

Задача А. Сумма на отрезке

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

sum.in	sum.out
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	

Задача В. Фонари

Имя входного файла:	<code>streetlamps.in</code>
Имя выходного файла:	<code>streetlamps.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В детском лагере «Берендеевы поляны» вряд ли найдётся кто-нибудь, кто не побоялся бы сходить ночью на хутор. В связи с этим администрация лагеря установила вдоль дороги N фонарей. Лампочки в фонарях могут перегорать, и, хотя в «Берендеевых полянах» есть электрик, он может заболеть и не успеть заменить перегоревшую лампочку на новую до приезда очередной комиссии.

Каждая посещающая лагерь комиссия считает своим долгом проинспектировать состояние фонарей, освещающих дорогу на хутор, но дорога эта очень длинная, поэтому каждый раз членов комиссии подвозят на машине к какому-нибудь фонарю, они выходят, идут пешком до другого фонаря, осматривая по пути все фонари, и там садятся в машину и уезжают.

Если они замечают хотя бы один перегоревший фонарь, в «Берендеевых полянах» устраивается штрафное посвящение для преподавателей.

По информации о происходящих событиях для каждого приезда комиссии выведите результат осмотра фонарей.

Формат входных данных

В первой строке входного файла записаны два числа N и k ($1 \leq N, k \leq 100\,000$) — количество фонарей и количество событий. Далее следуют k строк, описывающих события. События описываются следующим образом:

- перегорание фонаря задаётся двумя числами, первое из которых равно -1 , а второе задаёт номер перегоревшего фонаря (от 1 до N);
- замена лампочки электриком задаётся двумя числами, первое из которых равно 1, а второе задаёт номер заменённой лампочки (от 1 до N);
- приезд комиссии задается тремя числами, первое из которых равно 0, а два других (a и b) задают отрезок дороги, по которому проходит комиссия. А именно, члены комиссии видят все фонари с номерами от a до b включительно и только их. Гарантируется, что $1 \leq a \leq b \leq N$.

Изначально все фонари исправны.

Формат выходных данных

Для каждого приезда комиссии выведите в выходной файл одну строку. А именно, если члены комиссии не заметят ни одного перегоревшего фонаря, то выведите «PASSED», иначе — «PENALTY».

Пример

streetlamps.in	streetlamps.out
5 6	PASSED
-1 2	PENALTY
0 1 1	PASSED
0 2 5	
-1 4	
1 2	
0 2 3	

Задача С. Дерево отрезков с операцией на отрезке

Имя входного файла: `segment-tree.in`
 Имя выходного файла: `segment-tree.out`
 Ограничение по времени: 0.5 секунд
 Ограничение по памяти: 64 мегабайта

Реализуйте эффективную структуру данных для хранения элементов и увеличения нескольких подряд идущих элементов на одно и то же число.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leq N \leq 100\,000$) — количество чисел в массиве.

Во второй строке вводятся N чисел от 0 до 100 000 — элементы массива.

В третьей строке вводится одно натуральное число M ($1 \leq M \leq 30\,000$) — количество запросов.

Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (g — получить текущее значение элемента по его номеру, a — увеличить все элементы на отрезке).

Следом за g вводится одно число — номер элемента.

Следом за a вводится три числа — левый и правый концы отрезка и число add , на которое нужно увеличить все элементы данного отрезка массива ($1 \leq add \leq 100\,000$).

Формат выходных данных

Выведите в одну строку через пробел ответы на каждый запрос g .

Примеры

segment-tree.in	segment-tree.out
5	4
2 4 3 5 2	2
5	14
g 2	5
g 5	
a 1 3 10	
g 2	
g 4	

Задача D. 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:

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

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

Задача Е. Совет стаи

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

— Смотрите хорошенько, смотрите хорошенько, о волки!

Акела

Закон Джунглей очень ясно говорит, что каждый вновь женившийся волк может отделиться от своей стаи; однако едва его волчата вырастут настолько, чтобы хорошо держаться на ногах, он обязан привести их и представить Совету стаи, который обыкновенно собирается в полнолуние; это делается для того, чтобы остальные волки узнали их. После такого осмотра волчата имеют право бегать куда им угодно и пока они не поймают первого оленя.

