[7.50] Modify loop control variables within the loop

If you are familiar with other programming languages but not TI Basic, you may not be aware that you can modify all of the control variables of a *For ... EndFor* loop, within the loop block. While this is a dubious practice in terms of code readability and maintenance, it can sometimes result in smaller code.

For example, this loop will never terminate, because the index *i* never reaches the terminal value *n*:

```
1→n
For i,1,n
 i+1→n
EndFor
```

You can change the index step size within the loop:

```
1Ø→n
1→step
For i,1,n,step
 if i=4
  2→step
EndFor
```

For this example, *i* steps through 1, 2, 3, 4, 6, 8, 10.

The example below is a more realistic demonstration. This segment of code processes a list of expressions *e* to return a list of the individual terms of the expression. Initially, list *e* has only one element, but it grows as terms are separated, and shrinks as they are removed to the output list *o*.

```
while dim(e)≠Ø

 for i,1,dim(e)                              © Process each element of 'e'
   e[i]→ex
   part(ex)→px
   augment(left(e,i-1),right(e,dim(e)-i))→e  © 'e' reduced by one element
   if px=Ø or px=1 then
    augment(o,{ex})→o
   elseif px=2 then
    part(ex,Ø)→pxØ
    if pxØ="+" then
     augment(e,{part(ex,1),part(ex,2)})→e    © 'e' increased by two elements here ...
    elseif pxØ="-" then
     augment(e,{part(ex,1),⁻part(ex,2)})→e   © ... and here
    else
     augment(o,{ex})→o
    endif
   endif
 endfor

endwhile
```

The processing details are not relevant; the point is that the *For* loop will execute until the index reaches the current dimension of *e*, which is updated each pass through the loop.