Задача A. Хипуй!

Имя входного файла:	`heap.in`
Имя выходного файла:	`heap.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

В этой задаче вам необходимо организовать структуру данных Heap для хранения целых чисел, над которой определены следующие операции:

- `Insert(X)` — добавить в Heap число X;
- `Extract` — достать из Heap наибольшее число (удалив его при этом).

Формат входного файла

Во входном файле записано количество команд N ($1 \leqslant N \leqslant 100\,000$), потом последовательность из N команд, каждая в своей строке.

Каждая команда имеет такой формат: „`0 <число>`“ или „`1`“, что означает соответственно операции `Insert(<число>)` и `Extract`. Добавляемые числа находятся в интервале от 1 до 10^7 включительно.

Гарантируется, что при выполнении команды `Extract` в структуре находится по крайней мере один элемент.

Формат выходного файла

В выходной файл для каждой команды извлечения необходимо вывести число, полученное при выполнении команды `Extract`.

Примеры

heap.in	heap.out
7 0 100 0 10 1 0 5 0 30 0 50 1	100 50

Задача B. Минимум на отрезке

Имя входного файла:	`min.in`
Имя выходного файла:	`min.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Рассмотрим последовательность целых чисел длины N. По ней с шагом 1 двигается «окно» длины K, то есть сначала в «окне» видно первых K чисел, на следующем шаге в «окне» уже будут находиться K чисел, начиная со второго, и так далее до конца последовательности. Требуется для каждого положения «окна» определить минимум в нём.

Формат входного файла

В первой строке входного файла содержатся два числа N и K ($1 \leq K \leq N \leq 300000$) — длины последовательность и «окна» соответственно. На следующей строке находятся N чисел — сама последовательность.

Формат выходного файла

Выходной файл должен содержать $N-K+1$ строк — минимумы для каждого положения «окна».

Примеры

min.in	min.out
7 3 1 3 2 4 5 3 1	1 2 2 3 1