

Задача А. Apple Pen

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

У вас возникла идея нового клипа, который должен взорвать интернет. В клипе человек будет втыкать ручки в яблоки и утверждать, что у него получаются ананасы. Всё это будет сопровождаться весёлой музыкой и танцами. Одна ручка в магазине стоит X рублей, а одно яблоко стоит Y рублей. Для получения одного ананаса в клипе используется ровно одна ручка и ровно одно яблоко. Скажите, какое максимальное количество ананасов можно будет собрать из ручек и яблок, если бюджет на покупку составляет N рублей.

Формат входных данных

В первой строке содержится целое число N — бюджет на покупку ручек и яблок ($1 \leq N \leq 10^9$).

Во второй строке содержится целое число X — стоимость одной ручки в магазине ($1 \leq X \leq 10^9$).

В третьей строке содержится целое число Y — стоимость одного яблока в магазине ($1 \leq Y \leq 10^9$).

Формат выходных данных

В единственной строке выведите одно число — максимальное количество ананасов, которое можно собрать из ручек и яблок при заданном бюджете.

Примеры

стандартный ввод	стандартный вывод
3 2 1	1
5 1 10	0

Задача В. Камень-ножницы-бумага-ящерица-Спок

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вероятно, все участники этой олимпиады знают про игру «камень-ножницы-бумага». Возможно, некоторые даже знают про игру «Камень-ножницы-бумага-ящерица-Спок». Но на всякий случай мы объясним правила. В игре участвуют ровно 2 человека. Каждый из них может в качестве своего хода выбрать один из пяти вариантов:

- Камень (Rock)
- Ножницы (Scissors)
- Бумага (Paper)
- Ящерица (Lizard)
- Спок (Spock)

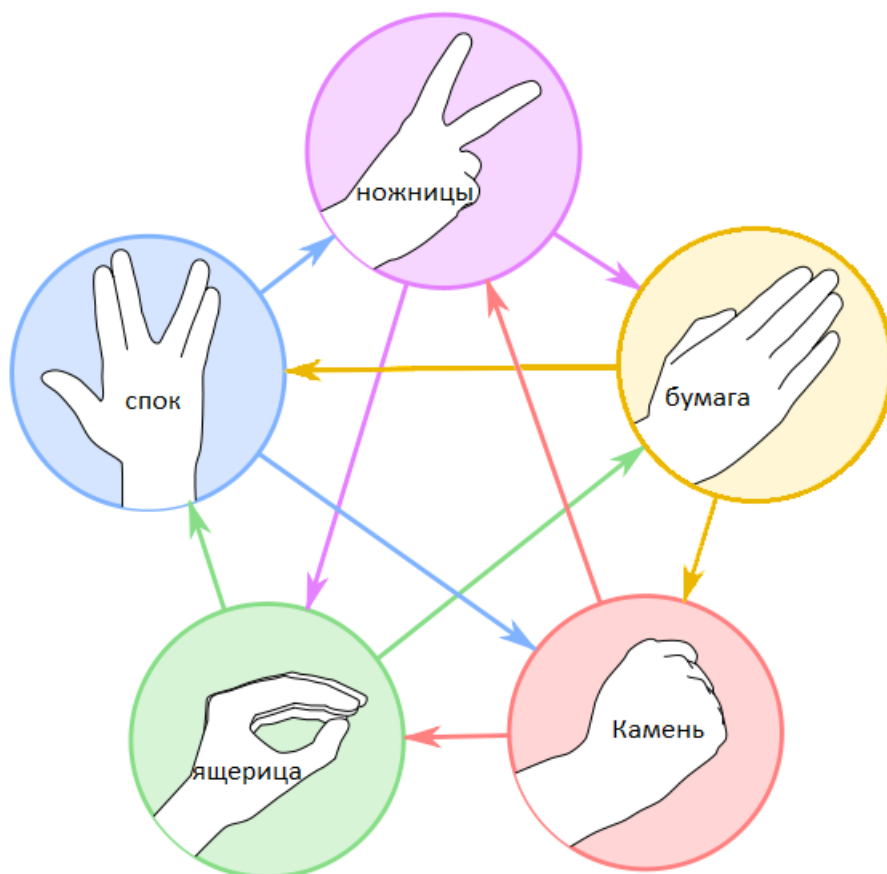
В скобочках к каждому варианту указано его английское название. Правила игры следующие:

