть для некоторой позиции k должны выполняться условия $Q_k > P_k$ и $Q_i = P_i$ при $i < k$, где Q и P — текущая и предыдущая последовательности трюков, соответственно. Для первого заезда предыдущей последовательностью является выбранная изначально.

Изменение порядка выполнения трюков относительно предыдущего заезда ограничено следующим условием: сноубордист должен выбрать одну подпоследовательность идущих подряд трюков и изменить порядок в ней на противоположный. Кроме того, выбранная подпоследовательность должна иметь длину $4 \times x + 2$ или $4 \times x + 3$, где x — любое целое неотрицательное число.

Если перед заездом сноубордист не может поменять порядок трюков таким образом, чтобы заезд был сложнее предыдущего, то тренировка заканчивается. Каждый сноубордист хочет выполнить самый последний заезд и доказать, что он катается лучше оппонента, поэтому при выборе последовательности трюков каждого заезда оба участника действуют оптимально. Вам необходимо определить, кто из участников выполнит последний заезд.

Input

В первой строке задано количество трюков в заездах N ($1 \leq N \leq 100$). Во второй строке через пробел заданы N идентификаторов трюков A_i ($1 \leq A_i \leq N$, $A_i \neq A_j$ при $i \neq j$) в той последовательности, которая была выбрана перед первым заездом.

Output

Если последний заезд будет выполнен сноубордистом, выполнявшим первый заезд, то выведите *First*, иначе выведите *Second*.

Examples

Standard input	Standard output
3 1 2 3	First
4 4 2 1 3	Second

Problem B. Очереди в кассу

Input file: Standard input
Output file: Standard output
Time limit: 1 секунда
Memory limit: 256 мегабайт

Многие поклонники горных лыж и сноуборда не могут себе позволить взять отпуск посреди зимы и отправиться отдыхать в горы. К счастью, в России есть праздничные дни в холодное время года — новогодние каникулы, 23-е февраля и 8-е марта. Во время любого из этих праздников все местные горнолыжные курорты наполняются толпами отдыхающих, приехавших покататься буквально на пару дней, а возле касс, прокатов и подъемников с самого утра вырастают огромные очереди.

Возле одного из пунктов проката до открытия собралось N человек, которые образовали две различных очереди длиной N : очередь A на получение лыжных ботинок и очередь B на получение лыж. При этом каждый человек занял по одному месту в каждой из очередей.

Обычно очереди двигаются симметрично, потому что для получения лыж и ботинок требуется одинаковое время. Но если в некоторый момент времени для одного человека одновременно наступает очередь на получение и ботинок и лыж, то он не может получить сразу и то и другое и задерживает всех остальных людей.

Чтобы избежать такой задержки, перед администратором курорта поставлена цель поменять порядок людей в очереди B таким образом, чтобы позиция любого человека в очереди A отличалась от его позиции в очереди B .

После внесения изменений позиции в очереди B для некоторых людей могут ухудшиться, что, конечно, вызовет их недовольство. Пусть позиция человека i в очереди B до изменений была равна p_i , а после изменений стала равна q_i . Тогда отклонением позиции является разность $q_i - p_i$. Считается, что позиция человека в очереди ухудшилась, если отклонение положительно.

Чтобы свести к минимуму недовольство людей в очередях администратор, хочет достичь цели таким образом, чтобы максимальное из отклонений в худшую сторону (т.е максимум из $q_i - p_i$ для всех i при которых $q_i \geq p_i$) было минимально.

Input

В первой строке задано число человек N ($1 \leq N \leq 10^5$).

В второй строке задаются N чисел A_i ($1 \leq A_i \leq N$, $A_i \neq A_j$ при $i \neq j$) — порядок людей в очереди A .

В третьей строке задаются N чисел B_i ($1 \leq B_i \leq N$, $B_i \neq B_j$ при $i \neq j$) — порядок людей в очереди B .

Output

Если решение есть, то в первой строке выведите минимально возможное максимальное отклонение позиции в худшую сторону, а во второй строке — итоговый порядок людей в очереди B . Если решений существует несколько, выведите любое.

