

## Задача А. Самое частое слово

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

Дан текст. Выведите слово, которое в этом тексте встречается чаще всего.

Если таких слов несколько, выведите то, которое меньше в лексикографическом порядке.

### Формат входных данных

Вводится текст.

### Формат выходных данных

Выведите ответ на задачу.

### Примеры

стандартный ввод	стандартный вывод
apple orange banana banana orange	banana
oh you touch my tralala mmm my ding dingdingg	ding

## Задача В. Предметный указатель

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

Профессор написал научную книгу и составил для неё предметный указатель. Это список ключевых слов, для каждого из которых указана страница, на которой это слово встречается. Теперь профессор хочет для каждой страницы выписать в алфавитном порядке все ключевые слова, которые на эту страницу попали (если такие вообще есть). Помогите профессору решить эту задачу.

### Формат входных данных

Сначала задано натуральное число  $n$ , не превосходящее 1000 — количество слов, которое требуется обработать. Далее идут  $n$  строк. В каждой строке сначала записано ключевое слово. Затем идёт натуральное число, также не превосходящее 1000, — номер страницы. Ключевые слова состоят из латинских букв, не бывают пустым и по длине не превосходят 16 символов. Слова в списке, конечно, могут повторяться.

### Формат выходных данных

Выпишите в порядке возрастания все страницы, на которых присутствуют ключевые слова. После каждого номера страницы через пробел выпишите в алфавитном порядке сами эти слова. Если на какой-то странице слово встретилось несколько раз, то повторять его не нужно. Завершающего пробела в конце строк быть не должно.

### Пример

стандартный ввод	стандартный вывод
5 derivative 10 function 2 function 10 function 10 limit 7	2 function 7 limit 10 derivative function

## Задача С. Симметрическая разность множеств

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

Реализуйте бинарную операцию над множествами — симметрическую разность. Ее результатом являются те элементы двух множеств, которые принадлежат ровно одному из множеств (но не принадлежат их пересечению).

### Формат входных данных

В первой строке записано натуральное число — количество элементов первого множества.

Во второй строке через пробел перечислены эти элементы (натуральные числа).

В третьей строке указано количество элементов второго множества (натуральное).

В четвертой строке перечислены через пробел элементы второго множества (натуральные числа).

### Формат выходных данных

Результат требуется вывести в том же формате: сначала количество элементов, полученных в результате операции, затем (если получено ненулевое количество элементов) вывести сами элементы в порядке возрастания.

Множества во входных данных могут быть неупорядочены и содержать равные элементы. Множество в выводе должно быть упорядочено и все элементы его должны быть различными.

### Пример

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

## Задача D. Минимум на отрезке

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Рассмотрим последовательность целых чисел длины  $N$ . По ней с шагом 1 движется «окно» длины  $K$ , то есть сначала в «окне» видно первые  $K$  чисел, на следующем шаге в «окне» уже будут находиться  $K$  чисел, начиная со второго, и так далее до конца последовательности. Требуется для каждого положения «окна» определить минимум в нём.

### Формат входных данных

В первой строке входных данных содержатся два числа  $N$  и  $K$  ( $1 \leq N \leq 150000$ ,  $1 \leq K \leq 100000$ ,  $K \leq N$ ) – длины последовательности и «окна», соответственно. На следующей строке находятся  $N$  чисел – сама последовательность. Числа последовательности не превосходят по модулю  $10^5$ .

### Формат выходных данных

Выходные данные должны содержать  $N - K + 1$  строк – минимумы для каждого положения «окна».

### Пример

стандартный ввод	стандартный вывод
7 3 1 3 2 4 5 3 1	1 2 2 3 1

## Задача Е. Англо-латинский словарь

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

Однажды, разбирая старые книги на чердаке, школьник Вася нашёл англо-латинский словарь. Английский он к тому времени знал в совершенстве, и его мечтой было изучить латынь. Поэтому попавшийся словарь был как раз кстати.

