

ЗАДАНИЕ 15.2. КОРОТКИЙ АЛГОРИТМ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON



ОСОБЕННОСТИ ЗАДАНИЯ

№ задания	Проверяемые элементы содержания	Уровень сложности	Максимальный балл за выполнение задания	Примерное время выполнения (мин)
15 (2)	Умение написать короткий алгоритм на языке программирования	В	2	45

Предполагаемый результат выполнения заданий высокого уровня – менее 40%.

ОСОБЕННОСТИ ЗАДАНИЯ

- ПЕРЕЧЕНЬ ЭЛЕМЕНТОВ СОДЕРЖАНИЯ, ПРОВЕРЯЕМЫХ ПРИ ВЫПОЛНЕНИИ ДАННОГО ЗАДАНИЯ НА ОСНОВНОМ ГОСУДАРСТВЕННОМ ЭКЗАМЕНЕ ПО ИНФОРМАТИКЕ И ИКТ:
 - 1) АЛГОРИТМ, СВОЙСТВА АЛГОРИТМОВ, СПОСОБЫ ЗАПИСИ АЛГОРИТМОВ. БЛОК-СХЕМЫ. ПРЕДСТАВЛЕНИЕ О ПРОГРАММИРОВАНИИ
 - 2) АЛГОРИТМИЧЕСКИЕ КОНСТРУКЦИИ.
 - 3) ЛОГИЧЕСКИЕ ЗНАЧЕНИЯ, ОПЕРАЦИИ, ВЫРАЖЕНИЯ
 - 4) РАЗБИЕНИЕ ЗАДАЧИ НА ПОДЗАДАЧИ, ВСПОМОГАТЕЛЬНЫЙ АЛГОРИТМ
 - 5) ОБРАБАТЫВАЕМЫЕ ОБЪЕКТЫ: ЦЕПОЧКИ СИМВОЛОВ, ЧИСЛА, СПИСКИ, ДЕРЕВЬЯ



ЗАДАЧИ (ПО ТИПУ ВВОДА ДАННЫХ)

- ИЗВЕСТНО КОЛИЧЕСТВО ЭЛЕМЕНТОВ ПОСЛЕДОВАТЕЛЬНОСТИ:

ПРОГРАММА ПОЛУЧАЕТ НА ВХОД КОЛИЧЕСТВО ЧИСЕЛ В ПОСЛЕДОВАТЕЛЬНОСТИ, А ЗАТЕМ САМИ ЧИСЛА. В ПОСЛЕДОВАТЕЛЬНОСТИ ВСЕГДА ИМЕЕТСЯ ЧИСЛО, КРАТНОЕ N.

- НЕИЗВЕСТНО КОЛИЧЕСТВО ЭЛЕМЕНТОВ ПОСЛЕДОВАТЕЛЬНОСТИ:

ПРОГРАММА ПОЛУЧАЕТ НА ВХОД НАТУРАЛЬНЫЕ ЧИСЛА, КОЛИЧЕСТВО ВВЕДЁННЫХ ЧИСЕЛ НЕИЗВЕСТНО, ПОСЛЕДОВАТЕЛЬНОСТЬ ЧИСЕЛ ЗАКАНЧИВАЕТСЯ ЧИСЛОМ 0 (0 — ПРИЗНАК ОКОНЧАНИЯ ВВОДА, НЕ ВХОДИТ В ПОСЛЕДОВАТЕЛЬНОСТЬ).

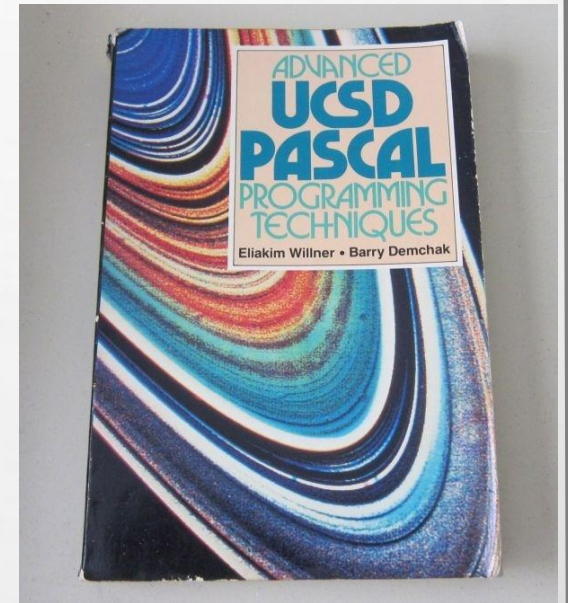
ИСПОЛЬЗУЕТСЯ ОПЕРАТОР ЦИКЛА WHILE (С ПРЕДУСЛОВИЕМ)

*КОЛИЧЕСТВО ЧИСЕЛ НЕ ПРЕВЫШАЕТ A. ВВЕДЁННЫЕ ЧИСЛА НЕ ПРЕВЫШАЮТ B.

ТИПЫ ЗАДАЧ (ПО СОДЕРЖАНИЮ УСЛОВИЯ)

- **НАХОЖДЕНИЕ СУММЫ ВСЕХ НАТУРАЛЬНЫХ ЧИСЕЛ, КРАТНЫХ a (ОКАНЧИВАЮЩИХСЯ НА b)**
- **НАХОЖДЕНИЕ СРЕДНЕГО АРИФМЕТИЧЕСКОГО (ПО УКАЗАННОМУ УСЛОВИЮ)**
- **НАХОЖДЕНИЕ КОЛИЧЕСТВА ЧИСЕЛ, КРАТНЫХ a (ОКАНЧИВАЮЩИХСЯ НА a / КОЛИЧЕСТВА ЧИСЕЛ ВСЕЙ ПОСЛЕДОВАТЕЛЬНОСТИ);**
- **НАХОЖДЕНИЕ МИНИМАЛЬНОГО (МАКСИМАЛЬНОГО / ЧИСЛА, ОКАНЧИВАЮЩЕЕСЯ НА a) ИЛИ ЕГО ПОРЯДКОВОГО НОМЕРА**

УСЛОВИЕ КРАТНОСТИ (ТОГО, ЧТО ЧИСЛО ОКАНЧИВАЕТСЯ НА ЗАДАННУЮ ЦИФРУ)



• **ОСТАТОК ОТ ДЕЛЕНИЯ!**

IF A % 2 == 0

(ЧИСЛО ЧЕТНОЕ)

IF A % 10 == N

(ЧИСЛО ОКАНЧИВАЕТСЯ НА N)

IF A % N == 0

(ЧИСЛО КРАТНО N)



Программирование циклов с заданным числом повторений

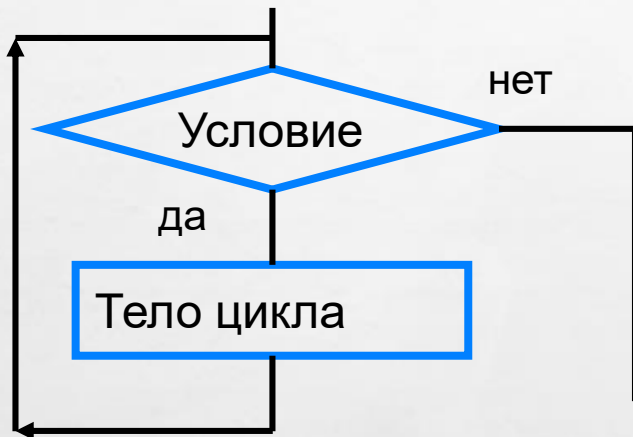


<счётчик цикла> -
переменная целого типа;
<начальное_значение> и
<конечное_значение> -
выражения того же типа, что
и счётчик;
<тело цикла> - простой или
составной оператор.

