

## Списки.

Вам предстоит реализовать класс, представляющий из себя двусвязный список.

### A. Добавление.

В этой задаче в первой строчке входного файла записывается  $N$  — количество элементов, которые будут добавлены в ваш список. Затем, в каждой из  $N$  строчек написано слово `front` или `back` и число, которое надо добавить, соответственно, в начало или конец списка.

Input	Output
4 front 5 front 3 back -1 front 0	0 3 5 -1

### B. Удаление.

В первой строке записано количество элементов в начальном списке. Во второй строчке элементы этого списка. Далее вводится  $M$  — количество операций удаления. Далее идут  $M$  строчек с описанием операций удаления. Операции удаления бывают трех типов.

- `front` — удаление первого элемента.
- `back` — удаление последнего элемента.
- `num` — нужно удалить все элементы списка, равные `num`.

Выведите список чисел, оставшихся после удалений. Если список уже пуст, а мы хотим удалить первый или последний элемент, выведите `break` и прекратите работу.

Input	Output
5 1 2 1 3 5 4 1 7 front back	3

### C. Добавление и удаление.

В первой строке вводится количество чисел в начальном списке. Во второй строчке элементы этого списка. Далее вводится  $M$  — количество запросов. Далее идут  $M$  строчек с описанием запросов. Запросы бывают 6 типов.

- `pop_front` — удаление первого элемента.
- `pop_back` — удаление последнего элемента.
- `delete num` — нужно удалить все элементы списка, равные `num`.
- `push_front num` — добавление `num` в начало.
- `push_back num` — добавление `num` в конец.
- `duplicate num` — нужно продублировать все элементы равные `num`.

Выведите список чисел, оставшихся после удалений. Если список уже пуст, а мы хотим удалить первый или последний элемент, выведите `break` и прекратите работу.

Input	Output
4 1 2 3 1 1 duplicate 1	1 1 2 3 1 1

### D. Сортировка.

В первой строке вводится количество чисел в начальном списке ( $N \leq 1000$ ). Во второй строчке элементы этого списка. В ответ выведите список отсортированный по возрастанию.

Input	Output
2 2 1	1 2

### E. Сортировка-2.

В первой строке вводится количество чисел в начальном списке ( $N \leq 100000$ ). Во второй строчке элементы этого списка. В ответ выведите список отсортированный по возрастанию.

Input	Output
2 2 1	1 2

F. Слабое звено.

В Берляндии по воскресеньям проходит известное шоу — игра «Слабое звено».

В игре принимают участие  $n$  игроков, которые выстраиваются в круг. Каждому игроку сопоставлен рейтинг — некоторое целое число  $a_i$ .

Игра проходит в несколько раундов, каждый из которых выглядит следующим образом:

В очередном раунде принимают участие все ещё не выбывшие игроки. Все игроки, которые по рейтингу строго слабее обоих своих соседей по кругу, объявляются слабым звеном и выбывают из игры. Все оставшиеся участники сдвигаются чуть плотнее, чтобы снова образовывать круг. Игра заканчивается, если после очередного раунда осталось ровно два человека, или если после очередного раунда не выбыл ни один человек. Иначе начинается новый раунд. Можно показать, что если до очередного раунда в игре оставалось хотя бы три участника, то после одного раунда гарантированно останется не менее двух участников.

Вам нужно выяснить для каждого участника, останется ли он до конца, или номер раунда, в котором он покинет игру.

*Входные данные.* В первой строке дано одно целое число  $n$  ( $2 \leq n \leq 200000$ ) — количество участников в игре.

Вторая строка содержит  $n$  целых чисел  $a_i$  ( $1 \leq a_i \leq 200000$ ) — рейтинги всех участников игры в том порядке, в котором они стоят, при этом участник с номером  $n$  является соседом участника с номером 1.

*Выходные данные* Выведите  $n$  целых чисел — номер раунда, в котором участник игры с номером  $i$  покинет игру, или 0, если этот игрок останется до конца игры.

Раунды нумеруются последовательными целыми числами начиная с 1.

Input	Output
5 4 5 5 2 3	3 0 0 1 2
5 5 1 3 1 5	0 1 2 1 0
4 6 5 5 6	0 0 0 0

*Примечание.* В первом примере игра проходит следующим образом (при помощи  $\_$  обозначаются выбывшие участники):

$[4; 5; 5; 2; 3] \rightarrow [4; 5; 5; \_; 3] \rightarrow [4; 5; 5; \_; \_] \rightarrow [\_; 5; 5; \_; \_]$

После этого игра заканчивается, так как осталось ровно два человека.

Во втором примере игра проходит следующим образом:

$[5; 1; 3; 1; 5] \rightarrow [5; \_; 3; \_; 5] \rightarrow [5; \_; \_; \_; 5]$

В третьем примере нет ни одного игрока, который был бы одновременно слабее обоих своих соседей, поэтому игра заканчивается, не успев начаться.