TEXAS INSTRUMENTS

TI-89
TI-92 Plus

指南手册

Advanced Mathematics
Software Version 2.0
(高等数学软件，版本 2.0)

美国专利号 4,405,829。由 RSA Data Security, Inc 特许。

© 1999 Texas Instruments(美国德州仪器公司)版权所有

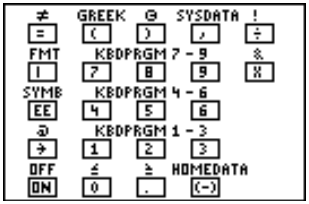
TI-89 快捷按键

综述

◆ APPS	Flash 应用程序列表。
2nd [±±]	在最近选择的两个应用程序之间切换或拆分屏幕。
◆ −, ◆ +	对比度明亮或暗淡调整。
◆ ENTER	计算近似答案。
◆ ⌃, ◆ ⌄	将光标移到顶部或底部(在编辑起中)。
↑ ⌃, ↑ ⌄	在历史区域中滚动巨大的对象。
↑ ‹, ↑ ›	将光标左方或右方突出显示。
2nd ⌃, 2nd ⌄	向上或向下翻页(在编辑起中)。
2nd ‹, 2nd ›	将光标移到最左或最右方。

屏上键盘布局图(◆ EE)

按下 ESC 来退出布局图。



键盘布局图是显示键盘上没有标记出的快捷方式。如下面所示，按下 ◆，然后再按下可用的按键。

◆ =	≠
◆ (	存取希腊字母 (参阅下一列)
◆)	◉ (注释)
◆ ,	将图形坐标复制到 sysdata
◆ ÷	! (阶乘)
◆ |	显示 FORMATS 对话框
◆ 1 – ◆ 9	运行程序 kbdprgm1() 到 kbdprgm9()
◆ ×	& (附加)
◆ EE	屏上键盘布局图
◆ STO▸	@
◆ ON	关闭单元，下次当您开启单元时，会返回当前的应用程序屏。
◆ 0 (零)	≤
◆ .	≥
◆ (-)	将图形坐标复制到 主屏幕历史区域。

Alpha 规则

alpha	键入一个小写字母。
↑	键入一个大写字母
2nd [a-lock]	小写 alpha lock
↑ alpha	大写 alpha lock
alpha	退出 alpha lock

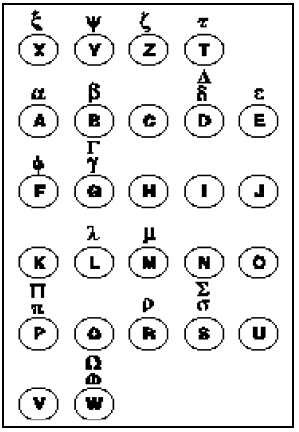
3D 作图

⌃, ⌄, ‹, ›	动态化图形
+, −	改变动态化速度
X, Y, Z	沿坐标轴查看
0	返回原始画面
|	改变图形格式样式
×	扩展 / 正常画面

希腊字母

◆ (	存取希腊字母集
◆ (alpha + 字母	存取小写希腊字母。例如: ◆ (alpha [W] 是显示 ω。
◆ (↑ + 字母	存取大写希腊字母。例如: ◆ (↑ [W] 是显示 Ω。

如您按下了非存取希腊字母的按键组合，那么您会得到该按键的正常字母。



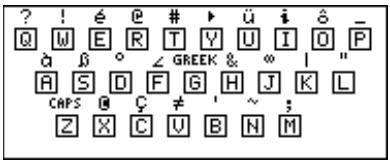
TI-92 Plus 快捷按键

综述

◆ APPS	Flash 应用程序列表
2nd [⊞]	在最近选择的两个应用程序之间切换或拆分屏幕
◆ D	将图形坐标复制到 sysdata(系统数据)
◆ F	显示 FORMATS(格式)对话框
◆ H	将图形复制到主屏幕历史区域
◆ N	创建新变元
◆ O	打开现存的变元
◆ S	另存为…
◆ −, ◆ +	对比度明亮或暗淡调节
◆ ENTER	计算近似答案
◆ ON	关闭单元，下次当您开启单元时，会返回当前的应用程序屏。
◆ 1 – ◆ 9	运行程序 kbdprgm1() 到 kbdprgm9()

屏上键盘布局图(◆ KEY)

按下 ESC 来退出布局图。



对于 TI-92 Plus 键盘上没有标出的快捷方式，请参阅下表。重音标记和希腊字母，请参阅下一列。

2nd Q	?
2nd W	! (阶乘)
2nd R	@
2nd T	# (间接)
2nd H	& (附加)
2nd X	◉ (注释)
◆ =	≠
◆ 0 (零)	≤
◆ .	≥

编辑

◆ ⊙	将光标移到顶部
◆ ⊙	将光标移到底部
2nd ⊙	将光标移到最左方
2nd ⊙	将光标移到最右方
⊙, ⊙	滚动历史区域的巨大对象
2nd ⊙, 2nd ⊙	向上或向下翻页
◆ X	剪切
◆ C	复制
◆ V	粘贴

3D 作图

⊙, ⊙, ⊙, ⊙	动态化图形
+, −	改变动态速度
X, Y, Z	沿坐标轴查看
0 (零)	返回原始画面
F	改变图形格式样式
×	扩展 / 正常画面

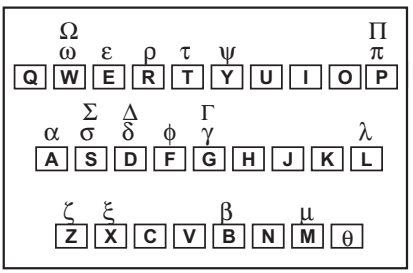
重音标记

2nd A + 字母	à、è、ì、ò、ù、À、È、Ì、Ò、Ù
2nd C + 字母	ç、Ç
2nd E + 字母	á、é、í、ó、ú、ý、Á、É、Í、Ó、Ú、Ý
2nd N + 字母	ã、ñ、õ、Ã、Ñ、Õ
2nd O + 字母	â、ê、î、ô、û、Â、Ê、Î、Ô、Û
2nd U + 字母	ä、ë、ï、ö、ü、ÿ、Ä、Ë、Ï、Ö、Ü

希腊字母

2nd G	存取希腊字母集
2nd G + 字母	存取小写希腊字母。例: 2nd G W 是显示 ω。
2nd G ↑ + 字母	存取大写希腊字母。例: 2nd G ↑ W 是显示 Ω。

如您按下了非存取希腊字母的按键组合，那么您会得到该按键的正常字母。



重要事项

对于任何程序和书面材料，德克萨斯仪器公司将不提供明示或默示的保证，包括但不限于任何可销售性和特定用途适合性的默示的保证，并仅将本材料以“现况如此，概不负责”的方式提供。

无论任何情况，德克萨斯仪器公司将不向任何人承担由于购买和使用这些材料而引起的或相关的特别、附随、附带或间接损害的责任，德克萨斯仪器公司唯一和独有的义务，无论行为的形式，不超过本设备的购买价格。此外，德克萨斯仪器公司对任何他方使用这些材料而造成的任何种类的损害不负责任。

内容表

本指南手册是叙述如何使用 TI-89 / TI-92 Plus 的。内容表可助您寻找“快速入门”信息和有关 TI-89 / TI-92 Plus 功能的信息。附表 A 提供了寻找 TI-89 / TI-92 Plus 每一功能和指令的详细信息的合适地方。

Flash 应用程序 v
击键差异 vii
最新消息 ix

第一章: 快速入门

准备 TI-89 2
准备 TI-92 Plus 3
设置对比度和选择语言 4
作一个函数图形 8
打开和关闭 TI-89 / TI-92 Plus 10
设置显示器对比度 11
TI-89 的键盘 12
TI-92 Plus 的键盘 13
修饰键 14
输入字母字符 17
主屏幕 19
输入数字 21
输入表达式和指令 22
显示结果的格式 25
在输入行中编辑表达式 28
菜单 30
自定义菜单的使用 33
选择应用程序 34
设置模式 36
利用清理菜单来开始一个新习题 39
目录对话框的使用 40
储存和重新调用变元值 43
重新利用前一输入项或上一答案 45
从历史区域自动粘贴输入项或答案 48
显示器状态行中的指示符 49
寻找软件版本和 ID 号码 51
输入复数 52
汇编语言程序 55
链接两个单元 57
传送变元、Flash 应用程序和文件夹 58
升级产品软件(基本码) 60

内容表

第二章: 综述

执行运算 66
符号操作 69
常数和测量单位 70
基础函数作图 71
参数作图 72
极坐标作图 73
序列作图 74
三维作图 75
微分方程作图 77
附加作图提示 79
表格 80
拆分屏幕 81
数据 / 矩阵编辑器 82
统计图和数据图 83
编程 87
文本操作 89
数字解答者 90
数字基 91
内存和变元管理 92

附件 A: 函数和指令

快速搜寻定位器 96
运算的字母顺序列表 100

支持、服务和保证信息

遇到困难时 219
电池信息 220
支持和服务信息 222
保修信息 223

Flash 应用程序

应用程序

Flash 功能允许从附上的 CD-ROM、TI 网址和其它计算器下载不同的应用程序到 TI-89 / TI-92 Plus 内。

在为 TI-89 / TI-92 Plus 下载新应用程序之前，请细致阅读和接受 TI-89 / TI-92 Plus 应用程序 CD-ROM 上的许可协议。

硬件 / 软件需求

安装 Flash 应用程序前，您必须具备:

- 一台计算机和串行端口；如从 CD-ROM 安装应用程序，则需要有 CD-ROM 驱动器。
- 作为软件另外提供的 TI-GRAPH LINK™ 以及连接计算机和计算器的电缆。
  如您需要 TI-GRAPH LINK 软件和连接电缆，请访问 TI 网址，具体地址是: **education.ti.com/guides**。

计算机硬件设置

欲设置:

1. 将 TI-GRAPH LINK 电缆的小端插在 TI-89 的底端口或 TI-92 Plus 的顶端口内。
2. 将另一端连接在计算机的串行端口，如有需要，请使用 25 到 9 插脚的适配器。

从 CD-ROM 安装 Flash 应用程序

欲安装应用程序:

1. 将 TI-89 / TI-92 Plus Applications CD-ROM 插入计算机的 CD-ROM 驱动器内；
2. 从计算机启动 TI-GRAPH LINK 软件；
3. 在 Link (链接)菜单上单击 Send Flash Software ► Applications and Certificates;
4. 找出 CD-ROM 内的 Flash 应用程序并双击之。此时，应用程序就会复制到计算器内。

备注: 有关从或到计算机的传送，请参阅 TI-GRAPH LINK 的指南手册。

运行 Flash 应用程序

欲运行应用程序:

1. 在 TI-89 / TI-92 Plus 上，按下 [◆] [APPS] 来显示 FLASH APPLICATIONS 菜单；
2. 用光标键 ⊕⊖ 将应用程序突出显示并按下 [ENTER]；

将 Flash 应用程序转移到另一台 TI-89 / TI-92 Plus 上

当接收或发送计算器显示出电池低电量的信息时，请勿尝试转移应用程序。

备注: 本指南手册是使用 TI-89 的画面。

1. 用与 TI-89 / TI-92 Plus 一起提供的计算器对计算器连接电缆来连接计算器。
2. 在发送计算器上:
   a. 按下 [2nd] [VAR-LINK]
   b. 按下:
      TI-89: [2nd] [F7]
      TI-92 Plus: [F7]
   c. 将 Flash 应用程序突出显示并按下 [F4] (此时，在选定的项目左方会显示出记号 ✓)
3. 在接收计算器上:
   a. 按下 [2nd] [VAR-LINK]
   b. 按下 [F3]
   c. 选择: 2:Receive
   d. 按下 [ENTER]

4. 在发送计算器上:
   a. 按下 [F3]
   b. 选择: 1:Send to TI-89/92 Plus
   c. 按下 [ENTER]

备份 Flash 应用程序

欲将应用程序备份到计算机:

备注: 有关从或到计算机的传送，请参阅 TI-GRAPH LINK 的指南手册。

1. 在计算器上，按下;
   TI-89: [HOME]
   TI-92 Plus: [♦] [HOME]
2. 从计算机上启动 TI-GRAPH LINK 软件;
3. 在 Link 菜单上单击 Receive Flash Software;
4. 选择一个或多个 Flash 应用程序并单击「添加」;
5. 单击「确定」;
6. 将应用程序保存到计算机内并记下此信息，以备日后参考之用。

删除 Flash 应用程序

欲从计算器删除应用程序:

备注: 欲选择所有 Flash 应用程序，可使用 [F5] All 菜单。

1. 按下 [2nd] [VAR-LINK] 以显示 VAR-LINK 屏;
2. 按下:
   TI-89: [2nd] [F7]
   TI-92 Plus: [F7]
2. 将 Flash 应用程序突出显示并按下 [F4] (此时，在选定的项目左方会显示出记号 ✓);
3. 按下 [F1] 并选择 1:Delete
   — 或 —
   按下 [←] (此时会显示出一个请求确认的信息);
4. 按下 [ENTER] 以确认删除。

击键差异

在使用 TI-89 / TI-92 Plus 进行各项运算时，会出现一些的击键差异。下表列出两计算器大部分命令的击键操作。

功能	TI-89	TI-92 Plus
字母		
一个小写字母(a-s, u, v, w)	[alpha] A-S, U-W	A-S, U-W
一个小写字母(t, x, y, z)	T, X, Y, Z	T, X, Y, Z
数个小写字母	[2nd] [a-lock]	
数个小写字母完毕	[alpha]	
数个大写字母	[↑] [a-lock]	[2nd] [CAPS]
数个大写字母完毕	[alpha]	[2nd] [CAPS]
功能键		
F6	[2nd] [F6]	[F6]
F7	[2nd] [F7]	[F7]
F8	[2nd] [F8]	[F8]
浏览		
在历史区域上下滚动巨大的对象	[↑] ⊖, [↑] ⊕	[⇧] ⊖, [⇧] ⊕
在输入行中将光标移到最左或最右	[2nd] ⊖, [2nd] ⊕	[2nd] ⊖, [2nd] ⊕
对角移动	⊖ 和 ⊖ ⊖ 和 ⊕ ⊕ 和 ⊖ ⊕ 和 ⊕	⊖ ⊕ ⊖ ⊕
功能		
显示主屏幕	[HOME]	[◆] [HOME]
剪切	[◆] [CUT]	[◆] X
复制	[◆] [COPY]	[◆] C
粘贴	[◆] [PASTE]	[◆] V
目录	[CATALOG]	[2nd] [CATALOG]
显示运算单位对话框	[2nd] [UNITS]	[◆] [UNITS]
正弦	[2nd] [SIN]	[SIN]
余弦	[2nd] [COS]	[COS]
正切	[2nd] [TAN]	[TAN]
LN	[2nd] [LN]	[LN]
e^x	[◆] [e^x]	[2nd] [e^x]
EE	[EE]	[2nd] [EE]

功能	TI-89	TI-92 Plus
符号		
▶ (换算三角形)	[2nd] [▶]	[2nd] [▶]
_ (下划线)	[◆] [_]	[2nd] [_]
θ (Theta)	[◆] [θ]	[θ]
| ("与")	[|]	[2nd] [|]
' (质数)	[2nd] [']	[2nd] [']
° (度)	[2nd] [°]	[2nd] [°]
∠ (角)	[2nd] [∠]	[2nd] [∠]
Σ (Sigma)	[CATALOG] Σ (	[2nd] [Σ]
x^{-1} (反商)	[CATALOG] ^-1	[2nd] [x^{-1}]
空格	[alpha][␣]	空格键
隐藏快捷方式		
将数据放在系统变元内	[◆] [,]	[◆] D
希腊字符	[◆] [(] [alpha] 或 [◆] [(] [↑]	[◆] G 或 [◆] G [↑]
键盘布局图	[◆] [EE]	[◆] [KEY]
将数据放在主屏幕历史区域	[◆] [(-)]	[◆] H
抑音(à, è, ì, ò, ù)	[2nd] [CHAR] 5	[2nd] A a, e, i, o, u
变音符(ç)	[2nd] [CHAR] 5 6	[2nd] C c
高音(á, é, í, ó, ú, ý)	[2nd] [CHAR] 5	[2nd] E a, e, i, o, u, y
颚化符(ã, ñ, õ)	[2nd] [CHAR] 5 6	[2nd] N a, n, o
脱字号(â, ê, î, ô, û)	[2nd] [CHAR] 5	[2nd] O a, e, i, o, u
元音变音符(ä, ë, ï, ö, ü, ÿ)	[2nd] [CHAR] 5	[2nd] U a, e, i, o, u, y
? (问号)	[2nd] [CHAR] 3	[2nd] Q
β (Beta)	[2nd] [CHAR] 5 6	[2nd] S
# (间接)	[2nd] [CHAR] 3	[2nd] T
& (附加)	[◆] [×] (乘)	[2nd] H
@ (任意常数)	[◆] [STO▶]	[2nd] R
≠ (不等符)	[◆] [=]	[2nd] V
! (阶乘)	[◆] [÷]	[2nd] W
注释(圆-C)	[◆] [)] ◉	[2nd] X ◉
新建	[F1] 3	[◆] N
打开	[F1] 1	[◆] O
副本另存为…	[F1] 2	[◆] S
格式对话框	[◆] [|]	[◆] F

最新消息

高等数学软件版本 2.0 简介

为了使 TI-89 和 TI-92 Plus 拥有可下载的计算器应用程序，TI 发展了 Advanced Mathematics Software Version 2.0(高等数学软件版本 2.0)。

高等数学软件版本 2.0 是当前的高等数学软件版本 1.xx 的增强版，备有 1.xx 版的所有功能，改良后的功能允许多个可下载的计算器软件和语言本土化。此新功能以用户数据档案与计算器软件应用程序之间超过 702_KB 的 Flash 内存重新分配为您的 TI-89 / TI-92 Plus。

以前所有的 TI-89 和 TI-92 Plus 模均可升级成 2.0 版。尽管如此，某些 TI-89 和所有 TI-92 Plus 模单元内的用户数据档案最大仅占 384-KB 的空间。702-KB Flash 内存与计算器软件应用程序共享。

您可从 TI 网址(**http://www.ti.com/calc/flash**)将高等数学软件版本 2.0 下载到您的计算器上，然后利用 TI-GRAPH LINK™ 软件和计算器对计算器连接电缆(另外提供)将其转移到您的 TI-89 / TI-92 Plus 上。您也可以利用计算器对计算器连接电缆将软件从一台 TI-89 / TI-92 Plus 转移到另一台上。高等数学软件可在 TI 网址(**http://www.ti.com/calc/flash**)上免费得到。

语言本土化

TI-89 / TI-92 Plus 可本土化成其它语言，这些免费应用程序可将提示、错误信息和大多数功能翻译成系列语种之一。

详情请参阅:
第一章

改良的用户界面

改良的用户界面允许文件夹瓦解 / 扩展和将 CATALOG 菜单扩展成包含应用程序功能和用户定义功能。

利用 Flash ROM 的可升级性

TI-89 / TI-92 Plus 采用 Flash 技术，可让您升级到的最新软件而无须购买一台新的计算器。

作为新的功能向用户提供，您可对您的 TI-89 / TI-92 Plus 进行电子升级。新软件版本包括免费的保养升级。同时，在 TI 网址备有新的软件和更多的升级程序供用户选购。

欲从 TI 网址下载升级程序，您必须具备连接了 Internet 的计算机、TI-GRAPH LINK™ 软件和计算机对计算器连接电缆(另外提供)。您也可利用提供的计算器对计算器连接电缆将软件产品(基本码)从一台 TI-89 / TI-92 Plus 转移到另一台上，这样，接收的计算器也有运行软件的许可。

自定义菜单

TI-92 Plus 的新功能就是自定义菜单，可让您创建自有的工具栏菜单。自定义菜单可含有任何可用的功能、指令或字符集。TI-92 Plus 备有默认的自定义菜单，对于该菜单，您可修改或重新定义之。

快速入门

1

准备 TI-89 2
准备 TI-92 Plus 3
设置对比度和选择语言 4
作一个函数图形 8
打开和关闭 TI-89 / TI-92 Plus 10
设置显示器对比度 11
TI-89 的键盘 12
TI-92 Plus 的键盘 13
修饰键 14
输入字母字符 17
主屏幕 19
输入数字 21
输入表达式和指令 22
显示结果的格式 25
在输入行中编辑表达式 28
菜单 30
自定义菜单的使用 33
选择应用程序 34
设置模式 36
利用清理菜单来开始一个新习题 39
目录对话框的使用 40
储存和重新调用变元值 43
重新利用前一输入项或上一答案 45
从历史区域自动粘贴输入项或答案 48
显示器状态行中的指示符 49
寻找软件版本和 ID 号码 51
输入复数 52
汇编语言程序 55
链接两个单元 57
传送变元、Flash 应用程序和文件夹 58
升级产品软件(基本码) 60

本章的内容可助您更快地掌握 TI-89 / TI-92 Plus 的使用。此外，还通过数个例子来向您说明 TI-89 / TI-92 Plus 的一些主要运算和函数作图功能。

欲掌握更多有关 TI-89 / TI-92 Plus 的信息，请参阅随 CD-ROM 附上的综合指南手册或访问德州仪器公司的网址:

education.ti.com

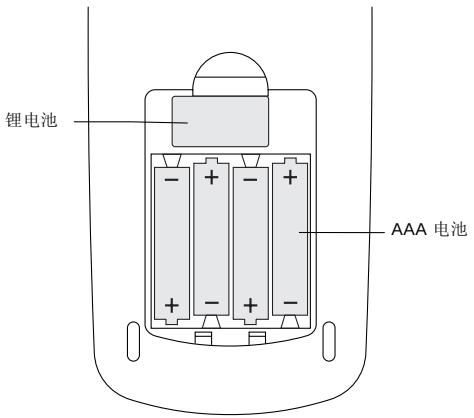
TI-89 是连同四节 AAA 电池一起提供的。本章是叙述如何安装这些电池。此外，也叙述了如何第一次打开计算器、设定显示器对比度、选择语言和从 TI-89 和 TI-92 Plus 查看主屏幕。

安装 AAA 电池

欲安装四节 AAA 电池:

1. 将 TI-89 面向下地放在一快柔软的布上，以避免刮花显示器的表面；
2. 在计算器背部，松开电池盒盖的扣，提起并取走电池盒盖；
3. 拆除新电池的包装并将其装入电池盒内。按照电池盒内的极性(+ 和 –)图示排好电池；
4. 装回电池盒盖。将盒盖的两插脚对准电池盒底部的两条槽后推动盖子，直至盒盖扣上为止。

重要事项: 日后更换电池时，请确认已通过按下 [2nd] [OFF] *来关闭 TI-89。*



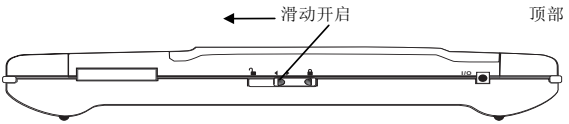
TI-92 Plus 是连同四节 AA 电池一起提供的。本章是叙述如何安装这些电池。此外，也叙述了如何第一次打开计算器、设定显示器对比度、选择语言和从 TI-92 Plus 和 TI-89 查看主屏幕。

安装 AA 电池

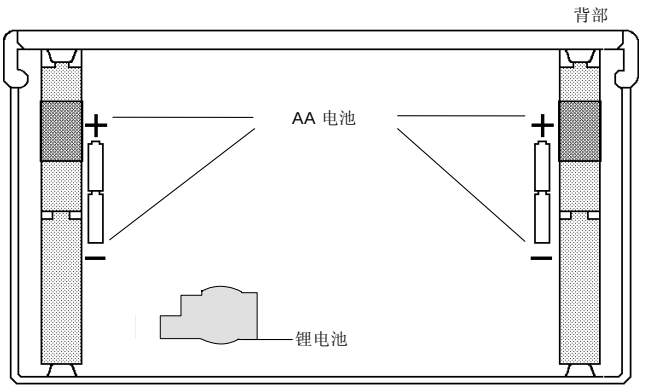
欲安装四节 AA 碱性电池:

1. 将 TI-92 Plus 计算器持直，用手指将顶部的闩推向左方的松锁位置；向下滑动后盖八分之一英寸并从单元上将其拆下。

重要事项: 日后放入电池时，应确保按下了 2nd [OFF] *来关闭 TI-92 Plus。*



2. 将 TI-92 Plus 底朝天地放在一块软布上，以免刮花显示器表面。
3. 装入四节 AA 电池，应确保电池的位置与机内的图示相符。每一电池的正极 (+) 端子均应向着计算器的顶部。



4. 重新装上后盖，并将顶部的闩推向右方的锁定位置，以把后盖在位锁定。

设置对比度和选择语言

打开单元并调整显示器对比度

在 TI-89 / TI-92 Plus 上安装了电池之后，请按下 [ON]。此时，显示器有可能会显得太暗或太亮。

欲按照您的喜好调整显示器的对比度，可按住 [◆] (绿色边框，内有钻石标记)并瞬间按下 [−] (减号键)来使显示变亮；按住 [◆] 并瞬间按下 [+] (加号键)来使显示变暗。

您会看到屏幕上列出一系列的语言供选。您计算器内的语言列表有可能与下面的图例有异。

TI-89 / TI-92 Plus 上备有的语言

英文以外的语言如 Flash 应用程序一样可用。英文是产品软件(基本码)的一部分。您可按照您的需要(视内存的限制)，在计算器内保留多中或少许几种语言，并可简便地在这些语言之间进行切换。在处理期间，您可有选择其它语言来保留或删除的可能。您也可通过 VAR-LINK 屏幕来添加或删除语言应用程序。

语言处理的重要信息

TI-89 / TI-92 Plus 可本土化成系列语言之一。本土化意味着所有菜单名称、对话框、错误信息等都会按您所选定的语言显示出来。

TI-89 / TI-92 Plus 每次仅可本土化成一种语言；然而，您可在单元中保留其它语言，并可随时切换到其它语言。

TI-89 / TI-92 Plus 的初始本土化需要三个阶段:

