

## Задача А. Прибытие короля

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

В управлении казны Тридевятого королевства переполох — прибывает король Иван Тридцать Третий с целью проверить работу своих поданных. Известно, что король очень любит аккуратность. Управляющему казной Иннокентию необходимо срочно навести порядок в трех королевских сундуках: уравнивать количество монет в них. Изначально в сундуках находятся  $a$ ,  $b$  и  $c$  монет, соответственно.

Чтобы уравнивать количество монет в сундуках, Иннокентий начал в панике добавлять в них монеты обеими руками. Каждую секунду он выбирает два различных сундука, и одновременно кладёт в каждый из них по монете.

Помогите Иннокентию посчитать, сколько секунд ему потребуется, чтобы уравнивать количество монет в сундуках.

### Формат входных данных

Ввод состоит из трёх строк.

В первой строке задано число  $a$ , во второй строке задано число  $b$ , в третьей строке задано число  $c$  ( $1 \leq a, b, c \leq 5 \cdot 10^8$ ).

### Формат выходных данных

Выведите одно число — минимальное количество операций, которые надо совершить Иннокентию, чтобы уравнивать количество монет в сундуках.

### Примеры

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

### Замечание

В первом тесте одним из оптимальных вариантов будет в первую секунду добавить по монете в 1 и 3 сундуки, затем в 1 и 2, и еще раз в 1 и 2.

Во втором тесте в сундуках изначально равное количество монет.

## Задача В. Кассовый разрыв

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

В недавно образовавшейся компании «NERC» (New Era Russian Coders), разумеется, есть бухгалтерия. И их очень беспокоит финансовое состояние компании. Особенно им не хочется, чтобы случился *кассовый разрыв* — ситуация, в которой требуется выплатить больше денег, чем есть сейчас на счёте.

На данный момент на счёте компании находится  $s$  рублей. Главный бухгалтер подготовил план операций на  $m$  ближайших дней. На этот период запланированы  $n$  операций по получению и переводу денег.

Для каждой операции известно на сколько после её выполнения изменяется счет компании, однако точная дата операции неизвестны. Деньги, переведённые клиентами, могут поступить на счёт не сразу, с другой стороны, требования оплатить счета могут поступить внезапно. Для каждой операции, таким образом, известен лишь диапазон дней, в которые эта операция может осуществиться. Если несколько операций выполняются в один и тот же день, они могут оказаться выполнены в любом порядке.

Помогите бухгалтерии проверить, верно ли, что при любой возможной последовательности выполнения операций не произойдёт кассового разрыва.

### Формат входных данных

В первой строке через пробел заданы числа  $n$ ,  $m$  и  $s$  — количество платежей, количество дней в плане и исходное количество денег на счете ( $1 \leq n, m \leq 1000$ ;  $1 \leq s \leq 10^6$ ).

В  $i$ -й из следующих  $n$  строк задано описание  $i$ -го платежа в формате «count from to», означающее, что в какой-то из дней с **from** по **to**, включительно, счет изменится на **count** рублей ( $-10^6 \leq \text{count} \leq 10^6$ ;  $1 \leq \text{from} \leq \text{to} \leq m$ ).

### Формат выходных данных

Выведите «YES», если в процессе выполнения операций может произойти кассовый разрыв, и «NO», если такая ситуация невозможна.

### Примеры

стандартный ввод	стандартный вывод
4 3 100 100 1 2 -100 1 2 1 2 3 0 3 3	NO
4 3 100 100 1 2 -100 1 2 1 2 3 -1 2 2	YES

### Замечание

В первом примере есть единственный перевод со счета, и при этом перед ним заведомо будет хотя бы 100 рублей, изначально имевшихся на счете, то есть кассовый разрыв невозможен.

Во втором примере могло случиться так, что оба перевода со счета: на 100 рублей и на 1 рубль, будут запрошены во второй день, раньше всех остальных переводов. В таком случае на момент второго из них денег на счету уже не останется — это кассовый разрыв.

