

Задача А. И снова сумма...

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
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);
- $sum(l, r)$ — вывести сумму всех элементов x из S , которые удовлетворяют неравенству $l \leq x \leq r$.

Формат входных данных

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 «? l r ». Операция «? l r » задает запрос $sum(l, r)$.

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 выходных данных

Для каждого запроса выведите одно число — ответ на запрос.

Пример

стандартный ввод	стандартный вывод
6	3
+ 1	7
+ 3	
+ 3	
? 2 4	
+ 1	
? 2 4	

Задача В. К-ый максимум

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	64 мегабайта

Напишите программу, реализующую структуру данных, позволяющую добавлять и удалять элементы, а также находить k -й максимум.

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество команд ($n \leq 100\,000$). Последующие n строк содержат по одной команде каждая. Команда записывается в виде двух чисел c_i и k_i — тип и аргумент команды соответственно ($|k_i| \leq 10^9$).

Поддерживаемые команды:

- 1 (или просто 1): Добавить элемент с ключом k_i .
- 0: Найти и вывести k_i -й максимум.
- -1: Удалить элемент с ключом k_i .

Гарантируется, что в процессе работы в структуре не требуется хранить элементы с равными ключами или удалять несуществующие элементы. Также гарантируется, что при запросе k_i -го максимума, он существует.

Формат выходных данных

Для каждой команды нулевого типа в выходной файл должна быть выведена строка, содержащая единственное число — k_i -й максимум.

Пример

стандартный ввод	стандартный вывод
11	7
1 5	5
1 3	3
1 7	10
0 1	7
0 2	3
0 3	
-1 5	
1 10	
0 1	
0 2	
0 3	

Задача С. Следующий

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

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

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

Формат входных данных

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) mod 109)`.

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

Формат выходных данных

Для каждого запроса выведите одно число – ответ на запрос.

Пример

стандартный ввод	стандартный вывод
6	3
+ 1	4
+ 3	
+ 3	
? 2	
+ 1	
? 4	

Замечание

В этой задаче запрещено использовать STL.

Задача D. НОД 2010

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

Вам предложили работу в секретном проекте Агентства Федеральной Безопасности под кодовым названием «НОД 2010». Основным объектом исследования является набор целых положительных чисел. Вы должны понять, как будет изменяться наибольший общий делитель всех чисел этого набора при добавлении в него новых чисел или удалении лежащих там чисел. В начале эксперимента набор чисел пуст.

Формат входных данных

В первой строке записано целое число q ($1 \leq q \leq 10^5$) — количество операций с набором.

Каждая из следующих q строк имеет вид «+ x » или «- x ». В первом случае число x добавляется в набор, а во втором случае — удаляется из него. Число x целое, положительное и не превосходит 10^9 . Гарантируется, что из набора будут удаляться только числа, которые в нём лежат.

Формат выходных данных

Выведите наибольший общий делитель всех чисел набора после каждой описанной операции. Согласно распоряжению 190p, наибольшим общим делителем пустого набора является единица.

Пример

стандартный ввод	стандартный вывод
5	8
+ 8	2
+ 6	2
+ 8	2
- 8	6
- 8	

Задача Е. Двоичное дерево поиска

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Реализуйте сбалансированное двоичное дерево поиска.

Формат входных данных

Входной файл содержит описание операций с деревом, их количество не превышает 100000. В каждой строке находится одна из следующих операций:

- **insert** x — добавить в дерево ключ x . Если ключ x в дереве уже есть, то ничего делать не надо.
- **delete** x — удалить из дерева ключ x . Если ключа x в дереве нет, то ничего делать не надо.
- **exists** x — если ключ x есть в дереве, выведите «true», иначе «false»
- **next** x — выведите минимальный элемент в дереве, строго больший x , или «none», если такого нет.
- **prev** x — выведите максимальный элемент в дереве, строго меньший x , или «none», если такого нет.

Все числа во входном файле целые и по модулю не превышают 10^9 .

Формат выходных данных

Выведите последовательно результат выполнения всех операций **exists**, **next**, **prev**. Следуйте формату выходного файла из примера.

