

Глава 4.10. Последовательный поиск в массиве

В программировании **поиск** - одна из наиболее часто встречающихся задач не вычислительного характера. Можно выделить следующие типовые задачи поиска:

- 1) найти наибольший (наименьший) элемент массива;
- 2) найти элемент массива, значение которого равно заданному значению.

```
•Program3.pas*
program n_5;
  var n, i: integer;
      a:array[1..10] of integer;
begin
  for i:=1 to 10 do a[i]:=random(60);
  for i:=1 to 10 do write (a[i], ' ');
  i:=0;
  repeat
    i:=i+1;
  until (a[i]=50) or (i=10);
  if a[i]=50 then write(['[',i,']']) else write('Нет')
end.
```

Окно вывода

42 27 50 52 29 24 7 57 52 30 [3]

Во многих случаях требуется найти первый из элементов, имеющий соответствующее значение, и дальнейший просмотр массива прекратить. Для этой цели можно использовать следующую программу:

```
•Program3.pas*
program n_3;
  var s, i, imax: integer;
      a:array[1..10] of integer;
begin
  s:=0;
  for i:=1 to 10 do a[i]:=random(50);
  for i:=1 to 10 do write (a[i], ' ');
  imax:=1;
  for i:=2 to 10 do
    if a[i]>a[imax] then imax:=i;
  write ('Наибольший элемент a[',imax,']=', a[imax])
end.
```

Окно вывода

9 1 26 35 37 11 30 11 11 43 Наибольший элемент a[10]=43

Подсчет количества элементов

Определите, количество каких элементов подсчитывается в следующей программе.

```
•Program3.pas*
program n_6;
  var k, i: integer;
      a:array[1..10] of integer;
begin
  for i:=1 to 10 do a[i]:=random(60);
  for i:=1 to 10 do write (a[i], ' ');
  k:=0;
  for i:=1 to 10 do
    if a[i]>50 then k:=k+1;
  write('k=', k)
end.
```

Окно вывода

46 58 46 32 55 58 12 45 37 6 k=3

Сумма значений элементов, удовлетворяющих условию

Определите, какому условию удовлетворяют элементы массива, значения которых суммируются в следующей программе.

```
•Program3.pas*
program n_7;
  var s, i: integer;
      a:array[1..10] of integer;
begin
  for i:=1 to 10 do a[i]:=random(60);
  for i:=1 to 10 do write (a[i], ' ');
  s:=0;
  for i:=1 to 10 do
    if (a[i]>10) and (a[i]<30) then s:=s+a[i];
  write('s=', s)
end.
```

Окно вывода

42 41 36 20 1 22 49 0 12 49 s=54

Глава 4.11. Сортировка массива

Под **сортировкой (упорядочением)** массива понимается перераспределение значений его элементов в некотором определённом порядке. Порядок при котором в массиве первый элемент имеет самое большое значение, а значение каждого следующего не больше значения предыдущего элемента, называют убывающим.

```
•Program3.pas*
program n_8;
  var n, i, j, x, imax: integer;
      a:array[1..10] of integer;
begin
  for i:=1 to 10 do read (a[i]);
  write('Введённый массив: ');
  for i:=1 to 10 do write (a[i], ' ');
  for i:=1 to 9 do
  begin
    imax:=i;
    for j:=i+1 to 10 do if a[j]>a[imax] then imax:=j;
    x:=a[i];
    a[i]:=a[imax];
    a[imax]:=x;
  end;
  writeln;
  write('Отсортированный массив: ');
  for i:=1 to 10 do write (a[i], ' ');
end.
```

Окно вывода

```
54
4
3
0
Введённый массив: 45 3 654 45 5 676 54 4 3 0
Отсортированный массив: 676 654 54 45 45 5 4 3 3 0
```

