

## Задача Bering. Спички детям не игрушка

|                         |  |
|-------------------------|--|
| Имя входного файла:     | <code>input.txt</code> или стандартный поток ввода   |
| Имя выходного файла:    | <code>output.txt</code> или стандартный поток вывода |
| Ограничение по времени: | 4 секунды  |
| Ограничение по памяти:  | 512 мегабайт   |

Лена играет со спичками. Естественный вопрос, посещающий любого школьника, играющего со спичками — а можно ли поджечь спичкой дерево?

Скажем, что дерево — это связный граф без циклов, вершины которого пронумерованы целыми числами  $1, 2, \dots, n$ , в каждой вершине которого также записано некоторое целое число  $p_v$ , являющееся приоритетом вершины  $v$ . Все приоритеты различны.

Оказывается, что если поджечь дерево, то оно, как и можно было ожидать, сгорит целиком. Однако процесс этот не быстрый. Сначала у дерева сгорает лист (*листом* называется вершина, имеющая ровно одного соседа) с минимальным приоритетом, затем сгорает лист с минимальным приоритетом из оставшихся вершин дерева, и так далее. Таким образом, вершины превращаются в листья и сгорают до тех пор, пока от дерева не останется лишь одна вершина, после чего она тоже сгорает.

Лена приготовила дерево из  $n$  вершин и в каждой вершине записала приоритет  $p_v = v$ . Лене с одной стороны интересно посмотреть, как горит дерево, но с другой она понимает, что если дерево поджечь, оно исчезнет насовсем. Лена добрая девочка, и деревья ей жалко, так что она хочет ограничиться выяснением ответов на некоторые вопросы про процесс сгорания дерева в уме. Лена хочет ответить на  $q$  вопросов, каждый из которых относится к одному из трёх следующих видов:

- «`up v`», присвоить вершине  $v$  приоритет  $1 + \max\{p_1, p_2, \dots, p_n\}$ ;
- «`when v`», выяснить, какой по счёту сгорит вершина  $v$ , если дерево поджечь сейчас;
- «`compare v u`», выяснить, какая из вершин  $v$  и  $u$  сгорит раньше, если дерево поджечь сейчас.

Заметим, что если приоритеты всех вершин сейчас различны, то и после выполнения запроса «`up`» они тоже останутся различными. Исходно они различны, поэтому в любой момент времени порядок сгорания листьев определён однозначно.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $q$  ( $2 \leq n \leq 200\,000$ ,  $1 \leq q \leq 200\,000$ ) — количество вершин дерева и количество вопросов.

В  $i$ -й из следующих  $n - 1$  строк находятся два целых числа  $v_i, u_i$  ( $1 \leq v_i, u_i \leq n$ ), задающие концы  $i$ -го ребра дерева.

Каждая из оставшихся  $q$  строк содержит операцию одного из трёх типов.

- «`up v`» ( $1 \leq v \leq n$ ) — присвоить новый приоритет вершине  $v$ ;
- «`when v`» ( $1 \leq v \leq n$ ) — определить момент сгорания вершины  $v$  для текущего дерева;
- «`compare v u`» ( $1 \leq v, u \leq n$ ,  $v \neq u$ ) — определить, какая из вершин  $v$  и  $u$  сгорит раньше для текущего дерева.

Гарантируется, что среди запросов хотя бы один имеет тип «`when`» или «`compare`».

### Формат выходных данных

Для каждого запроса типа «`when`» нужно вывести одно целое число от 1 до  $n$  — момент времени, когда сгорит вершина  $v$ .

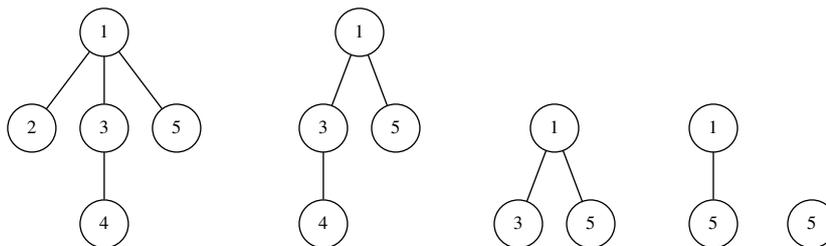
Для запроса типа «`compare`» выведите  $v$  или  $u$ , в зависимости от того, какая вершина сгорит раньше.