Общий вид оператора:

```
for i in range <конечное_значение> :  
<тело цикла>
```

Программирование циклов с заданным условием продолжения работы (оператор с предусловием)



Общий вид оператора:

```
while <условие> :  
<тело цикла>
```

Здесь:

<условие> - логическое выражение; пока оно истинно, выполняется тело цикла;

< тело цикла > - простой или составной оператор.



Программа вывода на экран целых чисел из интервала от 1 до n
(использование цикла **for**)

```
n=int(input())  
for i in range n:  
    print (' ',i)
```



Программа вывода на экран целых чисел из интервала от 1 до n
(использование оператора цикла **while**)

```
n=int(input())  
i:=1  
while i<=n:  
    print (' ',i)  
    i:=i+1
```

НАХОЖДЕНИЕ СУММЫ ВСЕХ НАТУРАЛЬНЫХ ЧИСЕЛ, КРАТНЫХ A (ОКАНЧИВАЮЩИХСЯ НА B)

- **ИВ! ВВОДИТСЯ ПЕРЕМЕННАЯ S (СУММА ЭЛЕМЕНТОВ МАССИВА, УДОВЛЕТВОРЯЮЩИХ ЗАДАННОМУ УСЛОВИЮ)**
- **ПРИМЕР ЗАДАЧИ:** НАПИШИТЕ ПРОГРАММУ, КОТОРАЯ В ПОСЛЕДОВАТЕЛЬНОСТИ НАТУРАЛЬНЫХ ЧИСЕЛ ОПРЕДЕЛЯЕТ **СУММУ ЧИСЕЛ**, КРАТНЫХ 6. ПРОГРАММА ПОЛУЧАЕТ НА ВХОД КОЛИЧЕСТВО ЧИСЕЛ В ПОСЛЕДОВАТЕЛЬНОСТИ, А ЗАТЕМ САМИ ЧИСЛА. В ПОСЛЕДОВАТЕЛЬНОСТИ ВСЕГДА ИМЕЕТСЯ ЧИСЛО, КРАТНОЕ 6. КОЛИЧЕСТВО ЧИСЕЛ НЕ ПРЕВЫШАЕТ 100. ВВЕДЁННЫЕ ЧИСЛА НЕ ПРЕВЫШАЮТ 300. ПРОГРАММА ДОЛЖНА ВЫВЕСТИ ОДНО ЧИСЛО — СУММУ ЧИСЕЛ, КРАТНЫХ 6.



Входные данные	Выходные данные
3	
12	
25	
6	18

⚠! Вводится переменная s (сумма элементов массива, удовлетворяющих заданному условию)

Решение:

```
program pr;  
var n, s, k, g: integer;  
begin  
s:=0;  
readln(n);  
for k:=1 to n do  
begin  
readln(g);  
if (g mod 6 = 0) then s:=s+g;  
end;  
writeln(s);  
end.
```

```
•Program1.pas*  
program pr;  
var n, s, k, g: integer;  
begin  
s:=0;  
readln(n);  
for k:=1 to n do  
begin  
readln(g);  
if (g mod 6 = 0) then s:=s+g;  
end;  
writeln(s);  
end.
```

Окно вывода

```
3  
12  
25  
6  
18
```

Окно вывода Сообщения компилятора

НАХОЖДЕНИЕ СУММЫ ВСЕХ НАТУРАЛЬНЫХ ЧИСЕЛ, КРАТНЫХ A (ОКАНЧИВАЮЩИХСЯ НА B)

