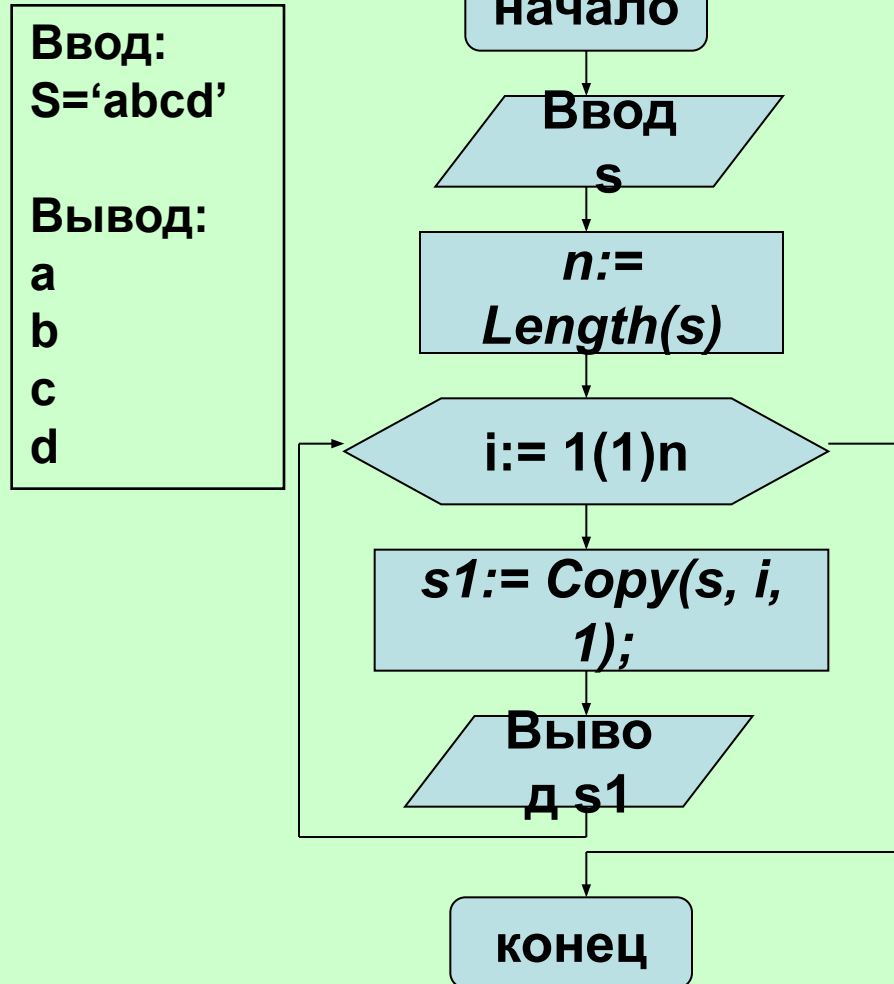


# **Обработка символьной информации (задачи, часть 3)**

**(TurboPascal)**

**Вывести на экран последовательно каждую букву слова. (способ 1)**



```
Program pr1;  
  uses crt;  
var  
  s, s1: string; i, n: integer;  
begin  
  clrscr;  
  readLn(s); n:= length(s);  
  for i := 1 to n do  
    begin  
      s1 := copy(s, i, 1);  
      writeLn(s1)  
    end  
  end.  
end.
```

# Строки как массивы символов

Строка – цепочка символов. Можно провести аналогию между строкой и одномерным массивом символов типа *char* (т.е. *array[1..N] of char*).

К любому символу в строке можно обращаться как к элементу одномерного массива, т.е. переменные типа **string** обрабатываются как массив. Например:

**var**

**s : string;**

**begin**

**s := 'Hello World!';**

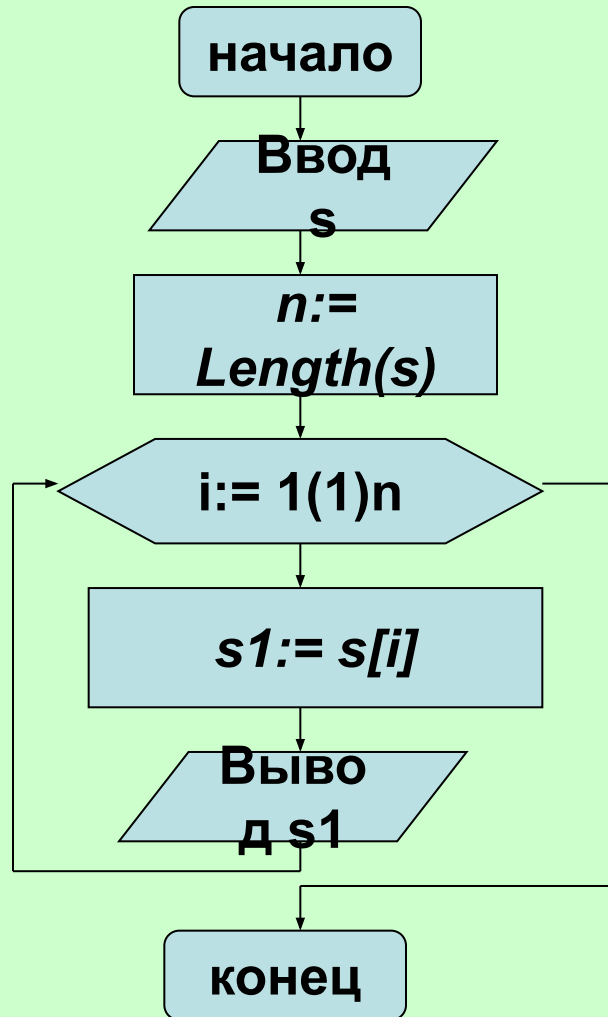
**if s[3]='l' then WriteLn('OK');**

**end.**

# Вывести на экран последовательно каждую букву слова. (способ 2)

**Ввод:**  
S='abcd'

**Вывод:**  
a  
b  
c  
d



```
var
    s,s1:string;
    i,n:integer;
begin
    writeln('vvedite slovo');
    readln(s);
    n:=length(s);
    for i:=1 to n do
        begin
            s1:=s[i];
            writeln(s1)
        end
    end.
end.
```

**Самый первый байт в строке содержит символ ,  
код которого - текущая длина строки.**

**`ord(s[0])` – длина строки.**

**Первый значащий символ строки занимает  
второй байт и имеет индекс 1 и т.д.**

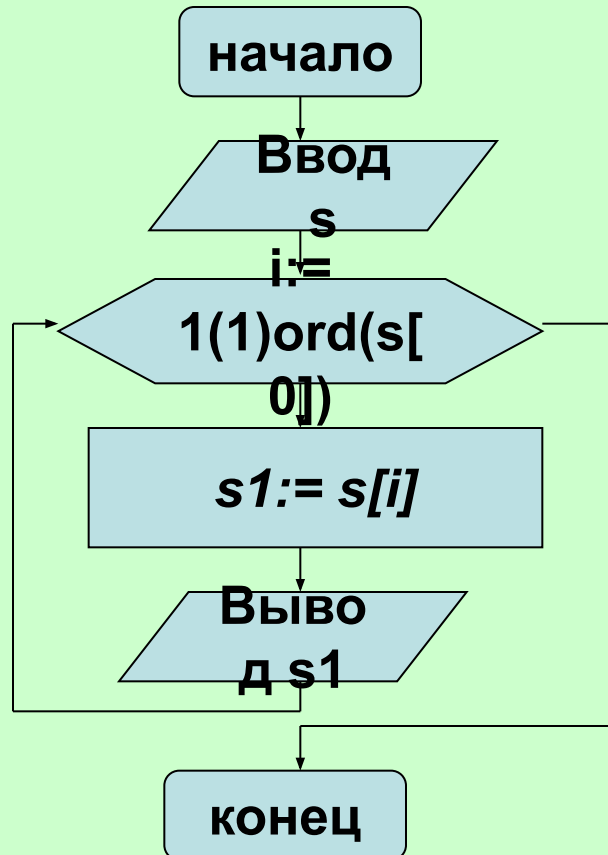
**n a m e**

**s[0] s[1] s[2] s[3] s[4]**

# Вывести на экран последовательно каждую букву слова. (способ 2а)

**Ввод:**  
S='abcd'

**Вывод:**  
a  
b  
c  
d



```
var
  s,s1: string;
  i: integer;
begin
  writeln ('vvedite slovo');
  readln(s);
  for i:=1 to ord(s[0]) do
  begin
    s1:=s[i];
    writeln(s1)
  end
end.
```

1. Найти количество искомой буквы в заданном слове.
2. Дано предложение. Заменить все буквы «е» на «и».

# Определить истинность утверждений:

Строка – цепочка символов.

**true**



Первый значащий символ строки  
занимает второй байт и имеет  
индекс 1.

**true**

Можно провести аналогию между  
строкой и двумерным массивом  
СИМВОЛОВ

**false**

Самый первый байт в строке  
содержит символ, код которого -  
текущая длина строки.

**true**

Переменные типа **string**  
обрабатываются как  
одномерный массив.

**true**

К любому символу в строке можно обращаться как к элементу одномерного массива.

**true**

**4. Найти длину слова «умозаключение» и получить из него слова «ум», «ключ», «замок», «зал», «люк».**

***var***

*s1, s2, s3, s4, s5, s6 : string; n: integer;*

***Begin***

*s1 := 'умозаключение';*

*n := Length(s1);*

*WriteLn ('Длина слова', n)*

*s2 := copy(s1,1,2);*

*s3 := copy(s1,6,4);*

*s4 := copy(s1,4,2)+ copy(s1,2,2)+ copy(s1,6,1);*

*s5 := copy(s1,4,2)+ copy(s1,7,1);*

*s6 := copy(s1,7,2)+ copy(s1,6,1);*

*WriteLn(s2, s3, s4, s5, s6)*

***end.***