

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 пусты.

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

Показать множества, в которых лежит число e ($0 \leq e \leq n$), 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. Прибавляем, суммируем

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Есть массив целых чисел длины $n = 2^{24}$, изначально заполненных нулями. Вам нужно сперва обработать m случайных запросов вида “прибавление на отрезке”. Затем обработать q случайных запросов вида “сумма на отрезке”.

Формат входных данных

На первой строке числа m, q ($1 \leq m, q \leq 2^{24}$). На второй строке пара целых чисел a, b от 1 до 10^9 , используемая в генераторе случайных чисел.

```
0. unsigned int a, b; // даны во входных данных
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до  $2^{24} - 1$ .
5. }
```

Каждый запрос первого вида генерируется следующим образом:

```
1. add = nextRand(); // число, которое нужно прибавить
2. l = nextRand();
3. r = nextRand();
4. if (l > r) swap(l, r); // получили отрезок [l..r]
```

Каждый запрос второго вида генерируется следующим образом:

```
1. l = nextRand();
2. r = nextRand();
3. if (l > r) swap(l, r); // получили отрезок [l..r]
```

Сперва генерируются запросы первого вида, затем второго.

Формат выходных данных

Выведите сумму ответов на все запросы второго типа по модулю 2^{32} .

Примеры

стандартный ввод	стандартный вывод
5 5 13 239	811747796
10 10 239017 170239	3460675938

Задача F. 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 невозрастания чисел.

Система оценки

Решения, корректно работающие при ограничении $n \cdot q \leq 10^8$, будут оцениваться исходя из 60 баллов.

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 G. Гистограмма

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.8 секунд
Ограничение по памяти:	256 мегабайт

Гистограмма является многоугольником, сформированным из последовательности прямоугольников, выровненных на общей базовой линии. Прямоугольники имеют равную ширину, но могут иметь различные высоты. Обычно гистограммы используются для представления дискретных распределений, например, частоты символов в текстах. Отметьте, что порядок прямоугольников очень важен. Вычислите область самого большого прямоугольника в гистограмме, который также находится на общей базовой линии.

Формат входных данных

В первой строке входного файла записано число N ($0 < N \leq 10^6$) – количество прямоугольников гистограммы. Затем следует N целых чисел $h_1 \dots h_n$, где $0 \leq h_i \leq 10^9$. Эти числа обозначают высоты прямоугольников гистограммы слева направо. Ширина каждого прямоугольника равна 1.

Формат выходных данных

Выведите площадь самого большого прямоугольника в гистограмме. Помните, что этот прямоугольник должен быть на общей базовой линии.

Примеры

стандартный ввод	стандартный вывод
7 2 1 4 5 1 3 3	8
3 2 1 2	3
1 0	0

Задача Н. Хорошие дни

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

Билл разрабатывает новую математическую теорию, описывающую человеческие эмоции. Его последние исследования посвящены изучению того, насколько хорошие и плохие дни влияют на воспоминания людей о различных периодах жизни.

Недавно Билл придумал методику, которая описывает, насколько хорошим или плохим был день человеческой жизни с помощью сопоставления дню некоторого неотрицательного целого числа. Билл называет это число эмоциональной значимостью этого дня. Чем больше это число, тем лучше этот день. Билл полагает, что значимость некоторого периода человеческой жизни равна сумме эмоциональных значимостей каждого из дней периода, помноженной на минимум эмоциональных значимостей дней этого периода. Эта методика отражает то, что период, который в среднем может быть весьма неплох, бывает испорчен одним плохим днем.

Теперь Билл хочет проанализировать свою собственную жизнь и найти в ней период максимальной значимости. Помогите ему это сделать.

Формат входных данных

Первая строка входного файла содержит число n — количество дней в жизни Билла, которые он хочет исследовать ($1 \leq n \leq 100\,000$). Оставшаяся часть файла содержит n целых чисел a_1, a_2, \dots, a_n , все в пределах от 0 до 10^6 — эмоциональные значимости дней. Числа во входном файле разделяются пробелами и переводами строки.

Формат выходных данных

В первой строке выходного файла выведите максимальную значимость периода жизни Билла.

Примеры

стандартный ввод	стандартный вывод
6 3 1 6 4 5 2	60
4 1 2 1 2	6
3 2 1 2	5

Задача I. 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