

Задача А. И вновь продолжается бой...

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

Перед обедом Виктор Александрович Матюхин раздал ЛКШатам карточки для игры в «Горца», и сразу выходить из столовой по одному и даже вдвоем стало опасно. Поэтому ЛКШата решили покидать столовую группами как минимум по три человека.

Все ЛКШата начинают кушать одновременно, но делают это с разной скоростью, то есть тому, кто съест обед быстрее всех, придется ждать как минимум двух товарищей. Аналогично тот, кто съест обед позже всех, должен будет попросить друзей его подождать.

ЛКШата не любят ждать и они хотят уходить из столовой такими группами, чтобы суммарное время ожидания было минимально.

Напишите программу, решающую эту задачу.

Формат входного файла

Первая строка входного файла содержит целое число n ($3 \leq n \leq 100\,000$) — количество ЛКШат. Во второй строке находятся n целых чисел t_1, t_2, \dots, t_n ($1 \leq t_i \leq 10^9$), где t_i — это время, необходимое i -ому ЛКШонку, чтобы пообедать.

Формат выходного файла

В первую строку выходного файла выведите через пробел два числа m и k — суммарное время ожидания и количество групп соответственно. Каждая из последующих k строк должна описывать одну группу выходящих школьников. Для каждой группы сначала выведите количество школьников и затем их номера, разделенные пробелом.

Пример

<code>battle.in</code>	<code>battle.out</code>
10	100 3
8 100 101 1 1 1 2 5 6 7	4 4 5 6 7 3 8 9 10 3 1 2 3

Задача В. Часы

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

Вася без ума от часов. На каждой руке он носит не меньше дюжины. Вася — коллекционер со стажем: в его коллекции есть и карманные часы, и настольные, и настенные, и напольные, и даже башенные. Недавно к нему попал интересный экземпляр — старинные часы с уникальным свойством показывать только четные минуты каждого часа.

На радость Васе, стоило ему завести часы, как они затикали, внутри что-то зашелестело, и минутная стрелка сместилась на деление. Однако через час наблюдения за часами у Васи появилось подозрение, что стрелки часов двигаются неправильно. Вася вооружился транспортиром, измерил углы отклонения стрелок от 00:00 в сторону их движения и записал свои измерения. Быстро обнаружив, что полученные углы выражаются целым числом градусов, он так задумался причинах этого эффекта, что забыл пометить, какой угол соответствует минутной, а какой — часовой стрелке.

Теперь Вася хочет проверить, существует ли такое время дня, при котором положение стрелок правильных часов

совпадает с положением стрелок Васиных часов в момент измерения.

Формат входного файла

Первая строка входного файла содержит целые числа deg_1 и deg_2 ($0 \leq deg_1, deg_2 \leq 359$) — пара измеренных Васей углов.

Формат выходного файла

В случае существования искомого времени выведите его в формате `hh:mm` в интервале от 00:00 до 11:59. В обратном случае выведите «No solution». (Кавычки — только для ясности).

Пример

<code>clock.in</code>	<code>clock.out</code>
0 30	01:00
0 31	No solution

Задача С. Криптография

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

После спецкурса «Введение в криптографию» Вася и Петя решили придумать свой собственный алгоритм шифрования. Вася отправил Пете зашифрованное сообщение, которое представляло собой правильную скобочную последовательность — результат шифрования новым алгоритмом. Но из-за помех в связи сообщение пришло по частям. Каждая часть представляет собой подстроку сообщения. Теперь Петя хочет понять, какое сообщение было отправлено. Для этого ему нужно так переставить полученные части, чтобы в результате получилась, если это возможно, правильная скобочная последовательность.

Формат входного файла

В первой строке входного файла находится число N ($1 \leq N \leq 100\,000$) — количество частей, на которые разбилось сообщение. В следующих N строках записаны сами части. В $(i + 1)$ -й строке находится i -я часть исходной скобочной последовательности A_i . Каждая из частей — это непустая строка, состоящая из открывающих и закрывающих круглых скобок. Гарантируется, что сумма длин всех частей не превосходит 10^5 .