- **ИВ! ВВОДИТСЯ ПЕРЕМЕННАЯ S (СУММА ЭЛЕМЕНТОВ МАССИВА, УДОВЛЕТВОРЯЮЩИХ ЗАДАННОМУ УСЛОВИЮ)**
- **ПРИМЕР ЗАДАЧИ:** НАПИШИТЕ ПРОГРАММУ, КОТОРАЯ В ПОСЛЕДОВАТЕЛЬНОСТИ НАТУРАЛЬНЫХ ЧИСЕЛ ОПРЕДЕЛЯЕТ **СУММУ ВСЕХ ЧИСЕЛ**, КРАТНЫХ 3 И ОКАНЧИВАЮЩИХСЯ НА 4. ПРОГРАММА ПОЛУЧАЕТ НА ВХОД НАТУРАЛЬНЫЕ ЧИСЛА, КОЛИЧЕСТВО ВВЕДЁННЫХ ЧИСЕЛ НЕИЗВЕСТНО, ПОСЛЕДОВАТЕЛЬНОСТЬ ЧИСЕЛ ЗАКАНЧИВАЕТСЯ ЧИСЛОМ 0 (0 — ПРИЗНАК ОКОНЧАНИЯ ВВОДА, НЕ ВХОДИТ В ПОСЛЕДОВАТЕЛЬНОСТЬ). КОЛИЧЕСТВО ЧИСЕЛ НЕ ПРЕВЫШАЕТ 1000. ВВЕДЁННЫЕ ЧИСЛА НЕ ПРЕВЫШАЮТ 30000. ПРОГРАММА ДОЛЖНА ВЫВЕСТИ ОДНО ЧИСЛО: СУММУ ВСЕХ НАТУРАЛЬНЫХ ЧИСЕЛ, КРАТНЫХ 3 И ОКАНЧИВАЮЩИХСЯ НА 4.

Входные данные	Выходные данные
16	78
24	
56	
22	
54	
0	

Решение:

```
program pr2;  
var a, s: integer;  
begin  
s := 0;  
readln(a);  
while a <> 0 do begin  
if (a mod 3 = 0) and (a mod 10 = 4) then  
s := s + a;  
readln(a);  
end;  
writeln(s)  
end.
```



ИВ! Вводится переменная s (сумма элементов массива, удовлетворяющих заданному условию)

The screenshot shows the PascalABC.NET IDE interface. The main window displays the source code for 'Program1.pas', which is identical to the code provided in the 'Решение' section. The code is color-coded: keywords are in black, variables are in blue, and literals are in green. The output window, titled 'Окно вывода', is open and shows the results of the program's execution: 16, 24, 56, 22, 54, 0, and 78, each on a new line. The IDE's menu bar includes 'Файл', 'Правка', 'Вид', 'Программа', 'Сервис', 'Модули', and 'Помощь'. The toolbar contains various icons for file operations, editing, and execution. The status bar at the bottom shows 'Окно вывода' and 'Список ошибок'.

Окно вывода

```
16  
24  
56  
22  
54  
0  
78
```

Окно вывода Список ошибок

НАХОЖДЕНИЕ СРЕДНЕГО АРИФМЕТИЧЕСКОГО (ПО УКАЗАННОМУ УСЛОВИЮ)