## Задача С. Реактивные поезда

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

3019 год. Всё человечество давно живёт в городах, парящих в атмосфере на аэростатах и реактивных двигателях. На планете есть  $n$  городов, между некоторыми парами городов курсируют реактивные поезда. Если между городами  $a$  и  $b$  курсируют реактивные поезда, то пассажиры могут перемещаться между этими городами в любом направлении. Будем говорить, что из города  $a$  достижим город  $b$ , если пассажир может добраться от  $a$  до  $b$  с использованием реактивных поездов, возможно с пересадками.

Люди в 3019 любят дружить, и дружат не парами, а целыми городами. Отношение дружбы взаимно: если город  $a$  дружит с городом  $b$ , то город  $b$  также дружит с городом  $a$ .

Время от времени в каком-нибудь городе решают организовать праздник и зовут всех своих друзей к себе в гости. Если праздник проходит в городе  $a$ , то гости из всех городов, которые дружат с  $a$ , и из которых достижим город  $a$ , прибывают на мероприятие, пользуясь реактивными поездами.

Для того, чтобы оценивать количество гостей на празднике, решено было разработать специальную информационную систему «Праздник 3019». Задан список текущих рейсов реактивных поездов, а также информация о том, какие города дружат. Информационная система должна обрабатывать запросы вида: «Если в городе  $v$  случится праздник, гости из сколько городов попадут на него?» Кроме того, должна поддерживаться возможность добавить информацию о новом рейсе реактивного поезда, а также о новой паре городов, которые начинают дружить. К счастью, рейсы не отменяются, и однажды подружившись, города навсегда остаются друзьями.

Помогите человечеству разработать описанную информационную систему.

### Формат входных данных

В первой строке заданы три целых числа  $n$ ,  $m$  и  $k$  — количество городов, пар исходно дружащих городов и рейсов реактивных поездов ( $1 \leq n \leq 10^5$ ,  $0 \leq m, k \leq 10^5$ ).

В следующих  $m$  строках находятся по два целых числа  $a$ ,  $b$  ( $1 \leq a, b \leq n$ ,  $a \neq b$ ) — пары дружащих городов. Каждая пара дружащих городов указана во вводе ровно один раз.

В следующих  $k$  строках находятся по два целых числа  $a$ ,  $b$  ( $1 \leq a, b \leq n$ ,  $a \neq b$ ) — пары городов, соединённых рейсами реактивных поездов. Каждая пара городов соединена не более, чем одним рейсом.

Следующая строка содержит число  $q$  ( $0 \leq q \leq 10^5$ ) — количество запросов, которые следует обработать, а в следующих  $q$  строках заданы сами запросы.

- Запрос «T  $a$   $b$ » означает, что между городами  $a$  и  $b$  запущен рейс реактивного поезда ( $1 \leq a, b \leq n$ ,  $a \neq b$ ). Гарантируется, что до этого запроса между этими городами рейса не было.
- Запрос «F  $a$   $b$ » означает, что города  $a$  и  $b$  начинают дружить ( $1 \leq a, b \leq n$ ,  $a \neq b$ ). Гарантируется, что до этого запроса эти города не дружили.
- Запрос «?  $v$ » означает, что к информационной системе сделан запрос: «Если в городе  $v$  случится праздник, гости из сколько городов попадут на него?» ( $1 \leq v \leq n$ )

### Формат выходных данных

Для каждого запроса «?  $v$ » выведите ответ на него в отдельной строке.

## Пример

стандартный ввод	стандартный вывод
4 2 2	1
1 2	2
1 3	3
1 2	
1 4	
5	
? 1	
F 4 1	
? 1	
T 4 3	
? 1	

## Замечание

Ответ на первый запрос равен единице, потому что только один из городов 2, 3 соединен с городом 1 рейсом поездов. Ответ на второй запрос равен двойке, потому что перед этим города 1 и 4 подружились, а они были соединены рейсом напрямую. Ответ на третий вопрос равен тройке, потому что от города 3 стало возможным добраться на поездах до города 1 следующим образом: сначала доехать до города 4, а потом от него до города 1.