Формат выходного файла

В выходной файл выведите «YES», если переставить полученные части возможно. Во второй строке следует вывести N чисел p_1, p_2, \dots, p_N , разделяя их одним пробелом. Последовательность должна представлять собой перестановку чисел от 1 до N . Строка $A_{p_1}A_{p_2}\dots A_{p_N}$ должна являться правильной скобочной последовательностью. Если искомым перестановкам несколько, выведите любую из них.

Если переставить части невозможно в единственной строке выведите слово «NO».

Пример

<code>code.in</code>	<code>code.out</code>
3 ((()	YES 2 3 1
2 ((()))	NO

Задача D. Мягкий отбой

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

22:30. Наступило время мягкого отбоя. Преподаватель хочет проверить, все ли дети находятся в своих домиках. Но так как он не успел запомнить, кто живет в каком домике, то он попросту проверяет, что в каждом домике столько детей, сколько должно быть. Известно, что в первом домике должно жить a_1 человек, во втором a_2 , и так далее. Всего N домиков, пронумерованных от 1 до N . Положение осложняется тем, что общее количество детей в момент отбоя вовсе не такое, какое должно быть (некоторые дети все еще не вернулись с клуба, а некоторые просто засиделись не в своем домике). А именно, в момент отбоя в домиках находится K человек. Прежде чем преподаватель начнет проверять домики, ЛКШата могут распределиться по домикам как хотят. В принципе, в домике помещается неограниченное количество человек.

Преподаватель последовательно проходит от первого до последнего домика и проверяет количество человек в каждом домике. Однако перед тем как преподаватель зайдет в какой-либо домик, дети могут отвлечь его (например, позвонив по телефону). Пока преподаватель занят, дети могут перебежать из любого домика в любой сколько угодно раз. Основная задача ЛКШат — перебежать таким образом, чтобы в тот момент, когда преподаватель входит в домик, там находилось нужное количество людей. Конечно же, лишний раз отвлекать преподавателя опасно, поэтому вам предстоит найти наименьшее количество раз, которое необходимо отвлечь преподавателя.

Формат входного файла

В первой строке входного файла содержатся числа N ($1 \leq N \leq 10^5$) и K ($1 \leq K \leq 10^9$). Во второй строке записано N чисел — a_i ($1 \leq a_i \leq 10^4$).

Формат выходного файла

Выведите единственное число — наименьшее количество раз, которое необходимо отвлечь преподавателя. Если же дети не могут перебежать таким образом, чтобы преподаватель обнаруживал в домике нужное количество человек, выведите «-1» (без кавычек).

Пример

<code>curfew.in</code>	<code>curfew.out</code>
3 1 1 1 1	2
3 3 1 1 1	0
1 10 1	-1

Задача E. Сравнение делителей

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

Недавно в группе А прошли алгоритмы на строках. Обычно сравнение строк осуществляется в лексикографическом порядке: две строки сравниваются сначала по первому символу, при равенстве — по второму символу, и так далее. Если же одна строка является префиксом другой, то она считается меньше. Например, строка «`product`» больше строки «`processor`», поскольку первые три символа совпадают, а четвертый символ «`d`» больше символа «`s`», а

строка «`process`» меньше строки «`processor`», поскольку она является ее префиксом.

А в группе А0 проходят теорию чисел. Натуральное число b является делителем натурального числа a , если a можно представить в виде $a = bc$ для некоторого натурального числа c . Число называется простым, если у него есть ровно два делителя: 1 и оно само.

Петя, который учится в В, заинтересовался: а как найти у заданного числа a минимальный простой делитель. При этом делители сравниваются в лексикографическом порядке как строки, содержащие их десятичные записи.

Формат входного файла

Входной файл содержит одно целое число x ($2 \leq x \leq 10^6$).

Формат выходного файла

Выведите в выходной файл одно число — минимальный в лексикографическом порядке простой делитель x .