## Примеры

| ВВОД  | ВЫВОД                    |
|---|--------------------------|
| <pre>5 7 1 5 1 2 1 3 4 3 when 1 when 2 when 3 when 4 when 5 compare 2 3 compare 3 4</pre> | <pre>4 1 3 2 5 2 4</pre> |
| <pre>5 5 1 5 1 2 1 3 4 3 up 1 compare 2 4 compare 4 3 compare 3 1 compare 1 5</pre>       | <pre>2 4 3 5</pre>       |

## Пояснение

В первом примере процесс сгорания исходного дерева проиллюстрирован на картинке:



В частности, порядок сгорания вершин следующий: [2, 4, 3, 1, 5].

Во втором примере после применения операции «up» порядок сгорания вершин станет следующим: [2, 4, 3, 5, 1]

## Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Открытая олимпиада школьников по программированию 2018/19, первый тур  
Москва, Сочи, 7 марта 2019

| Группа | Баллы | Доп. ограничения           | Необх.<br>группы | Комментарий                                      |
|--------|-------|----------------------------|------------------|--|
|        |       | $n, q$                     |                  |  |
| 0      | 0     | –                          | –                | Тесты из условия                                 |
| 1      | 9     | $n \leq 1000, q \leq 1000$ | 0                |  |
| 2      | 12    | –                          | –                | Только вопросы «when», «compare»                 |
| 3      | 23    | –                          | –                | Только вопросы «up», «compare»                   |
| 4      | 25    | –                          | –                | Граф является цепью, т.е. $v_i = i, u_i = i + 1$ |
| 5      | 31    | –                          | 0–4              | <b>Offline-проверка</b>                          |

## Задача Krusenstern. Трубка с шариками

|                         |  |
|-------------------------|--|
| Имя входного файла:     | <code>input.txt</code> или стандартный поток ввода   |
| Имя выходного файла:    | <code>output.txt</code> или стандартный поток вывода |
| Ограничение по времени: | 1 секунда  |
| Ограничение по памяти:  | 512 мегабайт   |

В запаянной с обеих сторон трубке, разделённой на  $n$  секторов, находятся  $m$  шариков. Шарик с номером  $i$  изначально находится в секторе с номером  $a_i$ , причём все шарики находятся в различных секторах. За одну секунду разрешается выполнить одну из двух операций:

1. Сдвинуть все шарики на один сектор влево. В этом случае все шарики по очереди, начиная с самого левого, перемещаются на одну позицию влево. Если в момент сдвига шарик  $i$  находится в самом левом секторе или в секторе слева расположен шарик, то шарик  $i$  не двигается.
2. Сдвинуть все шарики на один сектор вправо. В этом случае все шарики по очереди, начиная с самого правого, перемещаются на одну позицию вправо. Если в момент сдвига шарик находится в самом правом секторе или в секторе справа расположен шарик, шарик не двигается.

В  $k$  секторах трубки расположены переключатели, изначально находящиеся в выключенном состоянии. Как только в сектор попадает шарик, переключатель переходит во включённое состояние и **остаётся** в таком состоянии навсегда.

Вася хочет добиться того, чтобы все переключатели оказались включены. Для этого он хочет **один раз** выполнить одно из следующих действий:

1. Выбрать два целых числа  $x$  и  $y$  ( $x, y \geq 0$ ) и выполнить  $x$  раз первую операцию, затем  $y$  раз вторую операцию.
2. Выбрать два целых числа  $x$  и  $y$  ( $x, y \geq 0$ ) и выполнить  $x$  раз вторую операцию, затем  $y$  раз первую операцию.

За какое минимальное количество операций Вася сможет добиться того, чтобы все переключатели оказались включены?

### Формат входных данных

В первой строке заданы три целых числа  $n$ ,  $m$  и  $k$  ( $1 \leq n \leq 200\,000$ ,  $1 \leq m \leq n$ ,  $1 \leq k \leq n$ ) — число секторов в трубке, число шариков и переключателей, соответственно.

Во второй строке заданы  $m$  различных целых чисел ( $1 \leq a_1 < a_2 < \dots < a_m \leq n$ ) — сектора, в которых изначально располагаются шарики.

В третьей строке заданы  $k$  различных целых чисел ( $1 \leq b_1 < b_2 < \dots < b_k \leq n$ ) — сектора, в которых располагаются переключатели.

### Формат выходных данных

Выведите одно целое число — минимальное количество операций.

