

Задача А. Проценты

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Вася недавно выиграл в лотерею — целых n бурлей! Теперь он хочет положить все выигранные деньги на 6 месяцев в БерБанк. В БерБанке есть два вида вкладов на 6 месяцев:

- вклад «Попроще»: через 6 месяцев клиент получает назад сумму вклада плюс p процентов от нее,
- вклад «Посложнее»: в конце каждого месяца к текущей сумме вклада добавляется q процентов от текущей суммы.

Если, например, $n = 100$, $p = 77$ и $q = 10$, то с помощью вклада первого вида можно получить 177 бурлей. Если открыть вклад второго вида, то в конце первого месяца на счете будет 110 бурлей, в конце второго к этой сумме прибавится 10 процентов от нее и получится 121, в конце третьего месяца на счете будет 133.1 и так далее. В итоге с помощью такого вклада по истечении шести месяцев можно будет получить 177.1561 бурлей. Тем самым, лучше открыть вклад второго вида.

Вася долго думал, как ему выгоднее положить деньги. Он решил, что будет надежнее, если вы напишете программу, которая выясняет какой вклад необходимо открыть, чтобы через 6 месяцев получить максимальную сумму.

Формат входного файла

В первой строке входного файла содержатся три положительных целых числа n , p и q , не превосходящие 100. Гарантируется, что один из вкладов окажется более выгодным, чем другой.

Формат выходного файла

Выведите 1, если вклад «Попроще» выгоднее, чем вклад «Посложнее». В противном случае выведите 2.

Примеры

<code>input.txt</code>	<code>output.txt</code>
100 77 10	2

Задача В. Сдача

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

У тети Поли сегодня первый день на новой работе. Теперь она работает кассиром в супермаркете знаменитой Берляндской сети магазинов «Пучок». Надев опрятную зеленую униформу и густо накрашившись, тетя Поля в восемь утра заняла свое место в полной готовности встретить первого покупателя. Каким он будет? Высоким опрятно одетым джентельменом или прокуренным рабочим, заскочившим купить пачку сигарет по пути на завод, или зеленым мальчишкой, забежавшим за жвачкой, или же бабушкой, купившей десяток яиц и буханку хлеба? Тетя Поля была готова очаровать любого. И вот настал долгожданный момент. Элегантно перекинув через руку корзинку, небрежно доставая бумажник из кармана, перед ней стоял Он. Неторопливыми, полными изящества движениями тетя Поля поочередно поднесла товары из корзинки к сканеру. «С Вас x бурлей», произнесла она. Покупатель достал из бумажника и протянул ей хрустящую розовую бумажку в 1000 бурлей. «Ой, я должна дать ему сдачу», внезапно поняла тетя Поля. Ее взгляд упал на ящик кассового аппарата, в котором были аккуратно разложены имевшие хождение в то время купюры номиналами в 100, 50, 20, 10, 5 и 1 бурлей. В силу своей неопытности тетя Поля знала только один метод выдачи сдачи: набрать по максимуму 100-бурлевых купюр, затем по максимуму 50-бурлевых, и так далее до купюр в один бурль. "Ой, а вдруг не сойдется задрожала тетя Поля, «я же ему не понравлюсь». Мы знаем конец этой истории, а вам предлагаем решить следующую задачу: по заданному числу x и количеству купюр каждого номинала в кассе, выясните, смогла ли тетя Поля набрать сдачу таким алгоритмом, и, если да, то каким количеством купюр?

Формат входного файла

В первой строке записано целое число x ($1 \leq x \leq 1000$). Во второй строке записано 6 целых неотрицательных чисел $a_{100}, a_{50}, a_{20}, a_{10}, a_5, a_1$, не превосходящих 1000 — количества купюр соответствующих номиналов.

Формат выходного файла

Если описанным образом набрать необходимую сумму для сдачи невозможно, выведите «No solution». В противном случае выведите одно число — общее количество купюр всех номиналов, использованных для сдачи.

Примеры

<code>input.txt</code>	<code>output.txt</code>
964 0 0 1 1 1 1	4
877 2 2 2 0 2 3	5

Задача С. Исправление успеваемости (вручную)

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Как вы наверняка знаете, Вовочка не является примерным учеником. За учебный год он получил много разных оценок, и теперь не знает как показать дневник своему отцу. Отец Вовочки считает *хорошими* оценки 4 и 5, а все остальные считает *плохими*. Дело в том, что Вовочке было обещано лишить его отдыха в Летней школе по информатике в том случае, если он когда-либо получит три или более плохие оценки подряд. Вовочка уже давно научился стирать оценки из дневника, но с каждой стертой оценкой вероятность раскрытия такой секретной операции возрастает. В настоящий момент перед ним стоит непростая задача — удалить наименьшее количество оценок из имеющейся последовательности так, чтобы после удаления никакие три плохие оценки не шли подряд.

Помогите Вовочке решить задачу и подарить ему шанс участия в Летней школе по информатике.

Формат входного файла

В первой строке входного файла записано целое число n ($1 \leq n \leq 5000$), где n — количество оценок, полученных Вовочкой за год. Вторая строка содержит последовательность n целых чисел от 1 до 5. Оценки заданы в хронологическом порядке.

Формат выходного файла

В первую строку выведите t — наименьшее количество оценок, которые надо удалить из последовательности. Во вторую строку выведите последовательность оценок, которая получается после оптимального исправления успеваемости. Если возможных оптимальных решений несколько — выведите любое.

Примеры

input.txt	output.txt
6 1 2 3 5 5 3	1 1 2 5 5 3
4 1 4 1 5	0 1 4 1 5

Задача D. Кола

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Однажды Вася захотел купить бутылку своей любимой Pepsi Light в старом автомате по продаже колы. Известно, что внутреннее устройство автомата представляет собой прямоугольную таблицу размером $n \times m$. Некоторые ячейки пусты, в некоторых содержатся препятствия двух типов ('/' и '\'), а в одной из ячеек автомата изначально находится бутылка. После покупки она начинает падать вертикально вниз по следующим правилам.

- Если в ячейке, мимо которой пролетает кола, пусто, то она падает вниз.
- Если бутылка выпадает вниз с самого нижнего ряда, то она попадает в поддон и Вася может ее забрать.
- При столкновении с препятствием '/' ('\') бутылка, не меняя своей высоты, скатывается влево (вправо) на один столбец и продолжает падать вертикально вниз, если это возможно.
- Если движение бутылки невозможно, по причине наличия стенки в направлении перемещения, то она прекращает свое движение. Также бутылка прекращает движение если при перемещении от препятствия, в ячейке куда она должна переместиться расположено препятствие противоположного вида.
- Если бутылка скатывается к препятствию типа совпадающего с текущим, то она не застревает и продолжает движение.

Помогите Васе определить, попадет ли бутылка с колой в поддон. В случае положительного ответа необходимо найти номер столбца, из которого она упадет на поддон.

Формат входного файла

В первой строке входного файла заданы два целых числа n и m ($1 \leq n, m \leq 100$). Далее следует n строк по m символов — описание автомата по продаже колы: '.' (точка) означает отсутствие препятствия, 'P' — начальное положение бутылки колы, '/' и '\' — препятствия. Гарантируется, что описание автомата не содержит никаких других символов, а символ 'P' встречается ровно один раз.

Формат выходного файла

Если бутылка застрянет и не попадет в поддон, выведите «-1», иначе выведите номер столбца, из которого она упадет на поддон. Столбцы нумеруются слева направо начиная с 1.

Примеры

input.txt	output.txt
2 3 ./P ../	2
2 2 .P \	-1
2 2 .P.. .\.. ./.. /...	-1

Note

В третьем примере из условия бутылка из ячейки, помеченной буквой 'P' сначала перемещается вниз, затем вправо, вниз, влево, вниз, влево, вниз. На следующем шаге при попытке переместиться влево она упирается в стенку и застревает.

Задача Е. Экскурсия

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Однажды группа учащихся, состоящая из школьников (a человек) и школьниц (b человек) старших классов, собралась с учителем на экскурсию по родному городу. С самого начала мероприятия школьники пошли гулять по городу самостоятельно. Школьницы, наоборот, послушно присоединились к учителю. При этом в ходе экскурсии в моменты приближения к модным бутикам некоторые школьницы покидают группу, увлекаясь поиском и примеркой одежды до самого вечера, т. е. не возвращаясь назад к учителю. Румяные и повеселевшие школьники, напротив, возвращаются к учителю и с удовольствием сопровождают его до конца экскурсии. Время от времени добросовестный учитель аккуратно пересчитывает всю сопровождающую его группу учащихся и записывает численность группы в определенные моменты времени в свой блокнот.

