

## Задача А. Построение

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

Группа солдат-новобранцев прибыла в армейскую часть N666. После знакомства с прапорщиком стало очевидно, что от работ на кухне по очистке картофеля спасти солдат может только чудо.

Прапорщик, будучи не в состоянии запомнить фамилии, пронумеровал новобранцев от 1 до  $N$ . После этого он велел им построиться по росту (начиная с самого высокого). С этой несложной задачей могут справиться даже совсем необученные новобранцы, да вот беда, прапорщик уверил себя, что знает про некоторых солдат, кто из них кого выше, и это далеко не всегда соответствует истине.

После трех дней обучения новобранцам удалось выяснить, что знает (а точнее, думает, что знает) прапорщик. Помогите им, используя эти знания, построиться так, чтобы товарищ прапорщик остался доволен.

### Формат входных данных

Сначала на вход программы поступают числа  $N$  и  $M$  ( $1 < N \leq 100$ ,  $1 \leq M \leq 5000$ ) — количество солдат в роте и количество пар солдат, про которых прапорщик знает, кто из них выше. Далее идут эти пары чисел  $A$  и  $B$  по одной на строке ( $1 \leq A, B \leq N$ ), что означает, что, по мнению прапорщика, солдат  $A$  выше, чем  $B$ .

### Формат выходных данных

В первой строке выведите «Yes» (если можно построиться так, чтобы прапорщик остался доволен) или «No» (если нет). После ответа «Yes» на следующей строке выведите  $N$  чисел, разделенных пробелами, — одно из возможных построений.

### Примеры

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

## Задача В. Бюрократия

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

Знаете ли вы, что для проведения клуба по созерцанию процесса сварки, преподаватель в первую очередь должен получить три справки трёх различных типов в Судиславском Центре Документов (СЦД)?

СЦД состоит из  $n$  кабинетов. В каждом кабинете выдают справку только одного типа. При этом известно, что после посещения кабинета, выдающего справку типа 1, посетителя отправляют в кабинет, выдающий справку типа 2, после посещения кабинета, выдающего справку типа 2, посетителя отправляют в кабинет, выдающий справку типа 3. Для полного торжества бюрократии, из кабинета, выдающего справку типа 3, посетителя отправляют в кабинет, выдающий справку типа 1.

Известно, что  $m$  преподавателей ЛКШ уже посетили СЦД, при этом по воспоминаниям  $i$ -го преподавателя, из кабинета  $a_i$  его отправили в кабинет  $b_i$  (возможно, совпадающий с  $a_i$ ). Вам поручено для каждого кабинета вычислить тип справок, который в нём выдаётся, или определить, что преподаватели что-то напутали. Если возможных подходящих решений несколько, то выведите любое.

### Формат входных данных

В первой строке входных данных заданы числа  $n$  и  $m$  ( $1 \leq n \leq 100\,000$ ,  $0 \leq m \leq 100\,000$ ) — количество кабинетов в СЦД и количество преподавателей ЛКШ, посетивших СЦД.

В каждой из следующих  $m$  строк заданы два числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n$ ), означающие, что сотрудник по воспоминаниям преподавателя  $i$  из кабинета  $a_i$  его отправили в кабинет  $b_i$ .

### Формат выходных данных

Если номера всех кабинетов невозможно разбить на три непересекающихся множества, так что все перенаправления будут проходить из кабинетов первого множества в кабинеты второго множества, из кабинетов второго множества в кабинеты третьего множества, или из кабинетов третьего множества в кабинеты первого множества, то в первой строке выходных данных выведите «NO» (без кавычек).

В противном случае в первой строке выходных данных выведите «YES» (без кавычек), после чего выведите ещё три строки.

Строка с номером  $i+1$  должна начинаться с неотрицательного числа  $s_i$  — количества кабинетов в  $i$ -м множестве в разбиении, после чего должны следовать  $s_i$  различных чисел от 1 до  $n$ , являющихся номерами кабинетов из  $i$ -го множества, перечисленными в произвольном порядке. Каждый кабинет должен попасть ровно в одно из трёх множеств, произвольные множества можно оставить пустыми. Если подходящих разбиений несколько, то разрешается вывести любое из них.

## Примеры

tripart.in	tripart.out
6 6 2 5 3 1 1 5 5 6 6 4 3 4	YES 3 1 2 4 1 5 2 3 6
2 2 1 2 1 2	YES 1 1 1 2 0
4 3 1 2 2 3 4 4	NO

## Задача С. Островные государства

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

Суровые феодальные времена переживала некогда великая островная страна Байтландия. За главенство над всем островом борются два самых сильных барона. Таким образом, каждый город страны контролируется одним из правителей. Как водится издревле, некоторые из городов соединены двусторонними дорогами. Бароны очень не любят друг друга и стараются делать как можно больше пакостей. В частности, теперь для того чтобы пройти по дороге, соединяющей города различных правителей, надо заплатить пошлину — один байтландский рубль.

