

Строки символов

Программирование обработки
информации



Символьный тип данных

Величина символьного типа (char)

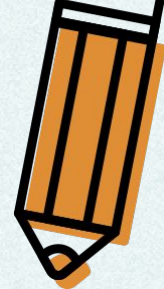
может хранить любой символ
компьютерного алфавита.

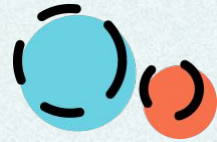
Кодовая таблица:

unicode-16.

Размерность переменной:

2 байта.





Строки символов

1

Строки и их
описание в
программе.

2

Хранение строк
в оперативной
памяти.

3

Операции, функции
и процедуры
обработки строк.

Строки

Строка —

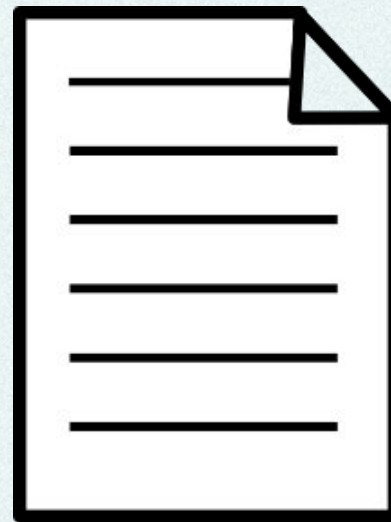
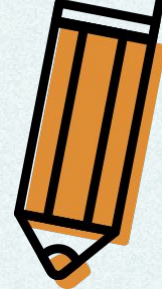
это последовательность символов.

Длина строки —

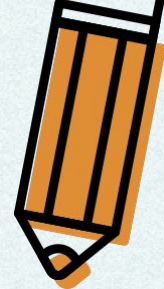
это количество символов, из которых она состоит.

Запись символьных величин:

'яблоко', 'машина', '578'.



Описание строковой переменной

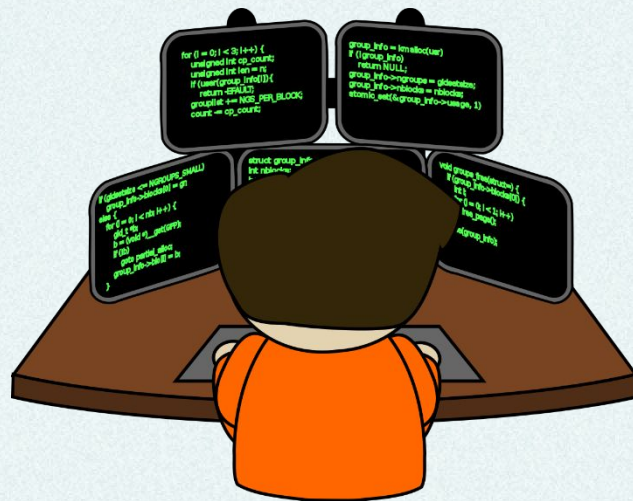


var

≤ 255

<имя>: **string** [<максимальная длина>];

s: **string**; – длина 255 символов.



Хранение строк



Размерность строк

t: `string` [6];

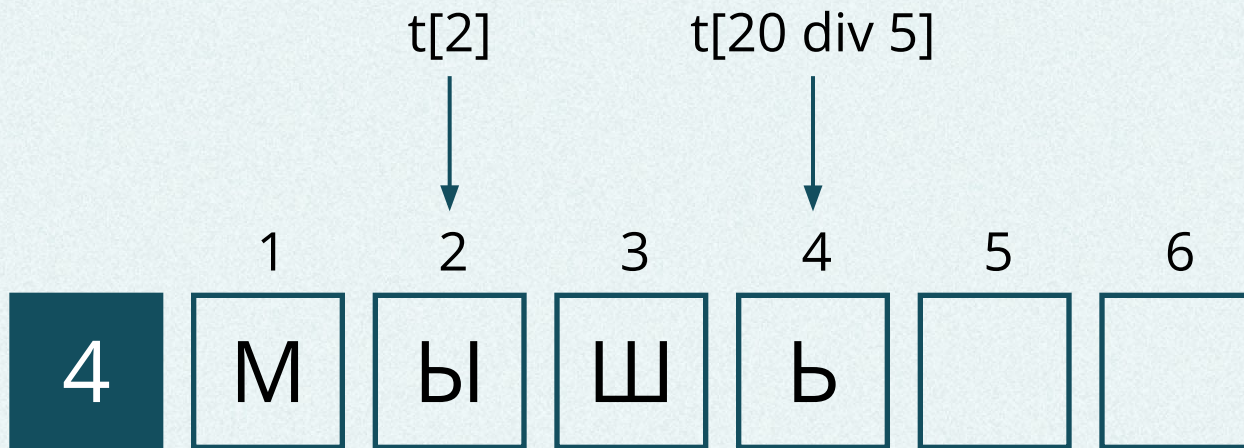
$6 \times 2 \text{ байта} + 1 \text{ байт} = 13 \text{ байт}$



s: `string`;

$255 \times 2 \text{ байта} + 1 \text{ байт} = 511 \text{ байт}$

	1	2	3	4	5	6
4	М	Ы	Ш	Ь		

Обращение к символам





Операции обработки строк

Сцепление (конкатенация)

$s1 + s2$

Тип результата: **string**

'стекловата'

