Текстовый и символьный типы данных

Паскаль

Вычислительные машины имеют дело не только с числами. Едва ли не больше времени они бывают заняты обработкой текста. В Паскале для этого есть специальный тип данных, который называется CHAR (от слова character – символ).

Тип CHAR (символьный или строковый или литерный).

Его значениями являются отдельные символы: буквы, цифры, знаки. Символьные константы заключаются в кавычки, например, ‘A’, ’B’, ’C’, ‘4’,‘7’, ‘ ‘(пробел).

Символьные переменные описываются предложением
Var имя переменной: char;

Символьные значения можно вводить и выводить, присваивать, сравнивать. Ниже приведен пример, где выполняются все эти действия.

```
Var     x,y:char;
Begin
Write(‘Введите символ‘);
Readln(x);
Y:=’A’;
If x<y then write (‘X’) else write (‘y’); {на экран буде выдан символ хранящийся в переменной X или Y в зависимости от проверки условия}
Readln;
     End.
```

Сравнивать символы можно благодаря тому, что в машинной памяти они хранятся в виде целых чисел (кодов символов). Из двух символов большим считается тот, код которого больше. Символы упорядочены следующим образом:

‘A’<’B’<…<’Z’

‘a’<’b’<…<’z’

‘0’<’1’<…<’9’

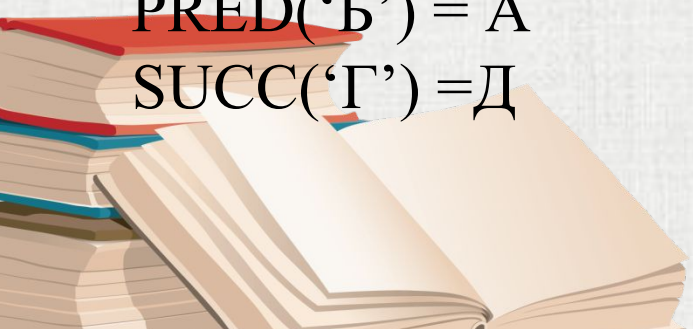
‘а’<’б’<…<’я’

‘А’<’Б’<…<’Я’

Для символов допустимы все шесть операций сравнения: =, <=,>=,<,>,<>.

Стандартные символьные функции.

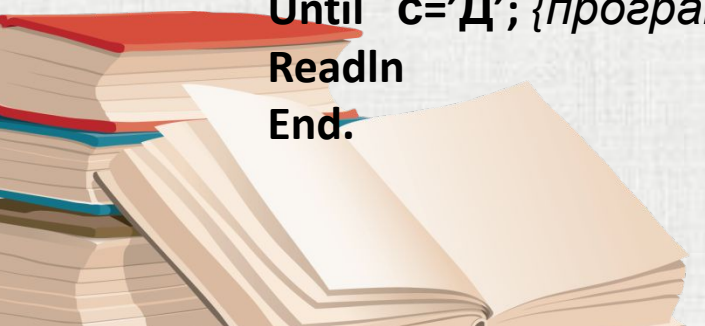
В Паскале имеются стандартные символьные функции:
CHR(N) – возвращает в программу символ с кодом N,
ORD(S) – возвращает код символа S,
PRED(S) –возвращает предыдущий символ
SUCC(S) – возвращает следующий символ
ПРИМЕРЫ:
CHR(128) = Б
ORD(‘:’) = 58
PRED(‘Б’) = А
SUCC(‘Г’) =Д



Каждый символ имеет свой уникальный двоичный код. Коды всех символов сведены в таблицу. Первая половина таблицы стала международным стандартом, который называется ASCII – American Standard Code Information Interchange (читается «аски код») в ней кроме прочего содержится латинский алфавит, вторая имеет разные варианты для разных языков. Кириллица (русский алфавит) имеет несколько стандартов. В Паскале используется стандарт КОИ-8.

ПРИМЕР использования переменной символьного типа.
Составить программу, по которой компьютер многократно вычисляет сумму A+B при различных значениях A и B. в конце каждого этапа появляется запрос о продолжении или прекращении вычислений: «Завершить программу?(Д/Н)».

```
Var   A,B:real; C : char;
Begin
repeat
Write('Введите два числа'); Readln(a,b);
Writeln(a+b:0:2);
Writeln('Завершить программу?(Д/Н)');
Readln(c);
Until   c='Д'; {программа завершит работу если будет введено Д}
Readln
End.
```



Тренировочные задания.

1. Что вернет функция CHR(ORD(X))?
2. Определить значения следующих функций:

CHR(68)
ORD(‘d’)
PRED(1)
SUCC(‘Я’)

3. Составить программу, по которой компьютер находит произведение нечетных чисел, начиная с единицы, и до тех пор, пока на вопрос, задаваемый после каждого шага вычислений: «Продолжить вычисления? (Д/Н)», отвечают ‘Д’.

Для обработки более крупных текстовых единиц - строк введен тип данных, который называется STRING (строка).

