
TI-89 / TI-92 Plus Tip List 10.0

Compiled by Doug Burkett, dburkett@infinet.com
Created 12 aug 99, revised 20 July 2002

Copyright 1999-2002 Doug Burkett. All rights reserved.

Dedicated to my wife,
Janet Katherine Scholz Burkett,
who cheerfully tolerates all this silliness...

Important!

Neither I, Doug Burkett, nor any of the contributors to this tip list are responsible in any way for any damage of any kind that you or anyone else may incur from using the information in this tip list. These tips are provided for educational purposes only. The contributors and I believe that all information is accurate and reliable, but we make no expressed or implied warrantee or guarantee to that effect. Some tips may involve hardware or software modifications to the calculator that may void the manufacturer's warrantee.

This document is a collection of useful tips and hints relating to the TI-89 and TI-92 Plus calculators. The tips come from user experience and many other sources, and apply to all facets of calculator operation and programming. The tip list has grown to include reference material, humor and oddities, so perhaps the name *Tip List*, is no longer appropriate, but it stands for now.

The tip list is not a buglist, a wishlist or a FAQ. I maintain a wishlist, and good FAQs are available. While a FAQ is, by definition, frequently asked questions, this list instead describes useful techniques which may be 'infrequently asked'. These are sophisticated calculators and some features are not obvious. All users have different skills, and what is obvious to some is not obvious to all.

I want the tip list to be the ultimate, exhaustive source of TI-89 / TI-92+ operation. If you find a good tip or clever solution, please email me. In general I give credit to the first person to submit a particular tip. Let me know if your tip is from a manual, or you really learned it from someone else. If a tip gives no credit, that only means I came up with it on my own, not that I am the original inventor.

Bhuvanesh Bhatt has generously volunteered to keep this list on his site:

<http://triton.towson.edu/~bbhatt1/ti/>

Thanks Bhuvanesh! Bhuvanesh also maintains a bug list for these calculators.

Through the gracious efforts of Andrew Cacovean, the tip list is also available here:

http://www.angelfire.com/realm/ti_tiplist/

Andrew developed and maintains this site. You can download the entire tip list or individual tips, tip list code and also the wishlist. Andrew has also written some very good examples of operation.

Some tips include code listings, which can be typed in. Most of the code is included in a zip file called *tlcode.zip*. I use lots of comments and white space in these listings, so you will save considerable RAM or archive memory by deleting these comments. Beginning with version 9.0, the code listings shown in the tips are not the same as those in the *tlcode.zip* file. The code shown in the tips contains explanatory comments. Those comments are not actually included in the *tlcode.zip* versions, to reduce RAM/flash requirements. Code from earlier tip list versions may still have all the comments included.

The tips are organized in these sections:

- [1] *Miscellaneous*: Tips that do not fit in any other category, as well as humor and hardware
- [2] *Computer Algebra System (CAS)*: Using the CAS for symbolic algebra
- [3] *Data structures*: Working with matrices, lists and data variables
- [4] *Graphing and plotting*: Function graphing, data plotting, operations on the graph screen
- [5] *Graph-Link Cable and Software*: Using the TI GraphLink cables and software
- [6] *Math*: Mostly numerical mathematics
- [7] *TI Basic programming*: Programming techniques, tricks and work-arounds
- [8] *String variables*: Manipulating string variables
- [9] *User interface*: User input and output, dialog boxes and results formatting.
- [10] *Units Conversions*: Using the built-in units conversions
- [11] *Solving*: Using the various solving functions, including *solve()*, *nsolve()* and *csolve()*
- [12] *C Programming*: A few tips on writing and using C and ASM programs

There are also a few appendices of reference material:

- [A] *Anti-tips - things that can't be done*: Summary of calculator and CAS limitations
- [B] *More resources*: Includes FAQs, TI documentation and useful web sites
- [C] *Glossary*: of common and obscure terms
- [D] *Quick reference guide*: to calculator commands and functions
- [E] *Tip list programs and functions*: Alphabetic list of all routines in *tlcode.zip*

Note that the table of contents includes active links to all the tips.

I hope you find this useful or interesting, and that you will consider contributing your ideas and corrections to this list.

