

Задача А. Постфиксная запись

Имя входного файла: `postfix.in`
 Имя выходного файла: `postfix.out`
 Ограничение по времени: 1 секунда
 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 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

<code>postfix.in</code>	<code>postfix.out</code>
8 9 + 1 7 - *	-102

Задача В. Хипуй!

Имя входного файла: `heap.in`
 Имя выходного файла: `heap.out`
 Ограничение по времени: 2 секунда
 Ограничение по памяти: 64 мегабайт

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

- Insert(N)** — добавить в **Неар** число N
- Extract** — достать из **Неар** наибольшее число (удалив его при этом)

Формат входного файла

Во входном файле записано количество команд N ($1 \leq N \leq 100000$), потом последовательность из N команд, каждая в своей строке. Каждая команда имеет такой формат: "0 <число>" или "1", обозначающие соответственно операции **Insert**(<число>) и **Extract**. Гарантируется, что при выполнении команды **Extract** в структуре находится по крайней мере один элемент.

Формат выходного файла

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

Пример

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

Задача С. Коллекция

Имя входного файла: `collect.in`
 Имя выходного файла: `collect.out`
 Ограничение по времени: 2 секунда
 Ограничение по памяти: 16 мегабайт

Имеется две коллекции натуральных чисел размеров N и M соответственно. Каждое число в этих коллекциях не превосходит $2 \cdot 10^8$. Для каждого числа K из второй коллекции нужно узнать, содержится ли оно в первой коллекции или нет.

Формат входного файла

В первой строке входного файла содержится число N ($1 \leq N \leq 100000$) — размер первой коллекции. В следующей строке записаны N чисел первой коллекции, упорядоченные по возрастанию. Все числа первой коллекции различны.

В третьей строке файла записан размер второй коллекции M ($1 \leq M \leq 100000$). В последней строке входного файла записаны M чисел второй коллекции.

Формат выходного файла

Выходной файл должен содержать M строк. Для каждого запроса выведите число 1, если данное число второй коллекции содержится в первой, и 0 в противном случае.

Пример

<code>collect.in</code>	<code>collect.out</code>
5	0
2 3 4 7 9	1
3	0
1 3 1000000	

Задача D. Минимум на отрезке

Имя входного файла: `interval.in`
 Имя выходного файла: `interval.out`
 Ограничение по времени: 2 секунда
 Ограничение по памяти: 16 мегабайт

Рассмотрим последовательность целых чисел длины N . По ней с шагом 1 движется "окно" длины K , то есть сначала в "окне" видно первых K чисел, на следующем шаге в "окне" уже будут находиться K чисел начиная со второго и так далее до конца последовательности. Требуется для каждого положения "окна" определить минимум в нём.

Формат входного файла

В первой строке входного файла содержатся два числа N и K ($1 \leq N \leq 150000$, $1 \leq K \leq 10000$, $K \leq N$) — длины последовательности и "окна" соответственно. На следующей строке находятся N чисел — сама последовательность.

Формат выходного файла

Выходной файл должен содержать $N - K + 1$ строк — минимумы для каждого положения "окна".

Пример

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