

Задача А. Стек

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 Мб

Реализуйте структуру данных "стек". Напишите программу, содержащую описание стека и моделирующую работу стека, реализовав все указанные здесь методы. Программа считывает последовательность команд и в зависимости от команды выполняет ту или иную операцию. После выполнения каждой команды программа должна вывести одну строчку. Возможные команды для программы:

`push n` Добавить в стек число n (значение n задается после команды). Программа должна вывести `ok`.

`pop` Удалить из стека последний элемент. Программа должна вывести его значение.

`back` Программа должна вывести значение последнего элемента, не удаляя его из стека.

`size` Программа должна вывести количество элементов в стеке.

`clear` Программа должна очистить стек и вывести `ok`.

`exit` Программа должна вывести `bye` и завершить работу.

Формат входных данных

Во входном файле дана последовательность команд по одной на строке.

Гарантируется, что набор входных команд удовлетворяет следующим требованиям: максимальное количество элементов в стеке в любой момент не превосходит 100, все команды `pop` и `back` корректны, то есть при их исполнении в стеке содержится хотя бы один элемент.

Формат выходных данных

Выведите результат работы программы.

Примеры

stack.in	stack.out
<code>push 2</code>	<code>ok</code>
<code>push 3</code>	<code>ok</code>
<code>push 5</code>	<code>ok</code>
<code>back</code>	<code>5</code>
<code>size</code>	<code>3</code>
<code>pop</code>	<code>5</code>
<code>size</code>	<code>2</code>
<code>push 7</code>	<code>ok</code>
<code>pop</code>	<code>7</code>
<code>clear</code>	<code>ok</code>
<code>size</code>	<code>0</code>
<code>exit</code>	<code>bye</code>

Задача В. Очередь

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Реализуйте работу очереди. 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 ($1 \leq M \leq 10^6$). 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

queue.in	queue.out
4	1
+ 1	10
+ 10	
-	
-	

Задача С. Скобки

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Рассмотрим последовательность, состоящую из круглых, квадратных и фигурных скобок. Программа должна определить, является ли данная скобочная последовательность правильной.

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

Формат входных данных

В единственной строке входного файла записано подряд N скобок ($1 \leq N \leq 255$).

Формат выходных данных

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

Пример

brackets.in	brackets.out
([])	YES
(({}))	NO

Задача D. Списки по классам

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

Формат входных данных

В каждой строке сначала записан номер класса (число, равное 9, 10 или 11), затем (через пробел) — фамилия ученика.

Формат выходных данных

Необходимо вывести список школьников по классам: сначала всех учеников 9 класса, затем — 10, затем — 11. Внутри одного класса порядок вывода фамилий должен быть таким же, как на входе.

Примеры

list.in	list.out
9 Иванов	9 Иванов
10 Петров	9 Григорьев
11 Сидоров	9 Сергеев
9 Григорьев	10 Петров
9 Сергеев	10 Яковлев
10 Яковлев	11 Сидоров

Задача Е. Постфиксная запись

Ограничение по времени: 2 секунды
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

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

Задача F. Парикмахерская

Ограничение по времени: 2 секунды
Ограничение по памяти: 64 Мб

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).

Так же 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 так же 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. 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. Если в какое-то время в парикмахерскую пришли несколько человек, то будет считаться что они принимают решения в том порядке, в котором они даны во входном файле.

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

saloon.in	saloon.out
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	