К сожалению, для полноценного изучения языка недостаточно только одного словаря: кроме англо-латинского необходим латинско-английский. За неимением лучшего он решил сделать второй словарь из первого.

Как известно, словарь состоит из переводимых слов, к каждому из которых приводится несколько слов-переводов. Для каждого латинского слова, встречающегося где-либо в словаре, Вася предлагает найти все его переводы (то есть все английские слова, для которых наше латинское встречалось в его списке переводов), и считать их и только их переводами этого латинского слова.

Помогите Васе выполнить работу по созданию латинско-английского словаря из англо-латинского.

### Формат входных данных

В первой строке содержится единственное целое число  $N$  — количество английских слов в словаре. Далее следует  $N$  описаний. Каждое описание содержится в отдельной строке, в которой записано сначала английское слово, затем отведённый пробелами дефис (символ номер 45), затем разделённые запятыми с пробелами переводы этого английского слова на латинский. Переводы отсортированы в лексикографическом порядке. Порядок следования английских слов в словаре также лексикографический.

Все слова состоят только из маленьких латинских букв, длина каждого слова не превосходит 15 символов. Общее количество слов на входе не превышает 100 000.

### Формат выходных данных

Выведите соответствующий данному латинско-английский словарь, в точности соблюдая формат входных данных. В частности, первым должен идти перевод лексикографически минимального латинского слова, далее — второго в этом порядке и т.д. Внутри перевода английские слова должны быть также отсортированы лексикографически.

### Пример

стандартный ввод	стандартный вывод
3 apple - malum, pomum, popula fruit - bacca, bacca, popum punishment - malum, multa	7 bacca - fruit bacca - fruit malum - apple, punishment multa - punishment pomum - apple popula - apple popum - fruit

## Задача F. Приготовление милкшейков

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

Недавно вы решили открыть небольшой ресторанчик на пляже Липецкого моря — излюбленном месте для туристов со всего мира. Так как в курортный сезон на улице очень жарко, вы собираетесь добавить в меню большое количество прохладительных напитков и, конечно же, авторских милкшейков.

Для приготовления милкшейков по различным рецептам требуется смешать множество ингредиентов в различных пропорциях. На заднем дворе ресторанчика есть склад, на котором хранятся ингредиенты. Для учета ингредиентов на складе и приема заказов от посетителей вы решили разработать программу, которая должна отвечать на запросы двух типов:

- Принять заказ на приготовление милкшейка под названием *name*, для которого требуется  $k_1$  условных единиц ингредиента  $s_1$ ,  $k_2$  условных единиц ингредиента  $s_2$ , и так далее,  $k_n$  условных единиц ингредиента  $s_n$ . В случае, если на складе недостаточно каких-либо ингредиентов для приготовления милкшейка, вам придется огорчить посетителя, после чего он наверняка уйдет из вашего заведения. В случае, если ингредиентов для приготовления милкшейка достаточно, их тут же забирают со склада и начинают приготовление.
- Из магазина на склад привозят  $k$  условных единиц ингредиента  $s$ , которые в дальнейшем можно использовать для приготовления милкшейков.

Напишите программу, которая сможет быстро обработать все запросы. До поступления первого запроса на складе нет ингредиентов.

### Формат входных данных

Первая строка содержит одно целое число  $q$  ( $1 \leq q \leq 10^5$ ) — количество запросов, которые нужно обработать.

Следующие строки содержат описание запросов в описанном далее формате. Описание каждого запроса начинается с целого числа  $t$  ( $1 \leq t \leq 2$ ), обозначающем тип запроса.

В случае, если  $t = 1$ , далее в этой же строке следует строка *name* ( $1 \leq |name| \leq 25$ ) — название милкшейка, который был заказан очередным посетителем.

Следующая строка содержит целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество различных ингредиентов, необходимых для приготовления милкшейка.

Каждая из следующих  $n$  строк содержит целое число  $k_i$  и строку  $s_i$  ( $1 \leq k_i \leq 10^9$ ,  $1 \leq |s_i| \leq 25$ ) — количество условных единиц и название очередного ингредиента, необходимого для приготовления милкшейка. Гарантируется, что все  $s_i$  в рамках одного запроса попарно различны.

