

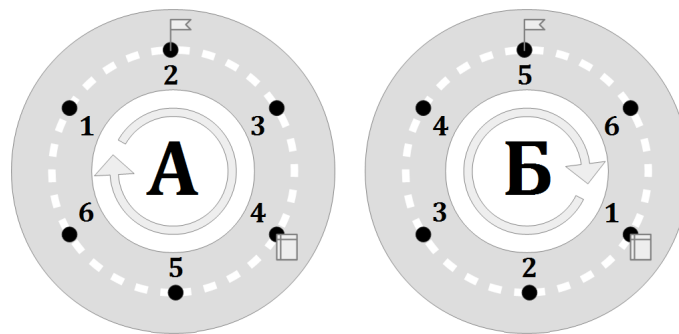
Задача А. Подсчёт столбов

Имя входного файла: `posts.in`
Имя выходного файла: `posts.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Будущие программисты Андрей и Борис вчера впервые поехали кататься с родителями по новой кольцевой дороге. Каждый из них выехал на дорогу в определённом месте, сделал полный круг и вернулся домой. От скуки они оба считали фонарные столбы, расположенные посередине дороги между встречными полосами движения, так что все N фонарных столбов у каждого из мальчиков получили номера от 1 до N . Но само значение N они не запомнили. При этом два столба обоим мальчикам запомнились особенно: на одном из них висел яркий плакат ко Дню города, а на другом — флаг Москвы. Каждый из мальчиков записал себе в тетрадку номер каждого из этих двух столбов.

Сегодня обе семьи, Андрея и Бориса, пошли на выставку кошек, и там мальчики, обсудив свои поездки, задались вопросом: сколько же всего фонарных столбов на новой кольцевой дороге? Единственное, что они смогли выяснить, в одном ли направлении ехали они по дороге.

Так сложилось, что Андрей — ваш младший брат, поэтому именно вам предстоит ответить на вопрос мальчиков. У вас есть серьёзное подозрение, что может не получиться однозначно найти ответ, а мальчики боятся больших чисел, поэтому вы решили сказать им лишь минимальное из возможных значений числа N .



В этом примере $N = 6$, $A_p = 4$, $A_f = 2$, $B_p = 1$, $B_f = 5$.

Формат входного файла

Первая строка входного файла содержит единственное целое число D , которое равно 1, если мальчики ехали в одном направлении, и -1 , если в разных. Вторая строка содержит 4 натуральных числа A_p , B_p , A_f , B_f , каждое из которых не превосходит 10^9 : A_p — номер столба с плакатом в нумерации Андрея, B_p — номер этого столба в нумерации Бориса, A_f — номер столба с флагом в нумерации Андрея, B_f — номер этого столба в нумерации Бориса. Соседние числа в строке разделены одним пробелом. Плакат и флаг могли оказаться на одном столбе — в этом случае каждый из мальчиков должен был бы получить два одинаковых числа, т. е. $A_p = A_f$ и $B_p = B_f$.

Формат выходного файла

Выведите единственное натуральное число N — минимально возможное количество столбов. Если мальчики где-то ошиблись, и таких чисел, как у них, не могло получиться ни при каком значении N , выведите число -1 .

Примеры

<code>posts.in</code>	<code>posts.out</code>
1 4 1 2 5	6
-1 4 9 4 7	-1

Задача В. В гору пойдет!

Имя входного файла: `tourism.in`
Имя выходного файла: `tourism.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Команда туристического клуба «В гору пойдёт!» только что вернулась из очередного похода. Прямо сейчас участники экспедиции с жаром спорят о том, какой же горный хребет они покорили.

Достоверно известно, что на маршруте было N стоянок, причём все — на разной целочисленной высоте от 1 до N над уровнем моря. Альпинисты заблаговременно прибыли на место первой стоянки, а потом шли по маршруту в течении $N - 1$ дня: в первый день они шли от 1-й стоянки до 2-й, во второй — от 2-й до 3-й и так далее, пока в последний день не совершили переход от стоянки под номером $N - 1$ до стоянки под номером N , завершив этим свой маршрут.

Участники экспедиции пытаются понять, на какой высоте находилась каждая стоянка. В их распоряжении имеются записи в журнале, содержащие информацию о том, в какой день они шли в гору, а в какой — спускались.

