

# Часть 2.

## Задания с развернутым ответом

Ковалев С.Н.  
учитель информатики «Лицей №67»

Номера заданий	Проверяемые требования	Уровень сложности	Кол-во баллов
№24 (C1)	Умение прочесть фрагмент программы на языке программирования и исправить допущенные ошибки.	Повышенный	3
№25 (C2)	Умение написать короткую (10-15 строк) простую программу (например, обработки массива) на языке программирования или записать алгоритм на естественном языке.	Высокий	2
№27 (C4)	Умение создавать собственные программы (30-50 строк) для решения задач средней сложности.	Высокий	4

# Задание №25 (С2)

## Алгоритмы обработки массивов

[Вычисление средних значений, суммы элементов](#)

[Поиск максимального элемента](#)

[Поиск минимального элемента](#)

[Другие алгоритмы](#)

# Поиск максимального (минимального) элемента массива



**Что взять за начальный максимум?**

**Что взять за начальный минимум?**

**Напоминаем!**

**За начальный максимум следует брать минимально возможное число.**

**За начальный минимум следует брать максимально возможное число.**

## Задача.

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения **от -10 000 до 10 000** включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести максимальный элемент массива, который делится на 5 (**наличие хотя бы одного такого элемента гарантируется**).

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**.

Алгоритмический язык	Паскаль
алг	const
нач	N = 20;
цел N = 20	var
целтаб а[1:N]	a: array [1..N] of integer;
цел i, j, k	i, j, k: integer;
нц для i от 1 до N	begin
ввод а[i]	for i := 1 to N do
кц	readln(a[i]);
...	...
кОН	end.

# Правильное решение задачи

```
const
  N = 20;
var
  a: array [1..N] of integer;
  i, j, k: integer;
begin
  for i := 1 to N do // ввод элементов массива
    readln(a[i]);

  j:= -10000; //начальный максимум

  // Поиск максимального элемента

  for i:=1 to N do
    if (a[i] mod 5 = 0) and (a[i] > j) then
      j:= a[i];

  write(j); //вывод результата
end.
```

# Поиск максимального (минимального) элемента массива

Дан массив, содержащий 70 целых чисел. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести наименьшее положительное нечетное число, содержащееся в массиве. Гарантируется, что в массиве есть хотя бы одно положительное нечетное число. Исходные данные объявлены так, как показано ниже.

Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

## Паскаль

```
const N=70;
var
a: array [1..N] of integer;
i, j, m: integer;
begin
  for i:=1 to N do
    readln(a[i]);
  ...
end.
```



Что взять за начальный минимум?

# Правильное решение задачи

```
const N=70;
var
a: array [1..N] of integer;
i, j, m: integer;
begin
  for i:=1 to N do
    read(a[i]);

  for i:=1 to N do
    if (a[i] > 0) and (a[i] mod 2 <> 0) then
      begin
        m:= a[i];
        break;

      for j:=i+1 to N do
        if (a[j] > 0) and (a[j] mod 2 <> 0) and (a[j] < m) then
          m:= a[j];

      write(m);
    end.
```

{Ищем начальный  
минимум}



# Решения учащихся

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от **-10 000 до 10 000** включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести максимальный элемент массива, который делится на 5 (**наличие хотя бы одного такого элемента гарантируется**).

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**.

Алгоритмический язык	Паскаль
алг	const
нач	N = 20;
цел N = 20	var
целтаб a[1:N]	a: array [1..N] of integer;
цел i, j, k	i, j, k: integer;
нц для i от 1 до N	begin
ввод a[i]	for i := 1 to N do
кц	readln(a[i]);
...	...
кон	end.

# Решения учащихся

25.  $k := 0$

for  $i := 1$  to  $N$  do

begin if  $a[i] \bmod 5 = 0$  then

$j := a[i];$

if  $j > k$  then

$k := j;$

end;

write ( $k$ )

end.

1. Неверно задано начальное значение переменной  $k$ .

