by 2. You can use any variable name for the name of the function, and a DEFined function can then be used in any formula. If you say

```
X = FN DOUBLE (Y)
```

then XYBASIC gives X the result of applying the function DOUBLE to the value of Y. Try the following program.

```
NEW
OK
10 DEF FN DOUBLE (I) = I * 2
20 INPUT "VALUE" X
30 PRINT "DOUBLE ("; X; ") ="; FN DOUBLE (X)
40 GOTO 20
RUN
VALUE? 2
DOUBLE ( 2 ) = 4
VALUE? 100
DOUBLE ( 100 ) = 200
VALUE? -165
DOUBLE (-165 ) =-330
VALUE? ^C
BREAK AT LINE 20
OK
```

DEF is legal only in program mode; an ID (Illegal Direct) error will occur if you try to use it in direct mode. An FC (Function Call) error will occur if you try to use a function DEFined in terms of itself, either directly or indirectly.

The variable I in the function DEFinition is called a dummy parameter. The value of the variable used as a dummy parameter is not changed when you evaluate the function. You can also write function DEFinitions with as many parameters as you desire or without any parameters. The following useful example uses a function without parameters to convert numbers (given in decimal, binary or hexadecimal) to octal.

```
NEW
OK
10 DEF FN OCTAL = TEST(N,I) + TEST(N,I+1)*2 + TEST(N,I+2)*4
20 INPUT "VALUE" N
30 PRINT "OCTAL ("; N; ")=";
40 PRINT TEST(N,15);
50 FOR I = 12 TO 0 STEP -3
60 PRINT FN OCTAL;
70 NEXT I
80 PRINT
90 GOTO 20
```