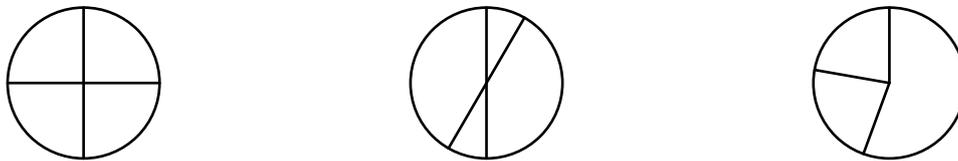
## Задача D. Нарезка пиццы

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Заработавшиеся допоздна члены технического комитета олимпиады заказали пиццу. Когда им доставили круглую, вкусную и сытную пиццу, они столкнулись с проблемой: им хочется разрезать пиццу как можно быстрее.

Каждый член технического комитета озвучил число  $\alpha_i$ , характеризующее сколько пиццы он хочет съесть. Для удобства они решили задавать угловую меру желаемого куска в градусах: число  $\alpha_i$  означает, что этот человек хочет получить ровно один кусок пиццы в форме сектора с угловой мерой  $\alpha_i$  градусов.

Для разрезания пиццы используется специальный нож для пиццы. С его помощью можно делать два типа разрезов: прямые разрезы от центра пиццы до края (нож проходит радиус круга) и прямые разрезы от края до края через центр (нож проходит диаметр круга). После разрезания полученные сектора можно раздать в любом порядке, по одному каждому едоку, а оставшиеся куски остаются в коробке.



Какое минимальное число разрезов необходимо сделать, чтобы каждому члену технического комитета достался в точности запрашиваемый им кусок пиццы?

### Формат входных данных

В первой строке находится целое число  $n$  — количество членов технического комитета ( $1 \leq n \leq 16$ ). В следующей строке находится  $n$  целых чисел  $\alpha_i$  — угловые меры в градусах желаемых секторов ( $1 \leq \alpha_i < 360$ ).

Гарантируется, что сумма всех  $\alpha_i$  не превосходит 360.

### Формат выходных данных

В первой строке выведите число  $m$  — минимальное количество разрезов, которое надо сделать. В следующих  $m$  строках выведите сами разрезы.

Каждый разрез задаётся двумя целыми числами  $\beta_i$  ( $0 \leq \beta_i < 360$ ) и  $type_i$  — углом поворота и типом разреза. Угол поворота отмеряется от зафиксированной оси в зафиксированном направлении (например, от оси направленной на север, по часовой стрелке). Тип разреза  $type_i$  равен 0, если разрез совпадает с радиусом, и равен 1, если разрез совпадает с диаметром.

### Примеры

стандартный ввод	стандартный вывод
4 90 90 90 90	2 0 1 90 1
2 30 30	2 0 1 30 1
3 200 80 80	3 0 0 200 0 280 0

## Задача Е. Единственное решение

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Владимир Геннадьевич готовит задание по высшей математике для студентов.

Задание планируется такое. Дано  $n$  целых чисел  $x_1, x_2, \dots, x_n$  и натуральное число  $m$ ,  $1 \leq m < 2^n$ .

Требуется выбрать  $n$  целых чисел  $a_1, a_2, \dots, a_n$ , каждое из которых равно 0,  $-1$  или  $1$ , при этом хотя бы одно из них должно быть отлично от 0, а сумма  $a_1x_1 + a_2x_2 + \dots + a_nx_n$  должна делиться на  $m$  без остатка.

Владимир Геннадьевич придумал ответ на задание, которое он хочет дать студентам: он выбрал  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $-1 \leq a_i \leq 1$ ), хотя бы одно из них не равно 0. Чтобы выполненные задания было легче проверить, он хочет придумать такие числа  $x_1, x_2, \dots, x_n$  и натуральное число  $m$ , чтобы выбранные им числа  $a_1, a_2, \dots, a_n$  были единственным подходящим решением. К сожалению, это невозможно, так как числа  $-a_1, -a_2, \dots, -a_n$  также подходят.

Тогда Владимир Геннадьевич ослабил свои требования: он хочет, чтобы других решений, кроме  $a_1, a_2, \dots, a_n$  и  $-a_1, -a_2, \dots, -a_n$  не существовало.

Помогите ему выбрать числа  $x_1, x_2, \dots, x_n$  и  $m$  для задания.

### Формат входных данных

В первой строке находится число  $n$  ( $1 \leq n \leq 30$ ).

В следующей строке дано  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $-1 \leq a_i \leq 1$ ). Гарантируется, что хотя бы одно из чисел  $a_i$  не равно 0.

