

Problem A. Ancient Diplomacy

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Вы являетесь сенатором в древней империи, находящейся под властью могущественного тирана. Вы присоединились к секретному комитету Сената, планирующего заговор с целью смещения тирана.

Для того, чтобы заговор был успешным, требуется единство губернаторов всех провинций страны; более того — они должны принадлежать к одной и той же партии.

На данный момент каждый губернатор принадлежит или к Зелёной партии, или к Фиолетовой партии. Так как в античном мире принадлежность к партии определялась, например, спортивными предпочтениями — выбранный цвет соответствовал цвету колесниц, участвующих в соревнованиях, то не имеет значения, какая из партий в итоге будет включать в себя всех губернаторов.

Детально разобравшись в политической ситуации, ваш комитет выяснил, что один губернатор влияет на другого, если они являются друзьями и принадлежат к одной и той же партии. Таким образом, каждый месяц комитет посылает к некоторому губернатору лоббиста, который убеждает того перейти в другую партию; после того, как губернатор перешёл, все его друзья-губернаторы, которые были в одной партии с ним, также переходят в другую партию, далее друзья друзей и так далее. Для того, чтобы избежать обвинений в подыгрывании той или иной партии, комитет отправляет лоббистов Зелёной и Фиолетовой партий по очереди. Начинать можно с любой партии.

У комитета есть все данные насчёт того, кто из губернаторов с кем дружит; более того, известно, что у каждого губернатора не менее одного друга среди остальных губернаторов и что не существует изолированных групп губернаторов (то есть групп, в которых губернаторы дружат только с членами этой же группы, и которые не совпадают со всем множеством губернаторов).

Требуется найти минимальное количество месяцев, которое потребуется для того, чтобы все губернаторы оказались в одной — какой-либо — партии.

Input

Вход состоит из нескольких (не более 44) тестовых примеров.

В первой строке каждого тестового примера заданы два целых числа n ($2 \leq n \leq 100$) и m ($n - 1 \leq m \leq n(n - 1)/2$), где n — количество губернаторов и m — количество пар друзей среди губернаторов.

Вторая строка содержит n целых чисел — нулей и единиц, задающая принадлежность губернаторов к партиям, перечисленных в порядке возрастания номеров губернаторов; 0 обозначает зелёную партию, 1 — фиолетовую.

В каждой из последующих m строк заданы два целых числа a и b ($1 \leq a < b \leq n$), обозначающие, что губернаторы a и b являются друзьями (если a является другом b , то и b является другом a). Все m пар (a, b) являются попарно различными. Вход завершается строкой, содержащей два нуля, обрабатывать которую не требуется.

Output

Для каждого тестового примера в отдельной строке выведите одно целое число — наименьшее количество месяцев, которое потребуется для того, чтобы все губернаторы стали принадлежать к одной партии.

Examples

standard input	standard output
4 3	1
0 1 0 0	2
1 2	
2 3	
2 4	
5 4	
0 1 1 0 1	
1 2	
2 3	
3 4	
4 5	
0 0	

Problem B. Bob and Banjo (Division 1 Only!)

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

Пройдя известные игры Banjo-Kazooie и Banjo-Tooie, Боб решил сыграть в не получивший широкого распространения секретный сиквел этих игр — Banjo-Threeie. Игра проходит на плоскости, на которой введены декартовы координаты. Задача игрока — провести главного героя игры, медведя Banjo, от старта до финиша. Скорость медведя равна 1.

Впрочем, в игре имеются и препятствия; в частности, заполненный лавой кратер, который имеет форму круга. Однако медведь настолько вынослив, что может идти по поверхности лавы с той же скоростью — но не более t секунд подряд. Если медведь проводит в лаве более t секунд подряд, то игрок проигрывает. При этом перерыв между двумя непрерывными отрезками пребывания в лаве может быть сколь угодно коротким.

Боб хочет выяснить наименьшее время, за которое он сможет провести медведя от старта до финиша в соответствии с правилами игры.

Input

Вход состоит из не более, чем 333 тестовых примеров.

