

[7.36] Input arguments for 'keyboard' programs

The TI-89 / TI-92 Plus support up to nine 'keyboard' programs. These programs can be executed from any folder by pressing [DIAMOND][1] for *kbdprgm1()*, [DIAMOND][2] for *kbdprgm2()*, and so on. Unlike other programs, keyboard programs cannot have input arguments. However, keyboard programs *can* use the *ans()* and *entry()* functions. *ans(n)* is a built-in function which returns answer *n* from the history display, and *entry(n)* returns entry *n*. These functions can be used to provide input to keyboard programs. As a simple example, suppose we want *kbdprgm2()* to convert a temperature to Kelvin degrees. The program is

```
kbdprgm2()  
Prgm  
dialog  
  text string(tmpcnnv(expr("ans(1)"),_°K))  
enddlg  
EndPrgm
```

Note that we cannot just use *ans(1)* as the *tmpCnnv()* argument. If we did, the TI Basic tokenizer would permanently replace *ans(1)*, in the program, with the actual value of *ans(1)* at the time the program ran. Using the expression *expr("ans(1)")* prevents this silly behavior.

To use the program, enter the temperature to convert, then press [DIAMOND][2], and a dialog box shows the converted temperature. For example, to convert 32°F to K, press

```
32 [ ] [°] [f] [ENTER]
```

to put 32_°F in the history display, then press [DIAMOND] [2] to run *kbdprgm2()*. A dialog box shows the result of 273.15K.

For conversion programs such as this, it would be even more useful if the result is returned to the history display, as with functions. This is not possible using built-in TI Basic functions, but it *can* be done with the *copyto_h()* ASM program described in tip [7.8]. This version of *kbdprgm2()* uses *copyto_h()* to return the result to the history display:

```
kbdprgm3()  
Prgm  
Local t,tk  
expr("ans(1)")→t  
tmpCnnv(t,_°K)→tk  
util\copyto_h(tmpcnnv("t","tk"))  
EndPrgm
```

As before, the temperature is entered in the history display. When [DIAMOND][3] is pressed to run *kbdprgm3()*, the result is directly returned to the history display.

(Credit to Andrew)