### Формат выходных данных

В первой строке выведите натуральное число  $m$  ( $1 \leq m < 2^n$ ).

В следующей строке выведите  $n$  целых чисел  $x_1, x_2, \dots, x_n$  ( $-2^{30} < x_i < 2^{30}$ ).

Если подходящих вариантов несколько, выведите любой из них.

Гарантируется, что решение всегда существует.

### Пример

стандартный ввод	стандартный вывод
2	3
1 -1	1 4

### Замечание

В примере студенты должны выбрать числа  $a_1$  и  $a_2$ , чтобы число  $a_1 + 4a_2$  делилось на 3. Подходят два решения:

- $a_1 = 1, a_2 = -1$  ( $a_1 + 4a_2 = 1 - 4 = -3$ , делится на 3) и
- $a_1 = -1, a_2 = 1$  ( $a_1 + 4a_2 = -1 + 4 = 3$ , делится на 3).

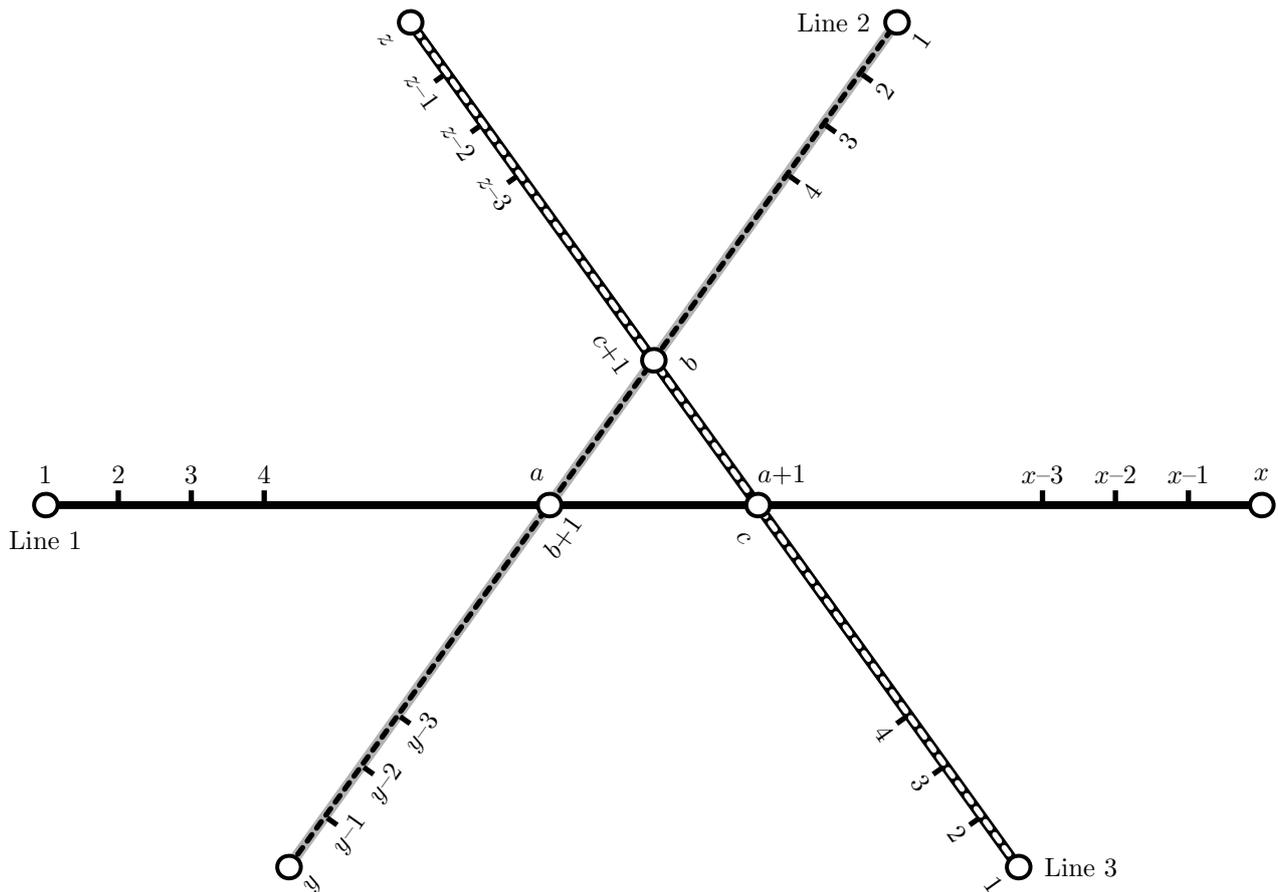
Требования Владимира Геннадьевича выполнены.