Каждый тестовый пример представляет собой одну строку, содержащую восемь целых чисел $x_1, y_1, x_2, y_2, x_c, y_c, r, t$, где (x_1, y_1) — координаты точки старта, (x_2, y_2) — координаты точки финиша, (x_c, y_c) — координаты центра кратера, r ($1 \leq r \leq 10^4$) — радиус кратера, и t ($0 \leq t \leq 10^4$) — максимальное время, которое медведь сможет провести в лаве без выхода из неё.

Все координаты удовлетворяют неравенству $-10^4 \leq x, y \leq 10^4$, все три заданные в условии точки попарно различны. Гарантируется, что ни начальная, ни конечная точки не лежат как в кратере, так и на его границе.

Вход завершается строкой из восьми нулей, обрабатывать которую не требуется.

Output

Для каждого тестового примера в отдельной строке выведите одно число — наименьшее время, которое требуется для того, чтобы провести медведя от старта до финиша, с абсолютной точностью не хуже 10^{-2} .

Examples

standard input	standard output
0 0 10 0 5 0 3 5 0 0 0 0 0 0 0 0	11.0

Problem C. Chess Knight's Poem

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Два шахматных коня решили совместно набрать короткий однострочный текст с помощью ноутбушной клавиатуры. Клавиатура показана на рисунке ниже. 30 клавиш заняты под различные символы, 4 — под клавишу Shift, 6 — под пробел.

Q q	W w	E e	R r	T t	Y y	U u	I i	O o	P p
A a	S s	D d	F f	G g	H h	J j	K k	L l	: ;
Z z	X x	C c	V v	B b	N n	M m	< ,	> .	? /
Shift	Shift	Space	Space	Space	Space	Space	Space	Shift	Shift

Обратите внимание, что пробел и клавиша Shift, которые обычно являются одиночными клавишами увеличенной длины, рассматриваются как много стандартных клавиш, каждая из которых имеет одну и ту же функцию.

Кони набирают текст, делая ходы. Кони ходят по обычным шахматным правилам, то есть конь может пойти на клетку, отстоящую от текущей на одну клетку по вертикали и две по горизонтали, или же на одну по горизонтали и две по вертикали. Например, если конь стоит на клавише с обозначением D, он может пойти на клавиши Q, Z, T, B, второй слева Shift или второй слева пробел.

Первый конь начинает движение с самого левого Shift-а, второй — с самого правого. Один конь может сделать несколько последовательных ходов. Первый ход может сделать любой из коней; при этом два коня не могут одновременно занимать одну и ту же клавишу.

Каждый ход коня приводит к тому, что к тексту справа добавляется не более одного символа. Если конь попадает на клавишу с символом или пробел, то добавляется один символ: в случае клавиши с символом печатается символ с нижней части клавиши, если второй конь при этом не стоит на клавише Shift, и с верхней — если стоит на ней. В случае пробела всегда выводится пробел (вне зависимости от положения второго коня). Если конь попадает на Shift, то добавления символа после этого хода не происходит.

По заданному тексту выясните, смогут ли кони его набрать.

Input

Вход содержит несколько (не более 222) тестовых примеров.

Каждый тестовый пример состоит из одной строки — текста длиной не менее одного и не более 100 символов, состоящего только из символов, присутствующих на описанной в задаче клавиатуре (включая пробелы). Гарантируется, что текст не начинается с пробела и не заканчивается пробелом.

Вход завершается строкой, содержащей один символ (*), обрабатывать которую не требуется.

Output

Для каждого тестового примера выведите 1, если кони смогут набрать соответствующий текст, и 0 в противном случае.

Examples

standard input	standard output
S,veA,eVE,aU	1
S,veA,eVE,aUc	0
CAlming eventa	1
CAL	1
*	

Problem D. Drone

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

Вы, как командир разведгруппы, получили информацию о передвижении колонны противника по ключевой магистрали. Благодаря фотографиям со спутника вы можете определить текущее местонахождение и скорость каждого грузовика на трассе (которую в практических целях можно рассматривать как бесконечную прямую).

