

Задача А. Стек (!) (1 балл)

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

Реализуйте работу стека. 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 стек числа N , по модулю не превышающего 10^9 . Команда “-” означает изъятие элемента из стека. Гарантируется, что не происходит извлечения из пустого стека. Гарантируется, что размер стека в процессе выполнения команд не превысит 10^6 элементов.

Формат входного файла

В первой строке входного файла содержится количество команд — M ($1 \leq M \leq 10^6$). Каждая последующая строка исходного файла содержит ровно одну команду.

Формат выходного файла

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

Пример

stack.in	stack.out
6	10
+ 1	1234
+ 10	
-	
+ 2	
+ 1234	
-	

Примечание

Необходимо написать свой стек.

Làm cho ngăn xếp hoạt động. Đối với mỗi thao tác xóa một phần tử hãy in kết quả của nó.

Ở đầu vào chương trình nhận các dòng chứa các lệnh. mỗi dòng chứa 1 lệnh. Lệnh - “+ N”, hoặc là “-”. Lệnh “+ N” biểu thị thêm vào стек số N, theo môđun không vượt quá 10 mũ 9. Lệnh “-” nghĩa là lấy ra phần tử từ стек. Sự bảo đảm rằng không có sự lấy ra từ стек. bảo đảm rằng kích thước của стек trong lúc thực hiện lệnh không vượt quá 10 mũ 6 phần tử

Задача В. Очередь (1 балл)

Имя входного файла: `queue.in`
Имя выходного файла: `queue.out`
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. Гарантируется, что размер очереди в процессе выполнения команд не превысит 10^6 элементов.

Формат входного файла

В первой строке содержится количество команд — M ($1 \leq M \leq 10^6$). В последующих строках содержатся команды, по одной в каждой строке.

Формат выходного файла

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

Пример

<code>queue.in</code>	<code>queue.out</code>
4	1
+ 1	10
+ 10	
-	
-	

Примечание

Необходимо написать свою очередь.

Задача С. Правильная скобочная последовательность (1 балл)

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

Входной файл содержит несколько строк, каждая из которых содержит последовательность символов '(', ')', '[' и ']'. Выясните, является ли она правильной скобочной последовательностью с двумя типами скобок.

Подсказка: используйте стек.

Формат входного файла

Входной файл содержит $1 \leq n \leq 500$ строк, каждая из которых содержит скобочную последовательность длиной $1 \leq l \leq 10^4$.

Формат выходного файла

Для каждой строки входного файла выведите в выходной файл «YES», если соответствующая последовательность является правильной скобочной последовательностью, или «NO», если не является.

Пример

<code>brackets.in</code>	<code>brackets.out</code>
<code>()()</code>	YES
<code>([])</code>	YES
<code>([)]</code>	NO
<code>(([])</code>	NO
<code>)()</code>	NO

Примечание

Необходимо написать свой стек.

Задача D. Постфиксная запись (2 балла)

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

Подсказка: используйте стек.

Формат входного файла

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

Формат выходного файла

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

Примечание

Необходимо написать свой стек.

Задача Е. Двоичный поиск (!) (1 балл)

Имя входного файла: `binsearch.in`
Имя выходного файла: `binsearch.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан массив из n элементов, упорядоченный в порядке неубывания, и m запросов: найти первое и последнее вхождение числа в массив.

Формат входного файла

В первой строке входного файла содержится одно число n — размер массива ($1 \leq n \leq 100000$). Во второй строке находится n чисел в порядке неубывания — элементы массива. В третьей строке находится число m — количество запросов. В следующей строке находится m чисел — запросы.

Формат выходного файла

Для каждого запроса выведите в отдельной строке номер первого и последнего вхождения этого числа в массив. Если числа в массиве нет, выведите два раза -1.

Пример

binsearch.in	binsearch.out
5	1 2
1 1 2 2 2	3 5
3	-1 -1
1 2 3	

Примечание

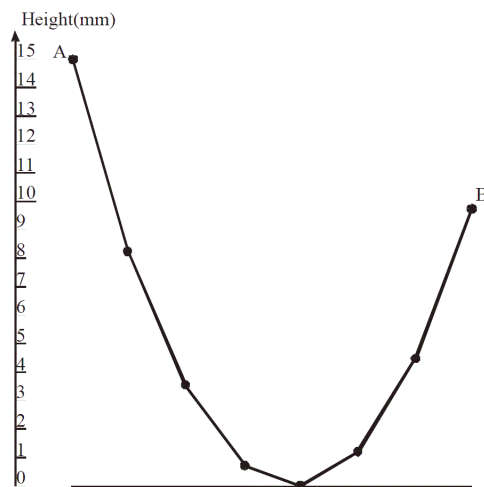
Необходимо написать свой бинарный поиск.

Задача F. Гирлянда (3 балла)

Имя входного файла: `garland.in`
Имя выходного файла: `garland.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Гирлянда состоит из n лампочек на общем проводе. Один её конец закреплён на заданной высоте A мм ($h_1 = A$). Благодаря силе тяжести гирлянда прогибается: высота каждой неконцевой лампы на 1 мм меньше, чем средняя высота ближайших соседей ($h_i = \frac{(h_{i-1} + h_{i+1})}{2} - 1$ для $1 < i < N$). Требуется найти минимальную высоту второго конца B ($B = h_n$) при условии, что лишь одна лампочка может касаться земли, а для остальных выполняется условие $h_i > 0$.

Подсказка: используйте двоичный поиск.



Формат входного файла

В первую строке входного файла содержится два числа n и A ($3 \leq n \leq 1000$, n — целое, $10 \leq A \leq 1000$, A — вещественное).

Формат выходного файла

Вывести одно вещественное число B с двумя знаками после запятой.

Пример

garland.in	garland.out
8 15	9.75
692 532.81	446113.34