

Задача А. Так приказал фараон...

Имя входного файла:	meteor.in
Имя выходного файла:	meteor.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 Mebibytes

Во время подготовки к восстанию против Харконненов Пол Атридес, желая как можно более чётко увидеть основную линию будущих событий, принял чрезмерное количество гошлана. В результате сбылось древнее пророчество Бене Джессерит — в памяти Пола соединилась память всех его предков, живших за много поколений до него, ожили образы давно исчезнувших стран на далёкой Земле...

Великий Фараон Эхнатон, Сын Солнца, не доверял жрецам. Произведённые ими расчёты времени разлива Нила по «неподвижным звёздам» и планетам казались Эхнатону неточными. Эхнатон повелел провести наблюдение за падающими звёздами. Всю ночь самые зоркие из рабов фараона, не смыкая глаз, пытались запомнить траектории метеоров, записывая координаты точек, через которые проходили эти траектории.

Так как египтяне считали небо плоским, то траекторию метеоров принимали за кривую на плоскости. Формула, задающая эту кривую, была известна с глубокой древности: $y = \max(p(x), 0)$, где $p(x) = a(x - h)^2 + k$, при этом $a < 0$ и $k > 0$.

Для построения новых таблиц фараону надо уметь отличить записи, сделанные разными наблюдателями, но относящиеся к одному и тому же метеору, если известно, что не существует двух различных метеоров, для которых $a_1 = a_2, k_1 = k_2, h_1 = h_2$.

Иначе говоря, необходимо по двум последовательностям точек, каждая из которых описывает движение одного метеора, определить, могут ли обе серии наблюдений относиться к одному и тому же метеору.

Формат входного файла

Входной файл содержит несколько тестовых примеров, отделённых друг от друга пустой строкой. Каждый тестовый пример состоит из 3 строк. Первая строка содержит два целых числа n_1 и n_2 ($1 < n_1, n_2 < 10$), разделённых пробелом — количество точек в первой и второй серии наблюдений. Вторая и третья строки содержат координаты соответственно n_1 точек первой серии и n_2 точек второй серии (в формате $x_1 y_1 x_2 y_2 \dots x_n y_n$). При этом $x_1 < x_2 < \dots < x_n, 0 \leq x_i \leq 10^5, 0 \leq y_i \leq 1000$. Последним примером во входном файле является пример с $n_1 = n_2 = 0$, который обрабатывать не надо.

Формат выходного файла

Для каждого тестового примера Ваша программа должна вывести в отдельной строке “same”, если обе серии наблюдений относятся к одному метеору, “unsure”, если они могут относиться как к одному метеору, так и к двум разным, и “different”, если они не могут относиться к одному метеору.

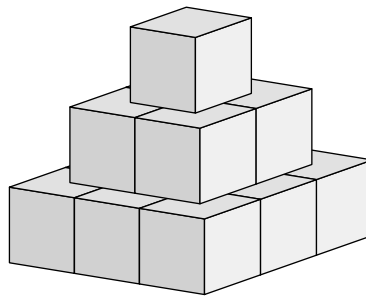
Пример

meteor.in	meteor.out
6 4	same
0 0 1 3 2 4 3 3 4 0 7 0	unsure
1 3 2 4 3 3 4 0	different
6 1	
0 0 1 3 2 4 3 3 4 0 7 0	
0 0	
6 2	
0 0 1 3 2 4 3 3 4 0 7 0	
1 3 2 5	
0 0	

Задача В. Постройка пирамиды

Имя входного файла: pyramid.in
Имя выходного файла: pyramid.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Mebibytes

Великий Фараон Эхнатон, Сын Солнца, повелел построить новую пирамиду. Для постройки были доставлены n каменных блоков, имеющих форму куба пяти локтей в длину, пяти в ширину и пяти в высоту (локоть — древнеегипетская мера длины, равная примерно 466 миллиметрам). Пирамида строится так: сначала выкладывается основание $k \times k$ блоков, на него кладётся следующий ярус размера $(k - 1) \times (k - 1)$ блоков, и так далее до последнего k -го яруса, состоящего всего лишь из одного каменного блока. Пирамида высоты в три каменных блока, состоящая из $9 + 4 + 1 = 14$ блоков, показана на рисунке.