Пример

<code>divisors.in</code>	<code>divisors.out</code>
273	13

Задача F. ПВШ в ЛКШ

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

Как известно, ЛКШ проходит в здании, напоминающем своей формой букву Ш. А именно, здание имеет N этажей. На каждом этаже расположен длинный коридор, из которого выходят K блоков. Конфигурация каждого этажа полностью одинакова, т.е. эти одинаковые блоки разных этажей расположены друг над другом, образуя K «ног». Каждый блок — это тамбур, из которого выходят три двери. Первая соединяет тамбур с этим большим коридором, вторая соединяет тамбур с собственно жилым отсеком блока, третья ведет на лестницу. Таким образом, в каждой «ноге» есть соединяющая все этажи лестница, соединенная дверями с тамбурами этажей. Итак, всего есть $3 \cdot N \cdot K$ дверей, по три в каждом блоке. В силу ряда причин некоторые двери могут быть заперты и отпереть их не представляется возможным.

За час до приезда школьников преподаватели обнаружили, что в одну из комнат поселили на одного школьника больше, чем там есть кроватей, а еще в одну — на одного меньше. Поэтому возникла срочная необходимость перенести кровать из одного блока в другой. Сделать это надо срочно, чтобы школьники ничего не заметили, но большую проблему по пути представляют собой двери. Через закрытую дверь пронести кровать, конечно, нельзя, но даже сквозь открытую дверь пронести кровать непросто. Поэтому им срочно понадобилось придумать, как перенести кровать, чтобы нужно было пройти через минимальное количество дверей.

Формат входного файла

На первой строке входного файла находятся четыре числа K , N и M , где K — количество «ног», N — количество этажей, M — количество закрытых дверей ($1 \leq N \leq 1000$, $1 \leq K \leq 1000$, $0 \leq M \leq 3000$).

На второй строке файла находятся четыре числа k_1 , n_1 , k_2 и n_2 — координаты блока, откуда нужно нести кровать, и блока, куда надо ее нести: номер «ноги» и номер этажа ($1 \leq n_1, n_2 \leq N$, $1 \leq k_1, k_2 \leq K$). Этажи нумеруются от земли с 1 до N , «ноги» — с одного конца здания от 1 до K .

Далее следуют M строк, на каждой из которых находятся три числа x_i , y_i и z_i , описывающие очередную закрытую дверь: x_i и y_i — координаты блока, где находится эта дверь

(номер «ноги» и этаж), а z_i равно 1 для двери между тамбуром и коридором, 2 для двери между тамбуром и лестницей, 3 для двери между тамбуром и жилым отсеком.

Формат выходного файла

В первую строку выходного файла выведите одно число L — минимальное количество дверей, через которые должен пролежать маршрут. Далее выведите L строк, описывающие эти двери в порядке прохода сквозь них в том же формате, как и во входном файле.

Если перенести кровать невозможно, выведите одно число: -1 .

Пример

lkshbfs.in	lkshbfs.out
2 2 1 2 1 1 2 2 1 2	6 2 1 3 2 1 1 1 1 1 1 1 2 1 2 2 1 2 3
2 1 1 1 1 2 1 1 1 3	-1

встают перед организаторами ЛКШ еще до ее начала? Например, о расселении школьников?

А это не так просто, как кажется. Школьников из одной группы необходимо селить в один домик. А домики совсем разные: в них разное количество комнат, да и сами комнаты разной вместимости. Кроме того, школьники из разных групп в одном домике не уживаются.

Все было бы не столь сложно, если бы школьники не дружили между собой. Но ведь каждый хочет жить со своими друзьями! Поэтому организаторы хотят расселить школьников так, чтобы каждый жил со своими друзьями в одной комнате.

Помогите сделать ваше пребывание в ЛКШ комфортнее.

Формат входного файла

В первой строке входного файла содержатся четыре целых числа: n , m , k и h ($1 \leq n \leq 250$; $0 \leq m \leq n^2$; $1 \leq k \leq 50$; $1 \leq h \leq 50$) — количество школьников, пар друзей среди школьников, учебных групп и домиков соответственно. Пусть школьники занумерованы числами от 1 до n , группы — числами от 1 до k , а домики — числами от 1 до h .

Вторая строка содержит n чисел от 1 до k , i -е число означает номер группы, в которой учится i -й школьник.

Следующие h строк содержат описания домиков по одному в строке. Домик описывается количеством комнат r ($1 \leq r \leq 3$), и r целыми числами — размерами комнат s_i ($1 \leq s_i \leq 50$).