$k := 10000;$

2. Переменную  $j$  можно было не использовать:

**if  $a[i] \bmod 5 = 0$  then**

**if  $a[i] > k$  then**

**$k := a[i];$**

# Решения учащихся

```
25) k := -11111;  
for i := 1 to N do begin  
  if a[i] mod 5 = 0 then begin  
    if a[i] > k then begin  
      j := i;  
      k := a[i];  
    end;  
  end;  
end;  
writeLn(a[j]);  
end.
```

В программе зачем-то определяется номер максимального элемента массива (скорее всего это делается, чтобы задействовать переменную j).



# Решения учащихся

Pascal ABC net

```
j := -10000;  
for i := 1 to N do begin  
  if (a[i] mod 5 = 0) and (a[i] > j) then  
    j := a[i];  
end;  
write  
writeln(a[i]);
```

А что выводим ?



# Решения учащихся

```
25. const
    N = 20;
var a: array [1..N] of integer;
    i, j, k: integer;
begin
    for i := 1 to N do
        readln(a[i]);
        k := -10000
        for j := 1 to N do
            begin
                if (a[i] mod 5 = 0) and (a[i] > k) then
                    k := a[i]
            end;
        writeln(k);
    end.
```

В этом решении попытка задействовать все переменные приводит к ошибке



# Решения учащихся

```
N 25
k := -10000;
For i := 1 to N do
  if a[i] > k and a[i] mod 5 = 0 then
k := a[i]; k := a[i]; end;
writeln (k);
```

1. Нет скобок в условном операторе.
2. Оператор end не имеет begin.

# Решения учащихся

```
25. k := -10000;  
for j:=1 to N do  
begin  
  if (a[j] > k) and (a[j] mod 5 = 0) then  
    k = a[j];  
end;  
writeln(k);
```

і и j – близнецы-братья?

# Задачи на второй (третий) максимум (минимум)

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать произвольные значения. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит значение второго максимума (элемента, который в отсортированном по невозрастанию массиве стоял бы вторым).

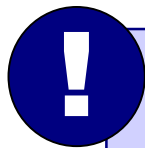
Дан целочисленный массив из 30 элементов. Элементы массива могут принимать произвольные целые значения. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит номера двух элементов массива, сумма которых минимальна. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.



# Контроль выхода за границы массива

Опишите на русском языке или одном из языков программирования алгоритм поиска трех последовательных элементов, сумма которых максимальна, в числовом массиве из 30 элементов.

```
Max := a[1] + a[2] + a[3];  
For i:=2 to N-2 do  
  If a[i] + a[i+1] + a[i+2] < Max then  
    Max:= a[i] + a[i+1] + a[i+2] <  
    Max;  
Write(Max);
```



Цикл должен заканчиваться не на N, а на N-2, чтобы не было выхода за границы массива!

# Использование двойного цикла

Дан целочисленный массив из 30 элементов. Элементы массива могут принимать произвольные целые значения. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит номера двух элементов массива, наименее отличающихся друг от друга. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

# За что снимают баллы?

- Задано неверное начальное значение переменных (или вообще не задано)
- Неверно указано условие завершения цикла
- "Забыли" изменять переменную цикла в цикле while (repeat)
- Перепутаны знаки < и >, логические операции or и and
- Неверно расставлены операторные скобки begin-end
- Программа не выводит результат или выводит не то, что спрашивают
- Синтаксические ошибки (знаки пунктуации – запятые, точки, точки с запятой; неверное написание ключевых слов) допускаются в разумных пределах (если они не искажают замысел автора)
- Не разрешается использовать стандартные процедуры и функции из библиотек (кроме функций чтения и записи)

## Задание №27 (С4)

Сложная задача на программирование

Для заданной последовательности вещественных чисел необходимо найти минимальное среднее арифметическое двух её элементов, номера которых различаются не менее чем на 9. Значение каждого элемента последовательности не превышает 1000. Количество элементов последовательности не превышает 10000.

Входные данные представлены следующим образом. В первой строке задаётся число  $N$  – общее количество элементов последовательности .

