DEF With Parameters

When your program encounters a statement containing a previously defined function-name with parameters, the parameters values are passed to the function in the same order in which they are listed. The expression is evaluated using those values, and the function is assigned the value of the expression at that time. String values can be passed only to string parameters. Numeric values can be passed only to numeric parameters.

If you define a function with parameters, it must appear with the same number of parameters when you use it in your program.

Recursive Definitions

A DEF statement may reference other defined functions (the expression may include previously defined function-names). However, a DEF statement may not be directly or indirectly recursive (self-referencing).

Direct recursion occurs when you use the function-name in the expression of the same DEF statement. (This would be similar to writing a dictionary definition that included the word you were trying to define.)

Indirect recursion occurs when the expression contains a function-name, and in turn the expression in the DEF statement of that function (or other function subsequently referenced) includes the original function-name. (This would be similar to looking up the dictionary definition of a word, finding that the definition included other words that you needed to look up, and then discovering that the definitions led you directly back to your original word.)

Examples

```
100 DEF PAY(OT)=40*RATE+1.5*RATE*OT
110 RATE=4.00
120 PRINT PAY(3)
RUN
178
```

Defines PAY so that each time it is encountered in a program the pay is figured using the RATE of pay times 40 plus 1.5 times the rate of pay times the overtime hours.

```
100 DEF RND20=INT(RND*20+1)
```

Defines RND20 so that each time it is encountered in a program an integer from 1 to 20 is given.

```
100 DEF FIRSTWORD$(NAME$)=SEG$(NAME$,1,POS(NAME$," ",1)-1)
```

Defines FIRSTWORD$ to be the part of NAME$ that precedes a space.