

## А. Скобки

1 с, 256 мегабайт

Требуется определить, является ли правильной данная последовательность круглых, квадратных и фигурных скобок.

### Входные данные

В единственной строке входного файла записано подряд  $N$  скобок ( $1 \leq N \leq 10^5$ ).

### Выходные данные

В выходной файл вывести «YES», если данная последовательность является правильной, и «NO» в противном случае.

входные данные
()
выходные данные
YES

входные данные
{[]}{}{}
выходные данные
YES

входные данные
[]{}{}
выходные данные
NO

Скобочная последовательность называется правильной, если ее можно получить из какого-либо математического выражения вычеркиванием всех символов, кроме скобок.

Формальное определение правильной скобочной последовательности таково:

- Пустая последовательность является правильной.
- Если  $A$  — правильная скобочная последовательность, то  $(A)$ ,  $[A]$  и  $\{A\}$  — правильные скобочные последовательности.
- Если  $A$  и  $B$  — правильные скобочные последовательности, то  $AB$  — правильная скобочная последовательность.

## В. Игра в пьяницу

1 с, 256 мегабайт

В игре в пьяницу карточная колода раздается поровну двум игрокам. Далее они вскрывают по одной верхней карте, и тот, чья карта старше, забирает себе обе вскрытые карты, которые кладутся под низ его колоды. Тот, кто остаётся без карт — проигрывает.

Для простоты будем считать, что все карты различны по номиналу, а также, что самая младшая карта побеждает самую старшую карту («шестерка берет туза»).

Игрок, который забирает себе карты, сначала кладёт под низ своей колоды карту первого игрока, затем карту второго игрока (то есть карта второго игрока оказывается внизу колоды).

Напишите программу, которая моделирует игру в пьяницу и определяет, кто выигрывает. В игре участвует  $n$  карт, имеющих значения от 0 до  $n - 1$ , большая карта побеждает меньшую, карта со значением 0 побеждает карту  $n - 1$ .

### Входные данные

Программа получает на вход три строки. В первой строке содержится целое чётное число  $n$  ( $2 \leq n \leq 100\,000$ ). Вторая строка содержит  $\frac{n}{2}$  чисел — карты первого игрока, а третья —  $\frac{n}{2}$  карт второго игрока. Карты перечислены сверху вниз, то есть каждая строка начинается с той карты, которая будет открыта первой. Гарантируется, что каждая из карт встречается в колодах игроков ровно один раз.

### Выходные данные

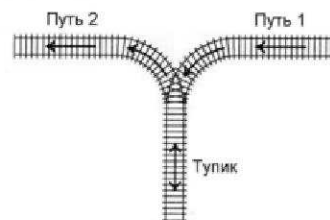
Программа должна определить, кто выигрывает при данной раздаче, и вывести слово «first» или «second», после чего вывести количество ходов, сделанных до выигрыша. Если на протяжении  $2 \cdot 10^5$  ходов игра не заканчивается, программа должна вывести слово «draw».

входные данные
10 1 3 5 7 9 2 4 6 8 0
выходные данные
second 5

## С. Сортировка вагонов

1 с, 256 мегабайт

К тупику со стороны пути 1 (см. рисунок) подъехал поезд. Разрешается отцепить от поезда один или сразу несколько первых вагонов и завести их в тупик (при желании, можно даже завести в тупик сразу весь поезд). После этого часть из этих вагонов вывезти в сторону пути 2. После этого можно завести в тупик еще несколько вагонов и снова часть оказавшихся вагонов вывезти в сторону пути 2. И так далее (так, что каждый вагон может лишь один раз заехать с пути 1 в тупик, а затем один раз выехать из тупика на путь 2). Заезжать в тупик с пути 2 или выезжать из тупика на путь 1 запрещается. Нельзя с пути 1 попасть на путь 2, не заезжая в тупик.



Известно, в каком порядке изначально идут вагоны поезда. Требуется с помощью указанных операций сделать так, чтобы вагоны поезда шли по порядку (сначала первый, потом второй и т.д., считая от головы поезда, едущего по пути 2 в сторону от тупика).

### Входные данные

Вводится число  $N$  — количество вагонов в поезде ( $1 \leq N \leq 2000$ ). Далее идут номера вагонов в порядке от головы поезда, едущего по пути 1 в сторону тупика. Вагоны пронумерованы натуральными числами от 1 до  $N$ , каждое из которых встречается ровно один раз.