Помогите альпинистам! Подскажите им хоть какой-нибудь вариант маршрута, не противоречащий записям в журнале.

Формат входного файла

В первой строке входного файла содержится натуральное число N — количество стоянок на маршруте ($2 \leq N \leq 1\,000\,000$). Во второй строке входного файла содержится последовательность длины $N - 1$, состоящая из знаков «<» и «>». Если на i -м месте в этой последовательности стоит знак «<», то в i -й день альпинисты шли в гору, а знак «>» означает, что в i -й день они спускались.

Формат выходного файла

Выведите строку, содержащую N различных целых чисел от 1 до N , разделённых пробелами, — маршрут, по которому могли пройти альпинисты. Маршрут описывается высотами стоянок в том порядке, в котором их посетили участники экспедиции.

Примеры

<code>tourism.in</code>	<code>tourism.out</code>
5 >><>	3 2 1 5 4

Задача С. Проще простого

Имя входного файла: `primes.in`
Имя выходного файла: `primes.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Что может быть проще простого числа? Казалось бы, объяснить, что такое простое число, можно даже человеку, совершенно далёкому от математики: целое число называется простым, если оно не меньше двух и не делится ни на какое целое положительное число, кроме единицы и самого себя. Это определение будет понятно даже третьекласснику, только-только познакомившемуся с делением. Что может быть проще? Но, как часто случается в математике, за кажущейся простотой определения скрывается очень глубокая теория с множеством нетривиальных фактов, многие из которых остаются недоказанными и по сей день.

Прочитав популярную книгу Д. Дербишира «*Простая одержжимость*», Леопольд узнал следующий занятный факт. Оказывается, существует *Теорема о распределении простых чисел*, гласящая, что количество простых чисел, не превышающих N , можно очень точно оценить как $\frac{N}{\ln(N)}$. Например, начиная с $N > 5000$, эта формула даёт ошибку, не более чем в 15% от реального значения. Более того, с ростом N относительная погрешность такой оценки падает, стремясь к нулю.

Леопольд крайне заинтересовался простыми числами и связанной с ними теорией. Он решил выдвинуть какую-нибудь не менее важную и серьёзную гипотезу, а потом доказать её, и назвать полученный факт теоремой Леопольда. Для этого ему нужна помощь в отыскании закономерностей, описывающих простые числа. Он просит вас написать для него программу, которая ищет для него Q отрезков, i -й из которых состоит из L_i последовательных натуральных чисел и содержит определённое количество K_i простых чисел. Для простоты анализа он просит вас ограничиться в поисках первыми **десятью миллионами** чисел. Помогите ему, и, возможно, вам с ним удастся оставить след в истории!

Формат входного файла

В первой строке входного файла задано целое число Q ($1 \leq Q \leq 100\,000$) — количество отрезков, которые требуются Леопольду.

В каждой из последующих Q строк задано по два целых числа L и K ($7000 \leq K \leq L \leq 100\,000$). **Обратите внимание**, подобные ограничения даны не случайно: Леопольд знает, что нередко закономерности начинают проявляться только при больших значениях.

Формат выходного файла

На каждый запрос Леопольда выведите в отдельной строке начальное и конечное число требуемого отрезка, либо -1 , если его не существует среди первых десяти миллионов чисел. Если требуемых отрезков несколько, выведите любой.

Примеры

<code>primes.in</code>	<code>primes.out</code>
3	-1
8000 8000	3632 83631
80000 7654	1482488 1582487
100000 7000	

Задача D. Матрёшки

Имя входного файла: `matrioshkas.in`
Имя выходного файла: `matrioshkas.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Склад завода по изготовлению матрёшек переполнен! Нужно как-то освободить место, поэтому директор завода принял волевое решение продать абсолютно всё, что там лежит. Больше всего места на складе занимают заготовки для матрёшек — нераскрашенные статуэтки целых положительных размеров, которые можно вставлять друг в друга. Увы, в таком неприглядном виде покупать их никто не хочет.

К счастью, завод сотрудничает с союзом художников по матрёшкам. В частности, был заключён договор, позволяющий заводу заказывать роспись матрёшек. В договоре указано, что матрёшка — это упорядоченный набор из M статуэток ($1 \leq M$) размеров a_1, a_2, \dots, a_M , где $a_1 + 1 = a_2$, $a_2 + 1 = a_3$, \dots , $a_{M-1} + 1 = a_M$. Там же прописано, что стоимость раскрашивания одной матрёшки равна одному тугрику, при этом количество статуэток, входящих в матрёшку, значения не имеет.

