

Задачи с использованием одномерных массивов

1. Опишите алгоритм подсчёта среднего значения положительных элементов в целочисленном массиве из 30 элементов. Если в массиве нет положительных элементов, сообщите об этом. Постарайтесь использовать минимальное количество операций.

```
1. Const N=30;  
   var a:array[1..N] of integer;  
   S,k,i:integer;
```

BEGIN

```
   S:=0; k:=0;
```

```
   For i:=1 to N do
```

```
       if a[i]>0 then begin S:=S + a[i]; k:=k + 1; end;
```

```
   If k>0 then writeln('Среднее значение положительных  
       элементов',S/k) else writeln('Положительных элементов  
       нет');
```

END.

2. Опишите алгоритм подсчёта суммы отрицательных элементов в целочисленном массиве из 30 элементов. Если в массиве нет отрицательных элементов, алгоритм должен сообщить об этом. Постарайтесь использовать минимальное количество операций.

2. Const N=30;

var a:array[1..N] of integer;

S,k,i:integer;

BEGIN

S:=0; k:=0;

For i:=1 **to** N **do**

if a[i]<0 **then**

begin

S:=S + a[i]; k:=k + 1;

end;

If k>0 **then** writeln('Сумма отрицательных элементов',S)

else writeln('Отрицательных элементов нет');

END.

3. Опишите алгоритм поиска двух элементов, сумма которых максимальна, в числовом массиве из 30 элементов. В качестве результата напечатайте сумму этих элементов. Постарайтесь использовать минимальное количество операций.

3. Const N=30;

var a:array[1..N] of real;

Max1,Max2,i:integer;

BEGIN

Max1:=a[1]; Max2:=a[1]; if a[2]>Max1 then Max1:=a[2] else
Max2:=a[2];

For i:=3 to N **do begin**

if a[i]>Max1 **then begin** Max2:=Max1; Max1:=a[i]; **end**

else if a[i]>Max2 **then** Max2:=a[i];

end ;

writeln('Сумма максимальных элементов',Max1+Max2);

END.

4. Опишите алгоритм вычисления 20-го числа в числовой последовательности, первые два элемента которой равны 3 и 2, а каждый последующий равен абсолютной величине разности двух предыдущих. Постарайтесь не заводить массив для хранения значений последовательности.

4. Const N=20;

```
var    First,Second,Last,i:integer;
```

```
BEGIN
```

```
    First:=3; Second:=2;
```

```
    For i:=3 to N do begin
```

```
        Last:=abs(First – Second);
```

```
        First:=Second;
```

```
        Second:=Last;    end;
```

```
    writeln('20-ое число числовой последовательности:',Last);
```

```
END.
```

ЗАДАЧА. Дан массив целых чисел. Определить, образуют ли элементы массива симметричную последовательность чисел (палиндром). Примечание: предполагается, что последовательность содержит больше одного элемента.

Метод: исследуем равенство элементов массива, которые расположены симметрично – на одинаковом расстоянии от центра последовательности.

Для этого сравниваем между собой значения элементов $a[i]$ и $a[n+1-i]$, где номер элемента i начинается с единицы и заканчивается в середине массива – на значении $n \text{ div } 2$.

•••

$i:=1;$

WHILE ($i \leq n \text{ div } 2$) and ($a[i] = a[n+1-i]$) do INC(i);

IF $i > n \text{ div } 2$ THEN WriteLn ('Последовательность симметрична')

ELSE WriteLn ('Последовательность не симметрична'); •••

ЗАДАЧА. Дан массив целых чисел. Определить, образуют ли элементы массива неубывающую последовательность чисел.
Примечание: предполагается, что последовательность содержит больше одного элемента.

...

i:=2;

WHILE (**i** <= **n**) and (a[**i**] >= a[**i** - 1]) do INC(i);

IF **i** > **n** THEN WriteLn ('Последовательность упорядочена')

ELSE WriteLn ('Последовательность не упорядочена');...

Задача. Сдвинуть одномерный массив на один элемент влево.

Например, исходный массив

1	2	3	4	5	6	7	8
5	7	8	2	0	1	6	3

Обработанный
массив:

1	2	3	4	5	6	7	8
7	8	2	0	1	6	3	5

Фрагмент программы:

...

t:=a[1];

For i:=1 To n-1 Do

a[i]:=a[i+1];

a[n]:=t;

...

Задача. Сдвинуть одномерный массив на k элементов влево

(k вводится с клавиатуры).

Например, исходный массив

1	2	3	4	5	6	7	8
5	7	8	2	0	1	6	3

Обработанный массив
(при k=3):

1	2	3	4	5	6	7	8
2	0	1	6	3	5	7	8

Фрагмент программы:

...

For j:=1 To k Do Begin

t:=a[1];

For i:=1 To n-1 Do

a[i]:=a[i+1];

a[n]:=t; End;

...

ЗАДАЧА. Дан массив целых чисел. Удалить из массива все нулевые элементы.

Метод 1: наиболее простая модель удаления элементов из последовательности – очередь: на место удалённого элемента надо записать то значение, которое находится справа. На место элемента справа передвигается следующий и т.д.

Если элемент равен нулю, то он удаляется, а если не равен – то переходим к следующему элементу ($i+1$)

```
...
i:=1;
WHILE (i < n) do
IF a[i] <> 0 THEN INC(i)
ELSE { Удаляем элемент a[i] }
begin
n:=n-1;
For j:= i To n Do a[j]:=a[j+1]
end; ...
```

ЗАДАЧА. Дан массив целых чисел. Вставить элемент **k** после первого максимального элемента.

Метод : пусть q – номер максимального элемента.

Сдвиг на одну позицию вправо и вставка на $q+1$ (после заданного элемента) элемента, равного **k** .

...

FOR **i**:= **n** DownTo **q**+1 Do

A[**i**+1]:=a[**i**];

A[**q**+1]:=k; ...