

### ***Информация о результатах оценивания решений во время тура для всех задач***

В течение тура можно не более 10 раз по каждой задаче запросить информацию о результатах оценивания решения на тестах жюри.

Запрос по каждой задаче можно делать не чаще одного раза в 5 минут. Для каждой подзадачи сообщаются баллы за эту подзадачу и результат проверки программы на каждом тесте.

В каждой задаче можно задать, какое из прошедших предварительную проверку решений будет оцениваться. В этом случае баллы начисляются за лучшее решение из следующих:

- выбранного явно;
- последнего принятого на проверку решения.

Если выбор не сделан, то будет оцениваться лучшее решение из следующих:

- тех решений, по которым просмотрены баллы;
- последнего принятого на проверку решения.

## Задача 5. «Мозаика»

Имя входного файла:	mosaic.in
Имя выходного файла:	mosaic.out
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	256 мегабайт
Максимальная оценка	100 баллов

Все элементы магнитной мозаики фирмы «АВВУУ» имеют прямоугольную форму. Два элемента можно соединить только в том случае, если у них совпадает хотя бы один из размеров: длина, ширина, или и то, и другое. Магнитные элементы поворачивать и переворачивать нельзя. Пару элементов мозаики, которые нельзя соединить, назовем *негармоничной*. Например, пара  $1 \times 2$  и  $2 \times 3$  является негармоничной, а пары  $2 \times 3$  и  $1 \times 3$  или  $2 \times 3$  и  $2 \times 3$  являются гармоничными.

Дизайнеры «АВВУУ» выложили все элементы мозаики в ряд, не соединяя их между собой. Назовем *набором* несколько подряд лежащих элементов мозаики в этом ряду. Они выбрали несколько наборов элементов, которые хотят оставить для создания инсталляции. Для каждого такого набора им нужно выяснить, есть ли в нем негармоничная пара элементов.

Требуется написать программу, которая для различных наборов подряд лежащих элементов мозаики определит номера элементов, образующих негармоничную пару, или сообщит, что такой пары нет.

### Формат входных данных

В первой строке входного файла записано одно число  $N$  – количество элементов, из которых состоит мозаика ( $2 \leq N \leq 100\,000$ ). В следующих  $N$  строках записаны по два целых числа  $A_i$  и  $B_i$ , задающих длину и ширину  $i$ -го элемента мозаики соответственно ( $1 \leq A_i, B_i \leq 10^9, 1 \leq i \leq N$ ).

В  $(N + 2)$ -й строке записано одно целое число  $K$  – количество наборов, в каждом из которых нужно определить номера двух негармоничных элементов ( $1 \leq K \leq 100\,000$ ). В следующих  $K$  строках записаны пары целых чисел  $N_1$  и  $N_2$  – номера первого и последнего элементов набора соответственно, в котором необходимо найти два негармоничных элемента мозаики ( $1 \leq N_1 < N_2 \leq N$ ).

### Формат выходных данных

Выходной файл должен содержать  $K$  строк, каждая из которых содержит два разделённых пробелом числа – номера элементов мозаики, образующих негармоничную пару в соответствующем наборе. Если решений несколько, можно вывести любое из них. Если в наборе негармоничная пара отсутствует, требуется вывести в соответствующей строке 0 0.

### Примеры входных и выходных данных

mosaic.in	mosaic.out
4	0 0
2 2	4 2
1 2	
1 3	
2 3	
2	
2 3	
2 4	

### Подзадачи и система оценки

Данная задача содержит четыре подзадачи. Для оценки каждой подзадачи используется своя группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы успешно пройдены.

#### Подзадача 1 (оценивается в 20 баллов)

Количество элементов мозаики  $N \leq 100$ , число наборов  $K \leq 100$ .

#### Подзадача 2 (оценивается в 30 баллов)

Количество элементов мозаики  $N \leq 1\,000$ , число наборов  $K \leq 1\,000$ .

#### Подзадача 3 (оценивается в 20 баллов)

Количество элементов мозаики  $N \leq 5\,000$ , число наборов  $K \leq 5\,000$ .

#### Подзадача 4 (оценивается в 30 баллов)

Количество элементов мозаики  $N \leq 100\,000$ , число наборов  $K \leq 100\,000$ .

## Задача 6. «Театр начинается с актеров»

Имя входного файла:	theatre.in
Имя выходного файла:	theatre.out
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	256 мегабайт
Максимальная оценка	100 баллов

Участники олимпиады пришли в казанский театр на спектакль, где играют  $N$  неизвестных для них актеров. В фойе театра висят портреты всех актеров труппы, которая в полном составе задействована в спектакле. Портреты не подписаны. Зрителям раздали программки, в которых для каждого действия спектакля приводится список фамилий участвующих в нем актеров, но не указаны их роли.