Получается, что для того чтобы продать всё содержимое склада, нужно сначала собрать заготовки в матрёшки и заказать роспись полученных матрёшек у художников. Помогите директору завода сделать это, потратив как можно меньше тугриков!

Формат входного файла

В первой строке входного файла содержится число N — количество заготовок на складе ($1 \leq N \leq 2 \cdot 10^5$). Во второй строке содержатся N целых чисел s_1, s_2, \dots, s_N , где s_i — это размер i -й заготовки ($1 \leq s_i \leq 2 \cdot 10^5$).

Формат выходного файла

В первой строке выходного файла выведите целое число T — минимальное количество тугриков, которое нужно заплатить художникам.

В следующих T строках выведите описания матрёшек, которые завод закажет у союза художников. Описание матрёшки состоит из строки, содержащей два разделённых пробелом числа, где первое число — размер самой маленькой статуэтки в матрёшке, а второе число — размер самой большой статуэтки в матрёшке.

Примеры

<code>matrioshkas.in</code>	<code>matrioshkas.out</code>
5	3
3 2 1 2 5	2 2
	1 3
	5 5

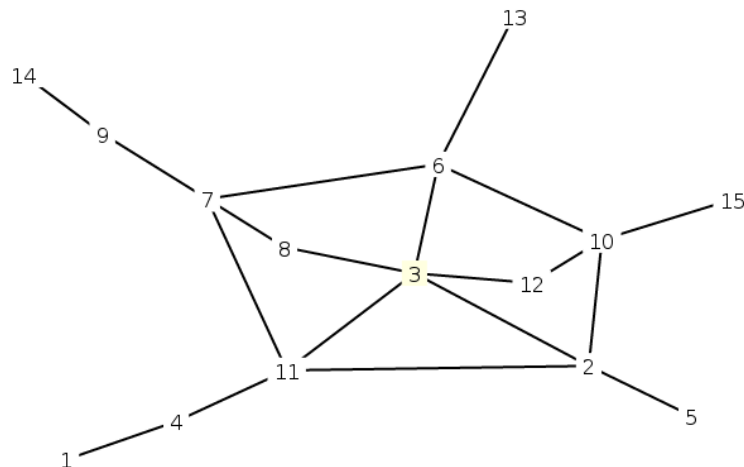
Задача Е. Карта метро

Имя входного файла: `map.in`
Имя выходного файла: `map.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В одном уездном городе Эн было решено построить собственное метро. Все силы города были мобилизованы на выкапывание станций и прокладку подземных путей дедовскими лопатами.

Вся эта история нас бы совершенно не интересовала, если бы однажды в мэрию города не пришло письмо из далёкой страны Емакира. Оказалось, что компания Алпер подозревает администрацию уездного города в нарушении их патента на `jMetro` и грозитя возбудить против города Эн дело. Согласно патенту, `jMetro` — это метро, в котором:

- существует ровно одна узловая станция, в которой начинаются все радиальные линии метро, и это единственная станция, принадлежащая хотя бы двум радиальным линиям;
- существует ровно одна кольцевая линия, все станции которой лежат на радиальных линиях, причём на каждой радиальной линии лежит ровно одна станция кольцевой линии;
- кольцевая линия не проходит через узловую станцию;
- кольцевая линия не проходит через конечные станции радиальных линий.



Поскольку компания Алпер известна своими необоснованными обвинениями в нарушениях патентов, мэрия города хочет проверить правомочность заявления компании.

Формат входного файла

В первой строке заданы два числа N и M ($1 \leq N, M \leq 2 \cdot 10^5$) — количество станций метро и перегонов между ними. Следующие M строк содержат описания перегонов: каждая из них содержит по два числа — номера станций, между которыми есть перегон. По каждому перегону составы могут ездить как в одну, так и в другую сторону. Между любыми двумя станциями существует не более одного перегона. Никакой перегон не соединяет станцию саму с собой.

Формат выходного файла

Выведите «YES», если метро уездного города Эн нарушает патент `jMetro`, и «NO» в противном случае.

