

Задача А. Хип ли?

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

Структуру данных Heap можно реализовать на основе массива.

Для этого должно выполняться *основное свойство Heap'a*, которое заключается в следующем. Для каждого $1 \leq i \leq n$ выполняются следующие условия:

- Если $2i \leq n$, то $a[i] \leq a[2i]$
- Если $2i + 1 \leq n$, то $a[i] \leq a[2i + 1]$

Дан массив целых чисел. Определите является ли он Heap'ом.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$). Вторая строка содержит n целых чисел по модулю не превосходящих $2 \cdot 10^9$.

Формат выходных данных

Выведите «YES», если массив является Heap'ом и «NO» в противном случае.

Примеры

<code>isheap.in</code>	<code>isheap.out</code>
5 1 0 1 2 0	NO
5 1 3 2 5 4	YES

Задача В. Хипуй!

Имя входного файла: `heap.in`
Имя выходного файла: `heap.out`
Ограничение по времени: 4 секунды
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 входных данных

Во входном файле записано количество команд N ($1 \leq N \leq 100\,000$), 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 выходных данных

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

Примеры

<code>heap.in</code>	<code>heap.out</code>
7	100
0 100	50
0 10	
1	
0 5	
0 30	
0 50	
1	

Задача С. Ярый коллекционер бабочек

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	64 мегабайта

Как известно, Андрей Сергеевич — ярый коллекционер бабочек. Он имеет огромную коллекцию, экспонаты которой собраны со всего мира. Будем считать, что в мире существует 2 000 000 000 видов бабочек.

Чтобы не запутаться, Андрей Сергеевич присвоил каждому виду уникальный номер. Нумерация видов бабочек начинается с единицы.

Теперь он хочет знать, есть ли бабочка с видом K в его коллекции, или же её придётся добывать, затрачивая уйму сил и денег.

Формат входных данных

В первой строке входного файла содержится единственное число N ($1 \leq N \leq 100\,000$) — количество видов бабочек в коллекции Андрея Сергеевича.

В следующей строке через пробел находятся N упорядоченных по возрастанию чисел — номера видов бабочек в коллекции.

Все виды бабочек в коллекции имеют различные номера.

В третьей строке файла записано число M ($1 \leq M \leq 100\,000$) — количество видов бабочек, про которых Андрей Сергеевич хочет узнать, есть ли они у него в коллекции или же нет. В последней строке входного файла содержатся через пробел M чисел — номера видов бабочек, наличие которых необходимо проверить.

Формат выходных данных

Выходной файл должен содержать M строчек. Для каждого запроса выведите “YES”, если бабочка с данным номером содержится в коллекции, и “NO” — в противном случае.

Примеры

стандартный ввод	стандартный вывод
7	YES
23 27 36 45 54 72 98	YES
4	YES
54 36 23 45	YES

Задача D. Шарики

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Организаторы детского праздника планируют надуть для него M воздушных шариков. С этой целью они пригласили N добровольных помощников, i -й среди которых надувает шарик за T_i минут, однако каждый раз после надувания Z_i шариков устает и отдыхает Y_i минут. Теперь организаторы праздника хотят узнать, через какое время будут надуты все шарики при наиболее оптимальной работе помощников, и сколько шариков надует каждый из них. (Если помощник надул шарик, и должен отдохнуть, но больше шариков ему надувать не придется, то считается, что он закончил работу сразу после окончания надувания последнего шарика, а не после отдыха).

Формат входных данных

В первой строке входных данных задаются числа M и N ($0 \leq M \leq 15000$, $1 \leq N \leq 1000$). Следующие N строк содержат по три целых числа — T_i , Z_i и Y_i соответственно ($1 \leq T_i \leq 100$, $1 \leq Y_i \leq 100$, $1 \leq Z_i \leq 1000$).

Формат выходных данных

В первой строке программа должна вывести значение T — минимальное время, за которое будут надуты все шарики. Во второй строке выведите N чисел — количество шариков, надутых каждым из приглашенных помощников. Разделяйте числа пробелами. Если распределений шариков несколько, выведите любое из них. Необходимо надуть ровно N шариков, нельзя надуть большее число шариков, даже если это занимает такое же время.

Примеры

стандартный ввод	стандартный вывод
10 3 1 2 3 3 10 3 2 4 3	8 4 2 4
1 3 1 1 100 2 1 100 3 1 100	1 1 0 0
1 2 1 1 1 1 1 1	1 1 0

Задача Е. Кружок рисования

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда (python3: 3 секунды)
Ограничение по памяти: 256 мегабайт

В один из дней на клубе рисования в ЛКШ изучалось современное искусство. Картины в направлении «кукарекизма» представляют собой длинную полоску бумаги, разбитую на ячейки, которые покрашены разными цветами. Будем считать, что каждый цвет можно представить как целое неотрицательное число.

Картины в этом жанре принято рисовать следующим образом: выбирается цвет краски и закрашиваются все ячейки от начала полоски до выбранной ячейки с номером d .

Длина полоски 10^9 ячеек, поэтому ходить вдоль картины всем лень. Напишите программу, которая подскажет ещё в процессе рисования, в какой цвет покрашена ячейка.

Изначально все ячейки покрашены в цвет 0.

Формат входных данных

Первая строка входных данных содержит одно число q ($1 \leq q \leq 100\,000$) — количество запросов.