Глава 4.12. Запись вспомогательных алгоритмов в Pascal

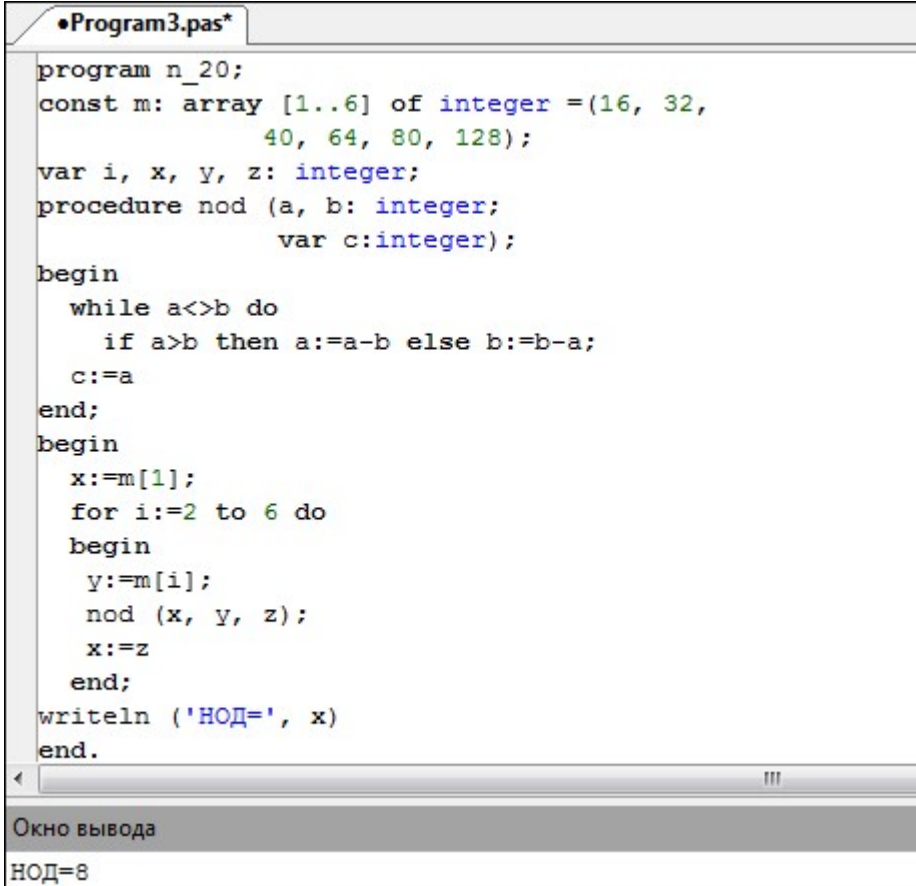
Процедуры

Процедура - подпрограмма, имеющая произвольное количество входных и выходных данных.

Описание процедуры имеет вид:

```
procedure <имя_процедуры> (<описание параметров-значений>);  
  var: <описание параметров-переменных>;  
  begin<операторы>  
  end;
```

Напишем процедуру для нахождения наибольшего общего делителя двух чисел с помощью алгоритма Евклида. Используем её для нахождения наибольшего общего делителя следующих шести чисел: **16, 32, 40, 64, 80 и 128**.



```
•Program3.pas*  
program n_20;  
const m: array [1..6] of integer =(16, 32,  
                                   40, 64, 80, 128);  
var i, x, y, z: integer;  
procedure nod (a, b: integer;  
              var c:integer);  
begin  
  while a<>b do  
    if a>b then a:=a-b else b:=b-a;  
    c:=a  
end;  
begin  
  x:=m[1];  
  for i:=2 to 6 do  
  begin  
    y:=m[i];  
    nod (x, y, z);  
    x:=z  
  end;  
writeln ('НОД=', x)  
end.
```

Окно вывода
НОД=8

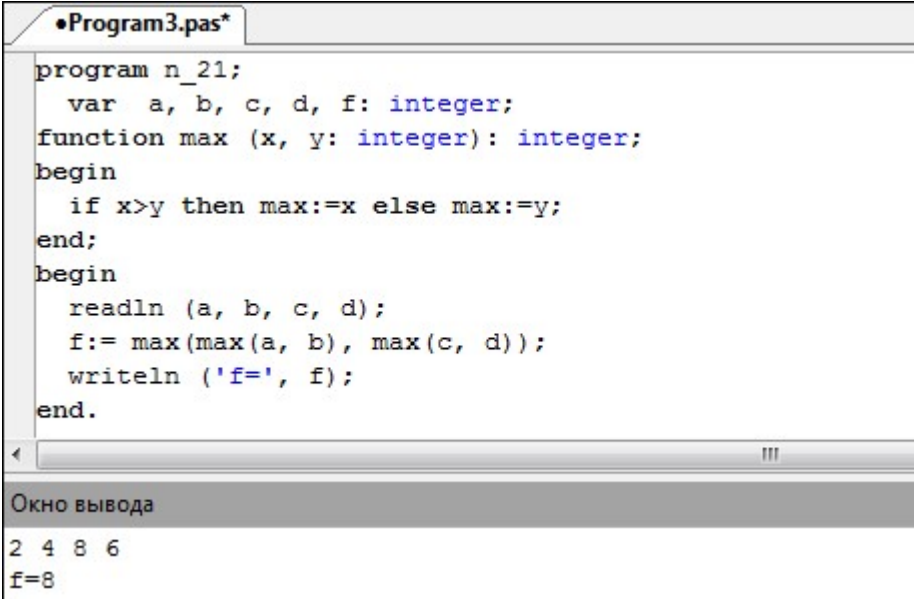
Измените программу так, чтобы с её помощью можно было найти:

- а) наибольшее общий делитель следующих пяти чисел: 12, 24, 30, 48 и 30;
- б) наибольший общий делитель произвольных десяти целых двузначных чисел.