Doug Burkett
Eaton, OH USA
dburkett@infinet.com

Contents

Contributors	0-9
Revision record: 9.0 to 10.0	0-10
1.0 Miscellaneous tips	
[1.1] Discussion Group etiquette	1-1
[1.2] TI92+ manual error for Logistic regression equation	1-2
[1.3] Alphabetic list of reserved variable names	1-2
[1.4] Importing and exporting PC data with the TI-89 / TI-92 Plus	1-3
[1.5] Creating picture variables on a PC	1-7
[1.6] Manual error for <i>SetMode()</i> codes for Angle modes	1-9
[1.7] Undocumented error codes	1-9
[1.8] Make a protective plastic 'skin' for your calculator	1-10
[1.9] Physical characteristics of TI-89 and TI-92 Plus	1-10
[1.10] Patents for TI-89 and TI-92+	1-12
[1.11] "Low Battery" indication voltages	1-16
[1.12] Processor clock speed and battery voltage	1-16
[1.13] Developer's guide describes calculator internal operation	1-17
[1.14] Cleaning a calculator after spills and other accidents	1-17
[1.15] Permission to copy TI guidebooks	1-19
[1.16] Use your TI-89 / TI-92 Plus as an on-screen ruler and protractor	1-20
[1.17] Quick tips	1-28
[1.18] Myth and legend from the TI discussion groups	1-29
[1.19] Kraftwerk's <i>Pocket Calculator</i> song	1-38
[1.20] TI-92 Plus internal photographs	1-40
[1.21] TI-89 Internal photographs	1-45
[1.22] TI Calc-Con '02 trip reports	1-49
[1.23] Ray Kremer interview on TI Student Features page	1-51
2.0 Computer Algebra System (CAS) tips	
[2.1] Force complete simplification with repeated evaluation	2-1
[2.2] Try constraints or <i>solve()</i> in symbolic equality tests	2-1
[2.3] Use <i>when()</i> for absolute values in integrals	2-2
[2.4] Use Exact mode to ensure <i>Limit()</i> success	2-3
[2.5] Try manual simplification to improve symbolic integration	2-3
[2.6] CAS uses common convention for integral of x^n	2-3
[2.7] Use <i>when()</i> in integration in AMS 2.03	2-4
[2.8] Use <i>expand()</i> with respect to variable for faster results	2-4
[2.9] Find more symbolic integrals, faster, with Real mode and constraints	2-4
[2.10] CAS square root simplification limitations	2-5
[2.11] Force <i>getNum()</i> and <i>getdenom()</i> to return correct results with <i>part()</i>	2-6
[2.12] Testing equivalence of symbolic expressions	2-7
[2.13] Try <i>comDenom()</i> to find limits	2-8
[2.14] Define functions that CAS can manipulate	2-8
[2.15] Use change of variable in limits of <i>when()</i> functions	2-9
[2.16] Find partial fractions with <i>expand()</i>	2-9
[2.17] Use a 'clean' or empty folder for CAS operations	2-10
[2.18] Delay evaluation to simplify expressions	2-10
[2.19] Restrict function arguments to integers	2-11
[2.20] <i>factor()</i> with respect to list elements	2-12
[2.21] Differentiation and integration fail with <i>part()</i>	2-12
[2.22] Methods for some CAS functions	2-13
[2.23] Try <i>comDenom()</i> for faster partial factoring	2-15
[2.24] Infinity ⁰ evaluates to 1	2-16

[2.25] Factor on complex i to show polynomial complex coefficients	2-18
[2.26] $\text{limit}()$ fails with piecewise functions	2-18

