[7.40] Recall expression without variable value substitution

You can use the RCL operation ([2nd] [STO] on the TI-89 and TI-92+) to recall an expression without substituting variable values. For example, suppose you store an equation to *eqn1*:

```
a+b=c→eqn1
```

then if you evaluate *eqn1* at the entry line, the result a+b=c is returned as expected. However, if you store numeric values to 1, 2, and 3 to *a*, *b* and *c*, respectively, then evaluating *eqn1* returns *true*, since 1+2 = 3. You can recall the original expression and avoid the variable value substitution with [RCL].

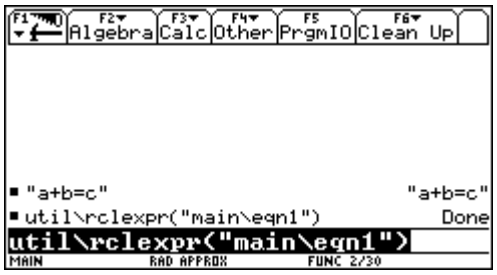
This operation can be accomplished in a TI Basic program, but the RCL operation is not programmable. Instead, we need to use a different method, as this code example shows.

```
rclexpr(ë)
Prgm
local ö,ü

Try:newFold(ä):else:endTry    © Create new folder if necessary
setFold(ä)→ö                  © Make new folder current and save old folder
#ë→ü                          © Evaluate expression
setFold(#ö)                   © Restore old folder
util\copyto_h("ü")            © Return expression as a string

EndPrgm
```

The expression is evaluated in a folder in which the variables do not exist, so substitution cannot take place. The expression is returned as a string, otherwise the variable values would be substituted. The argument ë is a string which specifies the variable and its folder. The folder must be specified, since the expression is evaluated in a different folder. As an example, suppose that the variable *eqn1* and the variables *a*, *b* and *c* are all in the *main*\ folder. If *rclexpr()* is in the *util*\ folder, then the call to get the expression in *eqn1* is

```
util\rclexpr("main\eqn1")
```

and the history display looks like this after execution:



Note that the expression is returned in the second history line, above the call to *rclexpr()*.

The *copyto_h()* function is used to send the result to the history area; see tip [7.8], *Copy program results to home screen*. The expression is evaluated in the folder *ä*, which is an empty folder containing no variables. *rclexpr()* creates this folder with *newFold()* if it does not already exist. *newFold()* is a block in a *Try ... endTry* structure, which prevents errors if folder *ä* already exists.

rclexpr() is useful in its own right, and the general idea can also be used in your own programs, but not in functions, because *Try ... endTry* is not allowed in functions.