Задача 1. Хип

Источник:	базовая
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда*
Ограничение по памяти:	разумное

Реализуйте кучу для поиска минимального элемента среди целых чисел.

Формат входных данных

В первой строке входного файла дано целое число Q — количество операций над кучей ($1 \leqslant Q \leqslant 2 \cdot 10^5$).

В следующих Q строках записаны производимые операции: добавление элемента в кучу и извлечение из кучи.

Операция добавления начинается со слова `push`, за которыим следует добавляемое в кучу число. Добавляемые числа по модулю не превосходят 10^5.

Операция извлечения значения из кучи обоозначается словом `pop`.

Формат выходных данных

Для каждой операции извлечения из кучи неоходимо в выходной файл в отдельную строку вывести число, которое было извлечено.

В случае, когда на момент извлечения куча была пуста, нужно напечатать `Heap is empty`.

Пример

input.txt	output.txt
12 push 1 push 2 pop push 3 pop pop pop push 4 pop pop push 5 pop	1 2 3 Heap is empty 4 Heap is empty 5

Задача 2. Сортировка кучей

Источник:	основная*
Имя входного файла:	`input.bin`
Имя выходного файла:	`output.bin`
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

В первых четырёх байтах входного файла задано целое число N — количество чисел в массиве A. Далее идут N четырёхбайтовых целых чисел — содержимое массива A. Размер массива лежит в диапазоне: $0 \leqslant N \leqslant 500\,000$.

Требуется отсортировать массив A по неубыванию, используя **алгоритм сортировки кучей**. В простейшем случае алгоритм состоит в том, чтобы добавить все элементы массива в двоичную кучу, а затем последовательно извлекать из кучи минимальный/максимальный элемент и записывать обратно в массив.

В выходной файл нужно вывести ровно N четырёхбайтовых целых чисел: содержимое массива A после сортировки.

Пример

input.bin
0A 00 00 00 1F 00 00 00 F2 FF FF FF 06 00 00 00 04 00 00 00 26 00 00 00 FD FF FF FF 1E 00 00 00 F6 FF FF FF 0A 00 00 00 F4 FF FF FF
output.bin
F2 FF FF FF F4 FF FF FF F6 FF FF FF FD FF FF FF 04 00 00 00 06 00 00 00 0A 00 00 00 1E 00 00 00 1F 00 00 00 26 00 00 00

Задача 3. Максимум в окне

Источник:	Повышенной сложности
Имя входного файла:	`input.txt`
Имя выходного файла:	`output.txt`
Ограничение по времени:	1 секунда*
Ограничение по памяти:	разумное

Дан массив A, в котором записано N целых чисел. По этому массиву перемещается окно. Окно — это подотрезок в массиве, начинающийся с L-ого элемента массива, и заканчивающийся на (R-1)-ом элементе (всего в окне $R - L$ элементов). При этом счётчики L и R постоянно изменяются. Нужно после каждого изменения счётчиков найти и вывести максимальное число в окне.

Изначально оба счётчика равны 0. Далее нужно выполнить $2N - 1$ операций, каждая операция имеет один из двух типов:

- `L` — увеличить счётчик L на 1, тем самым сдвинув начало окна.
- `R` — увеличить счётчик R на 1, тем самым сдвинув конец окна.

После выполнения каждой операции нужно вывести максимум среди чисел в окне, то есть максимум среди элементов массива, индекс которых попадает в диапазон $[L, R)$.

Формат входных данных

В первой строке содержится целое число N — количество элементов в массиве ($1 \leqslant N \leqslant 2 \cdot 10^5$). Во второй строке записано N целых чисел через пробел — содержимое массива. Все числа по абсолютной величине не превышают 10^9. В третьей и последней строке записано подряд $(2N - 1)$ символов — команды, которые требуется выполнить.

Гарантируется, что:

1. первая команда имеет тип `R`,
2. после выполнения каждой команды верно $R > L$,
3. окно всегда входит в массив, то есть $R \leqslant N$.

Формат выходных данных

Нужно вывести $2N - 1$ строк, в каждой из которых требуется записать максимум в текущем окне. Максимум нужно выводить после обработки каждой команды.

Пример

input.txt	output.txt
14 1 8 3 2 5 2 7 3 7 4 9 1 3 2 RRRRLLRLRRLLRRLLLRRRRLRLLLL	1 8 8 8 8 3 5 5 5 7 7 7 7 7 7 7 7 7 7 9 9 9 9 9 9 3 3 2