## Задача F. Метро 2345

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

В Кекополисе действует три линии метро, которые пересекаются в центре города, образуя треугольник из трёх перегонов. На первой линии  $x$  станций, на второй линии  $y$  станций, а на третьей линии  $z$  станций. Станции на каждой линии пронумерованы подряд, начиная с 1.

На станциях, где линии пересекаются, организованы пересадки, можно переходить между станцией  $a$  первой линии и станцией  $b + 1$  второй линии, станцией  $a + 1$  первой линии и станцией  $c$  третьей линии, а также станцией  $b$  второй линии и станцией  $c + 1$  третьей линии.



Линии строились в разное время, поэтому поезда по ним ходят с разной скоростью. Так, поезд на первой линии проезжает один перегон за  $t_1$  секунд, на второй за  $t_2$ , а на третьей за  $t_3$ . Для пересадки с одной линии на другую требуется  $d$  секунд. Время входа и выхода на станцию и в поезд, а также время остановки поезда на станциях, будем считать пренебрежимо малым.

Для того, чтобы попасть на Командную Олимпиаду Школьников Кекополиса по Информатике, Диме надо доехать от дома, который находится около станции  $i$  на  $k$ -й линии до Университета Кекополиса, который находится около станции  $j$  на  $l$ -й линии. Поскольку Дима проспал и опаздывает, он хочет доехать как можно быстрее.

Определите, за какое минимальное время Диме удастся доехать от дома до университета. Дима не входит и не выходит из метро на станции пересадки.

### Формат входных данных

Первая строка ввода содержит три целых числа  $x$ ,  $y$  и  $z$  ( $2 \leq x, y, z \leq 10^9$ ).

Вторая строка ввода содержат три целых числа  $a$ ,  $b$  и  $c$  ( $1 \leq a < x$ ,  $1 \leq b < y$ ,  $1 \leq c < z$ ).

Третья строка ввода содержит четыре целых числа  $t_1$ ,  $t_2$ ,  $t_3$  и  $d$  ( $1 \leq t_1, t_2, t_3, d \leq 10^9$ ).

Четвертая строка содержит четыре целых числа  $k$ ,  $i$ ,  $l$  и  $j$  ( $1 \leq k, l \leq 3$ , гарантируется, что на линии  $k$  есть станция с номером  $i$ , а на линии  $l$  есть станция с номером  $j$ , начальная и конечная

станции маршрута Димы различны и между ними нет прямой пересадки, то есть придётся проехать хотя бы один перегон).

### Формат выходных данных

Выведите одно число — минимальное время, которое Диме требуется, что доехать от начальной станции до конечной.

### Примеры

стандартный ввод	стандартный вывод
4 4 4 2 2 2 1 1 1 1 1 1 2 1	4
4 4 4 2 2 2 1 10 1 1 1 1 3 4	5
4 4 4 2 2 2 1 1 1 1 1 1 1 4	3

## Задача G. Переполнение

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Недавно Джон заказал на одном известном сайте электронные чашечные весы. У весов есть две чаши, на которые можно класть предметы. Весы показывают, какая из чаш перевешивает, либо сообщает, что веса наборов предметов на чашах равны.

Джон начал экспериментировать с весами и обнаружил, что показания весов иногда бывают довольно странными. Оказалось, что весы вычисляют суммарный вес предметов на каждой чаше, а затем сравнивают получившиеся значения. Однако поскольку софт для весов писали бывшие олимпиадники, весы вычисляют остаток от деления веса положенных на чашу предметов на число  $m$ .

У Джона есть  $n$  гирь,  $i$ -я гиря имеет вес  $a_i$ . Он хочет, чтобы весы сообщили, что чаши уравновешены, положив на весы некоторые из этих гирь.