- **第一阶段 -** 选择要将 TI-89 / TI-92 Plus 本土化的目标语言，线上更详细的指示会在选定语言后显示出来。
- **第二阶段 -** 阅读在第一阶段选定的语言的相关指示信息。
- **第三阶段 -** TI-89 / TI-92 Plus 会本土化成您在第一阶段所选定的语言。您可选择一个或多个应用程序(以后要切换到其它语言时使用)保留在计算器内。如有需要，日后您也可加载一或多个语言应用程序。计算器会随后自动删除未选定的语言。

备注: 在产品软件(基本码)中，英文是不能删除并一直会保持可用的。

TI-89 / TI-92 Plus 的本土化

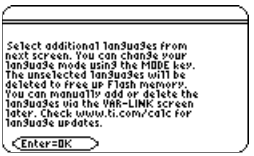
1. 按下光标键(⊖ 或 ⊕)，将指针指向您要设定 TI-89 / TI-92 Plus 的语言上。(您计算器内的语言列表有可能与右边的的图例有异。)

备注: 除非您完成了本土化过程，Select a Language (请选择一种语言)对话在您打开计算器后一直会显示出来。

2. 按下 ENTER 来将 TI-89 / TI-92 Plus 设定到选定的语言。(按下 ESC 可暂停本土化过程和显示主屏幕。)

3. 阅读显示出的信息并按下 ENTER。

此信息是以您先前选定的语言显示。



4. 按下光标键(⊖ 或 ⊕)来移动指针，然后按下 F1 来选择其它要保留的语言。
   — 或 —
   按下 F2 来选择并保留 *all(所有)*语言应用程序。

您不能将英文或在步骤 1 中选定的语言去除选择。

按下 F1 来开启或关闭 ✓ 标记。

5. 按下 ENTER 来完成本土化过程。如有的话，其它选定的语言将会保留在内存中，而未选定的语言将会被删除，以释放 Flash 内存。(按下 ESC 可暂停本土化过程和显示主屏幕。)

如您的 TI-89 / TI-92 Plus 内保留有其它语言应用程序，您可通过 Mode(模式)对话框(请参阅第 36 页)的 Page3 (F3)来改变本土化语言。您可通过 VAR-LINK 屏(请参阅第 58 页)添加或删除语言和其它 Flash 应用程序。

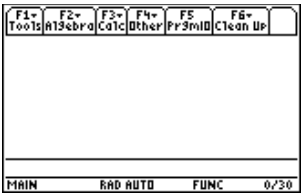
语言应用程序在随附的 CD 光盘内和在 Texas Instruments 的网址上备有。关于 Flash 应用程序和其它语言应用程序的更新，请查看 Texas Instruments 的网址:

education.ti.com

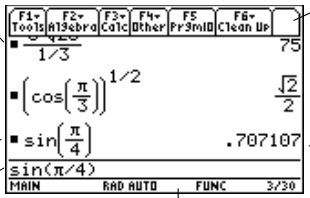
关于主屏幕

选择语言后，会显示出一个空白的主屏幕。

主屏幕可让您执行指令、计算表达式和查看结果。



以下例子含有先前输入的数据并说明主屏幕的主要部分。历史区域中的输入项 / 答案对是以“Pretty print”(工整印刷体)显示。在书写板或在文本薄上书写表达式后，同样会以工整印刷体显示出来。



历史区域
列出您已输入的输入项 / 答案对。当您输入新的输入项时，这些数据对会向上滚屏。

工具栏
让您为在主屏幕上可应用的操作选择显示菜单。欲显示工具栏，可按下 F1、F2 等。

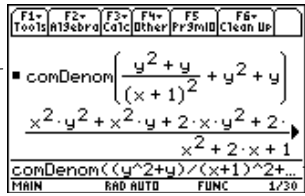
输入项
您的上一输入项。

上一答案
您上一输入项的结果。请记住，结果不会在输入行中显示出来。

输入行
在此输入表达式或指令。

状态行
显示计算器当前的状态。

以下例子显示答案是不与表达式同行。应注意，答案会比屏幕宽度长，箭头(▶)是表明还有后续的答案。输入行含有省略号(…)，该省略号表明输入项超出了屏幕的宽度。



上一输入项
"Pretty print" 处于 ON (开启)状态。当按传统方式书写后，指数、根、分数等都会以同样的格式显示出来。

答案继续
将答案突出显示，按下 ▶ 来向右滚动并查看其剩余部分。请注意，答案不会与表达式同处一行。

表达式继续
按下 ▶ 来向右滚动并查看其输入项的剩余部分。按下 2nd ◀ 或 2nd ▶ 来转到输入行的开头和结尾。

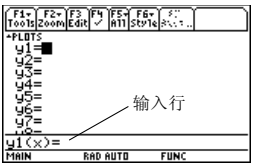
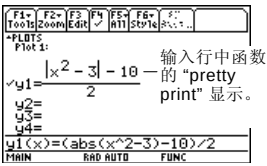
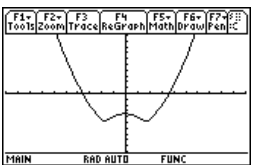
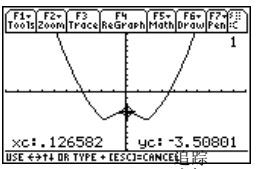
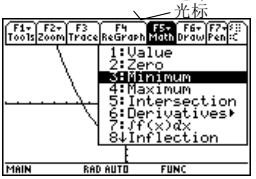
关闭 TI-89 / TI-92 Plus

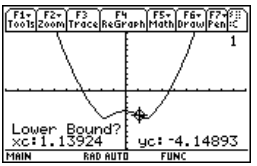
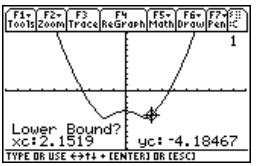
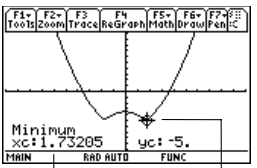
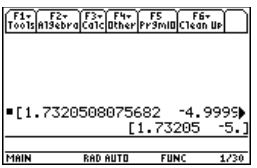
当您想关闭 TI-89 / TI-92 Plus 时，可按下 2nd [OFF]。
(备注: [OFF] 是 ON 键的第二功能。)

作一个函数图形

本章中的例子演示出 TI-89 / TI-92 Plus 的一些作图能力，并说明如何利用 Y=编辑器作一个函数图形。您将会学习到如何输入一个函数、制作一个函数图形、追踪曲线、求最低点和将最小坐标转移到主屏幕上。

通过作出函数 y=(|x²− 3|− 10)/2 的图形来研究 TI-89 / TI-92 Plus 的作图能力。

步骤	TI-89 的击键	TI-92 Plus 的击键	显示内容
1. 显示 Y= 编辑器。	◆ [Y=]	◆ [Y=]	
2. 输入函数 (abs(x²− 3)− 10)/2。	(CATALOG A ENTER X ^ 2 − 3) − 1 0) ÷ 2 ENTER	(2nd [CATALOG] A ENTER X ^ 2 − 3) − 1 0) ÷ 2 ENTER	
3. 显示函数的图形。 *按下* 6 *或将光标移到* 6:ZoomStd *并按下* ENTER *来选择* 6:ZoomStd。	F2 6	F2 6	
4. 打开 Trace(追踪)功能。 *追踪光标、x 和 y 坐标会显示出来。*	F3	F3	
5. 打开 MATH (数学)菜单并选择 3:Minimum。	F5 ⊖ ⊖ ENTER	F5 ⊖ ⊖ ENTER	

步骤	TI-89 的击键	TI-92 Plus 的击键	显示内容
6. 设定下限。 *按下 ▷ (右光标)来移动追踪光标，直至在第二次按下 ENTER 前，x 的下限刚好到达最小结点的左方。*	▷...▷ ENTER	▷...▷ ENTER	
7. 设定上限。 *按下 ▷ (右光标)来移动追踪光标，直至 x 的上限刚好到达最小结点的右方。*	▷...▷	▷...▷	
8. 在上限和下限间找出图形的最低点。	ENTER	ENTER	
9. 将结果转移到主屏幕，然后显示主屏幕。 *将图形坐标复制到主屏幕历史区域的快捷方式:* ***TI-89:*** ◆ (-) ***TI-92 Plus:*** ◆ H	◆ (-) HOME	◆ H ◆ [HOME]	

打开和关闭 TI-89 / TI-92 Plus

您可利用 [ON] 和 [2nd][OFF] (或 [◆][OFF])键来手控打开和关闭 TI-89 / TI-92 Plus。为了延长电池的寿命，APD™ (Automatic Power Down™ - 电源自动关闭)功能会自动让 TI-89 / TI-92 Plus 自行关闭的。

打开 TI-89 / TI-92 Plus

按下 [ON]。

- 如您按下 [2nd][OFF] 来关闭计算器，TI-89 / TI-92 Plus 会返回主屏幕。
- 如您按下[◆][OFF]来关闭计算器，或计算器已通过 APD 自行关闭的话，TI-89 / TI-92 Plus 将会返回您上次使用的任何应用程序处。

关闭 TI-89 / TI-92 Plus

备注: [OFF] *是* [ON] *键的第二功能。*

您也可使用以下的按键来光标 TI-89 / TI-92 Plus。

按键:	说明
[2nd][OFF] (按下 [2nd]，然后按下 [OFF])	设置值和内存内容会由 Constant Memory™ (常驻内存)功能保留下来。可是: • 如果显示出一个错误信息时，您则不能使用 [2nd][OFF]。 • 当您再次打开 TI-89 / TI-92 Plus 时，仍然是显示出主屏幕(与您使用的上一应用程序无关)。
[◆][OFF] (按下 [◆]，然后按下 [OFF])	与 [2nd][OFF] 相类似，不同的是: • 如果显示出一个错误信息，您仍可以使用 [◆][OFF]。 • 当您再次打开 TI-89 / TI-92 Plus 时，计算器会显示出与您退出时完全一样的屏幕。

APD (电源自动关闭)

时间经过了数分钟且没有任何动作时，TI-89 / TI-92 Plus 会自动关闭。此功能称为 APD。

当您按下 [ON]，TI-89 / TI-92 Plus 会显示出与您退出时完全一样的屏幕。

- 显示器、光标和任何错误情形会与您退出时的完全一样。
- 所有设置值和内存内容会得到保留。

在计算器或程序运行当中，APD 是不会进入运作的，除非程序已被暂停运行。

电池

TI-89 使用四节 AAA 碱性电池和一块备份锂电池。而 TI-92 Plus 则使用四节 AA 碱性电池，同样带有一块备份锂电池。

设置显示器对比度

显示器的亮度和对比度取决于室内光线、电池电量、观察角度和显示器对比度的调整。当 TI-89 / TI-92 Plus 关闭后，对比度设置会保留在内存中。

调整显示器对比度

您可调整显示器的对比度来适合您的观察角度和光照条件。

欲:	按下并按住:
增高对比度(暗淡)	◆ 和 −
降低对比度(明亮)	◆ 和 +

TI-89 对比度键　　TI-92 Plus 对比度键

如您按下并按住 ◆ − 或 ◆ + 的时间过长，显示器会完全变成黑屏或空白屏。欲进行精细调节，可按住 ◆ 并轻击 − 或 + 。

更换电池时

如电池电量低，显示器会变得暗淡(尤其是在运算过程中)，此时，您则需要增加对比度。如您要频繁地增加对比度，请尽快更换四节碱性电池。

提示: 在您更换电池后，显示可能会变得极暗。此时，您可利用 ◆ − 来增亮显示。

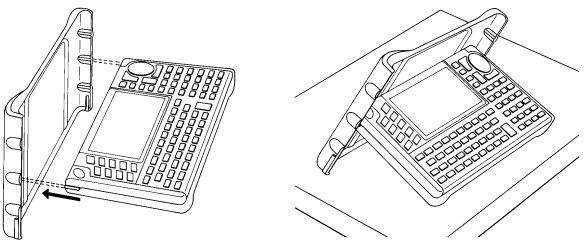
沿显示器下部的状态行也回给出电池的信息。

状态行中的指示符	说明
BATT	电池低电量。
BATT	请尽快更换电池。

将 TI-92 Plus 的滑扣面盖作为支座使用

当在桌面或台面上使用 TI-92 Plus 时，您可使用滑扣面盖将计算器以三个角度之一支撑起来，这样可便于在各种光照条件下查看显示器。

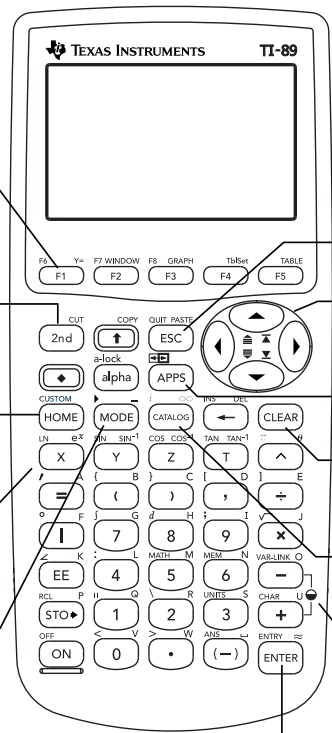
备注: 将 TI-92 Plus 顶边的突轴滑入面盖的孔内。



TI-89 的键盘

本章是让您熟悉 TI-89 键盘上各个按键的。大多数按键可执行两或多项功能，这取决于您首先按下哪个修饰符键。

一些重要按键的简介



光标移动

欲将光标移向某一特定方向，可按相应的光标键(◀、▶、▲ 或 ▼)。

某些 TI-89 应用程序可让您按下:

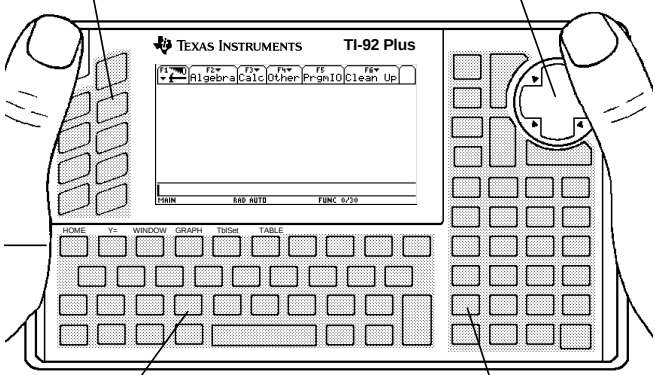
- 2nd ◀ 或 2nd ▶ 来移动到行的开头或结尾。
- 2nd ▲ 或 2nd ▼ 来一次上或下移一个屏幕。
- ◆ ▲ 或 ◆ ▼ 来移动到页面的顶部或底部。
- ▲ 和 ◀、▲ 和 ▶、▼ 和 ◀ 或 ▼ 和 ▶ 来进行对角移动。(同时按下指出的光标键。)

TI-92 Plus 的键盘

当您双手握住计算器时，TI-92 Plus 易于握持的外形及键盘布局，令您仍然能够迅速地接触键盘的任何区域。

键盘区

键盘分为相关按键的数个区域。



功能键
存取显示在屏幕顶部的工具栏菜单。

光标控制盘
将显示光标向八个方向移动，取决于应用程序。

应用程序快捷方式键
与 ◆ 一起使用，让您选择最为常用的应用程序。

QWERTY 键盘
输入文本字符就像使用打字机一样。

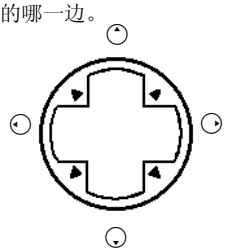
计算器键盘
执行多种多样的数学和科学运算。

光标控制盘

欲移动菜单，可按下光标控制盘的可下压边缘。本指南册使用按键标记如 ◀ 和 ▶ 来指示需要按下光标控制盘的哪一边。

例如，按下 ▶ 可将光标右移。

备注: 对角方向(如 ◥ 等)只用于几何及图形应用程序。



修饰键

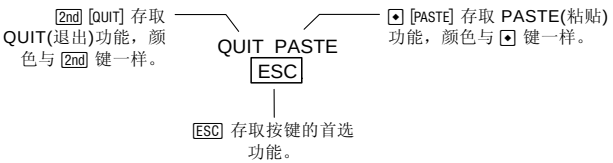
修饰键	说明
2nd (第二功能)	存取您要按下的下一按键的第二功能。在键盘上，第二功能是以 2nd 键相同的颜色印出。
◆ (钻石)	从键盘上选择某些应用程序(第 35 页)、菜单项和其它操作的启用键。在键盘上，应用程序快捷方式是与 ◆ 键相同的颜色印出。
↑ (shift)	键入您要按下的下一字母键的大写字符。处于编辑的目的，↑ 键也可与 ⊕ 和 ⊖ 同用，以使输入行中的字符突出显示。
alpha (仅对 TI-89)	用于键入字母字符，包括空格字符。在键盘上，均以 alpha 键相同的颜色印出。
✎ (手) (仅对 TI-92 Plus)	与光标控制盘一起使用，以操作几何对象。✎ 也用于在图形上画图。

备注: 有关使用 ↑ 和 alpha 的信息，请参阅第 17 页的 "输入字母字符"。

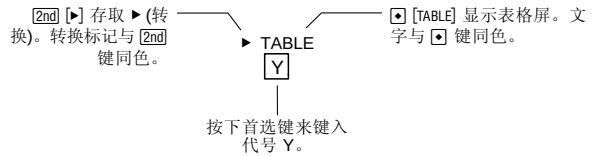
2nd 和 ◆ 修饰符举例

ESC 是数个可执行三种操作的按键之一，具体取决于您是先按下 2nd 还是 ◆。

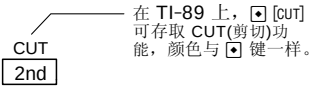
以下的 TI-89 例子中可以看出 2nd 或 ◆ 修饰键与 ESC 键一起使用的效果。



以下的 TI-92 Plus 例子可以看出 2nd 或 ◆ 修饰键与 Y 字母键一起使用的效果。



某些按键只可执行一种额外的操作，这些操作需要用到 2nd 或 ◆ 键，取决于印在键盘上的操作的颜色和定位在按键上的什么位置。



当您按下一个修饰键如 2nd 或 ◆ 时，显示器底部的状态行就会显示出 2ND 或 ◆ 指示符。如您意外按下了修饰键，可再次按下之(或按下 ESC) 来取消其作用。

其它您要熟悉的重要按键

备注: TI-89 与 TI-92 Plus 之间的击键略有不同。完整的列表请参阅本指南手册封面内容中的“击键差异”。

按键	说明
◆ [Y=]	显示 Y=编辑器。
◆ [WINDOW]	显示窗口编辑器。
◆ [GRAPH]	显示图形屏。
◆ [TblSet]	设定表格屏参数。
◆ [TABLE]	显示表格屏。
TI-89: ◆ [CUT] ◆ [COPY] ◆ [PASTE] **TI-92 Plus:** ◆ X (cut) ◆ C (copy) ◆ V (paste)	这些按键让进行剪切、复制或粘贴，以编辑您输入的信息。
2nd [⊞]	将活动边拆屏。
2nd [CUSTOM]	切换自定菜单的打开和关闭(请参阅第 33 请参阅第)。
2nd [▶]	转换计量单位。
TI-89: ◆ [_] **TI-92 Plus:** 2nd [_]	指定计量单位。
←	删除光标左方的字符(退格)。
2nd [INS]	信息输入时的插入和改写模式切换(请参阅第 33 页)。
◆ [DEL]	删除光标右方的字符

重要按键(续上)

按键	说明
TI-89: [I] **TI-92 Plus:** [2nd] [|]	输入用于符号运算的 "With" (与)算子。
[2nd] [∫], [2nd] [d]	执行积分和导数。
[2nd] [∠]	在极坐标、柱面和球面坐标中指定角。
[2nd] [MATH]	显示 MATH (数学)菜单。
[2nd] [MEM]	显示 MEMORY (内存)屏。
[2nd] [VAR-LINK]	显示管理变元和 Flash 应用程序的 VAR-LINK 屏。
[2nd] [RCL]	调用变元内容(请参阅第 44 页)。
TI-89: [2nd] [UNITS] **TI-92 Plus:** [♦] [UNITS]	显示 UNITS 对话框。
[2nd] [CHAR]	是显示 CHAR 菜单，让您选择希腊字母和国际重音字符等。
[2nd] [ANS], [2nd] [ENTRY]	相应调用前一输入项和上一答案(请参阅第 45 页)。

输入字母字符

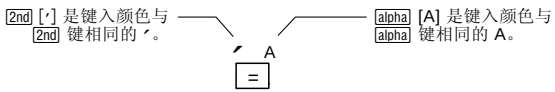
字母字符用在表达式(如 x^2+y^2)中输入变元名称(请参阅第 43 页)，或于文本编辑器中使用。

在 TI-89 中输入字母字符

字母 x、y、z 和 t 在代数表达式中最常用，为此，您要迅速输入它们。在 TI-89 的键盘上，这些字母都是首选键。

[X] [Y] [Z] [T]

其余的字母则作为其它按键的 [alpha] 功能可用，类似上一章介绍的 [2nd] 和 [◆] 修饰符。例如:



在 TI-89 / TI-92 Plus 上键入字母字符

备注: 在 TI-89 上，您不需要使用 [alpha] 或 alpha-lock 来键入 x、y、z 或 t，但必须使用 [↑] 或大写 ALPHA-lock 来键入 X、Y、Z 或 T。

备注: 在 TI-89 上，当您变动应用程序，如从文本编辑器转到主屏幕时，alpha-lock 功能一直是关闭的。

欲:	在 the TI-89 上，可按下:	在 TI-92 Plus 上，可按下:
键入单个小写 alpha 字符。	[alpha] 键，然后按下字母键(状态行会显示出 ◢)。	字母键。
键入单个大写 alpha 字符。	[↑] 键，然后然后按下字母键(状态行会显示出 ◆)。	[↑] 键，然后按下字母键(状态行会显示出 ◆)。
键入空格。	[alpha] [␣] ([(-)] 键的 alpha 功能)	空格键
打开小写 alpha-lock。	[2nd] [a-lock] (状态行会显示出 ◢)。	(不需动作)
打开大写 ALPHA-lock。	[↑] [a-lock] (状态行会显示出 ◢)。	[2nd] [CAPS]
关闭 alpha-lock。	[alpha] (关闭大小写锁定)。	[2nd] [CAPS] (关闭大写锁定)。

键入字母字符 ... (续上)

在 TI-89 上，无论哪种 alpha-lock 打开时:

- 欲键入一个按键首选功能的句号、逗号或其它字符，您则必须关闭 alpha-lock。
- 欲键入第二功能字符如 [2nd] [{]，您则无须关闭 alpha-lock 。在您键入字符后，alpha-lock 会保持打开。

TI-89对话框中的自动 Alpha-Lock

在某些情况下，您无须在 TI-89 上按下 [alpha] 或 [2nd] [a-lock] 来键入字母字符。无论何时，当首先显示出对话框时，自动 alpha-lock 功能是打开的。自动 alpha-lock 功能应用在以下的对话框:

对话框	Alpha-lock
Catalog 对话框	所有命令均会按字母顺序列出。按下一个字母来转到以该字母开始的第一个命令。更为详细的信息请参阅第 40 页。
Units 对话框	在每一单位类别中键入单位的第一个字母或常数。
带有输入项域的对话框。	这些包括但不限于: Create New Folder(创建新文件夹)、Rename(重命名)和 Save Copy As(重命名另存为...)。有关对话框更为详细的信息，请参阅第 31 页。

备注: 欲键入一个数字，可按下 [alpha] 来关闭 alpha-lock 。按下 [alpha] 或 [2nd] [a-lock] 来继续执行键入字母。

在要求仅为数字的输入项对话框中，Alpha-lock 是**不会**打开的。仅接受数字输入项的对话框有: Resize Matrix(调整矩阵)、Zoom Factors(缩放系数) 和 Table Setup(表格设置)。

特殊字符

利用 [2nd] [CHAR] 菜单可在各特殊字符中进行选择。

主屏幕

当您第一次开启您的TI-89 / TI-92 Plus 计算器时，主屏幕就会显示出来。主屏幕可让您执行指令、计算表达式和查看结果。

显示主屏幕

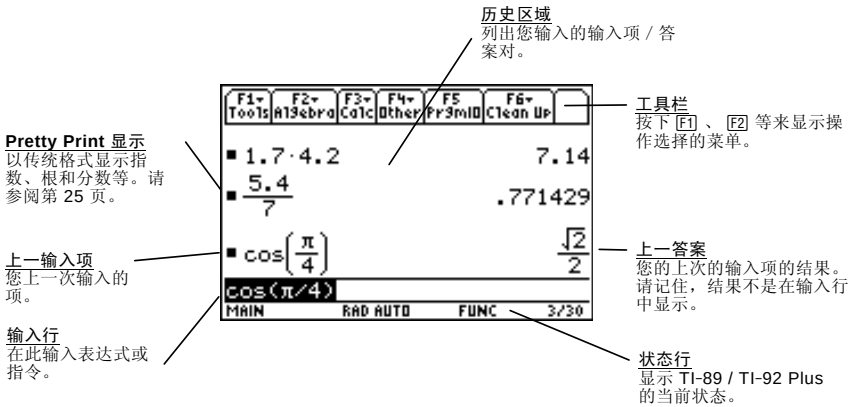
当您使用 [2nd] [OFF] 关闭后重新打开 TI-89 / TI-92 Plus 时，显示器会一直显示出主屏幕。(如 TI-89 / TI-92 Plus 是 APD™ 由自行关闭的，显示器会显示出上一屏幕，该屏幕也许是主屏幕，也许不是。)

欲在任何时候显示主屏幕:

- 按下:
  TI-89: [HOME]
  TI-92 Plus: [♦] [HOME]
  — 或 —
- 按下 [2nd] [QUIT]
  — 或 —
- 按下:
  TI-89: [APPS] [alpha] A
  TI-92 Plus: [APPS] A

主屏幕的各组成部分

以下例子给出了主屏幕各主要部分的简短说明。



历史区域

历史区域可显示多达八对以前的输入项 / 答案(视显示的表达式的复杂性和高度而定)。当显示满时，信息会滚出屏幕顶部。您可使用历史区域来:

- 审阅以前的输入项和答案。您可使用光标查看已滚出了屏幕外的输入项和答案。
- 在输入行上重新调用或自动粘贴前一输入项或答案，以便重新利用或编辑之。请参阅第 46 和 48 页。

滚动历史区域

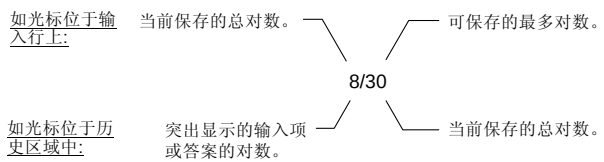
光标通常是位于输入行中。尽管如此，您可将光标移入历史区域中。

欲:	请进行如下:
查看已滚出屏幕外的输入项或答案。	1. 在输入行出，按下 ⊕ 来将上一答案突出显示。 2. 继续使用 ⊕ 来将光标从答案位置移到输入项上，从而进入历史区域中。
转到最旧或最新的历史对。	如光标位于历史区域，可反复按下 [◆] ⊕ 或 [◆] ⊖。
查看超出行长的输入项或答案(▶ 会显示在行尾)。	将光标移到输入项或答案上，反复使用 ⊖ 和 ⊕ 来左右滚动(或 [2nd] ⊖ 和 [2nd] ⊕ 来转到开头或结尾)。
使光标返回输入行。	按下 [ESC] 或按下 ⊖，直至光标返回输入行为止。

备注: 查看长答案的举例请参阅第 24 页。

状态行上的历史信息

有关输入项 / 答案对的信息，请使用历史指示符。例如:



作为默认设置，最后 30 个输入项 / 答案对会得到保存。当您输入新的输入项时，如历史区域已满(以 30/30 表示出)，新的输入项 / 答案对会得到保存，而最旧的则会被删除。历史指示符不能更改。

修改历史区域

欲:	请进行如下:
更改可保存的对数。	按下 [F1] 并选择 9:Format，或按下 **TI-89:** [◆] [|]；**TI-92 Plus:** [◆] F。然后按下 ⊕，利用 ⊕ 或 ⊖ 来将新数目突出显示，并按下 [ENTER] 两次。
清除历史区域并删除所有保存的对。	按下 [F1] 并选择 8:Clear Home，或在输入行中输入 **ClrHome**。
删除某一输入项 / 答案对。	将光标移到输入项或答案上，然后按下 [←] 或 [CLEAR]。

输入数字

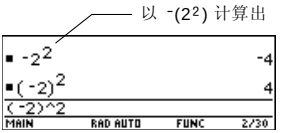
小键盘可让您为所进行的运算输入正数和负数。您也可以科学记数法输入数字。

输入负数

1. 按下负号键 [(-)]。(请勿使用减号键 [−]。)
2. 键入数字。

要记住，如 x^2 之类的函数是先于取负号运算的。

如您对负号运算有任何疑问，请使用 [(] 和 [)] 来括起来。



重要事项: 请使用 [−] 来进行减法运算和使用 [(-)] 来进行负数运算。

如您用了 [−] 而不是 [(-)] (或相反)，您将会得到一个错误信息和意外的的结果。例如:

- 9 [×] [(-)] 7 = ⁻63
  — 但 —
  9 [×] [−] 7 会显示一个错误信息。
- 6 [−] 2 = 4
  — 但 —
  6 [(-)] 2 = ⁻12 是由于可以蕴含乘法 6(⁻2) 演算。
- [(-)] 2 [+] 4 = 2
  — 但 —
  [−] 2 [+] 4 是从前一答案减去 2，然后加 4。

科学记数法输入数字

1. 在指数前键入数字部分。此数值可为表达式。
2. 按下:
   TI-89: [EE]
   TI-92 Plus: [2nd] [EE]
   E 会在显示器上出现。
3. 以最多 3 位的整数键入指数。您可使用负指数。

以科学记数法输入数字是不会导致答案以科学记数法或工程记数法显示的。

显示器格式是由模式设置(第 25 到 27 页)和数字的大小决定的。

输入表达式和指令

您可通过计算表达式来执行运算，也可通过适当的指令来起始一个行动。表达式和结果分别会按照第 25 页中述说的方式设置来计算和显示。

定义

备注: 附件 A 叙述了 TI-89 / TI-92 Plus 的所有内建函数和指令。

*备注: 本指南手册是使用文字 **command** 作为函数和指令的总称。*

表达式

由数字、变元、算子、函数以及它们计算成单一答案的变数构成。例如: $\pi r^2 + 3$。

- 以通常的书写方式输入相同顺序的表达式。
- 在很多情形下，您会被要求输入一个数字，但您可以输入表达式。

算子

执行一项运算，如 +、−、 ∗ 和 ^。

- 要求在算子的前后有变数的算子。例如: 4+5 和 5^2。

函数

对应一个数值。

- 在函数后要求有一或多个变数(包含在括号内)的函数。例如: √(5) 和 **min**(5,8)。

指令

开始行动。

- 指令不得用在表达式中。
- 某些指令不需要变数。例如: **ClrHome**.
- 某些指令则需要一或多个变数。
  例如: **Circle** 0,0,5。
  对于指令，不需将变数放在括号内。

蕴含乘法

TI-89 / TI-92 Plus 能辨认出蕴含乘法，故使用蕴含乘法不会与保留的记数法产生冲突。

	如您输入:	**TI-89 / TI-92 Plus 将其作为如下来演算:**
有效	2π	2∗ π
	4 sin(46)	4∗ sin(46)
	5(1+2) or (1+2)5	5∗ (1+2) 或 (1+2)∗ 5
	[1,2]a	[a 2a]
	2(a)	2∗ a
无效	xy	称为 xy 的单变元
	a(2)	函数调用
	a[1,2]	元素 a[1,2] 的矩阵目录

括号

表达式会按方程运算系统(EOS™)的等级得到计算。欲改变运算顺序或只保证按照您所要求的顺序来计算表达式，可使用括号来进行。

包含在括号对内的运算会首先进行。例如在 4(1+2) 中，EOS 首先计算(1+2)，然后再将答案乘 4。

输入表达式

键入表达式，然后按下 [ENTER] 来计算之。欲在输入行中输入一个函数或指令名称，您可:

- 如可用的话，按下它的按键。例如，按下 **TI-89:** [2nd] [SIN] 或 **TI-92 Plus:** [SIN]。
  — 或 —
- 如可用的话，在菜单上选择之。例如，在 MATH 菜单的子菜单 Number上选择 2:abs。
  — 或 —
- 从键盘上逐个字母地输入名称。(在 TI-89 上，利用 [alpha] 和 [2nd] [a-lock] 来键入字母。)您可使用任何大小写字母的混合。例如键入 **sin(** 或 **Sin(**。

例子

键入此例子中的函数名称。

计算 3.76 ÷ (⁻7.9 + √5) + 2 log 45。

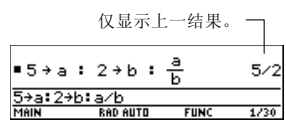
在 TI-89 上	在 TI-92 Plus 上	显示内容
3.76 [÷] [(] [(-)] 7.9 [+] [2nd] [√]	3.76 [÷] [(] [(-)] 7.9 [+] [2nd] [√]	3.76/(⁻7.9+√([2nd] [√] 是插入 √(，这是因为变数必须在括号内。
5 [)] [)]	5 [)] [)]	3.76/(⁻7.9+√(5)) 利用 [)] 一次来关闭 √(5)，再利用一次来关闭 (⁻7.9 + √5)。
[+] 2 [2nd] [a-lock] L O G [alpha] [(] 4 5 [)]	[+] 2 L O G [(] 4 5 [)]	3.76/(⁻7.9+√(5))+2log(45) **log** 要求有 ( ) 来围绕其变数。
[ENTER]	[ENTER]	$\blacksquare \frac{3.76}{-7.9+\sqrt{5}} + 2\cdot\log(45)$ 2.64258 3.76/(-7.9+√(5))+2log(45) MAIN RAD AUTO FUNC 1/30

备注: 您也可通过
TI-89: [CATALOG]
TI-92 Plus: [2nd] [CATALOG]
(第 40 页)来选择 **log**。

在一行上输入多个表达式

同时输入一个以上的表达式或指令，按下 [2nd] [:] 来用冒号将它们分隔开。

仅显示上一结果。

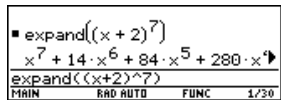


→ 当您按下 [STO▸] 来将数值存入变元时就会显示出。

如输入项或答案对于一行来讲太长

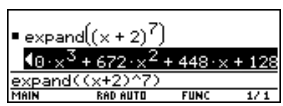
在历史区域，如输入项及其答案不能同在一行显示，则答案会在下一行显示出。

如输入项或答案过长而在一行中显示不出来，行尾就会显示出 ▶。



欲查看整个输入或答案:

1. 按下 ⊙ 来将光标从输入行上移至历史区域内。所突出显示的就是最后的答案。
2. 如有需要，可使用 ⊙ 和 ⊙ 来将您想查看的输入项或答案突出显示。例如，⊙ 可从答案移到输入项，直至历史区域。
3. 使用 ⊙ 和 ⊙ 或 [2nd] ⊙ 和 [2nd] ⊙ 来左右滚动。



备注: 当您向右滚动时，◀ 会显示在行的开端。

4. 欲返回输入行，可按下 [ESC]。

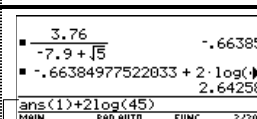
继续运算

当您按下 [ENTER] 来计算一个表达式时，TI-89 / TI-92 Plus 会将表达式留在输入行并将之突出显示。您可使用最后的答案或输入新的表达式来继续运算。

如您按下:	TI-89 / TI-92 Plus 就会:
[+]、[−]、[×]、[÷]、[^] 或 [STO▸]	用变元 ans(1) 取代输入行，该变元可让您将最后答案作为其它表达式的开始来使用。
任何其它按键	删去输入行并开始新的输入项。

例子

计算 3.76 ÷ (⁻7.9 + √5)，然后将 2 log 45 加到结果中。

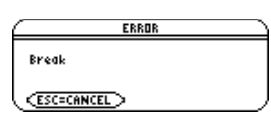
在 TI-89 上	在 TI-92 Plus 上	显示内容
3.76 [÷] [(] [(-)] 7.9 [+] [2nd] [√] 5 [)] [)] [ENTER] [+] 2 [2nd] [a-lock] L O G [alpha] [(] 4 5 [)] [ENTER]	3.76 [÷] [(] [(-)] 7.9 [+] [2nd] [√] 5 [)] [)] [ENTER] [+] 2 L O G [(] 4 5 [)] [ENTER]	 当您按下 [+] 时，输入行将会被表示最后答案的变元 ans(1) 所取代。

停止运算

当运算进行中时，BUSY 字符会状态行右边的结尾显示出来。欲停止运算，可按下 [ON]。

在显示出 "Break" 之前可能会有短暂的延迟。

按下 [ESC] 来返回当前的应用程序。



显示结果的格式

一个结果可以一系列的格式来进行运算和显示。本节是叙述影响到显示格式的 TI-TI-89 / TI-92 Plus92 的模式及其设置。欲检查或更改您当前的设置，请参阅第 36 页。

Pretty Print(工整印刷体)模式

默认设置为 Pretty Print = ON。指数、根、分数等均会以传统书写的相同形式显示。您可利用 [MODE] 来打开和关闭工整印刷体模式。

Pretty Print	
ON	**OFF**
$\pi^2, \frac{\pi}{2}, \sqrt{\frac{x-3}{2}}$	π^2, π/2, √((x− 3)/2)

输入行不能显示工整印刷体方式的表达式。如工整印刷体方式为开启状，在您按下 [ENTER] 后，历史区域内将显示出输入项及其结果。

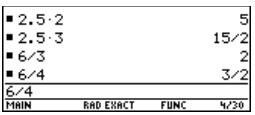
Exact(精确) / Approx(近似)模式

默认值为 Exact/Approx = AUTO。您可使用 [MODE] 来从三种设置中进行选择。

由于 AUTO 是其它两种设置的组合，所以，您应熟悉这三种设置。

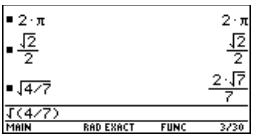
备注: 通过保留分数和符号方式，EXACT 可减少连锁运算中间结果导入的舍入错误。

EXACT(精确) — 任何不是完整数字的结果将会以分数或符号模式显示 (1/2、π、$\sqrt{2}$ 等)。



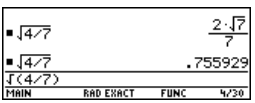
显示完整数字结果。

显示简化分数结果。



显示符号 π。

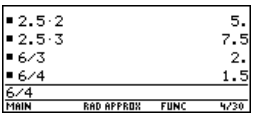
显示不能计算出完整数字的根的符号方式。



按下 [◆] [ENTER] 来临时改写 EXACT 设置并显示浮点结果。

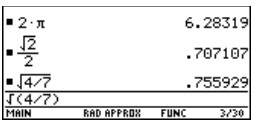
Exact(精确) / Approx(近似)模式 (续上)

APPROXIMATE(近似) — 所有数字结果在可能的时候会以浮点(小数)方式显示出来。



分数结果是以数字运算出。

备注: 结果是以 TI-89 / TI-92 Plus 的精确度舍入，并按当前的模式设置显示出来。

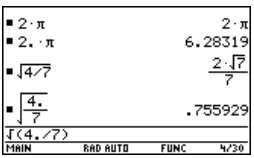


符号模式在可能的时候是以数字运算出。

由于不能计算未定义的变元，所以，它们将以代数的模式进行处理。例如，如变元 r 为未定义，则 $\pi r^2 = 3.14159 \cdot r^2$。

AUTO(自动) — 尽量使用 EXACT(精确)模式，但当您的输入项含有小数点时，则应使用 APPROXIMATE(近似)模式。同样，某些分数会显示 APPROXIMATE 结果，甚至您的输入项中不含小数点时。

提示: 欲保留 EXACT 形式，可使用分数而不用小数。例如，使用 3/2 而不用 1.5。



输入项中的一个小数强迫成为浮点结果。

以下表格列出了三种设置。

提示: 欲计算一个与当的前设置无关的近似输入项，可按下 ◆ ENTER。

输入项	Exact 结果	Approximate 结果	Auto 结果
8/4	2	2.	2
8/6	4/3	1.33333	4/3
8.5∗3	51/2	25.5	25.5
√(2)/2	$\frac{\sqrt{2}}{2}$	.707107	$\frac{\sqrt{2}}{2}$
π∗2	2·π	6.28319	2·π
π∗2.	2·π	6.28319	6.28319

输入项中的一个小数强迫成为自动的浮点结果。

显示数位模式

默认值是 Display Digits = FLOAT 6，这意味着结果会舍入到最多六个数位。您可使用 MODE 来选择不同的设置。设置会应用到所有的指数格式。

就内部而言，TI-89 / TI-92 Plus 对所有小数计算和保留高达 14 个有意义的数位(虽然最多显示 12 位)。

备注: 与显示数位设置无关，内部浮点运算会用全数值，以确保达到最高精确度。

备注: 如结果的量值不能按选定的位数显示，它就会自动以科学记数法显示出来。

设置	例子		说明
FIX (0 – 12)	123.	(FIX 0)	结果会舍到选定数目的小数位。
	123.5	(FIX 1)	
	123.46	(FIX 2)	
	123.457	(FIX 3)	
FLOAT	123.456789012		取决于结果，小数位的数目有所变动。
FLOAT (1 – 12)	1.E 2	(FLOAT 1)	结果会舍到选定数位的总数目。
	1.2E 2	(FLOAT 2)	
	123.	(FLOAT 3)	
	123.5	(FLOAT 4)	
	123.46	(FLOAT 5)	
	123.457	(FLOAT 6)	

指数格式模式

默认值是 Exponential Format = NORMAL。您可用 MODE 来从三种设置中进行选择。

备注: 在历史区域中，如绝对值小于 .001 的话，输入项中的数字就会以 SCIENTIFIC (科学)记数法模式显示出。

设置	例子	说明
NORMAL(普通)	12345.6	如结果不能按 Display Digits(显示数位)方式中指定的位数显示，TI-89 / TI-92 Plus 就会从 NORMAL 切换到 SCIENTIFIC(科学)，但仅对该结果而已。
SCIENTIFIC (科学)	1.23456E 4	1.23456×10^4 指数(10 的幂)。 小数点左方一直是 1 位数。
ENGINEERING (工程)	12.3456E 3	12.3456×10^3 指数是 3 的倍数。 小数点左方会是 1, 2 或 3 位数。

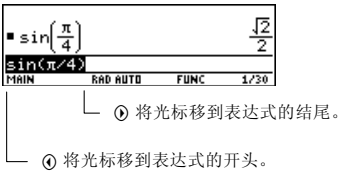
在输入行中编辑表达式

熟悉如何才能实际省时地编辑输入项。如您在键入表达式时出错，通常是改正错误较重新输入整个表达式为简单。

在前一输入项中消除突出显示

在您按下 [ENTER] 来计算表达式后，TI-89 / TI-92 Plus 会将该表达式留在输入行并将之突出显示。欲编辑表达式，您首先必须消除突出显示；要不然，您可通过在表达式上键入来清除意外的表达式。

欲清除突出显示，可将光标朝着您要编辑的表达式的边缘移动。



移动光标

消除突出显示后，将光标移到表达式内可应用的位置上。

备注: 如您没有按下 ◀ 或 ▶ 而意外地按下了 ⊖，光标会上移到历史区域内。此时，可按下 [ESC] 或按下 ⊖，直至光标返回输入行中。

欲将光标:	可按下:	
在表达式内左右移动。	◀ 或 ▶	按住光标控制盘来重复移动。
移到表达式的开头。	[2nd] ◀	
移到表达式的结尾。	[2nd] ▶	

删除字符

欲删除:	可按下:	
光标左方的字符。	[←]	按住 [←] 来删除多个字符。
光标右方的字符。	[◆] [←]	
光标右方的所有字符。	[CLEAR] (仅用一次)	如光标右方没有字符，则 [CLEAR] 可清除整个输入行。

清空输入行

欲清空输入行，可按下:

- [CLEAR] (如光标位于输入行的开头或结尾)；
  — 或 —
- [CLEAR] [CLEAR] (如光标不是位于输入行的开头或结尾)。第一次按下是删除光标右方的所有字符，第二次按下是清除输入行。

插入或改写字符

TI-89 / TI-92 Plus 具有插入和改写两种方式。TI-89 / TI-92 Plus 的默认值是插入方式。欲在插入和改写方式之间切换，可按下 [2nd] [INS]。

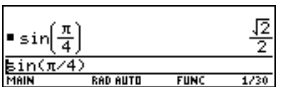
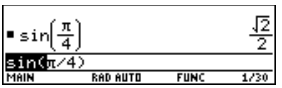
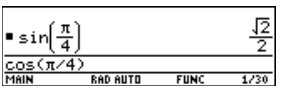
提示: 查看光标来了解您正处于插入或是改写方式。

如 **TI-89 / TI-92 Plus** 正处于:	您键入的下一字符:
Insert mode └ 字符间小光标	将在光标处插入。
Overtype mode └ 用光标将字符突出显示	将替换突出显示的字符。

替换或删除多个字符

首先将可用的字符突出显示，然后替换或删除所有突出显示的字符。

提示: 当您将字符突出显示来替换时，请记住，某些功能键会自动加入一个开口的左括号。

欲:	请这样进行:
将字符突出显示	1. 将光标移到您要突出显示的字符的任意一边。  欲利用 **cos(** 替换 **sin(**，可将光标放在 **sin** 旁。 2. 按住 [↑] 并按下 ◀ 后 ▶ 来使光标左右方的字符突出显示。  按住 [↑] 并按下 ▶ ▶ ▶ ▶。
替换突出显示的字符	键入新的字符。
— 或 —	
删除突出显示的字符	按下 [←]。

为保持键盘整洁，TI-89 / TI-92 Plus 是用菜单来存取众多的操作。本节中给出了如何在菜单上选择一个项目的综述。

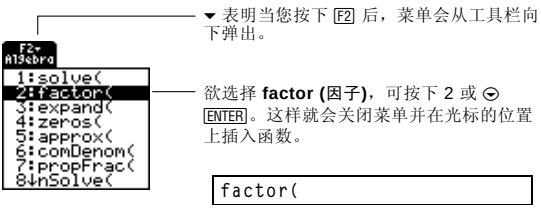
显示菜单

按下:	以显示:
F1、F2 等	工具栏菜单 — 在大多数应用程序屏幕的顶部向下弹出工具栏。让您选择通常用于该应用程序的操作。
APPS	APPLICATIONS (应用程序)菜单 — 让您从应用程序列表中进行选择。请参阅第 34 页。
2nd [CHAR]	CHAR (字符)菜单 — 让您从特殊字符(希腊字母、数学符号等)的类别中进行选择。
2nd [MATH]	MATH (数学运算)菜单 — 让您从数学运算操作的类别中进行选择。
TI-89: CATALOG **TI-92 Plus:** 2nd [CATALOG]	CATALOG (目录)菜单 — 让您从完整的、按字母排序的 TI-89 / TI-92 Plus 内建指令和函数表中进行选择。也让您选择用户定义功能和 Flash 应用程序功能(如已定义或已加载的话)。
2nd [CUSTOM]	CUSTOM (自定义)菜单 — 让您存取一个可自定义列出任何可用函数、指令或字符的菜单。TI-89 / TI-92 Plus 含有一个默认的自定义菜单，在此，您可加以修改或重新定义之。请参阅第 33 页。

从菜单选择一个项目

欲在显示出的菜单上选择一个项，可:

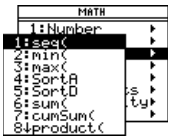
- 按下出现在该项左方的数字或字母；对于字母，在 TI-89 上可按下 alpha 并按下字母键。
  — 或 —
- 使用光标控制盘 ⊖ 和 ⊕ 来将项目突出显示，然后按下 ENTER。(注意，从第一个项开始，按下 ⊕ 是不能将突出显示移到最后的一个项的，反之亦然。)

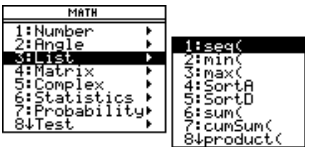


结尾带有 ▶ (子菜单)的项

如您选择以 ▶ 结尾的一个菜单项，此时就会显示出一个子菜单，您可从此子菜单上选择一个项目。

备注: 由于屏幕的大小所限，TI-89 会将这些菜单如下般重叠:





例如，**List** 会显示出能让您选择一个特定函数列表的子菜单。

↓ 表明您可使用光标控制盘来向下滚动，以获得其它的项目。

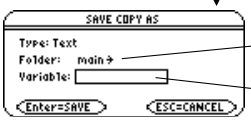
对于带有子菜单的项目，您可如以下般使用光标控制盘。

- 欲为突出显示的项目显示子菜单，可按下 ⊙。(此操作与该项目的选择相同。)
- 欲删除子菜单而不进行任何选择，可按下 ⊙。(此操作与按下 ESC 相同。)
- 欲从第一个菜单项直接换行到最后的菜单项，可按下 ⊙。欲从最后的菜单项直接换行到第一个菜单项，可按下 ⊙。

含有 "..." (对话框)的项目

如您选择一个带有"..." (省略号标记)的菜单项，会显示出一个对话框，以便您输入其它的信息。

例如，**Save Copy As ...** 会显示一个提示您选择文件夹名称和键入变元名称的对话框。



→ 表明您可按下 ⊙ 来从菜单上显示和选择。

输入框表明您必须键入一个数值。(对于 TI-89，Alpha-lock 功能会自动打开。请参阅第 18 页。)

在输入框内键入了变元后，您必须按下 ENTER 两次来保存信息并关闭对话框。

取消菜单

欲取消当前的菜单并不作出选择，可按下 ESC。取决于是否显示出子菜单，您也许必须按下 ESC 数次来取消显示的菜单。

从一个工具栏菜单移到另一个上

欲从一个工具栏菜单移到另一个上而不作出选择，您可:

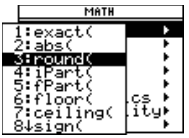
- 按下按键([F1]、[F2] 等) 来转到其它工具栏菜单上；
  — 或 —
- 利用光标控制盘来移到下一个(按下 ⊙)或前一个(按下 ⊙) 工具栏菜单。按下 ⊙ 从最后的菜单移到第一个菜单，反之亦然。

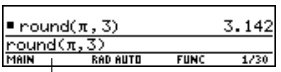
当使用 ⊙ 时，应确保子菜单中的项目不是突出显示。如果是的话，⊙ 就会显示该项目的子菜单而不是移到下一工具栏菜单。

例子: 选择菜单项

将 π 值舍成三位小数。从清空主屏幕上的输入行开始:

1. 按下 [2nd][MATH] 来显示 MATH 菜单。
2. 按下 1 来显示 Number 子菜单。(或按下 [ENTER]，由于第一项已自动地突出显示了。)
3. 按下 3 来选择 **round**。(或按下 ⊙⊙ 和 [ENTER]。)
4. 按下 [2nd][π][,]3[)]，然后再按下 [ENTER] 来计算表达式。





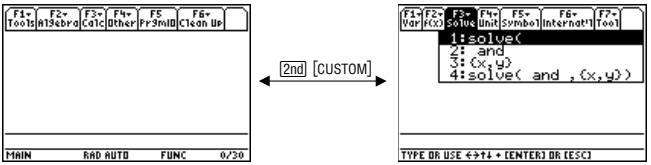
选择第 3 步的功能会自动在输入行中键入 **round(**。

TI-89 / TI-92 Plus 备有一个自定义菜单，您可随时把它打开或关闭。您可使用默认的自定义菜单或创建自有的菜单。

打开和关闭自定义 菜单

当您打开自定义菜单时，它就会取代普通的工具栏菜单。当您关闭它时，普通菜单就会重新出现。例如，在主屏幕的普通菜单上，按下 [2nd] [CUSTOM] 来切换自定义菜单的开和关。