Примеры

map.in	map.out
15 19 1 4 4 11 2 10 3 2 8 7 7 6 12 10 15 10 11 2 14 9 6 13 7 9 7 11 2 5 8 3 6 10 3 6 11 3 12 3	YES
5 4 2 1 2 3 2 5 2 4	NO

Примечание

Первый пример соответствует рисунку из условия.

Задача F. Свободная ячейка

Имя входного файла: `solitaire.in`
Имя выходного файла: `solitaire.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Лена — страстная любительница пасьянсов. Больше других ей нравятся стандартные пасьянсы на её стареньком рабочем компьютере под управлением доисторической операционной системы «вай-вай-вай-крософт миндоус ХР™», из которых особенное предподчтение она отдаёт «Свободной ячейке» (другое название этого пасьянса — «Солитер»). Все стандартные расклады уже давно решаются Леной за минуту, поэтому в свободное время она придумывает, как бы усложнить правила игры.

Она предлагает вам помочь ей со следующей постановкой. В её игре участвуют K карт одной масти достоинствами от 1 до K . Изначально они лежат в одном из слотов в следующем порядке при перечислении снизу вверх: $1, K, K-1, K-2, \dots, 3, 2$. Цель её пасьянса — переложить все карты кроме единицы в один из свободных слотов в порядке $K, K-1, \dots, 3, 2$, используя N дополнительных свободных слотов для стопок и F слотов для одиночных карт.

Правилами разрешается переложить верхнюю карту любой стопки в любой свободный слот, либо на карту на единицу большего достоинства при условии, что та не лежит в одиночном слоте. Обратите внимание, правилами **запрещается** перекладывать карту достоинством 1, которая лежит внизу изначальной стопки.

Лена не может определиться с тем, сколько именно карт может лежать в изначальной стопке и сколько должно быть слотов каждого вида. Она просит вас определить для некоторых наборов N , F и K , раскладывается ли пасьянс.

Формат входного файла

В первой строке входных данных находится единственное число $1 \leq T \leq 10^5$ — количество наборов, для которых нужно решить задачу.

Каждая из следующих T строк содержит по три целых числа N, F, K ($1 \leq N \leq 10^6$, $0 \leq F \leq 4$, $2 \leq K \leq 2 \times 10^9$).

Формат выходного файла

Выведите для каждого набора одно слово — “YES”, если пасьянс при очередных значениях сходится, либо “NO” в противном случае.

Примеры

<code>solitaire.in</code>	<code>solitaire.out</code>
2	YES
3 0 5	NO
2 0 5	
3	YES
2 1 5	NO
2 4 1000	YES
4 0 6	

Примечание

Пояснение к первому примеру. В обоих случаях у нас есть три свободных слота для формирования стопок и нет дополнительных слотов для одиночных карт. В первом случае начальная стопка состоит из пяти следующих карт 1, 5, 4, 3, 2 (в перечислении снизу вверх). Такой пасьянс сходится: например, сначала можно за три шага сложить стопку 3, 2 в одном из свободных слотов, затем положить карты 4 и 5 в два других слота, а затем уже собрать из этих четырёх карт стопку. С другой стороны, при двух свободных слотах 5 карт переложить уже нельзя.

Задача G. Развивающие игры

Имя входного файла:	games.in
Имя выходного файла:	games.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

У маленькой Даши скоро день рождения. Целыми днями она обдумывает, кого бы ей пригласить, какое платье надеть, и пытается угадать, что же ей подарят. А в это время её старший брат Серёжа занят гораздо более печальными мыслями — мама сказала, что именно он будет развлекать детей после застолья.

Подойдя к этому делу со всей свойственной ему ответственностью, Серёжа разработал игру, призванную повысить навыки детей в работе со строками. Правила этой игры таковы:

- N человек сидят за круглым столом на N мест. Перед каждым игроком лежит поднос с одинаковыми буквами-наклейками.
- Перед началом игры ведущий называет некоторое целое число K .
- Каждый играющий хватается букву, лежащую на подносе, перед которым он сидит, клеит ее себе на лоб, после чего все пробегают K позиций вдоль стола. Если $K > 0$, то бег осуществляется по часовой стрелке (от 1 к N), если же $K < 0$, то играющий бежит в направлении против часовой стрелки (от N к 1). Так, например, если за столом сидят 5 человек, и ведущий называет число -2 , то рассадка меняется с $(1, 2, 3, 4, 5)$ на $(3, 4, 5, 1, 2)$.
- Добежав до новой позиции, играющий берет букву, лежащую на подносе около этого места, и клеит ее себе на лоб после предыдущей наклеенной буквы. Таким образом, у него на лбу образуется строка.
- Такие перебежки вокруг стола повторяются до тех пор, пока игра не закончится.
- Игра заканчивается в тот момент, когда строка на лбу кого-то одного из игроков оказывается лексикографически меньше, чем все строчки на всех остальных лбах. Этот игрок становится победителем.