Кроме того, вам известно, что каждый грузовик движется с постоянной скоростью и что грузовики могут безопасно обгонять друга (или разъезжаться в случае, если они двигаются навстречу) без снижения скорости.

Вы хотите направить к магистрали разведывательный беспилотник, оборудованный специальными сенсорами, позволяющими выяснить, какой именно груз перевозится.

Запас батареи беспилотника позволяет однократно мгновенно получить всю информацию о грузовиках на некотором отрезке (зависящем от типа и настроек аппарата). Естественно, что Вы хотите направить беспилотник в такую точку и в такой момент, когда все грузовики находятся настолько близко друг к другу, насколько это возможно. Чтобы правильно выбрать аппарат, Вам требуется найти длину этого отрезка.

Input

Вход содержит несколько (не более 33) тестовых примеров.

Тестовый пример начинается строкой, содержащей целое число n ($1 \leq n \leq 10^5$) — количество грузовиков. Каждая из последующих n строк содержит два целых числа: x и v ($-10^5 \leq x, v \leq 10^5$), задающих начальную координату (x , в метрах) и скорость (v , в метрах в час) для этого грузовика. Знак скорости задаёт направление движения.

Вход завершается строкой, содержащей одно число 0. Обработать эту строку не требуется.

Output

Для каждого тестового примера выведите в отдельной строке длину в метрах минимального отрезка, на котором окажутся все грузовики в некоторый момент времени с абсолютной точностью не хуже 10^{-2} .

Examples

standard input	standard output
2	0.0
-100 1	1
100 -1	200.000
3	
-100 1	
100 -1	
101 -1	
3	
-100 -1	
0 0	
100 1	
0	

Problem E. Ed and The Legend of Zelda (Division 1 Only!)

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

Эд играет в недавно вышедший квест из серии «Legend of Zelda». В этой игре требуется выполнить n заданий.

При этом некоторые задания должны быть выполнены перед остальными. Для задания с номером i ($i = 2 \dots n$) всегда определено задание P_i , которое должно быть выполнено перед заданием i . Назовём P_i *обязательным* заданием для задания i .

Первое задание (которое всегда имеет номер 1) не требует выполнения какого-либо задания перед ним. Также гарантируется отсутствие циклов зависимостей, которые приводят к тому, что какое-либо задание должно быть выполнено перед самим собой.

В этой игрушке, как обычно, присутствуют и скрытые «читы». Для каждого i от 2 до n существует чит, который позволяет выполнить соответствующее задание не после задания P_i , а после задания P_{P_i} (то есть задания, которое требуется для выполнения задания P_i ; в случае, если $P_i = 1$, чит позволяет выполнить i -е задание вне зависимости от того, были ли выполнены какие-либо задания).

При этом существует следующее ограничение: каждое задание может быть задействовано в чите только один раз, то есть если был применён чит к заданию i (позволяющий выполнить задание i перед заданием P_i), то нельзя применять чит ни к заданию P_i , ни к какому-либо заданию, для которого обязательным является одно из заданий i и P_i .

Эдди хочет пройти игру, используя не более k читов. Найдите, сколькими способами он сможет это сделать. Способы считаются различными, если порядок выполнения заданий отличается. Ответ выведите по модулю $10^9 + 7$.

Input

Вход содержит несколько (не более 77) тестовых примеров.

Каждый тестовый пример состоит из двух строк. Первая строка тестового примера содержит два целых числа n ($1 \leq n \leq 200$) и k ($0 \leq k < n$) — количество заданий и максимально возможное количество читов. Во второй строке содержатся $n - 1$ целых чисел p_i ($1 \leq p_i \leq n$), разделённых пробелами — номера заданий P_i , требуемых для выполнения заданий $2, 3, \dots, n$ соответственно.

Вход завершается строкой, содержащей два нуля, обрабатывать которую не требуется.

Output

Для каждого тестового примера в отдельной строке выведите одно целое число — количество способов пройти игру, использовав не более, чем k читов, взятое по модулю $10^9 + 7$.

