

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

stdin	stdout
8 9 + 1 7 - *	-102

Задача В. Хип ли?

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Структуру данных *Неар* можно реализовать на основе массива.

Для этого должно выполняться *основное свойство Неар'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 *Неар'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 *Неар'ом* и «NO» в противном случае.

Примеры

stdin	stdout
5 1 0 1 2 0	NO
5 1 3 2 5 4	YES

Задача С. Хипуй!

Имя входного файла: `heap.in`
Имя выходного файла: `heap.out`
Ограничение по времени: 4 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

heap.in	heap.out
7	100
0 100	50
0 10	
1	
0 5	
0 30	
0 50	
1	

Задача D. Гемоглобин

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

Каждый день к Грегори Хаусу приходит много больных, и у каждого измеряется уровень гемоглобина в крови. Данные по всем пациентам заносятся в базу данных.

Но волчанка попадается один раз на миллион, а работать с остальными неинтересно. Чтобы Хаус не выгонял больных, Кадди иногда запрашивает статистику по k последним больным: ей хочется знать сумму их уровня гемоглобина.

Также Хаус — мизантроп: он смотрит уровень гемоглобина больного, который поступил к нему позже всех, и, видя, что это точно не волчанка, выписывает его из больницы и удаляет информацию о нем из базы.

Автоматизацию процесса Хаус поручил Чейзу. Но Чейз почему-то не справился с этой задачей и попросил вас ему помочь.

Формат входных данных

Первой строкой входного файла задано число n ($1 \leq n \leq 100\,000$) — число обращений к базе данных. Запросы к базе выглядят следующим образом: «+ x » ($1 \leq x \leq 10^9$) — добавить пациента с уровнем гемоглобина x в базу, «-» — удалить последнего пациента из базы, «? k » ($1 \leq k \leq 100\,000$) — вывести суммарный гемоглобин последних k пациентов. Гарантируется, что k не превосходит число элементов в базе. Также гарантируется, что запросов на удаление к пустой базе не поступает. Перед началом работы база данных пуста.

Формат выходных данных

Для каждого запроса «-» вывести уровень гемоглобина в крови пациента, а для каждого запроса «? k » — суммарный гемоглобин у последних k поступивших пациентов. Ответы выводите в порядке поступления запросов.

Примеры

стандартный ввод	стандартный вывод
7	5
+1	3
+2	2
+3	1
?2	
-	
-	
?1	

Задача Е. Гоблины и шаманы

Имя входного файла:	<code>shamans.in</code>
Имя выходного файла:	<code>shamans.out</code>
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

<code>shamans.in</code>	<code>shamans.out</code>
7	1
+ 1	2
+ 2	3
-	
+ 3	
+ 4	
-	
-	

Задача F. Дело в шляпах

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

Как-то раз в город Шляп заехал известный парикмахер. До его приезда парикмахеры были явно не очень, так как все жители города предпочитали ходить в шляпах. Но наконец-то настало время снять шляпы!

Парикмахер открыл свою временную парикмахерскую и работает без остановок, пока есть посетители. Жители приходят к нему в тот момент времени, когда им это удобно, и становятся в очередь. Каждому из них требуется своё время на создание индивидуальной стрижки. Парикмахер зовёт первого человека в порядке очереди, стрижёт его, и после ухода посетителя сразу зовёт следующего.

Стоять в очереди скучно, поэтому если подряд приходят двое или более людей в шляпах одинакового фасона — они начинают между собой активно общаться и необычайно гордиться своими шляпами (но всё равно заходят на стрижку, если уж их очередь подошла). Однако, если следом за ними в очередь встаёт человек в шляпе другого фасона, то вся группа подряд стоящих людей в одинаковых шляпах подозрительно смотрит на только что пришедшего «чужого» и совсем уходит из очереди. При этом очередь сдвигается и может появиться новая группа общающихся людей.

Так как обсуждение одинаковых шляп — это очень интересная тема, появление «чужого» человека в очереди привлекает внимание группы сильнее, чем парикмахер. Поэтому если одновременно пришёл человек в другой шляпе и парикмахер зовёт следующего — вся группа уходит, даже если один из них должен был сейчас зайти на стрижку. К парикмахеру при этом зайдёт следующий из оставшейся очереди, возможно даже только что пришедший «чужой».

Местного шляпника теперь интересует, каким жителям ему больше не нужно будет делать шляпы, так как они будут ходить с новыми стильными стрижками?

Формат входных данных

В первой строке содержится число N ($1 \leq N \leq 10^5$) — количество людей, которые придут к парикмахеру.

Каждая из следующих N строк обозначает пришедшего к парикмахеру жителя и содержит по три числа: фасон шляпы (все фасоны местного шляпника пронумерованы от 1 до 10^9), момент времени прихода s ($1 \leq s \leq 10^9$), и время на стрижку t ($1 \leq t \leq 10^9$). Строки упорядочены по времени прихода жителей. Гарантируется, что все приходят в разное время.

Так как парикмахер очень крут, гарантируется, что он успеет постричь всех жителей до момента времени $2 \cdot 10^9$, даже если бы из очереди никто не уходил.

Формат выходных данных

В единственной строке выведите через пробел номера людей в очереди в порядке возрастания, которых парикмахер всё-таки пострижёт. Люди нумеруются в порядке прихода в очередь, начиная с 1.

Примеры

стандартный ввод	стандартный вывод
5 1 2 7 2 4 3 2 6 2 1 7 3 3 8 2	1 4 5

Задача G. Монополия

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
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

стандартный ввод	стандартный вывод
10 11 12 13	4.6000000000
1 1	0.1000000000