

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на одном из языков программирования алгоритм, который находит количество элементов массива, больших 100 и при этом не кратных 4, а затем заменяет каждый такой элемент на число, равное найденному количеству. Гарантируется, что хотя бы один такой элемент в массиве есть. В качестве результата необходимо вывести изменённый массив, каждый элемент выводится с новой строки.

Например, для исходного массива из шести элементов:

```
141
256
92
148
511
4
```

программа должна вывести следующий массив:

```
2
256
92
148
2
4
```

Исходные данные объявлены так, как показано ниже на примерах для пяти языков программирования. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

Бейсик	Python
<pre>CONST N AS INTEGER = 30 DIM A (1 TO N) AS LONG DIM I AS LONG, J AS LONG, K AS LONG  FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre># допускается также # использовать две # целочисленные переменные j и k a = [] n = 30 for i in range(0, n): a.append(int(input())) ...</pre>
Паскаль	Алгоритмический язык
<pre>const N = 30; var a: array [1..N] of longint; i, j, k: longint; begin for i := 1 to N do readln(a[i]); ... end.</pre>	<pre>алг нач цел N = 30 цел таб a[1:N] цел i, j, k нц для i от 1 до N ввод a[i] кц ...  кон</pre>

C++

```
#include <iostream>
using namespace std;
const int N = 30;
int main() {
    long a[N];
    long i, j, k;
    for (i = 0; i < N; i++)
        cin >> a[i];
    ...
    return 0;
}
```

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 2.6). В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на Алгоритмическом языке).