3.0 Data structure tips

[3.1] Using indirection and $\text{setFold}()$ with matrix elements	3-1
[3.2] Calculate table indices instead of searching	3-1
[3.3] Delete matrix rows and columns	3-2
[3.4] Reverse list elements	3-2
[3.5] Unexpected $\text{NewMat}()$ operation in Cylindrical and Spherical vector modes	3-2
[3.6] Convert True/False list to 0's and 1's	3-3
[3.7] Replacement for Fill instruction in functions	3-4
[3.8] Work-around for writing to data variable elements	3-4
[3.9] Replace matrix rows and columns	3-5
[3.10] Appending is faster than $\text{augment}()$ for adding elements to lists; $\text{seq}()$ is even faster	3-6
[3.11] Store anything in a matrix	3-6
[3.12] Don't use Δlist function in data editor	3-7
[3.13] Find matrix minor and adjoint	3-7
[3.14] Randomize list elements	3-9
[3.15] Turn Pretty Print off to display commas in lists, matrices	3-10
[3.16] Use arrays with more than 2 dimensions	3-10
[3.17] Conditional tests on lists	3-18
[3.18] Swap list elements	3-18
[3.19] Test a list for identical elements	3-19
[3.20] Find matrix minimum and maximum	3-19
[3.21] Convert matrices to single-row or single-column vectors	3-20
[3.22] Create lists with logarithmic element spacing	3-21
[3.23] Use single bits as flags to save status information	3-21
[3.24] Use equations to build lists of lists and matrices	3-27
[3.25] Internal error with $\text{seq}()$ and Boolean operators in list index calculation	3-28
[3.26] Find indices of specific elements in lists and matrices	3-29
[3.27] Fast symbolic determinants	3-34
[3.28] Fill a list or matrix with a constant	3-35
[3.29] Convert data variables to matrices	3-35
[3.30] Singular value decomposition (SVD) of a matrix	3-36
[3.31] Store new list element at index greater than dimension	3-38

4.0 Graphing and plotting tips

[4.1] Plot "vertical" lines on graph screen	4-1
[4.2] Use DispG to force update to Δx and Δy	4-2
[4.3] Truth plots	4-2
[4.4] Plot data and functions simultaneously	4-4
[4.5] 3D parametric line and surface plots	4-6
[4.6] Graphing piece-wise defined functions with " " operator	4-6
[4.7] Graphing piece-wise defined function with a unit step function	4-7
[4.8] Plot functions that return lists	4-7
[4.9] Faster plots of slow functions	4-8
[4.10] Faster plots of integral functions	4-9
[4.11] 3D data plots	4-15
[4.12] Random scatter plots of 'noisy' functions	4-16
[4.13] Bode plots	4-18
[4.14] Completely clear Graph screen	4-22
[4.15] Delete statistics plots from programs	4-22

5.0 GraphLink Cable and Software Tips

[5.1] Unarchive variables to fix GraphLink transmission errors	5-1
[5.2] Backing up and restoring individual variables may be safer than Get Backup	5-1
[5.3] Do not restore a HW1 backup to a HW2 calculator	5-1
[5.4] Try Var-Link to fix GraphLink transmission errors	5-2

[5.5] Use TalkTI to develop PC applications which communicate with the TI-89 / TI-92 Plus	5-2
[5.6] Build your own GraphLink cable (Hess's BCC Serial Link)	5-2
[5.7] GraphLink switchbox aids CBR/CBL programming	5-13
[5.8] Opening variables in GraphLink changes some characters	5-13
[5.9] GraphLink software sends either TI-89 or TI-92 Plus files	5-15
[5.10] Run GraphLink under Windows XP	5-16
[5.11] Improve font size of printed program source code	5-16