- Ножницы режут бумагу
- Бумага заворачивает камень
- Камень давит ящерицу
- Ящерица травит Спока
- Спок ломает ножницы
- Ножницы отрезают голову ящерице
- Ящерица ест бумагу
- Бумага содержит улики против Спока
- Спок испаряет камень
- Камень затупляет ножницы

Если ходы двух людей совпадают, то объявляется ничья.

Для ясности на схеме ниже стрелочками указано, кто кого побеждает. Стрелочка идёт от победителя к проигравшему.

Даны ходы двух людей. Ваша задача — определить исход поединка.



Формат входных данных

В первой строке вводится ход первого игрока. Гарантируется, что это одно из пяти английских слов:

- Rock
- Scissors
- Paper
- Lizard
- Spock

Значения этих слов указаны в условии выше. Во второй строке в аналогичном формате вводится ход второго игрока.

Формат выходных данных

Если первый игрок победит второго, то выведите «First» (без кавычек).

Если второй игрок победит первого, то выведите «Second» (без кавычек).

Если же будет ничья, то выведите «Tie» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
Rock Paper	Second
Rock Rock	Tie
Lizard Spock	First

[illegible]

Пример

стандартный ввод	стандартный вывод
5 13)\...../(.)_..._V_/_...(..)_.....(.....'_',..... \...../ . . !_..._M_/_...! ..\...../'_',.....	7

Замечание

Поскольку настоящий тест занимает слишком много места, в примере вводятся два маленьких изображения в том же формате, а так же правильный ответ для этих двух изображений.

Обратите внимание, что ваша программа не должна выдавать правильный ответ на тест, указанный в примере. В задаче ровно один тест и он описан в условии.

Задача D. Напёрстки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Напёрсточники — те ещё жулики. Никогда не играйте с ними!

Сегодня, однако, вам придётся. Дело в том, что вы поспорили с другом, что легко сможете раскусить мошенничество, если напёрсточник попытается вас обмануть. Так что теперь придётся отдуваться.

Правила игры в напёрстки следующие:

1. У напёрсточника ровно три напёрстка.
2. В начале игры он выбирает один из них и складывает под него приз.
3. После этого он производит n манипуляций.
Каждая манипуляция заключается в том, что два напёрстка меняются местами.
4. После произведённых манипуляций вам нужно правильно угадать, под каким по счёту напёрстком находится приз.

В реальной жизни сложность этой игры заключается в том, что бывает трудно уследить за манипуляциями. Именно поэтому вы решили написать программу, которая бы делала это за вас.

Формат входных данных

В первой строке через пробел вводятся целые числа n, k — количество манипуляций и позиция напёрстка по счёту, под которым изначально лежит приз ($1 \leq n \leq 10^5, 1 \leq k \leq 3$).

В i -й из следующих n строк содержатся два целых числа a_i, b_i — позиции по счёту напёрстков, которые напёрсточник меняет местами во время i -й манипуляции ($1 \leq a_i, b_i \leq 3, a_i \neq b_i$).

Формат выходных данных

Выведите единственное число — позицию по счёту напёрстка, под которым лежит приз после n манипуляций.

Пример

стандартный ввод	стандартный вывод
2 1 1 2 2 3	3

Замечание

В примере изначально приз находится под первым по счёту напёрстком. После первой манипуляции первый и второй напёрсток меняются местами и теперь приз находится под вторым по счёту напёрстком. После второй манипуляции второй и третий напёрсток меняются местами и теперь приз находится под третьим напёрстком. Поэтому правильный ответ — 3.

Задача Е. Прокрастинация

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В психологии *прокрастинацией* называется склонность откладывать дела на потом. Валя — мастер прокрастинации. Каждый день мама с папой дают Вале очередное задание. Иногда они просят помыть полы в квартире, иногда — прибраться в комнате, иногда — сделать уроки... Валя же не хочет выполнять каждое задание по отдельности. Вместо этого, он хочет выполнять сразу много заданий, принадлежащих одной категории. Зачем мыть посуду каждый раз, когда можно дождаться, пока соберётся много посуды, а потом помыть всю разом?

