

Задача А. Постфиксная запись

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

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} .

Примеры

<code>stdin</code>	<code>stdout</code>
<code>8 9 + 1 7 - *</code>	<code>-102</code>

Задача В. Проверка ПСП

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дана строка, состоящая из круглых, квадратных и фигурных скобок. Нужно проверить, является ли она правильной скобочной последовательностью.

Формат входных данных

Во входном файле записана скобочная последовательность длиной не более 10 000 символов.

Формат выходных данных

Выведите `YES`, если скобочная последовательность является правильной, и `NO` в противном случае.

Примеры

<code>stdin</code>	<code>stdout</code>
<code>([])</code>	<code>YES</code>
<code>([]</code>	<code>NO</code>

Задача С. Марсианская парикмахерская

Имя входного файла: `saloon.in`
Имя выходного файла: `saloon.out`
Ограничение по времени: 1 second
Ограничение по памяти: 64 megabytes

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

В выходной файл 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

<code>saloon.in</code>	<code>saloon.out</code>
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 D. Гоблины и шаманы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	64 мегабайта

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.

Формат входных данных

В первой строке входных данных записано число N ($1 \leq N \leq 2 \cdot 10^5$) - количество запросов к программе. Следующие N строк содержат описание запросов в формате:

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

Формат выходных данных

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

Примеры

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

Задача Е. Сортировка вставками

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

Отсортируйте массив целых чисел в порядке невозрастания, **используя сортировку вставками**.

Формат входных данных

Первая строка входного файла содержит целое число N ($1 \leq N \leq 1000$), вторая строка — N целых чисел, по модулю не превышающих $2 \cdot 10^9$.

Формат выходных данных

Данные числа следует вывести в порядке невозрастания.

Примеры

<code>sort.in</code>	<code>sort.out</code>
5 9 2 7 1 2	9 7 2 2 1
5 5 4 3 2 1	5 4 3 2 1

Замечание

В этой задаче нельзя пользоваться методом `.sort()` и функцией `sorted()`, а также использовать метод `.reverse()`

Задача F. Имперский марш

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

На этот раз Император нагрязнул с ревизией не в какой-то там ангар, а в казармы 501-го легиона имперских штурмовиков. В связи с этим каждого штурмовика постригли «под ежика». Несмотря на развитие нанотехнологий, постригли плохо — в результате из-за различной длины волос штурмовики могут отличаться друг от друга по росту, но незначительно — разница не превышает 137 нанометров. Ваша задача — выстроить штурмовиков по росту.

Формат входных данных

Первая строка входного файла содержит целое число N — количество штурмовиков ($1 \leq N \leq 100\,000$), вторая строка N — натуральных чисел, не превышающих $2 \cdot 10^9$ каждое — рост штурмовика в нанометрах. Никакие два роста не различаются более, чем на 137 нм.

Формат выходных данных

Выведите роста штурмовиков в порядке убывания.

Примеры

<code>march.in</code>	<code>march.out</code>
5 12 1 2 1 13	1 1 2 12 13
1 1000000000	1000000000

Замечание

При решении этой задачи нельзя пользоваться стандартными функциями и методами `min`, `index`, `sort`, `sorted` и т. д.

Естественно, можно пользоваться функциями `min`, `max`, которые принимают два числа.

Задача G. Палиндром

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

Палиндром — это строка, которая читается одинаково как справа налево, так и слева направо.

На вход программы поступает набор больших латинских букв (не обязательно различных). Решается переставлять буквы, а также удалять некоторые буквы. Требуется из данных букв по указанным правилам составить палиндром наибольшей длины, а если таких палиндромов несколько, то выбрать первый из них в алфавитном порядке.

Формат входных данных

Входные данные содержат одну непустую строку, состоящую лишь из не более чем 10^5 заглавных латинских символов, без пробелов.

Формат выходных данных

В единственной строке выходных данных выведите искомый палиндром.

Примеры

стандартный ввод	стандартный вывод
AAB	ABA
QAZQAZ	AQZZQA