

Задача 3. Динамический поиск

Источник:	повышенной сложности*, обязательна на «отлично»
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда*
Ограничение по памяти:	разумное

Имеется множество целых чисел X , изначально оно пустое. Нужно выполнить M заданных операций над этим множеством.

Есть три типа операций:

1. **add** v — добавить число v в множество X . Если такого числа ещё не было в множестве, надо его добавить и напечатать слово **added**. Если такое число уже есть в множестве, нужно напечатать слово **dup** и ничего не делать.
2. **remove** v — удалить число v из множества X . Если такое число есть в множестве, нужно его удалить и напечатать слово **removed**. А если такого числа нет, нужно напечатать слово **miss** и ничего не делать.
3. **lower** v — найти минимальное число в множестве X , которое больше или равно заданному v (т.е. **lower_bound**). Если такое число в множестве есть, нужно напечатать в файл. А если его нет, то есть если v больше всех чисел множества X , то нужно напечатать **###** (три символа решётки, ASCII 35).

Внимание: операции нужно выполнять в режиме “online”: считывать операцию из файла разрешается только после того, как все предыдущие операции уже выполнены.

Задачу нужно решать **используя сбалансированное дерево поиска**.

Формат входных данных

В первой строке содержится целое число M — количество операций ($1 \leq M \leq 3 \cdot 10^5$). В остальных M строках записаны операции в порядке их выполнения. Все числа v в файле целые и по абсолютной величине не превышают 10^9 .

Формат выходных данных

Нужно вывести M строк, в каждой из которых требуется записать результат выполнения соответствующей операции.

Пример

input.txt	output.txt
16	added
add 7	added
add 3	added
add 5	dupe
add 5	added
add 10	dupe
add 7	miss
remove 6	removed
remove 5	added
add 5	removed
remove 3	5
lower 2	5
lower 5	added
add 1	1
lower 0	10
lower 10	###
lower 15	