Более формально, имеются n дней и k типов заданий, которые Вале дают родители. Типы заданий пронумерованы целыми числами от 1 до k . В i -й день Валя получает задание типа a_i . Валя может посвятить целиком один день выполнению всех накопившихся к этому моменту заданий одного типа (выданных в этот день или ранее). Либо можно ничего не делать целый день. Выполнять задания разных типов в течение одного дня нельзя.

Ваша задача — составить Вале план выполнения заданий таким образом, чтобы максимальное количество дней можно было ничего не делать, и при этом к концу n дней все выданные родителями задания были выполнены.

Формат входных данных

В первой строке через пробел вводятся целые числа n и k — количество прошедших дней и количество различных типов заданий ($1 \leq n, k \leq 10^5$).

Во второй строке дано n чисел, разделённых пробелами, где i -е число равно a_i — типу задания, выданному Вале в i -й день ($1 \leq a_i \leq k$). Возможна ситуация, когда задания некоторых типов не выдаются Вале совсем.

Формат выходных данных

В единственной строке выведите n чисел, разделённых пробелами, где i -е число должно быть равно 0, если Валя должен бездельничать в i -й день, либо типу задания, если Вале стоит выполнить все задания этого типа в этот день. Если существует несколько возможных вариантов ответа, выведите любой.

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 3	1 2 3
4 2 1 1 1 1	0 0 0 1