Examples

standard input	standard output
5 1	38
1 1 5 1	4
3 2	
1 1	
0 0	

Note

Упомянутые в ответе к первому тестовому примеру из условия 38 способов прохождения игры не более, чем с одним читом:

0 читов:

(1,2,3,5,4), (1,2,5,3,4), (1,2,5,4,3), (1,3,2,5,4), (1,3,5,2,4), (1,3,5,4,2), (1,5,2,3,4), (1,5,2,4,3), (1,5,3,2,4),
(1,5,3,4,2), (1,5,4,2,3), (1,5,4,3,2).

Чит, применённый ко второму заданию:

(2,1,3,5,4), (2,1,5,3,4), (2,1,5,4,3).

Чит, применённый к третьему заданию:

(3,1,2,5,4), (3,1,5,2,4), (3,1,5,4,2).

Чит, применённый к четвёртому заданию:

(1,2,3,4,5), (1,2,4,3,5), (1,2,4,5,3), (1,3,2,4,5), (1,3,4,2,5), (1,3,4,5,2), (1,4,2,3,5), (1,4,2,5,3), (1,4,3,2,5),
(1,4,3,5,2), (1,4,5,2,3), (1,4,5,3,2).

Чит, применённый к пятому заданию:

(5,1,2,3,4), (5,1,2,4,3), (5,1,3,2,4), (5,1,3,4,2), (5,1,4,2,3), (5,1,4,3,2), (5,4,1,2,3), (5,4,1,3,2).

Problem F. Fix and Solve (Division 1 Only!)

Input file: *standard input*
Output file: *standard output*
Time limit: 40 seconds
Memory limit: 256 mebibytes

Ваш учитель, всемирно известный автор задач, окончательно решил уйти на покой и заняться разведением пчёл. Однако напоследок он решил придумать ещё одну задачу и отправить предполагаемый шедевр в ICFP (International Committee for Fantastic Problems). Вам же поручено проверить работу учителя.

Задача оказалась связана с теоретико-числовыми построениями. Вы решили задачу, написали решение... решение не прошло. После нескольких часов отладки и проверки собственных выкладок вы осознали, что решение корректно, а некорректны входные данные — на этот раз Акела промахнулся...

В этой задаче вам дана целочисленная последовательность длины n $V_{1..n}$, при этом гарантируется, что в каждом отрезке из k последовательных её элементов (V_i, \dots, V_{i+k-1}) для всех i от 1 до $n - k + 1$ любые два элемента будут взаимно простыми. Однако в реальности во входных данных это требование далеко не везде выполнялось.

Пытаясь разобраться в том, как же на самом деле устроены данные, вы подсчитали, сколько отрезков длины k не соответствует условиям задачи. Ваш учитель начал исправлять данные. Он сделал m последовательных изменений; каждое изменение включает в себя выбор позиции a в последовательности и замена значения числа в этой позиции на b . После каждого изменения он предлагает вам снова подсчитать, сколько отрезков длины k не соответствует условиям задачи.

В конце он предложил решить задачу, подсчитав сумму получившейся после всех m изменений последовательности.

Ваша задача — выполнить все описанные действия.

Input

Вход состоит из нескольких (не более 12) тестовых примеров.

Первая строка каждого тестового примера содержит три целых числа n ($1 \leq n \leq 10^5$), k ($1 \leq k \leq n$) и m ($1 \leq m \leq 10^5$), где n — количество элементов в последовательности V , k — длина выбираемого отрезка и m — количество сделанных учителем изменений.

i -я из последующих n строк содержит одно целое число V_i ($1 \leq V_i \leq 10^5$) — i -й элемент последовательности. Каждая из последующих m строк содержит два целых числа a ($1 \leq a \leq n$) и b ($1 \leq b \leq 10^5$), обозначающие, что во время очередной правки автор изменил значение V_a на b .

Сумма всех n во входном файле не превосходит 10^6 . Сумма всех m во входном файле не превосходит 10^6 .

Вход завершается строкой, состоящей из четырёх нулей, обрабатывать которую не требуется.

Output

Для каждого тестового примера выведите $m + 2$ целых чисел, по одному числу на строку. Первое число должно обозначать количество отрезков длины k в первоначальных данных, не удовлетворяющих условию попарной взаимной простоты.