В случае, если  $t = 2$ , далее в этой же строке следует целое число  $k$  и строка  $s$  ( $1 \leq k \leq 10^9$ ,  $1 \leq |s| \leq 25$ ) — количество условных единиц и название ингредиента, которые были доставлены на склад.

Все названия милкшейков и ингредиентов состоят из строчных и заглавных латинских символов, а также символов подчеркивания («\_»).

Гарантируется, что суммарное количество различных ингредиентов, необходимое для приготовления всех милкшейков, не превосходит  $10^5$ . Иными словами,  $\sum_{i=1}^q n_i \leq 10^5$ .

### Формат выходных данных

Выведите  $q$  строк, в каждой из которых запишите ответ на  $i$ -й запрос.

В качестве ответа на запрос первого типа выведите сообщение «Milkshake *name* is ready» (без кавычек), где *name* — название милкшейка, в случае, если ингредиентов на складе хватит для его приготовления. В противном случае выведите сообщение «:(» (без кавычек).

В качестве ответа на запрос второго типа выведите сообщение « $s: k$ » (без кавычек), где  $s$  — название ингредиента, который был доставлен на склад, а  $k$  — количество условных единиц данного ингредиента на складе с учетом уже имевшихся до пополнения.

Все сообщения должны строго соответствовать формату, описанному выше. Для лучшего понимания формата ввода и вывода обратите внимание на пример.

## Система оценки

Помимо тестов из условия, данная задача содержит 25 тестов, каждый из которых оценивается независимо. За каждый успешно пройденный тест вы получите 4 балла.

## Пример

стандартный ввод	стандартный вывод
9	:(
1 Milk	Milk: 1000
1	Milkshake Milk is ready
100 Milk	Banana_syrup: 20
2 1000 Milk	Bubble_gum_syrup: 30
1 Milk	Ice_cream: 100
1	Milk: 801
200 Milk	Milkshake Banana_gum is ready
2 20 Banana_syrup	:(
2 30 Bubble_gum_syrup	
2 100 Ice_cream	
2 1 Milk	
1 Banana_gum	
4	
20 Banana_syrup	
20 Bubble_gum_syrup	
80 Ice_cream	
120 Milk	
1 Just_a_syrup	
1	
1 Banana_syrup	

## Замечание

Рассмотрим выполнение всех запросов в примере из условия:

- Заказан милкшейк с названием Milk, состоящий из 100 условных единиц ингредиента Milk. Так как склад изначально пуст, данный заказ не сможет быть выполнен.
- На склад поступило 1 000 условных единиц ингредиента Milk.
- Заказан милкшейк с названием Milk, на этот раз состоящий из 200 условных единиц ингредиента Milk. После выполнения заказа на складе останется  $1\,000 - 200 = 800$  условных единиц данного ингредиента.
- На склад поступило 20 условных единиц ингредиента Banana\_syrup.
- На склад поступило 30 условных единиц ингредиента Bubble\_gum\_syrup.
- На склад поступило 100 условных единиц ингредиента Ice\_cream.
- На склад поступила 1 условная единица ингредиента Milk — теперь этого ингредиента на складе 801 условная единица.

8. Заказан милкшейк с названием `Banana_gum`, для приготовления которого нужно 20 единиц `Banana_syrup`, 20 единиц `Bubble_gum_syrup`, 80 единиц `Ice_cream` и 120 единиц `Milk`. Данный заказ может быть выполнен.
9. Заказан милкшейк с названием `Just_a_syrup`, для приготовления которого на складе не хватает единственного ингредиента — `Banana_syrup`.



## Задача G. Вычислительная ихтиология

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

Игорь работает младшим лаборантом в НИИ ихтиологии. Ему вверены  $n$  аквариумов, стоящих в ряд, в каждом из которых живет колония рыбок гуппи. Про каждую колонию заранее известна ее численность.