## Примеры

| ВВОД                          | ВЫВОД |
|-------------------------------|-------|
| 5 1 3<br>2<br>1 3 5           | 5     |
| 10 3 4<br>1 7 9<br>2 6 8 10   | 3     |
| 11 5 3<br>2 5 7 9 11<br>5 7 9 | 0     |
| 15 2 2<br>1 15<br>2 14        | 2     |

## Пояснение

В первом примере Васе выгодно совершить первое действие с  $x = 1, y = 4$ , то есть суммарно будет выполнено 5 операций. После движения шарика влево переключатель в секторе 1 окажется во включённом состоянии. Затем Вася сдвинет шарик вправо два раза, после чего переключатель в секторе 3 окажется во включённом состоянии, после чего Вася сдвинет шарик вправо ещё два раза, в результате чего переключатель в секторе 5 окажется во включённом состоянии.

В третьем примере в каждом секторе с переключателем уже есть шарик, поэтому никаких операций выполнять не требуется. Можно считать, что Вася выполнит первое действие с  $x = 0, y = 0$ .

В четвёртом примере Васе выгодно совершить первое действие с  $x = 1, y = 1$ . После движения шариков на один сектор влево, переключатель в секторе 14 окажется включён, а шарик будет находиться на позициях 1 и 14. После этого Вася сдвинет шарик на 1 сектор вправо, и выключатель на позиции 2 окажется включён.

## Система оценки

Тесты к этой задаче состоят из семи групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

| Группа | Баллы | Дополнительные ограничения |                  |                  | Комментарий             |
|--------|-------|----------------------------|------------------|------------------|-------------------------|
|        |       | $n$                        | $m$              | $k$              |                         |
| 0      | 0     | –                          | –                | –                | Тесты из условия        |
| 1      | 6     | $n \leq 15$                | $m = 1$          | $k \leq 15$      |                         |
| 2      | 11    | $n \leq 15$                | $m \leq 15$      | $k \leq 15$      |                         |
| 3      | 13    | $n \leq 100$               | $m \leq 100$     | $k \leq 100$     |                         |
| 4      | 14    | $n \leq 500$               | $m \leq 500$     | $k \leq 500$     |                         |
| 5      | 11    | $n \leq 3000$              | $m \leq 3000$    | $k \leq 3000$    |                         |
| 6      | 21    | $n \leq 10\,000$           | $m \leq 10\,000$ | $k \leq 10\,000$ |                         |
| 7      | 24    | –                          | –                | –                | <b>Offline-проверка</b> |

## Задача Polo. Небоскребы

Имя входного файла: `input.txt` или стандартный поток ввода  
Имя выходного файла: `output.txt` или стандартный поток вывода  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Даша очень любит приключения. На днях она отправилась в очередное путешествие, вместе со своими верными спутниками — Башмачком, Рюкзаком и Картой. По пути им встретился удивительный город, представляющий из себя  $n$  улиц, направленных вдоль восточного направления, и  $m$  улиц, направленных вдоль южного направления, образующих естественным образом  $nm$  перекрестков. Каково же было удивление Дашы, когда, вместо привычного «Я карта!», она услышала, что на любом пересечении  $i$ -й улицы, идущей на восток, и  $j$ -й улицы, идущей на юг, можно увидеть монументальной величины небоскреб. Загоревшись любопытством, она решила исследовать высоту городских строений.

Проходя через перекресток на пересечении  $i$ -й и  $j$ -й улицы соответствующих направлений, Даша оглядывает две улицы, на которых находится. Узнав высоты всех небоскрёбов, которые находятся на этих двух улицах, Даша задаётся вопросом: как можно переназначить высоты всем небоскрёбам на этих двух улицах, чтобы максимальная высота была как можно меньше, а результат сравнения высот у любых двух небоскрёбов на одной улице не изменился.

Сформулируем задачу формально. На каждом из  $nm$  перекрёстков Даша независимо от других перекрёстков решает задачу. Сейчас она видит  $n + m - 1$  небоскрёбов, и для каждого видит его настоящую высоту. При этом любые два небоскрёба можно сравнить по высоте и получить результат «больше», «меньше» или «равно». Теперь Даша хочет выбрать некоторое целое  $x$ , после чего назначить каждому небоскрёбу целую положительную высоту от 1 до  $x$ . При назначении высот Даша хочет сохранить относительный порядок в каждой из улиц. То есть результат сравнения высоты любых двух небоскрёбов на текущей улице восточного направления не должен измениться, и результат сравнения высоты любых двух небоскрёбов на текущей улице южного направления не должен измениться. При этом небоскрёбы, находящиеся только на текущей улице восточного направления не сравниваются с небоскрёбами, находящимися на текущей улице южного направления, а небоскрёб, находящийся на пересечении, может сравниваться и с теми и с теми. Для каждого перекрёстка Даша хочет **независимо** вычислить минимальное возможное подходящее  $x$ .

