

Задача А. Минимум на отрезке

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Рассмотрим последовательность целых чисел длины N . По ней с шагом 1 двигается «окно» длины K , то есть сначала в «окне» видно первые 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 входных данных

В первой строке входных данных содержатся два числа N и K ($1 \leq N \leq 150000$, $1 \leq K \leq 10000$, $K \leq N$) – длины последовательности и «окна», соответственно. На следующей строке находятся N чисел – сама последовательность. Числа последовательности не превосходят по модулю 10^5 .

Формат выходных данных

Выходные данные должны содержать $N - K + 1$ строк – минимумы для каждого положения «окна».

Пример

стандартный ввод	стандартный вывод
7 3 1 3 2 4 5 3 1	1 2 2 3 1

Задача В. Множества

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.75 секунд
Ограничение по памяти:	256 мегабайт

На вступительном констесте в пилотную группу по программированию Вашему другу предложили реализовать структуру данных для хранения множеств чисел. Так как он специализируется на истории литературы, данную структуру придётся реализовать Вам.

Структура должна хранить $m + 1$ множеств чисел от 0 до n , пронумерованных от 0 до m включительно, при этом одно число может принадлежать сразу нескольким множествам. Изначально все множества пустые.

Вы должны реализовать следующие операции на этой структуре:

1. ADD e s

Добавить в множество N^s ($0 \leq s \leq m$) число e ($0 \leq e \leq n$).

2. DELETE e s

Удалить из множества N^s ($0 \leq s \leq m$) число e ($0 \leq e \leq n$). Гарантируется, что до этого число e было помещено в множество

3. CLEAR s

Очистить множество N^s ($0 \leq s \leq m$).

4. LISTSET s

Показать содержимое множества N^s ($0 \leq s \leq m$) в **возрастающем порядке**, либо -1 , если множество пусто.

5. LISTSETSOF e

Вывести номера множеств, в которых лежит число e ($0 \leq e \leq n$) в **возрастающем порядке**, либо -1 , если этого числа нет ни в одном множестве.

Формат входных данных

Сначала вводятся числа N ($1 \leq N \leq 9223372036854775807$), M ($1 \leq M \leq 100000$) и K ($0 \leq K \leq 100000$) — максимальное число, номер максимального множества и количество запросов к структуре данных. Далее следуют K строк указанного формата запросов.

Формат выходных данных

На каждый запрос LISTSET Ваша программа должна вывести числа — содержимое запрошенного множества или -1 , если множество пусто.

На каждый запрос LISTSETSOF Ваша программа должна вывести числа — номера множеств, содержащих запрошенное число, или -1 , если таких множеств не существует.

На прочие запросы не должно быть никакого вывода.

Гарантируется, что правильный вывод программы не превышает одного мегабайта.

Пример

стандартный ввод	стандартный вывод
10 10	1 2
9	1 2
ADD 1 1	2
ADD 1 2	-1
ADD 2 1	
LISTSET 1	
LISTSETSOFF 1	
DELETE 1 1	
LISTSET 1	
CLEAR 1	
LISTSET 1	

Замечание

Эту задачу можно (и нужно!) решать, используя `std::set` и `std::map`.

Задача С. Гоблины и очереди

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.6 секунд
Ограничение по памяти:	256 мегабайт

Гоблины Мглистых гор очень любят ходить к своим шаманам. Так как гоблинов много, к шаманам часто образуются очень длинные очереди. А поскольку много гоблинов в одном месте быстро образуют шумную толку, которая мешает шаманам проводить сложные медицинские манипуляции, последние решили установить некоторые правила касательно порядка в очереди.

Обычные гоблины при посещении шаманов должны вставать в конец очереди. Привилегированные же гоблины, знающие особый пароль, встают ровно в ее середину, причем при нечетной длине очереди они встают сразу за центром.

Так как гоблины также широко известны своим непочтительным отношением ко всяческим правилам и законам, шаманы попросили вас написать программу, которая бы отслеживала порядок гоблинов в очереди.

Формат входных данных

В первой строке входных данных записано число N ($1 \leq N \leq 10^5$) – количество запросов. Следующие N строк содержат описание запросов в формате:

- $+ i$ – гоблин с номером i ($1 \leq i \leq N$) встает в конец очереди.
- $* i$ – привилегированный гоблин с номером i встает в середину очереди.
- $-$ – первый гоблин из очереди уходит к шаманам. Гарантируется, что на момент такого запроса очередь не пуста.

Формат выходных данных

Для каждого запроса типа $-$ программа должна вывести номер гоблина, который должен зайти к шаманам.

Примеры

стандартный ввод	стандартный вывод
7 + 1 + 2 - + 3 + 4 - -	1 2 3
2 * 1 + 2	

Задача D. Машины

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Петя, которому три года, очень любит играть с машинками. Всего у Пети N различных машинок, которые хранятся на полке шкафа так высоко, что он сам не может до них дотянуться. Одновременно на полу комнаты может находиться не более K машинок. Петя играет с одной из машинок на полу и если он хочет поиграть с другой машинкой, которая также находится на полу, то дотягивается до нее сам. Если же машинка находится на полке, то он обращается за помощью к маме. Мама может достать для Пети машинку с полки и одновременно с этим поставить на полку любую машинку с пола. Мама очень хорошо знает своего ребенка и может предугадать последовательность, в которой Петя захочет играть с машинками. При этом, чтобы не мешать Петиней игре, она хочет совершить как можно меньше операций по подъему машинки с пола, каждый раз правильно выбирая машинку, которую следует убрать на полку. Ваша задача состоит в том, чтобы определить минимальное количество операций. Перед тем, как Петя начал играть, все машинки стоят на полке.

