

## Задача А. А+В

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

Для начала найдите сумму двух целых чисел.

### Формат входного файла

В первой и единственной строке входного файла два числа —  $A$  и  $B$ , по модулю не превышающие  $10^9$ .

### Формат выходного файла

В первой и единственной строке выходного файла одно число, равное сумме  $A$  и  $B$ .

### Пример

<code>aplusb.in</code>	<code>aplusb.out</code>
7 4	11

## Задача В. Скобки

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

Требуется определить, является ли правильной данная последовательность круглых, квадратных и фигурных скобок.

### Формат входного файла

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

### Формат выходного файла

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

### Пример

<code>brackets.in</code>	<code>brackets.out</code>
<code>()</code>	YES
<code>(({}))</code>	NO

## Задача С. 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 D. Очередь

Имя входного файла: `queue.in`  
Имя выходного файла: `queue.out`  
Ограничение по времени: 1 секунда  
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 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 изъятий из пустой очереди не производится.

#### Пример

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

### Задача Е. Результаты олимпиады

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

$N$  участников олимпиады получили уникальные номера от 1 до  $N$ . В результате решения задач на олимпиаде каждый участник получил некоторое количество баллов (целое число от 0 до 600). Известно, кто сколько баллов набрал.

Требуется перечислить участников олимпиады в порядке невозрастания набранных ими баллов.

### Формат входного файла

В первой строке входного файла содержится число  $N$  ( $0 \leq N \leq 101$ ). Далее записаны через пробел  $N$  чисел — количество набранных участниками баллов (1-е число — это баллы, набранные участником номер 1, 2-е — участником номер 2 и т.д.).

### Формат выходного файла

В выходной файл следует вывести  $N$  чисел — номера участников в порядке невозрастания набранных ими баллов (участники, набравшие одинаковое количество баллов могут быть выведены в любом порядке).

#### Пример

olymp.in	olymp.out
5	5 2 4 1 3
100 312 0 312 500	

### Задача F. Станция «Сортировочная»

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

На железнодорожной станции «Сортировочная» на пути находятся  $n$  товарных вагонов, из которых необходимо сформировать состав. Все вагоны имеют одинаковую длину, однако в них размещены различные грузы, поэтому вагоны могут иметь разную массу. Работники железнодорожной станции «Сортировочная» должны выстроить вагоны в порядке возрастания масс — тогда составу будет разрешено отправиться в путь.

Обычно для таких целей используются так называемые маневровые тепловозы и электровозы, однако на этой станции ведутся испытания экспериментального устройства для сортировки вагонов. Предполагается, что оно позволит существенно сократить затраты времени на формирование составов.

Это устройство на воздушной подушке перемещается над вагонами, его длина немного превышает длину двух вагонов. Оно может зависнуть над двумя соседними вагонами, поднять их оба в воздух и поменять местами. Однако, грузоподъемность устройства ограничена — указанную операцию оно может выполнить только, если суммарная масса двух вагонов не превышает  $M$ .

Ваша задача состоит в том, чтобы написать программу, которая определит, возможно ли с помощью экспериментального устройства для сортировки вагонов расставить вагоны, находящиеся на пути, в необходимом порядке.

### Формат входного файла

Первая строка входного файла содержит два числа: число вагонов  $n$  ( $2 \leq n \leq 100\,000$ ) и грузоподъемность экспериментального устройства  $M$  ( $1 \leq M \leq 10^9$ ). Вторая строка входного файла содержит массы вагонов  $m_1, \dots, m_n$  (для этих масс выполняются неравенства  $1 \leq m_i \leq 10^9$ , кроме этого массы вагонов попарно различны). Массы вагонов перечислены во входном файле в том порядке, в котором вагоны исходно стоят на пути.

### Формат выходного файла

В выходной файл выведите слово «Yes», если с помощью экспериментального

устройства для сортировки вагонов можно расставить вагоны в необходимом порядке, и слово «No» — иначе.

**Примеры**

sorting.in	sorting.out
4 10 5 6 3 4	Yes
4 9 5 6 3 4	No