

## Массивы — дополнительные задачи.

Во всех задачах открыты только первые 5 тестов.

### A. Сумма подряд идущих элементов

Дан массив, заполненный целыми числами и два числа  $A$  и  $k$ .

Найти минимальное значение порядкового номера элемента  $i$  (порядковый номер начинается с единицы, а не с нуля, как индекс массива), для которого  $x_i + x_{i+1} + \dots + x_{i+k} = A$

Если такого значения нет, вывести 0.

На вход программе даётся четыре строки.

В первой записано число  $N < 10^5$  — количество элементов массива, во второй число  $k \leq N$ , в третьей число  $A$ , в четвёртой вводятся  $N$  чисел, разделённых пробелами.

Программа должна вывести одно натуральное число — номер первого элемента, начиная с которого сумма  $k + 1$  элемента массива равна  $A$  или число 0, если такого элемента нет.

Input	Output
4	1
1	
22	
9 13 10 -11	

### B. Слияние упорядоченных массивов

Даны массивы  $a$  и  $b$ , содержащие  $N$  и  $M$  элементов соответственно, упорядоченные по неубыванию. Создать упорядоченный по неубыванию массив  $c$  размера  $N + M$ , содержащий элементы данных массивов с учётом кратности.

Нужен алгоритм с суммарным количеством операций (сравнения и присваивания), пропорциональным  $n + m$ .

В первой строке дано число  $N$ , в следующей — неубывающая последовательность из  $N$  чисел через пробел, затем число  $M$  и в последней строке неубывающая последовательность из  $M$  чисел через пробел.

Требуется вывести массив чисел в соответствии с условием.

Input	Output
3	1 2 4 4 5 5 7
1 5 7	
4	
2 4 4 5	

### C. Partition

Дан массив  $x$ , содержащий  $N$  элементов и число  $k$  ( $0 \leq k < N$ ) — индекс элемента в этом массиве. Элемент  $x[k]$  называется *опорным*.

В массиве переставить все элементы, строго меньшие, чем  $x[k]$  в начало, равные  $x[k]$  в середину, а строго большие  $x[k]$  — в конец. Относительный порядок элементов левее и правее опорного не имеет значения. Время работы программы должно быть пропорционально длине списка. Вспомогательные массивы использовать нельзя.

Такая процедура называется *partition* (разбиение) и используется в алгоритме быстрой сортировки (которая так и называется по английски — *quicksort*).

В первой строке дано число  $N$ , в следующей — последовательность из  $N$  чисел через пробел.

Требуется вывести массив чисел в соответствии с условием.

Input	Output
9	3 2 3 4 4 4 5 7 6
4 4 3 2 6 7 3 5 4	
1	

D. *Пересечение массивов*

Даны два массива, упорядоченных по неубыванию.

Создайте, а потом выведите массив, содержащий пересечение множеств элементов этих списков, то есть те числа, которые являются элементами обоих списков. Каждое число должно быть выведено один раз, числа упорядочены по возрастанию.

Время работы программы должно быть пропорционально сумме длин массивов. Множества и операции с ними использовать нельзя.

Ввод и вывод организованы так же, как и в задаче В.

Input	Output
5 1 3 4 7 9 5 2 3 7 10 11	3 7

E. *Разность массивов*

Даны два массива чисел, упорядоченных по неубыванию. Требуется создать и вывести массив, содержащий в порядке возрастания те и только те элементы первого массива, которых нет во втором массиве.

Время работы программы должно быть пропорционально сумме длин массивов. Множества и операции с ними использовать нельзя.

Ввод и вывод организованы так же, как и в задаче В.

Input	Output
5 1 2 4 9 10 7 1 3 3 6 8 9 10	2 4

F. *Симметрическая разность массивов*

Даны два массива, упорядоченных по неубыванию. Создать и вывести симметрическую разность этих массивов: упорядоченный по возрастанию массив, содержащий те и только те элементы, которые содержатся ровно в одном из данных массивов.

Время работы программы должно быть пропорционально сумме длин массивов. Множества и операции с ними использовать нельзя.

Ввод и вывод организованы так же, как и в задаче В.