备注: 您也可选择输入行上的 **CustmOn** *或* **CustmOff** *并按下* [ENTER] *来打开和关闭自定义菜单。*



主屏幕普通工具栏菜单　　自定义菜单

除非菜单被修改，否则会一直显示默认的自定义菜单。

提示: 自定义菜单可让您快速地存取最常用的项。

菜单	功能
[F1] Var	常用的变元名称。
[F2] f(x)	函数名称，如 f(x)、g(x) 和 f(x,y)。
[F3] Solve	与解方程相关的项。
[F4] Unit	常用的计量单位，如 _m、_ft 和 _l。
[F5] Symbol	符号，如 #、? 和 ~。
Internat'l **TI-89:** [2nd] [F6] **TI-92 Plus:** [F6]	常用的重音字符，如 è、é 和 ê。
Tool **TI-89:** [2nd] [F7] **TI-92 Plus:** [F7]	**ClrHome**、**NewProb** 和 **CustmOff**。

恢复默认自定义菜单

如显示出非默认的自定义菜单，而您又想恢复默认菜单的话:

1. 在主屏幕上，利用 [2nd] [CUSTOM] 来关闭自定义菜单和显示主屏幕普通工具栏菜单。

备注: 前一自定义菜单会被清除。如该菜单是由程序创建的，稍后您可再次运行程序来重新创建。

2. 显示 Clean Up 工具栏菜单并选择 3:Restore custom default。
   TI-89: [2nd] [F6]
   TI-92 Plus: [F6]
   此操作是将用来创建默认菜单的命令粘贴到输入行。

3. 按下 [ENTER] 来执行命令和恢复默认的。

TI-89 / TI-92 Plus 备有不同的应用程序来让您解决与探索各种问题。您可从菜单中选择某一应用程序或可直接从键盘上存取较为常用的一些应用程序。

从 APPLICATIONS(应用程序)菜单上选择

备注: 欲取消菜单并不作出选择，可按下 [ESC]。

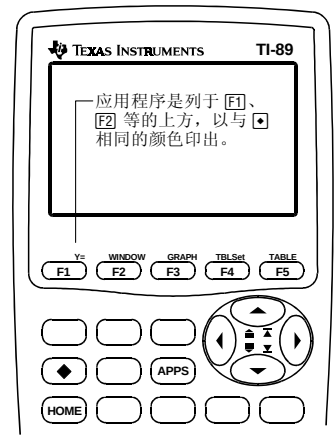
1. 按下 [APPS] 来显示列有应用程序的菜单。
2. 选择一个应用程序，可:
   - 使用光标控制盘 ⊖ 或 ⊕ 来将应用程序突出显示，然后按下 [ENTER]。
   — 或 —
   - 按下该程序左方的数字或字母。

应用程序:	可让您:
FlashApps	显示 Flash 应用程序的列表，如有的话。
Y= Editor	定义、编辑和选择图函数或方程。
Window Editor	设定查看图形的窗口大小。
Graph	显示图形。
Table	显示与输入的函数相对应的变元值表。
Data/Matrix Editor	输入和编辑数组、数据和矩阵。您可执行统计运算和作统计图。
Program Editor	输入和编辑程序和功能。
Text Editor	输入和编辑文本段。
Numeric Solver	输入表达式或方程、定义所有数值(只限一个变元)，然后求未知的变元。
Home	输入表达式和指令，执行运算。

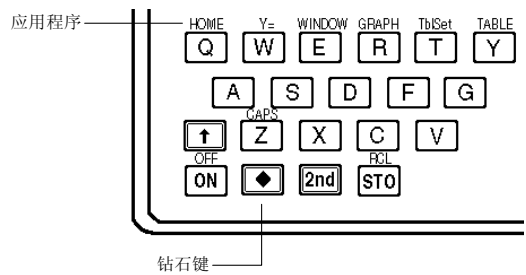
从键盘上选择

您可从键盘上存取较为常用的应用程序。例如，在 TI-89 上，◆ [Y=] 与按下 ◆ 并按下 F1 相同。本指南手册是使用表示法 ◆ [Y=]，与第二功能所表示的基数类似。

应用程序:	按下:
主屏幕	**TI-89:** HOME **TI-92 Plus:** ◆ [HOME]
Y=编辑器	◆ [Y=]
窗口编辑器	◆ [WINDOW]
图形	◆ [GRAPH]
表格设置	◆ [TblSet]
表格屏	◆ [TABLE]



在 TI-92 Plus 上，应用程序是列在 QWERTY 键上方。



设置模式

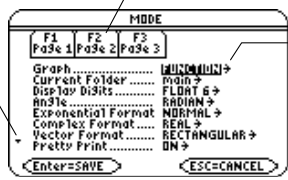
模式是控制数字和图形是如何得到显示和解释的。当 TI-89 / TI-92 Plus 关闭后，模式设置是由 Constant Memory™ 功能进行保留。所有数字，包括矩阵和数组元素均会按照当前的模式设置显示出。

检查模式设置

按下 MODE 来显示 MODE (模式)对话框，该对话框中列出了模式及其当前的设置。

共有三页模式设置列表。按下 F1、F2 或 F3 来快速显示某一指定页。

表明您可下滚来查看其它模式。



→ 表明您可按下 ▷ 或 ◁ 来从菜单上显示和选择。

备注: 不是当前有效的模式会暗淡显示。例如，在第 2 页上，当 Split Screen = FULL时，Split 2 App为无效。当您滚过列表时，光标会跳到暗淡显示的设置上。

更改模式设置

从 MODE 对话框上:

1. 将您要更改的设置突出显示，使用 ⊖ 或 ⊕ (连同 F1、F2 或 F3) 来滚动列表。
2. 按下 ▷ 或 ◁ 来显示列出有效设置的菜单。当前的设置为突出显示。
3. 选择可应用的设置，可:
   - 使用 ⊖ 或 ⊕ 来将设置突出显示，然后按下 ENTER;
     — 或 —
   - 按下该设置的数字或字母。
4. 如有需要，更改其它的模式设置。
5. 当您完成选择时，按下 ENTER 来保存变更并退出对话框。

提示: 欲取消菜单并返回 MODE 对话框而不作出选择，可按下 ESC

重要事项: 如您按下 ESC 而不是 ENTER 来退出 MODE 对话框，那么您对模式所作的任何更改将会被取消。

模式综述

备注: *关于特殊模式更详细的信息，请参阅本指南手册可应用的部分。*

模式	说明
Graph(作图)	作图的类型: FUNCTION、PARAMETRIC、POLAR、SEQUENCE、3D 或 DE。
Current Folder (当前文件夹)	用于储存和重新调用变元的文件夹。除非您创建了附加的文件夹，否则，只有 MAIN (主)文件夹为可用。
Display Digits (显示数位)	在浮点结果中显示的数位最大数目(FLOAT)或小数位的固定数目(FIX)，与设置无关。浮点结果中显示的数位不能超过 12 位。请参阅第 27 页。
Angle(角)	计算和显示角度值的单位: RADIAN 或 DEGREE。
Exponential Format (指数格式)	用于显示结果的记数法: NORMAL、SCIENTIFIC 或 ENGINEERING。请参阅第 27 页。
Complex Format (复数格式)	用于显示复数结果的格式，如有的话: REAL (复数结果不显示出，除非您是使用复数输入项)、RECTANGULAR 或 POLAR。
Vector Format (向量格式)	用于显示 2- 和 3-元素向量的格式: RECTANGULAR(矩形)、 CYLINDRICAL(柱形)或 SPHERICAL(球形)。
Pretty Print(工整印刷体)	将工整印刷体显示功能 OFF 或 ON。请参阅第25 页。
Split Screen (拆屏)	将屏幕拆分成两部分，并指定如何重排: FULL (不拆分屏幕)、 TOP-BOTTOM 或 LEFT-RIGHT。
Split 1 App	应用程序于拆分屏幕的顶部或左方。如您不在使用拆分屏幕，这就是当前的应用程序。
Split 2 App	应用程序于拆分屏幕的底部或右方。这仅对拆分屏幕为启用。
Number of Graphs	对于拆分屏幕，可让您将屏幕的两边设定成显示独立图形集。
Graph 2	如 Number of Graphs = 2，在屏幕的 Split 2 部分选择图形的类型。
Split Screen Ratio (拆屏比例)	拆分屏幕的两部分的比例大小: 1:1、1:2 或 2:1。(仅对 TI-92 Plus)
Exact/Approx (精确 / 近似)	以数字方式或有理 / 符号方式来计算表达式和显示结果: AUTO、EXACT 或 APPROXIMATE。请参阅第 25 页。

模式(续上)

模式	说明
Base(基)	让您以小数(DEC)、十六进制(HEX)或二进制(BIN)形式输入数字来执行运算。
Unit System (计量单位系统)	让您在表达式中为数值输入单位，如 6_m * 4_m 或 23_m/_s * 10_s，将数值在同一类别中从一种单位换算到另外一种，并创建您自有的自定单位。
Custom Units (自定计量单位)	让您选择自定的默认值。模式为暗淡显示，除非您选择 Unit System, 3:CUSTOM。
Language (语言)	让您将 TI-89 / TI-92 Plus 本土化成数种语言之一，这点取决于安装了哪种语言的 Flash 应用程序。

利用清理菜单来开始一个新习题

Clean Up(清理)菜单是在主屏幕上，Clean Up 工具栏菜单可让您在一个整洁的状态下开始新的运算而无须重新设定 TI-89 / TI-92 Plus 的内存。

清理工具栏菜单

从主屏幕上，通过按下以下按键来显示 Clean Up 菜单:

TI-89: [2nd] [F6]

TI-92 Plus: [F6]

菜单项	说明
Clear a–z	在当前的文件夹中清除(删除)所有单字符变元名称，除非变元为已锁定或已存档的。您会得到按下 [ENTER] 来确认动作的提示。 单字符变元名称通常是在以下的符号运算中用到: solve(a·x²+b·x+c=0,x) 如变元的任何一个已经赋了值，那么您的运算有可能得到一个误导的结果。为防止这情形产生，您可在运算开始前先选择 1:Clear a–z。
NewProb	将 **NewProb** 放在输入行中。您必须随后按下 [ENTER] 来执行命令。 **NewProb** 可执行范围广大的运算，让您在一个整洁的状态下开始新的习题而无须重新设定内存: • 清除当前文件夹中所有单字符的变元名称(与 1:Clear a–z 相同)，除非变元为锁定或已存档的。 • 关闭当前作图模式中的所有功能和统计作图(**FnOff** 和 **PlotsOff**)。 • 执行 **ClrDraw**、**ClrErr**、**ClrGraph**、**ClrHome**、**ClrIO** 和 **ClrTable**。
Restore custom default	如默认以外的一个自定义菜单正在运作，此功能可让您将其储存成默认值。请参阅第 33 页。

提示: 当您定义一个要保留的变元时，请在名称中使用一个以上的字符。这样可避免疏忽地被 1:Clear a–z *删除掉。*

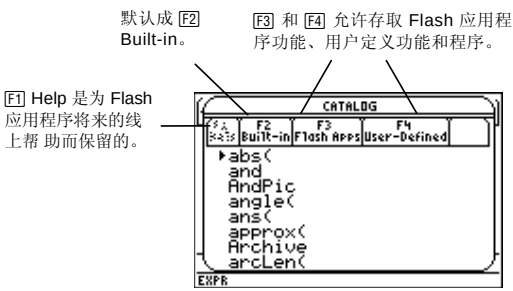
CATALOG(目录)提供了从一个便利的列表上存取任何内建于 TI-89 / TI-92 Plus 命令(函数和指令)的途径。此外，CATALOG 对话框更可让您选择用于 Flash 应用程序的功能或用户定义的函数(如已经加载或指定的话)。

CATALOG(目录)的显示

欲显示 CATALOG 对话框，可按下:

TI-89: [CATALOG]
TI-92 Plus: [2nd] [CATALOG]

CATALOG 默认为 [F2] Built-in，是显示所有预安装的 TI-89 / TI-92 Plus 命令(函数和指令)的字母列表。



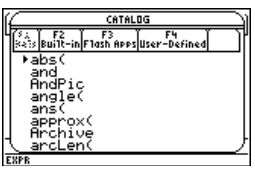
备注: 当前无效的选项为暗淡显示，例如，[F1] Help 是为 Flash 应用程序将来的线上帮助而保留的。如您没有安装 Flash 应用程序，[F3] Flash Apps 则为暗淡显示。如您没有创建的函数和程序，[F4] User-Defined 也为暗淡显示。

从 CATALOG 上选择一个内建命令

当您选择一个命令时，它的名称就会插入输入行光标所在的位置上。因此，您要在选择命令前先将光标按需定位。

1. 按下:
   TI-89: [CATALOG]
   TI-92 Plus: [2nd] [CATALOG]
2. 按下 [F2] Built-in。

备注: 当您第一次显示 Built-in 列表时，它会从顶部开始。下一次显示列表时，会在您离开它的相同位置显示出来。



- 命令是以字母顺利列出。对于非字母开头的命令，如(+、%、√、Σ、等)会显示在列表的尾部。
- 欲编辑 CATALOG 并不作出命令选择，可按下 [ESC]。

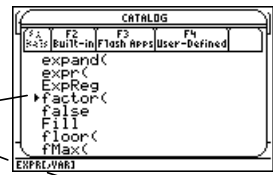
3. 将指示符 ▶ 移到命令上，然后按下 [ENTER]。

提示: 从列表的顶部按下 ⊖ 来移到底部。从列表的底部按下 ⊕ 来移到顶部。

欲按下列条件移动 ▶ 指示符:	可按下或键入:
一次一个命令	⊕ 或 ⊖
一次一页	[2nd] ⊕ 或 [2nd] ⊖
到带有指定字母开头的第一个命令	字母键。(在 TI-89 上，不要先按下 [alpha]。如您按下了，则必须在键入字母前再次按下 [alpha] 或 [2nd] [a-lock]。)

关于参数的信息

对于以 ▶ 指示的命令，状态行会显示出要求的和任选的参数，如有的且已键入的话。



指示出的命令及其参数。

中括号 [ ] 指出任选的参数。

备注: 有关参数更详细的信息，请参阅附件 A 中该命令的说明。

从上面的例子中可以看到 **factor** 的句法为:

factor(*表达式*) 要求的

— 或 —

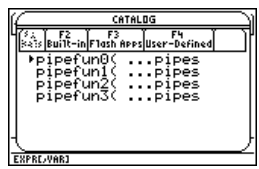
factor(*表达式, 变元*) 任选的

选择 Flash 应用程序函数

Flash 应用程序可含有一个或多个函数。当您选择一个函数时，它的名称就会插入输入行光标所在的位置上。因此，您要在选择函数前先将光标按需定位。

1. 按下:
   TI-89: [CATALOG]
   TI-92 Plus: [2nd] [CATALOG]

2. 按下 [F3] Flash Apps。(如 TI-89 / TI-92 Plus 没有已安装的 Flash 应用程序，此选项为暗淡显示。)



- 列表是按函数名字的字母排序。左边的列是列出函数，右边的列是列出含有函数的 Flash 应用程序。
- 与函数相关的信息会在状态行中显示出来。
- 欲退出而不作任何函数选择，可按下 [ESC]。

3. 将指示符 ▶ 移到函数上，然后按下 [ENTER]。

欲按下列条件移动 ▶ 指示符：	可按下或键入:
一次一个函数	⊖ 或 ⊕
一次一页	[2nd] ⊖ 或 [2nd] ⊕
到带有指定字母开头的第一个函数	字母键。(在 TI-89 上，**不要**先按下 [alpha]。如您按下了，则必须在键入字母前再次按下 [alpha] 或 [2nd] [a-lock]。)

选择用户定义函数或程序

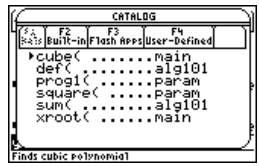
您可创建自己的函数或程序，然后利用 [F4] User-Defined 来存取它们。

当您选择一个函数或程序时，它的名称就会插入输入行光标所在的位置上。因此，您要在选择函数或程序前先将光标按需定位。

1. 按下:
   TI-89: [CATALOG]
   TI-92 Plus: [2nd] [CATALOG]

2. 按下 [F4] User-Defined。(如您还没有定义函数或创建程序，此选项会暗淡显示。)

备注: 可利用 VAR-LINK 屏幕来管理变元、文件夹和 Flash 应用程序。



- 列表是按函数 / 程序名字的字母排序。左边的列是列出函数和程序，右边的列是列出含有函数和程序的文件夹。
- 如函数或程序的第一行是注释，那么其内容就会显示于状态行中。
- 欲编辑并不作出函数或程序选择，可按下 [ESC]。

3. 将指示符 ▶ 移到函数或程序上，然后按下 [ENTER]。

欲按下列条件移动指示符 ▶:	可按下或键入:
一次一个函数或程序	⊖ 或 ⊕
一次一页	[2nd] ⊖ 或 [2nd] ⊕
到带有指定字母开头的第一个函数或程序	字母键。(在 TI-89 上，**不要**先按下 [alpha]。如您按下了，则必须在键入字母前再次按下 [alpha] 或 [2nd] [a-lock]。)

储存和重新调用变元值

当您储存一个数值时，是按已命名变元来储存的。在表达式中，您可用变元名取代其所表示的值。当 TI-89 / TI-92 Plus 在表达式中遇到名称时，会替换为变元的储存值。

变元名称的规则

一个变元名称:

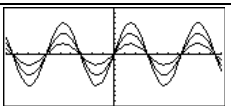
- 可使用包含有字母和数字的 1 到 8 个字符。这包括希腊字母 (π 除外)、重音字母及国际字母。不包括空格。
  - 第一个字符不可为数字。
- 可使用大小写字母。名称 AB22、 Ab22、 aB22 和 ab22 均相对于同一变元。
- 不得与 TI-89 / TI-92 Plus 预指定的名称相同。预指定包括:
  - 内建函数(如 **abs**)和指令(如 **LineVert**)。请参阅附件 A。
  - 系统变元(如 xmin 和 xmax，该等变元是用于储存图形相关数值的)。

例子

变元	说明
myvar	可以
a	可以
Log	不可以，该名称已预指定给 **log** 函数。
Log1	可以
3rdTotal	不可以，开头是一个数字。
circumfer	不可以，超过 8 个字符。

数据类型

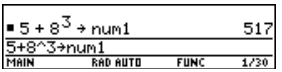
您可将任何的 TI-89 / TI-92 Plus 数据类型作为变元保存。数据类型的列表，请参阅附件 A 的 **getType()** 。下面备有一些例子:

数据类型	例子
表达式	2.54, 1.25E6, 2π, xmin/10, 2+3*i*, $(x-2)^2$, $\sqrt{2}/2$
数组	{2 4 6 8}, {1 1 2}
矩阵	[1 0 0], $\begin{bmatrix} 1 & 0 & 0 \\ 3 & 4 & 6 \end{bmatrix}$
字符串	“Hello”, “The answer is:”, “xmin/10”
图画	
函数	myfunc(arg), ellipse(x,y,r1,r2)

将数值存入变元

1. 输入您要储存的变元，可为一个表达式。
2. 按下 [STO▸] 。储存标记 (➔) 就会显示出。
3. 键入变元名称。
4. 按下 [ENTER]。

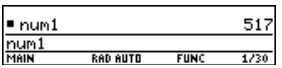
备注: 当键入变元名称时，TI-89 用户需要要使用 [alpha] 键。



欲临时储存一个变元，您可使用 “with” 算子。

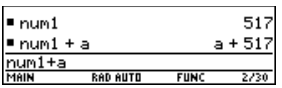
显示变元

1. 键入变元名称。
2. 按下 [ENTER] 。



如变元未定义，则变元名称会在结果中显示出。

在此例子中，变元 a 为未定义。因此，它是作为符号变元来使用。

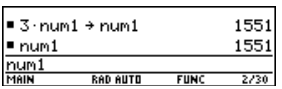


在表达式中使用变元

1. 将变元名称键入到表达式中。
2. 按下 [ENTER] 来计算表达式。

提示: 欲查看现有的变元名称表，可使用 [2nd] [VAR-LINK]。

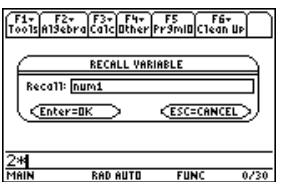
如您要用结果替换变元的前一数值，那么您就要储存结果。



重新调用变元的数值

某些情况下，您或许要用表达式中变元的实际数值来替代变元名称。

1. 按下 [2nd] [RCL] 来显示对话框。
2. 键入变元名称。
3. 按下 [ENTER] 两次。



在此例子中，储存在 num1 的数值将会插入在输入行的光标位置上。

重新利用前一输入项或上一答案

您可通过重新“照原样”执行输入项来重新利用前一输入项，或通过编辑输入项并重新执行之。您也可将上一计算答案插入在新表达式中来重新使用之。

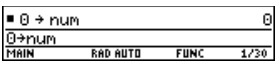
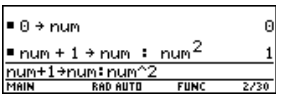
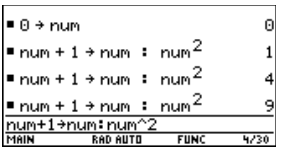
在输入行中重新使用表达式

当您按下 [ENTER] 来计算一条表达式时，TI-89 / TI-92 Plus 会将该表达式留在输入行并将其突出显示。如有需要，您可在输入项上键入内容或重新利用之。

例如，利用某变元来求 1、2、3 等的平方。

如下所述地设定起始变元值，然后输入变元表达式。随后，重新进入来增加变元并计算平方值。

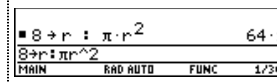
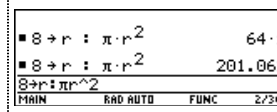
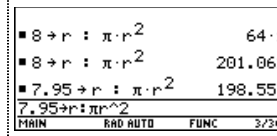
提示：“照原样”重新执行输入项通常是用于与变元相关的交互计算的。

在 **TI-89** 上	在 **TI-92 Plus** 上	显示内容
0 [STO▸] [2nd] [a-lock] N U M [ENTER]	0 [STO▸] N U M [ENTER]	
N U M [alpha] [+] 1 [STO▸] [2nd] [a-lock] N U M [2nd] [:] N U M [^] 2 [ENTER]	N U M [+] 1 [STO▸] N U M [2nd] [:] N U M [^] 2 [ENTER]	
[ENTER] [ENTER]	[ENTER] [ENTER]	

提示: 编辑输入项可让您实现最小的变更且不需重新键入整个输入项。

利用方程 A=πr²，使用试错法来求覆盖 200 平方厘米的圆的半径。

下面的例子将 8 作为第一个猜测值，然后以近似浮点形式显示答案。您可利用 7.95 来编辑并继续，直至答案精确到您想要的为止。

在 **TI-89** 上	在 **TI-92 Plus** 上	显示内容
8 [STO▸] [alpha] R [2nd] [:] [2nd] [π] [alpha] R [^] 2 [ENTER]	8 [STO▸] R [2nd] [:] [2nd] [π] R [^] 2 [ENTER]	
[◆] [ENTER]	[◆] [ENTER]	
[◀] [◆] [DEL] 7.95 [ENTER]	[◀] [◆] [DEL] 7.95 [ENTER]	

备注: 当输入项含有小数点时，结果就会自动以浮点形式显示出来。

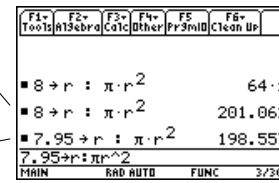
重新调用前一输入项

虽然输入项已滚出屏幕的顶部，您可重新调用任何储存在历史区域内的前一输入项。被调用的输入项会*覆盖*当前在输入行中所显示的。您可重新执行或编辑调用的输入项。

欲调用:	可按下:	结果:
上一输入项 (如您已经改变了输入行的话)	[2nd] [ENTRY] 一次	如上一输入项仍然显示在输入行中，这样可调用该输入项之前的一个。
前一输入项	[2nd] [ENTRY] 反复	每按一次按键，可调用显示在输入行中的输入项之前的一个输入项。

*备注: 您也可利用 **entry** 功能来调用前一输入项。详情请参阅附件 A 中的 **entry()**。*

例如:



如输入行中含有上一输入项，那么 [2nd] [ENTRY] 可调用该输入项。

如输入行被编辑过或清除过，那么 [2nd] [ENTRY] 可调用该输入项。

调用上一答案

每次您计算一条表达式时，TI-89 / TI-92 Plus 会将答案存如变元 ans(1) 中。欲将此变元插在输入行中，可按下 [2nd] [ANS]。

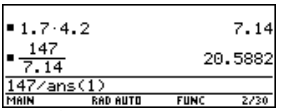
例如，先计算一个 1.7 米乘 4.2 米的花园地皮的面积。并由这块地产出总数为 147 个番茄，计算每平方米的产量。

1. 求面积。

   1.7 [×] 4.2 [ENTER]

2. 求单位面积产量。

   147 [÷] [2nd] [ANS] [ENTER]



变元 ans(1) 已插入，其数值也用在运算中。

备注: 请参阅附件 A 的 ***ans()****。*

就如 ans(1) 一直含有上一答案一样，ans(2)、ans(3) 等也含有前一答案。例如，ans(2) 含有上上一答案。

从历史区域自动粘贴输入项或答案

您可从历史区域中选择任何输入项或答案，并将它的一个复制件“自动粘贴”在输入行中。此功能可让您将前一输入项或答案插入输入行中而无须重新键入以前的信息。

为什么使用自动粘贴

使用自动粘贴的效果与先前章节中叙述的 2nd [ENTRY] 和 2nd [ANS] 相类似，但不尽相同。

对于输入项:

粘贴可让您:	2nd [ENTRY] 可让您:
在输入行中*插入*任何以前的输入项。	用任何以前的输入项*替代*输入行的内容。

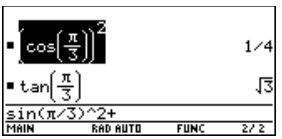
对于答案:

粘贴可让您:	2nd [ANS] 可让您:
将显示出的*任何以前答案*的值插入在输入行中。	插入仅*含有上一答案*的变元 ans(1)。每次您输入运算时，ans(1) 就会更新成最后答案。

