Работа со строками

Символьный тип данных — **Char**

Пример описания символьных переменных:

```
Var c1, c2: char;
```

Символьный тип относится к порядковым типам данных.

Из этого следует:

- Символы — упорядоченное множество
- У каждого символа в этом множестве есть свой порядковый номер
- Между символами работает соотношение «следующий — предыдущий»

Символьная величина занимает 1 или 2 байта памяти, в которой хранится код этого символа, соответствующий используемой кодовой таблице.

Функция `Ord(x)` — функция от аргумента порядкового типа, которая возвращает порядковый номер значения x в этом типе данных. Если x — символьная величина, то результатом функции будет десятичный код x в кодовой таблице.
Например, `Ord('A')=65, Ord ('5')=53`

Фукнция `Chr(x)` — функция от целочисленного аргумента, результатом которой является символ с кодом, равным x.
Например, `chr(65)='A', chr(53)='5'`

Пример 1. Составить программу на Паскале, по которой на экран будет выводиться таблица кодировки в диапазоне кодов от 32 до 255. (Символы с кодами, меньшими 32, являются управляющими). Значения выводятся парами: символ — код. В одной строке располагается 10 таких пар.

```
program tabl_code;
var kod: byte;
begin
  for kod:=32 to 255 do
  begin
   if (kod mod 10 = 0) then
writeln;
  write(chr(kod):3,  kod:4);
  end;
end.
```

В любой кодовой таблице выполняется принцип последовательного кодирования латинского (английского) алфавита и алфавита десятичной системы счисления.

Например, значение переменной C является прописной (заглавной) латинской буквой, если истинно логическое выражение `(C>='A') and (c<='Z')` и является цифрой, если истинно логическое выражение `(C>='0') and (c<='9')`.
В латинском алфавите 26 букв. Поэтому разница между кодами букв `'Z'` и `'A'`, а также букв `'z'` и `'a'` равна 25.

Пример 2. С помощью генератора случайных чисел заполнить массив `sim[0..10]` строчными английскими буквами. Затем массив отсортировать в алфавитном порядке.

```
program sort_mas;
uses crt;
var sim: array[0..10] of char;
    C: char; i, j: integer;
begin
  ClrScr;
  Randomize;
  writeln('исходный массив:');
  for i:=0 to 10 do
   begin
   sim[i]:=chr(random(26)+ord('a'));
   write(sim[i]);
   end;
writeln;
for i:=0 to 9 do
   for j:=0 to 9-i do
     if sim[j]>sim[j+1] then
      begin
      c:=sim[j];
      sim[j]:=sim[j+1];
      sim[j+1]:=c;
      end;
writeln('Отсортированный
массив:');
for i:=0 to 10 do
   write(sim[i]);
readln;
end.
```

Строковый тип данных — **`string`**
Строка — это последовательность символов.
Строковые величины могут быть переменными и константами.
Строковая константа записывается как последовательность символов, заключённых в апострофы.
Строковая переменная описывается в разделе описания переменных:

```
var <идентификатор>: string;
```

Может быть (но сейчас практически не используется) описана следующим образом:

```
var <идентификатор>: string[<максимальная длина строки>];
```

Чтобы записать в переменную строкового типа значение, используют оператор присваивания (например, `s:='Сегодня прекрасный день!'`) или оператор ввода с клавиатуры (`readln(s);`).
Важно! При вводе строк нужно использовать оператор `readln` вместо `read`.
Символы внутри строки нумеруются, начиная с единицы.
Каждый отдельный символ идентифицируется именем строки с индексом, заключённым в квадратные скобки.
Например, `name[5], name[i], slovo[j+1]`
Значение индекса может быть задано положительной константой, переменной, выражением целочисленного типа.
Тип `string` и тип `char` совместимы: строки и символы могут употребляться в одних и тех же выражениях.

