

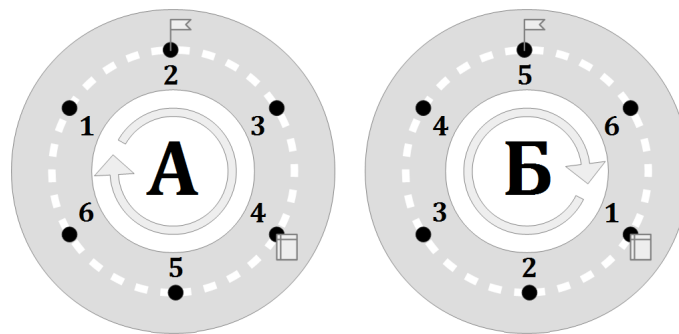
Задача А. Подсчёт столбов

Имя входного файла:	stdin
Имя выходного файла:	stdout
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 ввода содержит единственное целое число D , которое равно 1, если мальчики ехали в одном направлении, и -1 , если в разных. Следующие 4 строки содержат 4 натуральных числа A_p , B_p , A_f , B_f , по одному числу в строке, каждое из которых не превосходит 10^6 : 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.

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

stdin	stdout
1 4 1 2 5	6
-1 4 9 4 7	-1

Задача В. В гору пойдёт!

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
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 имеется запись в журнале, содержащая информацию о том, сколько дней они шли в гору, а сколько — спускались.

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

Формат входного файла

Входные данные содержат две строки. В первой строке записано целое неотрицательное число A — это количество дней, в которые альпинисты поднимались в гору. Вторая строка содержит целое неотрицательное число B — количество дней, в которые альпинисты спускались ($A + B + 1 = N$, $1 \leq N \leq 100\,000$).

Формат выходного файла

Выведите 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 могли посетить участники экспедиции.

Примеры

<code>stdin</code>	<code>stdout</code>
0 1	2 1
3 0	1 2 3 4

Задача С. Проще простого

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
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.

Леопольд очень интересуется всем, что связано с простыми числами. Недавно он узнал про *Постулат Бертрана* — оказывается, на отрезке между N и $2N$ всегда найдётся хотя бы одно простое число! Несмотря на простую формулировку и интуитивную очевидность этого утверждения, сформулированного французским математиком Жозефом Луи Франсуа Бертраном, оно было доказано только в середине 19 века русским математиком Пафнутием Львовичем Чебышёвым.

Впечатлённый тем, как можно увековечить своё имя на страницах истории математики, Леопольд 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 отрезок из L последовательных натуральных чисел, содержащий ровно K простых чисел. Чтобы результаты было легче анализировать, он просит вас ограничиться в поисках первыми **тридцатью тысячами** натуральных чисел. Помогите ему, и, возможно, и вам удастся оставить след в истории!

Формат входного файла

На вход программе подаются целые числа L и K ($1 \leq L \leq 30\,000$, $0 \leq K \leq L$), каждое в отдельной строке.

Формат выходного файла

Если в пределах до 30 000 найдётся отрезок из L подряд идущих натуральных чисел, среди которых ровно K простых, выведите минимальное и максимальное число на этом отрезке. В противном случае выведите единственное число -1 . Если существует несколько отрезков, удовлетворяющих условию, выведите любой.

Примеры

<code>stdin</code>	<code>stdout</code>
20 5	8 27
100 66	-1

Задача D. Матрёшки

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
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, если размер одной меньше чем размер другой. 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 $r_1 < r_2 < r_3 < \dots < r_M$. 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 входных данных содержится число N — количество заготовок на складе ($1 \leq N \leq 1000$). В последующих N строках содержатся целые числа $s_1, s_2, s_3, \dots, s_N$, где s_i — это размер i -й заготовки ($1 \leq s_i \leq 1000$).

Формат выходного файла

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

Далее выведите T строк, описывающие матрёшек, которые завод закажет у союза художников. Матрёшка описывается возрастающей последовательностью чисел $r_1, r_2, r_3, \dots, r_M$, где r_i — размер i -й заготовки, входящей в матрёшку.