$s := \text{'стекло'} + \text{'вата'}$

Операции отношения

$<, >, =, <>, >=, <=$

Тип результата: **boolean**

1072

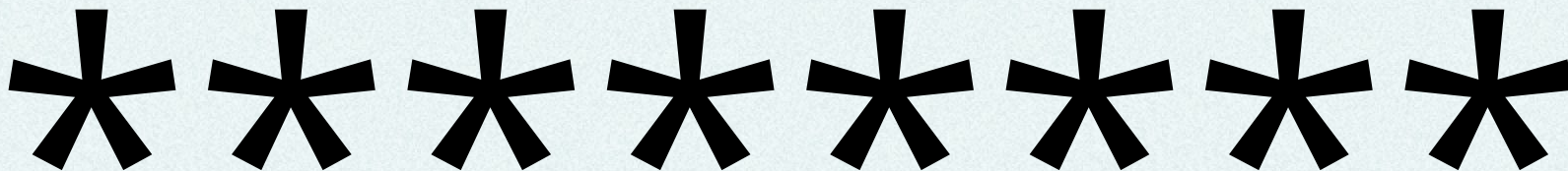
'Корабл' < 'Корыто'

1099

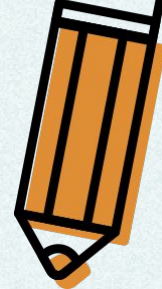
Задача

Написать программу, которая формирует строку, состоящую из **n** звёздочек.



$$1 \leq n \leq 255$$



Написание и тестирование программы



```
program zvezdochki;  
var  
  s: string;  
  n, i: byte;  
begin  
  writeln ('Программа, формирующая строку из n  
звёздочек. ');  
  write ('n=');  
  readln (n);  
  s:='';  
  for i:=1 to n do  
    s:=s+'*';  
  write ('Получившаяся строка: ', s);  
end.
```

Функции обработки строк

```
graph TD; A[Функции обработки строк] --> B[Выделение подстроки]; A --> C[Определение длины]; A --> D[Поиск подстроки];
```

Выделение подстроки

copy (s, poz, kol)

Тип результата:
string

Определение длины


length (s)

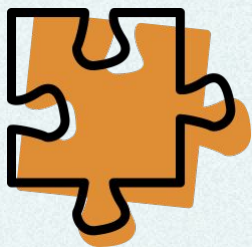
Тип результата:
byte

Поиск подстроки

pos (ps, s)

Тип результата:
byte





Процедуры обработки строк

Удаление
подстроки

`delete` (s, poz, kol)

Преобразование
числа в строку

`str` (x, s)

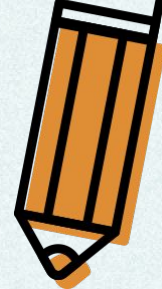
Вставка
подстроки

`insert` (ps, s, poz)

Преобразование
строки в число

`val` (s, x, e)





Задача

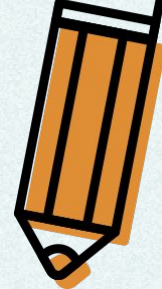
Заданная строка состоит из слов и целых чисел, разделённых одиночными пробелами. Сохранить отдельно слова и числа, которые встречаются в строке и вывести их на экран. Количество слов и чисел не превышает 50, длина каждого слова – не более 20 символов, а каждое число – не больше 2 000 000 000.

--	--	--	--	--	--	--	--

Слова: У, собак, лап, и, ушей.

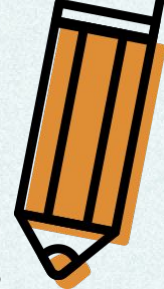
Числа: 10, 40, 20.

Написание программы



```
program slova_i_chisla;
var
  stroka: string;
  s: string [20];
  slova: array [1..50] of string [20];
  chisla: array [1..50] of integer;
  kol_slov, kol_chisel, space, i: byte;
  chislo, code: integer;
begin
  writeln ('Программа, выделяющая слова и
числа из строки. Введите строку.' );
  readln (stroka);
  kol_slov:=0;
  kol_chisel:=0;
```

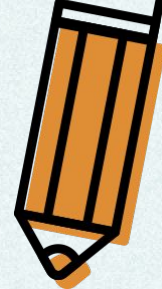
Написание программы



```
while stroka <> '' do  
begin  
  space:=pos(' ', stroka);  
  if space=0  
  then space:=length(stroka)+1;  
  s:=copy(stroka, 1, space-1);  
  val(s, chislo, code);  
  if code=0  
  then begin  
    kol_chisel:=kol_chisel+1;  
    chisla[kol_chisel]:=chislo;  
  end  
  else begin  
    kol_slov:=kol_slov+1;  
    slova[kol_slov]:=s;  
  end;  
  delete(stroka, 1, space);  
end;
```

```
write('Слова:');  
  for i:=1 to kol_slov do  
    write(' ', slova[i]);  
  writeln();  
  write('Числа:');  
  for i:=1 to kol_chisel do  
    write(' ', chisla[i]);  
end.
```

Строки символов



Строка —

это последовательность символов.

Описание строки:

<имя>: **string** [<максимальная длина>];

Операции над строками:

- ✓ сцепление (конкатенация): +;
- ✓ операции отношения: >, <, =, <>, >=, <=.

Функции обработки строк:

- ✓ выделение подстроки: **copy** (s, poz, kol);
- ✓ определение длины строки: **length** (s);
- ✓ поиск подстроки: **pos** (ps, s).

Процедуры обработки строк:

- ✓ удаление подстроки: **delete** (s, poz, kol);
- ✓ вставка подстроки: **insert** (ps, s, poz);
- ✓ преобразование числа в строку: **str** (x, s);
- ✓ преобразование строки в число: **val** (s, x, e).