Операции над строками:
Операция объединения (сцепления, конкатенации): +
Операции отношения: =, <, >, <=, >=, <>

Стандартные функции:
`copy(s, poz, n)` — выделение подстроки
`pos(s1, s2)` — определение первого вхождения в строку
`length (s)` — определение текущей длины строки
`concat (s1, s2, .., sn)` — сцепление (конкатенация) строк

Стандартные процедуры:
`delete(s, poz, n)` – удаление подстроки
`insert(s1, s2, poz)` — вставка подстроки

Функция `length (s)` определяет текущую длину строки `s`. Результат — значение целочисленного типа.
Например,

Значение S:	Выражение:	Результат:
`'Test-5'`	`length(s)`	`6`
`'(A+B)*C'`	`length(s)`	`7`

Функция `copy(s, poz, n)` выделяет из строки `s` подстроку длиной `n` символов, начиная с позиции `poz`. `N` и `poz` — целочисленные переменные.
Например,

Значение S:	Выражение:	Результат:
`'ABCDEFG'`	`copy (s, 2, 3)`	`BCD`
`'ABCDEFG '`	`copy (s, 4, 4)`	`DEFG`

Функция `pos(s1, s2)` обнаруживает первое появление в строке `s2` подстроки `s1`. Результат — целое число, равное номеру позиции, где находится первый символ подстроки `s1`. Если в `s2` не обнаружена постpока `s1`, то результат равен 0.

Например,

Значение S2 :	Выражение:	Результат:
`'abcdef'`	`pos('cd', s2)`	3
`'abcdcdef'`	`pos('cd', s2)`	3
`'abcdef'`	`pos('k', s2)`	0

Процедура `delete(s, poz, n)` удаляет n символов из строки `s`, начиная с позиции `poz`.

Например,

Исходное значение:	Оператор:	Конечное значение:
`'abcdefg'`	`delete(s, 3, 2)`	`'abefg'`
`'abcdefg'`	`delete(s, 2, 6)`	`'a'`

Процедура `insert(s1, s2, poz)` выполняет вставку строки `s1` в строку `s2`, начиная с позиции `poz`.

Например,

Исходное значение:	Оператор:	Конечное значение:
`'ЭВМ PC'`	`insert('IBM-',s2,5)`	`'ЭВМ IBM-PC'`
`'Рис. 2'`	`insert('N',s2,6)`	`'Рис. N2'`

Пример 3. Составить программу, формирующую символьную строку, состоящую из N звёздочек.

```
program stars;
var a: string; N, i: byte;
begin
  writeln('Введите число звёздочек');
  readln(N);
  a:='';
  for i:=1 to N do a:=a+'*';
  writeln (a);
readln;
end.
```

Пример 4. Ввести строку с клавиатуры, заменить в ней все буквы 'a' на буквы 'b' и вывести полученную строку на экран.

```
program replaceAB;
var s: string; i: integer;
begin
  writeln('Введите строку');
  readln(s);
  for i:=1 to length(s) do
     if s[i]='a' then s[i]:='b';
  writeln (s);
readln;
end.
```

Пример 5. В символьной строке, заданной с клавиатуры подсчитать количество цифр, предшествующих первому символу '!'.

```
program countdigit;
var s: string; k, i: integer;
begin
  writeln('Введите строку');
  readln(s);
  K:=0; i:=1;
  while (i<=length(s)) and (s[i]<>'!') do
  begin
     if (s[i]>='0') and (s[i]<='9') then k:=k+1;
     i:=i+1;
  end;
readln;
end.
```

Пример 6. С клавиатуры вводится строка, содержащая имя, отчество и фамилия человека. Каждые два слова разделены одним пробелом, в начале строки пробелов нет. В результате обработки должна получиться новая строка, содержащая фамилию и инициалы.

Например,
На вход: Иван Васильевич Потапенко
На выходе: Потапенко И.В.

```
program FIO;
var s, name, name2: string;
    n: integer;
begin
  write('Введите имя, отчество и фамилию:');
  readln(s);
  n:=pos(' ', s);
  name:=copy(s, 1, n-1); // вырезать имя
  delete(s, 1, n);
  n:=pos(' ', s);
  name2:=copy(s, 1, n-1); // вырезать отчество
  delete(s, 1, n);
  s:=s + ' ' + name[1] + '.' + name2[1] + '.';
  writeln(s);
  readln;
end.
```