备注: 您也可利用 F1 工具栏菜单粘贴信息。

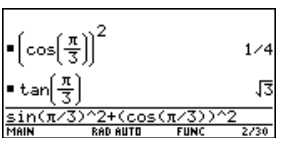
自动粘贴输入项或答案

1. 在输入行上，将光标置于您要插入输入项或答案的位置处。
2. 按下 ⊕ 来将光标上移入历史区域，这样可将上一答案突出显示。
3. 使用 ⊕ 和 ⊖ 来将要自动粘贴的输入项或答案突出显示。
   - ⊕ 可从答案上移到输入项，直至到达历史区域。
   - 您可使用 ⊕ 来将滚出屏幕外的项目突出显示。



4. 按下 ENTER。

   突出显示的项目就会插入到输入行上。



这样可把整个输入项或答案粘贴。如您只需要输入项或答案的部分，可编辑输入行来删除不想要的部分。

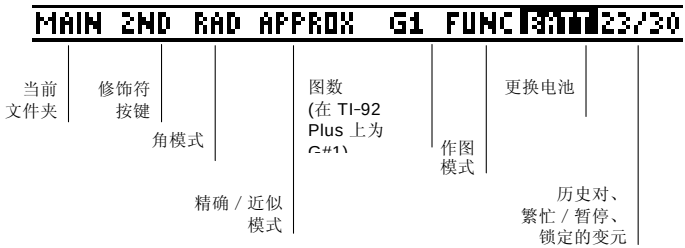
提示: 欲取消自动粘贴功能并返回输入行，可按下 ESC。

提示: 欲查看超出行长(由行结尾的 ► 标记指示出)的输入项或答案，可使用 ⊖ 和 ⊕ 或 2nd ⊖ 和 2nd ⊕。

显示器状态行中的指示符

状态行显示在所有应用程序屏的底部，它的作用是显示有关 TI-89 / TI-92 Plus 的当前状态，包括一系列重要的模式设置。

状态行指示符



指示符	意义
当前文件夹	显示当前文件夹的名称。MAIN 是默认文件夹，当您使用 TI-89 / TI-92 Plus 时就会自动设定。
修饰符按键	显示哪个修饰符正在起作用，如下面所述。
2nd	[2nd] — 将使用您下一按键的第二功能。
◆	[◆] — 将使用您下一按键的钻石功能。
(TI-89)	[alpha] — 将在您下一按键输入小写字母。
(TI-89)	[2nd] [a-lock] — 小写 alpha-lock 功能打开。除非您关闭它，否则您的每一按键是输入小写字母。欲取消 alpha-lock，可按下 [alpha]。
(TI-89)	[↑] [alpha] — 大写 ALPHA-lock 功能打开。除非您关闭它，否则您的每一按键是输入大写字母。欲取消 ALPHA-lock，可按下 [alpha]。
⬆	[↑] — 将在您下一按键输入大写字母。在 TI-89 上，您可使用 [↑] 来键入字母而不需通过 [alpha] 键。
角模式	显示演算和显示角数值的计量单位。欲改变 Angle 模式，可使用 [MODE] 键进行。
RAD	弧度
DEG	度

备注: 欲取消 [2nd]、[◆]、[alpha] 或 [↑]，可再次按下同一按键或按下另外一个修饰符键。

备注: 如您下次按下的键没有钻石功能或没有相关的字母，则该键只执行普通的操作。

状态行(续上)

指示符	意义
精确 / 近似模式	显示答案以何种形式被计算和显示的。请参阅第 25 页。欲改变 Exact/Approx 模式，可使用 MODE 键进行。
AUTO	自动
EXACT	精确
APPROX	近似
图数	如屏幕是拆分成显示两个独立的图形，这是表示哪个图形是活动的 — GR1 或 GR2 (TI-92 Plus 上的显示为 G#1 或 G#2。)
作图模式	表示可画出的图形的模式。欲改变 Graph 模式，可使用 MODE 键进行。
FUNC	y(x) 函数
PAR	x(t) 和 y(t) 参数方程
POL	r(θ) 极坐标方程
SEQ	u(n) 序列
3D	z(x,y) 3D 方程
DE	y'(t) 微分方程
电池	只在电池低电量时显示。如显示出 BATT 字样并屏幕变成黑屏，请尽快更换电池。
历史对、繁忙 / 暂停、存档	显示在状态行此部分中的信息取决于您所使用的应用程序。
23/30	主屏幕上显示的历史区域中的输入项 / 答案对数目。请参阅第 20 页。
BUSY	正在运算或正在作图。
PAUSE	您暂停了作图或程序。
🔒	当前编辑器(数据 / 矩阵编辑器、程序编辑器或文本编辑器)中打开的变元已锁定或已存档且无法修改。

寻找软件版本和 ID 号码

在某些情况下，您需要找出您的 TI-89 / TI-92 Plus 的有关信息，尤其是软件版本和设备的 ID 号码。

显示 "About" 屏

从主屏幕上按下 F1 并选择 A:About。

您的屏幕有可能与右边显示的不同。

按下 ENTER 或 ESC 来关闭该屏幕。

什么时候您需要此信息?

About 屏上的信息是与下列内容有关:

- 如您要为您的 TI-89 / TI-92 Plus 取得新的或升级的软件，那么您就需要提供当前使用的软件的版本和 / 或设备的 ID 号码。
- 如您的 TI-89 / TI-92 Plus 遇到困难并需要联系技术支持服务来寻求协助的话，了解软件的版本将会使问题的分析变得更简便。

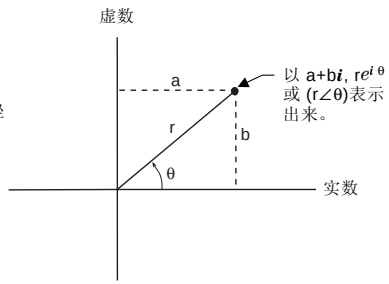
您可在极坐标形式(r∠θ)，其中 r 是数量，而 θ 是角，或在极坐标形式 $re^{i\theta}$ 中输入复数。您也可在直角坐标形式 a+b*i* 中输入复数。

复数综述

一个复数具有实数部分和虚数部分，以确定复平面中的一个点。这些部分是沿着与实数平面上的 x 和 y 轴相类似的实数轴和虚数轴上测得的。

点可以直角坐标形式或两种极坐标形式表示出来。

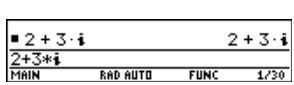
标记 *i* 是代表虚数 $\sqrt{-1}$。



如下面所示，您能够输入的形式是取决于当前的 Angle 模式。

您可使用的形式:	当角的模式设置为:
a+b*i*	Radian 或 Degree
$re^{i\theta}$	只有 Radian (在 Degree 角模式下，此形式会导致出现 Domain error。)
(r∠θ)	Radian 或 Degree

请利用以下的方法输入一个复数。

欲输入:	请如下进行:
直角坐标形式 a+b*i*	替换 a 和 b 的可应用值或变元名称。 a [+] b [2nd] [i] 例如: 

备注: 欲取得标记 i，可按下 [2nd] [i]。请勿随便键入字母 i。

欲输入:	请如下进行:
极坐标形式 $re^{i\theta}$ – 或 – (r∠θ) 对于 (r∠θ) 形式是要求使用括号的。	替换 r 和 θ 的可应用值或变元名称，其中 θ 是按角模式的设置算出的。 **TI-89:** [alpha] [R] [◆] [e^x] [2nd] [i] [◆] [θ] [)] – 或 – [(] [alpha] [R] [2nd] [∠] [◆] [θ] [)] **TI-92 Plus:** R [2nd] [e^x] [2nd] [i] [θ] [)] – 或 – [(] R [2nd] [∠] [θ] [)] 例如: ■ $2\cdot e^{\frac{i\cdot\pi}{4}}$ $\sqrt{2}+\sqrt{2}\cdot i$ ■ $\left(2\angle\frac{\pi}{4}\right)$ $\sqrt{2}+\sqrt{2}\cdot i$ (2∠π/4) MAIN RAD AUTO FUNC 2/30 结果是以直角坐标形式显示，但您可选择极坐标形式。

重要事项: 请勿在 Degree *模式下使用* $re^{i\theta}$*，它会造成* Domain error。

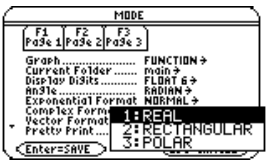
备注: 欲获得符号 e，可按下:
TI-89: [◆] [e^x]。
TI-92 Plus: [2nd] [e^x]
请勿随便键入字母 e。

提示*: 欲获得符号 ∠，可按下* [2nd] [∠]。

提示*: 欲以度为(r∠θ)输入符号，您可键入 ° 符号(比如 45°)。欲获得符号 °，可按下* [2nd] [°]。*您无需为* $re^{i\theta}$ *使用度。*

显示结果的复数格式模式

利用 [MODE] 来将 Complex Format 模式设定成三种设置之一。



您可在任何时候输入复数，与 Complex Format 模式的设置无关。尽管如此，模式设置还是决定了结果是如何显示的。

备注: 您可以任何形式(或所有形式的混合)来输入复数，这取决于 Angle *模式的设置。*

如复数为:	TI-89 / TI-92 Plus 会:
REAL	不显示复数结果，除非您: • 输入复数。 – 或 – • 使用复数功能，如 **cFactor()**、**cSolve()** 或 **cZeros()**。 如显示出复数结果，它们将会是 a+b*i* 或 $re^{i\theta}$ 形式。
RECTANGULAR	以 a+b*i* 显示复数结果。
POLAR	以下列形式显示复数结果: • $re^{i\theta}$，如 Angle 模式 = Radian – 或 – • (r∠θ)，如 Angle 模式 = Degree

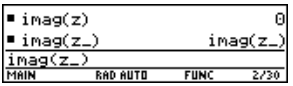
在符号运算中使用复数变元

未定义的变元会作实数进行处理，与 Complex Format 模式设置无关。欲执行复数符号分析，您可利用以下方法来设定复数变元。

方法 1: 使用下划线 _ (**TI-89:** ◆ [_] **TI-92 Plus:** 2nd [_])作为变元名称中的最后一个字符来指定复数变元。例如:

备注: 为使 **cSolve()** *和* **cZeros()** *的运算得到最佳的结果，请利用方法 1。*

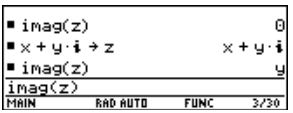
z_ 会作为复数变元(除非 z 已存在，此情形下，该变元会保留现存的数据类型。)处理。



方法 2: 定义一个复数变元。例如:

x+y*i*→z

随后，z 会被当成一个复数变元处理。



复数和度模式

Radian 角模式是建议给复数运算使用的。就内部而言，TI-89 / TI-92 Plus 会将所有输入的三角数值转换成弧度，但对于指数、对数或双曲函数，它不能进行这种转换。

备注: 如您使用 Degree *角模式，您则必须在形式(r∠θ)中输入极坐标输入项。在* Degree *角模式中，* $re^{i\theta}$ *输入项会造成错误。*

在 Degree 角模式下，复数恒等式如 $e^{\wedge}(\boldsymbol{i}\theta) = \cos(\theta) + \boldsymbol{i}\sin(\theta)$ 一般不成立，这是因为 cos 和 sin 的值会被转换成弧度，而 $e^{\wedge}()$ 则不会。例如， $e^{\wedge}(\boldsymbol{i}45) = \cos(45) + \boldsymbol{i}\sin(45)$ 内部会被当作 $e^{\wedge}(\boldsymbol{i}45) = \cos(\pi/4) + \boldsymbol{i}\sin(\pi/4)$ 处理。复数恒等式在 Radian 角模式下总是成立的。

您可运行以汇编语言为 TI-89 / TI-92 Plus 编写的程序。一般而言，与利用内建程序编辑器编写的击键程序相比，汇编语言程序的运行更快速，可提供更高的可控程度。

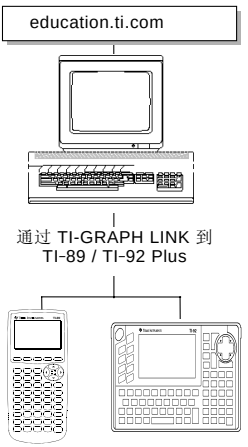
何处能取得 Assembly-Language (汇编语言)程序

汇编语言程序和击键程序均在 Texas Instruments 的网址上备有，具体地址:

education.ti.com

在此网址上备有的程序提供了内 TI-89 / TI-92 Plus 没有建立的额外功能和性能。请访问 Texas Instruments 的网址来取得最新的信息。

从网址下载了程序到您的计算机后，利用一条 TI-GRAPH LINK™ 计算机对计算器连接电缆(单独供应)来将程序发送到您的 TI-89 / TI-92 Plus 上。请参阅与 TI-GRAPH LINK 一起提供的使用手册。



关于 TI-GRAPH LINK 的注释

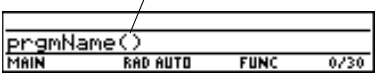
如您拥有 TI-GRAPH LINK 计算机对计算器连接电缆和用于 TI-92 的软件，应注意，TI-GRAPH LINK *软件* 与 TI-89 / TI-92 Plus 是不兼容的。而电缆则可与上述的两个单元一起工作。有关如何取得 TI-GRAPH LINK 软件或计算机对计算器连接电缆的信息，请访问 Texas Instruments 的网址，地址是: **education.ti.com/guides**，或通过电子邮件(**ti-cares@ti.com.**)联系 Texas Instruments Customer Support。

运行汇编语言程序

当 TI-89 / TI-92 Plus 汇编语言程序已储存在您的计算器内之后，您可象在主屏幕运行其它任何程序一样运行它。

如程序要求一个或多个变数，请将其键入括号( )内。请参阅程序的有关资料来熟悉所要求的变数。

提示: 如程序不在当前的文件夹，请确认特定的路径名。



您可在其它程序中把汇编语言程序作为子程序来调用及删除之，或作为任何其它程序一样来使用。

运行一个程序的快捷方式

在主屏幕上，您可利用键盘快捷方式来运行多达九个用户定义的程序或汇编语言程序。然而，程序必须具备以下的名称。

备注: 程序必须储存在 MAIN *文件夹中。同样，您不能利用快捷方式来运行要求使用变数的程序。*

在主屏幕上按下:	欲运行任何已命名的程序(如有的话):
◆ 1	kbdprgm1()
⋮	⋮
◆ 9	kbdprgm9()

如您有的程序带有不同的名称和您想利用快捷方式来运行之，可将现有的程序复制或重命名成 kbdprgm1() 等。

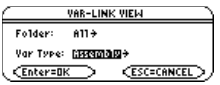
您不能编辑汇编语言程序

您不能利用 TI-89 / TI-92 Plus 来编辑汇编语言程序，内建的程序编辑器不能打开汇编语言程序。

显示汇编语言程序列表

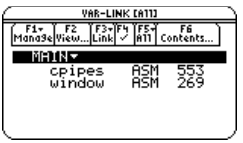
欲列出内存中储存的汇编语言程序:

1. 显示 VAR-LINK 屏(2nd [VAR-LINK])。
2. 按下 F2 View。
3. 选择可应用的文件夹(或 All 文件夹)并设定 Var Type = Assembly。



备注: 汇编语言程序备有 ASM 数据类型。

4. 按下 ENTER 来显示汇编语言程序的列表。



关于编写汇编语言程序的信息

教授初学者编写汇编语言程序所需的信息是超出本手册的发行目的。然而，如您拥有汇编语言方面的工作经验，请访问 Texas Instruments 的网址(**education.ti.com**)来取得有关如何存取 TI-89 / TI-92 Plus 功能的特殊信息。

备注: 您必须使用计算机来编写汇编语言程序。汇编语言程序不能在计算器键盘上创建。

TI-89 / TI-92 Plus 也含有执行由系列 Motorola 68000 运算码构成的字符串的 **Exec** 命令，这些编码象汇编语言程序的其它形式一样起作用。请访问 Texas Instruments 网址来获取备有的信息。

警告: Exec 可给您存取微计算机的所有功能。请记住，您很容易会出现错误而造成计算器锁死和丢失您的数据。我们建议您在尝试使用 **Exec** 命令之前，先备份计算器中的内容。

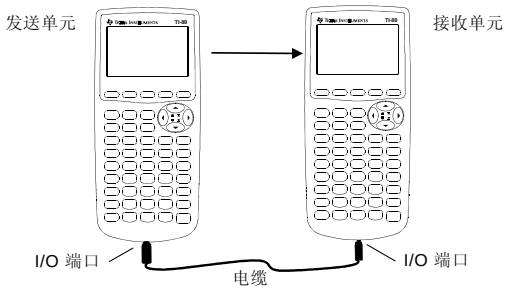
TI-89 和 TI-92 Plus 均备有一条让您链接两单元的电缆一同提供。当连接起来后，您就可以在两单元间传送信息了。

发送或接收之前的 连接

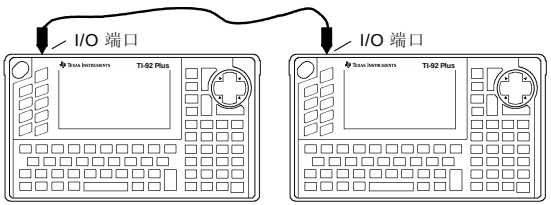
利用紧固连接口，将电缆的一端插入每一单元的 I/O 端口。两单元都可发送和接收信息，这取决于您如何在 VAR-LINK 屏上设定它们了。

下图示出如何将两台 TI-89 链接在一起。

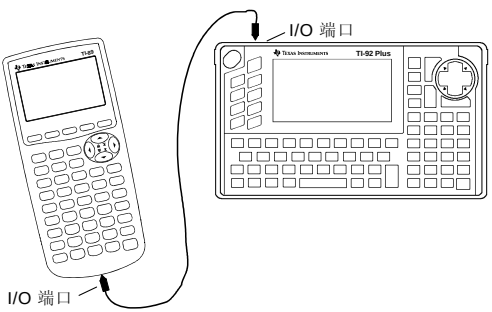
备注: 您可将一台 TI-89 或 TI-92 Plus 链接到另外一台 TI-89、TI-92 Plus 或 TI-92 上，但不能链接到某些其它图形计算器如 TI-81、TI-82、TI-83、TI-83 Plus、TI-85 或 TI-86 上。



下图示出如何将两台 TI-92 Plus 链接在一起。



您也可利用与计算器一同提供的 TI-GRAPH LINK 电缆将一台 TI-89 和一台 TI-92 Plus 链接起来。



传送变元、Flash 应用程序和文件夹

传送变元是共享列在 VAR-LINK 屏上的任何变元的便捷途径，这些变元包括有函数和程序等。您也可传送 Flash 应用程序和文件夹。

设置单元

大多数 Flash 应用程序只能从一台 TI-89 传送到一台 TI-89，或从一台 TI-92 Plus 传送到一台 TI-92 Plus。您不能将 Flash 应用程序发送到一台 TI-92 上，除非它带有 Plus 模块和 Advanced Mathematics (高等数学) 2.x 产品软件(基本码)。

1. 按照第 57 页中叙述的内容链接两个单元。
2. 在 **sending** (**发送**)单元上，按下 [2nd] [VAR-LINK] 来显示 VAR-LINK 屏。
3. 在**发送**单元上，选择您要发送的变元、文件夹或 Flash 应用程序。当选定后，折叠的文件夹会变成展开。
   - 欲选择单个变元或 Flash 应用程序，可移动光标将其突出显示。
   - 欲选择单个文件夹，可将其突出显示并按下 [F4]，以在旁边放置一个选中标记(✓)。这样会选择文件夹及其内容。
   - 欲选择多个变元、Flash 应用程序或文件夹，可将每一个突出显示并按下 [F4]，以在旁边放置一个选中标记(✓)。
   - 欲选择所有变元、Flash 应用程序或文件夹，可使用 [F5] All 1:Select All。

备注: *使用 [F4] 来选择多变元、Flash 应用程序或文件夹。再次使用 [F4] 来将您不想传送的任何项目去除选择。*

4. 在(**接收**)单元上，按下 [2nd] [VAR-LINK] 来显示 VAR-LINK 屏。(发送单元会保持显示在 VAR-LINK 屏上。)
5. 分别在接收*和*发送单元上按下 [F3] Link 来显示菜单选项。
6. 在**接收**单元上，选择 2:Receive。

   此时，接收单元状态行上会显示出信息 VAR-LINK: WAITING TO RECEIVE 和繁忙指示符 BUSY。
7. 在**发送**单元上选择如下:
   - 1:Send to TI-89/92 Plus

   — 或 —
   - 3:Send to TI-92

   这样就会开始传送了。

传送期间，在接收单元的状态行上会显示出一进度条。当传送完成后，接收单元上的 VAR-LINK 屏会得到更新。

变元、Flash 应用程序或文件夹的传送规则

发送和接收单元上带有相同名称的未锁定和未存档变元，会被来自发送单元的名称覆盖。

发送和接收单元上带有相同名称的已锁定和已存档变元，在被来自发送单元的名称覆盖前，必须先在接收单元上松锁或解除存档。

对于 Flash 应用程序或文件夹，您只能锁定但不能存档。

备注: 您不能向 TI-92 发送已存档的变元，您必须先将其解除存档。

备注: 在发送文件夹或其内容前，您必须展开该文件夹。

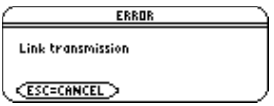
如您选择:	将会出现以下情况:
未锁定的变元	变元已传送到当前的文件夹，并在接收单元保持为未锁定形式。
锁定的变元	变元已传送到当前的文件夹，并在接收单元保持为锁定形式。
已存档的变元	变元已传送到当前的文件夹，并在接收单元保持为已存档形式。
未锁定的 Flash 应用程序	如接收单元备有准确的证书，Flash 应用程序就会被传送出去，它会在接收单元中保持为未锁定形式。
锁定的 Flash 应用程序	如接收单元备有准确的证书，Flash 应用程序就会被传送出去，它会在接收单元中保持为锁定形式。
未锁定的文件夹	文件夹及其选定的内容会被发送出去，文件夹在接收单元中保持为未锁定形式。
已锁定的文件夹	文件夹及其选定的内容会被发送出去，文件夹在接收单元中变成未锁定形式。

取消传送

对于发送和接收单元:

1. 按下 [ON]。

   会显示出一个错误信息。



2. 按下 [ESC] 或 [ENTER]。

升级产品软件(基本码)

您可升级您 TI-89 / TI-92 Plus 中的产品软件(基本码)。您也可从一台 TI-89 或 TI-92 Plus 将产品软件传送到另一台上，而接收单元则必须具有允许运行该软件的准确证书。

产品软件(基本码)升级

*产品软件*的期限条款包括以下这些基本码升级:

- 保养升级(免费)。
- 功能升级(其中一些要收费)。在 Texas Instruments 网址上下载您收费的功能升级文件之前，您必须提供您计算器的电子 ID 号码。此信息是用于创建自定义的电子证书，以指明您单元内的产品软件是经过许可运行的。

安装保养升级或功能升级产品均会将计算器内存重设回原始的工厂预设值。这意味着所有用户定义变元、程序、列表和 Flash 应用程序均会被删除。在执行基本码产品(保养或功能)升级之前，请先参阅下面有关电池的重要信息和第 61 页“安装产品软件(基本码)之前的单元备份”一节中叙述的重要信息。

重要产品软件(基本码)下载信息

开始下载基本码产品(保养或功能升级)之前，应先给单元装上崭新的电池。

当处于基本码产品下载模式时，Automatic Power Down™ (APD™)功能是不能运作的。如您在实际开始下载前，让计算器停留在下载模式的时间过度延长，那么电池的电量就会大量耗费。这样，您就必须在下载前用新电池替换耗尽的电池。

您也可利用单元对单元连接电缆将基本码产品从一台计算器传送到另一台上。如您传送完成之前意外地中断了操作，则必须使用一台计算机来重新安装之。此外，切记在下载前给单元装上新的电池。

如您没有什么把握进行上述操作，请按第 228 页中提供的资料联系 Texas Instruments。

安装产品软件(基本码)之前的单元备份

当您安装产品软件(基本码)升级时，其安装过程如下:

- 删除所有用户定义变元(在 RAM 和用户数据档案中)、函数、程序和文件夹。
- 删除所有 Flash 应用程序。
- 将所有系统变元和模式重设回原始的工厂设定值。此操作与使用 MEMORY 屏重设所有内存相同。

重要事项: *安装前请装上崭新的电池。*

欲保留任何现存的变元或 Flash 应用程序，可在*安装升级产品前*执行如下:

- 按照第 58 页叙述的内容将变元或 Flash 应用程序转移到另一台计算器上。

  — 或 —

备注: *计算机对计算器连接电缆和您的计算器一起提供的电缆有所不同。*

- 使用一条 TI-GRAPH LINK™ 计算机对计算器连接电缆(单独提供)和 TI-GRAPH LINK 软件(在 Texas Instruments 的网址上可免费获得)来将变元和 / 或 Flash 应用程序发送到计算机上。

如您有 TI-GRAPH LINK 计算机对计算器连接电缆和 TI-92 用软件的话，应注意，TI-92 TI-GRAPH LINK *软件*与 TI-89 或 TI-92 Plus 是不兼容的。尽管如此，该电缆还是能够与其它所有单元一起工作的。有关如何取得 TI-89 / TI-92 Plus 用 TI-GRAPH LINK 计算机对计算器连接电缆的信息，请访问 Texas Instruments 的网址:

education.ti.com

欲联系 Texas Instruments Customer Support:

电话: **1-800-TI-CARES (1-800-842-2737)**
电子邮件: **ti-cares@ti.com**

转移产品软件(基本码)

如发送的 TI-89 或 TI-92 Plus 带有正宗的产品软件(基本码)或免费保养升级产品，那么接收的 TI-89 或 TI-92 Plus 则不需要新的认可证书。其当前的证书是有效的，保养升级产品将会得到转移。

如发送的 TI-89 或 TI-92 Plus 带有收费的功能升级产品，那么接收单元的升级就要购买了。一旦购买后，就可以下载一份许可证书并装入接收单元中。安装了证书后，功能升级就可以转移了。

