

Задача А. Сумма

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

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 — число чисел в массиве и количество запросов. ($1 \leq N \leq 100\,000$), ($0 \leq K \leq 100\,000$). Следующие K строк содержат запросы

1. `A i x` — присвоить i -му элементу массива значение x ($1 \leq i \leq n$, $0 \leq x \leq 10^9$)
2. `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. RMQ

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

Дан массив `a[1..n]`. Требуется написать программу, обрабатывающую два типа запросов.

- Запрос `“max l r”`. Требуется найти максимум в массиве `a` от l -ой ячейки до r -ой включительно.
- Запрос `“add l r v”`. Требуется прибавить значение v к каждой ячейке массива `a` от l -ой до r -ой включительно.

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leq n, q \leq 10^5$) — длина массива и число запросов соответственно. Вторая строка содержит n целых чисел a_1, \dots, a_n ($|a_i| \leq 10^5$), задающих соответствующие значения массива. Следующие q строк содержат запросы.

В зависимости от типа запрос может иметь вид либо `“max l r”`, либо `“add l r v”`. При этом $1 \leq l \leq r \leq 10^5$, $|v| \leq 10^5$.

Формат выходных данных

Для каждого запроса вида `“max l r”` требуется в отдельной строке выдать значение соответствующего максимума.

Примеры

rmq.in	rmq.out
5 3	3
1 2 3 4 -5	7
max 1 3	
add 1 2 5	
max 1 3	

Задача С. Все минимумы

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

Дан массив целых чисел a_1, a_2, \dots, a_n .

Для каждого его подотрезка $[a_L..a_R]$ определим $F(L, R) := \min\{a_L, \dots, a_R\}$.

Найдите

$$\sum_{1 \leq L \leq R \leq n} F(L, R)$$

то есть сумму минимумов всех подотрезков.

Формат входных данных

Первая строка входных данных содержит натуральное число n ($1 \leq n \leq 100\,000$) — размер массива. Во второй строке через пробел заданы элементы массива, все числа целые от -10^6 до 10^6 .

Формат выходных данных

Выведите единственное число — сумму минимумов всех подотрезков массива a .

Примеры

minsum.in	minsum.out
1 5	5
2 -10 1	-19
4 1 2 3 4	20
5 -3 2 -4 1 -5	-52

Задача D. Количество инверсий

Имя входного файла: `invcnt.in`
Имя выходного файла: `invcnt.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Дан массив случайных целых чисел, нужно найти количество инверсий.

Формат входных данных

На первой строке числа n ($1 \leq n \leq 1\,000\,000$) — размер массива и m ($1 \leq m \leq 2^{24}$) — числа в массиве от 0 до $m-1$). На второй строке пара целых чисел a, b от 1 до 10^9 , используемая в генераторе случайных чисел.

```
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand24() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до 224 - 1.
5. }
```

Элементы массива генерируются последовательно. $x_i = \text{nextRand24}() \% m$;

Формат выходных данных

Выведите количество инверсий

Примеры

invcnt.in	invcnt.out
20 5 19 18	63

Замечание

Сгенерированный массив: 01142210424031343330.

Задача E. Знакопереключение

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

Реализуйте структуру данных из n элементов $a_1, a_2 \dots a_n$, поддерживающую следующие операции:

- присвоить элементу a_i значение j ;
- найти знакопеременяющуюся сумму на отрезке от l до r включительно ($a_l - a_{l+1} + a_{l+2} - \dots \pm a_r$).

Формат входных данных

В первой строке входного файла содержится натуральное число n ($1 \leq n \leq 10^5$) — длина массива. Во второй строке записаны начальные значения элементов (неотрицательные целые числа, не превосходящие 10^4).

В третьей строке находится натуральное число m ($1 \leq m \leq 10^5$) — количество операций. В последующих m строках записаны операции:

- операция первого типа задается тремя числами $0 \ i \ j$ ($1 \leq i \leq n, 1 \leq j \leq 10^4$).
- операция второго типа задается тремя числами $1 \ l \ r$ ($1 \leq l \leq r \leq n$).

Формат выходных данных

Для каждой операции второго типа выведите на отдельной строке соответствующую знакопеременяющуюся сумму.

Пример

signchange.in	signchange.out
3	-1
1 2 3	2
5	-1
1 1 2	3
1 1 3	
1 2 3	
0 2 1	
1 1 3	