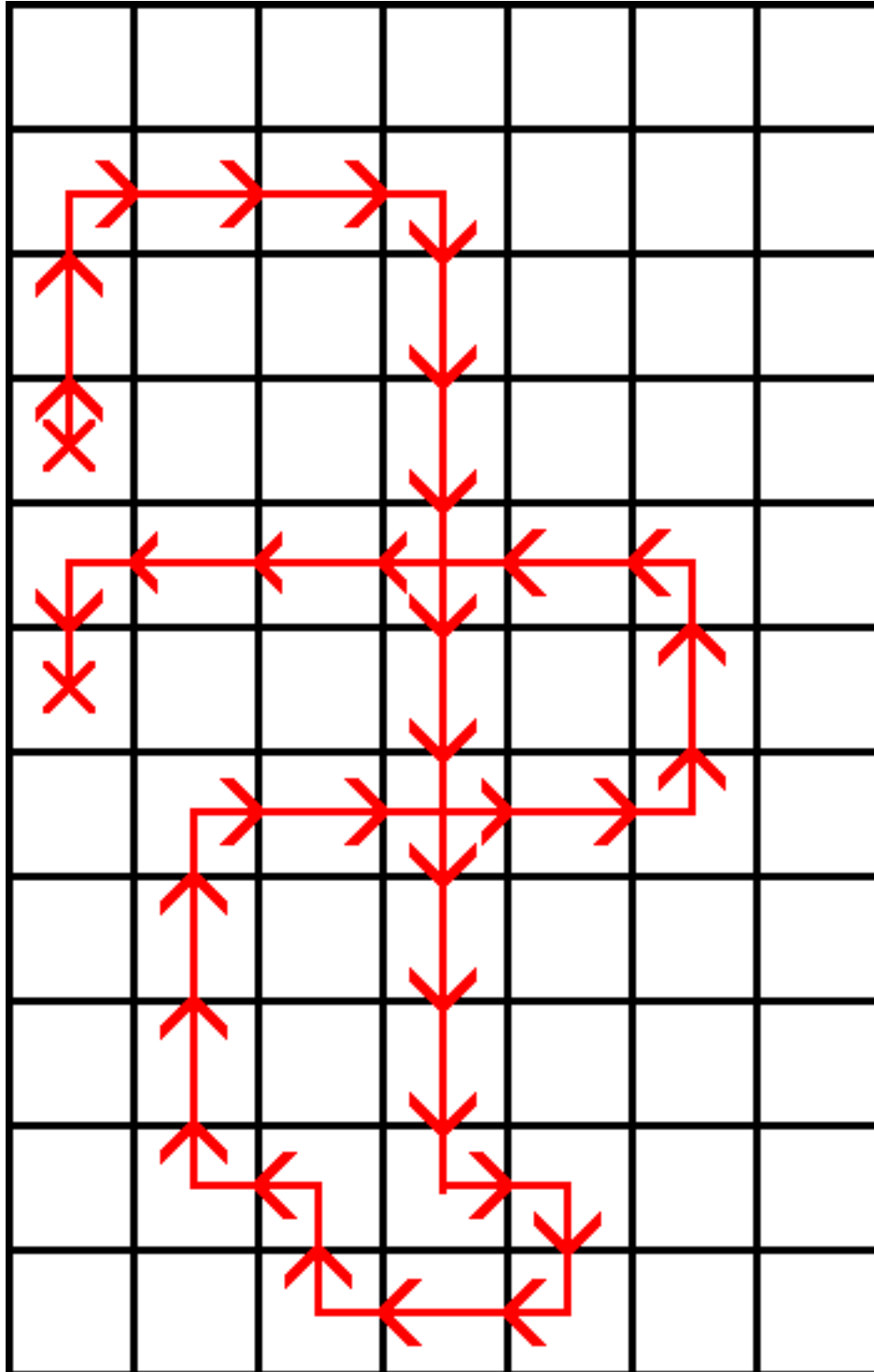
Замечание

В первом примере Вале каждый день дают задание нового типа, поэтому он вынужден каждый день выполнять новое задание родителей.

Во втором примере Вале четыре дня подряд дают задание типа 1, поэтому он может выполнить их все в последний день. В таком случае он сможет бездельничать целых три дня.

Задача F. Кубик

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт



По клетчатому полю катается кубик. Точнее, ещё только планирует кататься. План его будущих передвижений уже составлен. При перемещении в соседнюю клетку кубик перекачивается через соответствующее ребро.

Прежде чем кубик начнёт свои перемещения по полю, вам разрешено на каждой грани кубика чернилами написать одно натуральное число от 1 до 6. При этом нужно соблюдать условие, что каждое число должно быть написано ровно на одной из граней. Каждый раз, когда кубик будет вставать некоторой гранью на клетку, на этой клетке будет отпечатываться след от кубика в виде числа, которое было напечатано на этой грани кубика. Причём, если до этого на клетке кубик уже оставлял след, то старый след перекроется новым. На клетке, в которой кубик начинает, отпечатывается число на его нижней грани.

Вам дана траектория движения кубика. Требуется определить, какое максимальное значение суммы отпечатанных чисел на клетках поля можно получить, правильным образом написав числа на гранях кубика.

Формат входных данных

В данной задаче ровно один тест, описывающий траекторию движения кубика, изображённую на рисунке в условии задачи. Крестиками указаны начальная и конечная точка маршрута. Стрелочками — переходы между клетками в ходе маршрута. На ввод подаётся одна строка, каждый символ которой является одним из четырёх:

- «L» (соответствует движению влево)
- «R» (соответствует движению вправо)
- «U» (соответствует движению вверх)
- «D» (соответствует движению вниз)

Каждый символ строки описывает одно перемещение кубика в соответствующем символу направлении. Перемещения в строке указаны в том порядке, в котором кубик их совершает. Так, например, строке «LLUR» соответствуют перемещения кубика «влево-влево-вверх-вправо».

Программа, которую вы напишете, может считать тест из входных данных и обработать его, или сразу выдать правильный ответ, не читая входные данные.

Формат выходных данных

Требуется вывести одно число — максимальное значение суммы отпечатанных чисел на клетках поля.