您可以看到您的 TI-89 / TI-92 Plus 内的软件是哪个版本的，只要在主屏幕上按下 F1 并选择 A:About 即可。

产品软件(基本码)只能从一台 TI-89 到另一台 TI-89 或从一台 TI-92 Plus 到另一台 TI-92 Plus 地转移。您不能发送 Advanced Mathematics 2.x 产品软件(基本码)到一台 TI-92 中，除非它该带有 Plus 模。

欲进行单元对单元的产品软件(基本码)转移:

重要事项: 对于每一接收单元，请记住将所需的信息备份和安装新电池。

1. 按照第 57 页的说明链接两单元。
2. 在接收*和*发送单元上，按下 2nd [VAR-LINK] 来显示 VAR-LINK 屏。
3. 在接收*和*发送单元上，按下 F3 Link 来显示菜单选项。

重要事项: 请确认发送和接收单元均处于 VAR-LINK 屏。

4. 在接收单元上，选择 5:Receive Product SW。

   此时，会显示出一个警告信息。可按下 ESC 来停止转移或按下 ENTER 来执行转移。按下 ENTER 后，接收单元的状态行中会显示出信息 VAR-LINK: WAITING TO RECEIVE 和 BUSY。

5. 在发送单元上，选择 4:Send Product SW。

   此时，会显示出一个警告信息。可按下 ESC 来停止转移或按下 ENTER 来开始发送。

转移产品软件(续上)

在传送期间，接收单元会显示出传送的进程，当传送完成后:

- 发送单元会返回 VAR-LINK 屏。
- 接收单元会返回主屏幕。您也许需要利用 ◆ ⊟ (明亮)或 ◆ ⊞ (暗淡)来调整对比度。

切勿尝试取消产品软件(基本码)的传送

开始转移后，接收单元内现有的基本码会被实际地删除，如您在转移完成之前中断了操作，接收单元就会运作不正确。您必须使用一台计算机来重新安装基本码(保养或功能)升级产品。

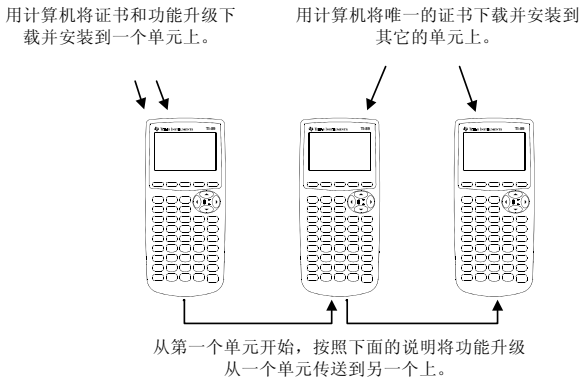
如您要在多单元上升级产品软件(基本码)

欲在多单元上进行保养升级，您可从一个单元转移到另一个上进行升级，不需使用计算机来安装在每一单元上。保养升级是免费提供的，在下载和安装前，您无须先获得许可证书。

备注: *同样备有群组证书。*

在安装一个收费的功能升级前，每一台 TI-89 或 TI-92 Plus 都必须具备自有的许可证书。在下载和安装期间，您可选择证书和功能升级，或者只选证书。以下图示表明了为收费功能升级而预备多单元的最有效途径。

提示: *通常而言，单元对单元的基本码升级传送较使用计算机安装为快。*



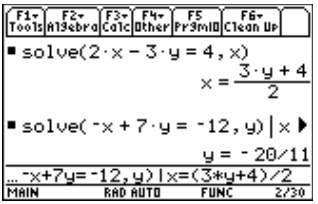
为多台 TI-92 Plus 的收费功能升级而预备的工作方式与上图相同。

综述

2

执行运算 66
符号操作 69
常数和测量单位 70
基本函数作图 71
参数作图 72
极坐标作图 73
序列作图 74
三维作图 75
微分方程作图 77
附加作图提示 79
表格 80
拆分屏幕 81
数据 / 矩阵编辑器 82
统计图和数据图 83
编程 87
文本操作 89
数值解答器 90
数字基 91
内存和变元管理 92

本章中的每一例子均提供了逐步过程、实际击键和实例显示。

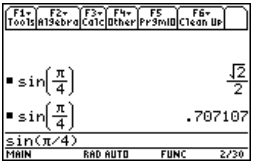
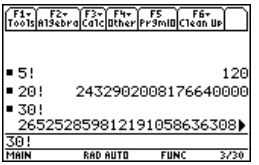
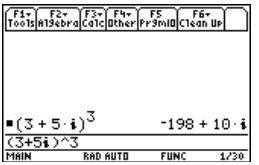
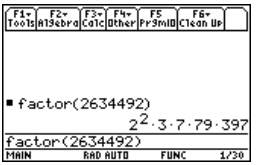


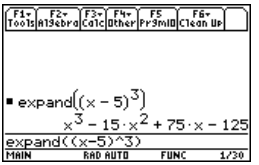
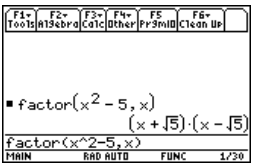
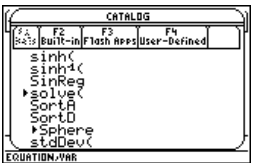
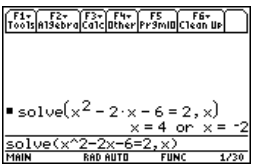
欲掌握有关 TI-89 / TI-92 Plus 的资料，请参阅随附的 CD-ROM 上的综合指南手册或访问德州仪器公司的网址:

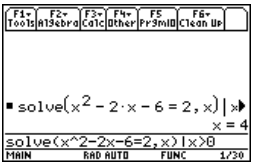
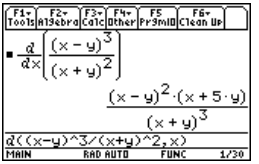
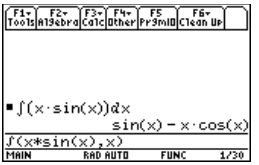
education.ti.com

执行运算

本节为您的操作提供了一系列的例子，从而演示出 TI-89 / TI-92 Plus 的一些计算功能。在执行每一例子之前，每一屏幕的历史区域均可通过按下 [F1] 并选择 8:Clear Home 来清除，以便仅显示出该例击键的结果。

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
运算显示			
1. 计算 sin(π/4) 并以符号及数字格式显示结果。 *欲清除前一运算的历史区域，可按下 [F1] 并选择* 8:Clear Home。	[2nd] [SIN] [2nd] [π] [÷] 4 [)] [ENTER] [◆] [ENTER]	[SIN] [2nd] [π] [÷] 4 [)] [ENTER] [◆] [ENTER]	
求数字的阶乘			
1. 计算一系列数字的阶乘来查看 TI-89 / TI-92 Plus 是如何处理极大的整数。 *欲取得阶乘算子 (!)，可按下 [2nd] [MATH]，选择* 7:Probability *后再选择* 1:! 。	5 [2nd] [MATH] 7 1 [ENTER] 2 0 [2nd] [MATH] 7 1 [ENTER] 3 0 [2nd] [MATH] 7 1 [ENTER]	5 [2nd] W [ENTER] 2 0 [2nd] W [ENTER] 3 0 [2nd] W [ENTER]	
展开复数			
1. 计算 $(3+5i)^3$ 来观看 TI-89 / TI-92 Plus 是如何处理包含复数的运算。	[(] 3 [+] 5 [2nd] [i] [)] [^] 3 [ENTER]	[(] 3 [+] 5 [2nd] [i] [)] [^] 3 [ENTER]	
求质数因子			
1. 计算有理数 2634492 的因子。 *您可在键盘上键入* factor，*或可通过按下* [F2] *并选择* 2:factor(，*从而达到在输入行上输入 "factor" 的目的。* 2. 请您亲自输入其它数字。(*任选项*)	[F2] 2 2 6 3 4 4 9 2 [)] [ENTER]	[F2] 2 2 6 3 4 4 9 2 [)] [ENTER]	

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
展开表达式			
1. 展开表达式 $(x-5)^3$。 *您可在键盘上键入* expand，*或可通过按下* F2 *并选择* 3:expand(，*从而达到在输入行上输入 "expand" 的目的。*	F2 3 (X − 5) ^ 3) ENTER	F2 3 (X − 5) ^ 3) ENTER	
2. 请您亲自输入其它表达式。(*任选项*)			
化简表达式			
1. 将表达式 $(x^2-2x-5)/(x-1)$ 化简成它的最简形式。 *您可在键盘上键入* propfrac，*或可通过按下* F2 *并选择* 7:propFrac(，*从而达到在输入行上输入 "propfrac" 的目的。*	F2 7 (X ^ 2 − 2 X − 5) ÷ (X − 1)) ENTER	F2 7 (X ^ 2 − 2 X − 5) ÷ (X − 1)) ENTER	
将多项式分解因式			
1. 对关于 x 的多项式 (x^2-5) 进行分解因式。 *您可在键盘上键入* factor，*或可通过按下* F2 *并选择* 2:factor(，*从而达到在输入行上输入 "factor" 的目的。*	F2 2 X ^ 2 − 5 , X) ENTER	F2 2 X ^ 2 − 5 , X) ENTER	
解方程			
1. 解关于 x 的方程 $x^2-2x-6=2$。 *您可在键盘上键入 solve、在 Catalog(目录)菜单上选择* "solve(" *或按下* F2 *并选择* 1:solve(，*从而达到在输入行上输入 "solve" 的目的。* *状态行区域会显示出 Catalog 菜单中标记的•项所要求的句法。*	F2 1 X ^ 2 − 2 X − 6 = 2 , X) ENTER	F2 1 X ^ 2 − 2 X − 6 = 2 , X) ENTER	 

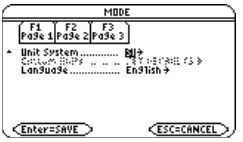
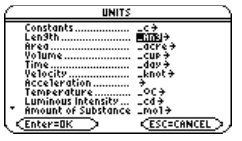
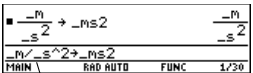
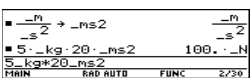
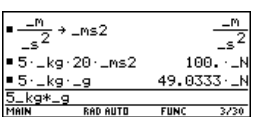
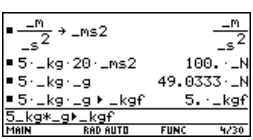
步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
解具有域约束的方程			
1. 解有关 x 的方程 $x^2-2x-6=2$，其中 x 大于零。 *"with" (|)算子提供了域约束。* ***TI-89:*** [I] ***TI-92 Plus:*** [2nd] [|]	[F2] 1 X [^] 2 [−] 2 X [−] 6 [=] 2 [,] X [)] [I] X [2nd] [>] 0 [ENTER]	[F2] 1 X [^] 2 [−] 2 X [−] 6 [=] 2 [,] X [)] [2nd] [|] X [2nd] [>] 0 [ENTER]	
求函数的导数			
1. 求有关 x 的函数 $(x-y)^3/(x+y)^2$ 的导数。 *本例子是说明微积分微分法功能的使用和函 数是如何在历史区域以 "工整印刷体"方式显示出来的。*	[2nd] [d] [(] X [−] Y [)] [^] 3 [÷] [(] X [+] Y [)] [^] 2 [,] X [)] [ENTER]	[2nd] [d] [(] X [−] Y [)] [^] 3 [÷] [(] X [+] Y [)] [^] 2 [,] X [)] [ENTER]	
求函数的积分			
1. 求有关 x 的函数 x∗sin(x) 的积分。 *本例子说明微积分积分法功能的使用。*	[2nd] [∫] X [×] [2nd] [SIN] X [)] [,] X [)] [ENTER]	[2nd] [∫] X [×] [SIN] X [)] [,] X [)] [ENTER]	

解方程组 2x – 3y = 4 和 ⁻x + 7y = ⁻12。解第一个方程，使 x 由 y 表示。将 x 的表达式代入第二个方程中并解出 y 值，然后再将 y 值代入第一个方程中来求 x 值。

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 显示主屏幕并清空输入行。解方程 2x – 3y = 4 的 x。 F2 1 是从 Algebra *(代数)*菜单上选择 **solve(**。*您也可直接在键盘上键入* **solve(** *或从* Catalog *菜单上选择之。*	HOME CLEAR CLEAR F2 1 2 X − 3 Y = 4 , X) ENTER	♦ HOME CLEAR CLEAR F2 1 2 X − 3 Y = 4 , X) ENTER	solve(2·x − 3·y = 4, x) x = (3·y + 4)/2 solve(2x−3y=4,x) MAIN RAD AUTO FUNC 1/30
2. 开始解出方程 ⁻x + 7y = ⁻12 中的 y，但还是不要按下 ENTER。	F2 1 (−) X + 7 Y = (−) 1 2 , Y)	F2 1 (−) X + 7 Y = (−) 1 2 , Y)	
3. 使用 “with” 算子(2nd K)来替代第一方程中计算出的 x 的表达式。这样就给出了 y 值。 *在屏幕上，“with” 算子是以* **|** *显示出来。* *使用自动粘贴功能来将历史区中的上一答案突出显示并将其粘贴在输入行中。*	I ▲ ENTER ENTER	2nd [I] ▲ ENTER ENTER	solve(2·x − 3·y = 4, x) x = (3·y + 4)/2 solve(−x + 7·y = −12, y) | x ▶ y = −20/11 ...−x+7y=−12,y)|x=(3*y+4)/2 MAIN RAD AUTO FUNC 2/30
4. 将历史区中 x 的方程突出显示。	▲▲▲	▲▲▲	solve(2·x − 3·y = 4, x) x = (3·y + 4)/2 solve(−x + 7·y = −12, y) | x ▶ y = −20/11 ...−x+7y=−12,y)|x=(3*y+4)/2 MAIN RAD AUTO FUNC 2/2
5. 自动粘贴已突出显示的表达式到输入行中，然后替换从第二方程中计算出 的 y 值。 解即为: x = ⁻8/11 和 y = ⁻20/11	ENTER I ▲ ENTER ENTER	ENTER 2nd [I] ▲ ENTER ENTER	solve(−x + 7·y = −12, y) | x ▶ y = −20/11 x = (3·y + 4)/2 | y = −20/11 x = −8/11 x=(3*y+4)/2|y=−20/11 MAIN RAD AUTO FUNC 3/30

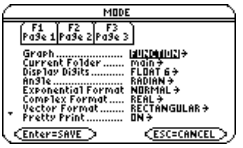
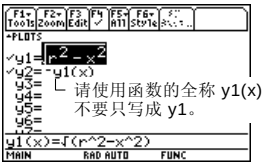
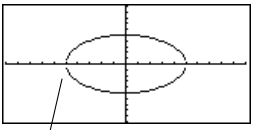
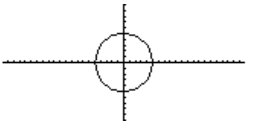
此例子是符号操作的演示。对于解方程组，备有一步解出的功能。

利用方程 f = m∗ a，计算当 m = 5 千克和 a = 20 米 / 秒^{2}时的力。问当a = 9.8 米 / 秒^{2}时的力又如何?(这是的重力加速度，其中常数成为 _g)。请将力的结果从牛顿换算成千克。

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 显示第 3 页中的 MODE 对话框。对于 Unit System 模式，请为测量的公制单位选择 SI。 *结果会按这些默认的单位显示出来。*	MODE F3 ▷ 1 ENTER	MODE F3 ▷ 1 ENTER	
2. 为米 / 秒^{2}创建加速度单位，称作 _ms2. *UNITS 对话框可让您从类别字母列表上选择单位。您可使用 2nd ▽ 和 2nd △ 来在类别中每次滚动一页。* *至此，您就可使用 _ms2 而不必在需要时每次重新输入 _m/_s^2 了。* *此外，您可使用 UNITS 对话框来从加速度类别中选择 _ms2。*	2nd [UNITS] ▽ ▷ M ENTER ÷ 2nd [UNITS] ▽ ▽ ▽ ▷ S ENTER △ 2 STO▸ ◆ [_] 2nd [a-lock] M S alpha 2 ENTER	◆ [UNITS] ▽ ▷ M ENTER ÷ ◆ [UNITS] ▽ ▽ ▽ ▷ S ENTER △ 2 STO▸ 2nd [_] M S 2 ENTER	  如您使用 UNITS 对话框来选择单位，那么符号 _ 就会自动输入的。
3. 计算当 m = 5 千克(_kg)和 a = 20 米 / 秒^2(_ms2)时的力。 *如果您懂得单位的缩写，也可从键盘键入之。*	5 ◆ [_] 2nd [a-lock] K G alpha × 2 0 ◆ [_] 2nd [a-lock] M S alpha 2 ENTER	5 2nd [_] K G × 2 0 2nd [_] M S 2 ENTER	
4. 利用同一个 m 来计算重力加速度的力(常数为 _g)。 *对于 _g，您可使用 UNITS 对话框中可用的预定义常数，也或键入 _g。*	5 ◆ [_] 2nd [a-lock] K G alpha × 2nd [UNITS] ▷ alpha G ENTER ENTER	5 2nd [_] K G × ◆ [UNITS] ▷ G ENTER ENTER	
5. 换算成千克力(_kgf)。 *2nd [▸] 可显示换算算子 ▸。*	▷ 2nd [▸] ◆ [_] 2nd [a-lock] K G F alpha ENTER	▷ 2nd [▸] 2nd [_] K G F ENTER	

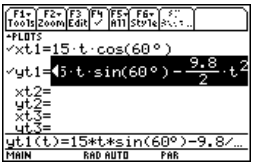
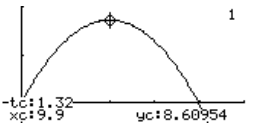
基本函数作图

作出半径为 5，圆心位于坐标系原点的圆。利用标准观测窗(ZoomStd)来查看圆，然后使用 ZoomSqr 来调整观察窗。

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 显示 MODE 对话框，对于 Graph 模式请选择 FUNCTION。	MODE ▶ 1 ENTER	MODE ▶ 1 ENTER	
2. 显示主屏幕，然后将半径 5 储存在变元 r 内。	HOME 5 STO▸ alpha R ENTER	◆ [HOME] 5 STO▸ R ENTER	5→ r
3. 显示并清空 Y= 编辑器，然后定义圆的上半部分的函数 y1(x) = $\sqrt{r^2 - x^2}$。 *在函数作图模式下，您必须单独定义圆的上半部分和下半部分。*	◆ [Y=] F1 8 ENTER ENTER 2nd [√] alpha R ^ 2 − X ^ 2) ENTER	◆ [Y=] F1 8 ENTER ENTER 2nd [√] R ^ 2 − X ^ 2) ENTER	
4. 定义圆的下半部分的函数 y2(x) = $^-\sqrt{r^2 - x^2}$。 *下半部分是上半部分的负数，所以，您可以定义成 y2(x) = ⁻y1(x)。*	ENTER (-) Y 1 (X) ENTER	ENTER (-) Y 1 (X) ENTER	 请使用函数的全称 y1(x)，不要只写成 y1。
5. 选择自动作出函数图形的 ZoomStd 观察窗。 *x 和 y 轴范围的标准观察窗是从 ⁻10 到 10。然而，此范围沿 x 轴的伸展比 y 轴的长。因此，圆会呈椭圆形。*	F2 6	F2 6	 注意上下半圆结合部位的轻微间隙。
6. 选择 ZoomSqr。 *ZoomSqr 可增大沿 x 轴的范围，所以，圆和正方形会呈正确的比例。*	F2 5	F2 5	

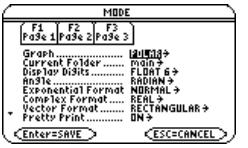
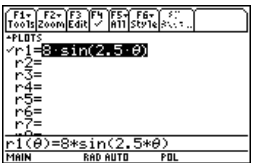
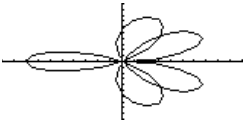
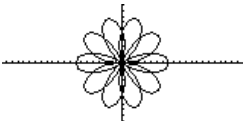
备注: 圆的上半部与下半部间之所以存在间隙，其原因是每一半部是一个独立的函数。每一半部的数学终点为 (-5,0) 和 (5,0)。取决于观察窗，然而，每一半部*画出*的终点也许会与它们的*数学*终点略有不同。

作出参数方程的图形，表示一个以 60° 度角(θ)、初速度(v_0)为 15 米 / 秒踢出的球的轨迹。重力常数 g = 9.8 米 / 秒^2。忽略空气阻力和其它阻力，问球的最大高度是多少?何时碰撞地面?

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 显示 MODE 对话框，对于 Graph 模式请选择 PARAMETRIC。	[MODE] ▼ 2 [ENTER]	[MODE] ▼ 2 [ENTER]	
2. 显示和清空 Y= 编辑器。然后定义水平分量 xt1(t) = v_0t cos θ。 *输入 v_0 和 θ 的值。* **TI-89:** 键入 T [×] [2nd] [COS]，*而不是* T [2nd] [COS]。 **TI-92 Plus:** *键入* T [×] [COS]，*而不是* T [COS]。 *按下* [2nd] [°] *或* [2nd] [MATH] 2 1 *之一来输入符号 "°"。这样可确保以度来解释，与角模式无关。*	[◆] [Y=] [F1] 8 [ENTER] [ENTER] 1 5 T [×] [2nd] [COS] 6 0 [2nd] [°] [)] [ENTER]	[◆] [Y=] [F1] 8 [ENTER] [ENTER] 1 5 T [×] [COS] 6 0 [2nd] [°] [)] [ENTER]	xt1(t)=15t*cos(60°)
3. 定义垂直分量yt1(t) = v_0t sin θ – (g/2)t^2。 *输入 v_0、θ 和 g 的数值。*	[ENTER] 1 5 T [×] [2nd] [SIN] 6 0 [2nd] [°] [)] [−] [(] 9 . 8 [÷] 2 [)] T [^] 2 [ENTER]	[ENTER] 1 5 T [×] [SIN] 6 0 [2nd] [°] [)] [−] [(] 9 . 8 [÷] 2 [)] T [^] 2 [ENTER]	
4. 显示窗口编辑器，输入适合此例子的窗口变元。 *您可按下 ▼ 或 [ENTER] 之一来输入数值及移到下一变元。*	[◆] [WINDOW] 0 ▼ 3 ▼ . 0 2 ▼ [(−)] 2 ▼ 2 5 ▼ 5 ▼ [(−)] 2 ▼ 1 0 ▼ 5	[◆] [WINDOW] 0 ▼ 3 ▼ . 0 2 ▼ [(−)] 2 ▼ 2 5 ▼ 5 ▼ [(−)] 2 ▼ 1 0 ▼ 5	tmin=0. tmax=3. tstep=.02 xmin=-2. xmax=25. xscl=5. ymin=-2. ymax=10. yscl=5.
5. 作出参数方程的图形来模仿球的轨迹。	[◆] [GRAPH]	[◆] [GRAPH]	
6. 选择 Trace (跟踪)，然后，将光标沿轨迹移动来寻找: • 最大高度时的 y 值。 • 球与地面碰撞处的 t 值。	[F3] ▶ 或 ◀ (如有需要的话)	[F3] ▶ 或 ◀ (如有需要的话)	

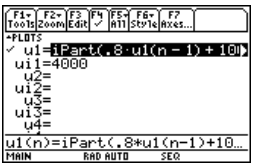
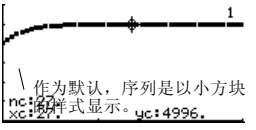
极坐标作图

作出极坐标方程 A sin Bθ 的图形，以构成玫瑰的形状。作出 A=8 和 B=2.5 的玫瑰图。然后用其它 A 和 B 值来研究玫瑰的外观。

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 显示 MODE 对话框，对于 Graph 模式请选择 POLAR。对于 Angle 模式，请选择 RADIAN。	[MODE] ▶ 3 ▼ ▼ ▼ ▶ 1 [ENTER]	[MODE] ▷ 3 ▽ ▽ ▽ ▷ 1 [ENTER]	
2. 显示和清空 Y= 编辑器，定义极坐标方程 r1(θ) = A sin Bθ。 *将 8 和 2.5 相应输入 A 和 B 中。*	[◆] [Y=] [F1] 8 [ENTER] [ENTER] 8 [2nd] [SIN] 2 . 5 [◆] [θ] [)] [ENTER]	[◆] [Y=] [F1] 8 [ENTER] [ENTER] 8 [SIN] 2 . 5 [θ] [)] [ENTER]	
3. 选择作方程图形的 ZoomStd 观察窗。 • *图形中仅显示出五块玫瑰瓣。* – *在标准观察窗中，Window (窗口)变元 θmax = 2π。其余花瓣的 θ 值均大于 2π。* • *玫瑰不能对称显示。* – *x 和 y 轴的范围均是从 −10 到 10。然而，此范围沿着 x 轴的伸展较 y 轴为长。*	[F2] 6	[F2] 6	
4. 显示窗口编辑器，并将 θmax 改为 4π。 *4π 将会在您离开窗口编辑器时计算成一个数字。*	[◆] [WINDOW] ▼ 4 [2nd] [π]	[◆] [WINDOW] ▽ 4 [2nd] [π]	θmin=0. θmax=4π θstep=.13089969389957 xmin=-10. xmax=10. xscl=1. ymin=-10. ymax=10. yscl=1.
5. 选择重新作出方程图形的 ZoomSqr。 *ZoomSqr 可增大沿 x 轴的范围，所以，图形看起来会成正确的比例。*	[F2] 5	[F2] 5	
6. 如有需要，您可改变 A 和 B 的值并重 新作出方程图形。			

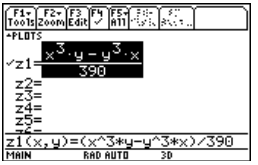
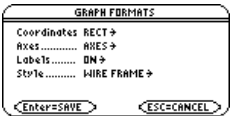
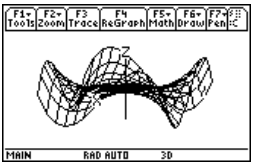
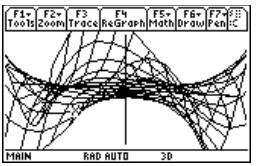
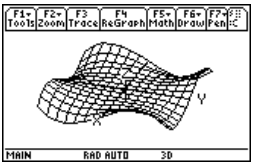
一个小树林有 4000 棵树，每年将砍伐 20% 的树木(保留 80%)并种植 1000 棵新树。请使用序列方式来计算每年年底的树木数量。问树木能否稳定在某一数目?

