

```

RUN
VALUE? 257
OCTAL ( 257 )= 0 0 0 4 0 1
VALUE? -1
OCTAL (-1 )= 1 7 7 7 7 7
VALUE? #FOFO
OCTAL (-3856 )= 1 7 0 3 6 0
VALUE? &10110101000101
OCTAL ( 11589 )= 0 2 6 5 0 5
VALUE? ^C
BREAK AT LINE 20
OK

```

This program DEFINes a function OCTAL which calculates one digit of the octal representation of N by TESTING three binary digits (which correspond to one octal digit). A FOR loop calls OCTAL to get successive octal digits of N.

In Extended XYBASIC you can DEFINE functions of any type -- integer, floating point or string. The function name you choose determines the result type of the function. The parameters of the function may also be of any type. A TM (Type Mismatch) error will occur if the actual parameters in the FN call are incompatible with the dummy parameters in the DEFINition, or if the result of evaluating the function body is incompatible with the function's result type. For example, the following commands define three different functions.

```

10 DEF FN F (X, Y) = (X + Y) / 2
20 DEF FN ROT$ (A$, I) = MID$ (A$, I+1) + LEFT$ (A$, I)
30 DEF FN BAD (X) = "BAD" + STR$ (X)

```

The floating point function FN F returns the average of its two floating point arguments. The string function FN ROT\$ rotates its string argument left by the number of characters given by its numeric argument; examples of its use are given under String Functions in Section 4, where you will also find definitions of the functions +, MID\$ and LEFT\$. The function FN BAD is a floating point function, but evaluating its function body gives a string value; therefore any call of FN BAD will result in a TM error.

### Variable Types

In Extended XYBASIC variables can be of three types: floating point, integer, or string. As noted in Section 2, variable names may consist of a letter followed by letters or digits, optionally followed by one of the characters !, %, or \$. Variable names ending in a letter or digit, or in !, are floating point variables; variable names ending in % are integer variables; and variable names ending in \$ are string variables. The optional type characters are not considered part of the variable name, so for example DOG and DOG! represent the same floating point variable. However, the string variable DOG\$ and the integer variable DOG% are different from the floating point variable DOG, and different from each other.

Initially the default variable type for all variables in Extended XYBASIC is floating point, so any variable name not ending in % or \$ is assumed to be the name of a floating point variable. You can change the default