Каждое из последующих m чисел должно обозначать аналогичное количество отрезков, подсчитанное после очередного изменения данных автором.

Последнее число должно представлять собой сумму последовательности, получившейся после всех m изменений.

Examples

standard input	standard output
6 3 4	2
7	3
2	3
3	3
4	2
5	42
6	
4 3	
5 9	
4 10	
6 11	
0 0 0	

Problem G. Gold Bandits

Input file: *standard input*
Output file: *standard output*
Time limit: 5 seconds
Memory limit: 256 mebibytes

В далёкой-далёкой стране среди высоких гор находится долина, в которой построено много посёлков, соединённых дорогами. Этой страной правит жестокий король, который требует, чтобы каждый посёлок ежегодно платил ему дань золотыми монетами. Жители посёлка должны как можно быстрее принести золото королю по его первому требованию. Замок короля расположен в одном из посёлков.

Кроме того, один из посёлков является разбойничьим посёлком. Разбойники не копят золото для короля — как и положено разбойникам, они тратят всё поступающее в посёлок золото сразу же. Если король требует дани, разбойники отправляются из своего посёлка в королевский замок, при этом в каждом посёлке, через который они проходят, они принимают решение — ограбить посёлок, забрав всё накопленное золото, или же не делать этого.

В соответствии с требованием короля путь разбойников к замку должен проходить по дорогам через минимально возможное количество посёлков. Кроме того, после визита к королю разбойники должны вернуться назад, а проходить через посёлки, которые они ограбили, разбойники обоснованно опасаются.

Определите, какое наибольшее количество золота разбойники могут награть по пути в замок, оставив себе возможность безопасно вернуться назад. При этом оптимальность обратного пути разбойников не заботит.

Input

Вход состоит из нескольких (не более 44) тестовых примеров.

Каждый тестовый пример начинается строкой, содержащей два целых числа n ($3 \leq n \leq 36$) и m ($n - 1 \leq m \leq n(n - 1)/2$), где n — количество посёлков и m — количество дорог. Посёлки пронумерованы последовательными целыми числами от 1 до n . Посёлок с первым номером — разбойничий посёлок, в посёлке 2 расположен королевский замок. Следующая строка содержит $n - 2$ целых чисел g ($1 \leq g \leq 5,000$) — количество золота в каждом из остальных посёлков (в порядке 3, 4, ..., n). Каждая из последующих m строк содержит два целых числа a и b ($1 \leq a < b \leq n$), обозначающие, что между посёлками a и b есть двунаправленная дорога. Все m пар (a, b) попарно различны. Гарантируется, что из любого посёлка в любой другой можно пройти по дорогам.

Вход завершается строкой, содержащей два нуля, обрабатывать которую не требуется.

Output

Для каждого тестового примера в отдельной строке выведите одно целое число — максимальное количество золота, которое могут награть бандиты по пути в замок и после этого вернуться назад.

Examples

standard input	standard output
3 3	0
1	24
1 2	800
2 3	700
1 3	
4 4	
24 10	
1 3	
2 3	
2 4	
1 4	
6 8	
100 500 300 75	
1 3	
1 4	
3 6	
4 5	
3 5	
4 6	
2 5	
2 6	
7 7	
90 1000 700 2000 800	
1 3	
1 4	
1 5	
3 7	
5 6	
2 6	
3 6	
0 0	

Problem H. How Many Values?

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

По заданной последовательности A , состоящей из n целых чисел, определим $f(lo, hi)$, $1 \leq lo \leq hi \leq n$, как наибольший общий делитель всех элементов последовательности от A_{lo} до A_{hi} включительно. Заметим, что lo и hi — это индексы, а не сами элементы последовательности.

По заданному массиву выясните, сколько различных значений может принимать функция $f(lo, hi)$ для всех возможных пар lo и hi .

Input

Вход содержит несколько (не более 99) тестовых примеров.

Каждый тестовый пример начинается строкой, содержащей одно целое число n ($1 \leq n \leq 10^5$) — длину последовательности.

i -я из последующих n строк содержит одно целое число A_i ($1 \leq A_i \leq 100$) — i -й элемент последовательности.