Например, если выбранный перекрёсток и улицы ему соответствующие выглядят следующим образом:

|    |    |    |    |    |  |
|----|----|----|----|----|--|
|    |    | 10 |    |    |  |
|    |    | 20 |    |    |  |
| 11 | 13 | 14 | 15 | 16 |  |
|    |    | 10 |    |    |  |

То можно заменить высоты небоскрёбов следующим образом (заметим, что все сравнения на меньше, больше или равно внутри улицы южного направления и внутри улицы восточного направления при этом не поменяются).

|   |   |   |   |   |  |
|---|---|---|---|---|--|
|   |   | 1 |   |   |  |
|   |   | 4 |   |   |  |
| 1 | 2 | 3 | 4 | 5 |  |
|   |   | 1 |   |   |  |

Наибольшее использованное число равно 5, а значит ответом для этого перекрёстка было бы 5.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 1000$ ) — количество улиц, идущих вдоль восточного направления и южного соответственно.

Каждая из следующих  $n$  содержит по  $m$  целых чисел  $a_{i,j}$  ( $1 \leq a_{i,j} \leq 10^9$ ) в каждой. Число  $a_{i,j}$ , находящееся на  $j$ -й позиции в  $i$ -й строке, характеризует высоту небоскреба, находящегося на пересечении  $i$ -й улицы восточного направления и  $j$ -й улицы южного направления.

### Формат выходных данных

Выведите  $n$  строк, по  $m$  целых положительных чисел, где число  $x_{i,j}$ , находящееся на  $j$ -й позиции в  $i$ -й строке — это ответ на задачу в перекрестке на пересечении  $i$ -й улицы восточного направления и  $j$ -й улицы южного направления.

### Примеры

| ВВОД                  | ВЫВОД          |
|-----------------------|----------------|
| 2 3<br>1 2 1<br>2 1 2 | 2 2 2<br>2 2 2 |
| 2 2<br>1 2<br>3 4     | 2 3<br>3 2     |

### Пояснение

В первом примере из условия ни на одном из перекрестков максимальную высоту уменьшить не получится, а значит можно просто не менять высоты.

Во втором примере из условия имеем следующие ответы:

- Для пересечения первой строки и первого столбца:  $\begin{bmatrix} 1 & 2 \\ 3 & \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 \\ 2 & \end{bmatrix}$
- Для пересечения первой строки и второго столбца:  $\begin{bmatrix} 1 & 2 \\ & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 \\ & 3 \end{bmatrix}$
- Для пересечения второй строки и первого столбца:  $\begin{bmatrix} 1 \\ 3 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 2 & 3 \end{bmatrix}$
- Для пересечения второй строки и второго столбца:  $\begin{bmatrix} & 2 \\ 3 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} & 1 \\ 1 & 2 \end{bmatrix}$

### Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп.

| Группа | Баллы | Дополнительные ограничения |                          | Необх. группы | Комментарий       |
|--------|-------|----------------------------|--------------------------|---------------|-------------------|
|        |       | $n, m$                     | $a_{i,j}$                |               |                   |
| 0      | 0     | –                          | –                        | –             | Тесты из условия. |
| 1      | 14    | $1 \leq n, m \leq 2$       | –                        | 0             |                   |
| 2      | 24    | $1 \leq n, m \leq 100$     | –                        | 0, 1          |                   |
| 3      | 28    | $1 \leq n, m \leq 1000$    | $1 \leq a_{i,j} \leq 20$ | 0             |                   |
| 4      | 34    | $1 \leq n, m \leq 1000$    | –                        | 0, 1, 2, 3    |                   |

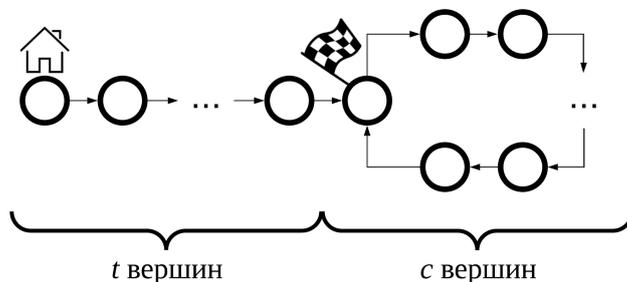
## Задача Cook. Кооперативная игра

|                         |   |
|-------------------------|---|
| Имя входного файла:     | input.txt или стандартный поток ввода   |
| Имя выходного файла:    | output.txt или стандартный поток вывода |
| Ограничение по времени: | 1 секунда                               |
| Ограничение по памяти:  | 512 мегабайт                            |

Это интерактивная задача с открытыми тестами. Все тесты будут высланы вам в виде сообщения в тестирующей системе в начале соревнования.