Гарантируется, что  $N > 9$ . В каждой из следующих  $N$  строк задаётся одно неотрицательное целое число – очередной элемент последовательности .

*Пример входных данных:*

10

8

10

45

-55

25

35

25

14

16

9

Программа должна вывести одно число – описанное в условии среднее арифметическое.

*Пример выходных данных для приведённого выше примера входных данных: **8.5***

*Вам предлагаются два задания, связанные с этой задачей: задание А и задание Б. Вы можете решать оба задания А и Б или одно из них по своему выбору.*

□ *Итоговая оценка выставляется как **максимальная** из оценок за задания А и Б. Если решение одного из заданий не представлено, то считается, что оценка за это задание составляет 0 баллов.*

*Задание Б является усложненным вариантом задания А, оно содержит дополнительные требования к программе.*

**А.** Напишите на любом языке программирования программу для решения поставленной задачи, в которой входные данные будут запоминаться в массиве, после чего будут проверены все возможные пары элементов. Перед программой укажите версию языка программирования. **Обязательно** укажите, что программа является решением **задания А**.

Максимальная оценка за выполнение задания А – 2 балла.

**Б.** Напишите программу для решения поставленной задачи, которая будет эффективна как по времени, так и по памяти (или хотя бы по одной из этих характеристик). Программа считается эффективной по времени, если время работы программы пропорционально количеству элементов последовательности  $N$ , т.е. при увеличении  $N$  в  $k$  раз время работы программы должно увеличиваться не более чем в  $k$  раз.

**Обязательно** укажите, что программа является решением задания Б.

Максимальная оценка за правильную программу, эффективную по времени и по памяти, – 4 балла. Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, – 3 балла.



Напоминаем! Не забудьте указать, к какому заданию относится каждая из представленных Вами программ.

# Решение для задания А (на 2 балла)

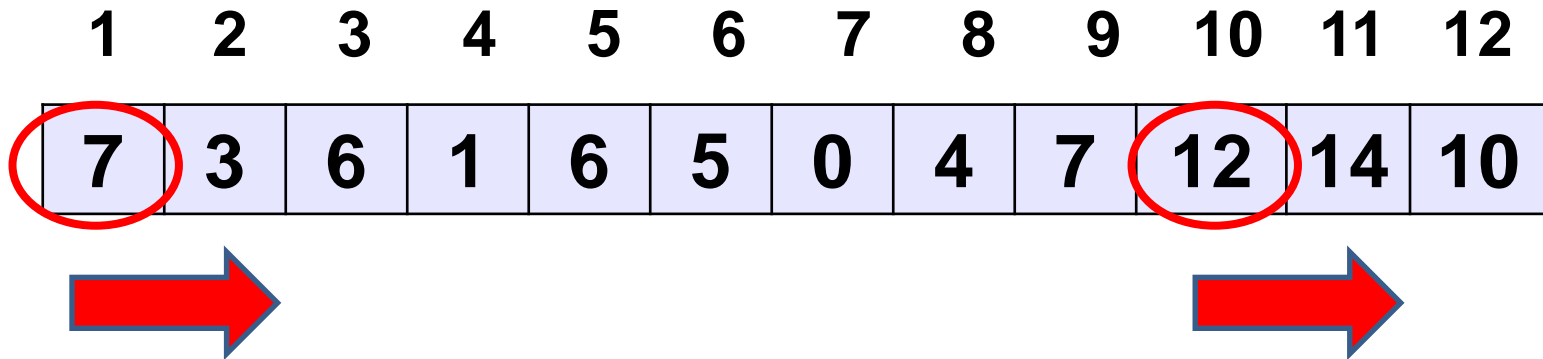
	Паскаль
<p>Пример правильной программы на языке Паскаль, не эффективной ни по времени, ни по памяти.</p>	<pre>Const d = 9; var N: integer; a: array[1..10000] of real; {хранение всех элементов} m: real;      {минимальное значение суммы} i, j: integer; begin   readln(N);   for i:=1 to N do readln(a[i]); {Ввод значений элементов}    m := 2001.0;    for i := 1 to N-d do     for j := i+d to N do       if a[i] + a[j] &lt; m then m := a[i] + a[j];    writeln(m/2.0) end.</pre>