6.0 Math Tips

[6.1] Simulate <i>poly</i> function of TI-85/86	6-1
[6.2] Use rectangular complex mode for faster results	6-1
[6.3] Improving floating-point solutions to simultaneous equations	6-2
[6.4] Gamma and log-gamma functions	6-3
[6.5] Round numbers to significant digits	6-5
[6.6] Linear regression through a fixed point	6-6
[6.7] Complex derivatives	6-7
[6.8] Convert trigonometric expressions to exponential format	6-8
[6.9] Convert floating-point numbers to exact fractions	6-9
[6.10] Exact solutions to cubic and quartic equations	6-9
[6.11] Rounding floating point numbers	6-14
[6.12] Find faster numerical solutions for polynomials	6-14
[6.13] Find coefficients of determination for all regression equations	6-17
[6.14] Use <i>norm()</i> to find root mean square (RMS) statistic for matrices	6-19
[6.15] Convert equations between rectangular and polar coordinates	6-19
[6.16] Transpose operator and dot product find adjoint and complex scalar product	6-20
[6.17] Use dd.mmssss format to convert angles faster	6-21
[6.18] Use <i>iPart()</i> and <i>int()</i> effectively	6-22
[6.19] Sub-divide integration range to improve accuracy	6-22
[6.20] Generating random numbers	6-23
[6.21] Evaluating polynomials	6-24
[6.22] Linear Interpolation	6-24
[6.23] Step-by-step programs	6-25
[6.24] Fast Fibonacci Numbers	6-26
[6.25] Polar and rectangular coordinate conversions	6-26
[6.26] Accurate numerical derivatives with <i>nDeriv()</i> and Ridder's method	6-27
[6.27] Find Bernoulli numbers and polynomials	6-39
[6.28] Bilinear interpolation	6-44
[6.29] Dirac's delta (impulse) and Heaviside (step) functions	6-55
[6.30] Spherical coordinate conventions	6-56
[6.31] Fast 2nd-order interpolation with QuadReg	6-57
[6.32] Accurate approximate solutions to quadratic equations with large coefficients	6-60
[6.33] Sine and cosine integrals	6-61
[6.34] Error function for real arguments	6-64
[6.35] Cumulative normal distribution and inverse	6-66
[6.36] Integration may return 'undef' for identical integration limits	6-70
[6.37] Random number generator algorithm	6-71
[6.38] Finance Flash Application function finds days between two dates	6-71
[6.39] Convert equations to a parameterized form	6-71
[6.40] Write functions with multiple input argument types	6-72
[6.41] Integration error in AMS 2.05	6-73
[6.42] <i>stdDev()</i> and <i>variance()</i> find sample (not population) statistics	6-75
[6.43] Dot product (scalar product) for complex vectors	6-77
[6.44] Nest <i>min()</i> in <i>max()</i> to limit an argument	6-78
[6.45] Determine if a point is inside a triangle	6-78
[6.46] <i>solve()</i> and <i>nsolve()</i> ignore constraints with local variables	6-79
[6.47] Unsigned infinity displays as undef	6-81
[6.48] Use R>Pθ for four-quadrant arc tangent function	6-81
[6.49] <i>Taylor()</i> polynomial function finds tangent line	6-82

[6.50] The Savage benchmark	6-83
[6.51] Review of TI-89 / TI-92 Plus Calculus Tools	6-88
[6.52] Return complex results in Real mode with 0i	6-90
[6.53] Implicit derivatives	6-91
[6.54] Delete variables after using numeric solver	6-92
[6.55] Algorithms for <i>factor()</i> and <i>isPrime()</i>	6-93
[6.56] Fourth-order splice joins two functions	6-94
[6.57] Sum binary "1" digits in an integer	6-110
[6.58] Extract floating-point mantissa and exponent	6-111
[6.59] Accelerate series convergence	6-112
[6.60] Faster, more accurate exponential integrals	6-114
[6.61] Find more accurate polynomial roots	6-116
[6.62] Try asymptotic expansion for functions of large arguments	6-127
[6.63] Faster numerical matrix inverses	6-130
[6.64] Calculate through undef and get correct results	6-131

