[7.47] Global variables not evaluated when local variable has the same name

Unexpected behavior can result when global variable names are used as input in a *Request* statement. If the program has defined a local variable of the same name, the global variable will not be evaluated. This situation arises, for example, when you use global variables to save floating-point numbers. The program below demonstrates:

```
t()
Prgm

local x

clrio

dialog
 request "x",x
enddlog

expr(x)→x
disp "x is",x

EndPrgm
```

Expected operation ensues if you supply a number for *x* at the *Request* prompt. For example, if you enter 4.22 at the prompt, then the Program I/O screen shows

```
x is
4.22
```

However, suppose you have previously stored the value 4.17723887656 to a global variable *x*, for use in this program. To avoid re-typing the value, you just enter x at the prompt. In this case, *x* is not evaluated, and the Program I/O screen shows

```
x is
x
```

A work-around is to use unusual local variable names, in the optimistic hope that the program's user will not have a global variable of the same name. Characters from the International sub-menu of the CHAR menu may be appropriate.