

## Задача 2. Односвязный список

|                         |                         |
|-------------------------|-------------------------|
| Источник:               | базовая                 |
| Имя входного файла:     | <code>input.txt</code>  |
| Имя выходного файла:    | <code>output.txt</code> |
| Ограничение по времени: | 1 секунда               |
| Ограничение по памяти:  | разумное                |

Требуется реализовать односвязный список на массиве. В каждом узле списка хранится строковое значение и индекс следующего узла. Дано начальное состояние списка, а также последовательность операций двух видов: добавление и удаление узла. Нужно выполнить все операции, и после этого вывести значения всех узлов списка в порядке их следования в цепи.

У каждого узла списка есть индекс в едином массиве, в котором всё хранится. В начальном состоянии списка есть  $N$  узлов, имеющих индексы от 0 до  $N - 1$  в порядке их задания. При добавлении нового узла он дописывается в конец массива. То есть первый добавленный узел имеет индекс  $N$ , второй добавленный —  $N + 1$ , и так далее.

Нужно выполнять операции двух видов:

0. *Добавление узла.* При этом указывается индекс узла, после которого нужно вставить новый узел, и строковое значение для нового узла. Если индекс равен -1, то узел надо вставить в начало списка (перед самым первым элементом). После выполнения операции нужно вывести индекс нового узла (они назначаются по порядку, см. выше).
1. *Удаление узла.* При этом указывается индекс узла, и удалить нужно тот узел, который идёт сразу после него. Гарантируется, что узел с указанным индексом не последний. Если указан индекс -1, значит нужно удалить самый первый элемент списка (гарантируется, что он есть). После выполнения операции нужно вывести строковое значение удалённого узла.

В задаче используется “мультитест”: в одном входном файле записано много отдельных тестов.

### Формат входных данных

В первой строке файла записано одно целое число  $T$  — количество тестов в файле. Далее в файле идут тесты ( $T$  штук) подряд, один за другим.

Первая строка теста начинается с трёх целых чисел:  $N$  — изначальное количество узлов в связном списке,  $F$  — индекс первого элемента списка и  $Q$  — количество операций, которые нужно выполнить ( $0 \leq F < N \leq 10^5$ ,  $0 \leq Q \leq 10^5$ ).

Затем идёт  $N$  строк, в которых описываются узлы связного списка в порядке увеличения индекса. Описание каждого узла состоит из его строкового значения и индекса следующего узла в списке (или -1, если следующего нет).

Наконец, в тесте идут  $Q$  строк, которые описывают операции над списком. В каждой строке сначала записан тип операции: 0 — добавление, 1 — удаление. Затем указан индекс узла, после которого нужно вставить/удалить узел. Если описывается операция вставки узла, то в конце также задано строковое значение нового узла.

У каждого узла строковое значение имеет длину от 1 до 7 символов включительно, и состоит из произвольных печатаемых символов ASCII кроме пробела (такие символы имеют код от 33 до 126 включительно).

Сумма  $N$  по всем тестам не превышает  $10^5$ , аналогично для суммы всех  $Q$ .

## Формат выходных данных

Для каждого теста нужно вывести следующее:

1. результаты выполнения операций ( $Q$  строк)
2. строка "===" (три знака равенства)
3. строковые значения всех узлов списка после выполнения операций (в порядке их следования в списке)
4. снова строка "==="

## Пример

| input.txt | output.txt |
|-----------|------------|
| 3         | ===        |
| 5 3 0     | 1.111      |
| 4.283 2   | 2.718      |
| 2.718 4   | 3.141      |
| 5.0 -1    | 4.283      |
| 1.111 1   | 5.0        |
| 3.141 0   | ===        |
| 1 0 5     | 1          |
| zero -1   | 2          |
| 0 -1 one  | 3          |
| 0 -1 two  | 4          |
| 0 1 three | one        |
| 0 0 four  | ===        |
| 1 2       | two        |
| 4 2 2     | three      |
| \$45\$ 1  | zero       |
| %drill# 3 | four       |
| &qw6: 0   | ===        |
| *a-+r -1  | *a-+r      |
| 1 1       | 4          |
| 0 2 \num\ | ===        |
|           | &qw6:      |
|           | \num\      |
|           | \$45\$     |
|           | %drill#    |
|           | ===        |

## Пояснение к примеру

В примере три теста.

Первый тест полностью совпадает с примером к задаче “Распечатать список”: дано 5 узлов и 0 операций. Т.к. операций нет, в ответе записана сразу строка "===". Потом записан ответ как в задаче “Распечатать список” и ещё одна строка "===".

Во втором тесте изначально есть только один узел. Заметим, что в каждом узле значение равно индексу, записанному по-английски: **zero**, **one**, **two**, **three**, **four**.

Первые четыре операции задают вставку элементов: сначала вставляется два узла в начало, получается список **two**, **one**, **zero**. Затем вставляется узел после узла **one** и ещё один узел после узла **zero**, получается список **two**, **one**, **three**, **zero**, **four**. Последняя операция удаления: удаляется узел, который стоит **после** узла **two**, то есть узел **one**.