### Выходные данные

Если сделать так, чтобы вагоны шли в порядке от 1 до  $N$ , считая от головы поезда, когда поезд поедет по пути 2 из тупика, можно, выведите действия, которые нужно проделать с поездом. Каждое действие описывается двумя числами: типом и количеством вагонов:

- если нужно завезти с пути 1 в тупик  $K$  вагонов, должно быть выведено сначала число 1, а затем — число  $K(K \geq 1)$ ,
- если нужно вывезти из тупика на путь 2  $K$  вагонов, должно быть выведено сначала число 2, а затем — число  $K(K \geq 1)$ .

Если возможно несколько последовательностей действий, приводящих к нужному результату, выведите любую из них. Если выстроить вагоны по порядку невозможно, выведите одно число 0.

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

входные данные
4 4 1 3 2
выходные данные
1 1 1 1 2 1 1 1 1 1 2 1 2 1 2 1

входные данные
3 2 3 1
выходные данные
0

D. Минимум на стеке

2 секунды, 256 мегабайт

Вам требуется реализовать структуру данных, выполняющую следующие операции:

1. Добавить элемент  $x$  в конец структуры.
2. Удалить последний элемент из структуры.
3. Вывести минимальный элемент в структуре.

Входные данные

В первой строке входного файла задано одно целое число  $n$  — количество операций ( $1 \leq n \leq 10^6$ ). В следующих  $n$  строках заданы сами операции. В  $i$ -ой строке число  $t_i$  — тип операции (1, если операция добавления. 2, если операция удаления. 3, если операция минимума). Если задана операция добавления, то через пробел записано целое число  $x$  — элемент, который следует добавить в структуру ( $-10^9 \leq x \leq 10^9$ ). Гарантируется, что перед каждой операцией удаления или нахождения минимума структура не пуста.

Выходные данные

Для каждой операции нахождения минимума выведите одно число — минимальный элемент в структуре. Ответы разделяйте переводом строки.

входные данные
8 1 2 1 3 1 -3 3 2 3 2 3
выходные данные
-3 2 2

E. Шарики

0.5 секунд, 256 мегабайт

В одной компьютерной игре игрок выставляет в линию шарiki разных цветов. Когда образуется непрерывная цепочка из трех и более шариков одного цвета, она удаляется из линии. При этом все шарики 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 момент 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, которые будут уничтожены.

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

F. Астроград

0.5 с, 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 ]

Г. 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 толпу, 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

Н. Постфиксная запись

1 секунда, 256 мегабайт

В постфиксной записи (или обратной польской записи) операция записывается после двух операндов. Например, сумма двух чисел  $A$  и  $B$  записывается как  $A\ B\ +$ . Запись  $B\ C\ +\ D\ *$  обозначает привычное нам  $(B + C) * D$ , а запись  $A\ B\ C\ +\ D\ * +$  означает  $A + (B + C) * D$ . Достоинство постфиксной записи в том, что она не требует скобок и дополнительных соглашений о приоритете операторов для своего чтения.

Дано выражение в обратной польской записи. Определите его значение.

Входные данные

В единственной строке записано выражение в постфиксной записи, содержащее однозначные числа и операции  $+$ ,  $-$ ,  $*$ . Строка содержит не более 100 чисел и операций.

Выходные данные

Необходимо вывести значение записанного выражения.  
 Гарантируется, что результат выражения, а также результаты всех промежуточных вычислений по модулю меньше  $2^{31}$ .

<b>входные данные</b>
8 9 + 1 7 - *
<b>выходные данные</b>
-102

## I. Сортировка стека

1 секунда, 256 мегабайт

Даны два стека А и В. Изначально стек А содержит целые числа от 1 до  $n$  в некотором порядке, а стек В пуст. Вы можете выполнять два типа операций:

1. push: взять верхний элемент из стека А и поместить его в стек В,
2. pop: взять верхний элемент из стека В и вывести его в выходной поток.

Ваша задача - вывести все элементы стека А в отсортированном порядке.

### Входные данные

Первая строка содержит целое число  $n$  ( $1 \leq n \leq 2000$ ). Вторая строка содержит  $n$  целых чисел - элементы в стеке А, при этом левый элемент является верхним в стеке.

### Выходные данные

Выведите последовательность операций, которая выводит все элементы в отсортированном порядке. Если решения не существует, выведите impossible.

<b>входные данные</b>
5 5 3 1 2 4
<b>выходные данные</b>
push push push pop push pop pop push pop pop

<b>входные данные</b>
3 2 3 1
<b>выходные данные</b>
impossible