

---

## Appendix A: Anti-tips - things that can't be done

---

This section lists some things that you may want to do, or might think you should be able to do, but the TI-89 / TI-92 Plus or TI Basic do not support. This section might save you some time in finding out that you can't implement a particular feature.

1. Your programs cannot, in general, use the text editor or matrix editor - but see tip [7.10].
2. You cannot write to a text variable.
3. Many built-in functions, for example, *nSolve()* and *NewPlot*, won't accept local variables as arguments. Use global variables, and delete them with *DelVar* when your program exits.
4. The TI-89 / TI-92 Plus do not have built-in functions to create cascading drop-down menus, as used, for example, on the home screen toolbar items. However, the program *MnuPak* (<http://www.calc.org/programs/proginfo.php?id=2716>) can be used to do this.
5. Variables cannot be larger than about 64K. Trying to create or use a variable larger than this will cause a memory error, and subsequent memory leakage, in which the available RAM mysteriously decreases. The only known fix is a reset.
6. A program cannot archive itself. However, one program can archive and unarchive another.
7. Functions can read from global variables, but cannot store to (write to) global variables. Programs can read and write to global variables. Functions *can* write to their argument variables, which are considered local variables.
8. Functions cannot write to the program I/O display.
9. Programs cannot, in general, return results to the history area, but see tip [7.8].
10. You cannot reference single-row or single-column matrices (vectors) with a single index. Example: *matrix[3]* won't work. This works: *matrix[3, 1]*.
11. You cannot use just *ans()* in a function. The TI Basic tokenizer will replace *ans(1)* with the actual value of *ans(1)* the first time the program runs. Instead, use *expr("ans(1)")*. See tip [9.10] for an example. This behavior can also be avoided by archiving the function before running it. *entry()* exhibits the same behavior.
12. You cannot delete statistics plot definitions within a program or function. There is no command to do this. However, see tip [4.15] for a possible work-around.
13. You cannot directly write to data variable elements. See tip [3.8] for a workaround.
14. Number base conversions (►Hex, ►Dec, ►Bin) only work in Auto or Exact modes. In Approximate modes, the base conversions return a *Domain error* message. Keep this in mind if you write programs that use the number base conversions.
15. You must use global variables, not local variables, to do symbolic math in programs. This is in the *89/92+ Guidebook*, page 291.
16. You cannot archive variables beginning with the underscore character '\_'. The operating system considers these to be system variables and prevents the archive. The underscore character is typically used to define custom user units.
17. You cannot pass Data variables as program or function arguments. If this is necessary, convert the data variable to a matrix (if all the columns have the same number of rows), and pass the matrix. See tip [3.29] for details.
18. You cannot use elements of lists or matrices as arguments in the *Request* command. To accomplish this, first store the element to a variable, use the variable in *Request*, then store the variable to the element.