Если решения не существует, выведите -1 .

Examples

Standard input	Standard output
2 1 2 2 1	0 2 1
3 1 2 3 3 2 1	1 2 3 1

Problem C. Хели-ски (Division 1 Only!)

Input file: Standard input
Output file: Standard output
Time limit: 1 секунда
Memory limit: 256 мегабайт

Хели-ски — это экстремальная разновидность горнолыжного спорта, в которой путь до горных склонов преодолевается на вертолете, а спуск осуществляется по нетронутой снежной целине вдали от цивилизации.

Команда фрирайдеров, пролетая на вертолете над неизведанным ранее горным хребтом заметила среди непроходимых скал большой участок целины, идеально подходящий для спуска. Погода не позволяла высадиться там в этот же день, поэтому было решено сделать несколько снимков участка, чтобы тщательнее продумать маршрут и покорить склон на следующий день.

По прибытии в лагерь фрирайдеры распечатали N фотографий, каждая из которых имеет равную высоту H и некоторую ширину W_i , разложили их в ряд на столе и отправились отдыхать. Собравшись вечером для обсуждения предстоящего маршрута, они обнаружили, что порядок фотографий был перепутан.

Чтобы приблизительно оценить размеры участка, было решено поменять порядок фотографий в ряду таким образом, чтобы после склейки соседних фотографий площадь максимального прямоугольного участка, свободного от скал, была наибольшей. При этом в процессе перестановок фотографии нельзя переворачивать, склейка осуществляется по всей высоте расположенных в соседних позициях фотографий, а стороны найденного участка должны быть параллельны сторонам фотографий.

Input

В первой строке через пробел заданы числа N и H ($1 \leq N, H \leq 10^5$).

Далее следует N описаний фотографий. В первой строке каждого описания задана ширина фотографии W_i ($1 \leq H \times \sum_{i=1}^n W_i \leq 10^5$). В следующих H строках по W_i символов задается фотография, причем символам 0 соответствуют подходящие для спуска участки, а символам 1 — участки, занятые скалами. Гарантируется, что хотя бы один участок пригоден для спуска.

Output

В первой строке необходимо вывести перестановку из N чисел от 1 до N , позволяющую получить наибольший прямоугольный участок, подходящий для спуска.

Во второй строке необходимо вывести координаты левого верхнего и правого нижнего углов найденного участка после склейки фотографий в порядке, заданном перестановкой. После склейки левый верхний угол первой фотографии имеет координаты $(1, 1)$, а правый нижний угол последней фотографии — координаты $(H, \sum_{i=1}^n W_i)$. Если существует несколько оптимальных решений, выведите любое.

Examples

Standard input	Standard output
3 4	2 1 3
4	2 3 3 5
1001	
0000	
0010	
1001	
3	
000	
010	
000	
011	
2	
00	
10	
01	
00	

Note

В 1-м тестовом примере после склейки фотографии располагаются следующим образом (искомый участок выделен жирным шрифтом):

000100100

01**0000**010

00**000**1001

011100100

Problem D. Апре-ски

Input file: Standard input
Output file: Standard output
Time limit: 1 секунда
Memory limit: 256 мегабайт

Апре-ски в буквальном переводе с французского звучит, как "после лыж". На самом деле под этим термином скрывается весь спектр отдыха на горнолыжных курортах, который следует непосредственно после катания — вечеринки, клубы, экскурсии, шопинг и, конечно, азартные игры.

Двое лыжников, впервые отдыхающих в Европе, недооценили уровень цен на местных курортах, поэтому уже через несколько дней их апре-ски состоял только из скудного ужина и четырёх стен гостиничного номера. Чтобы хоть как-то развлечься после катания, они придумали новую игру.

На листе бумаге перед игрой записывается последовательность из N различных целых положительных чисел A_i . Игроки ходят по очереди. За один ход игрок может отнять 1 от одного из чисел последовательности. Если после хода игрока в последовательности появляются два одинаковых числа, то одно из них вычеркивается. Если после хода игрока в последовательности появился 0, то он тоже вычеркивается. Игрок, который не может походить, объявляется проигравшим, а его оппонент — победителем.