Примеры

stdin	stdout
5	2
1	1 2 3
3	2 3
2	
2	
3	
6	3
4	1 2
2	2 4
1	2 4
2	
4	
2	

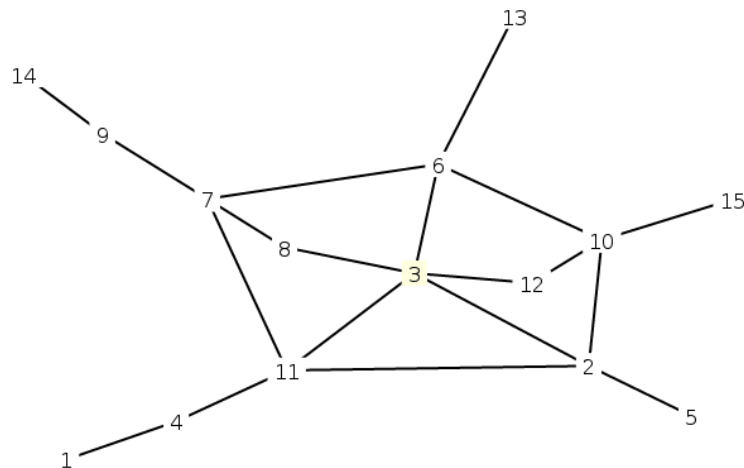
Задача Е. Карта метро

Имя входного файла:	stdin
Имя выходного файла:	stdout
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 Аlрer подозревает администрацию уездного города в нарушении их патента на jMetro и грозитя возбудить против города Н 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;
- количество радиальных линий не меньше **пяти**;
- 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 Аlрer 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 ($1 \leq N, M \leq 300$) — 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.

Формат выходного файла

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

Примеры

stdin	stdout
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
4 3 2 1 2 3 2 4	NO

Примечание

Первый пример соответствует рисунку из условия.

Задача F. Свободная ячейка

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
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 XP™», 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 дополнительно N свободных слотов.

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

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

Формат входного файла

В первой строке находится натуральное число N ($1 \leq N \leq 6$) — количество свободных слотов для карт (не включая слот с исходной колодой).

Во второй строке находится натуральное число K — количество карт в исходной колоде. ($2 \leq K \leq 20$)

Формат выходного файла

Выведите “YES”, если пасьянс сходится, либо “NO”, если нет.

Примеры

stdin	stdout
3 5	YES
2 5	NO

Примечание

Пояснение к примерам. В первом примере 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. При двух свободных слотах (пример 2) 5 карт переложить уже нельзя.

Задача G. Развивающие игры

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

У Даши тоже 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 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 по часовой стрелке (от 1 к N , после N опять следует 1). Так, например, если за столом сидят 5 человек, и ведущий называет число 2, то рассадка меняется с $(1, 2, 3, 4, 5)$ на $(4, 5, 1, 2, 3)$.
- 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 из игроков (обязательно только у одного) оказывается в алфавитном порядке 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 ($1 \leq N \leq 10^3$, $0 \leq K \leq N$). Следующая строка входного файла содержит строку из N строчных латинских букв: первая буква соответствует буквам, лежащим на подносе перед первым игроком, вторая — перед вторым игроком 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. В случае, если ребята обречены бегать бесконечно долго, выведите в первой строке слово «Infinite» (без кавычек), а во второй строке выведите в возрастающем порядке номера игроков, которые на момент наступления Апокалипсиса будут иметь минимальные в алфавитном порядке строки на своих лбах.

Примеры

stdin	stdout
6 2 cabcba	Finite 6
4 2 abaa	Infinite 1 3
5 0 ababa	Infinite 1 3 5

Задача Н. Вечеринка в Нью-Йорке

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 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 ($1 \leq N, M \leq 1000$). В следующей строке содержится единственное число C — количество хостелов ($1 \leq C \leq 1000$). 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 безопасных отелей ($1 \leq H \leq 1000$). 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

stdin	stdout
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».

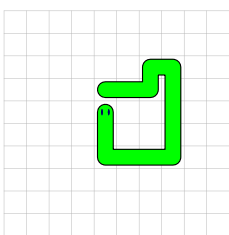
Суть этой известной игры в следующем. Удав перемещается по прямоугольному полю, не переползая через самого себя, и ест яблоки, которые ему сверху скидывает попугай.

Как известно, измерять длину удава можно в слонятах, 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 прямоугольную сетку из 10 строк по 10 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, таким образом, ещё одну клетку поля.

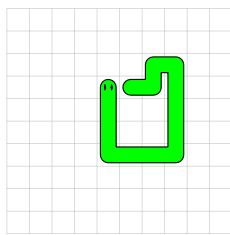
Более формально, удав длины L попугаев представляет собой последовательность из L различных клеток поля A_1, A_2, \dots, A_L таких, что любые две соседние в этой последовательности клетки A_i и A_{i+1} являются соседними на поле, т. е. имеют общую сторону. Головой удава считается клетка A_L . За один 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 разжимаясь и продвигая голову, поэтому это корректный ход).

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.

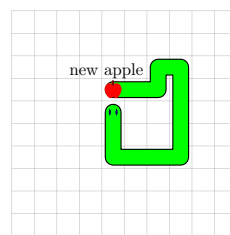
Перед каждым ходом удава на поле находится ровно одно яблоко. Каждый раз, когда удав съедает яблоко, его длина увеличивается на одного попугая. Это происходит следующим образом: удав делает ход, и если после этого в какой-нибудь из клеток удава (т. е. из клеток A_2, A_3, \dots, A_L, H в обозначениях выше) оказывается яблоко, оно считается съеденным (как именно удав может есть яблоки любой частью своего тела — не менее запутанная история, чем объяснение таким скромным размерам удава в начале игры, поэтому её мы также опустим). Если яблоко было съедено, то оно 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 поле. **Обратите внимание**, что если удав ест яблоко, то в этом случае он уже не имеет права выбирать в качестве нового положения головы H старую клетку своего хвоста, потому что тогда к началу следующего хода она будет дважды встречаться в теле удава, что недопустимо!