7.0 TI Basic Programming Tips

[7.1] Create evaluated Y= Editor equations in programs	7-1
[7.2] Using the built-in function documentation in CATALOG	7-1
[7.3] Using language localization	7-2
[7.4] Return error codes as strings	7-4
[7.5] Considerations for referencing external programs	7-5
[7.6] Recursion limits	7-6
[7.7] Use <i>return</i> instead of <i>stop</i> in programs for better flexibility, and to avoid a crash	7-6
[7.8] Return program results to home screen	7-8
[7.9] Passing optional parameters to functions and programs	7-10
[7.10] Calling built-in applications from your programs	7-11
[7.11] Run TI Basic programs before archiving for better execution speed	7-11
[7.12] Access a variable from any folder with "_" (underscore)	7-12
[7.13] Write to program & function arguments	7-13
[7.14] Determine calculator model and ROM version in programs	7-14
[7.15] Avoid <i>For ... EndFor</i> loops	7-16
[7.16] Use <i>when()</i> instead of <i>if...then...else...endif</i>	7-16
[7.17] Returning more than one result from a function	7-17
[7.18] Simplest (?) application launcher	7-18
[7.19] Bypass programs locked with <i>ans(1)</i> and 4→errornum:passerr	7-20
[7.20] Running programs within a program	7-20
[7.21] Use <i>undef</i> as an argument	7-21
[7.22] Local documentation for functions and programs	7-21
[7.23] Passing user function names as program/function arguments	7-21
[7.24] Use a script for self-deleting set-up programs	7-23
[7.25] Use scripts to test program and functions	7-24
[7.26] <i>dim()</i> functions slow down with lists and matrices with large elements	7-25
[7.27] <i>getMode()</i> returns strings in capital letters	7-25
[7.28] Use long strings with <i>setMode()</i> in custom menus	7-26
[7.29] Table of <i>getConfig()</i> list elements	7-28
[7.30] <i>sign()</i> function is not compatible with other programming languages	7-29
[7.31] <i>return</i> shows variables during debugging	7-29
[7.32] Documenting programs	7-30
[7.33] Local functions must be declared as local variables	7-33
[7.34] Use linefeed char(10) for separate lines in program Help	7-34
[7.35] Conditional test in <i>If ... EndIf</i> evaluates all 'and' arguments	7-35
[7.36] Input arguments for 'keyboard' programs	7-35
[7.37] Simulate SWITCH or CASE statement with GOTO and indirection	7-36
[7.38] Why use TI Basic?	7-37
[7.39] Quickly delete locked, archived variables	7-39
[7.40] Recall expression without variable value substitution	7-40
[7.41] Indirection bug with local variables	7-41

[7.42] Find variable names used in expressions	7-42
[7.43] Faster function calls with list and matrix arguments	7-45
[7.44] Local subroutines execute faster than global subroutines	7-52
[7.45] Display text in multiple fonts and Pretty-Print expressions	7-53
[7.46] <i>Custom</i> command makes menu labels for trapped function keys	7-60
[7.47] Global variables not evaluated when local variable has the same name	7-62
[7.48] Create functions in programs	7-62
[7.49] Read and write text variables	7-63
[7.50] Modify loop control variables within the loop	7-64

8.0 String Variable Tips

[8.1] Convert integers to strings without extra characters	8-1
[8.2] String substitutions	8-1
[8.3] Creating strings that include quote characters	8-2
[8.4] Count occurrences of a character in a string	8-2
[8.5] <i>string()</i> uses the current number display format	8-3
[8.6] Convert strings to upper- and lower-case	8-3
[8.7] Replacement for <i>mid()</i> and <i>right()</i> functions use position instead of count	8-4
[8.8] Relational operators compare strings	8-5

9.0 User Interface Tips

[9.1] Use icons in toolbars	9-1
[9.2] User interface considerations	9-2
[9.3] Take advantage of <i>ok</i> system variable in dialog boxes	9-4
[9.4] Displaying more lines on the TI-92 Plus program I/O screen	9-4
[9.5] Default values for variables in Request	9-4
[9.6] Position cursor with <i>char(2)</i>	9-5
[9.7] Creating 'dynamic' dialog boxes	9-6
[9.8] Dialog box limitations	9-9
[9.9] Display all 14 significant digits	9-10
[9.10] Group fraction digits for easier reading	9-10
[9.11] Access all custom menus from any custom menu	9-12
[9.12] Use one <i>Request</i> (in a Dialog box) for more than one variable	9-13
[9.13] Disable ALPHA default in TI-89 AMS 2.05 dialog boxes	9-17
[9.14] Change Auto/Exact/Approx modes fast	9-18
[9.15] [2nd] [ENTER] quickly selects, copies entries to command line	9-20
[9.16] Use "Save Copy As" to display the current variable name in editors	9-20
[9.17] Speed up the keyboard	9-21
[9.18] Use multiple commands on entry line with ":"	9-22
[9.19] Use a more efficient RPN interface	9-23
[9.20] Use a graphical equation editor (equation writer)	9-24
[9.21] Programs for simple equation libraries	9-25
[9.22] Minimize TI-89 alpha entry inefficiencies	9-29