Следующие q строк описывают действия. Каждая строка начинается с числа $type$. Если $type = 1$, то дальше идут ещё 2 числа r и c ($1 \leq r, c \leq 10^9$), это обозначает, что красятся первые r ячеек в цвет c . Если же $type = 2$, то дальше идёт число x , это обозначает, что спрашивается, в какой цвет окрашена ячейка с номером x .

Формат выходных данных

Выведите в отдельных строках ответы на запросы, в какой цвет покрашена очередная ячейка.

Примеры

стандартный ввод	стандартный вывод
3 1 5 1 2 3 2 6	1 0
4 1 5 1 2 3 1 4 2 2 3	1 2

Задача F. Разреженные таблицы

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

Дан массив из n чисел. Требуется написать программу, которая будет отвечать на запросы следующего вида: найти минимум на отрезке между u и v включительно.

Формат входных данных

В первой строке входного файла даны три натуральных числа n , m ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^7$) и a_1 ($0 \leq a_1 < 16\,714\,589$) — количество элементов в массиве, количество запросов и первый элемент массива соответственно. Вторая строка содержит два натуральных числа u_1 и v_1 ($1 \leq u_1, v_1 \leq n$) — первый запрос.

Элементы a_2, a_3, \dots, a_n задаются следующей формулой:

$$a_{i+1} = (23 \cdot a_i + 21563) \bmod 16714589.$$

Например, при $n = 10$, $a_1 = 12345$ получается следующий массив: $a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095)$.

Запросы генерируются следующим образом:

$$u_{i+1} = ((17 \cdot u_i + 751 + ans_i + 2i) \bmod n) + 1,$$
$$v_{i+1} = ((13 \cdot v_i + 593 + ans_i + 5i) \bmod n) + 1,$$

где ans_i — ответ на запрос номер i .

Обратите внимание, что u_i может быть больше, чем v_i .

Формат выходных данных

В выходной файл выведите u_m , v_m и ans_m (последний запрос и ответ на него).

Примеры

<code>sparse.in</code>	<code>sparse.out</code>
10 8 12345 3 9	5 3 1565158

Замечание

Пояснение к тесту из примера: запросы и результаты.

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
12345	305498	7048017	11694653	1565158	2591019	9471233	570265	13137658	1325095

#	u	v	ans
1	3	9	570265
2	10	1	12345
3	1	2	12345
4	10	10	1325095
5	5	9	570265
6	2	1	12345
7	3	2	305498
8	5	3	1565158

Задача G. Сумма на отрезке им. Фенвика

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

Дан массив из N элементов, нужно научиться находить сумму чисел на отрезке.

Формат входных данных

Первая строка входного файла содержит два целых числа N и K — количество чисел в массиве и количество запросов ($1 \leq N \leq 100\,000$, $0 \leq K \leq 100\,000$). Следующие K строк содержат следующие запросы:

- A i x — присвоить i -му элементу массива значение x ($1 \leq i \leq n$, $0 \leq x \leq 10^9$);
- Q l r — найти сумму чисел в массиве на позициях от l до r ($1 \leq l \leq r \leq n$).

Изначально в массиве живут нули.

Формат выходных данных

На каждый запрос вида Q l r нужно вывести единственное число — сумму на отрезке.

Примеры

<code>sum.in</code>	<code>sum.out</code>
5 9	0
A 2 2	2
A 3 1	1
A 4 2	2
Q 1 1	0
Q 2 2	5
Q 3 3	
Q 4 4	
Q 5 5	
Q 1 5	

Замечание

Сдавайте данную задачу при помощи дерева Фенвика. Остальные решения будут баниться

Задача Н. Друзья и последовательности

Имя входного файла: friends.in
Имя выходного файла: friends.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Майк и !Майк соперничают еще со школьных лет, они противоположны во всем что делают, кроме программирования. Сегодня у них возникла проблема, которую сами друзья сами решить не могут, но вместе с вами — кто знает?

Каждый из них знает две последовательности n чисел a и b . По запросу в виде пары целых чисел (l, r) Майк может сразу сообщить значение $\max_{i=l}^r a_i$, а !Майк — значение $\min_{i=l}^r b_i$.

Предположим, что робот задает им каждый из возможных различных запросов в виде пары целых чисел (l, r) ($1 \leq l \leq r \leq n$) (то есть он сделает ровно $n(n+1)/2$ запросов) и считает, сколько раз их ответы на один и тот же запрос совпадают, то есть для скольких пар выполняется $\max_{i=l}^r a_i = \min_{i=l}^r b_i$.

Сколько случаев совпадения посчитает робот?

Формат входных данных

В первой строке содержится единственное целое число n ($1 \leq n \leq 200\,000$).

Во второй строке содержатся n целых чисел a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) — элементы последовательности a .

В третьей строке содержатся n целых чисел b_1, b_2, \dots, b_n ($-10^9 \leq b_i \leq 10^9$) — элементы последовательности b .

Формат выходных данных

Выведите одно целое число — количество совпадений ответов, которые посчитает робот, то есть для скольких пар выполняется $\max_{i=l}^r a_i = \min_{i=l}^r b_i$.

Примеры

friends.in	friends.out
6 1 2 3 2 1 4 6 7 1 2 3 2	2
3 3 3 3 1 1 1	0

Замечание

Эта задача взята с Codeforces Round 361. Оригинал: <http://codeforces.com/contest/689/problem/D>