В данной игре можно считать, что ребята выполняют все движения синхронно, а буквы на подносах бесконечны, как и свободное место на лбу каждого участника.

Маме эта игра показалась очень забавной, но она сразу заметила два слабых места в плане Серёжи. Во-первых, может случиться так, что игра никогда не закончится. Во-вторых, дети носятся очень быстро, и уследить за словами у них на лбах практически невозможно, поэтому ведущий (то есть сам Серёжа) может не заметить победителя. Для того, чтобы избежать подобных неприятностей, Серёжа попросил Вас, как своего друга-программиста, написать программу, предсказывающую исход игры по начальным данным.

Формат входного файла

В первой строке входного файла содержатся два целых числа N и K . $1 \leq N \leq 3 \cdot 10^5$, $|K| \leq 3 \cdot 10^5$. Следующая строка входного файла содержит строку из N строчных латинских букв. Первая буква соответствует буквам, лежащим на подносе перед первым игроком, вторая — перед вторым и так далее.

Формат выходного файла

Если игра с имеющимися начальными данными когда-нибудь закончится, то выведите в первой строке слово «Finite» (без кавычек), а во второй строке номер победителя (игроки нумеруются начиная с единицы). В случае, если ребята обречены бегать бесконечно, выведите в первой строке слово «Infinite» (без кавычек), а во второй строке выведите номера игроков, которые на момент наступления Апокалипсиса будут иметь лексикографически минимальные строки на своих лбах. Номера необходимо выводить в возрастающем порядке. Игроки нумеруются с единицы.

Примеры

games.in	games.out
3 3 aba	Infinite 1 3
3 2 aba	Finite 1
5 0 break	Finite 4

Задача Н. Вечеринка в Нью-Йорке

Имя входного файла: `grid.in`
Имя выходного файла: `grid.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Скоро у Лизы день рождения! Родители долго думали, чем порадовать свою любимую дочку, и решили, что лучше всего подарить ей путешествие в Нью-Йорк. Ведь это — такой оригинальный подарок! Так как Лиза — уже самостоятельная девочка, она вполне может сама выбрать, где ей поселиться в Нью-Йорке. Однако родители очень беспокоятся за нее, поэтому они разрешили ей выбирать только из H безопасных пятизвездочных отелей.

Лиза — общительная девочка, и у неё очень много друзей. Узнав, что она будет отмечать день рождения в Нью-Йорке, друзья немедленно отправились туда и поселились в C хостелах.

Лиза по случаю дня рождения устраивает маленькую вечеринку. К сожалению, никто из её друзей не хочет пропустить очередную серию нового реалити-шоу «Home-2», но как только серия закончится, все они одновременно выйдут из своих хостелов и поедут поздравлять Лизу. Она — девочка нетерпеливая, поэтому хочет выбрать отель так, чтобы последний друг приехал к ней как можно раньше, т.е. находился как можно ближе. Так как она тоже не хочет отвлекаться от просмотра высокоинтеллектуального шоу, она попросила вас помочь ей с выбором отеля. Конечно, за это она тоже пригласит вас на вечеринку!

Как известно, Нью-Йорк представляет собой идеальную прямоугольную сетку улиц: с севера на юг тянутся N авеню, а с запада на восток — M улиц. Наличием Бродвея мы в этой задаче пренебрежем. Можно считать, что каждый объект находится в узле этой сетки (и, таким образом, однозначно задается номером авеню и номером улицы, возле пересечения которых он расположен), а два соседних узла сетки находятся на расстоянии одного километра.