Последние m строк содержат по два числа каждая: a_i и b_i ($1 \leq a_i, b_i \leq n$; $a_i \neq b_i$). Это означает, что школьники под номерами a_i и b_i дружат. Каждая пара друзей встречается в списке ровно один раз.

Формат выходного файла

В первую строку выходного файла выведите k чисел — номера домиков, в которые необходимо поселить учебные группы.

Во вторую строку выведите n чисел: i -е число означает номер комнаты, в которую необходимо поселить i -го школьника внутри домика, соответствующего его группе. Комнаты нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Если расселить школьников не получится, выведите единственное число « -1 » (без кавычек). Если решений несколько, выведите любое.

Пример

settling.in
16 9 3 4 1 1 1 1 1 1 1 2 2 2 2 2 3 3 3 3 1 4 2 3 5 3 1 1 2 2 2 3 1 2 1 3 4 5 5 6 6 7 16 15 15 14 15 13 16 13
settling.out
2 4 1 1 1 1 2 2 2 2 1 1 2 2 2 1 1 1 1

Задача G. Слоники

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

Перестановкой называется последовательность из n различных чисел от 1 до n . Например, $\langle 3, 1, 4, 2, 5 \rangle$ представляет собой перестановку. Перестановки можно упорядочить в лексикографическом порядке: сначала по первому числу, при равном первом — по второму, затем по третьему, и т. д. Например, перестановки размера 3 оказываются упорядочены в следующем порядке: $\langle 1, 2, 3 \rangle$, $\langle 1, 3, 2 \rangle$, $\langle 2, 1, 3 \rangle$, $\langle 2, 3, 1 \rangle$, $\langle 3, 1, 2 \rangle$, $\langle 3, 2, 1 \rangle$. Пронумеруем их в этом порядке от 1 до $n!$. Например, четвертая перестановка размера 3 в этом порядке — $\langle 2, 3, 1 \rangle$.

Пусть задана последовательность целых чисел a_1, a_2, \dots, a_n . Тогда k -й порядковой статистикой в этой последовательности называется число, которое после сортировки оказалось бы в этой последовательности на k -й позиции. Например, 3-я порядковая статистика в последовательности 1, 1, 7, 10, 2, 8 равна 2.

По заданным числам k , n и p найдите k -ю порядковую статистику в перестановке с номером p среди всех перестановок размера n .

Формат входного файла

Входной файл содержит три числа: k , n и p ($1 \leq k \leq n \leq 100$; $1 \leq p \leq \min\{10^9, n!\}$).

Формат выходного файла

Выведите в выходной файл одно число — искомую порядковую статистику.

Пример

perm.in	perm.out
1 2 1	1

Задача H. Расселение

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

Задумывались ли вы когда-нибудь о задачах, которые

Задача I. Две сортирующие перестановки

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

Перестановкой порядка n называется последовательность из попарно различных целых положительных чисел p_1, p_2, \dots, p_n , где каждое $1 \leq p_i \leq n$. Если перестановку применить к массиву a , состоящему из n чисел, элементы массива переставятся так, что i -й элемент массива a перейдет на p_i -е место. Перестановка является сортирующей для массива, если при ее применении она переставляет элементы так, что массив становится отсортированным по возрастанию.

Например, если применить перестановку $\langle 3, 1, 2, 4 \rangle$ к массиву $(5, 1, 1, 10)$, то получится массив $(1, 1, 5, 10)$. Заметим, что он является отсортированным, поэтому эта перестановка является сортирующей для этого массива.

В этой задаче будут даны две перестановки порядка n . Вам необходимо придумать такой массив из n целых положительных чисел, что обе эти перестановки будут сортирующими для этого массива, а количество различных элементов в этом массиве будет максимально.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \leq n \leq 100\,000$) — порядок перестановок. В следующих двух строках описаны две перестановки — по n различных чисел от 1 до n .

Формат выходного файла

В выходной файл выведите ответ — искомый массив из n целых положительных чисел, не превосходящих 10^9 .

Пример

sortperm.in	sortperm.out
4	1 1 1 10
2 1 3 4	
3 2 1 4	

Задача J. Вечорка

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

Вечор, ты помнишь, был кефир?
В десятом домике шёл пир...

Вымысел. Все совпадения случайны.