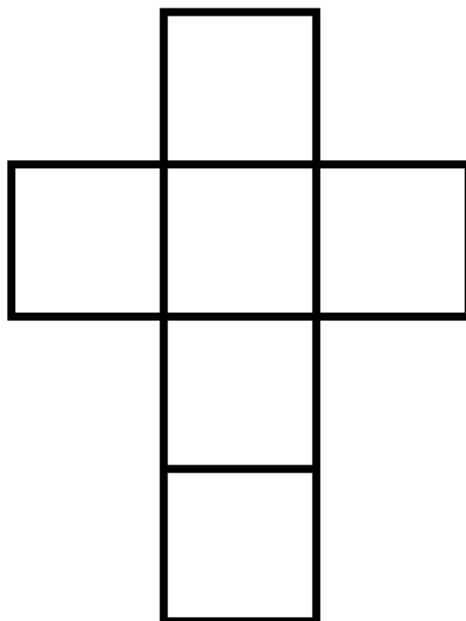
Пример

стандартный ввод	стандартный вывод
UURRRDDDDDDDRDLLULUURRRRUULLLLLD	0

Замечание

Обратите внимание, что ответ в примере является неверным и приведён только для демонстрации формата вывода.

На следующей странице мы напечатали для вас заготовку, из которой можно быстро изготовить кубик. Для этого нужно аккуратно оторвать куски листа, оставив только нарисованный крест, а далее согнуть его во по всем чёрным линиям, после чего его можно будет свернуть в кубик.



Задача G. Уральские блинчики

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

После квалификационного тура довольные, но голодные программисты зашли в ресторан «Уральские блинчики» и заказали себе несколько фирменных блинчиков. Для того чтобы приготовить блинчик, повар должен прожарить каждую из его сторон на сковороде.

Фирменный блинчик имеет форму квадрата со стороной 1. Сковорода повара имеет форму прямоугольника ширины m и высоты n . Она разбита на $n \cdot m$ клеток. К сожалению, некоторые из клеток подгорели, и в них нельзя класть блинчики. При этом не подгоревшие клетки образуют связную область, то есть из любой не подгоревшей клетки можно попасть в любую другую, перемещаясь каждый раз в соседнюю неподгоревшую клетку. Соседними называются такие две клетки, что либо они находятся в одной и той же строке и в соседних столбцах, либо в одном и том же столбце и в соседних строках.

В некоторые из не подгоревших клеток повар положил блинчики (не более одного на клетку), и они уже поджарились с одной стороны. Теперь их нужно перевернуть на другую сторону. Для этого можно поддеть любой блинчик лопаткой снизу и перевернуть его, переместив при этом на соседнюю клетку. Перемещать блинчики за пределы сковороды нельзя. Естественно, если перевернуть блинчик дважды, он снова будет расположен прожаренной стороной вниз. Если переместить блинчик на клетку, где уже лежит другой блинчик, они образуют стопку. Одновременно на одной из клеток могут находиться сколько угодно блинчиков, но переворачивать можно только верхний из них.

Сумеет ли повар добиться того, чтобы каждый блинчик лежал прожаренной стороной вверх, и на каждой клетке лежало не более одного блинчика? При этом нельзя перемещать блинчики на подгоревшие клетки, даже временно.

Формат входных данных

В первой строке через пробел даны целые числа n и m — размер сковороды ($1 \leq n, m \leq 100$).

Далее даны n строк длины m каждая, описывающих сковороду. Строки состоят из символов «.» (код 46), «#» (код 35) и «P» (код 80). Символ «.» обозначает, что клетка свободна; символ «#» обозначает, что клетка подгоревшая; символ «P» обозначает, что клетка занята блинчиком.

Гарантируется, что на сковороде есть хотя бы один блинчик, и что не подгоревшие клетки образуют связную область.

Формат выходных данных

Если повар сможет перевернуть все блинчики на неподжаренную сторону, следуя условиям задачи, в единственной строке выведите «YES» (без кавычек). Иначе, выведите «NO» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
1 3 P.P	NO
2 2 PP PP	YES
2 2 PP P#	NO

Задача Н. Светофоры

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Валя учится в школе в четвёртом классе. Все эти годы он составлял оптимальный маршрут от дома до здания школы. Эту задачу осложняет то, что время ходьбы по одному и тому же маршруту не всегда одинаковое: оно зависит от того, как долго придётся стоять на светофорах. Чтобы не стоять слишком долго, выгодно переходить на перекрёстках, где зелёный свет горит то в одном, то в другом направлении. Стоя на таком перекрёстке Валя и придумал эту задачу.

