

Дан массив, содержащий 70 неотрицательных целых чисел. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести наименьшую чётную сумму двух соседних элементов массива. Гарантируется, что в массиве есть соседние элементы с чётной суммой. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Паскаль	Бейсик
<pre>const N=70; var a: array [1..N] of integer; i, j, x, y: integer; begin for i: 1 to N do readln(a[i]); ... end.</pre>	<pre>N=70 DIM A(N) AS INTEGER DIM I, J, X, Y AS INTEGER FOR I=1 TO N INPUT A(I) NEXT I ... END</pre>
Си++	Алгоритмический язык
<pre>#include <iostream> using namespace std; #define N 70 int main(){ int a[N]; int i, j , x, y; for (i = 0; i < N; i++) cin >> a[i]; ... }</pre>	
Python	
<pre># допускается также # использовать три # целочисленные переменные j, x, y a = [] n = 70 for i in range(0, n): a.append(int(input())) ...</pre>	

В качестве ответа Вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.4) или в виде блок-схемы. В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии.