

Основные вопросы:

1. Действия над массивами и над элементами массива.
2. Поиск оптимального элемента.

Первый вопрос: Действия над массивами и над элементами массива.

Для работы с массивом как единым целым используется идентификатор массива без указания индекса в квадратных скобках. Массив может участвовать только в операциях отношения «равно», «не равно», и в операторе присваивания. Массивы, участвующие в этих действиях, должны быть идентичны по структуре, т.е. иметь одинаковые типы индексов и одинаковые типы компонентов. Например, если массивы A и B описаны так:

```
var A,B:array[1..20] of real;
```

то применение к ним допустимых операций даст следующий результат:

Выражение	Результат
A=B	true , если значение каждого элемента массива A равно соответствующему значению элемента массива B
A<>B	true , если хотя бы одно значение элемента массива A не равно значению соответствующего элемента массива B
A:=B	все значения элементов массива B присваиваются соответствующим элементам массива A. Значения элементов массива B остаются неизменными.

Действия над элементами массива.

После объявления массива каждый его элемент можно обработать, указав идентификатор (имя) массива и индекс элемента в квадратных скобках. Например, запись MAS[2], VECTOR[10] позволяет обратиться ко второму элементу массива MAS и десятому элементу массива VECTOR.

Индексированные элементы массива наз. *индексированными переменными* и могут быть использованы так же, как и простые переменные. Элементы массива могут стоять как в левой части оператора присваивания, так и в выражениях. Над элементами массива можно производить те же операции, которые допустимы для данных его базового типа.

```
B[5]:=B[3]+1;
```

```
SUM:=SUM-C[k];
```

```
PI:=A[2*i+1];
```

Для ввода и вывода числовых значений массива используются циклы, т.к. язык TURBO PASCAL 7.0 не имеет средств ввода-вывода элементов массива сразу.

Инициализация (присваивание начальных значений) массива заключается в присваивании каждому элементу массива одного и того значения, соответствующего базовому типу. Наиболее эффективно эта операция выполняется с помощью оператора цикла с параметром, например:

```
for i:=1 to 4 do A[i]:=0;
```

Под вводом понимается ввод значений элементов массива. Под выводом массива понимается вывод на экран значений элементов массива. Если в программе необходимо ввести или вывести значения всех элементов массива, то для этого удобно использовать оператор цикла с параметром, значение параметра которого может быть использовано как индекс элемента массива.

Например:

```
for i:=1 to 30 do readln(A[i]);
```

Копирование массивов наз. присваивание значений всех элементов одного массива всем соответствующим элементам другого массива. Копирование можно выполнить одним оператором присваивания, например $A:=D$; или с помощью оператора цикла с параметром:

```
for i:=1 to 4 do A[i]:=D[i];
```

II. Копирование элементов массива

Копированием массивов называется присваивание всех элементов одного массива всем соответствующим элементам другого массива.

```

program zapoln1;
var n, i: integer;
a, b: array[1..100] of integer;
begin
  randomize;
  writeln;
  write('vvedite kol-vo elementov ->'); readln(n);
  writeln('Massiv A:');
  for i:=1 to n do begin
    a[i]:=random<10>-15;
    write (a[i]:3)
  end;

  writeln;
  writeln('Massiv B:');
  for i:=1 to n do begin
    b[i]:=a[i];
    write (b[i]:3)
  end;
end.

```

массив А копируется
в массив B

15

Сумма выбранных элементов

```

program qq;
const N = 5;
var A: array [1..N] of integer;
    i, S: integer;
begin
  { здесь надо заполнить массив }
  S:= 0;
  for i:=1 to N do
    if A[i] > 0 then S:= S + A[i];
  writeln('Сумма полож. элементов: ', S);
end.

```

перебираем все
элементы массива

Задания для самостоятельной работы

1. Заполнить массив из 5 элементов и вывести на печать сначала все, ниже третий.

```

program alfa;
uses crt;
var i:integer; a:array[1..5] of integer;
begin
  clrscr;
  a[1]:=3;a[2]:=7;a[3]:=6;a[4]:=9;a[5]:=2;
  for i:=1 to 5 do
    write(a[i],' '); writeln;
  write('3 элемент= ',a[3]);
  readkey;
end.
  
