Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 10000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести произведение всех таких элементов массива, которые являются двузначными числами с чётной суммой цифр. Гарантируется, что в исходном массиве есть хотя бы один элемент, значение которого является двузначным числом, и при этом сумма его цифр чётна. Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных. Исходные данные всегда подобраны так, что результат произведения не выходит за пределы объявленных типов данных.

Бейсик

```
      N = 30
  DIM A(N) AS LONG
DIM I, J, P AS LONG
   FOR I = 1 TO N
     INPUT A(I)
       NEXT I
         ...
        END
```

Паскаль

```
         const
        N = 30;
          var
  a: array [1..N] of
       longint;
   i, j, p: longint;
         begin
  for i := 1 to N do
     readln(a[i]);
           ...
         end.
```

Си++

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

Алгоритмический язык

```
         алг
         нач
    цел N = 30
   целтаб a[1:N]
    цел i, j, p
нц для i от 1 до N
     ввод a[i]
         кц
         ...
         кон
```

Естественный язык

Объявляем массив A из 30 элементов.
Объявляем целочисленные переменные I, J, P.
В цикле от 1 до 30 вводим элементы массива A с 1-го по 30-й. …

Python

```
      # допускается также
       # использовать две
# целочисленные переменные j, p
             a = []
             n = 30
     for i in range(0, n):
    a.append(int(input()))
              ...
```

В качестве ответа Вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), 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 (например, в образце, записанном на естественном языке).