

Вопросы к теме:
« Одномерный массив »

Что такое одномерный массив?

Одномерный массив-это упорядоченная совокупность однотипных переменных, последовательно расположенных в памяти ЭВМ. Весь массив имеет одно общее имя, а каждый элемент массива- свой номер.



В чем заключается главное свойство одномерного массива?

Главное свойство массива- возможность прямого доступа к любому его элементу путем указания имени массива и номера элемента в квадратных скобках.



Какие вы знаете способы заполнения и вывода на экран одномерного массива?

1. Способы заполнения массива:

С клавиатуры:

```
for i:=1 to n do read(a[i]);
```

Случайным образом:

```
randomize;
```

```
for i:=1 to n do
```

```
  a[i]:= -10+random(21);
```

Из файла:

```
assign(f,'input.dat');
```

```
reset(f);
```

```
for i:=1 to n do read(f,a[i]);
```

```
close(f);
```

2. Способы вывода массива на экран:

В строку:

```
for i:=1 to n do write(a[i], ' ');
```

Типы задач на обработку одномерных массивов.

- Поиск суммы и произведения элементов;
- Поиск количества;
- Поиск минимального и максимального элементов и их номеров;
- Исследование массива на некоторое свойство.



Алгоритмы реализации задачи на языке Паскаль.

□ алгоритм поиск суммы элементов:

```
s:=0;  
for i:=1 to n do  
if a[i]>0 then s:=s+a[i];  
writeln(s);
```

□ алгоритм поиска количества элементов:

```
k:=0;  
for i:=1 to n do  
if a[i]>0 then k:=k+1;  
writeln(k);
```

□ алгоритм поиска максимального элемента:

```
m:=-maxint;
```

▶ for i:=1 to n do

на экран номера первого четного элемента
целочисленного массива из 30 элементов или вывести
на экран сообщение о том, что четных элементов в
массиве нет.

```
var a: array [1..30] of integer; f, i, k: integer;
begin f:=0;
for i:=1 to 30 do
begin
read(a[i]);
if (a[i] mod 2=0) and (f=0) then
begin f:=1; k:= i;
end;
end;
writeln;
if f=0 then write('четных нет') else write(k);
```

```
const n=30;
```

```
var a: array [1..n] of integer; i: integer;
```



Методы сортировки одномерных массивов.



Метод прямого выбора.

Алгоритм сортировки массива по возрастанию методом прямого выбора может быть представлен так:

- Просматривая массив с первого элемента, найти минимальный и поменять его местами с первым элементом.*
 - Просматривая массив со второго элемента, найти минимальный и поменять его местами со вторым элементом.*
 - И, так далее, до последнего элемента.*
-



Алгоритм использует вложенные циклы. Внешний цикл (счетчик шагов) последовательно выбирает номер элемента массива, куда следует записывать найденный в неупорядоченной части массива минимальный элемент. Внутренний цикл перебирает номера неупорядоченных элементов при поиске минимального элемента. Для внешнего цикла достаточно шагов на один меньше, чем элементов в массиве.

```
for i:=1 to n-1 do
  for j:=1 to n-i do
    if a[j]>a[j+1] then begin
      m:=a[j]; a[j]:=a[j+1]; a[j+1]:=m;
    end;
```



Метод пузырька.

В основе алгоритма лежит обмен соседних элементов массива. Каждый элемент массива, начиная с первого, сравнивается со следующим и если он больше следующего, то элементы меняются местами. Таким образом, элементы с меньшим значением продвигаются к началу массива (всплывают), а элементы с большим значением – к концу массива (тонут). Этот процесс повторяется на единицу меньше раз, чем элементов в массиве.



```
for i:=1 to n-1 do
  for j:=1 to n-i do
    if a[j]>a[j+1] then begin
      m:=a[j]; a[j]:=a[j+1]; a[j+1]:=m;
    end;
```