- **ИВ!** ВВОДИТСЯ ПЕРЕМЕННЫЕ S (СУММА ЭЛЕМЕНТОВ МАССИВА, УДОВЛЕТВОРЯЮЩИХ ЗАДАННОМУ УСЛОВИЮ), ПОТОМ НАХОДИТСЯ СРЕДНЕЕ АРИФМЕТИЧЕСКОЕ $SR:=S/N$ (N -КОЛИЧЕСТВО ЭЛЕМЕНТОВ МАССИВА)
- **ПРИМЕР ЗАДАЧИ:** НАПИШИТЕ ПРОГРАММУ ДЛЯ РЕШЕНИЯ СЛЕДУЮЩЕЙ ЗАДАЧИ. КАМЕРА НАБЛЮДЕНИЯ РЕГИСТРИРУЕТ В АВТОМАТИЧЕСКОМ РЕЖИМЕ СКОРОСТЬ ПРОЕЗЖАЮЩИХ МИМО НЕЁ АВТОМОБИЛЕЙ, ОКРУГЛЯЯ ЗНАЧЕНИЯ СКОРОСТИ ДО ЦЕЛЫХ ЧИСЕЛ. **НЕОБХОДИМО ОПРЕДЕЛИТЬ СРЕДНЮЮ ЗАРЕГИСТРИРОВАННУЮ СКОРОСТЬ** ВСЕХ АВТОМОБИЛЕЙ, ОКРУГЛИТЬ ЗНАЧЕНИЕ ДО ЦЕЛЫХ ЧИСЕЛ. ЕСЛИ НЕ МЕНЕЕ ДВУХ АВТОМОБИЛЕЙ ДВИГАЛИСЬ СО СКОРОСТЬЮ НЕ БОЛЬШЕ 40 КМ/Ч, ВЫВЕДИТЕ «YES», ИНАЧЕ ВЫВЕДИТЕ «NO». ПРОГРАММА ПОЛУЧАЕТ НА ВХОД ЧИСЛО ПРОЕХАВШИХ АВТОМОБИЛЕЙ N ($1 \leq N \leq 30$), ЗАТЕМ УКАЗЫВАЮТСЯ ИХ СКОРОСТИ. ЗНАЧЕНИЕ СКОРОСТИ НЕ МОЖЕТ БЫТЬ МЕНЬШЕ 1 И БОЛЬШЕ 300.ПРОГРАММА ДОЛЖНА СНАЧАЛА ВЫВЕСТИ СРЕДНЮЮ СКОРОСТЬ, ЗАТЕМ «YES» ИЛИ «NO».

Входные данные	Выходные данные
4	
74	
69	76
63	NO
96	

Решение

```
program pr3;
var N, a, i, num: integer; s, sr: real;
begin
s := 0; num := 0;
readln(N);
for i := 1 to N do
begin
readln(a); s := s+a;
if a <= 40 then num := num + 1;
end;
sr := s/N;
writeln(sr:6:0);
if num >= 2 then writeln('YES')
else writeln('NO')
end.
```

Внимание! Вводятся переменные s (сумма элементов массива, удовлетворяющих заданному условию), потом находится среднее арифметическое $sr:=s/n$ (n - количество элементов массива)

The screenshot shows a Pascal IDE with the following source code in the main window:

```
•Program1.pas*
readln(N);
for i:= 1 to N do
begin;
readln(a); s:= s+a;
if a <= 40 then num:= num + 1;
end;
sr:= s/N;
writeln(sr:6:0);
if num >= 2 then writeln('YES')
else writeln('NO')
end.
```

Three 'Окно вывода' (Output Window) windows are shown, each displaying the results of a program run with different input values:

- Window 1: Input 1, sum 35, average 35, output NO.
- Window 2: Input 3, sum 150, average 60, output YES.
- Window 3: Input 2, sum 80, average 40, output YES.

There is also a fourth, larger 'Окно вывода' window in the background showing a list of numbers: 4, 74, 69, 63, 96, 76, NO.

НАХОЖДЕНИЕ КОЛИЧЕСТВА ЧИСЕЛ, КРАТНЫХ A (ОКАНЧИВАЮЩИХСЯ НА A / КОЛИЧЕСТВА ЧИСЕЛ ВСЕЙ ПОСЛЕДОВАТЕЛЬНОСТИ)