Для оценки охвата учащихся внеклассной работой учитель поручил вам по сделанным им записям определить возможное расписание присоединения и ухода из экскурсионной группы учащихся.

Формат входного файла

В первой строке входного файла заданы два целых числа a и b ($1 \leq a, b \leq 100$) — число школьников и школьниц соответственно. Вторая строка содержит целое число n ($1 \leq n \leq 100$) — количество записей в блокноте учителя. Третья строка входного файла содержит n целых неотрицательных чисел, не превосходящих 200 — записи учителя в порядке их следования в блокноте.

Формат выходного файла

Если в записях учителя содержится ошибка, и не существует ни одного расписания, подходящего под эти данные, выведите «ERROR». Иначе выведите в выходной файл n строк. Строка с номером i должна содержать два целых неотрицательных числа — количество присоединившихся к группе школьников и количество покинувших группу школьниц соответственно перед тем, как учитель пересчитал учащихся в i -тый раз. Если возможных расписаний несколько, выведите любое.

Примеры

input.txt	output.txt
3 3 2 2 3	1 2 2 1
3 3 3 1 2 5	ERROR
2 2 1 2	0 0

Задача F. Елки

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

В процессе подготовки к столетию Берляндского государственного университета было решено засадить елками прямоугольную площадку размером n на m метров. Площадка была разбита на $n \times m$ одинаковых квадратных клеток с длиной стороны 1 метр.

Расположение деревьев тщательно выбиралось, и в итоге был выработан план. Для каждой клетки этой площадки определили, сколько елок должно быть посажено в этой клетке. Оказалось, что все эти количества от 1 до 5, включительно.

Для реализации этого плана была приобретена машина по посадке елок. За одну операцию можно сделать следующее: указать этой машине какую-то клетку (двумя её координатами: от 1 до n , и от 1 до m соответственно), и в этой клетке будет посажена елка. Однако уже после покупки выяснилось, что машина обладает следующей неприятной особенностью. При каждой операции эта машина посадит елку не только в указанную клетку, но и в соседние клетки площадки, в которые уже посажено хотя бы по одной елке. Соседними для клетки называются те, которые имеют с ней одну общую сторону. Кроме того, по техническим причинам машине нельзя дважды указывать одну и ту же клетку.

В итоге очень важная часть плана по подготовке к столетию под угрозой срыва. К вам обратились с просьбой написать программу, которая для заданного плана найдёт последовательность nm операций, приводящую к требуемой посадке елок, или сообщит, что сделать это невозможно.

Формат входного файла

В первой строке входного файла записаны два целых числа n и m , задающих размеры поля ($1 \leq n, m \leq 200$). Далее идут n строк по m целых чисел от 1 до 5, задающих, сколько елок должно быть посажено в каждой клетке поля. Числа в строках разделяются пробелами.

Формат выходного файла

Если решение существует, выведите nm строк по два числа в каждой: номер строки и номер столбца, в которой сажается елка на очередном шаге.

Если решения не существует, выведите строку «No solution» (без кавычек).

Примеры

input.txt	output.txt
1 2 1 2	1 2 1 1
1 3 1 3 1	1 2 1 3 1 1
3 3 2 4 2 2 1 2 3 2 3	1 2 3 1 3 3 1 1 1 3 3 2 2 1 2 3 2 2
2 1 1 1	No solution

Задача G. Поступай правильно

Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайт

В Берляндском государственном университете новые правила для поступления. Для каждой из n специальностей опубликован список тестов, и балл абитуриента на эту специальность складывается из набранных баллов по этим тестам.

В Берляндии утвержден список из k тестов и каждый абитуриент имел возможность пройти эти тесты. В настоящий момент каждый бывший школьник имеет некоторый балл по каждому из k тестов (если он этот тест не сдавал, то его балл считается равным 0).

В соответствии с текущими правилами поступления абитуриенты могут подавать заявления на произвольное количество специальностей. Если абитуриент подал заявление на несколько специальностей, то считается, что порядок подачи соответствует его предпочтениям (первая специальность — самая желанная и т.д.).

Каждая специальность имеет ограничение w_i — максимальное количество студентов, которые могут на нее поступить. Следующая процедура повторяется несколько раз для определения того, кто куда в результате поступил. Для каждой специальности создают список из w_i абитуриентов (из подавших на нее заявки) с наибольшим количеством баллов (то есть абитуриентов упорядочивают по количеству набранных баллов и берут w_i лучших). Если на специальность всего было подано заявлений менее w_i , то в список попадают все подавшие на специальность заявления. Если существует неоднозначность в выборе w_i абитуриентов, то при сортировке при равенстве баллов лучшим считается тот абитуриент, кто подал документы раньше (порядок подачи документов совпадает с порядком описания студентов во входных данных).