Формат входного файла

В первой строке входных данных содержится два числа N и M — размеры города ($1 \leq N, M \leq 10^9$). В следующей строке содержится единственное число C — количество хостелов ($1 \leq C \leq 10^5$). Далее в C строках содержатся описания хостелов, каждый из них задается двумя координатами x и y ($1 \leq x \leq N, 1 \leq y \leq M$). В следующей строке содержится одно число H — количество отелей ($1 \leq H \leq 10^5$). В следующих строках содержатся описания отелей, в том же формате, что и хостелы.

Отель и хостел могут располагаться возле одного и того же перекрестка.

Формат выходного файла

В первой строке выходных данных выведите одно число — искомое оптимальное расстояние. В следующей строке выведите номер любого из отелей, гарантирующих данное расстояние.

Примеры

<code>grid.in</code>	<code>grid.out</code>
10 10	6
2	2
1 1	
3 3	
2	
1 10	
4 4	

Задача I. 38 попугаев

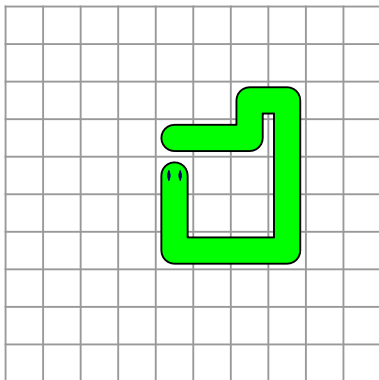
Имя входного файла:	stdin
Имя выходного файла:	stdout
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Как-то раз удав, слонёнок, мартышка и попугай играли в замечательную игру «змейка».

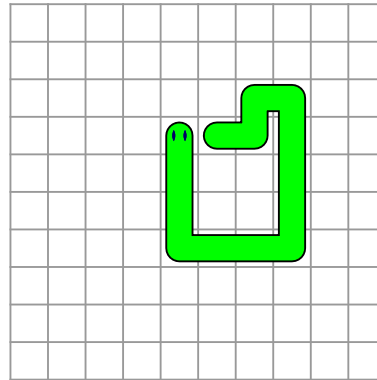
Суть этой известной игры в следующем. Удав перемещается по прямоугольному полю, не переползая через самого себя, и ест яблоки, которые ему сверху скидывает попугай.

Как известно, мерять длину удава можно в слонёнках, мартышках или попугаях. Но в попугаях удав выходит гора-а-а-здо длиннее, поэтому мы остановимся на них в качестве единицы измерения. Поле представляет собой N строк по M ячеек размером 1 попугай \times 1 попугай. Изначально длина удава составляет всего три попугая (каким образом удав становится таким коротким в начале игры — долгая и запутанная история, поэтому мы не будем на ней останавливаться в рамках этой задачи. . .), он занимает три крайних левых ячейки в верхней строке поля и смотрит направо. Каждый раз, съедая яблоко, удав удлиняется на одного попугая и занимает тем самым ещё одну клетку поля.

Более формально — удав длины L попугаев представляет собой последовательность из L различных клеток поля $A_1, A_2 \dots A_L$, таких что любые две соседних в этой последовательности клетки A_i и A_{i+1} являются соседними на поле, т. е. имеют общую сторону. Головой удава считается клетка A_L . За один ход удав может продвинуться на одну клетку — он выбирает новую клетку H для своей головы (естественно, смежную с текущей клеткой головы A_L и не выходящую за границы поля), а его хвост подтягивается следом на одну клетку. То есть, после хода удав будет представлять собой последовательность из клеток A_2, A_3, \dots, A_L, H . **Обратите внимание**, по определению выше удав после выполнения хода не должен дважды содержать одну и ту же клетку, но новое положение головы H вполне может совпадать со старым положением хвоста A_1 (наш удав перемещается подобно гусенице, сначала сжимаясь и подтягивая хвост, а потом расжимаясь и продвигая голову, поэтому это корректный ход).



вот так...



...ходить можно

Заметим, что из этого определения сразу следует, что в каждый момент времени у удава есть возможность пойти в не более чем трёх направлениях — продолжить движение в направлении, куда смотрит его голова, свернуть налево или свернуть направо.