Давайте представим себе идеальную ЛКШ. Да, это сложно, но мы постараемся. Как ни парадоксально, в идеальной ЛКШ сразу появляется множество проблем!

Взять хотя бы такую банальность, как вечерний кефирчик. Естественно, в идеальной ЛКШ кефирчик выдают так, что всем хватает. И даже остается немного лишнего, чем незамедлительно пользуются преподаватели. И тут возникает вопрос: а что бы такое сделать, чтобы оставалось побольше кефирчика? Преподаватели справились с этой задачей — надо правильно распределять школьников по параллелям!

Как известно, в идеальной ЛКШ каждая параллель живет в своем собственном коттедже, поэтому вечерку выдают сразу на всю параллель. Одного пакета кефирчика хватает на двух с половиной человек. При этом работники столовой тоже любят кефирчик, и потому выдают каждой параллели минимальное возможное количество пакетов.

Так как преподаватели в идеальной ЛКШ очень дружные (впрочем, как и в любой ЛКШ), то они хотят максимизировать суммарный лишний кефирчик. Чтобы в идеальной ЛКШ было хотя бы одной проблемой меньше, необходимо заранее распределить школьников по параллелям.

Формат входного файла

Входной файл содержит два числа: n и k ($1 \leq k \leq n \leq 100\,000$) — количество школьников и параллелей соответственно.

Формат выходного файла

В выходной файл выведите k чисел — количество школьников в каждой параллели. Естественно, каждый школьник должен попасть ровно в одну параллель, и в каждой параллели должно быть не менее одного школьника. Если решений несколько, выведите любое.

Пример

supper.in	supper.out
10 2	2 8
6 3	3 2 1

Задача K. Футболки

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

На зарядку у столовой пришло n ЛКШат, они построились в ряд. Разумеется, ребята ходят в разноцветных ЛКШатских футболках разных лет. Владимир Андреевич заметил, что можно попросить некоторых ребят присесть, и тогда для ребят, которые останутся стоять, будет выполнено следующее: последовательность цветов их футболок при перечислении слева направо будет такой же как и последовательность при перечислении справа налево, то есть будет *палиндромом*.

Например, если на зарядку пришли Маша в зеленой футболке, Паша в желтой, Сережа в красной и Ваня в зеленой, то можно попросить присесть Пашу, тогда последовательность цветов будет «зеленый, красный, зеленый» как слева направо, так и справа направо. Аналогично можно попросить присесть Сережу (последовательность будет «зеленый, желтый, зеленый»), Пашу и Сережу одновременно, или одного из троих ребят. Таким образом, всего есть 7 способов добиться того, чтобы последовательность цветов была палиндромом.

Помогите Владимиру Андреевичу найти количество способов попросить некоторых ЛКШат присесть, чтобы последовательность цветов футболок оставшихся стоять была палиндромом. Поскольку это число может быть очень большим, выведите его по модулю 10^9 .

Формат входного файла

Первая строка входного файла содержит число n — количество ЛКШат, пришедших на зарядку ($1 \leq n \leq 2000$). Вторая строка содержит n целых чисел, каждое из которых задает цвет футболки ЛКШонка и изменяется в пределах от 1 до 10^9 . Разные цвета задаются разными числами, а одинаковые — одинаковыми.

Формат выходного файла

Выведите в выходной файл одно число — искомое количество способов по модулю 10^9 .

Пример

tshirts.in	tshirts.out
4	7
1 2 3 1	

Задача L. Не-не-не-нечестная игра

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

Дэвид Блейн и Вася сидят на уроке математики и скучают. Через некоторое время Дэвид Блейн предлагает поиграть в следующую игру:

- он записывает на листочке некоторое большое целое число и ходит первым;
- игроки ходят по очереди;
- на каждом ходу игрок обязан разделить текущее число на его простой делитель p . На этом же ходу игрок может умножить (а может и нет) результат деления на простое число q ($1 < q < p$);
- проигрывает тот, кто не может сделать ход.

Мягко говоря, Вася не очень доверяет Дэвиду Блейну и боится, что тот выписывает только проигрышные для Васи начальные числа. Помогите ему определить, так это или нет.

Формат входного файла

Первая строка входного файла содержит единственное число n ($1 \leq n \leq 10^{12}$).