Театрал Виталий решил узнать, как выглядит каждый из актеров, упомянутых в программке. Для этого в антракте после каждого действия он выходил в фойе и сопоставлял портреты с увиденными актерами.

Требуется написать программу, которая по заданному числу актеров  $N$  и списку фамилий актеров, участвующих в каждом из  $M$  действий, определяет номер действия, после которого впервые становится возможным установить соответствие между фамилией актера из программки и его портретом.

### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $N$  – число актеров и  $M$  – количество действий в спектакле ( $1 < N \leq 100\,000$ ,  $1 \leq M \leq 100\,000$ ). В каждой из следующих  $M$  строк сначала записано количество актеров  $K_i$ , участвующих в  $i$ -ом действии ( $1 \leq K_i \leq N$ ,  $K_1 + K_2 + \dots + K_M \leq 100\,000$ ), а затем  $K_i$  различных натуральных чисел, не превосходящих  $N$ , обозначающих фамилии этих актеров. Соседние числа в каждой строке разделены пробелом.

### Формат выходных данных

Выходной файл должен содержать одну строку, состоящую из  $N$  записанных через пробел чисел.  $i$ -е число этой строки – это номер действия, после которого впервые становится возможным установить соответствие между  $i$ -м актером и его портретом. Если к концу спектакля установить соответствие между каким-либо актером и его портретом так и не удалось, то соответствующее число в строке должно быть равно нулю.

### Примеры входных и выходных данных

theatre.in	theatre.out
3 3 2 1 2 2 3 2 2 1 2	2 2 1
5 3 3 1 2 3 3 2 3 1 2 1 3	0 3 0 0 0
4 3 1 1 1 3 1 2	1 3 2 3

### Комментарий

В первом примере три актера участвуют в спектакле с тремя действиями. В первом действии участвуют два актера с номерами 1 и 2. Так как актеров всего трое, то после первого акта становится понятно, какой портрет соответствует актеру с номером 3, поэтому третье число строки выходного файла равно 1.

Во втором действии участвуют два актера с номерами 3 и 2. Поскольку только второй актер участвовал и в первом, и во втором действиях, то его портрет можно определить после второго действия. А так как портретов всего три, то после второго действия можно установить, что последний портрет соответствует актеру номер 1. Третье действие на ответ не влияет.

**Подзадачи и система оценки**

Данная задача содержит три подзадачи. Для оценки каждой подзадачи используется своя группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы пройдены.

**Подзадача 1 (оценивается в 30 баллов)**

Количество актеров  $N$  не превосходит 100, количество действий  $M$  не превосходит 100,  
 $K_1 + K_2 + \dots + K_M \leq 100$ .

**Подзадача 2 (оценивается в 30 баллов)**

Количество актеров  $N$  не превосходит 10000, количество действий  $M$  не превосходит 10000,  
 $K_1 + K_2 + \dots + K_M \leq 10000$ .

**Подзадача 3 (оценивается в 40 баллов)**

Количество актеров  $N$  не превосходит 100000, количество действий  $M$  не превосходит 100000,  
 $K_1 + K_2 + \dots + K_M \leq 100000$ .



- 1 означает, что Ёжик позвал Лошадь, и она действительно оказалась в той же клетке, что и он. В этом случае другие два числа равны 0, и программа-решение должна закончить свою работу.

Программа-решение не должна делать более 10 000 ходов.

### Пример взаимодействия

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

### Комментарий

Ёжик находился в клетке (1, 2). Сначала он попробовал позвать Лошадь в той же клетке (вывод: 0 0 1), но Лошади там не оказалось, и она сместилась вправо (ввод: 0 1 0). Ёжик сместился по диагонали, но Лошадь звать не стал (вывод: 1 -1 0), а Лошадь осталась на месте (ввод: 0 0 0). Ёжик сместился вправо и позвал Лошадь (вывод: 1 0 1). Лошадь оказалась в той же клетке и отозвалась (ввод: 1 0 0). Значит, изначально Лошадь находилась в клетке (2, 1), а встретились они в клетке (3, 1). Ёжик при этом сделал три хода и дважды запросил местоположение Лошади.