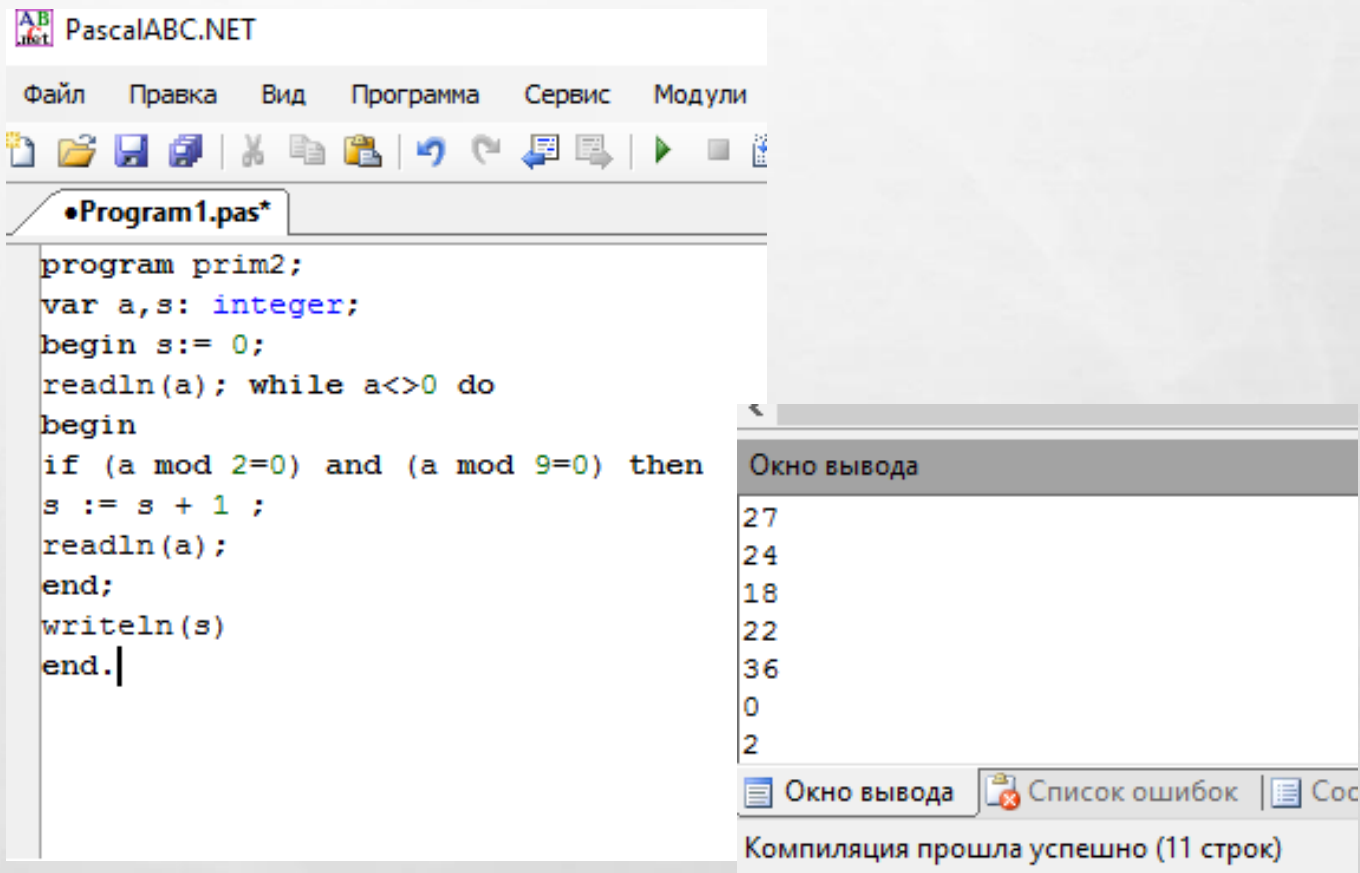
- **ИВ!** ВВОДИТСЯ ПЕРЕМЕННАЯ S (СЧЕТЧИК - СЧИТАЕТ КОЛИЧЕСТВО ЭЛЕМЕНТОВ МАССИВА, УДОВЛЕТВОРЯЮЩИХ ЗАДАННОМУ УСЛОВИЮ)
- **ПРИМЕР ЗАДАЧИ:** НАПИШИТЕ ПРОГРАММУ, КОТОРАЯ В ПОСЛЕДОВАТЕЛЬНОСТИ НАТУРАЛЬНЫХ ЧИСЕЛ ОПРЕДЕЛЯЕТ **КОЛИЧЕСТВО ВСЕХ ЧЁТНЫХ ЧИСЕЛ**, КРАТНЫХ 9. ПРОГРАММА ПОЛУЧАЕТ НА ВХОД НАТУРАЛЬНЫЕ ЧИСЛА, КОЛИЧЕСТВО ВВЕДЁННЫХ ЧИСЕЛ НЕИЗВЕСТНО, ПОСЛЕДОВАТЕЛЬНОСТЬ ЧИСЕЛ ЗАКАНЧИВАЕТСЯ ЧИСЛОМ 0 (0 — ПРИЗНАК ОКОНЧАНИЯ ВВОДА, НЕ ВХОДИТ В ПОСЛЕДОВАТЕЛЬНОСТЬ). КОЛИЧЕСТВО ЧИСЕЛ НЕ ПРЕВЫШАЕТ 1000. ВВЕДЁННЫЕ ЧИСЛА НЕ ПРЕВЫШАЮТ 30 000. ПРОГРАММА ДОЛЖНА ВЫВЕСТИ ОДНО ЧИСЛО: КОЛИЧЕСТВО ВСЕХ ЧЁТНЫХ ЧИСЕЛ, КРАТНЫХ 9.