起始数	**1** 年之后	**2** 年之后	**3** 年之后	**. . .**
4000	.8 x 4000 + 1000	.8 x (.8 x 4000 + 1000) + 1000	.8 x (.8 x (.8 x 4000 + 1000) + 1000) + 1000	. . .

步骤	**TI-89** 击键内容	**TI-92 Plus** 击键内容	显示
1. 显示 MODE 对话框，对于 Graph 模式请选择 SEQUENCE。	[MODE] ▶ 4 [ENTER]	[MODE] ▷ 4 [ENTER]	
2. 显示并清空 Y= 编辑器。序列定义为 u1(n) = iPart(.8∗ u1(n− 1)+1000)。 *使用* **iPart** *来获取结果的整数部分。树木的砍伐是以棵计算。* *欲存取* **iPart(**，*可使用* [2nd][MATH] *并键入之，或从* CATALOG *菜单上选择。*	[◆][Y=] [F1] 8 [ENTER] [ENTER] [2nd][MATH] 1 4 . 8 [alpha] U 1 [(] [alpha] N [−] 1 [)] [+] 1 0 0 0 [)] [ENTER]	[◆][Y=] [F1] 8 [ENTER] [ENTER] [2nd][MATH] 1 4 . 8 U 1 [(] N [−] 1 [)] [+] 1 0 0 0 [)] [ENTER]	
3. 把作为第一项使用的 ui1 定义成起始值。	[ENTER] 4 0 0 0 [ENTER]	[ENTER] 4 0 0 0 [ENTER]	
4. 显示窗口编辑器，设定 n 和 plot 窗口变元。 *nmin=0 和 nmax=50 是计算 50 年后树林的规模。*	[◆][WINDOW] 0 ▼ 5 0 ▼ 1 ▼ 1 ▼	[◆][WINDOW] 0 ▽ 5 0 ▽ 1 ▽ 1 ▽	nmin=0. nmax=50. plotStrt=1. plotStep=1. xmin=0. xmax=50. xscl=10. ymin=0. ymax=6000. yscl=1000.
5. 将窗口变元 x 和 y 设定成与本例子相应的数值。	0 ▼ 5 0 ▼ 1 0 ▼ 0 ▼ 6 0 0 0 ▼ 1 0 0 0	0 ▽ 5 0 ▽ 1 0 ▽ 0 ▽ 6 0 0 0 ▽ 1 0 0 0	
6. 显示作图屏幕。	[◆][GRAPH]	[◆][GRAPH]	
7. 选择 Trace。移动光标来进行逐年追踪。多少年(nc)才能使树木数量(yc)稳定? *从 nc=0 开始追踪。* *nc 是年数。* *xc = nc(是因为 nc 已在 x 轴上画出。)* *yc = u1 (n)，树木的量就在 nc 年处。*	[F3] ▶ 和 ◀ (如有需要的话)	[F3] ▷ 和 ◁ (如有需要的话)	

三维作图

作出 3D (三维)图形方程 z(x,y) = ($x^3y - y^3x$) / 390 的图，通过光标交互地改变控制您的观察角的窗口变元值来令图形动态化，然后在不同图形格式下观察该图形。

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 显示 MODE 对话框，对于 Graph 模式请选择 3D。	MODE ▶ 5 ENTER	MODE ▷ 5 ENTER	
2. 显示并清空 Y= 编辑器。 定义 3D 方程 z1(x,y) = (x^3 y − y^3 x) / 390。 *注意击键中使用的隐含乘法。*	◆ [Y=] F1 8 ENTER ENTER (X ^ 3 Y − Y ^ 3 X) ÷ 3 9 0 ENTER	◆ [Y=] F1 8 ENTER ENTER (X ^ 3 Y − Y ^ 3 X) ÷ 3 9 0 ENTER	
3. 改变图形格式来显示和标明坐标轴，同时设定 Style = WIRE FRAME。 *您可使任何格式样式的图形动态化，但 WIRE FRAME 是最快速的。*	◆ [I] ▼▶ 2 ▼▶ 2 ▼▶ 1 ENTER	◆ F ▼ ▷ 2 ▼ ▷ 2 ▼ ▷ 1 ENTER	
4. 选择 ZoomStd 观察正方形。这样可自动作出方程的图形。 *在方程计算后(作图前)，屏幕左上角会显示出信息 "valuation percentages"。*	F2 6	F2 6	
备注: *如您已使用 3D 作图，那么图形就会呈扩展外观。当您将图形动态化时，屏幕就会自动返回正常外观。(除动态化外，您可在扩展和正常外观下进行同样的操作。)*	⊠ (按下 ⊠ 来在扩展和正常外观之间切换)	⊠ (按下 ⊠ 来在扩展和正常外观之间切换)	
5. 通过减少 eyeϕ 窗口变元值来使图形动态化。 ▼ 或 ▲ 会影响 eyeθ *和* eyeψ，*但较* eyeϕ *值的延伸少。* *欲使图形连续动态化，可按下并按住光标约一秒钟然后松开即可。要停止动态化，可按下* ENTER。	▼▼▼▼ ▼▼▼▼	▽▽▽▽ ▽▽▽▽	

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
6. 将图形返回原始方向，然后沿观察轨道环绕图形来移动观察角。	0 (零，不是字母 O) ◀◀◀	0 (零，不是字母 O) ◀◀◀	
7. 沿 x 轴、y 轴及 z 轴检视图形。	X	X	
图形沿 y 轴和 x 轴具有相同的形状。	Y	Y	
	Z	Z	
8. 返回原始方向。	0	0	
9. 以不同的格式样式显示图形。	I (按下 I 来从每一样式到下一样式切换)	F (按下 F 来从每一样式到下一样式切换)	

隐藏表面

轮廓层次
(需要更多的时间来计算轮廓)

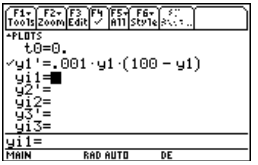
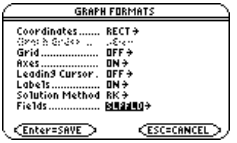
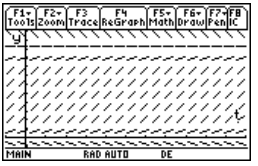
线条和轮廓

线框

备注: 您也可通过 GRAPH FORMATS 对话框(F1 9 或 **TI-89:** ◆ I **TI-92 Plus:** ◆ F)将图形作为隐含图显示。如您按下 **TI-89:** I **TI-92 Plus:** F 来切换样式，隐含图将不会显示出来。

微分方程作图

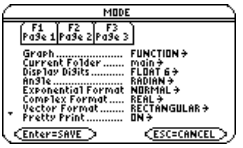
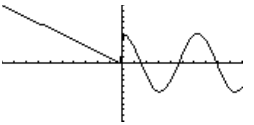
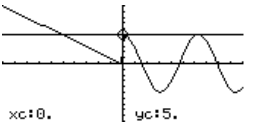
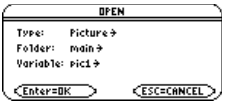
作出逻辑一阶微分方程 y' = .001y∗(100−y)的解。从仅画出斜率场开始，然后在 Y=编辑器中输入起始条件并在图形屏幕中交互。

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 显示 MODE 对话框，对于 Graph 模式请选择 DIFF EQUATIONS。	[MODE] ▷ 6 [ENTER]	[MODE] ▷ 6 [ENTER]	
2. 显示和清空 Y= 编辑器。定义微分方程: y1'(t)=.001y1∗(100−y1) *按下 [×] 来输入上面示出的 ∗。请勿在变元与括号之间使用蕴含乘法。如使用了，它将会被视作功能调用来处理。保留起始条件 yi1 空白。*	[◆] [Y=] [F1] 8 [ENTER] [ENTER] . 0 0 1 Y 1 [×] [(] 1 0 0 [−] Y 1 [)] [ENTER]	[◆] [Y=] [F1] 8 [ENTER] [ENTER] . 0 0 1 Y 1 [×] [(] 1 0 0 [−] Y 1 [)] [ENTER]	 重要事项: 选定了 y1' 时，TI-89 / TI-92 Plus 将会作出 y1 解曲线，并非导数 y1'。
3. 显示 GRAPH FORMATS 对话框，设定 Axes = ON、Labels = ON、Solution Method = RK 和 Fields = SLPFLD。 ***重要事项:** 欲作一个微分方程，Fields 必须设定成 SLPFLD 或 FLDOFF。如 Fields=DIRFLD，当您作图时就会出现错误。*	[◆] [I] ▽ ▽ ▷ 2 ▽ ▽ ▷ 2 ▽ ▷ 1 ▽ ▷ 1 [ENTER]	[◆] F ▽ ▽ ▷ 2 ▽ ▽ ▷ 2 ▽ ▷ 1 ▽ ▷ 1 [ENTER]	
4. 显示窗口编辑器并如右方所示设定窗口变元。	[◆] [WINDOW] 0 ▽ 1 0 ▽ . 1 ▽ 0 ▽ [(−)] 1 0 ▽ 1 1 0 ▽ 1 0 ▽ [(−)] 1 0 ▽ 1 2 0 ▽ 1 0 ▽ 0 ▽ . 0 0 1 ▽ 2 0	[◆] [WINDOW] 0 ▽ 1 0 ▽ . 1 ▽ 0 ▽ [(−)] 1 0 ▽ 1 1 0 ▽ 1 0 ▽ [(−)] 1 0 ▽ 1 2 0 ▽ 1 0 ▽ 0 ▽ . 0 0 1 ▽ 2 0	t0=0. tmax=10. tstep=.1 tplot=0. xmin=-10. xmax=110. xscl=10. ymin=-10. ymax=120. yscl=10. ncurves=0. diftol=.001 fldres=20.
5. 显示作图屏。 *由于您没有指定起始条件，故只能画出斜率场(如通过 GRAPH FORMATS 对话框中 Fields=SLPFLD 指定的)。*	[◆] [GRAPH]	[◆] [GRAPH]	

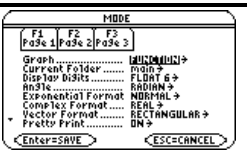
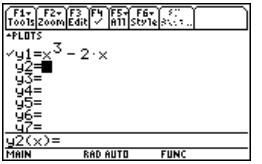
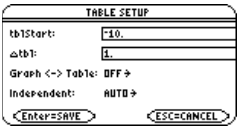
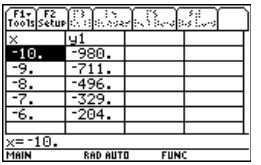
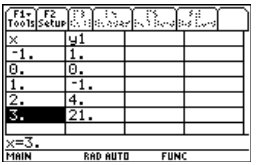
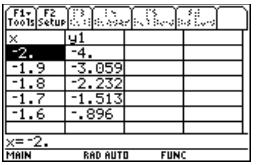
步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
6. 返回 Y= 编辑器并输入一个起始条件: yi1=10	◆ [Y=] ENTER 1 0 ENTER	◆ [Y=] ENTER 1 0 ENTER	F1 Tools F2 Zoom F3 Edit F4 ✓ F5 All F6 Style *PLOTS t0=0. ✓y1'=.001·y1·(100 − y1) yi1=10
7. 返回作图屏。 *在 Y=编辑器中输入的起始条件通常是在 t0。图形从起始条件开始并向右画出，然后再向左画。*	◆ [GRAPH]	◆ [GRAPH]	起始条件用一个圆标记出。
8. 返回 Y=编辑器并改变 yi1: yi1={10,20}	◆ [Y=] ⊕ ENTER 2nd [{] 1 0 , 2 0 2nd [}] ENTER	◆ [Y=] ⊕ ENTER 2nd [{] 1 0 , 2 0 2nd [}] ENTER	F1 Tools F2 Zoom F3 Edit F4 ✓ F5 All F6 Style *PLOTS t0=0. ✓y1'=.001·y1·(100 − y1) yi1={10 20}
9. 返回作图屏。	◆ [GRAPH]	◆ [GRAPH]	
10. 欲交互地选择一个起始条件，可按下: **TI-89:** 2nd [F8] **TI-92 Plus:** F8 当出现提示时，输入 t=40 和 y1=45。 *当您交互地选择一个起始条件时，您可给 t 指定一个在 Y=编辑器或窗口编辑器中输入的 t0 以外的值。* *按下 **TI-89:** 2nd [F8]* ***TI-92 Plus:** F8 后，作为输入 t 和 y1 的替代，您可将光标移到屏幕的一点上并按下 ENTER。* *您可使用 F3 来为 Y=编辑器中指定的起始条件追踪曲线。然而，您不能为交互选择的起始条件追踪曲线。*	2nd [F8] 4 0 ENTER 4 5 ENTER	F8 4 0 ENTER 4 5 ENTER	Initial Conditions=? t=40 y1=45

附加作图提示

在主屏幕上，作出已定义的分段函数: y = −x (当 x < 0) 和 y = 5 cos(x) (当 x ≥ 0)。 画一条与余弦曲线顶部相交的水平线，然后将显示出的图形保存成图象。

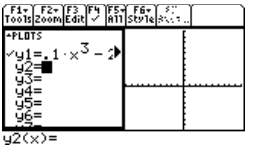
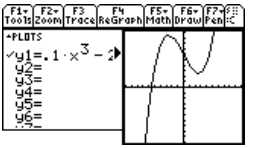
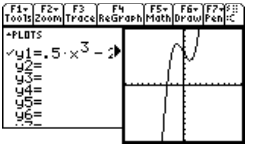
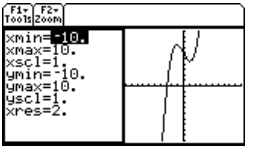
步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 显示 MODE 对话框，对于 Graph 模式，选择 FUNCTION。 对于 Angle 模式，请选择 RADIAN。	MODE ▶ 1 ▼▼▼ ▶ 1 ENTER	MODE ▶ 1 ▼▼▼ ▶ 1 ENTER	
2. 显示主屏幕，利用 **Graph** 命令和 **when** 功能来指定已定义的分段函数。 *F4 2 可从 Other 工具栏菜单中选择 **Graph** 并自动加入一个空格。*	HOME F4 2 2nd [a-lock] W H E N alpha (X 2nd [<] 0 , (-) X , 5 × 2nd [COS] X))	◆ [HOME] F4 2 W H E N (X 2nd [<] 0 , (-) X , 5 × COS X))	Graph when(x<0,-x,5*cos(x))
3. 执行 **Graph** 命令，这样会自动显示作图屏。 *作图会采用当前的窗口变元，这是被假定成此例子中的标准值(F2 6)。*	ENTER	ENTER	
4. 画出一条交余弦曲线顶部的水平线。 *计算器保持在 "line" 模式，直至您选择了不同的操作或按下了 ESC。*	2nd [F7] 5 ▲ (直至线段定位为止) ENTER	F7 5 ▲ (直至线段定位为止) ENTER	
5. 将图形保存为图象，利用 PIC1 作为该图象的变元名称。 *应确认设定了 Type = Picture，默认设置是 GDB。*	F1 2 ▶ 2 ▼ ▼ P I C alpha 1 ENTER ENTER	F1 2 ▶ 2 ▼ ▼ P I C 1 ENTER ENTER	
6. 清除画出的水平线。 *您也可按下 F4 来重作图形。*	2nd [F6] 1	F6 1	
7. 打开保存的图象变元来重新显示带有水平线的图形。 *应确认设定了 Type = Picture，默认设置是 GDB。*	F1 1 ▶ 2 (如不显示出来，在设定 Variable = pic1) ENTER	F1 1 ▶ 2 (如不显示出来，在设定 Variable = pic1) ENTER	

计算函数 $y=x^3 - 2x$ 在 ⁻10 与 10 间每一个整数的函数值? 有多少符号改变了? 发生在什么地方?

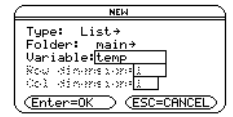
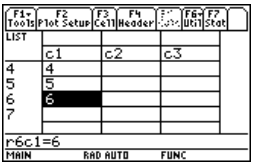
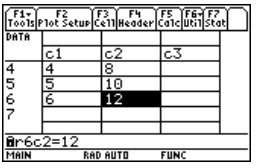
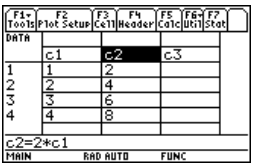
步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 显示 MODE 对话框。对于 Graph 模式，请设定 FUNCTION。	[MODE] ▶ 1 [ENTER]	[MODE] ▷ 1 [ENTER]	
2. 显示并清空 Y= 编辑器。然后定义 y1(x) = $x^3 - 2x$。	◆ [Y=] [F1] 8 [ENTER] [ENTER] X [^] 3 [−] 2 X [ENTER]	◆ [Y=] [F1] 8 [ENTER] [ENTER] X [^] 3 [−] 2 X [ENTER]	
3. 将表格参数设定成: tblStart = ⁻10 Δtbl = 1 Graph <-> Table = OFF Independent = AUTO	◆ [TblSet] [(-)] 1 0 ▼ 1 ▼ ▶ 1 ▼ ▶ 1 [ENTER]	◆ [TblSet] [(-)] 1 0 ▽ 1 ▽ ▷ 1 ▽ ▷ 1 [ENTER]	
4. 显示表格屏。	◆ [TABLE]	◆ [TABLE]	
5. 滚动表格，注意， y1 在 x = ⁻1、1 和 2 处会改变符号。 *欲一次滚动一页，可使用* [2nd] ▼ *和* [2nd] ▲。	▼ 和 ▲ (需要的话)	▽ 和 △ (需要的话)	
6. 通过改变以下的表格参数来放大 x = ⁻2 和 x = ⁻1 之间符号变更之处: tblStart = ⁻2 Δtbl = .1	[F2] [(-)] 2 ▼ . 1 [ENTER] [ENTER]	[F2] [(-)] 2 ▽ . 1 [ENTER] [ENTER]	

拆分屏幕

对屏幕进行拆分，以显示 Y= 编辑器和作图屏，然后研究多项式系数变化时，多项式图形的性态。

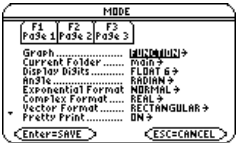
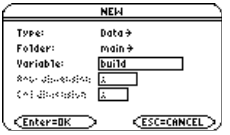
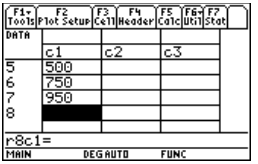
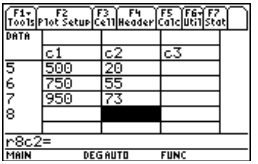
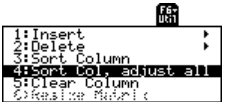
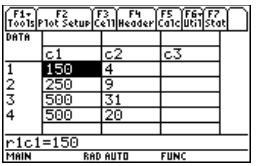
步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 显示 MODE 对话框。对于 Graph 模式，请选择 FUNCTION。 对于 Split Screen，请选择 LEFT-RIGHT。 对于 Split 1 App，请选择 Y= Editor。 对于 Split 2 App 请选择 Graph。	[MODE] ▶ 1 [F2] ▶ 3 ▼▶ 2 ▼▶ 4 [ENTER]	[MODE] ▷ 1 [F2] ▷ 3 ▼ ▷ 2 ▼ ▷ 4 [ENTER]	
2. 清空 Y= 编辑器并关闭统计数据图。定义 y1(x) = .1x³–2x+6。 *环绕 Y= 编辑器的粗边框表明编辑器为活动的。当编辑器活动时，输入行是屏幕的全长度。*	[F1] 8 [ENTER] [F5] 5 [ENTER] . 1 X [^] 3 [−] 2 X [+] 6 [ENTER]	[F1] 8 [ENTER] [F5] 5 [ENTER] . 1 X [^] 3 [−] 2 X [+] 6 [ENTER]	
3. 选择 ZoomStd 观察窗，这样会切换到图形屏幕并作出函数图形。 *粗边框现在是环绕图形屏幕。*	[F2] 6	[F2] 6	
4. 切换到 Y= 编辑器，然后编辑 y1(x) 来将 .1x³ 改变成 .5x³。 [2nd] [⊞] *是* [APPS] *的第二功能。* *粗边框是环绕 Y= 编辑器。*	[2nd] [⊞] ▲ [ENTER] ◀ ▶ ▶ ← 5 [ENTER]	[2nd] [⊞] ↑ [ENTER] ← → → ← 5 [ENTER]	
5. 切换到图形屏幕，重新作出编辑的函数的图形。 *粗边框是环绕作图屏。*	[2nd] [⊞]	[2nd] [⊞]	
6. 切换到 Y = 编辑器，然后在里面的位 置打开窗口编辑器。	[2nd] [⊞] ◆ [WINDOW]	[2nd] [⊞] ◆ [WINDOW]	
7. 打开主屏幕，然后退出到全屏幕显示的主屏幕。	[2nd] [QUIT] [2nd] [QUIT]	[2nd] [QUIT] [2nd] [QUIT]	

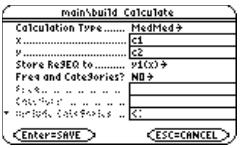
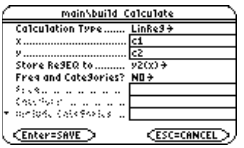
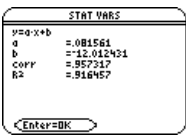
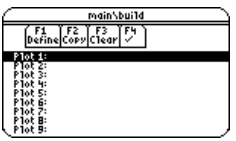
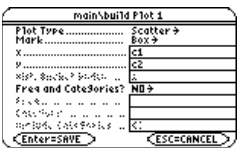
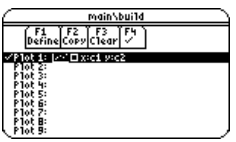
使用数据 / 矩阵编辑器来创建单列数组变元，然后添加信息的第二列。注意，数组变元(可仅为单列)会自动转换成数据变元(可为多列)。

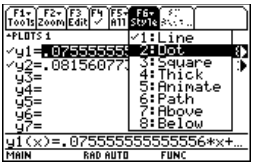
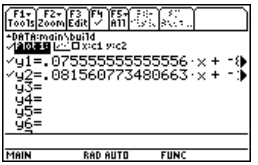
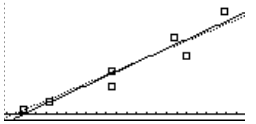
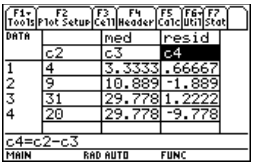
步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 启动数据 / 矩阵编辑器来创建一个名为 TEMP 的新数组变元。	APPS 6 3 ▶ 3 ▼ ▼ T E M P ENTER ENTER	APPS 6 3 ▶ 3 ▼ ▼ T E M P ENTER ENTER	
2. 输入数字的列，然后将光标上移一个单元格(仅是为了使单元格的突出显示数值显示在输入行中)。 *显示在左上角的 LIST (数组)表明一个数组变元。* *您可使用 ▼ 而不使用 ENTER 来在单元格中输入信息。*	1 ENTER 2 ENTER 3 ENTER 4 ENTER 5 ENTER 6 ENTER ▲	1 ENTER 2 ENTER 3 ENTER 4 ENTER 5 ENTER 6 ENTER ▲	
3. 移到列 2 并定义该列的页眉，使其成 为列 1 的数值的两倍。 *显示在左上角的 DATA (数据)表明该数组变元已转换成数据变元。*	▶ F4 2 × alpha C 1 ENTER	▶ F4 2 × C 1 ENTER	 🔒 表明单元格于已定义的列中。
4. 移到列 2 的页眉单元格，在输入行中显示其定义值。 *当光标位于页眉单元格时，您无须按下 F4 来定义它，只要键入表达式即可。*	2nd ▲ ▲	2nd ▲ ▲	
5. 转到主屏幕，然后返回当前的变元上。	HOME APPS 6 1	◆ HOME APPS 6 1	
6. 清除变元的内容。 *简单地清除数据而不将数据变元转换回数组变元。*	F1 8 ENTER	F1 8 ENTER	

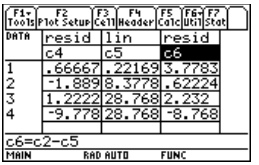
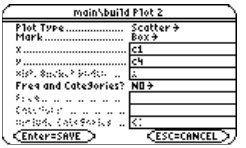
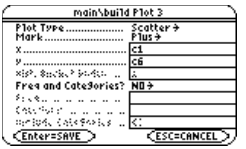
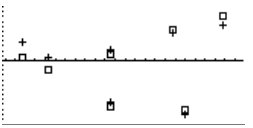
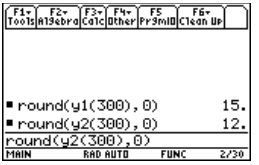
提示: 如您不需要保存当前的变元，可将其作 "*便笺簿*" 使用，下次当您需要一个变元作为临时数据时，可清除当前的变元并重新使用之。这样可让您输入临时数据而不需每次创建耗用计算器内存的新变元。

基于一个拥有七个城市的例子，输入把人口与 12 层以上的建筑物的数量联系起来的数据。 使用中位数－中位数和线性回归方程来进行计算、求解和绘图，以将数据拟合。对于每一回归方程，预测一个拥有 30 万人口的城市需要多少栋 12 层以上的建筑物。

