There is a lot of statements you can use besides **read, readln, write,** and **writeln**. Let's take an instance where you want something to happen that depends on the situation. There are a few ways you can effect this. There is the **while**_____**do** expression , the **if** _____ **then** expression, and the **case** _____ **of** expression (where the underlined space represents the condition you will insert).

These are called **condition expressions**. You identify the conditions that will trigger a specific response.

We will cover all three in the first grade. See how easy things are in the first grade?

This lesson will concentrate on the while_____do expression.

Here is a sample of how it is used.

```
PROGRAM while_do;                    {this will generate a sequential list of numbers}

VAR
x:integer;                           {defines x as an integer)

begin
x:=1;                                {gives x an inital value of positive 1}
  while x <= 10 do                   {sets up the condition}

    begin
    writeln (x:2);                   {writes the next sequential value of  x and a <cr>}
    x:= x +1;                        {adds 1 to the previous value of  x}
    end;                             {of  listing}

readln;                              {allows you to see what happened by requiring a <cr>}

end.                                 {of program}
```

Look at the program and try and determine what it does. It is a simple program. A picture of the results of this program is included on page 2. Don't look right away.

Take a look at the Program name line. The name is "while_do". Remember, no spaces are allowed in a program name so a '_' is used where a space would normally be placed.
Next, skip down to the line that reads

x:= 1;

this provides the initial value of the variable x. In this case it is positive 1. The statement could have read
x:= 1 ;
but it could not have read

x : = 1;

The initialization operator ':=' must follow immediately after the variable...no spaces allowed.

One last thing before we look at the program. note the line

writeln (x:2);

this says write the value of x and give it 2 spaces then do a <cr>.

OK. Here is the result of the program. Did you figure it out?

That's lesson one. If you are happy about it get out a bag of m&m's and feast. If you aren't happy about it, then gt out a baag of m&m's and save it for the next lesson.... things may start to heat up soon.