10.0 Units Conversion Tips

[10.1] Units calculations are approximate in Auto mode	10-1
[10.2] Convert compound units to equivalent units	10-1
[10.3] Remove units from numbers	10-2
[10.4] Add units to undefined variables	10-2
[10.5] Use <i>tmpcnv()</i> to display temperature conversion equations	10-2
[10.6] Add units to built-in Units menu	10-3

11.0 Solving Tips

[11.1] Try <i>nsolve()</i> if <i>solve()</i> and <i>csolve()</i> fail	11-1
[11.2] <i>zeros()</i> ignores constraint for complex solutions	11-1
[11.3] Try <i>cZeros()</i> and <i>cSolve()</i> to find real solutions	11-1
[11.4] Using <i>solve()</i> with multiple equations and solutions in programs	11-2
[11.5] Using bounds and estimates with <i>nsolve()</i>	11-4

[11.6] Use <i>solve()</i> as multiple-equation solver	11-6
[11.7] Saving multiple answers with <i>solve()</i> in program	11-7
[11.8] Try <i>solve()</i> for symbolic system solutions	11-8
[11.9] <i>nSolve()</i> may return "Questionable Accuracy" warning even with good solutions	11-8
[11.10] <i>solve()</i> may return false solutions for trigonometric expressions	11-10
[11.11] <i>exp►list()</i> fails with <i>solve()</i> in user functions	11-11
[11.12] <i>nSolve()</i> may ignore solution constraints	11-11

12.0 C Programming Tips

[12.1] Advantages of TIGCC over TI SDK	12-1
[12.2] Use C functions in expressions with AMS 2.05	12-1
[12.3] TI Basic extension app template	12-2

Appendix A: Anti-tips A-1

Appendix B: More resources

[B.1] FAQs	B-1
[B.2] TI Documentation	B-29
[B.3] Web sites	B-30
[B.4] Bibliography	B-33

Appendix C: Glossary C-1

Appendix D: Command quick reference guide D-1

Appendix E: Tip list programs and functions E-1

Contributors

Contributors are credited in each tip. I would also like to list them here, and thank them again.

Andrew
Alex Astashyn
Billy
cj
Martin Daveluy
ES
Glenn E. Fisher
John Gilbertson
Titmité Hassan
Paul King
Kosh DV
Daniel Lloyd
Francesco Orabona
Rick A.
TM
Frank Westlake

Andy
Bez
Damien Cassou
Jordan Clifford
Stuart Dawson
Fabrizio
Lars Frederiksen
Mike Grass
Rick Homard
Eric Kobrin
Ray Kremer
Christopher Messick
Roberto Perez-Franco
Samuel Stearley
Rafael Velazquez
Hank Wu

Kenneth Arnold
Bhuvanesh Bhatt
Andrew Cacovean
cybernesto
George Dorner
Larry Fasnacht
Gp
Frank Mori Hess
Sam Jordan
Kevin Kofler
Cass Lewart
Olivier Miclo
Mike Roberts
TipDS
Gary Wardall
ZLX

Revision record

Various changes:

Page numbering now section-local. New section 12, *C Programming Tips*. 'More resources material' divided into lettered appendices. New Appendix E, *List of programs and functions*.

Contributors:

Correct spelling of Samuel Stearley. New contributors: cybernesto, Carlos Becker, Francesco Orabona, Hernan Rivera, Rafael Velazquez..

Tips changed:

Introduction: Update Bhuvanesh' URL.

[1.16], [7.8], [7.36], [7.40] Update for new version of *copyto_h()*

[1.1] Links fixed

[1.10] Add more patents

[1.7] Add Bhuvanesh' method

[2.10] Considerably clarified, thanks to Carlos Becker

[2.13] Limit problem fixed in AMS 2.05

[3.4] Add *listrev()* function

[4.1] Added *when()* method

[6.4] Rename to *Gamma, log-gamma and factorial functions*. Add factorial function.