### Подзадачи и система оценки

В данной задаче две подзадачи. Каждый тест в обеих подзадачах оценивается отдельно. Оценка за тест вычисляется по формуле  $\min\{10, \text{round}(10 \times (J/S)^2)\}$ , где 10 – оценка в баллах за тест,  $S$  – количество ходов, которое потребовалось программе-решению, чтобы обнаружить Лошадь,  $J$  – количество ходов, которое требуется заданному эталонному решению при том же начальном положении Ёжика. Округление ведется по правилам математики.

#### Подзадача 1 (оценивается из 40 баллов)

$$2 \leq N, M \leq 10.$$

#### Подзадача 2 (оценивается из 60 баллов)

$2 \leq N, M \leq 30$ . В этой подзадаче количество запросов о том, есть ли Лошадь в текущей клетке, не должно превышать  $N \times M$ .

### Вспомогательная программа

Для тестирования своего решения вы можете использовать вспомогательную программу «runpair», которая находится у вас в каталоге «с:\work\runpair».

Она позволяет запустить две программы и перенаправить стандартный поток вывода первой программы на стандартный поток ввода второй программы и наоборот.

Для тестирования с ее помощью программы-решения вам придется помимо программы-решения написать программу, моделирующую поведение лошади. Тогда можно запустить одновременно программу, моделирующую поведение лошади, и программу-решение, с помощью команды

```
runpair «исполнимый файл лошади» «исполнимый файл ёжика»
```

и на экране будет отображен их диалог.

## Задача 8. «Ордынское войско»

Имя входного файла:	army.in
Имя выходного файла:	army.out
Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	256 мегабайт
Максимальная оценка	100 баллов

Готовясь к бою, хан Гирей пронумеровал всех воинов своего войска натуральными числами от 1 до  $N$ . Поскольку воины умеют сражаться, но не умеют считать, при любом построении в шеренгу они выстраиваются в произвольном порядке.

Одного или несколько воинов, стоящих в шеренге, будем называть *отрядом*. Отряд назовем *правильным*, если номера этих воинов в том порядке, в котором они стоят в шеренге, образуют упорядоченную по возрастанию последовательность чисел. Среди всех правильных отрядов хан Гирей выбирает *ударный отряд* – самый большой по количеству воинов. Так, в шеренге 1 3 2 4 из четырех воинов ударными являются отряды 1 3 4 и 1 2 4, а отряд 1 4 – один из правильных, но не ударный.

Некоторые воины являются личными телохранителями хана Гирея.

Требуется составить программу, определяющую количество таких шеренг, в которых телохранители хана образуют ударный отряд.

### Формат входных данных

В первой строке входного файла задано натуральное число  $N$  – общее количество воинов ( $1 \leq N \leq 15$ ). Во второй строке задано натуральное число  $K$  – количество телохранителей хана ( $1 \leq K \leq N$ ). В третьей строке через пробел указаны  $K$  различных натуральных чисел, не превосходящих  $N$ , – номера телохранителей хана в порядке возрастания.

### Формат выходных данных

Выходной файл должен содержать единственное число – количество различных расстановок всех воинов в шеренгу так, чтобы все телохранители хана были ударным отрядом в каждой из таких расстановок.

### Примеры входных и выходных данных

army.in	army.out
5 3 1 3 4	11
3 3 1 2 3	1
1 1 1	1

### Комментарий

В первом примере войско состоит из пяти воинов. Ударный отряд должен состоять из трех воинов с номерами 1, 3 и 4. Этому условию удовлетворяют следующие 11 шеренг: (1, 3, 2, 5, 4), (1, 3, 5, 2, 4), (1, 3, 5, 4, 2), (1, 5, 3, 2, 4), (1, 5, 3, 4, 2), (2, 1, 3, 5, 4), (2, 1, 5, 3, 4), (2, 5, 1, 3, 4), (5, 1, 3, 2, 4), (5, 1, 3, 4, 2), (5, 2, 1, 3, 4).

**Подзадачи и система оценки**

Данная задача содержит семь подзадач. Для оценки каждой подзадачи используется своя группа тестов. Баллы за подзадачу начисляются только в том случае, если все тесты из этой группы успешно пройдены.

**Подзадача 1 (оценивается в 40 баллов)**

$$1 \leq N \leq 8.$$

**Подзадача 2 (оценивается в 10 баллов)**

$$9 \leq N \leq 10.$$

**Подзадача 3 (оценивается в 10 баллов)**

$$N = 11.$$

**Подзадача 4 (оценивается в 10 баллов)**

$$N = 12.$$

**Подзадача 5 (оценивается в 10 баллов)**

$$N = 13.$$

**Подзадача 6 (оценивается в 10 баллов)**

$$N = 14.$$

**Подзадача 7 (оценивается в 10 баллов)**

$$N = 15.$$