Ваша задача, зная последовательность чисел на бумаге, определить победителя игры.

Input

В первой строке задана длина последовательности N ($1 \leq N \leq 3$).

Во второй строке через пробел заданы N элементов последовательности A_i ($1 \leq A_i \leq 10^{18}$). Все элементы последовательности различны.

Output

Если победителем станет игрок, ходивший первым, выведите 1, иначе выведите 2.

Examples

Standard input	Standard output
1 42	2
2 4 2	2
3 2 3 4	1

Problem E. Земля в Красной Поляне

Input file: Standard input
Output file: Standard output
Time limit: 2 секунды
Memory limit: 256 мегабайт

Приближающиеся XXII зимние Олимпийские игры в г. Сочи изменили до неузнаваемости соседний посёлок Красная Поляна, в котором будут проходить лыжные, санные и многие другие соревнования. За короткий срок здесь было возведено 3 новых горнолыжных комплекса, полностью переработана транспортная инфраструктура, построены сотни гостиниц... И стройка даже не думает останавливаться.

Администрация посёлка выделила для строительства новых гостиниц большой прямоугольный участок земли, равномерно поделенный на $N \times M$ одинаковых секторов, расположенных в N рядов и M столбцов. Все сектора были выставлены на продажу частным компаниям и для каждого из них установлена стоимость C_{ij} .

Есть две конкурирующие сети гостиниц — A и B . Как это часто бывает в крупном бизнесе, несмотря на видимую конкуренцию, обе они принадлежат одному владельцу. Владелец хочет построить N гостиниц сети A так, чтобы в каждом ряду выделенного участка было по одной гостинице A , и M гостиниц сети B так, чтобы в каждом столбце участка было по одной гостинице B . При этом для строительства каждой гостиницы требуется ровно один сектор, и никакие две гостиницы не могут быть построены в одном и том же секторе.

Ваша задача, определить минимальную стоимость S , которую предстоит заплатить владельцу за покупку земли под строительство гостиниц, согласно описанной схеме.

Input

В первой строке задано число рядов и столбцов земельного участка: N и M ($N \times M \leq 10^6$, $2 \leq N, M$).

В следующих N строках задана стоимость секторов в рядах, по M чисел C_{ij} ($1 \leq C_{ij} \leq 10^9$) в каждом.

Output

В первой строке выведите минимальную стоимость покупки земли S .

Examples

Standard input	Standard output
3 4 1 3 10 8 2 1 9 2 6 7 4 6	19

Note

В тестовом примере предлагается расположить гостиницы сети A в секторах $(1, 1)$, $(2, 2)$ и $(3, 1)$, а гостиницы сети B в секторах $(2, 1)$, $(1, 2)$, $(3, 3)$ и $(2, 4)$.

Problem F. Трансфер

Input file: Standard input
Output file: Standard output
Time limit: 1 секунда
Memory limit: 256 мегабайт

Чем ближе гостиница расположена к подъёмникам, тем дороже стоит проживание в ней — это правило знает любой горнолыжник. Понятное дело, что мало кому понравится идти до подъёмников пешком несколько километров или ездить каждое утро в общественном транспорте с громоздким инвентарём. Но к счастью в последнее время многие недорогие гостиницы, расположенные на удалении от курортов, стали предлагать отдыхающим собственные трансферные услуги.

Автобус, везущий утром N пассажиров из гостиницы к подъёмникам, делает во время пути одну остановку возле проката лыж. С вероятностью Pa_i каждый пассажир в момент остановки может спать — такие пассажиры не просыпаются до самого конца пути. С вероятностью Pb_i пассажир сидит в автобусе и бодрствует в течение всего времени остановки. С вероятностью Pc_i пассажир выходит из автобуса в прокат.