Представьте, что вы стоите на перекрёстке, и в обе стороны горит красный свет. На всех светофорах стоит счётчик, показывающий, сколько секунд осталось ждать. В одну сторону счётчик показывает A секунд, в другую — B секунд. Каждую секунду числа на обоих счётчиках одновременно уменьшаются на 1. Как только один из счётчиков достигнет нуля, загорится зелёный свет.

Пока ещё горит красный, интересны все такие моменты, когда числа на счётчиках будут различаться в целое число раз, то есть когда их частное — целое число. Посчитайте, сколько раз такое случится.

Формат входных данных

В единственной строке через пробел вводятся целые числа A и B — числа на первом и втором счётчиках соответственно ($1 \leq A, B \leq 10^9$).

Формат выходных данных

В единственной строке выведите одно целое число — количество раз, когда числа на счётчиках будут различаться в целое число раз до того, как один из них достигнет нуля.

Примеры

стандартный ввод	стандартный вывод
3 30	2
16 4	4

Задача I. Персеантовка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Известен факт, что если изменить порядок букв внутри слова, то слово всё равно сможет иметь смысл текста, если правая и левая буква окажутся на месте.

Вам дан **текст**. **Текст** — это строка, состоящая из строчных латинских букв и пробелов, завершающаяся точкой. **Словом** называется непустая строка, состоящая из строчных латинских букв, такая, что перед первой буквой **слова** находится начало **текста** или пробел, а после последней буквы **слова** находится точка или пробел. Гарантируется, что между каждой парой соседних **слов** стоит ровно один пробел. Гарантируется, что точка в **тексте** содержится ровно одна. Она следует строго за последним **словом**. Пробела между последним **словом** и точкой быть не может. Гарантируется, что в тексте есть по крайней мере одно **слово**.

В тексте авторы переставили некоторые буквы внутри **слов** местами, при этом не изменяя первую и последнюю букву **слов**. У вас имеется **словарь** всех возможных **слов**, которые могли встречаться в тексте до того, как буквы были переставлены. Никакие другие **слова** не могли встречаться в исходном тексте. Ваша задача — написать программу, которая сможет восстановить исходный текст (то, каким он был до перестановки букв).

Формат входных данных

В первой строке вводится **текст**, полученный после некоторой перестановки символов некоторых **слов**. Гарантируется, что суммарная длина текста не превышает $5 \cdot 10^5$ символов.

В следующей строке вводится целое число n — количество **слов** в **словаре** ($1 \leq n \leq 5 \cdot 10^5$).

Каждая из следующих n строк содержит одно **слово** из **словаря**. Гарантируется, что суммарная длина всех **слов** в **словаре** не превышает $5 \cdot 10^5$ символов. Также гарантируется, что все **слова** в **словаре** различны.

Формат выходных данных

Выведите единственную строку — исходный текст до перестановки символов. Если существует несколько возможных вариантов ответа, выведите любой из них. Если не существует ни одного корректного варианта ответа, выведите «No solution» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
hello wolrd. 2 hello world	hello world.
hello wlrd. 2 hello world	No solution
tihs is vrey sceret txet. 7 text secret serect scret is very this	this is very secret text.

Задача J. Катамари

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

«Катамари» — это магический шар, к которому прилипает всё, что его касается. Он катится по Земле, собирая всё большие и большие объекты, пока не вырастет достаточно громадным, чтобы стать новым небесным телом.

В этот раз катамари занесло на прямоугольное клетчатое поле размера n строк на m столбцов. Будем говорить, что клетка, находящаяся в строке i в столбце j , имеет координаты (i, j) ($1 \leq i \leq n$, $1 \leq j \leq m$). В каждой клетке расположен какой-нибудь объект. Размер объекта, находящегося в клетке с координатами (i, j) , равен a_{ij} . При этом мир настолько разнообразен, что для любого числа существует **не более трёх** объектов такого размера.