Пример

стандартный ввод	стандартный вывод
insert 2	true
insert 5	false
insert 3	5
exists 2	3
exists 4	none
next 4	3
prev 4	
delete 5	
next 4	
prev 4	

Задача F. Декартовы деревья

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

Недавно на лекции в университете Вася узнал, что такое декартово дерево. Декартово дерево — это двоичное дерево, в каждом узле которого хранятся два значения: ключ и приоритет. Оно является деревом поиска по множеству ключей и кучей на максимум по приоритетам, то есть

1. ключ любой вершины в левом поддереве вершины v меньше, чем ключ вершины v ;
2. ключ любой вершины в правом поддереве вершины v больше, чем ключ вершины v ;
3. приоритеты сыновей вершины v не больше, чем приоритет самой вершины v .

На контрольной работе Васе досталась следующая задача: дано n пар чисел вида (ключ, значение), i -я из которых выглядит как (i, y_i) , и нужно узнать, сколько существует способов построить декартово дерево, используя как ключ вершины i число i , а как приоритет — y_i . Поскольку это число может оказаться достаточно большим, требуется найти остаток от его деления на число $10^9 + 7$.

Формат входных данных

Первая строка содержит одно натуральное число t — число тестовых примеров во входных данных. Далее следуют описания тестов. Описание каждого теста состоит из двух строк. Первая строка содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество вершин в дереве. Вторая строка содержит n целых чисел y_i ($1 \leq y_i \leq 10^9$) — приоритет i -й вершины дерева. Сумма n по всем тестам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого теста в отдельной строке выведите одно целое число — количество различных декартовых деревьев, которые можно построить на данном наборе приоритетов, по модулю $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
2	1
4	10
2 4 1 3	
6	
7 3 3 1 1 3	

Задача G. Ксюша и загруженное множество

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Ксюша решила основать компанию по производству игр. Чтобы выделиться среди конкурентов и добиться успеха, она решила написать свой игровой движок. Движок должен поддерживать множество, изначально состоящее из n различных целых чисел a_1, a_2, \dots, a_n .

К множеству будут последовательно применяться m операций. Операции бывают следующих типов:

- Вставить элемент x в множество;
- Удалить элемент x из множества;
- Сообщить k -загруженность множества.

k -загруженностью множества называется такое **минимальное** целое **положительное** число d , что числа $d, d + 1, \dots, d + (k - 1)$ не встречаются в этом множестве. Например, 3-загруженность множества $\{3, 4, 6, 11\}$ равна 7, так как числа 7, 8, 9 отсутствуют в множестве, а никакое меньшее значение не подходит.

Ксюша занята менеджерскими делами, поэтому движок придётся писать вам. Реализуйте эффективную поддержку описанных операций.

Формат входных данных

В первой строке дано целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Далее следуют описания наборов.

В первой строке дано целое число n ($1 \leq n \leq 2 \cdot 10^5$) — начальный размер множества.

Во второй строке даны n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_1 < a_2 < \dots < a_n \leq 2 \cdot 10^6$) — начальное состояние множества.

В третьей строке дано целое число m ($1 \leq m \leq 2 \cdot 10^5$) — количество операций.

В следующих m строках даны операции. Операции даны в следующем формате:

- $+ x$ ($1 \leq x \leq 2 \cdot 10^6$) — вставить элемент x в множество (гарантируется, что x отсутствует в множестве);
- $- x$ ($1 \leq x \leq 2 \cdot 10^6$) — удалить элемент x из множества (гарантируется, что x присутствует в множестве);
- $? k$ ($1 \leq k \leq 2 \cdot 10^6$) — вывести значение k -загруженности множества.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$, то же самое гарантируется для m .

Формат выходных данных

Для каждого набора входных данных выведите ответы на операции типа «?».

Пример

стандартный ввод	стандартный вывод
3	2 2 1 6 3 8 8 2000001
5	9 9 9 7
1 2 5 905 2000000	1 1
15	
- 2	
? 2	
? 1	
- 1	
? 1	
+ 4	
+ 2	
? 2	
+ 6	
- 4	
+ 7	
? 2	
? 3	
? 4	
? 2000000	
5	
3 4 5 6 8	
9	
? 5	
- 5	
? 5	
+ 1	
? 2	
- 6	
- 8	
+ 6	
? 5	
5	
6 7 8 9 10	
10	
? 5	
- 6	
? 4	
- 10	
+ 5	
- 8	
+ 3	
+ 2	
- 3	
+ 10	