Через некоторое время водителю надоедает ждать вышедших пассажиров и он спрашивает по громкой связи: "все на своих местах?". Спящие пассажиры ответить на вопрос не могут. Вышедшие из автобуса не слышат вопроса и не знают о том, что автобус собирается уезжать. Каждый бодрствующий пассажир в автобусе отвечает "нет", если в автобусе нет кого-то из его друзей, и молчит, если все его друзья на месте. В случае если водитель услышал хотя бы один ответ "нет", он просит пассажиров связаться по телефону со своими друзьями и позвать их в автобус. Пассажиры, ответившие "нет", обзванивают своих друзей и те сразу же возвращаются в автобус. После этого водитель снова задаёт вопрос по громкой связи.

Описанная процедура повторяется до тех пор, пока после заданного водителем вопроса ни один пассажир не ответит "нет" — в этом случае автобус уезжает, а люди, не вернувшиеся в автобус, остаются в прокате.

Вам необходимо определить математическое ожидание числа забытых во время остановки пассажиров, если Вы знаете список друзей каждого пассажира. Известно, что дружественные связи образуют единую древовидную структуру, то есть каждый пассажир дружит хотя бы с одним другим пассажиром и среди дружественных связей нет циклов (если A и B друзья, B и C друзья, то A не может быть другом C). Примите во внимание, что ни один вышедший из автобуса пассажир не вернётся в автобус, пока ему не позвонит друг из автобуса.

Input

В первой строке задано число пассажиров N ($1 \leq N \leq 50000$). В следующих N строках заданы вероятности событий для каждого пассажира в процентах: Pa_i, Pb_i, Pc_i ($0 \leq Pa_i, Pb_i, Pc_i \leq 100$, $Pa_i + Pb_i + Pc_i = 100$).

Следующие $N - 1$ строк содержат описание дерева дружественных связей: каждая строка содержит одну пару друзей u_i и v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$).

Output

Математическое ожидание числа забытых во время остановки пассажиров. Ответ должен отличаться от эталонного не более чем на 10^{-6} .

Examples

Standard input	Standard output
1 32 46 22	0.220000000000
2 32 46 22 54 23 23 1 2	0.293600000000
5 78 12 10 17 13 70 39 19 42 22 16 62 93 1 6 1 5 2 5 2 3 4 5	1.640445408000

Problem G. Строительство подъемников

Input file:	Standard input
Output file:	Standard output
Time limit:	1 секунда
Memory limit:	256 мегабайт

В последние несколько лет в России стартовали сразу несколько амбициозных проектов по строительству горнолыжных курортов. Одним из важнейших этапов при создании нового курорта является исследование окружающей местности и разработка проекта трасс, который является отправной точкой для планирования транспортной и гостиничной инфраструктуры. Помимо проекта трасс необходимо с самого начала предусмотреть строительство горнолыжных подъемников.

Проект трасс представляет собой множество из M направленных участков трасс, соединяющих некоторые пары из N опорных точек на склонах. Для каждой опорной точки известна её высота H_i над уровнем моря, а направление участков трасс соответствует движению от верхних опорных точек к нижним. Множество опорных точек является связным, то есть существует путь по участкам трасс без учета их направлений от каждой опорной точки до любой другой.

Инвесторы, участвующие в строительстве нового курорта, заинтересованы в максимально быстрой отдаче от своих вложений. Вам поручено для готового проекта трасс разработать стартовый проект строительства простых бугельных подъемников, соответствующий нескольким критериям.

1. Подъемник может соединять любые две различных опорных точки S_i и F_i , но позволяет отдыхающим перемещаться только по направлению от точки S_i к точке F_i , причём должно выполняться условие $H_{S_i} \leq H_{F_i}$.
2. После строительства подъемников должна быть возможность добраться из каждой опорной точки до любой другой, осуществляя движение по направлению участков трасс и/или используя подъемники.
3. Суммарный перепад высот $Z = \sum_{i=1}^K H_{F_i} - H_{S_i}$ строящихся подъемников должен быть минимально возможным.
4. Среди всех решений, удовлетворяющих первым трём критериям, необходимо выбрать решение с минимальным числом K подъемников в проекте.

Input

В первой строке задано число опорных точек N ($1 \leq N \leq 100$) и число участков трасс M ($1 \leq M \leq 10000$).

Во второй строке заданы N чисел H_i ($0 \leq H_i \leq 10000$) - высоты опорных точек. Гарантируется, что все H_i попарно различны.