В лабораторных условиях НИИ ихтиологии колония рыбок гуппи растет по следующему правилу: достигнув популяции в  $f$  рыбок, колония живет в течение  $\max(1000 - f, 1)$  секунд, после чего на свет появляется новая рыбка. От начального момента времени до рождения первой рыбки колония размера  $f$  также ждет  $\max(1000 - f, 1)$  секунд.

Например, колония с начальным размером 996 будет размножаться следующим образом:

время	число рыб	время до очередной рыбки
0	996	4
4	997	3
7	998	2
9	999	1
10	1000	1
11	1001	1
...	...	...

Появление на свет каждой новой рыбки Игорь должен фиксировать в специальном журнале. Будем считать, что запись он делает мгновенно, но при этом он должен в момент рождения новой рыбки находиться рядом с аквариумом, в котором это произошло.

На перемещение от одного аквариума к соседнему у Игоря уходит одна секунда. В начальный момент времени Игорь стоит около первого аквариума.

Вычислите, в течение какого наибольшего периода времени Игорь сможет добросовестно выполнять свою работу.

### Формат входных данных

В первой строке входного файла содержится целое число  $n$  ( $2 \leq n \leq 50$ ) — количество аквариумов с рыбками гуппи в НИИ ихтиологии. Каждая из следующих  $n$  строк содержит одно целое число  $a_i$  ( $1 \leq a_i \leq 2007$ ) — численность  $i$ -й колонии.

### Формат выходных данных

В выходной файл выведите момент времени, когда родится первая рыбка гуппи, запись о рождении которой Игорь сделать не сможет.

### Пример

стандартный ввод	стандартный вывод
3 996 1 994	7

### Замечание

В приведенном примере Игорь сначала ждет у первого аквариума появления рыбки на 4-й секунде. После этого он бежит к третьему аквариуму (на это у него уходит 2 секунды) и как раз успевает к рождению рыбки на 6-й секунде. Однако вернуться к первому аквариуму, где следующая рыбка родится на 7-й секунде, он уже не успевает.

## Задача Н. Ладьи-защитники

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

У вас есть квадратная шахматная доска размера  $n \times n$ . Строки пронумерованы сверху вниз числами от 1 до  $n$ , а столбцы — слева направо числами от 1 до  $n$ . Таким образом, каждая клетка доски задается парой целых чисел  $(x, y)$  ( $1 \leq x, y \leq n$ ), где  $x$  — номер строки, а  $y$  — номер столбца.

От вас требуется выполнить  $q$  запросов трех типов:

- Поставить новую ладью в клетку  $(x, y)$ .
- Убрать ладью из клетки  $(x, y)$ , куда ранее была поставлена ладья.
- Проверить, верно ли, что каждая клетка *подпрямоугольника*  $(x_1, y_1) - (x_2, y_2)$  доски атакована хотя бы одной ладьей.

*Подпрямоугольником* называется множество клеток доски  $(x, y)$ , для которых верно, что  $x_1 \leq x \leq x_2$  и  $y_1 \leq y \leq y_2$ .

Напомним, что клетка  $(a, b)$  атакована ладьей, находящейся в клетке  $(c, d)$ , если  $a = c$  или  $b = d$ . В частности, клетка, в которой находится ладья, атакована этой ладьей.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $q$  ( $1 \leq n \leq 10^5$ ,  $1 \leq q \leq 2 \cdot 10^5$ ) — размер доски и также количество запросов, соответственно.

