

Задача А. Честное столовластие

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

После разработки закона о введении налога на постройку зданий выше двух этажей у n чиновников короля Маркеля Весёлого осталось k бюджетных золотых монет. Эти деньги они решили частично поделить между собой. Поскольку на службе у Короля состоят лишь честные чиновники, делить их они решили справедливо. Они знают, какой вклад w_i внес в законопроект i -й чиновник. Деньги должны быть распределены так, что если i -й чиновник получил c_i монет, а j -й — c_j монет, то числа c_i и c_j должны иметь такой же результат сравнения, как числа w_i и w_j . Это значит, что $c_i < c_j$ должно быть верно тогда и только тогда, когда $w_i < w_j$. В частности, из этого следует, что при равенстве вкладов чиновники должны получить одинаковое количество монет.

Среди чиновников у короля Маркеля Весёлого есть m фаворитов. Он хочет, чтобы фаворит с минимальным вкладом в законопроект получил в итоге как можно больше денег. Помогите честным госслужащим разделить деньги по справедливости, соблюдая требование короля Маркеля. Заметьте, что в задачу не входит максимизация прибыли чиновников, не являющихся фаворитами Маркеля Весёлого.

Формат входных данных

В первой строке следуют целое положительное число n и два целых неотрицательных числа m и k , разделённых пробелами — количество чиновников, количество фаворитов Маркеля и количество монет для раздела соответственно ($m \leq n \leq 100\,000$, $k \leq 10^{18}$).

Во второй строке находятся n целых положительных чисел w_i , разделённых пробелами ($1 \leq w_i \leq 10^9$). i -е из них обозначает вклад, который привнес i -й чиновник.

В третьей строке записаны m различных натуральных чисел f_i — номера чиновников, являющихся фаворитами короля ($1 \leq f_i \leq n$).

Формат выходных данных

Если справедливое распределение монет существует, первой строкой выведите слово «POSSIBLE» (без кавычек). Второй же строкой выведите n целых неотрицательных чисел через пробел. i -е из них должно равняться количеству монет, которые получит i -й чиновник.

Помните, что раздавать все монеты не обязательно, а Маркель Весёлый волнуется только о том, чтобы фаворит с минимальным вкладом (а значит и минимальным количеством монет) получил как можно больше, а в остальном чиновники вольны распределять монеты как угодно. Это значит, что и вы можете вывести любой правильный ответ.

Если же поделить деньги справедливо невозможно, выведите слово «IMPOSSIBLE» (без кавычек)

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Доп. ограничения	Необх. подзадачи
0	0	Тесты из условия	
1	25	$n \leq 10, k \leq 50$	
2	25	$n \leq 5000, k \leq 5000$	1
3	50	нет	1, 2

Примеры

стандартный ввод	стандартный вывод
5 2 15 15 2 33 12 8 3 4	POSSIBLE 4 0 5 3 1
4 2 3 1 2 3 2 1 2	IMPOSSIBLE

Замечание

Обратите внимание, что некоторая часть денег может остаться неподделённой. В первом примере входных данных излишек монет можно было бы также раздать второму и пятому чиновнику, но, так как они не являются фаворитами короля, делать это не обязательно.

Во втором примере входных данных корректный делёж потребует не менее четырёх монет, тогда как их осталось всего три.

Задача В. Турист Петр

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

На отдыхе в Теплой Стране Вера познакомилась с симпатичным волейболистом-трактористом Петром. Турист Петр, кстати, собирается после отличного отдыха в Теплой Стране отправиться в путешествие по городам Европы. Как известно, Европа обладает развитой транспортной системой: в Европе есть v интересующих Петра городов и e маршрутов ночных поездов. Каждый маршрут соединяет два различных города, время в пути составляет одну ночь. Поезда по маршруту ходят в обоих направлениях.

Основной целью поездки Петра является осмотр местных достопримечательностей. Поскольку Петр — невероятно занятой человек, то он решил, что все путешествие должно занимать не более четырех дней. Петр уже многое повидал, поэтому на осмотр достопримечательностей в каждом городе Петр тратит ровно один день. Он хочет составить наиболее практичный тур: каждый день он будет тратить на осмотр города, а каждую ночь — на переезд ночным поездом между городами. Разумеется, Петр не имеет ни малейшего желания посещать один город несколько раз.

Но на этом прагматичность Петра не заканчивается: Петр, как настоящий турист, хочет посмотреть на самые красивые европейские достопримечательности. Он долго изучал справочники и для каждого города оценил свою ожидаемую радость от его посещения p_i . Теперь он хочет найти маршрут, при котором его суммарная радость от посещения городов будет наибольшей. Помогите Петру найти такой маршрут.