Помогите Джону решить, какие гири на какие чаши ему необходимо положить. Джон должен положить на весы хотя бы одну гирю.

### Формат входных данных

В первой строке ввода даны два натуральных числа  $n$  и  $m$  — количество гирь у Джона и значение, по модулю которого весы вычисляют вес ( $1 \leq n \leq 25$ ;  $1 \leq m < 4 \cdot 10^7$ ).

В следующей строке перечислены  $n$  чисел  $a_1, \dots, a_n$  — веса гирь Джона ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Если ответ существует, на первой строке выведите  $k$  — количество гирь, которые Джон должен положить на первую чашу весов. На второй строке выведите номера этих гирь.

На следующей строке выведите  $m$  — количество гирь, которые Джон должен положить на вторую чашу весов. На следующей строке выведите номера этих гирь.

Гири пронумерованы от 1 до  $n$  в порядке, в котором их веса заданы во вводе.

Если ответа нет, выведите единственное число «-1».

### Примеры

стандартный ввод	стандартный вывод
4 14 1 3 7 10	1 4 2 2 3
3 7 1 2 4	3 1 2 3 0

## Задача Н. Проверка теста

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Сегодня студенты писали тест по БЖД. В тесте было  $n$  вопросов, на каждый вопрос было предложено 4 варианта ответа «А», «В», «С» и «D», из которых нужно было выбрать один правильный ответ.

Преподаватель знает, что студенты часто списывают друг с друга, поэтому он хочет найти все пары похожих работ. Он считает две работы похожими, если в каждой из них больше половины правильных ответов совпадают с ответами в другой работе, и больше половины неправильных ответов совпадают с ответами в другой работе.

Помогите ему найти все пары похожих работ.

### Формат входных данных

Первая строка ввода содержит число  $n$  — количество вопросов в тесте ( $1 \leq n \leq 100$ ).

Вторая строка содержит правильные ответы, они задаются строкой длины  $n$ , состоящей из символов «А», «В», «С» и «D»,  $i$ -й символ строки равен правильному ответу на  $i$ -й вопрос теста.

Третья строка содержит число  $m$  — количество студентов, писавших тест ( $1 \leq m \leq 100$ ).

Далее следуют  $m$  строк, содержащие ответы студентов на тест в том же формате, что и строка с правильными ответами.

### Формат выходных данных

В первой строке выведите количество пар похожих работ. В следующих строках выведите все пары похожих работ в любом порядке. Работы пронумерованы натуральными числами от 1 до  $m$  в том порядке, в котором они даны во входных данных. Элементы внутри пары тоже можно выводить в любом порядке.

### Примеры

стандартный ввод	стандартный вывод
3 AAA 4 ABA ABA CBA CAA	1 1 2
6 ABCDAB 3 ABCCCC BCDCC ACCDCC	3 1 2 1 3 2 3

## Задача I. Магический фокус

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Артём пришёл в цирк, и сейчас ему предстоит принять участие в фокусе.

Артём загадывает перестановку — массив из  $n$  различных целых чисел от 1 до  $n$ . Пусть он загадал перестановку  $[a_1, a_2, \dots, a_n]$ . Для каждого  $i$  от 1 до  $n$  он рассматривает набор чисел  $\{a_i, a_{i+1}, a_{i+2}\}$  (будем считать, что  $a_{n+1} = a_1$ ,  $a_{n+2} = a_2$ ).

Числа в каждом наборе он перемешивает произвольным образом, а затем перемешивает и сами наборы. После чего сообщает получившиеся тройки Иллюзионисту.

Вы — Иллюзионист, угадайте перестановку, которую загадал Артём.

### Формат входных данных

Первая строка содержит целое число  $n$  ( $3 \leq n \leq 200\,000$ ) — количество элементов в перестановке.

Каждая из следующих  $n$  строк содержит три различных целых числа  $a_{i,1}, a_{i,2}, a_{i,3}$  ( $1 \leq a_{i,j} \leq n$ ) — наборы, которые Артём сообщил Иллюзионисту.

Гарантируется, что данные Артёма соответствуют хотя бы одной перестановке.