Каждая из следующих  $q$  строк содержит в себе описание очередного запроса. Описание запроса начинается с целого числа  $t$  ( $t \in \{1, 2, 3\}$ ), которое обозначает тип запроса:

- Если  $t = 1$ , далее следуют два целых числа  $x$  и  $y$  ( $1 \leq x, y \leq n$ ) — координаты клетки, на которую нужно поставить новую ладью. Гарантируется, что в момент данного запроса в клетке  $(x, y)$  не стоит ладья.
- Если  $t = 2$ , далее следуют два целых числа  $x$  и  $y$  ( $1 \leq x, y \leq n$ ) — координаты клетки, с которой нужно убрать ладью. Гарантируется, что в момент данного запроса в клетке  $(x, y)$  стоит ладья.
- Если  $t = 3$ , далее следуют четыре целых числа  $x_1, y_1, x_2$  и  $y_2$  ( $1 \leq x_1 \leq x_2 \leq n$ ,  $1 \leq y_1 \leq y_2 \leq n$ ) — подпрямоугольник, для которого нужно проверить, верно ли, что каждая его клетка атакована хотя бы одной ладьей.

Гарантируется, что среди  $q$  запросов есть хотя бы один запрос третьего типа.

### Формат выходных данных

Для каждого запроса третьего типа в отдельной строке выведите «Yes» (без кавычек), если каждая клетка данного подпрямоугольника атакована хотя бы одной ладьей.

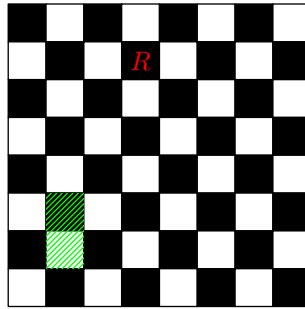
В противном случае выведите «No» (без кавычек).

## Пример

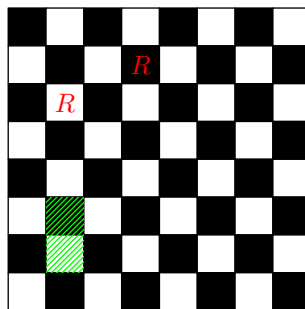
стандартный ввод	стандартный вывод
8 10	No
1 2 4	Yes
3 6 2 7 2	Yes
1 3 2	No
3 6 2 7 2	Yes
1 4 3	
3 2 6 4 8	
2 4 3	
3 2 6 4 8	
1 4 8	
3 2 6 4 8	

## Замечание

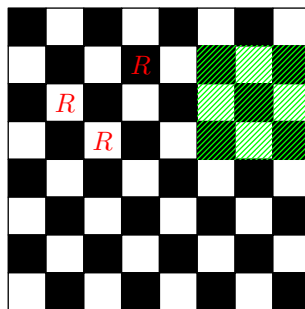
Рассмотрим пример. После первых двух запросов доска будет выглядеть следующим образом (буквой *R* обозначены клетки, в которых находится ладья, а зеленым выделен прямоугольник запроса третьего типа):



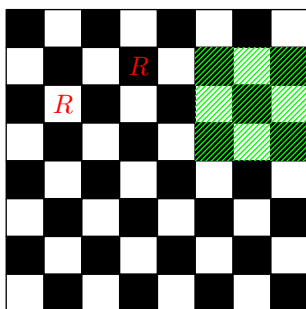
Доска после выполнения третьего и четвертого запросов:



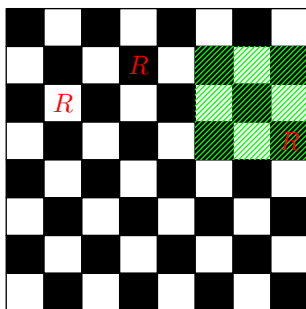
Доска после выполнения пятого и шестого запросов:



Доска после выполнения седьмого и восьмого запросов:



Доска после выполнения последних двух запросов:



## Задача I. Машинки

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

Петя, которому три года, очень любит играть с машинками. Всего у Пети  $N$  различных машинок, которые хранятся на полке шкафа так высоко, что он сам не может до них дотянуться. Одновременно на полу комнаты может находиться не более  $K$  машинок. Петя играет с одной из машинок на полу и если он хочет поиграть с другой машинкой, которая также находится на полу, то дотягивается до нее сам. Если же машинка находится на полке, то он обращается за помощью к маме. Мама может достать для Пети машинку с полки и одновременно с этим поставить на полку любую машинку с пола. Мама очень хорошо знает своего ребенка и может предугадать последовательность, в которой Петя захочет играть с машинками. При этом, чтобы не мешать Петиней игре, она хочет совершить как можно меньше операций по подъему машинки с пола, каждый раз правильно выбирая машинку, которую следует убрать на полку. Ваша задача состоит в том, чтобы определить минимальное количество операций. Перед тем, как Петя начал играть, все машинки стоят на полке.

