

Задача А. В начало строя!

Имя входного файла: `movetofront.in`
Имя выходного файла: `movetofront.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Капрал Питуца любит командовать своим отрядом. Его любимый приказ «в начало строя». Он выстраивает свой отряд в шеренгу и оглашает последовательность приказов. Каждая приказ имеет вид «Солдаты с l_i по r_i — в начало строя!»

Пронумеруем солдат в начальном положении с 1 до n , слева направо. Приказ «Солдаты с l_i по r_i — в начало строя!» означает, что солдаты, стоящие с l_i по r_i включительно, перемещаются в начало строя, сохраняя относительный порядок.

Например, если в некоторый момент солдаты стоят в порядке 2, 3, 6, 1, 5, 4, после приказа: «Солдаты с 2 по 4 — в начало строя!» порядок будет 3, 6, 1, 2, 5, 4.

По данной последовательности приказов найти конечный порядок солдат в строю.

Формат входного файла

В первой строке два целых числа n and m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 100\,000$) — количество солдат и количество приказов. Следующие m строк содержат по два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$).

Формат выходного файла

Выведите n целых чисел — порядок солдат в конечном положении после выполнения всех приказов.

Пример

<code>movetofront.in</code>	<code>movetofront.out</code>
6 3 2 4 3 5 2 2	1 4 5 2 3 6

Задача В. Декартово дерево

Имя входного файла: `tree.in`
Имя выходного файла: `tree.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам даны пары чисел (a_i, b_i) , Вам необходимо построить декартово дерево, такое что i -ая вершина имеет ключи (a_i, b_i) , вершины с ключом a_i образуют бинарное дерево поиска, а вершины с ключом b_i образуют кучу.

Формат входного файла

В первой строке записано число N — количество пар. Далее следует N ($1 \leq N \leq 50\,000$) пар (a_i, b_i) . Для всех пар $|a_i|, |b_i| \leq 30\,000$. $a_i \neq a_j$ и $b_i \neq b_j$ для всех $i \neq j$.

Формат выходного файла

Если декартово дерево с таким набором ключей построить возможно, выведите в первой строке YES, в противном случае выведите NO. В случае ответа YES, выведите N строк, каждая из которых должна описывать вершину. Описание вершины состоит из трёх чисел: номер предка, номер левого сына и номер правого сына. Если у вершины отсутствует предок или какой-либо из сыновей, то выводите на его месте число 0.

Если подходящих деревьев несколько, выведите любое.

Пример

tree.in	tree.out
7	YES
5 4	2 3 6
2 2	0 5 1
3 9	1 0 7
0 5	5 0 0
1 3	2 4 0
6 6	1 0 0
4 11	3 0 0

Задача С. Следующий

Имя входного файла: `next.in`
Имя выходного файла: `next.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Реализуйте структуру данных, которая поддерживает множество S целых чисел, с которым разрешается производить следующие операции:

- $add(i)$ — добавить в множество S число i (если он там уже есть, то множество не меняется);
- $next(i)$ — вывести минимальный элемент множества, не меньший i . Если искомый элемент в структуре отсутствует, необходимо вывести -1.

Формат входного файла

Исходно множество S пусто. Первая строка входного файла содержит n — количество операций ($1 \leq n \leq 300\,000$). Следующие n строк содержат операции. Каждая операция имеет вид либо «+ i », либо «? i ». Операция «? i » задает запрос $next(i)$.

Если операция «+ i » идет во входном файле в начале или после другой операции «+», то она задает операцию $add(i)$. Если же она идет после запроса «?», и результат этого запроса был y , то выполняется операция $add((i + y) \bmod 10^9)$.

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до 10^9 .

Формат выходного файла

Для каждого запроса выведите одно число — ответ на запрос.

Пример

next.in	next.out
6	3
+ 1	4
+ 3	
+ 3	
? 2	
+ 1	
? 4	

Задача D. Вставка ключевых значений

Имя входного файла: `key.in`
Имя выходного файла: `key.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вас наняла на работу компания MacroHard, чтобы вы разработали новую структуру данных для хранения целых ключевых значений.

Эта структура выглядит как массив A бесконечной длины, ячейки которого нумеруются с единицы. Изначально все ячейки пусты. Единственная операция, которую необходимо поддерживать — это операция $Insert(L, K)$, где L — положение в массиве, а K — некоторое положительное целое ключевое значение.

Операция выполняется следующим образом:

- Если ячейка $A[L]$ пуста, то присвоить $A[L] := K$.
- Если ячейка $A[L]$ непуста, выполнить $Insert(L + 1, A[L])$, а затем присвоить $A[L] := K$.

По заданной последовательности из N целых чисел L_1, L_2, \dots, L_N вам необходимо вывести содержимое этого массива после выполнения следующей последовательности операций:

$Insert(L_1, 1)$
 $Insert(L_2, 2)$
 \dots
 $Insert(L_N, N)$

Формат входного файла

В первой строке входного файла содержится N — число операций $Insert$ и M — максимальный номер позиции, которую можно использовать в операции $Insert$. ($1 \leq N \leq 131\,072$, $1 \leq M \leq 131\,072$).

В следующей строке даны N целых чисел L_i , которые описывают операции $Insert$ ($1 \leq L_i \leq M$).

Формат выходного файла

Выведите содержимое массива после выполнения данной последовательности операций $Insert$. На первой строке выведите W — номер последней несвободной позиции в массиве. Далее выведите W целых чисел — $A[1], A[2], \dots, A[W]$. Для пустых ячеек выводите нули.

Пример

key.in	key.out
5 4	6
3 3 4 1 3	4 0 5 2 3 1