Функции

Функция - подпрограмма, имеющая единственный результат, записываемый в ячейку памяти, имя которой полностью совпадает с именем функции. Поэтому в блоке обязательно должен присутствовать оператор <имя_функции>:=<результат>

function <имя_функции> (<описание входных данных>): <тип_функции>;
begin
<операторы>;
<имя_функции> := <результат>
end;



```
•Program3.pas*
program n_21;
  var a, b, c, d, f: integer;
  function max (x, y: integer): integer;
  begin
    if x>y then max:=x else max:=y;
  end;
begin
  readln (a, b, c, d);
  f:= max(max(a, b), max(c, d));
  writeln ('f=', f);
end.
```

Окно вывода

```
2 4 8 6
f=8
```

Измените программу так, чтобы с её помощью можно было найти:

- а) максимальное из чисел a,b,c;
- б) максимальное из чисел b,c,d;
- в) минимальное из четырёх чисел;
- г) разность максимального и минимального чисел;

Глава 5.1. Относительные, абсолютные и смешанные ссылки

Относительные ссылки. Присутствующая в формуле относительная ссылка фиксирует расположение ячейки, в которой записана формула. При изменении позиции ячейки, содержащей формулу, изменяется и ссылка.

При копировании формулы вдоль столбца и вдоль строки относительная ссылка автоматически корректируется так:

- смещение на один столбец приводит к изменению в ссылке одной буквы в имени столбца;
- смещение на одну строку приводит к изменению в ссылке на единицу номера строки.

Задача 1. Рассмотрим задачу о численности населения некоторого города, ежегодно увеличивающейся на 5%. Проведём в электронных таблицах расчёт предполагаемой численности населения города в ближайшие 5 лет, если в текущем году она составляет 40 000 человек. Внесём в таблицу исходные данные, в ячейку **B3** введём формулу **=B2+0,05*B2** с относительными ссылками; скопируем формулу из ячейки **B3** в диапазон ячеек **B4:B7**.

	A	B
1	Год	Численность населения
2	Текущий	40000
3	1	=B2+0,05*B2
4	2	=B3+0,05*B3
5	3	=B4+0,05*B4
6	4	=B5+0,05*B5
7	5	=B6+0,05*B6
8		

Абсолютные ссылки. Абсолютная ссылка в формуле всегда ссылается на ячейку, расположенную в определённом (фиксированном) месте. В абсолютной ссылке перед каждой буквой и цифрой помещается знак \$, например **\$A\$1**. При изменении позиции ячейки, содержащей формулу, абсолютная ссылка не изменяется. При копировании формулы вдоль строк и вдоль столбцов абсолютная ссылка не корректируется.

Задача 2. Некий гражданин открывает в банке счёт на сумму 10 000 рублей. Ему сообщили, что каждый месяц сумма вклада будет увеличиваться на 1,2%. Для того чтобы узнать возможную сумму и приращение вклада через 1, 2, ..., 6 месяцев, гражданин провёл следующие расчёты.

	A	B	C	D
1	Начальная сумма вклада		10000	
2	Месяц	Сумма	Приращение	
3	1	=C1+C1*0,012	=B3-\$C\$1	
4	2	=B3+B3*0,012	=B4-\$C\$1	
5	3	=B4+B4*0,012	=B5-\$C\$1	
6	4	=B5+B5*0,012	=B6-\$C\$1	
7	5	=B6+B6*0,012	=B7-\$C\$1	
8	6	=B7+B7*0,012	=B8-\$C\$1	

Выполните аналогичные расчёты для начального вклада, равного 15 000 рублям.

Смешанные ссылки. Смешанная ссылка содержит либо абсолютно адресуемый столбец и относительно адресуемую строку (**\$A1**), либо относительно адресуемый столбец и абсолютно адресуемую строку (**A\$1**). При изменении позиции ячейки, содержащей формулу, относительная часть адреса изменяется, а абсолютная не изменяется.

Задача 3. Требуется составить таблицу сложения чисел первого десятка, т.е. заполнить таблицу следующего вида:

	A	B	C	D	E	F	G	H	I	J
1		1	2	3	4	5	6	7	8	9
2	1									
3	2									
4	3									
5	4									
6	5									
7	6									
8	7									
9	8									
10	9									

Внесите в ячейку **B2** формулу **= $\$A2+B\1** и скопируйте её на весь диапазон **B2:J10**. У вас должна получиться таблица сложения, знакомая каждому первокласснику.

Самостоятельная работа:

1. Измените формулу так, чтобы из таблицы сложения сделать таблицу умножения.
2. Модифицируй таблицу, отразив числа второго десятка (20 на 20).