Демонстрационный вариант ЕГЭ 2019 г. – задание №25 — Дан целочисленный **массив из 30 элементов**. Элементы массива могут принимать натуральные значения **от 1 до 10 000 включительно**. Опишите на одном из языков программирования алгоритм, который находит **минимум** среди элементов массива, **не делящихся нацело на 6**, а затем **заменяет каждый элемент**, **не делящийся нацело на 6**, на число, **равное найденному минимуму.**

Гарантируется, что хотя бы один такой элемент в массиве есть. В качестве результата необходимо **вывести изменённый массив**, каждый элемент выводится с новой строчки.
Например, для исходного массива из шести элементов:
14
6
11
18
9
24
программа должна вывести следующий массив
9
6
9
18
9
24

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

Бейсик

```
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
```

Алгоритмический язык

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

C++

Python

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

Паскаль

```
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
```

```
#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)

Решение:

На языке Паскаль

```
k := 10000;
for i := 1 to N do
  if (a[i] mod 6 <> 0) and (a[i] < k) then
    k := a[i];
for i := 1 to N do begin
  if (a[i] mod 6 <> 0) then
    a[i] := k;
  writeln(a[i]);
end;
```

На Алгоритмическом языке

```
k := 10000
нц для i от 1 до N
  если mod(a[i], 6) <> 0 и a[i] < k
  то
    k := a[i]
  все
кц
нц для i от 1 до N
  если mod(a[i], 6) <> 0
  то
    a[i] := k
  все
  вывод a[i], нс
кц
```

На языке Бейсик

```
K = 10000
FOR I = 1 TO N
  IF A(I) MOD 6 <> 0 AND A(I) < K THEN
    K = A(I)
  END IF
NEXT I
FOR I = 1 TO N
```

```
  IF A(I) MOD 6 <> 0 THEN
    A(I) = K
  END IF
  PRINT A(I)
NEXT I
```

На языке C++

```
k = 10000;
for (i = 0; i < N; i++)
  if (a[i] % 6 != 0 && a[i] < k)
    k = a[i];
for (i = 0; i < N; i++) {
  if (a[i] % 6 != 0)
    a[i] = k;
  cout << a[i] << endl;
}
```

На языке Python

```
k = 10000
for i in range(0, n):
    if (a[i] % 6 != 0 and a[i] < k):
        k = a[i]
for i in range(0, n):
    if (a[i] % 6 != 0):
        a[i] = k
    print(a[i])
```

Дан массив, содержащий **2014 положительных целых чисел**. Напишите на одном из языков программирования программу, которая находит в этом массиве **количество элементов**, значение которых более чем **в два раза превосходит значение предшествующего элемента**.

Например, для массива из 6 элементов, содержащего числа 2, 5, 10, 15, 40, 100, программа должна выдать ответ 3 (условию соответствуют элементы со значениями 5, 40 и 100). Программа должна **вывести общее количество подходящих элементов**, значения элементов выводить не нужно.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из описанных переменных.

Pascal

```
const N = 2014;
var a: array [1..N] of integer;
i, j, k: integer;
begin
for i:=1 to N do
  readln(a[i]);
…
end.
```

C++

```
#include<stdio.h>
int main(void) {
const int N=2014;
int a[N];
int i, k, j;
for (i=0; i<N; i++)
  scanf("%d", &a[i]);
…
}
```

Python

```
# допускается также
# использовать две
# целочисленные переменные j и k
a = []
n = 2014
for i in range(0, N):
    a.append(int(input()))
...
```

Решение:

Паскаль

```
k:=0;
for i:=2 to N do
  if a[i] > 2*a[i-1] then
    k:=k+1;
writeln(k);
```

C++

```
k=0;
for (i=1; i<N; i++)
  if (a[i] > 2*a[i-1])
    k=k+1;
cout<<k;
```

Python

```
k = 0
for i in range(1, N):
    if a[i] > 2 * a[i - 1]:
        k += 1
print(k)
```

Дан целочисленный **массив из 40 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 **количество пар** элементов массива, в которых **десятичная запись хотя бы одного числа оканчивается на 5**.

В данной задаче под парой подразумевается два подряд идущих элемента массива.

Pascal

```
const N = 40;
var
  a: array [1..N] of integer;
  i, j, k: integer;
begin
  for i := 1 to N do
    readln(a[i]);
  ...
end.
```

C++

```
#include <stdio.h>
#define N 40
int main() {
  int a[N];
  int i, j, k;
  for (i = 0; i<N; i++)
    scanf("%d", &a[i]);
  ...
  return 0;
}
```

Python

```
# допускается также
# использовать две
# целочисленные переменные j и k
a = []
N = 40
for i in range(0, N):
    a.append(int(input()))
...
```

Решение:

Паскаль

```
k := 0;
for i := 1 to N-1 do
  if (a[i] mod 10 = 5) or (a[i+1] mod 10 = 5) then
    k := k + 1;
writeln(k);
```

C++

```
k = 0;
for (i=0; i<N-1; i++)
  if (a[i] % 10 == 5 || a[i+1] % 10 == 5)
    k = k + 1;
cout<<k;
```

Python

```
k = 0
for i in range(0, N - 1):
    if a[i] % 10 == 5 or a[i + 1] % 10 == 5:
        k += 1
print(k)
```

Дан целочисленный **массив из 40 элементов**. Элементы массива могут принимать целые значения **от 1 до 10000 включительно**. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести **наибольший** из элементов массива, **шестнадцатеричная запись которого оканчивается на букву F**. Если таких чисел нет, нужно вывести ответ .

Pascal