Миша очень любит играть в кооперативные игры с неполной информацией. Сегодня он предложил десяти своим друзьям сыграть в кооперативную игру «Озеро».

Миша уже нарисовал игровое поле для предстоящей игры. Оно представляет из себя **ориентированный** граф, состоящий из двух частей. Одна из этих частей — дорожка вдоль берега озера, являющаяся циклом из  $s$  вершин. Вторая часть — тропинка от домика к озеру, представляющая из себя цепь из  $t$  вершин, причём из последней вершины этой цепи проведено ребро в одну из вершин дорожки вокруг озера — в ту, из которой на озеро открывается наилучший вид. Игровое поле Миша решил держать в секрете, так что никто кроме него не знает ни  $t$ , ни  $s$ .



В начале игры игровые фишки всех десяти игроков, для удобства пронумерованных цифрами от 0 до 9, располагаются в начале тропинки, в вершине у домика. Затем не более  $q$  раз Миша будет одновременно передвигать в следующие вершины фишки игроков, высказавших такое желание на этом ходу, а после этого объявлять, фишки каких игроков находятся в одной вершине, а каких — в разных. Обратите внимание, что у каждой вершины игрового поля есть только одна следующая, при этом каждая кроме стартовой и вершины с лучшими видами является следующей только для одной вершины. Стартовая вершина не является следующей ни для какой вершины, а вершина с лучшими видами является следующей для последней вершины цепочки и для одной из вершин цикла. Мише так понравилась идея не сообщать друзьям  $t$  и  $s$ , что и  $q$  он решил тоже не объявлять.

Цель игры — перевести игровые фишки всех игроков в вершину с наилучшим видом на озеро, то есть в ту, которая отмечена финишным флагом. Мишины друзья не представляют, как можно выиграть в такой игре не зная ни  $s$ , ни  $t$ , ни  $q$ , но, к счастью для них, они ещё и ваши друзья. Помогите им: скоординируйте их действия так, чтобы победить.

### Протокол взаимодействия

Для удобства пронумеруем друзей Миши целыми числами от 0 до 9 включительно.

Для того, чтобы отдать команды перемещения друзьям, выведите «next» а затем через пробел номера друзей, которым необходимо отдать команды. Например, чтобы отдать команду перемещения друзьям с номерами 0, 2, 5 и 9 выведите «next 0 2 5 9». На каждом ходу обязательно перемещать хотя бы одного из друзей.

В качестве ответа проверяющая программа выдаст сначала число  $k$ , а потом 10 цифр, разбитых на  $k$  групп, разделённых пробелами. Друзья, соответствующие цифрам, находящимся в одной группе, находятся в одной вершине. Друзья, соответствующие цифрам, находящимся в разных группах, находятся в разных вершинах. Цифры в каждой группе следуют в порядке возрастания.

Например ответ проверяющей программы «2 05 12346789» означает, что друзья с номерами 0 и 5 находятся в одной вершине, а все остальные друзья в одной и той же, но другой вершине. Ответ проверяющей программы «4 01 567 234 89» означает, что друзья Миши находятся в четырёх

различных вершинах: в первой находятся друзья с номерами 0 и 1, во второй — друзья с номерами 5, 6 и 7, в третьей — друзья с номерами 2, 3 и 4, а в четвёртой — друзья с номерами 8 и 9.

Если вы превысите лимит на количество ходов или нарушите протокол взаимодействия, то вместо информации о местоположении друзей вашей программе на вход будет передана строка «stop». Считав такую строку, ваша программа должна немедленно завершить работу, иначе она может получить вердикт тестирования, отличный от настоящего.

После того, как все друзья соберутся в вершине с лучшим видом необходимо вывести «done» и завершить работу программы.

После вывода каждой строки сбрасывайте буфер вывода — для этого используйте `flush(output)` на языке Паскаль или Delphi, `fflush(stdout)` или `cout.flush()` в C/C++, `sys.stdout.flush()` на языке Python, `System.out.flush()` на языке Java.