На совете стаи волки представляют своих детенышей другим волкам, чтобы они могли принять решение об их вступлении в стаю. Совет проходит в полночь, и волчата располагаются вдоль полоски лунного света, чтобы их можно было лучше рассмотреть.

Процесс выбора вступающих в стаю волчат не совсем определен, но зависит от решения каждого из волков. Волк, оценивающий потомство, как правило, не изучает всех прибывших детенышей, но ограничивается лишь некоторыми, стоящими подряд. Также, иногда та или другая волчица выталкивает носом своего детёныша в полосу лунного света, желая, чтобы его непременно заметили. В таком случае какой-то другой волчонок уходит в тень, а новый встает на его место.

Показателем будущих способностей маленького волка является его рост, главным образом зависящий от длины его ног. Взрослые волки внимательно рассматривают волчат и выбирают из них несколько рядом стоящих, по их мнению, наиболее приспособленных для охоты. Приспособленность детеныша волки определяют, как число $h - e$, где h — рост волчонка, а e — некоторая эталонная величина, передающаяся волками из поколения в поколение. Таким образом, среди кандидатов, рассматриваемых им, каждый волк выбирает подряд стоящих, так, чтобы сумма их приспособленностей была максимальна. При этом, каждый волк понимает важность преемственности и выбирает хотя бы одного волчонка, даже если все кандидаты неважные.

Предскажите выбор каждого из волков на совете стаи.

Формат входных данных

В первой строке входного файла записано одно число N ($1 \leq N \leq 100\,000$) — количество волчат-кандидатов, выстроившихся в ряд в начале процесса отбора. Во второй строке записаны N целых чисел x_i ($|x_i| \leq 10^9$) — приспособленности детенышей.

В третьей строке записано одно число M ($0 \leq M \leq 100\,000$) — количество событий, произошедших на совете. В следующих M строках описываются события:

- 1 $a_i x_i$ ($1 \leq a_i \leq N, |x_i| \leq 10^9$) означает, что мать-волчица поставила своего детеныша с приспособленностью x_i на позицию a_i ;
- 2 $l_i r_i$ ($1 \leq l_i \leq r_i \leq N$) означает, что волк производит выбор, просматривая детенышей на позициях $[l_i, r_i]$.

Формат выходных данных

Для каждого события типа 2 выведите ответ — максимальную возможную сумму приспособленностей расположенных подряд волчат на отрезке $[l_i, r_i]$.

Примеры

council.in	council.out
3	6
1 2 3	3
5	3
2 1 3	1
1 2 -1	
2 1 3	
2 2 3	
2 1 2	

Задача F. Художник

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

Не успев дорисовать свой гениальный футуристический шедевр, М. Калевич увлёкся рисованием одномерных чёрно-белых картин. Он пытается найти оптимальное местоположение и количество чёрных участков картины. Для этого он проводит на прямой белые и чёрные отрезки и после каждой из таких операций хочет знать количество чёрных отрезков на получившейся картине и их суммарную длину.

Изначально прямая белая. Ваша задача — написать программу, которая после каждой такой операции выводит в выходной файл интересующие художника данные.

Формат входных данных

В первой строке входного файла содержится общее количество нарисованных отрезков ($1 \leq N \leq 100\,000$). В последующих N строках содержится описание операций. Каждая операция описывается строкой вида $c\ x\ l$, где c — цвет отрезка ('W' для белых отрезков и 'B' для чёрных), а сам отрезок имеет вид $[x; x + l]$, причём координаты обоих концов — целые числа, по модулю не превосходящие 500 000. Длина задаётся положительным целым числом.

Формат выходных данных

После выполнения каждой из операций необходимо вывести в выходной файл на отдельной строке количество чёрных отрезков на картине и их суммарную длину, разделённые одним пробелом.

Примеры

<code>painter.in</code>	<code>painter.out</code>
7	0 0
W 2 3	1 2
B 2 2	1 4
B 4 2	1 4
B 3 2	2 6
B 7 2	3 5
W 3 1	0 0
W 0 10	