### Формат выходных данных

Выведите перестановку из  $n$  элементов, которую загадал Артём.

Если однозначно восстановить перестановку нельзя, выведите произвольную подходящую перестановку.

### Примеры

стандартный ввод	стандартный вывод
6 3 4 1 5 1 6 5 4 2 2 4 3 2 5 6 6 1 3	1 3 4 2 5 6
3 1 2 3 2 3 1 1 2 3	1 2 3

### Замечание

Во втором примере подойдёт произвольная перестановка чисел 1, 2 и 3.

## Задача J. Суперперестановки

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Однажды на уроке информатики Дима решил все возможные задачи про перестановки. Тогда он придумал задачу про *суперперестановку*.

*Суперперестановкой* порядка  $n$  называется такая последовательность чисел от 1 до  $n$ , что каждая перестановка чисел от 1 до  $n$  входит в неё как подотрезок.

Дима быстро придумал, что получать суперперестановки он будет с помощью следующего алгоритма:

- $s_1 = [1]$ .
- Исходно  $s_{n+1}$  приравняем к  $s_n$ .
- Будем рассматривать подотрезки  $s_n$  длины  $n$  слева направо, по увеличению левой границы.
- Если очередной подотрезок  $s_n[l, l+1, \dots, l+n-1]$  является перестановкой чисел от 1 до  $n$ , то есть, на этом отрезке встречаются все числа от 1 до  $n$  по одному разу, то вставим в  $s_{n+1}$  числа  $n+1, s_n[l], s_n[l+1], \dots, s_n[l+n-1]$  сразу после последнего элемента  $s_n[l+n-1]$  соответствующего подотрезка.

Посмотрим, как получаются суперперестановки порядка 1, 2, 3 и 4:

По определению  $s_1 = [1]$ .

Положим  $s_2 = [1]$ . Рассмотрим единственный отрезок длины 1 в  $s_1$ :  $[1]$  является перестановкой, вставляем после него  $[2, 1]$  в  $s_2$ , получаем  $s_2 = [1, 2, 1]$ .

Положим  $s_3 = [1, 2, 1]$ . Рассмотрим отрезки длины 2 в  $s_2$ .  $[1, 2]$  является перестановкой, вставляем после него  $[3, 1, 2]$ , получаем  $s_3 = [1, 2, 3, 1, 2, 1]$ .  $[2, 1]$  также является перестановкой, после его последнего элемента в  $s_3$  вставляем  $[3, 2, 1]$ , получаем  $s_3 = [1, 2, 3, 1, 2, 1, 3, 2, 1]$ .

Положим  $s_4 = [1, 2, 3, 1, 2, 1, 3, 2, 1]$ . Рассмотрим отрезки длины 3 в  $s_3$ .

- $[1, 2, 3]$  является перестановкой, вставляем после него  $[4, 1, 2, 3]$ , получаем  $s_4 = [1, 2, 3, 4, 1, 2, 3, 1, 2, 1, 3, 2, 1]$ .
- $[2, 3, 1]$  является перестановкой, вставляем после него  $[4, 2, 3, 1]$ , получаем  $s_4 = [1, 2, 3, 4, 1, 2, 3, 1, 4, 2, 3, 1, 2, 1, 3, 2, 1]$ .
- $[3, 1, 2]$  является перестановкой, вставляем после него  $[4, 3, 1, 2]$ , получаем  $s_4 = [1, 2, 3, 4, 1, 2, 3, 1, 4, 2, 3, 1, 2, 4, 3, 1, 2, 1, 3, 2, 1]$ .
- $[1, 2, 1]$  перестановкой не является, поэтому ничего не делаем и переходим к следующему отрезку.
- $[2, 1, 3]$  является перестановкой, вставляем после него  $[4, 2, 1, 3]$ , получаем  $s_4 = [1, 2, 3, 4, 1, 2, 3, 1, 4, 2, 3, 1, 2, 4, 3, 1, 2, 1, 3, 4, 2, 1, 3, 2, 1]$ .
- Аналогично делаем для двух оставшихся перестановок длины 3:  $[1, 3, 2]$  и  $[3, 2, 1]$ .