Input	Output
5 1 2 4 9 10 7 1 3 3 6 8 9 10	2 3 4 6 8

G. *Объединение массивов*

Даны два массива, упорядоченных по неубыванию. Создать и вывести объединение этих массивов: упорядоченный по возрастанию массив, содержащий те же числа, что и в данных массивах, но по одному разу.

Время работы программы должно быть пропорционально сумме длин массивов. Множества и операции с ними использовать нельзя.

Ввод и вывод организованы так же, как и в задаче В.

Input	Output
5 1 2 4 9 10 7 1 3 3 6 8 9 10	1 2 3 4 6 8 9 10

## Н. Разложение перестановки на циклы

Дан массив  $x$  из  $N$  элементов,  $0 \leq x[i] < N$ , каждое число встречается *ровно* один раз. Используя вспомогательный массив, разложите данную перестановку на циклы. Выведите элементы каждого цикла в новой строке. Исходный массив изменять запрещается.

*Указание:* используйте вспомогательный массив логических значений для хранения информации об элементе массива: был ли он уже использован в каком-то цикле?

*Входные данные*

На вход подаётся натуральное число  $N$  ( $N \leq 10^5$ ). Затем на вход подаётся строка, в которой через пробел записаны  $N$  разных чисел от 0 до  $N - 1$ .

*Выходные данные*

Требуется выписать разложение данной перестановки на циклы. Циклы должны быть упорядочены лексикографически, причём под номером элемента цикла понимается его индекс в данном массиве.

Input	Output
10 9 4 8 2 5 7 6 1 3 0	9 0 4 5 7 1 8 3 2 6

*Комментарий к тесту:*

- 9 4 8 2 5 7 6 1 3 0 первый цикл
- 9 4 8 2 5 7 6 1 3 0 второй цикл начинается с первого не использованного ранее элемента
- 9 4 8 2 5 7 6 1 3 0 третий цикл — порядок элементов в цикле отличается от порядка в массиве
- 9 4 8 2 5 7 6 1 3 0 четвёртый цикл

Первый цикл начинается с девятки, элемента исходного массива с индексом 0. Второй цикл начинается с четвёрки — первого после девятки неиспользованного в первом цикле элемента массива и т.д.

## I. Шарiki

В одной компьютерной игре игрок выставляет в линию шарiki разных цветов. Когда образуется непрерывная цепочка из трех и более шариков одного цвета, она удаляется из линии. Все шарiki при этом сдвигаются друг к другу, и ситуация может повториться.

Напишите программу, которая по данной ситуации определяет, сколько шариков будет сейчас «уничтожено». Естественно, непрерывных цепочек из трёх и более одноцветных шаров в начальный момент может быть не более одной.

*Входные данные*

На вход подаётся натуральное число  $N$  ( $N \leq 1000$ ). Затем на вход подаётся строка, в которой через пробел записаны  $N$  чисел.

*Выходные данные*

Требуется вывести одно число — ответ на вопрос задачи.

Input	Output
5 1 3 3 3 2	3

Ж. *Максимальная сумма элементов, удалённых не менее, чем на  $K$*

Найти в массиве два числа, сумма которых максимальна и расстояние (разность индексов) между которыми не меньше данного  $K$ . Решить задачу за время, пропорциональное длине массива, дополнительных массивов не использовать.

Если таких пар элементов несколько — вывести ту, чей левый элемент имеет меньший индекс. Если есть несколько пар с одинаковым левым индексом — вывести ту, индекс правого элемента которой меньше.

*Входные данные*

На вход подаётся натуральное число  $N$  ( $N \leq 10^5$ ).

Затем на вход подаётся натуральное число  $K$  ( $0 < K < N$ ).

Далее на вход подаётся строка, в которой через пробел записаны  $N$  целых чисел. Каждое число не превосходит по модулю  $10^9$ .

*Выходные данные*

Программа должна вывести два числа, разделённые пробелом — индексы левого и правого элемента. Элементы должны иметь максимальную сумму, а модуль разности их индексов не меньше  $K$ . Если таких пар индексов несколько, то требуется вывести ту, у которой второй индекс меньше, а при равных правых индексах ту, у которой меньше левый индекс.

Элементы массива индексируются, начиная с нуля.