[9.5] Add method of saving defaults as a list

[9.14] Substantially changed

[B-1] Add actual TI FAQs contents, not just topics.

[B-4] New bibliography

[C] Many corrections and additions.

[D] Correct entries for *avgRC()*, *setMode()*, *Table*, *Logistic*.

New tips added:

[1.15] Permission to copy TI guidebooks

[1.16] Use your TI-89 / TI-92 Plus as an on-screen ruler and protractor

[1.17] Quick tips

[1.18] Myth and legend from the TI discussion Groups

[1.19] Kraftwerk's "Pocket Calculator" song

[1.20] TI-92 Plus internal photographs

[1.21] TI-89 internal photographs

[1.22] TI Calc-Con '02 trip report

[1.23] Ray Kremer interview on TI Student Feature page

[2.23] Try *comDenom()* for faster partial factoring

[2.24] Infinity⁰ evaluates to 1

[2.25] Factor on complex *i* to show polynomial complex coefficients

[2.26] *limit()* fails with piecewise functions

[3.24] Use equations to build lists of lists and matrices

[3.25] Internal error with *seq()* and Boolean operators in list index calculation

[3.26] Find indices of specific elements in lists and matrices

[3.27] Fast symbolic determinants

[3.28] Fill a list or matrix with a constant

- [3.29] Convert data variables to matrices
- [3.30] Singular value decomposition (SVD) of a matrix
- [3.31] Store new list element at index greater than dimension
- [4.14] Completely clear Graph screen
- [4.15] Delete statistics plots from programs
- [5.8] Opening variables in GraphLink changes some characters
- [5.9] GraphLink software sends either TI-89 or TI-92 Plus files
- [5.10] Run GraphLink under Windows XP
- [5.11] Improve font size of printed program source code
- [6.48] Use $R>P\theta()$ for four-quadrant arc tangent function
- [6.49] *Taylor()* polynomial function finds tangent line
- [6.50] The Savage Benchmark
- [6.51] Review of TI-89 / TI-92 Plus Calculus Tools
- [6.52] Return complex result in Real mode with 0i
- [6.53] Implicit derivatives
- [6.54] Delete variables after using numeric solver
- [6.55] Algorithms for *factor()* and *isPrime()*
- [6.56] Fourth-order splice joints two functions
- [6.57] Sum binary "1" digits in an integer
- [6.58] Extract floating-point mantissa and exponent
- [6.59] Accelerate series convergence
- [6.60] Faster, more accurate exponential integrals
- [6.61] Find more accurate polynomial roots
- [6.62] Try asymptotic expansion for functions of large arguments
- [6.63] Faster numerical matrix inverses
- [6.64] Calculate through undef and get correct results
- [7.38] Why use TI Basic?
- [7.39] Quickly delete locked, archived variables
- [7.40] Recall expression without variable value substitution
- [7.41] Indirection bug with local variables
- [7.42] Find variable names used in expressions
- [7.43] Faster function calls with list and matrix arguments
- [7.44] Local subroutines execute faster than global subroutines
- [7.45] Display text in multiple fonts, and Pretty-Print expressions
- [7.46] 'Custom' command makes menu labels for trapped function keys
- [7.47] Global variables not evaluated when local variable has the same name
- [7.48] Create functions in programs
- [7.49] Read and write text variables
- [7.50] Modify loop control variables within the loop
- [8.6] Convert strings to upper- and lower-case
- [8.7] Replacement for *mid()* and *right()* functions use position instead of count
- [8.8] Relational operators compare strings
- [9.17] Speed up the keyboard
- [9.18] Use multiple commands on entry line with ":"
- [9.19] Use a more efficient RPN interface
- [9.20] Use a graphical equation editor (equation writer)
- [9.21] Programs for simple equation libraries
- [9.22] Minimize TI-89 alpha entry inefficiencies
- [11.12] *nSolve()* may ignore solution constraints
- [12.1] Advantages of TICGG over TI SDK
- [12.2] Use C functions in expressions with AMS 2.05
- [12.3] TI Basic extension app template

Changes to anti-tips:

New anti-tips 17, 18.

Anti-tip 11: Fix reference to [9.11] to [9.10]