

Задача А. Баржа

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

На барже располагается K грузовых отсеков. В каждый отсек можно поместить некоторое количество бочек с одним из 10 000 видов топлива. Причём извлечь бочку из отсека можно лишь в случае, если все бочки, помещённые в этот отсек после неё, уже были извлечены. Таким образом в каждый момент времени в каждом непустом отсеке имеется ровно одна бочка, которую можно извлечь не трогая остальных. Будем называть такие бочки крайними.

Изначально баржа пуста. Затем она последовательно проплывает через N доков, причём в каждом доке на баржу либо погружается бочка с некоторым видом топлива в некоторый отсек, либо выгружается крайняя бочка из некоторого отсека. Однако, если указанный отсек пуст, либо если выгруженная бочка содержит не тот вид топлива, который ожидалось, следует зафиксировать ошибку. Если на баржу оказывается погружено более P бочек или если после прохождения всех доков она не стала пуста, следует также зафиксировать ошибку. От вас требуется либо указать максимальное количество бочек, которые одновременно пребывали на барже либо зафиксировать ошибку.

Формат входного файла

В первой строке три целых числа N , K и P ($1 \leq N, K, P \leq 100\,000$). Далее следует N строк с описанием действия, выполняемого в очередном доке. Если в нём происходит погрузка, то строка имеет вид «+ $A B$ », где A — номер отсека, в который помещается бочка, а B — номер вида топлива в ней. Если же док занимается разгрузкой, то строка имеет вид «- $A B$ », где A — номер отсека, из которого извлекается бочка, а B — номер ожидаемого вида топлива.

Формат выходного файла

Вывести либо одно число, равное искомому максимуму в случае безошибочного прохождения баржной маршрута, либо вывести слово «Error» в противном случае.

Примеры

ship.in	ship.out
10 1 5	5
+ 1 1	
+ 1 2	
+ 1 3	
+ 1 4	
+ 1 5	
- 1 5	
- 1 4	
- 1 3	
- 1 2	
- 1 1	

Задача В. Деки на 6 мегабайтах

Имя входного файла: `deques.in`
Имя выходного файла: `deques.out`
Ограничение по времени: 1 second
Ограничение по памяти: 6 megabytes

Напишите программу, которая умеет оперировать большим количеством деков. Дек — это «очередь с двумя концами».

Формат входного файла

Первая строка входного файла содержит общее количество команд n ($0 \leq n \leq 150\,000$). Каждая из следующих n строк содержит описание команды:

- «pushfront $A B$ » — вставить число B в начало дека A ;
- «pushback $A B$ » — вставить число B в конец дека A ;
- «popfront A » — удалить первый элемент дека A ;
- «popback A » — удалить последний элемент дека A .

Для каждой команды параметры A и B — целые числа от 1 до 150 000 включительно.

Формат выходного файла

Для каждой команды `popfront` или `popback` выведите удаляемое число. Гарантируется, что перед выполнением команды удаления соответствующий дек не пуст.

Примеры

deques.in	deques.out
9	71819
pushfront 1 71819	1
pushback 2 71820	11
pushback 1 1	71820
popfront 1	
popfront 1	
pushfront 2 10	
pushback 2 11	
popback 2	
popback 2	

Задача C. Range Sum Query

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

Дан массив целых чисел, изначально заполненный нулями. С ним необходимо производить два типа операций.

- $! i x$ — присвоить ячейке i значение x ;
- $? l r$ — узнать сумму значений на отрезке $[l, r]$.

Формат входного файла

Первая строка входного файла содержит два числа n и m - размер массива и количество запросов соответственно ($1 \leq n \leq (10^5 + 7)$, $1 \leq m \leq 2 \cdot 10^5$). За ним следуют m запросов по одному на строке.

В процессе работы программы необходимо поддерживать два числа P и Q . Изначально $P = 91$, $Q = 47$.

Запрос вида « $! A B$ » обозначает, что в ячейку $(A + P) \bmod n$ необходимо записать число $(B + Q) \bmod (10^9 + 7)$.

Запрос вида « $? A B$ » обозначает, что необходимо посчитать сумму между элементами $(A + P) \bmod n$, $(B + Q) \bmod n$ включительно. Пусть ответ равен Z . Тогда необходимо изменить числа P и Q следующим образом:

$$P = (P \cdot 31 + Z) \bmod (10^9 + 7),$$

$$Q = (Q \cdot 29 + Z) \bmod (10^9 + 7).$$

Нумерация элементов массива начинается с нуля.

Все числа во входном файле не превосходят $10^9 + 7$.

Формат выходного файла

Для каждого запроса на сумму нужно вывести ответ на него на отдельной строке.

Примеры

rsq.in	rsq.out
10 8	52
! 1 4	1416
! 2 5	42558
? 3 6	103
! 4 1	
? 5 9	
! 1 4	
? 5 9	
? 5 9	
15 6	97
! 1 1	0
! 5 2	0
? 5 14	
! 8 7	
? 1 4	
? 4 13	

Задача D. Range Variation Query

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

В начальный момент времени последовательность a_n задана следующей формулой:
 $a_n = n^2 \bmod 12345 + n^3 \bmod 23456$.

Требуется много раз отвечать на запросы следующего вида:

- найти разность между максимальным и минимальным значениями среди элементов a_i, a_{i+1}, \dots, a_j ;
- присвоить элементу a_i значение j .

Формат входного файла

Первая строка входного файла содержит натуральное число k — количество запросов ($1 \leq k \leq 100\,000$). Следующие k строк содержат запросы, по одному на строке. Запрос номер i описывается двумя целыми числами x_i, y_i .

Если $x_i > 0$, то требуется найти разность между максимальным и минимальным значениями среди элементов a_{x_i}, \dots, a_{y_i} . При этом $1 \leq x_i \leq y_i \leq 100\,000$.

Если $x_i < 0$, то требуется присвоить элементу $a_{|x_i|}$ значение y_i . В этом случае $-100\,000 \leq x_i \leq -1$ и $|y_i| \leq 100\,000$.

Формат выходного файла

Для каждого запроса первого типа в выходной файл требуется вывести одну строку,

содержащую разность между максимальным и минимальным значениями на соответствующем отрезке.

Примеры

rvq.in	rvq.out
7	34
1 3	68
2 4	250
-2 -100	234
1 5	1
8 9	
-3 -101	
2 3	