Программист Вася живет в городе номер 1. С наступлением лета он собирается съездить в город  $N$  на Всебайтландское сборище программистов. Разумеется, он хочет затратить при этом как можно меньше денег и помочь ему здесь, как обычно, предлагается Вам.

**В этой задаче нельзя использовать алгоритм Дейкстры!**

### Формат входных данных

В первой строке входного файла записано два числа  $N$  и  $M$  ( $1 \leq N, M \leq 100\,000$ ) — количество городов и количество дорог соответственно.

В следующей строке содержится информация о городах —  $N$  чисел 1 или 2 — какому из баронов принадлежит соответствующий город.

В последних  $M$  строках записаны пары  $1 \leq a, b \leq N, a \neq b$ . Каждая пара означает наличие дороги из города  $a$  в город  $b$ . По дорогам Байтландии можно двигаться в любом направлении.

### Формат выходных данных

Если искомого пути не существует, выведите единственное слово `impossible`. В противном случае в первой строке напишите минимальную стоимость и количество посещенных городов, а во вторую выведите эти города в порядке посещения. Если минимальных путей несколько, выведите любой.

### Примеры

стандартный ввод	стандартный вывод
7 8	0 5
1 1 1 1 2 2 1	1 2 3 4 7
1 2	
2 5	
2 3	
5 4	
4 3	
4 7	
1 6	
6 7	

## Задача D. Числа

Имя входного файла: `numbers.in`  
Имя выходного файла: `numbers.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Витя хочет придумать новую игру с числами. В этой игре от игроков требуется преобразовывать четырехзначные числа не содержащие нулей при помощи следующего разрешенного набора действий:

1. Можно увеличить первую цифру числа на 1, если она не равна 9.
2. Можно уменьшить последнюю цифру на 1, если она не равна 1.
3. Можно циклически сдвинуть все цифры на одну вправо.
4. Можно циклически сдвинуть все цифры на одну влево.

Например, применяя эти правила к числу 1234, можно получить числа 2234, 1233, 4123 и 2341 соответственно.

Точные правила игры Витя пока не придумал, но пока его интересует вопрос, как получить из одного числа другое за минимальное количество операций.

### Формат входных данных

Во входном файле содержится два различных четырехзначных числа, каждое из которых не содержит нулей.

### Формат выходных данных

Программа должна вывести последовательность четырехзначных чисел, не содержащих нулей. Последовательность должна начинаться первым из данных чисел и заканчиваться вторым из данных чисел, каждое последующее число в последовательности должно быть получено из предыдущего числа применением одного из правил. Количество чисел в последовательности должно быть минимально возможным.

### Примеры

<code>numbers.in</code>	<code>numbers.out</code>
9876	9876
8876	8769
	8768
	8876

## Задача Е. Новогодние хлопоты

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

Дольше всего к Новому Году готовится, конечно, Дед Мороз, ведь ему надо подарить подарки огромному количеству детей. А прежде, чем дарить подарки их нужно выбрать, упаковать и написать на каждом адрес получателя. Но даже на этом работа не заканчивается. Прежде, чем начать развозить подарки нужно, сложить их все в большой красный мешок, подготовить сани Деда Мороза, накормить оленей и позвать Снегурочку.

Эти и другие необходимые дела секретарь Деда Мороза записал и пронумеровал. Также он выписал такие пары дел, что одно из них обязательно должно быть сделано до начала работы над вторым. Например, прежде, чем складывать подарки в мешок их нужно упаковать, а никак не наоборот. Гарантируется, что из таких зависимостей между делами никогда не следует, что некоторое дело должно быть сделано до себя самого.

Теперь секретарю интересно, за какое минимальное время можно выполнить все нужные дела. Можно считать, что на выполнение каждого дела уходит ровно один час. Помощников у Деда Мороза много, поэтому они могут работать над несколькими делами параллельно.

### Формат входных данных

В первой строке записаны два числа  $N$  и  $M$  — количество предновогодних дел и количество зависимостей между делами соответственно.  $1 \leq N, M \leq 10^5$ .

Следующие  $M$  строк описывают сами зависимости. В каждой из них записана пара чисел  $a, b$  и это означает, что дело  $a$  должно быть сделано до дела  $b$ .

### Формат выходных данных

Выведите минимальное время в часах, за которое можно закончить работу над всеми предновогодними делами.