Формат входных данных

В первой строке содержатся три числа N , K и P ($1 \leq K, N \leq 100000$, $1 \leq P \leq 500000$). В следующих P строках записаны номера машинок в том порядке, в котором Петя захочет играть с ними.

Формат выходных данных

Выведите единственное число: минимальное количество операций, которое надо совершить Петиней маме.

Пример

стандартный ввод	стандартный вывод
3 2 7 1 2 3 1 3 1 2	4

Замечание

В этой задаче можно использовать STL.

Пояснения к примеру:

Операция 1: снять машинку 1

Операция 2: снять машинку 2

Операция 3: поднять машинку 2 и снять машинку 3

Операция 4: поднять машинку 3 или 1 и снять машинку 2

Задача Е. Высокие горы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В мире существует огромное количество горных систем. Самые высокие из них, как правило, образовались в течение последних 50 млн. лет и называются *молодыми*. Ученые собрали статистику о высоте гор в виде неубывающего массива чисел — a_1, a_2, \dots, a_n , означающих высоты гор в сантиметрах.

Оказалось, что можно предсказать высоты гор в будущем. Для каждого года j в течение q последующих лет ученые предположили, что k_j самых высоких гор вырастут на x_j сантиметров.

Ученые предоставили вам необходимую информацию и попросили вас узнать, какими будут высоты гор через q лет. Напишите программу, которая поможет найти искомые высоты.

Формат входных данных

На вход в первой строке подается натуральное число n ($1 \leq n \leq 10^5$) — количество исследуемых гор. Во второй строке идут n чисел — a_i ($1 \leq a_i \leq 10^9$) — текущие высоты гор в сантиметрах.

В третьей строке находится число q ($1 \leq q \leq 10^5$) — количество последующих лет, в которые происходили изменения. Каждая из следующих q строк состоит из 2 чисел — k_j и x_j ($1 \leq k_j \leq n$, $1 \leq x_j \leq 10^4$), где k_j — количество самых высоких гор, которые вырастут в j -й год, а x_j — высота, на которую они увеличатся.

Формат выходных данных

Нужно вывести n чисел — высоты гор через q лет в сантиметрах в порядке неубывания чисел.

Пример

стандартный ввод	стандартный вывод
3 1000 2000 3000	1000 4000 8000
2 2 2000	
1 3000	

Замечание

В первом примере всего 3 горы высотой 1000, 2000 и 3000 сантиметров. В следующем году высоты второй и третьей гор увеличатся на 2000 сантиметров, а через год — вырастет только третья гора на 3000 сантиметров

Задача F. Менеджер памяти-1

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Пете поручили написать менеджер памяти для новой стандартной библиотеки языка $\varphi++$. В распоряжении у менеджера находится массив из N последовательных ячеек памяти, пронумерованных от 1 до N . Задача менеджера – обрабатывать запросы приложений на выделение и освобождение памяти. Запрос на выделение памяти имеет один параметр K . Такой запрос означает, что приложение просит выделить ему K последовательных ячеек памяти. Если в распоряжении менеджера есть хотя бы один свободный блок из K последовательных ячеек, то он обязан в ответ на запрос выделить такой блок. При этом непосредственно перед самой первой ячейкой памяти выделяемого блока не должно располагаться свободной ячейки памяти. После этого выделенные ячейки становятся занятыми и не могут быть использованы для выделения памяти, пока не будут освобождены. Если блока из K последовательных свободных ячеек нет, то запрос отклоняется. Запрос на освобождение памяти имеет один параметр T . Такой запрос означает, что менеджер должен освободить память, выделенную ранее при обработке запроса с порядковым номером T . Запросы нумеруются, начиная с единицы. Гарантируется, что запрос с номером T – запрос на выделение, причем к нему еще не применялось освобождение памяти. Освобожденные ячейки могут снова быть использованы для выделения памяти. Если запрос с номером T был отклонен, то текущий запрос на освобождение памяти игнорируется. Требуется написать менеджер памяти, удовлетворяющий приведенным критериям.

Формат входных данных

Первая строка входного файла содержит числа N и M – количество ячеек памяти и количество запросов соответственно ($1 \leq N \leq 2^{31} - 1$; $1 \leq M \leq 10^5$). Каждая из следующих M строк содержит по одному числу: $(i+1)$ -я строка входного файла ($1 \leq i \leq M$) содержит либо положительное число K , если i -й запрос – запрос на выделение с параметром K ($1 \leq K \leq N$), либо отрицательное число $-T$, если i -й запрос – запрос на освобождение с параметром T ($1 \leq T < i$).

Формат выходных данных

Для каждого запроса на выделение памяти выведите в выходной файл результат обработки этого запроса: для успешных запросов выведите номер первой ячейки памяти в выделенном блоке, для отклоненных запросов выведите число -1 . Результаты нужно выводить в порядке следования запросов во входном файле.

Примеры

стандартный ввод	стандартный вывод
42 9 7 3 8 -2 6 5 -5 9 4	1 8 11 19 25 30 19
128 12 1 2 4 -2 8 -3 16 -5 32 -7 64 -1	1 2 4 8 16 32 64