```

2. Заполнить массив из 5 элементов с клавиатуры и найти их среднее арифметическое.

```

program alfa;
uses crt;
var i:integer;
a:array[1..5] of integer;
sr:real;
begin
  clrscr;
  sr:=0;
  for i:=1 to 5 do
  begin
    write('Введите ',i,' элемент ');read(a[i]);
    sr:=sr+a[i];
  end;
  sr:=sr/5;
  write('Среднее арифметическое= ',sr:1:2);
  readkey;
end.
  
```

Найти количество положительных элементов массива.

```

const n=4;
var a:array [1..n] of integer;i,s:integer;
begin
  for i:=1 to n do
  begin
    write ('a[' ,i, ']=');
    readln(a[i]);
  end;
  for i:=1 to n do
  begin
    if a[i]>0 then s:=s+1;
  end;
  writeln ('количество=',s);
end.
  
```

Второй вопрос: Поиск оптимального элемента.

Алгоритм поиска минимального (максимального) элемента массива довольно очевиден: делается предположение, что первый элемент массива является минимальным (максимальным), затем остальные элементы массива сравниваются с этим элементом. Если обнаруживается, что проверяемый элемент меньше (больше) принятого за минимальный (максимальный) и продолжается проверка оставшихся элементов. Ниже на рис. представлена блок-схема алгоритма поиска минимального элемента в массиве из 10 целых чисел.

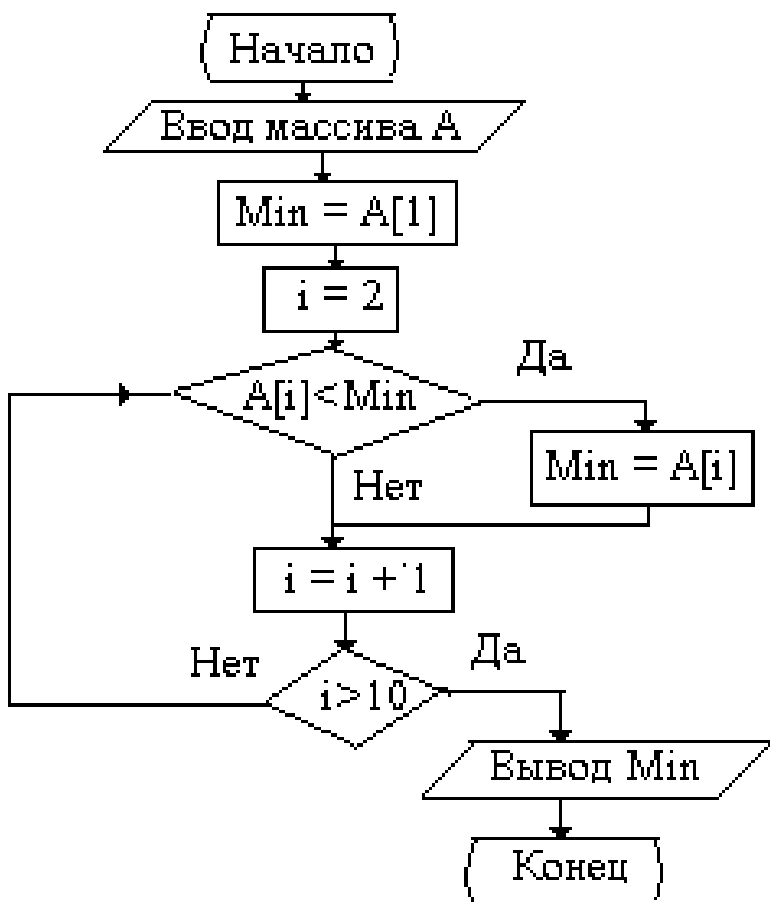


Рис. Блок-схема алгоритма поиска минимального элемента.

Программа:

Program Example ;

Var

A : Array[1..10] of integer ; { массив целых чисел } ;

Min : Integer ; { номер минимального элемента массива }

i : Integer ; { номер элемента сравниваемого с минимальным }

Begin

WriteLn(' Введите 10 целых чисел в одной строке через пробел ') ;

for i := 1 to 10 do ReadLn(A[i]) ; { Ввод массива целых }

Min := A[1] ;

for i := 2 to 10 do

if A[i] < Min then Min := A[i] ;

WriteLn(' Минимальный элемент массива : ' , Min:3) ;

End.