### Примеры

<code>stdin</code>	<code>stdout</code>
5 3 1 2 2 3 4 5	3
10 0	1

## Задача F. Кратчайший путь коня

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

На шахматной доске размером  $8 \times 8$  заданы две клетки. Соедините эти клетки кратчайшим путем коня.

### Формат входных данных

Программа получает на вход координаты двух клеток, каждая в отдельной строке. Координаты клеток задаются в виде буквы (от "a" до "h") и цифры (от 1 до 8) без пробелов.

### Формат выходных данных

Программа должна вывести путь коня, начинающийся и заканчивающийся в данных клетках и содержащий наименьшее число клеток.

### Примеры

<code>stdin</code>	<code>stdout</code>
a1	a1
h8	b3
	c5
	d7
	e5
	f7
	h8

## Задача G. Доставка кефирчика

Имя входного файла: `kefir.in`  
Имя выходного файла: `kefir.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 64 мегабайта

Во время проведения очередной Межгалактической Летней Компьютерной Школы (МЛКШ) организаторы столкнулись с проблемой доставки кефирчика для вечерки. Дело в том, что кефирчик производят на планете под номером 1, а сами школьники живут на планете  $n$ , поэтому на доставку кефирчика тратится довольно большое время, а значит он успеваеет испортиться.

К счастью, галактическая транспортная система «Берендеев-Экспресс» постепенно внедряет новые кефиропроводы, способные передавать кефир со скоростью, в два раза превышающей скорость старых моделей. А именно, с любой планеты на любую по старым кефиропроводам кефир проходит за два года, а по новым — за один.

Разумеется, грешно было бы не воспользоваться инновационными технологиями, поэтому директор МЛКШ попросил вас написать программу, которая по данным о имеющихся кефиропроводах (как новых, так и старых) узнает кратчайший путь от планеты 1 до планеты  $n$ .

### Формат входных данных

В первой строке входного файла даны два целых числа  $n$  и  $m$  ( $1 \leq n \leq 100\,000$ ,  $0 \leq m \leq 100\,000$ ) — количество планет и количество кефиропроводов соответственно. В последующих  $m$  строках даны тройки натуральных чисел  $u_i$ ,  $v_i$  и  $c_i$ . Числа  $u_i$  и  $v_i$  обозначают номера планет, соединенных  $i$ -м кефиропроводом, а  $c_i$  ( $c_i = 1$  или  $c_i = 2$ ) — количество лет, которое потребуется, чтобы передать кефир с одной планеты на другую через  $i$ -й кефиропровод. Планеты во входном файле нумеруются с единицы. Кефир по трубопроводам можно передавать в обоих направлениях.

### Формат выходных данных

В выходной файл требуется вывести одно число — количество лет, которое требуется, чтобы доставить кефир с планеты 1 на планету  $n$ . Если доставка невозможна, то в выходной файл требуется вывести «-1».

### Примеры

<code>kefir.in</code>	<code>kefir.out</code>
3 2 1 2 2 2 3 1	3
3 1 2 3 1	-1
2 5 1 2 1 1 2 2 1 2 1 1 1 2 2 2 1	1

## Задача Н.

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

Нужно найти кратчайший путь из одной строки из нулей и единиц в другую, если можно делать следующие действия:

- развернуть любой префикс строки;
- развернуть любой суффикс строки;
- поменять любой бит на противоположный.

### Формат входных данных

В первой строке входного файла дано число  $N$  ( $1 \leq N \leq 14$ ) — количество битов в изначальной строке. Во второй строке входного файла дана изначальная строка из нулей и единиц размера  $N$ . В третьей строке выходного файла дана конечная строка из нулей и единиц размера  $N$ .

### Формат выходных данных

В первой строке выходного файла выведите  $K$  — количество действий, которое нужно сделать, чтобы получить из изначальной строки конечную. В следующих  $K + 1$  строках выведите все промежуточные строки на этом пути.

### Пример

стандартный ввод	стандартный вывод
5	2
01011	01011
11110	01111
	11110

## Задача I. Цивилизация (версия для Python)

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

Карта мира в компьютерной игре «Цивилизация» версии 1 представляет собой прямоугольник, разбитый на квадратики. Каждый квадратик может иметь один из нескольких возможных рельефов, для простоты ограничимся тремя видами рельефов — поле, лес и вода. Поселенец перемещается по карте, при этом на перемещение в клетку, занятую полем, необходима одна единица времени, на перемещение в лес — две единицы времени, а перемещаться в клетку с водой нельзя.