Главный землемер фараона хочет узнать, какова максимальная высота пирамиды, которую можно построить из n блоков — не зная заранее высоты, он не может начать строительство. К сожалению, эта задача оказалась ему не по силам, и он обратился за помощью к вам — своему помощнику.

Формат входного файла

В первой строке входного файла записано одно целое число n — количество каменных блоков ($1 \leq n \leq 32\,000$).

Формат выходного файла

В первой строке выходного файла выведите одно целое число — максимальную возможную высоту пирамиды в блоках.

Примеры

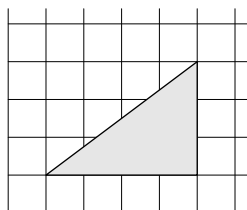
pyramid.in	pyramid.out
9	2
14	3

Задача С. Треугольная комната

Имя входного файла: room.in
Имя выходного файла: room.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Mebibytes

После укладки основания пирамиды (это оказался квадрат из 45×45 блоков) пришла пора вырубить в ней комнату — будущую усыпальницу для одного из родственников фараона. Эхнатон решил, что усыпальница в этот раз, вопреки всем канонам, будет треугольной. Стороны её должны иметь длины, кратные размерам блоков, из которых построена пирамида. Одна сторона должна быть в длину как a блоков, стоящих в ряд друг за другом, другая — как b блоков, третья — как c блоков. Числа a , b и c — это три знаменательных даты в жизни родственника фараона.

Главный землемер, руководящий строительством, хочет, кроме того, чтобы каждый из трёх углов комнаты находился на стыке четырёх блоков, из которых составлена пирамида. Иными словами, если нарисовать основание пирамиды на плоскости и ввести координаты так, чтобы один угол пирамиды оказался в начале координат, а противоположный — в точке $(45; 45)$, то вершины треугольника, соответствующего комнате, должны оказаться в точках с целыми координатами. Кроме того, углы комнаты должны быть строго внутри основания пирамиды. Пример такой комнаты для $a = 3$, $b = 5$ и $c = 4$ показан на рисунке.



Помогите главному землемеру и в этот раз — выясните, каким образом вырубить комнату с данными длинами сторон так, чтобы выполнить его требования, или укажите, что это невозможно.

Формат входного файла

В первой строке входного файла записаны через пробел три целых числа a , b и c — длины стен треугольной комнаты ($1 \leq a, b, c \leq 43$). Числа a , b и c таковы, что из трёх отрезков с такими длинами можно составить треугольник, площадь которого строго положительна.

Формат выходного файла

Если комнату с данными длинами стен указанным образом поместить в пирамиду невозможно, в первой строке выходного файла выведите "NO". В противном случае в первой строке выведите "YES", а в следующих трёх строках — координаты углов комнаты, по одному углу на строке. Порядок углов не имеет значения. Все координаты должны быть целыми числами в промежутке от 1 до 44, включительно.

Примеры

room.in	room.out
3 5 4	YES 1 1 5 1 5 4
2 3 4	NO

Задача D. Ловушки

Имя входного файла: traps.in
Имя выходного файла: traps.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Mebibytes

Эхнатон, Сын Солнца, хочет защитить новую пирамиду от воров и мародёров. Для этого в пирамиде, кроме усыпальницы, будет вырублено ещё несколько комнат поменьше. Все комнаты, включая усыпальницу, будут соединены системой коридоров таким образом, чтобы из любой комнаты в любую другую существовал ровно один путь, не проходящий ни по какому коридору дважды. Во многих комнатах будут установлены смертоносные ловушки.

У Эхнатона есть верный слуга — шаман родом из далёких южных стран, владеющий магией вуду. Фараон повелел шаману наложить заклятие на пирамиду: как только непрошенный посетитель оказывается в тупике — комнате, в которую ведёт только один коридор — волны отчаяния и безысходности должны захлестнуть его разум и подтолкнуть несчастного к ближайшей ловушке. Заклятие, однако, не должно действовать в священной комнате — усыпальнице, — даже если из неё ведёт всего один коридор.