Формат выходного файла

В выходной файл выведите `Vasya`, если у Васи есть выигрышная стратегия, независимо от ходов Дэвида Блейна. Иначе выведите `David`.

Пример

<code>unfair.in</code>	<code>unfair.out</code>
16	Vasya

Задача M. Завалинка

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

КОЖЫПОГ — ГОПЫЖОК наоборот.

Из завалинки ЛКШ2008.Кострома.

Завалинка — это командная интеллектуальная игра, целью которой является обман других команд.

В этой игре есть набор малоизвестных слов русского языка, к которым команды должны придумать и записать в бланки неправильные определения, очень похожие на правильные определения из словаря. Правильное определение писать нельзя, за это команда получает -10 баллов.

После того, как все команды написали определения, игра переходит во вторую фазу. Эта фаза состоит из нескольких туров, по одному на слово. В каждом туре команды слушают определения друг друга и голосуют за правильное, по их мнению, определение. При этом к определениям команд добавляется правильное определение слова и неправильное определение, придуманное жюри. Все эти определения читаются в случайном порядке. Каждая команда пишет номер определения, за которое она голосует. Определения нумеруются с единицы в порядке прочтения.

Если команда угадала правильное определение, она получает 2 балла. Если же команда указала определение другой команды, то команда-автор определения получает за

это 1 балл. Голосовать за собственное определение запрещено, за это команда получает -10 баллов.

Баллы за все туры суммируются. Команда, набравшая наибольшее число баллов, становится победителем.

Вам предлагается по информации о процессе проведения завалинки подсчитать результат каждой команды.

Формат входного файла

Первая строка входного файла содержит два целых числа: t и w ($1 \leq t, w \leq 50$) — количество команд и слов соответственно. Каждая из последующих t строк содержит название очередной команды. В следующих w строках содержатся слова, по одному в строке. Далее во входном файле содержатся w блоков, каждый из которых задает результаты одного тура. Перед каждым блоком содержится пустая строка.

Каждый блок описывается играемым словом, порядком прочтения ответов команд и вариантами ответов, за которые проголосовали команды.

В первой строке блока содержится играемое слово.

В последующих $t + 2$ строках содержатся названия команд, описывающие порядок прочтения вариантов ответов, по одному в строке. При этом строка «\$answer\$» означает правильный ответ, а строка «\$jury\$» — вариант жюри (неправильное определение). Каждая команда встречается в этом списке ровно один раз.

Последние t строк блока содержат номера ответов, за которые голосовали команды. Каждая строка содержит название команды s и число i ($1 \leq i \leq t + 2$), разделенные пробелом. Такая строка означает, что команда s выбрала вариант i . Каждая команда встречается в этом списке ровно один раз.

Входной файл не содержит лишних пробелов. Все слова и названия команд состоят из строчных латинских и русских букв, их длина не превышает 20 символов. Предполагается, что ни одна команда не написала правильное определение ни на какое слово.

Формат выходного файла

В выходной файл выведите $t + 1$ строку. В каждую из первых t строк выведите результат одной команды в формате « $s r$ », где s — название команды, а r — количество очков, набранных этой командой. Выводите команды в том порядке, в каком они первый раз перечислены во входном файле.

В последней строке выведите «\$jury\$ r », где r — количество голосований команд за варианты жюри.

Пример

zavalinka.in	zavalinka.out
4 3	отбой -6
отбой	солнышко 4
солнышко	анихомикос 4
анихомикос	сепаратор 0
сепаратор	\$jury\$ 2
нахзац	
похерц	
афакия	
афакия	
отбой	
\$jury\$	
сепаратор	
анихомикос	
солнышко	
\$answer\$	
отбой 6	
солнышко 1	
анихомикос 2	
сепаратор 5	
нахзац	
\$answer\$	
сепаратор	
\$jury\$	
анихомикос	
отбой	
солнышко	
солнышко 1	
отбой 5	
анихомикос 5	
сепаратор 3	
похерц	
анихомикос	
солнышко	
\$jury\$	
\$answer\$	
отбой	
сепаратор	
анихомикос 4	
солнышко 1	
отбой 1	
сепаратор 2	