Далее каждый абитуриент из тех специальностей, в списки которых он попал, выбирает ту, которая для него более предпочтительная. В результате некоторая часть абитуриентов становится студентами, и некоторая часть мест на специальностях становится занятой. После этого из w_i каждой специальности вычитается количество занятых мест по результатам текущего распределения и процедура повторяется заново, начиная с построения списков. Повторение происходит до тех пор, пока оно имеет смысл, то есть в результате которого хотя бы один абитуриент становится студентом.

Напишите программу, которая для каждой специальности подготовит список поступивших на нее студентов.

Формат входного файла

В первой строке входного файла содержатся два числа n ($1 \leq n \leq 20$) и k ($1 \leq k \leq 10$), где n — это количество специальностей, а k — количество тестов соответственно. Далее в следующих n строках содержатся описания специальностей.

Каждая из строк содержит целые числа w_i ($1 \leq w_i \leq 50$), c_i ($1 \leq c_i \leq k$) и последовательность из c_i различных чисел b_{ij} ($1 \leq b_{ij} \leq k$), где w_i — максимальное число студентов на i -ой специальности, c_i — количество тестов, которые учитываются при приеме на эту специальность и b_{ij} — номера тестов.

Далее содержится число s ($1 \leq s \leq 200$), s — общее количество абитуриентов. Далее следует $2s$ строк, описывающих абитуриентов. Абитуриенты перечислены в порядке подачи ими документов. Информация об одном абитуриенте содержится в двух строках. Первая из этих двух строк содержит k чисел — оценки за тесты (это целые числа от 0 до 100, включительно). Вторая строка содержит целое число q_i ($1 \leq q_i \leq n$), q_i — количество специальностей, куда абитуриент подал заявления, и затем q_i различных чисел от 1 до n — номера специальностей в порядке предпочтения студентов.

Формат выходного файла

Выведите n строк. Строки должны описывать результаты зачисления по специальностям от первой до n -ой. Выведите в каждую строку количество зачисленных на нее абитуриентов d_i , и последовательность из d_i различных чисел — список поступивших в любом порядке. Для обозначения

абитуриентов используйте их номера от 1 до s , определяемые порядком их следования во входных данных.

Примеры

input.txt	output.txt
2 1 1 1 1 1 1 1 2 1 2 1 2 0 2 1 2	1 1 1 2
3 2 7 2 1 2 1 2 1 2 23 1 2 3 61 64 2 3 1 4 29 2 2 3 63 39 2 1 3	1 3 1 2 1 1

Note

В первом тесте всего есть два студента, каждый из них поступает на обе специальности, но каждому из них приоритетнее поступить на первую специальность. Поскольку на каждой специальности имеется только одно место, и у первого студента 1 балл, а у второго 0, то первый из них поступает на первую специальность, а второму приходится довольствоваться второй.

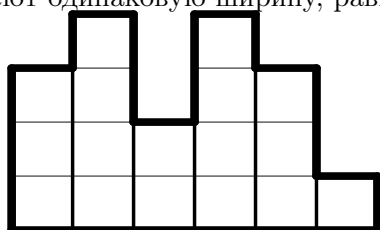
Задача Н. Доски из забора

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

В тяжелые времена мирового экономического кризиса особое значение приобретает так называемое перераспределение ресурсов без увеличения расходов. В частности, это означает использование уже имеющихся материалов вместо покупки новых. Так, для ремонта XXV корпуса Берляндского государственного университета понадобились доски. Поскольку денег на новые доски нет, то было решено использовать часть досок из забора, окружающего местный заброшенный завод по заготовке рогов и копыт. Однако эта инициатива встретила возмущение местных жителей, которые годами любовались видом этого прекрасного забора. Строители уже было хотели по традиции наплевать на мнение жителей, однако в дело неожиданно вмешалось министерство эстетики, которое встало на сторону обитателей окрестных домов. В спешном порядке созданная комиссия по согласованию пришла к следующему заключению: поскольку эстетичность забора определяется прежде всего его периметром (см. рисунок), то из забора разрешается забрать какие-то доски таким образом, что если плотно сдвинуть оставшиеся доски, то новый периметр забора будет не меньше, чем половина периметра исходного забора.

