[7.13] Write to program & function arguments

TI BASIC passes function and program arguments by value, not by reference. For example,

```
myprog(y)
```

actually passes the contents of the variable *y*, not the address of *y* or a pointer to *y. myprog()* can write to *y*, but if *y* is a global variable, *y* is not changed. For example, suppose you have a global variable *y* and you call this program, with *myprog(y):*

```
myprog(x)
Prgm
7→x
EndPrgm
```

The program runs without error, but the global variable *y* is not changed to 7, even though *x* contains 7 within the program.

To bypass this limitation, pass program argument variable names as strings:

```
myprog("y")
```

Use indirection within *myprog()* to access the variable:

```
myprog(x)
Prgm
local k
...
#x→k  ©Save y to k
...
k→#x  ©Save k to y
...
EndPrgm
```

This method works for programs, not functions, because functions cannot write to global variables. You can write to the function arguments as local variables, though, and indirection is not needed:

```
myfunc(x)
Func
...
x→k   ©Save y to k
...
k→x   ©Save k to y
...
EndFunc
```

The comments show what happens with the call *myfunc(y).*