

Задача А. Постфиксная запись

Имя входного файла: postfix.in
Имя выходного файла: postfix.out
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

postfix.in	postfix.out
8 9 + 1 7 - *	-102

Задача В. Парикмахерская

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

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 100 — количество клиентов.

В следующих N строках вводятся времена прихода клиентов — по два числа, обозначающие часы и минуты (часы — от 0 до 23, минуты — от 0 до 59) и степень его нетерпения (неотрицательное целое число не большее 100) — максимальное количество человек, которое он готов ждать впереди себя в очереди. Времена указаны в порядке возрастания (все времена различны).

Гарантируется, что всех клиентов успеют обслужить до полуночи.

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 можно 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 С. 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 D. Хип ли?

Имя входного файла: `isheap.in`
Имя выходного файла: `isheap.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Структуру данных Heap можно реализовать на основе массива.

Для этого должно выполняться *основное свойство Heap'a*, которое заключается в следующем. Для каждого $1 \leq i \leq n$ выполняются следующие условия:

- Если $2i \leq n$, то $a[i] \leq a[2i]$
- Если $2i + 1 \leq n$, то $a[i] \leq a[2i + 1]$

Дан массив целых чисел. Определите является ли он Heap'ом.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$). Вторая строка содержит n целых чисел по модулю не превосходящих $2 \cdot 10^9$.

Формат выходных данных

Выведите «YES», если массив является Heap'ом и «NO» в противном случае.

Примеры

<code>isheap.in</code>	<code>isheap.out</code>
5 1 0 1 2 0	NO
5 1 3 2 5 4	YES

Задача Е. Хипуй!

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

В этой задаче вам необходимо организовать структуру данных `Heap` для хранения целых чисел, над которой определены следующие операции:

- `Insert(X)` — добавить в `Heap` число X ;
- `Extract` — достать из `Heap` наибольшее число (удалив его при этом).

Формат входных данных

Во входном файле записано количество команд N ($1 \leq N \leq 100\,000$), потом последовательность из N команд, каждая в своей строке.

Каждая команда имеет такой формат: „0 <число>“ или „1“, что означает соответственно операции `Insert(<число>)` и `Extract`. Добавляемые числа находятся в интервале от 1 до 10^7 включительно.

Гарантируется, что при выполнении команды `Extract` в структуре находится по крайней мере один элемент.

Формат выходных данных

В выходной файл для каждой команды извлечения необходимо вывести число, полученное при выполнении команды `Extract`.

Примеры

<code>heap.in</code>	<code>heap.out</code>
7	100
0 100	50
0 10	
1	
0 5	
0 30	
0 50	
1	

Задача F. Сортировка кучей

Имя входного файла: `qsort.in`
Имя выходного файла: `qsort.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Отсортируйте список с помощью `heapsort`.

Формат входных данных

В единственной строке входного файла содержится последовательность, содержащая не более чем 100000 целых чисел.

Формат выходных данных

В единственной строке выходного файла выведите последовательность в неубывающем порядке.

Примеры

<code>qsort.in</code>	<code>qsort.out</code>
4 1 4 8 6 6 5	1 4 4 5 6 6 8

Задача G. Монополия

Имя входного файла: `monopoly.in`
Имя выходного файла: `monopoly.out`
Ограничение по времени: 15 секунд
Ограничение по памяти: 256 мегабайт

В новом варианте игры “Монополия” появилась возможность объединять несколько предприятий в одно для увеличения приносимого ими дохода. При это в игре действуют следующие правила:

1. За один ход можно объединить ровно два предприятия в одно. При этом стоимость нового предприятия равна сумме стоимостей двух предприятий до объединения.
2. За совершение операции по объединению предприятий необходимо заплатить налог в размере 5% от стоимости объединяемых предприятий.

Коля уже заработал в игре много денег и теперь хочет объединить все свои предприятия в одно. Он заметил, что общая сумма уплаченного налога зависит от того, в каком порядке будут объединяться предприятия. Например, пусть у Коли есть четыре предприятия стоимостью 10, 11, 12 и 13. Если Коля сначала объединит предприятия 10 и 11 (это обойдется ему в \$1.05), потом результат — с 12 (\$1.65), и затем — с 13 (\$2.3), то всего он заплатит \$5. Если же сначала отдельно объединить 10 и 11 (\$1.05), потом — 12 и 13 (\$1.25) и, наконец, объединить два полученных предприятия (\$2.3), то в итоге он заплатит лишь \$4.6.

Помогите Коле определить минимальную сумму денег, необходимую для объединения всех его предприятий в одно.

Формат входных данных

В единственной строке входного файла записано N натуральных чисел ($2 \leq N \leq 200\,000$), каждое из которых не превосходит 10000 — стоимости Колиных предприятий.

Формат выходных данных

В выходной файл выведите минимальную сумму денег необходимую для объединения всех Колиных предприятий в одно

Примеры

<code>monopoly.in</code>	<code>monopoly.out</code>
10 11 12 13	4.6000000000
1 1	0.1000000000