```
const N = 40;
var
  a: array [1..N] of integer;
  i, j, k: integer;
begin
  for i := 1 to N do
    readln(a[i]);
  ...
end.
```

C++

```
#include <stdio.h>
#define N 40
int main() {
  int a[N];
  int i, j, k;
  for (i = 0; i<N; i++)
    scanf("%d", &a[i]);
  ...
  return 0;
}
```

Python

```
# допускается также
# использовать две
# целочисленные переменные j и k
a = []
N = 40
for i in range(0, N):
    a.append(int(input()))
...
```

Алгоритмический язык

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, j, k
  нц для i от 1 до N
    ввод a[i]
  кц
  ...
кон
```

Решение:

Паскаль

```
k := 0;
for i := 1 to N do
  if (a[i] mod 16 = 15) and (a[i] > k) then
    k := a[i];
writeln(k);
```

C++

```
k = 0;
for (i = 0; i<N; i++)
  if (a[i] % 16 == 15 && a[i] > k)
    k = a[i];
cout<<k;
```

Python

```
k = 0
for i in range(0, N):
    if a[i] % 16 == 15 and a[i] > k:
        k = a[i]
print(k)
```

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 1 до 10000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести **наибольший** из элементов массива, **восьмеричная запись которого содержит ровно три цифры**. Если таких чисел нет, нужно вывести ответ .

Pascal

C++

```
const N = 40;
var
  a: array [1..N] of integer;
  i, j, k: integer;
begin
  for i := 1 to N do
    readln(a[i]);
  ...
end.
```

Python

```
# допускается также
# использовать две
# целочисленные переменные j и k
a = []
n = 40
for i in range(0, N):
    a.append(int(input()))
...
```

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, j, k
  нц для i от 1 до N
    ввод a[i]
  кц
  ...
кон
```

Решение:

Паскаль

```
k := 0;
for i := 1 to N do
  if (64 <= a[i]) and (a[i] < 512) and (a[i] > k) then
    k := a[i];
writeln(k);
```

C++

```
k = 0;
for (i = 0; i<N; i++)
  if (64 <= a[i] && a[i] < 512 && a[i] > k)
    k = a[i];
cout<<k;
```

Python

```
k = 0
for i in range(0, N):
    if a[i] >= 64 and a[i] < 512 and a[i] > k:
        k = a[i]
print(k)
```

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 1 до 10000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести **наименьший** из элементов массива, **шестнадцатеричная запись которого содержит не менее трёх цифр и оканчивается на букву С**. Если таких чисел нет, нужно вывести ответ .

Pascal	Алгоритмический язык

```
const N = 40;
var
  a: array [1..N] of integer;
  i, j, k: integer;
begin
  for i := 1 to N do
    readln(a[i]);
  ...
end.
```

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, j, k
  нц для i от 1 до N
    ввод a[i]
  кц
  ...
кон
```

C++

```
#include <stdio.h>
#define N 40
int main() {
  int a[N];
  int i, j, k;
  for (i = 0; i<N; i++)
    scanf("%d", &a[i]);
  ...
  return 0;
}
```

Python

```
#допускается также использовать
#две целочисленные переменные
#j и k
a = []
n = 40
for i in range(n):
  a.append(int(input()))
```

Решение:

Паскаль

```
k := 10001;
for i := 1 to N do
  if (256 <= a[i]) and (a[i] mod 16 = 12) and (a[i] < k) then
    k := a[i];
if k > 10000 then
  writeln(0)
else
  writeln(k);
```

C++

```
k = 10001;
for (i = 0; i<N; i++)
  if (256 <= a[i] && a[i] % 16 = 12 && a[i] < k)
    k = a[i];
if (k > 10000)
  cout<<0;
else
  cout<<k;
```

Python

```
k = 10001
for i in range(0, N):
    if a[i] >= 256 and a[i] % 16 == 12 and a[i] < k:
        k = a[j]
if k > 10000:
    print(0)
else:
    print(k)
```

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 1 до 10000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести **количество элементов массива**, **восьмеричная** запись которого содержит **ровно две цифры**, причём **первая (старшая) цифра меньше второй (младшей)**.

Pascal

```
const N = 40;
var
  a: array [1..N] of integer;
  i, j, k: integer;
begin
  for i := 1 to N do
    readln(a[i]);
  ...
end.
```

Алгоритмический язык

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, j, k
  нц для i от 1 до N
    ввод a[i]
  кц
  ...
кон
```

C++

```
#include <stdio.h>
#define N 40
int main() {
  int a[N];
  int i, j, k;
  for (i = 0; i<N; i++)
    scanf("%d", &a[i]);
  ...
  return 0;
}
```

Python

```
#допускается также использовать
#две целочисленные переменные
#j и k
a = []
n = 40
for i in range(n):
  a.append(int(input()))
```

Решение:

Паскаль

```
k := 0;
for i := 1 to N do
  if (8 <= a[i]) and (a[i] <= 63) and
     (a[i] div 8 < a[i] mod 8) then
    k := k + 1;
writeln(k);
```

C++

```
k = 0;
for (i = 0; i<N; i++)
  if (8 <= a[i] && a[i] <= 63 && (a[i]/8 < a[i]%8))
    k = k + 1;
cout<<k;
```

Python

```
k = 0
for i in range(0, N):
    if a[i] >= 8 and a[i] <= 63 and (a[i] / 8 < a[i] % 8):
        k += 1
```

```
print(k)
```