В любой момент времени на поле находится ровно одно яблоко. Каждый раз, когда удав съедает яблоко, его длина увеличивается на одного попугая. Это происходит следующим образом — если удав делает ход, после которого его голова занимает клетку, в которой расположено яблоко, оно считается съеденным. В таком случае яблоко сразу продвигается по телу удава вплоть до бывшего положения хвоста (т. е. до клетки A_1), и на этом месте возникает новая клетка удава. Таким образом, удав приобретает длину в $(L+1)$ попугай и занимает к началу следующего хода клетки $(A_1, A_2, \dots, A_K, H)$. В этот же момент попугай скидывает следующее яблоко на какую-то из незанятых клеток поля.

Удав проигрывает, если он делает некорректный ход. Помогите удаву съесть все яблоки!

Формат взаимодействия с тестирующей системой

Это — интерактивная задача.

В первой строке стандартного потока ввода идут два целых числа N и M ($3 \leq N, M \leq 25$) — количество строк и столбцов поля соответственно.

Далее ваше общение с тестирующей программой будет устроено следующим образом. На каждое яблоко, скидываемое попугаем, вашей программе будут подаваться на стандартный на стандартный поток ввода два целых числа R и C ($1 \leq R \leq N, 1 \leq C \leq M$) — соответственно номер строки и столбца клетки, содержащей яблоко. Строки нумеруются с 1 по N сверху вниз, столбцы нумеруются с 1 по M слева направо.

После этого вы должны вывести последовательность команд для удава в виде строки, состоящей из латинских букв 'R', 'U', 'L' и 'D', которые обозначают что удав должен сделать ход, передвинув свою голову соответственно на клетку правее, выше, левее или ниже. По итогам этой последовательности команд удав должен съесть яблоко, не допуская некорректных ходов. **Обратите внимание**, яблоко должно быть съедено именно последним ходом этой последовательности! После вывода каждой строки сбрасывайте буфер вывода — для этого используйте `flush(output)` на языке Паскаль или Delphi, `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` на языке Python, `System.out.flush()` на языке Java.

В момент окончания игры (как успешного, так и неуспешного — например, если удав делает некорректный ход) вашей программе на ввод подаётся строка, состоящая из двух нулей. **Обратите внимание**, считав эти два нуля, ваша программа обязательно должна **сразу** завершиться! В противном случае в качестве результата проверки вы можете получить вердикт, отличный от настоящего!

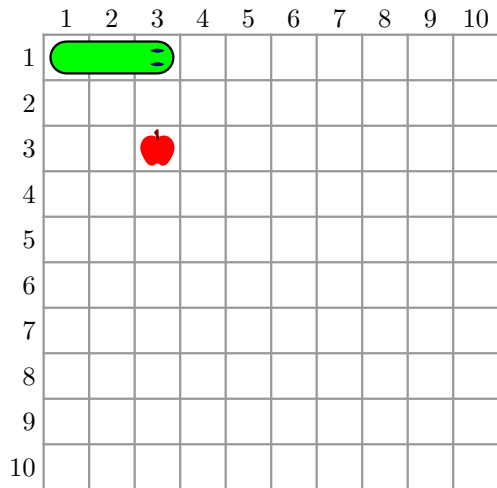
Примеры

stdin	stdout
4 6 3 3	
4 6	DD
1 3	DRRR
0 0	UUULLL
3 5 2 3	
3 4	D
3 5	DR
2 2	R
	ULLL

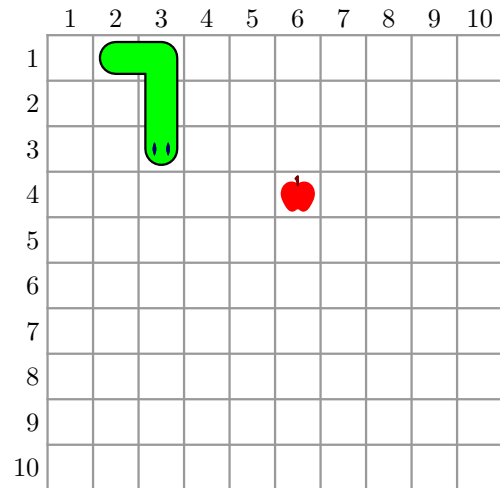
Примечание

Пояснение к первому тесту. Положение удава по ходу игры показано на следующей странице.