Ваша задача — прицепить все объекты, перекатывая катамари по полю. Начать можно в любой клетке. Две клетки называются соседними, если номера их столбцов совпадают, а номера строк отличаются на единицу, либо если номера их строк совпадают, а номера столбцов отличаются на единицу. Перемещаться из клетки можно только в соседнюю клетку, выходить за пределы поля нельзя. При этом одну клетку нельзя посещать дважды.

Необходимо также выполнить следующее ограничение: размеры объектов, прицепленные к катамари, должны нестрого возрастать. Иными словами, для любых двух объектов разного веса, меньший из них должен быть приклеен раньше, чем больший.

Составьте такой маршрут для катамари, чтобы каждый объект был собран, и все ограничения были выполнены.

Формат входных данных

В первой строке через пробел вводятся два целых числа n и m — размеры поля ($1 \leq n, m \leq 100$).

Следующие n строк описывают поле. В каждой из них содержится m целых чисел a_{ij} , разделённых пробелом ($1 \leq a_{ij} \leq 10^4$) — размеры объектов в соответствующих клетках.

Гарантируется, что для любого числа существует **не более трёх** объектов такого размера.

Формат выходных данных

Если составить требуемый маршрут невозможно, в единственной строке выведите -1 .

Иначе, выведите $n \cdot m$ строк. В k -й строке через пробел выведите два числа i_k и j_k — координаты k -й клетки маршрута. Маршрут должен обходить всё поле, не посещая одну клетку дважды, а размеры объектов в порядке обхода должны составлять неубывающую последовательность. Если возможных ответов несколько, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
2 4 1 4 8 16 2 4 16 16	1 1 2 1 2 2 1 2 1 3 1 4 2 4 2 3
3 3 1 2 2 1 2 3 1 3 3	-1

Задача К. Малыш и Карлсон

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На соревнованиях по программированию не раз встречались задачи про Малыша и Карлсона, делящих между собой сладости. Чтобы не повторяться, каждый автор задачи хочет добавить в своё условие изюминку, которая бы выделила задачу на фоне остальных. То Малыш и Карлсон делят многослойный торт, то они играют в игру на плитке шоколада. Дошла очередь и до самих героев: какую задачу можно придумать, если бы Малыш и Карлсон жили в параллельной вселенной?

Малыш и Карлсон из параллельной вселенной — образованные и воспитанные люди. Они не страдают от лишнего веса, и каждый из них готов отдать другу половину всех сладостей. Сегодня они выиграли в командном соревновании по программированию большой торт (в параллельной вселенной команды состоят из двух человек). Торт имеет форму выпуклого N -угольника. Малыш и Карлсон тут же решили поделить его на две равные по площади части, причём сделать это надо с математической точностью.

Для этого они ввели декартову систему координат и измерили координаты вершин торта. Все координаты оказались целыми числами. Осталось провести прямолинейный разрез. Чтобы не допустить даже малейшей погрешности, было принято решение выбрать две точки A и B с целочисленными координатами и сделать разрез по прямой, проходящий через эти две точки.

После долгого соревнования Малыш и Карлсон устали, поэтому они обратились к своим братьям и сёстрам из параллельной вселенной за помощью. Найдите две различные точки A и B , чтобы прямая, проходящая через них, разделила торт на две равные по площади части. Координаты точек должны быть целыми числами, не превышающими 10^{18} по модулю.

Формат входных данных

В первой строке находится целое число N — количество вершин многоугольника ($3 \leq N \leq 10^3$).

В следующих N строках через пробел записаны целые числа x_i и y_i — координаты i -й вершины многоугольника ($-10^5 \leq x_i, y_i \leq 10^5$).

Гарантируется, что вершины задают выпуклый многоугольник в порядке обхода против часовой стрелки, никакие две точки не совпадают и никакие три точки не лежат на одной прямой.

Формат выходных данных

Если решения не существует, в единственной строке выведите -1 .

Иначе, выведите две строки. В первой строке через пробел выведите целые числа x_a и y_a — координаты точки A ($-10^{18} \leq x_a, y_a \leq 10^{18}$). Во второй строке в аналогичном формате выведите координаты точки B . Точки не должны совпадать.

Пример

стандартный ввод	стандартный вывод
4	-1 4
0 3	4 4
3 0	
3 6	
0 7	