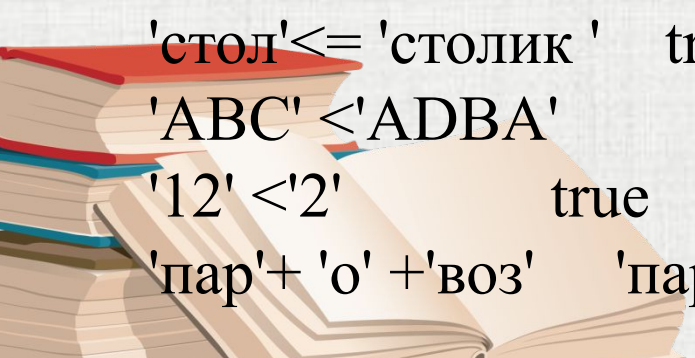
Значениями этого типа являются строки любых символов длиной до 255.

Переменные строки должны быть описаны предложением:

VAR имя: STRING

Строки можно присваивать, сравнивать, вводить, выводить и соединять. Соединение обозначается знаком "+". Вот примеры некоторых операций сравнения над строками:

'стол'<= 'столик ' true
'ABC' <'ADBA' true
'12' <'2' true
'пар'+ 'о' +'воз' 'паровоз'



Среди всевозможных значений строк есть пустая строка. Она изображается двумя апострофами (одинарными кавычками), между которыми ничего нет. Чтобы ввести этот символ в состав строки, надо повторить его дважды. Например, оператор

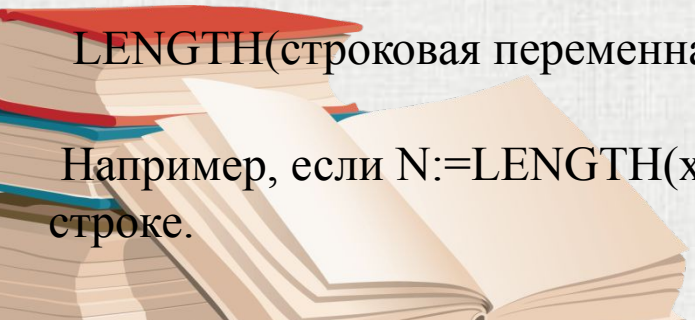
write ('об''явление')
выведет на экран: об'явление.

Программисту доступны отдельные символы строковой переменной, для этого кроме имени переменной надо указать порядковый номер символа в строке. Например, если описана переменная X:STRING, то X[1] - это первый символ строки, X[2] - второй и т.д.

У X[0] особая роль - хранить длину строки. Значением X[0] является символ, код которого равен количеству символов в строке. Но для определения длины строковой переменной обычно используется функция

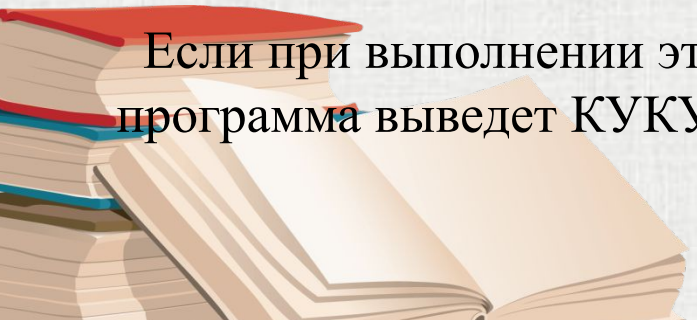
LENGTH(строковая переменная).

Например, если N:=LENGTH(x); - N присвоится значение равное числу символов в строке.



При описании строковой переменной мы можем ограничить длину строки, указав ее максимально возможный размер, тогда в строке будет храниться только указанное число символов.

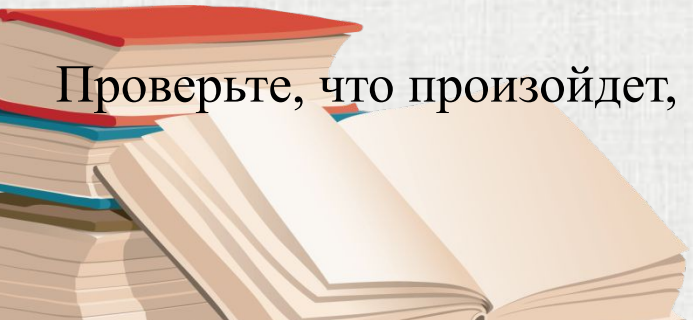
```
Var
  a,b:string[4];
begin
  write('введите слово');
  readln(a);
  write(a);
  readln
end.
```



Если при выполнении этой программы ввести слово КУКУРУЗА, то программа выведет КУКУ.

ЗАПОМНИТЕ. Если при выполнении программы необходимо ввести значение для нескольких строковых переменных, для каждой из них должен быть указан свой оператор ввода READLN. Например,

```
Var
  a,b,c:string;
begin
  readln(a);
  readln(b);
  readln(c);
  write(a+b+c);
  readln
end.
```



Проверьте, что произойдет, если записать READLN(a,b,c); или READ(a,b,c).