### Формат входных данных

В первой строке содержатся три числа  $N$ ,  $K$  и  $P$  ( $1 \leq K, N \leq 100000$ ,  $1 \leq P \leq 500000$ ). В следующих  $P$  строках записаны номера машинок в том порядке, в котором Петя захочет играть с ними.

### Формат выходных данных

Выведите единственное число: минимальное количество операций, которое надо совершить Петиней маме.

### Пример

стандартный ввод	стандартный вывод
3 2 7 1 2 3 1 3 1 2	4

### Замечание

В этой задаче можно использовать STL.

Пояснения к примеру:

Операция 1: снять машинку 1

Операция 2: снять машинку 2

Операция 3: поднять машинку 2 и снять машинку 3

Операция 4: поднять машинку 3 или 1 и снять машинку 2

## Задача J. Трамвай

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

С окраины в центр города каждое утро по одному маршруту едут в трамвае  $N$  человек. За долгое время поездок они достаточно хорошо узнали друг друга. Чтобы никому не было обидно, они захотели решить, кто из них и между какими остановками маршрута должен сидеть, а кто должен стоять. Все остановки пронумерованы от 1 до  $P$ .

Один из пассажиров оказался знатоком теории математического моделирования. Он предложил рассмотреть значение суммарного удовлетворения пассажиров. Для каждого  $i$ -го пассажира он оценил две величины —  $a_i$  и  $b_i$ . Если в течение одного переезда между остановками пассажир сидит, то к суммарному удовлетворению прибавляется  $a_i$ , если же он стоит, то прибавляется  $b_i$ .

Всего в трамвае  $M$  сидячих мест. Вставать и садиться пассажиры могут мгновенно на любой остановке. Кроме того, некоторые пассажиры предпочитают ехать стоя, даже если в трамвае есть свободные места (для них  $a_i < b_i$ ).

Требуется написать программу, которая вычисляет значение максимально достижимого суммарного удовлетворения, если для каждого  $i$ -го пассажира известны величины  $a_i$  и  $b_i$ , а также номера остановок, на которых он садится и выходит из трамвая.

### Формат входных данных

Первая строка входного файла содержит разделенные пробелом три целых числа  $N$ ,  $M$  и  $P$  — число пассажиров, число сидячих мест и число остановок на маршруте соответственно ( $1 \leq N, M, P \leq 100000$ ;  $2 \leq P$ ).

Каждая из следующих  $N$  строк содержит информацию об очередном пассажире в виде четырёх целых чисел  $a_i, b_i, c_i, d_i$ , где первые два числа определяют вклад в параметр счастья, третье — номер остановки, на которой пассажир садится в трамвай, и последнее — номер остановки, на которой он выходит из трамвая ( $-10^6 \leq a_i, b_i \leq 10^6$ ;  $1 \leq c_i < d_i \leq P$ ).

### Формат выходных данных

В выходной файл необходимо вывести одно целое число — максимальное суммарное удовлетворение, которого могут добиться пассажиры.

### Пример

стандартный ввод	стандартный вывод
4 2 4 10 -10 2 3 -1 -3 1 4 6 -6 1 3 7 4 2 4	28

### Замечание

Максимальное суммарное довольство достигается следующим образом:

На первой остановке входят и садятся второй и третий пассажиры;

На второй остановке входят первый и четвёртый пассажиры, второй уступает место первому;

На третьей остановке встают и выходят первый и третий пассажиры, второй и четвёртый садятся на их места;

На четвёртой остановке выходят второй и четвёртый пассажиры.