# Решение для задания Б (на 3 балла)

	Паскаль
Пример правильной программы на языке Паскаль, эффективной по времени, но неэффективной по памяти.	<pre>const d = 9; var N: integer; a: array[1..10000] of real;    {хранение всех элементов} mn:real;    {минимальное введенное число}            {не считая d последних} m:real;    {минимальное значение суммы} i: integer; begin   readln(N);{Ввод всех элементов последовательности}   for i:=1 to N do readln(a[i]);   mn := 1001.0;   m := 2001.0;   for i := d + 1 to N do   begin     if a[i-d] &lt; mn then mn := a[i-d];     if a[i] + mn &lt; m then m := a[i] + mn   end;   writeln(m/2.0) end.</pre>

# Разбор решение задания Б (на 3 балла)



При  $i = 10$   
 $a[1] = 7$   $mn = 7$   $m = a[10] + 7 = 19$

При  $i = 11$   
 $a[2] = 3$   $mn = 3$   $m = a[11] + 3 = 17$

При  $i = 12$   
 $a[3] = 6$   $mn = 3$   $m = a[12] + 3 = 13$

```
mn := 1001.0;  
m := 2001.0;  
for i := 10 to N do  
begin  
    if a[i - 9] < mn then  
        mn := a[i - 9];  
    if a[i] + mn < m then  
        m := a[i] + mn  
end;
```

# Решение для задания Б (на 4 балла)

Правильная программа на языке Паскаль, эффективная по времени по памяти должна использовать минимальный по объему массив для хранения 9 последних минимумов, который циклически заполняется по мере ввода данных. Так как размер этого массива не зависит от общего количества введенных элементов, то данное решение будет эффективным по памяти.

Ссылка на разбор подобной задачи на сайте Полякова

[Материалы семинара в г. Тольятти, архив ZIP  
\(6 580 Кб\)](#)

# За что снимают баллы в задаче №27?

4 балла ставилось, если программа была эффективной, но допускалось несколько (до 3-5 синтаксических ошибок), которые были бы выловлены при компиляции на компьютере. **Если ошибок было уже более, то уже снимался один балл.**

Если были ошибки **в использовании алгоритмических конструкций** в совокупности с синтаксическими, то уже снималось 2 балла

Если же были **логические ошибки** с учетом ошибок, указанных выше, но **в программе присутствовала рациональная часть**, в отдельных случаях можно выставлять один балл.

# Задание №24 (С1)

## Исправление ошибок в программе

На обработку поступает натуральное число, не превышающее 109. Нужно написать программу, которая выводит на экран минимальную чётную цифру числа. Если в числе нет чётных цифр, требуется на экран вывести «NO». Программист написал программу неправильно.

```
var N,digit,minDigit: longint;  
begin  
  readln(N);  
  minDigit := N mod 10;  
  while N > 0 do  
  begin  
    digit := N mod 10;  
    if digit mod 2 = 0 then  
      if digit < minDigit then  
        minDigit := digit;  
    N := N div 10;  
  end;  
  if minDigit = 0 then  
    writeln('NO')  
  else  
    writeln(minDigit)  
End.
```

# Задание №24 (С1)

## Исправление ошибок в программе

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 231.

2. Приведите пример такого трёхзначного числа, при вводе которого программа выдаёт верный ответ.

3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:

1) выпишите строку, в которой сделана ошибка;

2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

## Указания по оцениванию. Баллы.

### 3 балла

Выполнены все четыре необходимых действия, и ни одна верная строка не указана в качестве ошибочной.

### 2 балла

Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций:

- а) выполнены три из четырёх необходимых действий. Ни одна верная строка не указана в качестве ошибочной;
- б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки

### 1 балл

Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два необходимых действия из четырёх

### 0 баллов

Не выполнены условия, позволяющие поставить 1, 2 или 3 балла 0

**Спасибо за внимание!**