Пример 1.

Составить программу определяющую, какая из двух фамилий длиннее. Фамилии имеют разную длину.

```
Var
  a,b:string;
begin
  readln(a);
  readln(b);
  if length(a)>length(b) then write(a) else write(b);
  readln
end.
```

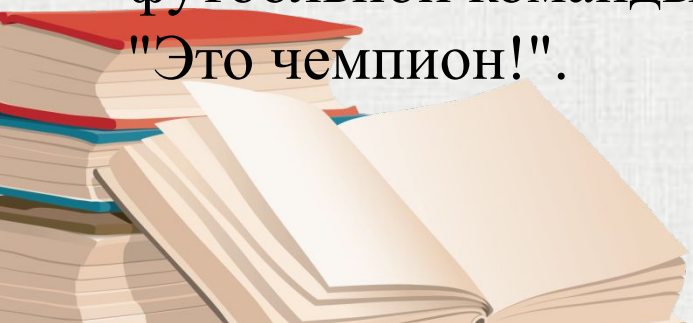
Пример 2.

Даны два слова. Составить программу определяющую верно ли, что первое слово начинается на ту же букву, которой оканчивается второе слово.

```
Var  x,y:byte;
     a,b:string;
begin
  readln(a);
  readln(b);
  x:=length(b); {определяем длину слова b, чтобы узнать номер последнего символа}
  if a[1]=b[x] then write('верно') else write('неверно');
  readln
end.
```

Тренировочные задания.

1. Дано название города. Определить, четно или не четно количество символов в нем.
2. Дано слово. Вывести на экран его третий символ и дважды его последний символ.
3. Дано слово. Верно ли, что оно начинается и оканчивается на одну и ту же букву?
4. Дано слово. Получить и вывести на экран буквосочетание, состоящее из его третьего и последнего символа.
5. Составить программу, которая запрашивает название футбольной команды и повторяет его на экране со словами: "Это чемпион!".



Для работы со строковыми переменными в Паскале существует набор стандартных процедур и функций. Их применение упрощает решение задач. Хочу напомнить что результат выполнения функции должен быть запомнен в переменной соответствующего типа, если конечно она, функция, не является элементом выражения.

Функция копирования строки или ее части.

S:=COPY(*строка*, *позиция*, N);

Функция копирования называется также "вырезкой". Результатом выполнения функции будет часть строки начиная с указанной позиции длиной N.

Пример: Дано предложение. Определить порядковый номер первой встреченной буквы 'к'. Если такой буквы нет, сообщить об этом.

```
Var  x: integer;
     a: string;
begin
  write('Введите предложение');
  readln(a);
  x:=pos('к',a);
  if x=0 then writeln(' Такой буквы нет') else writeln(x);
  readln
end.
```

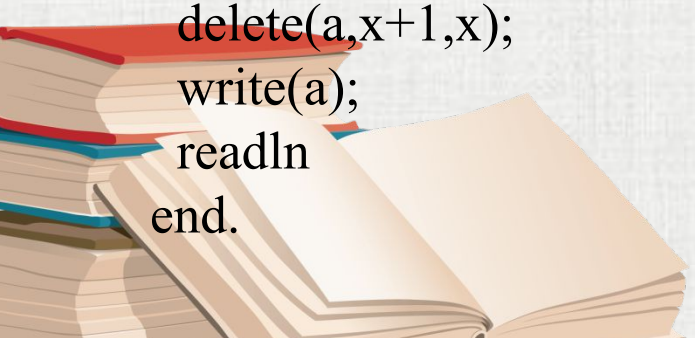
Процедура удаления части строки

DELETE(*строка*, *начальный номер*, *количество символов*)

Удаляет из исходной строки указанное количество символов.

Пример : Дано слово, состоящее из четного числа букв. Вывести на экран его первую половину.

```
Var  i,x:byte;   a,p:string;
begin
  repeat
    write('Введите слово из четного числа букв');
    readln(a);
    x:=length(a);          {определяем длину слова}
  until (x mod 2 = 0);
  x:= x div 2;          {применяем целочисленное деление}
  delete(a,x+1,x);
  write(a);
  readln
end.
```



Процедура вставки подстроки в строку

INSERT(*строка1*, *строка2*, *позиция*);
Строка1 вставляется в строку2 начиная с указанной позиции.

Тренировочные задания.

1. Дано предложение. Определить число вхождений в него некоторого символа.
2. Дано предложение. Заменить в нем все вхождения буквосочетания "ах" на "ух".
3. Дано слово. Проверить, является ли оно "перевертышем", т.е. читается одинаково как с начала, так и с конца.
4. Дано слово:
   a. удалить из него первую из букв "о", если такая буква есть;
   b. удалить из него последнюю из букв "т", если такая буква есть.
5. Дано предложение. Удалить из него все буквы "с".