

Задача 4. Двусвязный список

Источник:	основная
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 двусвязный список. В каждом узле списка хранится целочисленное значение и индексы следующего и предыдущего узлов. 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:

1. *Добавление узла спереди.* При этом указывается индекс узла, после которого нужно вставить новый узел, и 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. *Добавление узла сзади.* При этом указывается индекс узла, перед которым нужно вставить новый узел, и значение для нового узла. Если индекс равен -1, то узел надо вставить в конец списка (после самого последнего элемента). После выполнения операции нужно вывести индекс нового узла.
0. *Удаление узла.* При этом указывается индекс узла, который нужно удалить. После выполнения операции нужно вывести значение удалённого узла.

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

Подсказка: Возможно, удобнее будет хранить связный список в виде двунаправленного кольцевого списка со вспомогательным элементом, например, сдвинув нумерацию узлов в программе на один.

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

Затем идёт N строк, в которых описываются узлы связного списка в порядке увеличения индекса. Описание каждого узла состоит из его целочисленного значения и двух индексов: следующего узла в списке и предыдущего (или -1, если одного из них нет).

Наконец, в тесте идут Q строк, которые описывают операции над списком. В каждой строке сначала записан тип операции: 1 — добавление спереди, -1 — добавление сзади, 0 — удаление. Затем указан индекс узла. Если описывается операция вставки, то в конце также задано целочисленное значение нового узла.

Все значения узлов лежат в диапазоне от 0 до 10^6 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. целочисленные значения всех узлов списка после выполнения операций (в порядке их следования в списке)
4. снова строка "==="

Пример

input.txt	output.txt
3	===
5 3 2 0	1111
4283 2 4	2718
2718 4 3	3141
5000 -1 0	4283
1111 1 -1	5000
3141 0 1	===
1 0 0 5	1
0 -1 -1	2
1 -1 1000	3
-1 -1 2000	4
1 1 3000	2000
-1 0 4000	===
0 2	1000
4 2 3 2	3000
45 1 2	4000
74 3 0	0
16 0 -1	===
56 -1 1	74
0 1	4
1 2 35	===
	16
	35
	45
	56
	===

Пояснение к примеру

В примере три теста, похожие на тесты из задачи “Односвязный список”. Во всех тестах заменены строковые значения на целочисленные, во втором тесте также другой порядок узлов.