Input	Output
7	0 3
3	
6 0 9 9 0 0 6	

К. *Максимальная непрерывная сумма - I*

В массиве из  $N$  элементов, заполненном произвольными целыми числами найдите непрерывный фрагмент, сумма чисел в котором максимальна.

Требуется найти такие  $i$  и  $j$  ( $i \leq j$ ), что сумма элементов  $A[i] + A[i + 1] + \dots + A[j]$  будет максимальна.

Если таких пар индексов несколько, то  $j$  должно быть минимально возможным, а при равных  $j$  значение  $i$  должно быть максимально возможным.

На вход подаётся строка, в которой записано натуральное число  $N \leq 1000$ . В следующих  $N$  строках записано по одному целому числу.

Программа должна вывести два числа — ответ на вопрос задачи. Решение должно иметь сложность  $O(N^2)$ .

Input	Output
5	1 2
-1 2 3 -2 2	

Л. *Максимальная непрерывная сумма - II*

Решите предыдущую задачу за время  $O(N)$ ,  $N \leq 10^5$ .

М. *Купи-продай*

Дан массив из  $N$  положительных чисел  $x$ , где  $x_i$  — это цена одной акции в день номер  $i$ . Разрешается сделать одну покупку и одну продажу одной акции (покупка должна предшествовать продаже).

Предложите алгоритм для решения следующей задачи: в какие два (разные) дня следует сделать покупку и продажу, чтобы максимизировать прибыль и каков размер максимальной прибыли? Если решений несколько, то покупку надо сделать как можно раньше. Если с одним временем покупки есть несколько дней, когда можно сделать продажу — то её тоже надо сделать как можно раньше.

Время работы  $O(N)$ , дополнительная память  $O(1)$ .

На вход подаётся две строки. В первой записано натуральное число  $N \leq 3 \cdot 10^5$ . В следующей строке записано  $N$  натуральных чисел, разделённых пробелами.

Программа должна вывести три числа. Первые два — номера дней, когда надо сделать покупку и продажу соответственно. Третье — размер прибыли. Дни нумеруются, начиная с нуля.

Input	Output
16 6 19 17 16 18 17 7 3 3 5 5 2 5 9 7 15	0 1 13

N\* *Найти повторяющийся элемент*

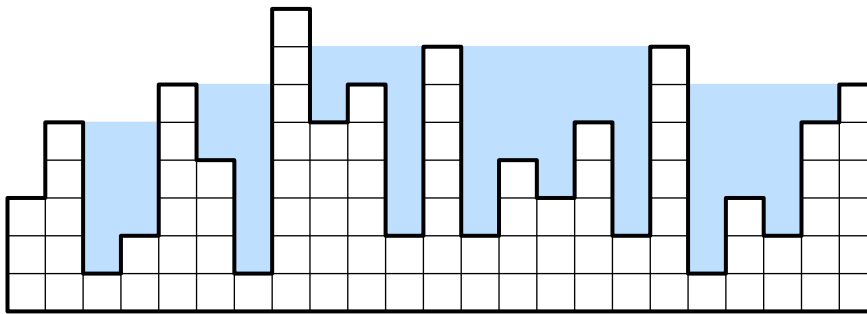
Дан массив размера  $N$  ( $0 < N \leq 2 \cdot 10^5$ ) заполненный числами  $0 \dots N - 2$  (не обязательно всеми из них, в произвольном порядке).

Требуется вывести любое число, которое встречается более одного раза. Массив изменять нельзя, новый массив создавать нельзя. Время работы должно быть пропорционально  $N$ .

Input	Output
6 2 1 0 3 4 3	3

О. *«Расчёска» с водой*

Дан массив натуральных чисел. Представим себе каждое число в виде столбика из соответствующего количества клеток  $1 \times 1$  и составим из таких столбиков фигуру, которую вертикально «опустим» в воду, а потом оттуда вынем (см. рисунок).



Требуется посчитать количество воды, которое осталось, т.е. количество клеток, которое занимает вода, которая не смогла утечь.

На вход подаётся число  $N \leq 10^6$ , а в следующей строке  $N$  натуральных чисел, не превосходящих 1000.

Программа должна вывести одно целое число. Гарантируется, что ответ не превосходит  $10^9$ .

Input	Output
7 5 4 3 2 4 3 2	3

Иллюстрация к примеру:

