

у2018-1-2. Стек, очередь, СНМ

A. Minimum on Stack

2 seconds, 256 megabytes

You have to implement the data structure that supports the following operations:

- 1. 1  $x$  — add  $x$  to an end of the data structure.
- 2. 2 — retrieve the last element from the data structure.
- 3. 3 — find the minimal element in the data structure.

Input

The first line of the input contains one integer  $n$  ( $1 \leq n \leq 10^6$ ) — the number of operations. Next  $n$  lines contain the description of operations, one per line. The argument  $x$  in an operation of the first type lies in  $[-10^9, 10^9]$ . It is guaranteed that before retrieval the data structure is not empty.

Output

Output the result for each operation of the third type, one per line.

input
8
1 2
1 3
1 -3
3
2
3
2
3
output
-3
2
2

Statement is not available on English language

В. Шарик

2 секунды, 256 мегабайт

В одной компьютерной игре игрок выставляет в линию шарик разных цветов. Когда образуется непрерывная цепочка из трех и более шариков одного цвета, она удаляется из линии. Все шарик при этом сдвигаются друг к другу, и ситуация может повториться.

Напишите программу, которая по данной ситуации определяет, сколько шариков будет сейчас уничтожено. Естественно, непрерывных цепочек из трех и более одноцветных шаров в начальный может быть не более одной.

Входные данные

Даны количество шариков в цепочке (не более  $10^5$ ) и цвета шариков (от 0 до 9, каждому цвету соответствует свое целое число).

Выходные данные

Требуется вывести количество шариков, которое будет уничтожено.

входные данные
5 1 3 3 3 2
выходные данные
3

входные данные
10 3 3 2 1 1 1 2 2 3 3
выходные данные
10

Statement is not available on English language

С. Астроград

1 секунда, 256 мегабайт

В Астрополисе прошел концерт популярной группы Астроград. За пару дней до концерта перед кассой выстроилась огромная очередь из людей, желающих туда попасть. Изначально очередь была пуста. В каждый из  $n$  моментов времени происходило следующее:

- 1. В очередь пришел новый человек с уникальным номером  $id$ , он встает в очередь последним.
- 2. Человеку, стоящему спереди очереди, удалось купить билет. Он уходит.
- 3. Человеку, стоящему последнему в очереди, надоело ждать. Он уходит.
- 4. Человек с уникальным номером  $q$  хочет знать, сколько людей стоит в очереди спереди него.
- 5. Очередь хочет знать, человек с каким уникальным номером стоит сейчас первым и задерживает всех.

Вам необходимо написать программу, которая умеет обрабатывать описанные события.

Входные данные

В первой строке дано целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество событий. В каждой из следующих  $n$  строк дано описание событий: номер события, а также число  $id$  ( $1 \leq id \leq 10^5$ ) для событий типа 1 и число  $q$  для событий типа 4. События происходили в том порядке, в каком они описаны во входном файле. Гарантируется корректность всех событий.

Выходные данные

Выведите ответы для событий типа 4 и 5 в том порядке, в каком они описаны во входном файле.

входные данные
7
1 1
5
1 3
3
2
1 2
4 2
выходные данные
1
0

В примере из условия происходили следующие события:

- 1. В очередь пришел человек с  $id = 1$ . Очередь: [ 1 ]
- 2. Первым в очереди стоит человек с  $id = 1$ . Очередь: [ 1 ]
- 3. В очередь пришел человек с  $id = 3$ . Очередь: [ 1, 3 ]
- 4. Последнему в очереди надоело стоять и он уходит. Очередь: [ 1 ]

5. Первому в очереди удалось купить билет и он уходит. Очередь: [ ]
6. В очередь пришел человек с  $id = 2$ . Очередь: [ 2 ]
7.  $q = 2$  хочет знать, сколько человек стоит перед ним. Очередь: [ 2 ]

Statement  
is not  
available  
on  
English  
language

D. Гоблины и шаманы

2 секунды, 256 мегабайт

Гоблины Мглистых гор очень любят ходить к своим шаманам. Так как гоблинов много, к шаманам часто образуются очень длинные очереди. А поскольку много гоблинов в одном месте быстро образуют шумную толку, которая мешает шаманам проводить сложные медицинские манипуляции, последние решили установить некоторые правила касательно порядка в очереди.

Обычные гоблины при посещении шаманов должны вставать в конец очереди. Привилегированные же гоблины, знающие особый пароль, встают ровно в ее середину, причем при нечетной длине очереди они встают сразу за центром.