Дима заметил, что это довольно эффективный способ построения суперперестановки, поскольку каждая перестановка встречается ровно один раз. Чтобы убедиться, что Дима не ошибся, он хочет по заданной перестановке  $a_1, \dots, a_n$  найти её вхождение в суперперестановку  $s_n$ . Позиции в суперперестановке нумеруются, начиная с единицы.

Так как длина  $s_n$  достаточно большая, то необходимо вывести остаток от деления номера первого элемента последовательности  $s_n$ , с которого начинается вхождение в неё заданной перестановки, на число  $10^9 + 7$ .

### Формат входных данных

В первой строке содержится одно целое число  $n$  — количество элементов в перестановке ( $1 \leq n \leq 300\,000$ ).

Следующая строка входных данных содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$  — элементы перестановки ( $1 \leq a_i \leq n$ , все числа  $a_i$  различны).

### Формат выходных данных

Выведите единственное число — позицию вхождения перестановки  $a_1, a_2, \dots, a_n$  в суперперестановку порядка  $n$ . Ответ необходимо вывести по модулю  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
3 2 3 1	2
4 2 3 1 4	6
4 4 3 1 2	14

## Задача К. Подготовка тестов

Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Коля готовится к проведению олимпиады. До тура осталось не так много времени, а Коля все еще не сделал тесты к одной из задач. Поскольку времени на подготовку осталось мало, Коля принял взвешенное решение — сделать в этой задаче мультитест: несколько тестов, перечисленных один за другим во входных данных.

Каждый тест в этой задаче должен начинаться с числа  $m$ , а далее содержать  $m$  пар целых неотрицательных чисел  $a_i$  и  $b_i$ . Эти пары чисел задают ребра в неориентированном графе, вершины которого пронумерованы целыми числами от 0 до бесконечности. Граф во вводе не должен иметь петель и кратных рёбер, а также должен являться лесом — графом, не содержащим циклов.

Приведём примеры корректных и некорректных входных данных.

Входные данные	Описание
3 1 2 2 0 2 3	Один тест, содержащий дерево с 4 вершинами 0, 1, 2 и 3, рёбра (1, 2), (2, 0), (2, 3). Остальные вершины изолированы.
1 2 1 2 2 0 2 3	Два теста, первый содержит дерево с двумя вершинами 1 и 2, второй — дерево с вершинами 0, 2 и 3.
2 0 1 2 3	Один тест, два дерева: с вершинами 0 и 1 и с вершинами 2 и 3.
0	Один тест, граф не содержит рёбер.
0 0 0	Три теста, в каждом граф не содержит рёбер.
1 1 1	Некорректный тест, петля в вершине 1.
2 1 2 1 2	Некорректный тест, два кратных ребра.
2 1 2 2 1	Некорректный тест, два кратных ребра.
3 1 2 1 3 2 3	Некорректный тест, граф содержит цикл.
2 1 2 1 3 2 3	Некорректные входные данные, второй тест начинается с числа 2 и должен содержать описание двух рёбер, а содержит только одно число.

Упростив себе задачу, он тем не менее не решил вопрос, где взять столько тестов. Тогда он вспомнил, что недавно уже делал задачу, входные данные к которой представляли собой массив целых неотрицательных чисел. Он решил выбрать отрезки подряд идущих элементов этого массива в качестве тестов к новой задаче.

Теперь его интересует вопрос — сколько различных подотрезков массива являются корректными тестами? Два подотрезка считаются различными, если у них не совпадают границы.

### Формат входных данных

В первой строке дано одно целое число  $n$  — длина массива ( $1 \leq n \leq 300\,000$ ).

Во второй строке даны  $n$  целых чисел  $a_i$  — массив ( $0 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите одно целое число — количество различных подотрезков массива, которые являются корректными тестами для задачи.

### Примеры

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

## Замечание

В первом тесте корректными являются следующие подотрезки (выделены квадратными скобками):

- [2 1 3 4 1]
- 2 [1 3 4] 1

Во втором тесте корректными являются следующие подотрезки:

- [1 3 1] 2 2 0 2 3
- 1 [3 1 2 2 0 2 3]
- 1 3 1 [2 2 0 2 3]
- 1 3 1 2 2 [0] 2 3
- [1 3 1 2 2 0 2 3]