Вход завершается строкой, содержащей одно число 0, обрабатывать которую не следует.

Output

Для каждого тестового примера в отдельной строке выведите одно целое число — количество различных значений функции $f(lo, hi)$.

Examples

standard input	standard output
2	3
4	5
6	
3	
3	
6	
8	
0	

Problem I. Integer Estate Agent

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

После протестов жителей Флатландии против ограничения их прав правительство Флатландии отменило некоторые сословные ограничения. В частности, покупать здания на знаменитой Нулевой улице сейчас могут не только знатные многоугольники, но и пятиугольники, квадраты, а также треугольники (конечно, если у них хватит денег на покупку).

Как сотрудник компании «Integer Estate Agency», вы отвечаете за одну из сторон улицы. Нумерация домов начинается от Нулевой площади. Дом 1 находится прямо на площади, дом 2 идёт сразу после дома 1, дом 3 идёт после дома 2 и так далее до бесконечности. Дом k стоит $k + 1$ флатландских тугриков.

Перспективный покупатель планирует потратить в точности n флатландских тугриков на покупку группы подряд идущих домов. Сколько вариантов вы ему сможете предложить? Например, если он собирается потратить 5 тугриков, у него есть два варианта — купить дома 1 и 2 ($2+3=5$ тугриков) или купить дом 4.

Input

Вход состоит из не более, чем 555555 тестовых примеров.

Каждый тестовый пример состоит из одной строки, содержащей целое число n ($1 \leq n \leq 10^6$) — количество флатландских тугриков, которые готов потратить покупатель.

Вход заканчивается строкой с $n = 0$, обрабатывать которую не требуется.

Output

Для каждого тестового примера выведите в отдельной строке количество различных вариантов, удовлетворяющих требованиям покупателя.

Examples

standard input	standard output
1	0
2	1
5	2
0	

Problem J. John and Super Mario 169

Input file: *standard input*
Output file: *standard output*
Time limit: 5 seconds
Memory limit: 256 mebibytes

Тестер Джон играет в последнюю версию игры Super Mario 169 (которая последует сразу за Super Mario 64). Действие этой игры происходит в океане, который может быть представлен как трёхмерное пространство с введёнными в нём декартовыми координатами. Задача игрока — управляя плавающим по океану Марио, собрать все монеты, которых может быть не более 169.

Изначально в игре нет ни одной монеты; при этом есть не более 13 переключателей. Марио может подплыть к любому переключателю и нажать на него. Нажатие каждого из переключателей приводит к появлению определённого числа (не более 13) монет с фиксированными координатами. При этом каждый переключатель может быть активирован только один раз, и все монеты, которые не были собраны к моменту нажатия на переключатель, исчезают и Марио уже не сможет их собрать.

Чтобы убедиться, что он играет оптимально, Джон хочет узнать минимальное расстояние, которое должен проплыть Марио, чтобы собрать все монеты. Переключатели и монеты считать точками.

Input

Вход состоит из нескольких (66 или менее) тестовых примеров.

Каждый тестовый пример начинается строкой, содержащей четыре целых числа n , mx , my , mz , где n ($1 \leq n \leq 13$) — количество переключателей, а в точке (mx, my, mz) Марио находится в момент старта игры.

Далее следует n блоков описания переключателей; каждый блок начинается строкой, содержащей четыре целых числа k , sx , sy , sz , где k ($1 \leq k \leq 13$) — количество монет, которые появляются при нажатии на соответствующий переключатель, а координаты (sx, sy, sz) задают точку, в которой находится переключатель. В каждой из последующих k строк заданы три целых числа cx , cy , cz , — координаты точек (cx, cy, cz) , в которых расположены монеты.

Все координаты не превосходят 1000 по абсолютной величине. Никакие две точки в одном тестовом примере (точка старта Марио, координаты переключателей и монет) не совпадают.

Вход заканчивается строкой, содержащей четыре нуля, обрабатывать которую не требуется.