Формат входных данных

В первой строке входных данных заданы два целых числа v и e ($1 \leq v, e \leq 300\,000$) — количество городов и маршрутов поездов, соответственно.

В следующей строке заданы v целых чисел p_i ($1 \leq p_i \leq 10^8$), где p_i обозначает ожидаемую радость от посещения города с номером i .

В следующих e строках заданы описания маршрутов поездов. Каждое описание состоит из пары различных чисел a_i и b_i ($1 \leq a_i, b_i \leq v$) — номеров городов, между которыми курсирует этот маршрут поезда. Гарантируется, что между каждой парой городов существует не более одного маршрута поезда.

Формат выходных данных

В первой строке выходных данных выведите число k ($1 \leq k \leq 4$) — количество городов в оптимальном маршруте туриста Петра.

В следующей строке выведите номера этих городов в порядке посещения. Города нумеруются начиная с единицы.

Если оптимальных маршрутов несколько, выведите любой из них.

Примеры

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

Замечание

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Доп. ограничения	Необх. подзадачи
0	0	Тесты из условия	
1	20	$v, e \leq 100$	
2	20	$v, e \leq 1\,000$	1
3	30	$v \leq 3\,000, e \leq 60\,000$	1, 2
4	30	нет	1, 2, 3

Задача С. Ксилофон

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

Это интерактивная задача.

ЮИ-кун купил ксилофон, состоящий из n деревянных дощечек. Дощечки расположены в ряд и пронумерованы целыми числами от 1 до n слева направо. Дощечка с номером i имеет длину a_i ($1 \leq a_i \leq n$). Известно, что длины дощечек попарно различны. Также ЮИ-кун знает, что **дощечка с минимальной длиной расположена левее, чем дощечка с максимальной длиной**.

Вы хотите купить себе такой же ксилофон, а поэтому хотите выяснить длину каждой дощечки в музыкальном инструменте ЮИ-куна. Для этого вы можете выбрать два целых числа l и r , таких что $1 \leq l \leq r \leq n$, и сообщить их ЮИ-куну. В качестве ответа он сообщит вам разность длин самой длинной и самой короткой дощечек среди дощечек a_l, a_{l+1}, \dots, a_r .

Сделав данный запрос не более, чем 10 000 раз, выясните длины всех дощечек в ксилофоне ЮИ-куна!

Протокол взаимодействия

В начале ваша программа должна считать одно целое число n ($2 \leq n \leq 5\,000$) — количество дощечек в ксилофоне.

После этого вы можете не более, чем 10 000 раз отправить запрос программе жюри. Для этого вы должны вывести запрос в следующем формате: «? l r » ($1 \leq l \leq r \leq n$). После этого ваша программа должна считать одно целое число d ($0 \leq d \leq n - 1$) — разницу в длинах самой длинной и самой короткой дощечек среди дощечек с номерами $l, l + 1, \dots, r$.

После того, как вы выясните длины всех дощечек, вы должны вывести ответ в следующем формате: «! a_1 a_2 ... a_n » ($1 \leq a_i \leq n$). Все a_i должны быть попарно различны. Данное сообщение не учитывается при подсчете количества сделанных запросов. После этого сообщения ваша программа должна завершить работу.

Не забывайте выполнять операцию `flush` после каждого запроса (в том числе и последнего).

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Доп. ограничения	Необх. подзадачи
0	0	Тесты из условия	
1	11	$n \leq 100$	
2	36	$n \leq 1\,000$	1
3	15	$n \leq 3\,000$	1, 2
4	10	$n \leq 4\,950$	1, 2, 3
5	28	нет	1, 2, 3, 4

Пример

стандартный ввод	стандартный вывод
5	? 1 5
4	? 3 5
2	! 2 1 5 3 4

Задача D. Парковка

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

После не самого удачного выступления на одной известной олимпиаде, сборная одного известного города отправилась работать на парковке. Их задача заключалась в том, чтобы переставлять машины.

На парковке находятся $2n$ машин, которые покрашены в n различных цветов, при этом для каждого цвета есть ровно 2 машины такого цвета. Далее будем считать, что цвета пронумерованы от 1 до n .

Сама парковка состоит из m парковочных ячеек, пронумерованных от 1 до m , при этом в каждой ячейке одновременно может находиться не более двух машин. В каждую ячейку есть ровно один въезд, и машина может занять или покинуть место, если и только если никакая другая машина не блокирует въезд. Машина называется *верхней*, если она занимает верхнюю половину ячейки. Аналогично, машина является *нижней*, если она занимает нижнюю половину ячейки. Оказалось, что машины припаркованы так, что: либо ячейка полностью заполнена, либо занята только нижняя ее половина, либо она полностью пуста.

