

## Задача А. Переворот

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

64 мегабайта

Дано натуральное число  $N$  и последовательность из  $N$  элементов. Требуется вывести эту последовательность в обратном порядке.

### Формат входных данных

В первой строке записано натуральное число  $N$  ( $N \leq 10^3$ ). В следующих  $N$  строках идут  $N$  целых чисел, по модулю не превосходящих 1000, — элементы последовательности.

### Формат выходных данных

Выведите заданную последовательность в обратном порядке.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 2                | 4                 |
| 3                | 3                 |
| 4                |                   |

### Замечание

Для вывода последовательности в обратном порядке необходимо использовать рекурсивную функцию.

## Задача В. Реализуйте стек

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Требуется реализовать стек с двумя операциями.

Первая операция `push` добавляет элемент в стек, а вторая операция `pop` — удаляет. На каждую вторую операцию необходимо вывести удаленное число. Гарантируется, что всегда есть, что удалить.

Добавление элемента, называемое также проталкиванием `push`, возможно только в вершину стека (добавленный элемент становится первым сверху). Удаление элемента, называемое также выталкиванием `pop`, возможно также только из вершины стека, при этом, второй сверху элемент становится верхним.

### Формат входных данных

В первой строчке находится число операций  $n$  ( $1 \leq n \leq 100\,000$ ). В следующих  $n$  строчках первое число — номер операции (1 для `push`, 2 для `pop`), второе (только для `push`) — добавляемое число (это число натуральное и не превосходит 100 000).

### Формат выходных данных

Выведите все удаленные числа по одному в строке.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 6                | 6                 |
| 1 3              | 8                 |
| 1 4              |                   |
| 1 6              |                   |
| 2                |                   |
| 1 8              |                   |
| 2                |                   |

## Задача С. Реализуйте очередь

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Для каждой операции изъятия элемента выведите ее результат.

На вход программе подаются строки, содержащие команды. Каждая строка содержит одну команду. Команда — это либо “+ N”, либо “-”. Команда “+ N” означает добавление в очередь числа  $N$ , по модулю не превышающего  $10^9$ . Команда “-” означает изъятие элемента из очереди.

### Формат входных данных

В первой строке содержится количество команд —  $m$  ( $1 \leq m \leq 10^5$ ). В последующих строках содержатся команды, по одной в каждой строке.

### Формат выходных данных

Выведите числа, которые удаляются из очереди, по одному в каждой строке. Гарантируется, что изъятий из пустой очереди не производится.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4                | 1                 |
| + 1              | 10                |
| + 10             |                   |
| -                |                   |
| -                |                   |

## Задача D. Скобки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 64 мегабайта

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

### Формат входных данных

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

### Формат выходных данных

Вывести «YES», если данная последовательность является правильной, и «NO» в противном случае.

### Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| ()               | YES               |
| ([])             | YES               |

### Замечание

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

Формальное определение правильной скобочной последовательности таково:

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

## Задача Е. Постфиксная запись

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

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.

### Формат входных данных

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

### Формат выходных данных

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

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 8 9 + 1 7 - *    | -102              |

## Задача F. Игра в пьяницу

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 2 секунды         |
| Ограничение по памяти:  | 256 мегабайт      |

В игре в пьяницу карточная колода раздаётся 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  $n - 1$ .

### Формат входных данных

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.

### Формат выходных данных

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**».

### Пример

| стандартный ввод             | стандартный вывод |
|------------------------------|-------------------|
| 10<br>1 3 5 7 9<br>2 4 6 8 0 | second 5          |

## Задача G. Марсианская парикмахерская

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Пока я не поспал, «сегодня» не наступило

мистер Грин

В парикмахерской работает один мастер. Он тратит на одного клиента ровно 20 минут, а затем сразу переходит к следующему, если в очереди кто-то есть, либо ожидает, когда придет следующий клиент.

Даны времена прихода клиентов в парикмахерскую (в том порядке, в котором они приходили).

Также у каждого клиента есть характеристика, называемая *степенью нетерпения*. Она показывает, сколько человек может максимально находиться в очереди перед клиентом, чтобы он дождался своей очереди и не ушел раньше. Если в момент прихода клиента в очереди находится больше людей, чем степень его нетерпения, то он решает не ждать своей очереди и уходит. Клиент, который обслуживается в данный момент, также считается находящимся в очереди.

Требуется для каждого клиента указать время его выхода из парикмахерской.

### Формат входных данных

В первой строке вводится натуральное число  $N$ , не превышающее  $10^5$  — количество клиентов.

В следующих  $N$  строках вводятся времена прихода клиентов — по два числа, обозначающие часы и минуты (часы — от 0 до 16 000 000, минуты — от 0 до 59) и степень его нетерпения (неотрицательное целое число не большее  $10^5$ ) — максимальное количество человек, которое он готов ждать впереди себя в очереди. Времена указаны в порядке возрастания (все времена различны).

Если для каких-то клиентов время окончания обслуживания одного клиента и время прихода другого совпадают, то можно считать, что в начале заканчивается обслуживание первого клиента, а потом приходит второй клиент.

### Формат выходных данных

Выведите  $N$  пар чисел: времена выхода из парикмахерской 1-го, 2-го, ...,  $N$ -го клиента (часы и минуты). Если на момент прихода клиента человек в очереди больше, чем степень его нетерпения, то нужно считать, что время его ухода равно времени прихода.

### Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3                | 10 20             |
| 10 0 0           | 10 40             |
| 10 1 1           | 10 2              |
| 10 2 1           |                   |
| 5                | 1 20              |
| 1 0 100          | 2 20              |
| 2 0 0            | 2 1               |
| 2 1 0            | 2 40              |
| 2 2 3            | 2 3               |
| 2 3 0            |                   |

## Задача Н. Очередь

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 3 секунды         |
| Ограничение по памяти:  | 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 производить 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.

### Формат входных данных

В первой строке записано одно целое число  $N$  ( $1 \leq N \leq 3 \cdot 10^5$ ) — число запросов к вашей программе. В следующих  $N$  строках заданы описания запросов в следующем формате:

- «+  $i$ » — к очереди присоединяется гoblin  $i$  ( $1 \leq i \leq N$ ) и встает в ее конец;
- «\*  $i$ » — привилегированный гoblin  $i$  встает в середину очереди ( $1 \leq i \leq N$ );
- «-» — гoblin выходит из очереди и заходит к шаманам. Гарантируется, что на момент каждого такого запроса очередь будет не пуста.

### Формат выходных данных

Для каждого запроса третьего типа в отдельной строке выведите номер гоблина, который должен зайти к шаманам.

### Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 7                | 1                 |
| + 1              |                   |
| + 2              | 2                 |
| -                |                   |
| + 3              | 3                 |
| + 4              |                   |
| -                |                   |
| -                |                   |