

## Задача Next. Следующий

Имя входного файла: `next.in`  
 Имя выходного файла: `next.out`  
 Ограничение по времени: 3 секунды  
 Ограничение по памяти: 64 мегабайта

Реализуйте структуру данных, которая поддерживает множество  $S$  целых чисел, с которым разрешается производить следующие операции:

- $add(i)$  — добавить в множество  $S$  число  $i$  (если он там уже есть, то множество не меняется);
- $next(i)$  — вывести минимальный элемент множества, не меньший  $i$ . Если искомым элемент в структуре отсутствует, необходимо вывести -1.

### Формат входного файла

Исходно множество  $S$  пусто. Первая строка входного файла содержит  $n$  — количество операций ( $1 \leq n \leq 300\,000$ ). Следующие  $n$  строк содержат операции. Каждая операция имеет вид либо «+  $i$ », либо «?  $i$ ». Операция «?  $i$ » задает запрос  $next(i)$ .

Если операция «+  $i$ » идет во входном файле в начале или после другой операции «+», то она задает операцию  $add(i)$ . Если же она идет после запроса «?», и результат этого запроса был  $y$ , то выполняется операция  $add((i + y) \bmod 10^9)$ .

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до  $10^9$ .

### Формат выходного файла

Для каждого запроса выведите одно число — ответ на запрос.

### Пример

next.in	next.out
6	3
+ 1	4
+ 3	
+ 3	
? 2	
+ 1	
? 4	

## Задача Sum. 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 входного файла

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 l r x` — присвоить элементам массива с позициями от  $l$  до  $r$  значение  $x$  ( $1 \leq l \leq r \leq n$ ,  $0 \leq x \leq 10^9$ )
2. `Q l r` — найти сумму чисел в массиве на позициях от  $l$  до  $r$ . ( $1 \leq l \leq r \leq n$ )

Изначально массив заполнен нулями.

### Формат выходного файла

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

## Примеры

sum.in	sum.out
5 9	3
A 2 3 2	2
A 3 5 1	3
A 4 5 2	4
Q 1 3	2
Q 2 2	7
Q 3 4	
Q 4 5	
Q 5 5	
Q 1 5	

## Задача Kth. К-ый минимум

Имя входного файла: `kth.in`  
 Имя выходного файла: `kth.out`  
 Ограничение по времени: 0.5 секунды  
 Ограничение по памяти: 64 мегабайта

Напишите программу, которая находит  $k$ -ое в возрастающем порядке число в массиве  $A = \langle a_1, a_2, \dots, a_n \rangle$ .

Массив  $A$  задается с помощью полинома  $P(x) = 132x^3 + 77x^2 + 1345x + 1577$ :  $a_i = P(i) \bmod 1743$ .

### Формат входного файла

Входной файл содержит два натуральных числа  $n$  и  $k$  ( $1 \leq k \leq n \leq 4\,000\,000$ ).

### Формат выходного файла

В выходной файл выведите одно число — ответ на задачу.

### Пример

kth.in	kth.out
1 1	1388
10 1	402

## Задача Sum2. Суммы без модуля

Имя входного файла: `sum.in`  
 Имя выходного файла: `sum.out`  
 Ограничение по времени: 3 секунды  
 Ограничение по памяти: 64 мегабайта

Участники сборов приезжают на сборы группами, и их необходимо заселить в гостиницу. Недолго думая, администратор гостиницы селит  $i$ -ю группу в комнаты с номерами с  $l_i$  по  $r_i$  включительно, по одному человеку в комнату (соответственно, в  $i$ -й группе  $r_i - l_i + 1$  человек). Комнаты не резиновые, а именно вмещают лишь  $k - 1$  человек. Как только в комнату заселяется  $k$ -й человек, все обитатели этой комнаты обижаются и уезжают домой (включая только что заселившегося).

Вдохновленный новым эффективным методом заселения, администратор решил применить подобный метод для завтраков, обедов и ужинов участников сборов. А именно, на  $j$ -й прием пищи приглашаются лишь участники из комнат с номерами с  $s_j$  по  $t_j$  включительно. Вам необходимо подсчитать, сколько порций нужно готовить на каждый прием пищи.

### Формат входного файла

В первой строке записаны три натуральных числа — число комнат  $n$  ( $1 \leq n \leq 100\,000$ ), характеристика вместимости комнаты  $k$  ( $2 \leq k \leq 5$ ), и количество произошедших на сборах событий  $m$  ( $1 \leq m \leq 100\,000$ ). В последующих  $m$  строках описаны сами произошедшие события в хронологическом порядке, по одному на строке.

Каждое событие описывается тремя целыми числами. Заезд очередной группы участников описывается как «1  $l$   $r$ », где  $l$  и  $r$  задают диапазон номеров комнат для заселения ( $1 \leq$

$l \leq r \leq n$ ). Очередной прием пищи описывается как «2 s t», где  $s$  и  $t$  ( $1 \leq s \leq t \leq n$ ) задают диапазон номеров комнат, приглашенных в столовую.

### Формат выходного файла

На каждый запрос второго вида выведите количество купающихся участников на отдельной строке.

### Пример

sum.in	sum.out
3 3 9	3
1 1 3	2
1 1 2	1
1 1 1	1
2 1 3	
2 1 2	
1 1 3	
1 3 3	
2 1 3	
2 1 2	