Ученики хотят перепарковать машины так, что каждая пара машин одинакового цвета будет находиться в одной ячейке. При этом им абсолютно плевать на то, какая ячейка будет занята каким цветом, а также на то, какая конкретно машина будет верхней, а какая нижней.

Они будут переставлять машины следующим образом:

1. Для начала надо выбрать любую машину, которая может выехать из своей ячейки.
2. Затем надо выбрать либо полностью пустую ячейку, либо ячейку, содержащую машину **того же цвета**, что и текущая.
3. Затем выбранная машина переедет в выбранную ячейку. Если ячейка была пустой, то машина займет нижнюю половину. Иначе будет занята верхняя половина.

У учеников нет водительских прав, поэтому им хочется минимизировать количество операций, которое потребуется для перестановки машин.

Формат входных данных

В первой строке даны числа n и m ($1 \leq n \leq m \leq 200\,000$) — количество цветов и количество парковочных ячеек.

В следующих m строках содержатся пары чисел b_i и t_i ($0 \leq b_i, t_i \leq n$), которые обозначают соответственно нижнюю и верхнюю машины в i -й ячейке. Если $b_i = 0$, то вся ячейка пустая. Иначе, если $t_i = 0$, то нижняя половина занята машиной с цветом b_i . Иначе нижняя половина занята машиной с цветом b_i , а верхняя — машиной с цветом t_i . Гарантируется, что если $b_i = 0$, то и $t_i = 0$ (иначе говоря, не может быть заполнена только верхняя половина).

Формат выходных данных

Если не существует последовательности операций, с помощью которой можно добиться необходимого расположения машин, выведите -1 .

Иначе в первой строке выведите число k — минимальное количество операций, необходимое для достижения требуемого результата.

В следующих k строках выведите сами операции. В i -й строке выведите числа x_i и y_i ($1 \leq x_i, y_i \leq m, x_i \neq y_i$), означающие, что ученики должны переставить машину из ячейки x_i в ячейку y_i . Необходимо, чтобы в ячейке x_i была хотя бы одна машина, а ячейка y_i либо пустая, либо содержит ровно одну машину, которая должна быть того же цвета.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

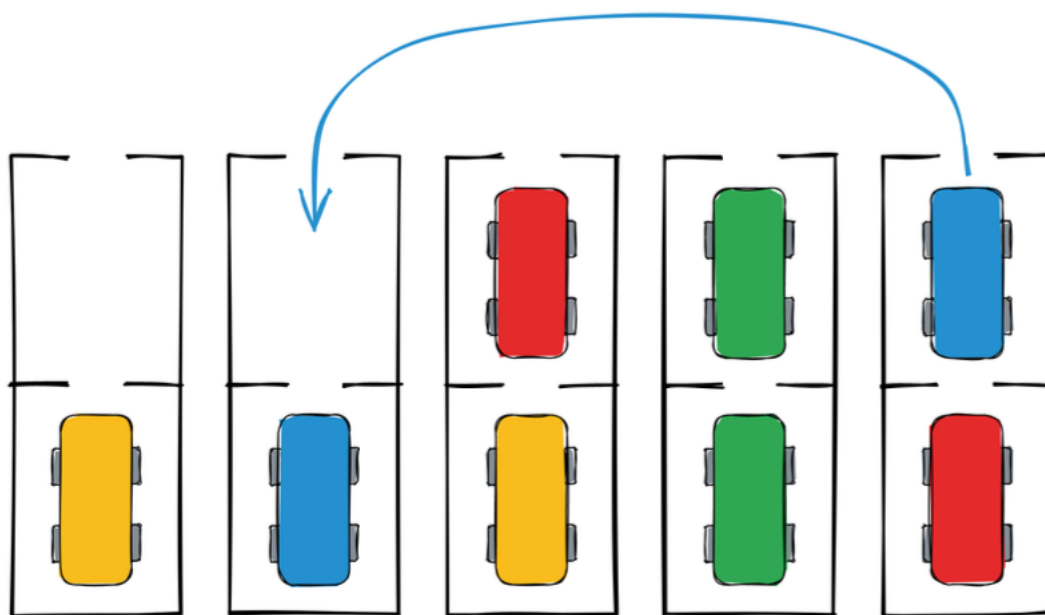
Подзадача	Баллы	Доп. ограничения	Засисит от
0	0	Тесты из условия	
1	10	$m \leq 4$	
2	10	$2 \cdot n \leq m$	
3	25	Изначально каждая ячейка пустая или полная, $n \leq 1000$	
4	15	Изначально каждая ячейка пустая или полная	3
5	25	$n \leq 1000$	3
6	15	нет	1 — 5

Примеры

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

Замечание

В первом примере машины изначально расставлены следующим образом:



Стрелочкой показан единственный вариант для первой операции.