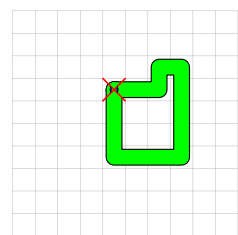
вот так...



...ходить можно



а вот так...



...нельзя

Удав проигрывает, если он делает некорректный ход. Некто (не будем говорить, кто, хотя это был слонёнок) рассказал вам, в какие клетки поля и в каком порядке попугай собирает яблоки. Помогите удаву построить выигрышную стратегию перемещения!

Формат входного файла

В первой строке ввода записано натуральное число K — количество яблок, которое скинет попугай на поле. Длина удава не превзойдёт 38 попугаев, т. е. $K + 3 \leq 38$.

В каждой из последующих K строк задано по два целых числа R_i, C_i ($1 \leq R_i, C_i \leq 10$) — номера строки и столбца, куда попугай скинет i -е яблоко. Строки нумеруются с 1 по 10 сверху вниз, столбцы нумеруются с 1 по 10 слева направо. Позиции, где возникают яблоки, могут повторяться.

Формат выходного файла

Сначала выведите целое число T — количество ходов, которое потребуется удаву, чтобы съесть все яблоки.

Во второй строке выведите строку, задающую последовательность ходов удава, состоящую из T латинских букв 'R', 'U', 'L', 'D', обозначающих, что на очередном ходу клетка головы удава смещается соответственно вверх, вправо, влево или вниз.

Будет зачтена любая последовательность ходов, которая не приводит к проигрышу удава, по итогам которой все яблоки будут съедены, и длина T которой не превосходит 10 000 в целях экономии времени играющих. Жюри гарантирует, что при данных ограничениях задачу всегда возможно решить.

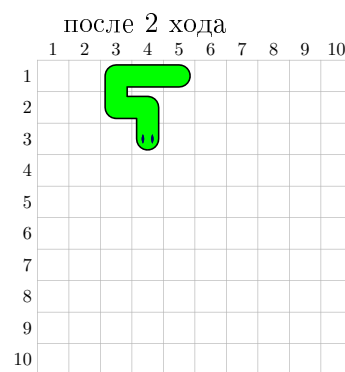
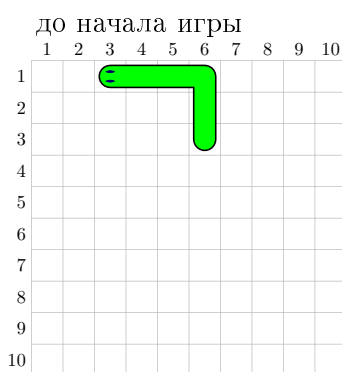
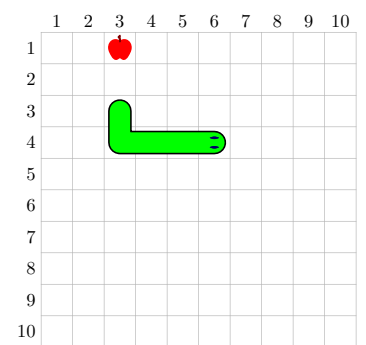
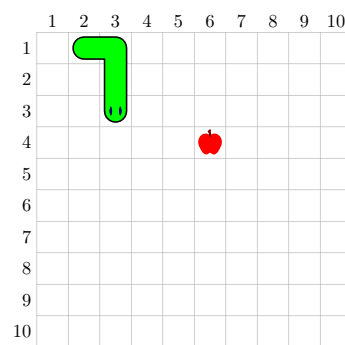
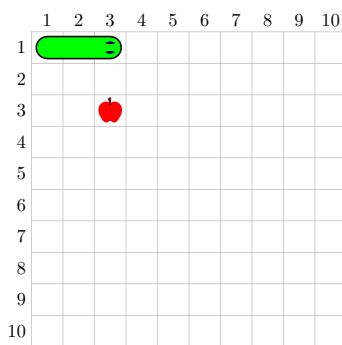
Примеры

stdin	stdout
3 3 3 4 6 1 3	16 DDRRRRUUULLLLDRD
4 2 3 1 2 3 5 2 2	8 DDRRULLL

Примечание

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

Пояснение ко второму тесту. Второе яблоко выпадает прямо на удава, и сразу оказывается съеденным, т. е. перед вторым ходом удав состоит из клеток $(1, 1)$, $(1, 2)$, $(1, 3)$, $(2, 3)$, а после второго хода — из клеток $(1, 1)$, $(1, 2)$, $(1, 3)$, $(2, 3)$, $(3, 3)$. Также 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.



после 12 хода

в конце игры