В следующих M строках заданы участки трасс в виде пар соединяемых ими номеров опорных точек: U_i и V_i ($1 \leq U_i, V_i \leq N$, $H_{U_i} > H_{V_i}$). Во входных данных нет участков трасс, соединяющих одинаковые пары опорных точек.

Output

В первой строке выведите суммарный перепад высот в подъемниках Z .

Во второй строке выведите количество подъемников в Вашем проекте K .

В следующих K строках выведите номера опорных точек S_i и F_i , соединяемых подъемниками.

Если существует несколько решений, соответствующих условию, выведите любое.

Examples

Standard input	Standard output
4 3	10
10 8 6 0	2
1 2	4 3
2 3	3 1
2 4	

Problem H. Потерянный ключ (Division 1 Only!)

Input file:	Standard input
Output file:	Standard output
Time limit:	3 секунды
Memory limit:	256 мегабайт

Группа горнолыжников арендовала для проживания коттедж на одном из европейских курортов. После возвращения с очередного дня катания выяснилось, что единственный выданный на группу ключ от коттеджа оказался потерян.

Обычно ключ оставляли под дверью, чтобы любой вернувшийся со склонов член группы мог без проблем войти внутрь. Но сегодня один из участников группы случайно забрал его с собой перед катанием и потерял где-то на склоне. Поняв, что из-за его ошибки остальной группе предстоит несколько часов мёрзнуть у входа в коттедж, потерявший ключ участник решил до завтрашнего дня не показываться на глаза своим товарищам и отправился в ночной клуб, надеясь, что им удастся связаться с хозяином и получить запасной ключ.

Однако все попытки группы связаться с хозяином коттеджа не увенчались успехом. В отчаянии участники решили отправиться на поиски ключа на склоны. Чтобы сократить область поисков, было решено записать со слов членов группы маршруты, на которых сегодня видели потерявшего ключ участника. Однако прежде чем отправляться на поиски, необходимо проверить по карте трасс достоверна ли собранная информация или нет.

Курорт, на котором отдыхает группа, входит в большой регион катания — множество объединённых единой инфраструктурой горнолыжных курортов. Карта трасс региона катания представляет собой связное множество из N опорных точек, соединённых $N - 1$ трассами, каждая из которых помечена некоторой меткой c_i . В силу того что регион катания объединяет несколько курортов, метки, которыми обозначаются трассы, могут повторяться. Кроме того на карте, которая попала в руки участников группы, не обозначена высота опорных точек, поэтому при решении задачи можно считать, что по трассам разрешается двигаться в обе стороны.

Рассмотрим кратчайший путь по трассам между опорными точками u и v : $P_1, P_2, P_3, \dots, P_k$, где $P_1 = u$, $P_k = v$ и k — количество опорных точек в кратчайшем пути. Выпишем метки трасс в порядке обхода пути: S_1, S_2, \dots, S_{k-1} , где S_i метка на ребре (P_i, P_{i+1}) . Тогда последовательность меток S является корректным маршрутом, а пара опорных точек (u, v) — положением маршрута на карте трасс.

Задан набор из M последовательностей меток P . Для каждой последовательности необходимо проверить, является ли она корректным маршрутом, и, если да, то найти одно из его положений на карте трасс.

Input

Первая строка содержит единственное целое число N ($1 \leq N \leq 50000$) — количество опорных точек на карте трасс.

Следующие $N - 1$ строк содержат описание трасс в виде троек чисел u_i, v_i и c_i ($1 \leq u_i, v_i \leq N$, $0 \leq c_i \leq 10^5$), где u_i, v_i — соединяемые трассой опорные точки, а c_i — метка трассы.

В следующей строке задано единственное целое число M ($1 \leq M \leq 50000$) — количество последовательностей, которые необходимо проверить.

Каждая из следующих M строк содержит последовательность P_i в следующем формате: длина L_i ($1 \leq L_i$) последовательности и L_i элементов последовательности $P_{i,j}$ ($0 \leq P_{i,j} \leq 10^5$). Все числа целые и разделены одним пробелом.