Так как гоблины также широко известны своим непочтительным отношением ко всяческим правилам и законам, шаманы попросили вас написать программу, которая бы отслеживала порядок гоблинов в очереди.

Входные данные

В первой строке входных данных записано число  $N$  ( $1 \leq N \leq 5 \cdot 10^5$ ) - количество запросов к программе. Следующие  $N$  строк содержат описание запросов в формате:

- „+ i” - гoblin с номером  $i$  ( $1 \leq i \leq N$ ) встает в конец очереди.
- „\* i” - привилегированный гoblin с номером  $i$  встает в середину очереди.
- „-” - первый гoblin из очереди уходит к шаманам. Гарантируется, что на момент такого запроса очередь не пуста.

Выходные данные

Для каждого запроса типа „-” программа должна вывести номер гоблина, который должен зайти к шаманам.

входные данные
7 + 1 + 2 - + 3 + 4 - -
выходные данные
1 2 3

E. Postfix Notation

1 second, 256 megabytes

In a postfix notation (Reverse Polish notation), two operands are followed by an operation. For example, the sum of two numbers  $A$  and  $B$  is written as  $A\ B\ +$ . The expression  $B\ C\ +\ D\ *$  means  $(B + C) * D$ , and the expression  $A\ B\ C\ +\ D\ * +$  means  $A + (B + C) * D$ . The advantage of a postfix notation is that it does not require brackets and additional operator precedence agreements for its reading.

An expression is given in the reverse Polish notation. Calculate the result.

Input

The input contains the expression in the postfix notation containing single-digit numbers and the operations  $+$ ,  $-$ ,  $*$ . The string contains no more than 100 numbers and operations.

Output

Output the result of the expression. It is guaranteed that the result of the expression, as well as the results of all intermediate calculations is less than  $2^{31}$  by an absolute value.

input
8 9 + 1 7 - *
output
-102

Statement  
is not  
available  
on  
English  
language

F. Сортировка стеком

1 секунда, 256 мегабайт

входные данные
5 5 3 1 2 4
выходные данные
push push push pop push pop pop push pop pop

входные данные
3 2 3 1
выходные данные
impossible

G. Disjoint-Sets

2 seconds, 256 megabytes

Your task is to maintain DSU. Moreover, you'll have to find minimum and maximum element and number of elements in the subset.

Initially, each element is in it's own subset.

Input

First line of input consists of a single integer  $n$  — number of elements in set ( $1 \leq n \leq 300\,000$ ).

Each of the next lines contains a single operation. For each of operation `get`, you should output minimum and maximum element and number of elements in corresponding subset.

Output

Output the sequence of results of operations `get`.

input

```
5
union 1 2
get 3
get 2
union 2 3
get 2
union 1 3
get 5
union 4 5
get 5
union 4 1
get 5
```

output

```
3 3 1
1 2 2
1 3 3
5 5 1
4 5 2
1 5 5
```

Statement  
is not  
available  
on  
English  
language

Н. Подсчет опыта

2 секунды, 64 мегабайта

В очередной онлайн игре игроки, как обычно, сражаются с монстрами и набирают опыт. Для того, чтобы сражаться с монстрами, они объединяются в кланы. После победы над монстром, всем участникам клана, победившего его, добавляется одинаковое число единиц опыта. Особенностью этой игры является то, что кланы никогда не распадаются и из клана нельзя выйти. Единственная доступная операция — объединение двух кланов в один.

Поскольку игроков стало уже много, вам поручили написать систему учета текущего опыта игроков.

Входные данные

В первой строке входного файла содержатся числа  $n$  ( $1 \leq n \leq 200000$ ) и  $m$   $1 \leq m \leq 200000$  — число зарегистрированных игроков и число запросов.

В следующих  $m$  строках содержатся описания запросов. Запросы бывают трех типов:

- `join X Y` — объединить кланы, в которые входят игроки  $X$  и  $Y$  (если они уже в одном клане, то ничего не меняется).
- `add X V` — добавить  $V$  единиц опыта всем участникам клана, в который входит игрок  $X$  ( $1 \leq V \leq 100$ ).
- `get X` — вывести текущий опыт игрока  $X$ .

Изначально у всех игроков 0 опыта и каждый из них состоит в клане, состоящим из него одного.

Выходные данные

Для каждого запроса `get X` выведите текущий опыт игрока  $X$ .

входные данные
3 6 add 1 100 join 1 3 add 1 50 get 1 get 2 get 3
выходные данные
150 0 50