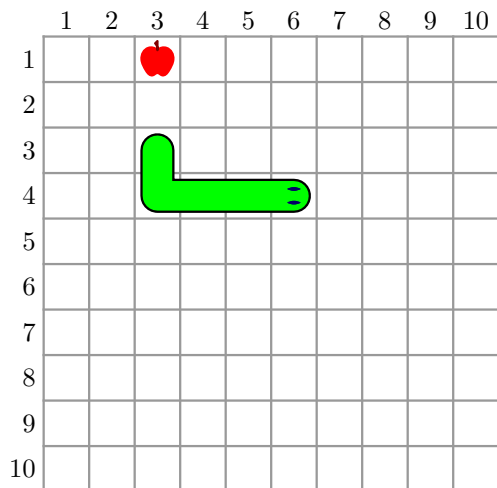
Пояснение ко второму тесту. Обратите внимание, что после шестого хода происходит ситуация, проиллюстрированная в условии: в качестве нового положения головы выбирается клетка (2, 3), являющаяся на тот момент клеткой хвоста.



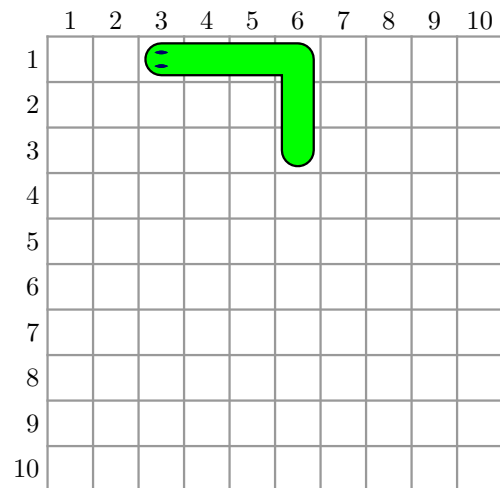
до начала игры



после 2 хода



после 6 хода



после 12 хода

Задача J. Сладкоежка

Имя входного файла: `cake.in`
Имя выходного файла: `cake.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Тимофей — настоящий сладкоежка. Сегодня он купил большой и вкусный круглый торт с клубникой, чтобы полакомиться самому и угостить друзей.



Однако Тимофей угощает друзей очень изощренным способом. А именно, он отрезает кусочки с краев торта прямолинейными разрезами ножа. При этом, чтобы разрезы получились красивыми, нож нужно вести непрерывно, разрезы не должны друг друга пересекать, а закончить процесс отрезания нужно в начальной его точке. Более того, так как Тимофей — большой любитель клубники, то он хочет, чтобы все ягоды оказались внутри центрального куска торта, который, конечно же, достанется ему самому.

Однако оказалось, что разрезать торт таким образом весьма непросто. Помогите Тимофею! Быть может, в награду он и вас угостит кусочком этого замечательного торта.

Будем считать, что торт является кругом на плоскости, а ягоды представляют собой точки. Необходимо построить **любой** выпуклый многоугольник, вершины которого лежат на окружности, ограничивающей круг, а все заданные точки находятся внутри него или на его границе.

Формат входного файла

В первой строке входных данных содержатся четыре числа N, R, X, Y : количество ягод, радиус торта и координаты центра торта ($1 \leq N \leq 100, 1 \leq R \leq 1000, 0 \leq |X|, |Y| \leq 1000$). В следующих N строках содержатся описания ягод. Каждая ягода задается двумя числами x_i, y_i — своими координатами ($0 \leq |x_i|, |y_i| \leq 1000$).

Формат выходного файла

В первой строке выходных данных выведите единственное число M — количество вершин в построенном вами многоугольнике ($3 \leq M \leq 222$). В следующих M строках выведите координаты вершин многоугольника, в порядке обхода против часовой стрелки.

Ваш ответ будет считаться верным, если все выведенные вами точки различны, действительно лежат на окружности, если они образуют выпуклый многоугольник, и если все ягоды лежат внутри или на границе многоугольника.

Проверяющая программа производит все проверки с точностью 10^{-6} , в частности, независимо от того, сколько знаков вы выведете после десятичной точки, две вершины многоугольника будут считаться совпадающими, если они отличаются и по x -координате, и по y координате не более, чем на 10^{-6} .

Примеры

cake.in	cake.out
4 6 2 3	4
2 4	-4.000000 3.000000
3 4	-2.242640 -1.242640
1 2	8.000000 3.000000
-4 3	2.000000 9.000000