Output

Для каждого тестового примера выведите одно число — минимальное расстояние, которое должен проплыть Марио, чтобы собрать все монеты, с точностью не хуже 10^{-2} .

Example

standard input	standard output
2 5 5 0 4 6 0 0 7 0 0 -11 -1 0 -11 1 0 -10 0 0 2 5 0 0 0 0 0 0 5 0 0 0 0 0	44.22

Problem K. Keys (Division 2 Only!)

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

Однажды маленькому Пете подарили кучу разнообразных ключей. Петя поиграл с ключами и разбросал их по комнате. Уходя из комнаты, Петя посмотрел на разбросанные ключи и задался вопросом — сколько троек ключей лежат на одной прямой?

Считая, что ключи представляют собой точки, координаты которых Вам известны, помогите Пете ответить на его вопрос.

Input

Первая строка входа содержит целое число T ($1 \leq T \leq 30$) — количество тестовых примеров.

В первой строке каждого тестового примера задано неотрицательное целое число $N \leq 1000$ — количество ключей. Во второй строке заданы $2N$ целых чисел, не превосходящих 10^4 по абсолютной величине — координаты ключей (в порядке $x_1, y_1, x_2, y_2 \dots$). Никакие два ключа не находятся в одной точке.

Output

Для каждого тестового примера в отдельной строке выведите одно число — количество троек ключей, лежащих на одной прямой.

Example

standard input	standard output
4	1
3	2
0 0 0 1 0 2	4
5	4
0 0 0 1 0 2 1 1 2 2	
4	
0 0 1 1 2 2 3 3	
5	
0 0 -10 -10 4 1732 2 2 40 40	

Problem L. Letter Game (Division 2 Only!)

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

В игре с буквами для смартфона игрок должен составлять слова из заданных на экране 9 букв, 8 из которых расположены на окружности, а одна — в центре. Каждое слово должно состоять из четырёх или более букв и содержать центральную букву (буква может использоваться не более того количества раз, которое она встречается). Напишите программу, которая будет проверять корректность предложенных игроком слов.

Input

Входной файл состоит из нескольких тестовых примеров. Первая строка каждого тестового примера содержит последовательность из 9 заглавных латинских букв — слово, заданное на экране. Центральная буква находится в центре слова (то есть является 5-ой).

Вторая строка содержит целое положительное число N , за которым следуют N строк, содержащие слова, составленные из не более, чем девяти заглавных латинских букв — предложенные игроком слова.

Входной файл заканчивается строкой, содержащей число 0, обрабатывать которую не требуется.

Общее количество предложенных игроком слов во всех тестовых примерах не превосходит 1000.

Output

Для каждого предложенного игроком слова X выведите “ X is valid”, если предложенное игроком слово является корректным с точки зрения правил игры, или “ X is invalid” в противном случае.

Examples

standard input	standard output
PHONEGAME	FAME is invalid
5	PEG is invalid
FAME	GAME is valid
PEG	GAPHON is invalid
GAME	GENOME is valid
GAPHON	
GENOME	
0	

Problem M. Math Honors (Division 2 Only!)

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

Каждый семестр тысячи студентов выбирают курс математики. Однако на усиленный курс математики берут далеко не всех. Критерий выбора для усиленного курса следующий: студент должен иметь GPA не менее 2.50 и оценку не хуже «С» по информатике.

Напишите программу, которая по данным GPA и оценке по информатике определяет, получится ли у студента выбрать усиленный курс математики.

Input

Первая строка входа содержит одно целое число $T \leq 100$ — количество тестовых примеров.

Каждый тестовый пример записан в отдельной строке и состоит из вещественного числа $0.0 < X \leq 4.0$, заданного с двумя знаками после десятичной точки, и символа Y — значения GPA и оценку по информатике соответственно. Оценка представляется одной из заглавных латинских букв от 'A' до 'F'.

Output

Для каждого тестового примера в отдельной строке выведите текст “Eligible”, если студент с соответствующими данными может быть допущен к курсу, и “Ineligible” в противном случае.

Example

standard input	standard output
3	Ineligible
2.49 A	Eligible
3.56 C	Ineligible
3.56 D	