У вас есть один поселенец, вы определили место, где нужно построить город, чтобы как можно скорее завладеть всем миром. Найдите маршрут переселенца, по которому можно прийти в место строительства города за минимальное время. На каждом ходе переселенец может перемещаться в клетку, имеющую общую сторону с той клеткой, где он сейчас находится.

### Формат входных данных

Во входном файле записаны два натуральных числа  $N$  и  $M$ , не превосходящих 300 — размеры карты мира ( $N$  — число строк в карте,  $M$  — число столбцов). Затем заданы координаты начального положения поселенца  $x$  и  $y$ , где  $x$  — номер строки,  $y$  — номер столбца на карте ( $1 \leq x \leq N$ ,  $1 \leq y \leq M$ ), строки нумеруются сверху вниз, столбцы — слева направо. Затем аналогично задаются координаты клетки, куда необходимо привести поселенца.

Далее идет описание карты мира в виде  $N$  строк, каждая из которых содержит  $M$  символов. Каждый символ может быть либо «.» (точка), обозначающим поле, либо «W», обозначающим лес, либо «#», обозначающим воду.

Гарантируется, что начальная и конечная клетки пути переселенца не являются водой.

### Формат выходных данных

В первой строке выходного файла выведите количество единиц времени, необходимое для перемещения поселенца (перемещение в клетку с полем занимает 1 единицу времени, перемещение в клетку с лесом — 2 единицы времени). Во второй строке выходного файла выведите последовательность символов, задающих маршрут переселенца. Каждый символ должен быть одним из четырех следующих: «N» (движение вверх), «E» (движение вправо), «S» (движение вниз), «W» (движение влево). Если таких маршрутов несколько — выведите любой из них.

Если дойти из начальной клетки в конечную невозможно, выведите число -1.

### Примеры

<code>civ.in</code>	<code>civ.out</code>
<pre>4 8 1 1 4 8 ...WWW .#####. .#..W... ...WWW.</pre>	<pre>13 SSSEENEEEEES</pre>
<pre>4 7 2 2 3 6 ##### #WW#.# #WW#.# #####</pre>	<pre>-1</pre>

## Задача J. Кони ходят по очереди

Имя входного файла: knight3.in  
Имя выходного файла: knight3.out  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Переведите каждого из двух коней из одной клетки в другую за наименьшее общее число ходов. Два коня не могут одновременно находиться в одной клетке. **Ходы коней должны чередоваться.**

### Формат входных данных

Во входном файле записаны координаты первого и второго коня, затем координаты клеток, куда нужно их переместить.

### Формат выходных данных

Программа должна вывести последовательность ходов коней в виде нескольких строк. Первым символом в строке должен быть номер коня (1 или 2), затем, через пробел, координаты клетки, в которую он переставляется. Необходимо вывести любое из возможных оптимальных решений. Кони должны ходить по очереди, первым может ходить любой из коней, кони могут сделать различное число ходов.

### Примеры

knight3.in	knight3.out
a1	1 b3
c2	2 a1
c2	1 d4
a1	2 b3
	1 c2
	2 a1

## Задача К. Вокруг света за 80 дней

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Филлеасу Фоггу и Паспарту опять пришлось поспорить со Стюартом о том, что они смогут добраться из города Виллабаджо с номером  $s$  в город Вилларибо с номером  $t$  не более, чем за  $k - 1$  день. А так как Филлеас Фогг перфекционист, то он хочет добраться из первого города в последний за минимально возможное число дней. Задание показалось Филлеасу простым, однако внезапно в Англии ввели новую систему поездов, согласно которой поезда идут из города с номером  $v$  в город с номером  $u$  тогда и только тогда, когда  $u$  является делителем  $v$  ( $u \leq v$ ) или  $u - v$  является простым делителем  $v$  или единицей ( $u \geq v$ ). Помогите Филлеасу Фоггу и Паспарту понять, смогут ли они доехать из Виллобаджо в Вилларибо не более, чем за  $k - 1$  день, если поездка по железной дороге между двумя различными городами занимает ровно один день.

Помните, что железные дороги в Англии односторонние.

### Формат входных данных

На первой и единственной строке вводятся три числа  $s, t, k$  ( $1 \leq s, t \leq 3 \cdot 10^5, 0 \leq k \leq 10^3$ ).

### Формат выходных данных

Если Филлеас Фогг и Паспарту не смогут выполнить условия спора, то выведите в первой строке  $-1$  и завершите программу, иначе в первой строке выведите  $n$  — минимальное число городов, которые посетят герои за свое путешествие, а во второй строке выведите  $n$  чисел — города в порядке посещения.

### Примеры

стандартный ввод	стандартный вывод
2 7 3	-1
2 7 4	4 2 3 6 7
57 179 7	7 57 58 59 118 177 178 179