Чтобы наложить заклятие, шаману необходимо знать точное количество тупиков в пирамиде. Чтобы узнать это число, он позвал вас — помощника главного землемера. К счастью, у вас сохранился план пирамиды. Ответьте шаману, сколько тупиков насчитывается на этом плане.

Формат входного файла

В первой строке входного файла записаны через пробел два целых числа n и m — количество комнат и количество коридоров, соответственно ($2 \leq n \leq 20$, $m > 0$). Следующие m строк содержат описания коридоров; в i -й из этих строк записаны два числа u_i и v_i через пробел, означающие, что i -й коридор соединяет комнаты с номерами u_i и v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$). Комнаты нумеруются с единицы; комната с номером 1 — это усыпальница. Любые две комнаты соединены не более чем одним коридором.

Формат выходного файла

В первой строке выходного файла выведите одно целое число — количество тупиков. Помните, что усыпальница тупиком не считается.

Примеры

traps.in	traps.out
3 2 1 2 1 3	2
3 2 1 2 2 3	1

Задача E. Магия вуду

Имя входного файла: voodoo.in
Имя выходного файла: voodoo.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Mebibytes

Чтобы наложить на пирамиду заклятие, шаман Эхнатона должен использовать один или несколько магических кристаллов. Каждый кристалл характеризуется целым числом — своей силой. От силы используемых кристаллов зависит мощность заклятия: при использовании k кристаллов, имеющих силы f_1, f_2, \dots, f_k , заклятие будет иметь силу $(f_1 \cdot f_2 \cdot \dots \cdot f_k) \bmod m$, где m — магическое число вуду, равное $2^{30} = 1\,073\,741\,824$. Выражение " $a \bmod b$ " означает остаток от деления a на b .

Шаман уже убедился в ваших способностях к вычислениям, и дал новое задание. Он снабдил вас информацией об имеющихся у него магических кристаллах. Теперь шаман просит сказать ему, какой максимальной мощности может достичь заклятие при правильном выборе из имеющихся кристаллов одного или нескольких, которые надо использовать.

Формат входного файла

В первой строке входного файла записано целое число n — количество магических кристаллов, имеющихся в распоряжении шамана ($1 \leq n \leq 12$). Во второй строке записаны n целых чисел p_1, p_2, \dots, p_n через пробел — силы магических кристаллов ($1 \leq p_i \leq 10\,000$).

Формат выходного файла

В первой строке выходного файла выведите одно число — максимальную возможную силу заклятия.

Примеры

voodoo.in	voodoo.out
2 1 3	3
3 10000 11 9999	99990000

Задача F. Код Эхнатона

Имя входного файла: `code.in`
Имя выходного файла: `code.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Mebibytes

Не полагаясь только на лабиринт комнат и магические ловушки, Великий Фараон Эхнатон, Сын Солнца, приказал оборудовать дверь усыпальницы хитрым замком.

Замок устроен следующим образом: Имеется блок из N рычагов, каждый из которых может быть в положении A или B . Конструкция блока позволяет за одну минуту установить любое количество подряд идущих рычагов в одинаковое состояние (то есть или все в A , или все в B , например, из $ABABBB$ можно за одно переключение сделать $AAAAAB$, установив рычаги с первого по четвёртый в положение A).

Однако Эхнатон не уверен, что, даже зная положение рычагов, открывающее дверь, можно быстро проделать необходимые действия. Поэтому он хочет выяснить, какое минимальное время понадобится для перевода блока из начального положения в положение, открывающее дверь.

Формат входного файла

В первой строке входного файла задано целое число $1 \leq t \leq 100$ — количество тестовых примеров. Каждый тестовый пример состоит из двух строк, каждая из которых состоит из $0 \leq N \leq 200$ символов A и B . Первая строка описывает начальное состояние рычагов, а вторая — состояние рычагов, открывающее дверь.

Формат выходного файла

Для каждого тестового примера выведите на отдельной строке минимальное время в минутах, которое потребуется для установки рычагов в требуемое состояние.

Пример

<code>code.in</code>	<code>code.out</code>
2	1
BB	2
AA	
BAAAB	
ABBA	