

Задача А. Найти сумму

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

Даны массив целых чисел a_1, a_2, \dots, a_n и натуральные числа k и m . Найдите минимальное значение i такое, что $a_i + a_{i+1} + \dots + a_{i+k} = m$. (т.е. сумма $k + 1$ подряд идущих элементов массива равна m).

Формат входных данных

На вход программе сначала подаются значения n , k и m ($0 \leq k \leq n \leq 1\,000\,000$, n — количество элементов в массиве). В следующей строке входных данных расположены сами элементы массива — целые числа, по модулю не превосходящие 100.

Формат выходных данных

Выведите минимальное искомое i . Если такого значения нет, то выведите 0.

Примеры

findsum.in	findsum.out
8 1 5 4 0 5 0 3 2 3 3	2

Задача В. Пересечение массивов

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

Заданы два неупорядоченных массива целых чисел, по модулю не превосходящих 10 000. Будем рассматривать их как множества с повторяющимися элементами. Требуется распечатать пересечение двух множеств. Так, массивы $\{2, 1, 2, 3\}$ и $\{2, 4, 2, 2\}$ имеют пересечение $2, 2$.

Формат входных данных

На вход программе сначала подается значение n ($n \leq 100$) — количество элементов в первом массиве. В следующей строке входных данных расположены элементы первого массива. Далее на вход программе подается значение m ($m \leq 100$) — количество элементов во втором массиве. В следующей строке входных данных расположены элементы второго массива. Элементы в обоих массивах — целые числа, по модулю не превосходящие 10 000.

Формат выходных данных

Выведите общие элементы данных массивов в том порядке, в котором они встречаются в первом массиве.

Примеры

intersec.in	intersec.out
5 2 1 2 3 4 4 2 4 2 2	2 2 4

Задача С. k -е простое

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

По введенному натуральному числу k выдать k -е по счету простое число. Используйте массив для запоминания уже найденных простых чисел.

Формат входных данных

На вход дается одно число k ($k \leq 100\,000$).

Формат выходных данных

Выведите искомое простое число.

Примеры

prime.in	prime.out
1	2
4	7

Задача D. Максимальный кусок

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

В одномерном массиве целых чисел, найдите непрерывный кусок, сумма чисел в котором максимальна. Фактически требуется найти такие i и j ($i \leq j$), что сумма всех элементов массива от a_i до a_j включительно будет максимальна.

Формат входных данных

На вход программе сначала подается значение n ($n \leq 1\,000\,000$) — количество элементов в массиве. В следующей строке входных данных расположены сами элементы массива — целые числа, по модулю не превосходящие 30 000.

Формат выходных данных

Выведите пару искоемых значений индексов. Если таких пар несколько, то j должно быть минимально возможным, а при равных j значение i должно быть максимально возможным.

Примеры

findmax.in	findmax.out
5 -1 2 3 -2 2	2 3
7 2 -2 3 -1 5 -2 7	3 7

Задача Е. Пилообразная последовательность

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

Числовая последовательность называется пилообразной, если каждый ее элемент (кроме первого и последнего) либо больше обоих своих соседей, либо меньше обоих соседей. Например, последовательность 1, 2, 1, 3, 2 является пилообразной, а 1, 2, 3, 1, 2 — нет, поскольку $1 < 2 < 3$. Любая последовательность из одного элемента является пилообразной. Последовательность из двух элементов является пилообразной, если ее элементы не равны.

Дана последовательность. Требуется определить, какое наименьшее количество ее элементов нужно вычеркнуть, чтобы оставшаяся последовательность оказалась пилообразной.

Формат входных данных

На вход программе сначала подается значения n ($1 \leq n \leq 100\,000$) — количество членов последовательности. Во второй строке записано n натуральных чисел, не превосходящих 10 000 — члены последовательности.

Формат выходных данных

Выведите одно число — минимальное количество элементов, которые необходимо вычеркнуть.

Примеры

saw.in	saw.out
5 1 2 3 1 2	1
5 1 2 1 3 2	0
5 1 2 3 4 5	3
5 1 1 2 1 1	2