Входные данные	Выходные данные
27	2
24	
18	
22	
36	
0	

Внимание! Вводится переменная s (счетчик - считает количество элементов массива, удовлетворяющих заданному условию)

Решение

```
program prim2;  
var a, s: integer;  
begin s:= 0;  
readln(a); while a<>0 do  
begin  
if (a mod 2=0) and (a mod 9=0) then  
s := s + 1 ;  
readln(a);  
end;  
writeln(s)  
end.
```



The screenshot shows the PascalABC.NET IDE with a menu bar (Файл, Правка, Вид, Программа, Сервис, Модули) and a toolbar. The main window displays the Pascal code from the previous block. A separate 'Окно вывода' (Output Window) is open, showing the output of the program: 27, 24, 18, 22, 36, 0, 2. At the bottom, a status bar indicates 'Компиляция прошла успешно (11 строк)' (Compilation successful (11 lines)).

```
program prim2;  
var a,s: integer;  
begin s:= 0;  
readln(a); while a<>0 do  
begin  
if (a mod 2=0) and (a mod 9=0) then  
s := s + 1 ;  
readln(a);  
end;  
writeln(s)  
end.
```

Окно вывода

27
24
18
22
36
0
2

Окно вывода | Список ошибок | Со...

Компиляция прошла успешно (11 строк)

НАХОЖДЕНИЕ МИНИМАЛЬНОГО (МАКСИМАЛЬНОГО / ЧИСЛА, ОКАНЧИВАЮЩЕЕСЯ НА А) ИЛИ ЕГО ПОРЯДКОВЫЙ НОМЕР

- **ИВ! ОРГАНИЗУЕТСЯ ПЕРЕБОР ВСЕХ ЭЛЕМЕНТОВ, НАЧИНАЯ С ПЕРВОГО С ЦЕЛЬЮ НАЙТИ МИНИМАЛЬНЫЙ (МАКСИМАЛЬНЫЙ/**

*ИЗ ДЕМОВЕРСИИ 2019

20.2 Напишите программу, которая в последовательности натуральных чисел определяет минимальное число, оканчивающееся на 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 4. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число – минимальное число, оканчивающееся на 4.

Пример работы программы:

Входные данные	Выходные данные
3	14
24	
14	
34	



```

var n,i,a,min: integer;
begin
  readln(n);
  min := 30001;
  for i := 1 to n do
    begin
      readln(a);
      if (a mod 10 = 4) and (a < min)
        then min := a;
    end;
  writeln(min)
end.

```

№	Входные данные	Выходные данные
1	3 2 4 19	4
2	3 14 24 44	14
3	3 24 44 4	4

Pascal