Гарантируется, что $M \leq \sum_{i=1}^M L_i \leq 50000$.

Output

Выведите M строк по одной на каждую последовательность P_i . Если последовательность P_i является корректным маршрутом, выведите пару вершин u_i, v_i , описывающих положение маршрута на карте трасс. Если положений маршрута существует несколько, выведите любое. Если строка P_i не является корректным маршрутом, то выведите два нуля через пробел.

Examples

Standard input	Standard output
6	2 1
1 2 1	2 6
2 3 7	6 2
1 6 2	3 4
6 4 3	0 0
6 5 3	4 5
14	0 0
1 1	4 2
2 1 2	4 3
2 2 1	6 3
4 7 1 2 3	0 0
5 7 1 2 3 3	0 0
2 3 3	3 2
2 1 1	0 0
3 3 2 1	
4 3 2 1 7	
3 2 1 7	
3 7 1 7	
3 2 3 2	
1 7	
1 5	
4	1 3
1 2 2	3 1
1 4 1	2 1
2 3 3	3 2
9	0 0
2 2 3	0 0
2 3 2	4 2
1 2	4 3
1 3	3 4
1 4	
2 1 1	
2 1 2	
3 1 2 3	
3 3 2 1	

Problem I. Вверх-вниз (Division 2 Only!)

Input file: Standard input
Output file: Standard output
Time limit: 1 секунда
Memory limit: 256 мегабайт

Катаясь на некоторых горнолыжных курортах, можно поймать себя на мысли, что чаще приходится подниматься вверх на подъемниках, чем ехать вниз на лыжах. Обычно причиной тому является плохая инфраструктура курорта.

Чтобы объективно оценить качество инфраструктуры курортов, некоторые горнолыжники ведут статистику своих перемещений. Результаты записываются в строку S : в случае спуска на лыжах к строке добавляется символ '>', в случае перемещения вверх на подъёмнике — символ '<'.

Вам необходимо по известной строке S восстановить последовательность высот H , на которых мог побывать горнолыжник. Последовательность H должна иметь длину равную $|S| + 1$ и состоять из целых чисел $0 \leq H_i \leq 10^9$. При этом если в позиции i строки S находится символ '>', то должно выполняться условие $H_i > H_{i+1}$, а если символ '<', то условие $H_i < H_{i+1}$. Из всех допустимых последовательностей необходимо выбрать такую, в которой количество различных высот H_i минимально.

Input

Единственная строка S ($1 \leq |S| \leq 10^5$), состоящая только из символов '<' и '>'.

Output

Искомая последовательность высот H .

Если решений существует несколько, выведите любое.

Examples

Standard input	Standard output
>>><<<	9 6 3 0 3 6 9
>>><><<	3 2 1 0 1 0 1 2

Problem J. Экипировка (Division 2 Only!)

Input file: Standard input
Output file: Standard output
Time limit: 1 секунда
Memory limit: 256 мегабайт

Для комфортного катания на лыжах очень важно хорошо экипироваться. Вам потребуются не только лыжи, палки и ботинки, но также и специальные лыжные штаны, куртка, маска, шлем, перчатки и множество других мелочей.

Перед поездкой Вы наметили покупку N различных товаров в спортивном магазине, но точная цена каждого из них Вам неизвестна. Однако со слов друзей Вы знаете, что стоимость i -го товара целая и находится на отрезке $[L_i, R_i]$.

Необходимо посчитать количество различных комбинаций P цен на товары, в которых цена каждого товара удовлетворяет ограничению $L_i \leq P_i \leq R_i$, а побитовая сумма $S = P_1 \oplus P_2 \oplus \dots \oplus P_N$ (операция XOR) всех купленных товаров составляет ровно X .

Input

Первая строка содержит два числа N и X ($2 \leq N \leq 8$, $0 \leq X \leq 10^{18}$).

В следующих N строках записаны пары чисел L_i и R_i ($0 \leq L_i \leq R_i \leq 10^{18}$).

Output

Выведите искомое число комбинаций цен по модулю 1 000 000 007.

Examples

Standard input	Standard output
3 5 0 3 4 7 2 5	8