При данных ограничениях университету требуется изъять из забора набор досок с максимальной суммарной высотой.

Забор представляет собой выровненную по нижнему краю последовательность досок. Все доски имеют одинаковую ширину, равную 1 и целочисленные высоты.



Периметр забора считается как периметр фигуры, полученной объединением прямоугольников, соответствующих доскам. Например, периметр забора, изображенного на рисунке, выделен жирной линией и его длина равна 24.

Формат входного файла

В первой строке записано целое число n ($1 \leq n \leq 50$) — количество досок в заборе. Во второй строке записано n целых чисел h_i — высоты досок в заборе ($1 \leq h_i \leq 100$). Доски перечислены слева направо так, как они образуют забор.

Формат выходного файла

В первой строке выведите s — максимальную суммарную высоту набора досок, которые можно унести без нарушения описанных в условии правил. Во второй строке выведите целое число k — число досок в искомом наборе. В третьей строке выведите k чисел — номера досок, которые следует использовать для ремонта. Доски нумеруются с 1 в том порядке, в котором они перечислены во входном файле. Номера выводите в любом порядке. Если решений несколько, то выведите любое.

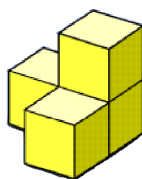
Примеры

<code>input.txt</code>	<code>output.txt</code>
6	12
3 4 2 4 3 1	4
	1 3 4 5

Задача I. Пирамида

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунды
Ограничение по памяти: 64 мегабайт

Рассматривается невыпуклая фигура типа «уголок» в пространстве, состоящая из четырех одинаковых кубов. Один из кубов имеет общую грань с любым из остальных, а также существует единственная точка, которая принадлежит всем четырем кубам и для каждого куба является вершиной (см. рис). Каждый «уголок» имеет две характеристики: цвет и целочисленную длину стороны каждого из кубов (размер), из которых состоит одна фигура, при этом длина равна степени двойки, поэтому задается только показатель степени. Задача состоит в том, чтобы сконструировать большой «уголок» из данных фигур, используя минимально возможное количество цветов, если задана длина составляющего этот «уголок» куба.



Формат входного файла

В первой строке входного файла содержится два числа, разделенных одним пробелом: c — число цветов и k — показатель степени двойки, которая равна размеру большого «уголка» ($1 \leq c, k \leq 1000$). Следующие c строк содержат по k чисел a_{ij} , a_{ij} равно количеству «уголков» цвета i и размера 2^j ($0 \leq a_{ij} \leq 1000$ для $1 \leq i \leq c, 0 \leq j < k$).

Формат выходного файла

В первую строку выходного файла следует вывести n — количество типов фигур, которые использовались в построении. В следующие n строк выведите a_i, b_i, c_i , соответственно цвет, размер, количество «уголков» этого типа, разделяя эти числа пробелом. Строки разрешается выводить в произвольном порядке. Каждый тип «уголка» должен выводиться не более одного раза. Если решений несколько, то выведите любое. Если решений не существует, то выведите «NO SOLUTION» (без кавычек).

Примеры

input.txt	output.txt
1 1 8	1 1 0 8
2 2 1 7 7 0	3 1 1 7 1 0 1 2 0 7

Задача J. Двери

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 мегабайт

Недавно в Берляндии открылся Музей Современной Техники. Если кому-то из вас уже довелось его посетить, то вы, наверное, заметили, что технические достижения в музее используются не только как экспонаты. Автомат продает билеты, управляемый лифт позволяет подниматься к потолку, чтобы можно было получше рассмотреть огромные машины, освещение автоматически отключается в залах, откуда вышли посетители... Находясь среди такого разнообразия техники, даже странно слышать скрип старых входных дверей.

По распоряжению директора музея были закуплены две автоматические двери, которые работают по следующему принципу. При установке двери задается ее период ожидания t , который может принять любое целочисленное значение от 1 до 10^9 секунд. Если в момент времени p через дверь проходит человек, то она открывается в момент времени $p - t$ и закрывается в момент времени $p + t$. Исключение составляет случай, когда несколько человек идут подряд, «цепочкой», с промежутком между соседними людьми в «цепочке» не более $2t$ секунд. Тогда дверь открывается только один раз, за t секунд до того, как через нее пройдет первый человек в «цепочке», и закрывается через t секунд после того, как через нее пройдет последний человек.