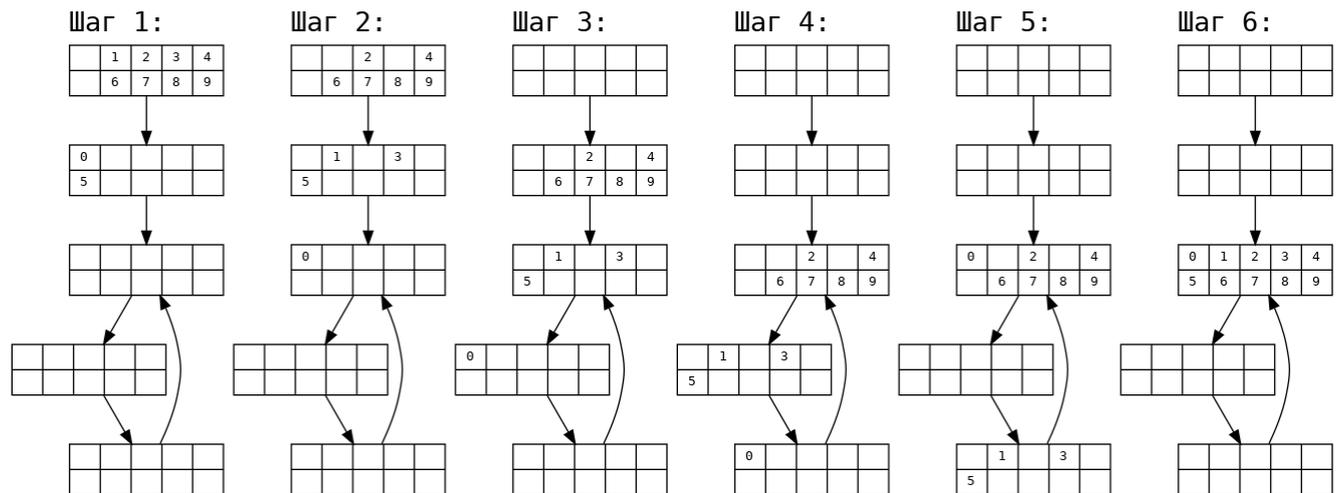
### Пример

| ВВОД           | ВЫВОД                    |
|----------------|--------------------------|
| 2 05 12346789  | next 0 5                 |
| 3 246789 135 0 | next 0 1 3               |
| 3 246789 0 135 | next 2 3 0 1 4 5 6 7 8 9 |
| 3 246789 0 135 | next 9 8 7 6 5 4 3 2 1 0 |
| 2 135 0246789  | next 0 1 3 5             |
| 1 0123456789   | next 1 3 5               |
|                | done                     |

### Пояснение

В условии в примере взаимодействия вводимые и выводимые данные расположены для удобства восприятия в хронологическом порядке, при реальном взаимодействии никакие «лишние» переводы строк возникать не должны.

В примере перемещение друзей происходит следующим образом:



### Система оценки

Тесты к этой задаче состоят из четырёх групп. Каждая группа тестируется только при прохождении всех тестов **необходимых** групп. В каждой тестируемой группе ваша программа будет запущена на **всех** тестах, каждый пройденный тест будет оценен в 1 балл. Группа считается пройденной только при прохождении всех тестов в ней.

Открытая олимпиада школьников по программированию 2018/19, первый тур  
Москва, Сочи, 7 марта 2019

| Группа | Тесты  | Ограничения         |                                       | Необх. группы | Комментарий     |
|--------|--------|---------------------|---------------------------------------|---------------|-----------------|
|        |        | $(t + c)$           | $q$                                   |               |                 |
| 0      | 1      | $(t + c) \leq 65$   | $6600 \leq q \leq 10\,000$            | —             | Тест из условия |
| 1      | 2-10   | $(t + c) \leq 65$   | $6600 \leq q \leq 10\,000$            | —             |                 |
| 2      | 11-20  | $(t + c) \leq 100$  | $(t + c)^2 \leq q \leq 10\,000$       | 0, 1          |                 |
| 3      | 21-61  | $(t + c) \leq 1000$ | $4 \cdot (t + c) \leq q \leq 10\,000$ | 0, 1          |                 |
| 4      | 62-100 | $(t + c) \leq 1000$ | $3 \cdot (t + c) \leq q \leq 10\,000$ | 0, 1          |                 |