Очень важно правильно отрегулировать двери. Ведь если они будут слишком часто открываться и закрываться, их постоянное хлопанье будет мешать посетителям. С другой стороны, если обе двери будут долго стоять открытыми, возникнет сквозняк и кто-нибудь обязательно простудится.

Более формально: вам дан список моментов времени $p_1 < p_2 < \dots < p_n$, в которые люди проходили через первую дверь, и моменты времени $q_1 < q_2 < \dots < q_m$, когда люди проходили через вторую дверь. Необходимо по имеющейся статистике так выбрать t_1 и t_2 — периоды ожидания первой и второй дверей соответственно, чтобы выполнялись два условия.

1. Суммарное количество раз, когда двери открываются, должно быть минимальным.
2. Не существует непрерывного промежутка времени, длительности строго больше d секунд, в течение которого обе двери стоят открытыми.

Формат входного файла

В первой строке входного файла заданы три целых числа n , m и d ($1 \leq n, m \leq 5000$, $1 \leq d \leq 10^9$). Вторая строка содержит числа p_i , третья строка — числа q_i , заданные в порядке возрастания ($1 \leq p_i, q_i \leq 10^9$).

Формат выходного файла

В первой строке выходного файла выведите два числа t_1 и t_2 . Если решений несколько, разрешается выводить любое. Если решения не существует, выведите «No solution».

Примеры

input.txt	output.txt
3 2 4 1 6 13 7 11	3 2

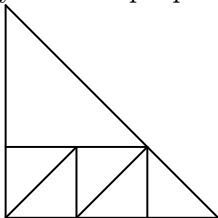
Задача К. Замощение площади

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 мегабайт

В честь 100-летия Берляндского государственного университета было принято решение замостить новую актовую площадь университета брусчаткой из гранатового амфиболита. По традиции, был объявлен конкурс на лучший проект замощения. Актовая площадь представляет собой равнобедренный прямоугольный треугольник с длиной катета n . Для того, чтобы в конкурсе могли принять участие не только художники, но и программисты, была введена система координат таким образом, что вершина треугольника, противолежащая гипотенузе, оказалась в точке $(0, 0)$, а две другие вершины оказались в точках $(0, n)$ и $(n, 0)$. Для замощения разрешается использовать плитки, представляющие собой равнобедренные прямоугольные треугольники с целочисленной длиной катета, причем располагать их во введенной системе координат можно только так, чтобы один из катетов был параллелен оси абсцисс, а другой был параллелен оси ординат. В комиссию поступило такое количество проектов, что стало ясно, что вручную все их рассмотреть невозможно. Тогда председателю комиссии пришла в голову светлая мысль: сократить число проектов, автоматически отбросив все некорректные замощения. Замощение считается некорректным, если для него верно хотя бы одно из перечисленных условий:

- используются треугольники, отличные от описанных в условии,
- какие-то два треугольника пересекаются,
- какой-то треугольник не содержится полностью внутри актовой площади,
- какая-то часть треугольника площади осталась не покрытой треугольником из набора.

В таком виде задача стала понятна местным программистам, и они за час написали проверяющую эти условия программу. А вы способны повторить их подвиг?



Пример корректного замощения

Формат входного файла

Во входном файле задано несколько наборов входных данных. В первой строке записано целое положительное число $t \leq 10$ — количество тестовых наборов. Далее идут сами тестовые наборы. В первой строке каждого набора записано два целых числа n и m — длина катета треугольника актовой площади и число треугольников в замощении ($1 \leq n \leq 25000, 1 \leq m \leq 100000$). Далее в каждом тесте идет по m строк, в каждой из которых записано по 6 целых неотрицательных чисел $x_1, y_1, x_2, y_2, x_3, y_3$, не превосходящих 25000 — координаты вершин треугольников в замощении. Гарантируется, что все треугольники имеют ненулевую площадь. Суммарное число треугольников по всем тестовым наборам не превосходит 200000.

Формат выходного файла

Для каждого тестового набора выведите в отдельной строке «YES», если приведенное замощение корректно, и «NO» в противном случае.

Примеры

input.txt	output.txt
3	YES
1 1	NO
0 0 0 1 1 0	YES
2 3	
0 0 1 1 0 1	
0 1 1 1 0 2	
1 1 2 0 1 0	
3 6	
0 0 0 1 1 1	
0 0 1 0 1 1	
1 0 1 1 2 1	
1 0 2 0 2 1	
2 0 2 1 3 0	
0 1 2 1 0 3	