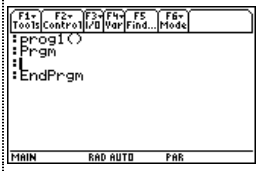
步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 显示 MODE 对话框。 对于 Graph 模式，请选择 FUNCTION。	MODE ▶ 1 ENTER	MODE ▷ 1 ENTER	
2. 显示数据 / 矩阵编辑器并创建名为 BUILD 的新数据变元。	APPS 6 3 ▼ ▼ B U I L D ENTER ENTER	APPS 6 3 ▽ ▽ B U I L D ENTER ENTER	
3. 使用下面例子的数据，在列 1 中输入人口数量。 人口(千) 12 层以上建筑物 150 4 500 31 800 42 250 9 500 20 750 55 950 73	1 5 0 ENTER 5 0 0 ENTER 8 0 0 ENTER 2 5 0 ENTER 5 0 0 ENTER 7 5 0 ENTER 9 5 0 ENTER	1 5 0 ENTER 5 0 0 ENTER 8 0 0 ENTER 2 5 0 ENTER 5 0 0 ENTER 7 5 0 ENTER 9 5 0 ENTER	
4. 将光标移到列 2 的行 1(r1c2)，然后输入对应建筑物的数量。 *⊞ ⊖ 可将光标移到页的顶部。键入单元格数据后，您可按下 ENTER 或 ⊖ 来输入数据，并将光标下移一个单元格。按下 ⊖ 可输入数据和将光标上移一个单元格。*	▶ ⊞ ⊖ 4 ENTER 3 1 ENTER 4 2 ENTER 9 ENTER 2 0 ENTER 5 5 ENTER 7 3 ENTER	▷ 2nd ⊖ 4 ENTER 3 1 ENTER 4 2 ENTER 9 ENTER 2 0 ENTER 5 5 ENTER 7 3 ENTER	
5. 将光标移到列 1 的行 1(r1c1)，将人口按递增顺序排序。 *这样可对列 1 排序，并调整其它列，使它们象列 1 一样保留相同的顺序。对于保持数据列之间的关系，这是至关重要的。欲对列 1 进行排序，光标可放在列 1 的任何位置上。此例子是当成您按下了* ***TI-89:*** *⊞ ⊖* ***TI-92 Plus:*** *2nd ⊖，所以，您可看到所有的数据。*	◀ ⊞ ⊖ 2nd F6 4	◁ 2nd ⊖ F6 4	 

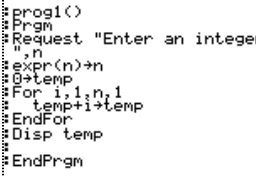
步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
6. 显示 Calculate 对话框，设定: Calculation Type = MedMed x = C1 y = C2 Store RegEQ to = y1(x)	F5 ▶ 7 ▼ C alpha 1 ▼ alpha C 2 ▼ ▶ ▼ ENTER	F5 ▷ 7 ↓ C 1 ↓ C 2 ↓ ▷ ↓ ENTER	
7. 执行计算，以显示中位数－中位数回 归方程。 *由于此方程已在 Calculate 对话框中定义，所以，方程会存储在 y1(x) 内。*	ENTER	ENTER	
8. 关闭 STAT VARS 屏幕。这时会显示出数据 / 矩阵编辑器。	ENTER	ENTER	
9. 显示 Calculate 对话框，设定: Calculation Type = LinReg x = C1 y = C2 Store RegEQ to = y2(x)	F5 ▶ 5 ▼ ▼ ▼ ▶ ▼ ENTER	F5 ▷ 5 ↓ ↓ ↓ ▷ ↓ ENTER	
10. 执行计算，以显示 LinReg 线性回归方程。 *此方程会存储在 y2(x) 内。*	ENTER	ENTER	
11. 关闭 STAT VARS 屏幕。这时会显示出数据 / 矩阵编辑器。	ENTER	ENTER	
12. 显示 Plot Setup 屏幕。 *由于默认设置的原因， Plot 1 为突出显示。* F3 *可让您清除突出显示的 Plot 设置。*	F2	F2	
13. 将 Plot 1 定义成: Plot Type = Scatter Mark = Box x = C1 y = C2 *请注意此对话框与 Calculate 对话框之间相似之处。*	F1 ▶ 1 ▼ ▶ 1 ▼ C alpha 1 ▼ alpha C 2	F1 ▷ 1 ↓ ▷ 1 ↓ C 1 ↓ C 2	
14. 保存绘图定义值并返回 Plot Setup 屏幕。 *注意 Plot 1 定义值的速记符号。*	ENTER ENTER	ENTER ENTER	

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
15. 显示 Y= 编辑器，对于 y1(x)，中位数－中位数回归方程会将显示样式设定到 Dot (点)。 ***备注:*** *您也许要将光标移到 y1 上，这取决于 Y= 编辑器上一次所含的内容。* *PLOTS 1 位于屏幕顶部是表明 Plot 1 已经选定。* *注意，当回归方程存储时， y1(x) 和 y2(x) 即为选定。*	◆ [Y=] 2nd [F6] 2	◆ [Y=] F6 2	
16. 向上滚动，使 Plot 1 突出显示。 *显示出的速记定义与 Plot Setup 屏上的相同。*	⊕	⊕	
17. 使用 ZoomData 来作 Plot 1 以及 y1(x) 和 y2(x) 回归方程的图。 *ZoomData 是检查所有选定的统计图的数据 并调整观察窗，以使所有点能够包含在内。*	F2 9	F2 9	
18. 返回数据 / 矩阵编辑器的当前工作期。	APPS 6 1	APPS 6 1	
19. 输入列 3 的标题，将列 3 的页眉定义成由中位数－中位数线预测出的数值。 *欲输入一个标题，光标必须在列的最顶部将单元格突出显示。* F4 *可让您从列的任何位置定义页眉。当光标位于页眉单元格时，无须按下* F4 *。*	⊕⊕⊕⊕ 2nd [a-lock] M E D alpha ENTER F4 Y 1 (alpha C 1) ENTER	⊕⊕⊕⊕ M E D ENTER F4 Y 1 (C 1) ENTER	
20. 输入列 4 的标题，将列 4 的页眉定义成中位数－中位数的离差 (观察到的与预测到的值之间的差)。	⊕⊕ 2nd [a-lock] R E S I D alpha ENTER alpha C 2 − alpha C 3 ENTER	⊕⊕ R E S I D ENTER F4 C 2 − C 3 ENTER	
21. 输入列 5 的标题，将列 5 的页眉定义成由 LinReg 线性回归线预测出的数值。	⊕⊕⊕ 2nd [a-lock] L I N alpha ENTER F4 Y 2 (alpha C 1) ENTER	⊕⊕ L I N ENTER F4 Y 2 (C 1) ENTER	

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
22. 输入列 6 的标题，将列 6 的页眉定义成 LinReg 的离差。	▶ ▲ [2nd] [a-lock] R E S I D [alpha] [ENTER] [F4] [alpha] C 2 [−] [alpha] C 5 [ENTER]	▶ ▲ R E S I D [ENTER] [F4] C 2 [−] C 5 [ENTER]	
23. 显示 Plop Setup 绘图设置屏幕并删除 Plot 1。	[F2] [F4]	[F2] [F4]	
24. 把 Plot 2 突出显示并将其定义成: Plot Type = Scatter Mark = Box x = C1 y = C4 (中位数－中位数离差)	▼ [F1] ▼ ▼ C [alpha] 1 ▼ [alpha] C 4 [ENTER] [ENTER]	▼ [F1] ▼ ▼ C 1 ▼ C 4 [ENTER] [ENTER]	
25. 把 Plot 3 突出显示并将其定义成: Plot Type = Scatter Mark = Plus x = C1 y = C6 (LinReg 离差)	▼ [F1] ▼ ▶ 3 ▼ C [alpha] 1 ▼ [alpha] C 6 [ENTER] [ENTER]	▼ [F1] ▼ ▶ 3 ▼ C 1 ▼ C 6 [ENTER] [ENTER]	
26. 显示 Y= 编辑器并关闭所有 y(x) 功能。 *在 [F5] 上选择* 3:Functions Off，*不要选* 1:All Off。 *Plots 2 和 3 仍然为选定。*	◆ [Y=] [F5] 3	◆ [Y=] [F5] 3	
27. 使用 ZoomData 来作离差图。 □ *标明中位数－中位数离差;* + *标明 LinReg 离差。*	[F2] 9	[F2] 9	
28. 显示主屏幕。	[HOME]	◆ [HOME]	
29. 使用中位数－中位数 (y1(x)) 和 LinReg (y2(x)) 回归方程来计算 x = 300 (30万人)的值。 **round** (舍入)*功能* ([2nd] [MATH] 13) *可确保结果能以整数显示建筑物的数量。* *计算完第一个结果后，编辑输入行来将 y1 改成 y2。*	[2nd] [MATH] 1 3 Y 1 [(] 3 0 0 [)] [,] 0 [)] [ENTER] ▶ ◀ ◀ ◀ ◀ ◀ ◀ ◀ ◀ [←] 2 [ENTER]	[2nd] [MATH] 1 3 Y 1 [(] 3 0 0 [)] [,] 0 [)] [ENTER] ▶ ◀ ◀ ◀ ◀ ◀ ◀ ◀ ◀ [←] 2 [ENTER]	

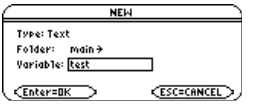
编写一个程序来提示用户输入一个整数，并把从 1 到该输入了的整数之间的所有整数相加，并将结果显示出来。

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 在程序编辑器上启动新的程序。	APPS 7 3	APPS 7 3	
2. 键入 Prog1 (不含空格)作为新程序的变元名称。	⊖⊖ P R O G alpha 1	⊖⊖ P R O G 1	
3. 显示新程序的 “template”(模板)，这时，程序名称、 **Prgm** 和 **EndPrgm** 就会自动显示出来。 *当在输入框(如 Variable)内键入后，您必须按下 ENTER 两次。*	ENTER ENTER	ENTER ENTER	

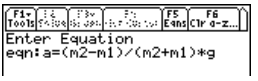
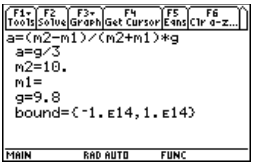
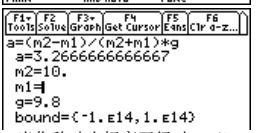
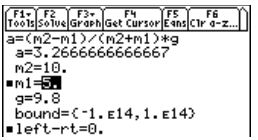
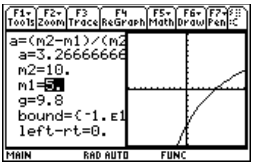
4. 键入以下的程序行。
`Request "Enter an integer",n`
显示一个 “Enter an integer”(请输入一个整数)的提示对话框，等候用户输入一个数值并将其存储(作为字符串)在 n 变元中。
`expr(n)→n`
将字符串转换成为数字表达式。
`0→temp`
创建名为 temp 的变元并将其初始化为 0。
`For i,1,n,1`
启动基于变元 i 的 For 循环。第一次的循环是直到结尾，i = 1。在循环的结尾，i 增加 1。循环继续进行，直至 i > n。
`temp+i→temp`
将 i 的当前值添加入 temp 中。
`EndFor`
标记 For 循环的结尾。
`Disp temp`
显示 temp 的最终值。
	如图所示键入程序行。
	在每一行的结尾处按下 ENTER。
		如图所示键入程序行。
		在每一行的结尾处按下 ENTER。
			

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
5. 转到主屏幕，在输入程序名称后，再键入一个括号对。 *您必须包含 ( )，即使当程序没有变数之时。* *程序会显示一个带有在程序中指定的提示的对话框。*	[HOME] [2nd] [a-lock] P R O G [alpha] 1 [(] [)] [ENTER]	[◆] [HOME] P R O G 1 [(] [)] [ENTER]	prog1()
6. 在显示出来的对话框中键入 5。	5	5	F1 Tools F2 Algebra F3 Calc F4 Other F5 PrgmIO F6 Clean Up Enter an integer: 5 Enter=OK ESC=CANCEL prog1() MAIN RAD AUTO PAR 0/30
7. 继续执行程序，**Disp 显示**命令可在程序 I/O (输入 / 输出)屏幕上显示结果。 *结果就是从 1 到 5 的整数的和。* *尽管程序 I/O 屏幕看起来类似主屏幕，但它仅是用于程序的输入和输出而已。您不能在程序 I/O 屏幕上执行计算。*	[ENTER] [ENTER]	[ENTER] [ENTER]	F5 PrgmIO 1 3 6 10 15 自其它程序的输出内容将仍然显示在屏幕上。 整数 5 的结果。 MAIN RAD AUTO PAR 5/30
8. 离开程序 I/O 屏并返回主屏幕。 *您也可按下* [ESC]、[2nd] [QUIT] *或* ***TI-89:*** [HOME] ***TI-92 Plus:*** [◆] [HOME] 来返回主屏幕。	[F5]	[F5]	F1 Tools F2 Algebra F3 Calc F4 Other F5 PrgmIO F6 Clean Up ■ prog1() Done prog1() MAIN RAD AUTO PAR 1/30

启动文本编辑器的一个新工作期，然后使用文本编辑器，键入您想要的文本内容来进行练习。当您键入时，练习会移动文本光标并纠正您可能输入的错误。

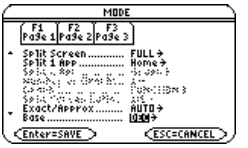
步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 在文本编辑器上启动新工作期。	[APPS] 8 3	[APPS] 8 3	
2. 创建一个名为 TEST 的文本变元，该变元会将您在新工作期内输入的任何文本自动存储起来。 *使用象 NEW (新建)对话框的默认显示一样的 MAIN (主)文件夹。* *当在输入框(如 Variable)内键入后，您必须按下 [ENTER] 两次。*	⊖ T E S T [ENTER] [ENTER]	⊖ T E S T [ENTER] [ENTER]	
3. 键入一些示例文本。 • *欲键入单个大写字母，可按下 [↑] 并按下字母。* *仅对 **TI-89**:* – *欲键入空格，可按下 [alpha] [␣] ([(-)] 键的 alpha 功能)。* – *欲键入句号，可按下 [alpha] 来关闭 alpha-lock，再按下 [.]，然后按下 [2nd] [a-lock] 以重新打开 alpha-lock 。* *练习是使用以下方式来编辑您的文本的:* • *用光标控制盘移动文本光标。* • *用 [←] 或 [♦] [DEL] 来重复删除光标左边或右边的字符。*	[2nd] [a-lock] 键入您要的任何内容	键入您要的任何内容	
4. 离开文本编辑器并显示主屏幕。 *您的文本工作期会象您所键入的一样得到自动存储。所以，在退出文本编辑器之前，您无须手控保存工作期。*	[HOME]	[♦] [HOME]	
5. 返回文本编辑器的当前工作期。	[APPS] 8 1	[APPS] 8 1	
6. 注意，工作期会与您离开时完全相同的样式显示出来。			

思考方程式 a=(m2− m1)/(m2+m1)∗g，其中已知的数值为 m2=10 和 g=9.8。如您假设 a=1/3 g，求出 m1 的值。

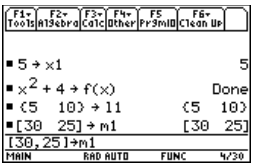
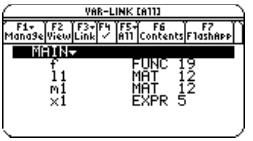
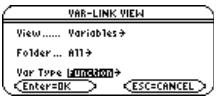
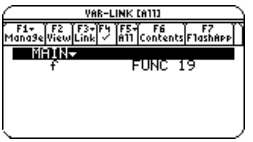
步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 显示数值解答器。	APPS 9	APPS 9	
2. 输入方程式。 *当您按下 ENTER 或 ⊖ 时，屏幕会列出方程中用到的变元。*	alpha A = (alpha M 2 − alpha M 1) ÷ (alpha M 2 + alpha M 1) × alpha G ENTER	A = (M 2 − M 1) ÷ (M 2 + M 1) × G ENTER	
3. 为每一变元输入数值，未知变元 m1 除外。 *首先定义 m2 和 g，然后再定义 a。(在按照 g 来定义 a 前，您必须先定义 g。)接受 bound 的默认值。如变元已预先定义了的话，它的值会成为默认值显示。*	⊖ 1 0 ⊖ ⊖ 9 . 8 ⊕ ⊕ ⊕ alpha G ÷ 3	⊖ 1 0 ⊖ ⊖ 9 . 8 ⊕ ⊕ ⊕ G ÷ 3	
4. 将光标移到未知的变元 m1 上。 *作为任选项，您可以为 m1 输入初始猜测值。甚至如果您为所有变元输入数值，数值解答器就会将光标标记出的那个变元解出来。*	⊖ ⊖	⊖ ⊖	 当您移动光标离开行时，g/3 就会被计算出来。
5. 解出未知变元。 *欲检查方程解的准确度，方程的左边和右边会被分开计算。差别会以 left− rt 显示。如方程正确，那么 left− rt=0。*	F2	F2	 ▪ 标记计算出的值。
6. 利用 ZoomStd 观察窗作出方程的图形。 *图形在拆分的屏幕中显示，您可通过追踪、缩放等来探讨图形。*	F3 3	F3 3	
7. 返回数值解答器并退出拆分屏幕模式。 *您可按下 ENTER 或 ⊖ 来重新显示变元列表。*	2nd [+□] F3 2	2nd [+□] F3 2	被光标标出的变元(未知变元 m1)是在 x 轴，而 left−rt 是在 y 轴。

数字基

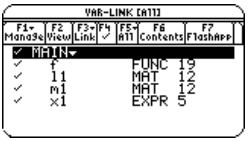
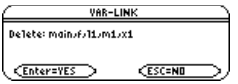
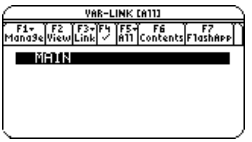
计算 10 二进制(基 2) + F 十六进制(基 16) + 10 小数(基 10)，然后使用 ▸ 算子来将整数从一种基转换成另一种。最后，查看改变 Base 模式会如何影响显示的结果。

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 显示 MODE 对话框(第 2 页)。对于 Base 模式，选择 DEC 作为默认数字基。 *整数结果是按照 Base 模式显示出来。分数和浮点小数结果一直是以小数形式显示。*	MODE F2 (利用 ⊖ 来移到 Base 模式) ⊕ 1 ENTER	MODE F2 (利用 ⊖ 来移到 Base 模式) ⊖ 1 ENTER	
2. 计算 0b10+0hF+10。 *欲输入二进制或十六进制数字，您必须使用 0b 或 0h 前缀(零和字母 B 或 H)。否则，输入项会被当作小数处理。*	0 alpha B 1 0 + 0 2nd [a-lock] H F alpha + 1 0 ENTER	0 B 1 0 + 0 H F + 1 0 ENTER	0b10 + 0hF + 10 27 0b10+0hf+10 MAIN RAD AUTO FUNC 1/30 **重要事项:** 0b 或 0h 前缀是零，并非字母 O，后面跟随有 B 或 H。
3. 往结果中加 1 并将其转换成十进制。 *2nd [▸] 是显示 ▸ 换算算子。*	+ 1 2nd [▸] 2nd [a-lock] B I N alpha ENTER	+ 1 2nd [▸] B I N ENTER	
4. 往结果中加 1 并将其转换成十六进制。	+ 1 2nd [▸] 2nd [a-lock] H E X alpha ENTER	+ 1 2nd [▸] H E X ENTER	
5. 往结果中加 1 并让其保持默认的小数基。	+ 1 ENTER	+ 1 ENTER	0b10 + 0hF + 10 27 (27 + 1)▸Bin 0b11100 (0b11100 + 1)▸Hex 0h1D 0h1D + 1 30 ans(1)+1 MAIN RAD AUTO FUNC 4/30 结果是以 0b 或 0h 前缀识别基的。
6. 将 Base 模式换成 HEX。 *当 Base = HEX 或 BIN 时，结果的数量会被限制在一定的大小。*	MODE F2 (利用 ⊖ 来移到 Base 模式) ⊕ 2 ENTER	MODE F2 (利用 ⊖ 来移到 Base 模式) ⊖ 2 ENTER	
7. 计算 0b10+0hF+10。	0 alpha B 1 0 + 0 2nd [a-lock] H F alpha + 1 0 ENTER	0 B 1 0 + 0 H F + 1 0 ENTER	0b10 + 0hF + 10 0h1B 0b10+0hf+10 MAIN RAD AUTO FUNC 1/30
8. 将 Base 模式换成 BIN。	MODE F2 (利用 ⊖ 来移到 Base 模式) ⊕ 3 ENTER	MODE F2 (利用 ⊖ 来移到 Base 模式) ⊖ 3 ENTER	
9. 重新输入 0b10+0hF+10。	ENTER	ENTER	0b10 + 0hF + 10 0h1B 0b10 + 0hF + 10 0b11011 0b10+0hf+10 MAIN RAD AUTO FUNC 2/30

为众多的变元数据类型赋值。利用 VAR-LINK 屏查看已定义变元的列表，然后将一个变元移到用户数据档案内存中，并分析您可以用哪种方式存取已存档的变元，哪种不可以。(存档的变元为自动锁定的。) 最后，将变元解除存档并删除不使用的变元，以免占用内存。

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
1. 从主屏幕上指定带有以下变元类型的变元。 表达式: 5 → x1 函数: X^2 +4 → f(x) 数组: {5,10} → l1 矩阵: [30,25] → m1	[HOME] [CLEAR] 5 [STO▸] X 1 [ENTER] X [^] 2 [+] 4 [STO▸] [alpha] F [(] X [)] [ENTER] [2nd] [{] 5 [,] 1 0 [2nd] [}] [STO▸] [alpha] L 1 [ENTER] [2nd] [[] 3 0 [,] 2 5 [2nd] []] [STO▸] [alpha] M 1 [ENTER]	[♦] [HOME] [CLEAR] 5 [STO▸] X 1 [ENTER] X [^] 2 [+] 4 [STO▸] F [(] X [)] [ENTER] [2nd] [{] 5 [,] 1 0 [2nd] [}] [STO▸] L 1 [ENTER] [2nd] [[] 3 0 [,] 2 5 [2nd] []] [STO▸] M 1 [ENTER]	
2. 假定您要利用一个函数变元来开始执行一项运算，但您又忘记了该变元的名称。	5 [×]	5 [×]	5*
3. 显示 VAR-LINK 屏。 *本例子是假设上面指定的变元只有一个已定义了。*	[2nd] [VAR-LINK]	[2nd] [VAR-LINK]	
4. 改变屏幕的外观，只显示函数变元。 *尽管此方式在一个带有四变元的例子中不甚有用，但如有众多类型的变元存在的话，应考虑其作用如何。*	[F2] [▾] [▾] [▸] 5 [ENTER]	[F2] [▾] [▾] [▸] 5 [ENTER]	 

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
5. 将 f 函数变元突出显示并查看其内容。 *注意，该函数是通过 f(x)指定的，但在屏幕上是显示成 f。*	⊖ 2nd F6	⊖ F6	x^2+4
6. 关闭内容窗口。	ESC	ESC	
7. 在 f 变元仍然突出显示时，关闭 VAR-LINK 并将变元名称粘贴到输入行上。	ENTER	ENTER	5*f(注意，“ (” 已粘贴上去了。
8. 完成运算。	2) ENTER	2) ENTER	5*f(2)
<u>变元存档:</u>			
9. 重新显示 VAR-LINK 屏并将您要存档的变元突出显示。 *前一外观的变更不再生效，屏幕会显示出所有已定义的变元。*	2nd VAR-LINK (使用 ⊖ 来将 x1 突出显示)	2nd VAR-LINK (使用 ⊖ 来将 x1 突出显示)	VAR-LINK [A11] F1▾ Manage F2 View F3▾ Link F4 ✓ F5▾ All F6 Contents F7 FlashApp MAIN▾ f FUNC 19 l1 LIST 10 m1 MAT 12 t1 FIG 26 x1 EXPR 5
10. 使用 F1 Manage 工具栏菜单来进行变元存档。	F1 8	F1 8	F1▾ Manage 2:Copy 3:Rename 4:Move 5:Create Folder 6:Lock 7:UnLock 8:Archive Variable 9:Unarchive Variable VAR-LINK [A11] F1▾ Manage F2 View F3▾ Link F4 ✓ F5▾ All F6 Contents F7 FlashApp MAIN▾ f FUNC 19 l1 MAT 12 m1 MAT 12 ¤ x1 EXPR 5 ¤ 表示变元已存档。
11. 返回主屏幕并在计算中使用已存档的变元。	HOME 6 × X 1 ENTER	◆ HOME 6 × X 1 ENTER	■ 5·f(2) 40 ■ 6·x1 30 6*x1 MAIN RAD AUTO FUNC 6/30
12. 尝试将不同的数值存入已存档的变元中。	1 0 STO▸ X 1 ENTER	1 0 STO▸ X 1 ENTER	ERROR Variable is locked, protected, or archived ESC=CANCEL
13. 取消错误信息。	ESC	ESC	

步骤	TI-89 击键内容	TI-92 Plus 击键内容	显示
14. 使用 VAR-LINK 解除变元存档。	2nd [VAR-LINK] (使用 ⊖ 来将 x1 突出显示) F1 9	2nd [VAR-LINK] (使用 ⊖ 来将 x1 突出显示) F1 9	
15. 返回主屏幕并将不同的数值存入已解除存档的变元中。	HOME ENTER	◆ [HOME] ENTER	
<u>删除变元:</u>			
16. 显示 VAR-LINK 屏并利用 F5 All 工具栏菜单去选择所有的变元。 ✓ *标记指示所选定的项目。注意，这样一来 MAIN 文件夹也被选定。* ***备注:*** *作为 F5 (如您不想删除所有的变元)的替代，您也可以选择单个变元，将每个要删除的变元突出显示并按下 F4。*	2nd [VAR-LINK] F5 1	2nd [VAR-LINK] F5 1	 
17. 利用 F1 来删除。 ***备注:*** *您可按下 ← (替代 F1 1)来删除已标记的变元。*	F1 1	F1 1	
18. 确认删除。	ENTER	ENTER	
19. 由于 F5 1 同时也选定 MAIN 文件夹，会有一个错误信息说明您不能删除 MAIN 文件夹。请承认此信息。 *当 VAR-LINK 重新显示出来后，已删除的变元就不会列出。*	ENTER	ENTER	
20. 关闭 VAR-LINK 并返回当前的应用程序(此例子中的主屏幕)。 *当您使用 ESC (替代 ENTER)来关闭 VAR-LINK 屏时，突出显示的名称就不会粘贴在输入行上。*	ESC	ESC	

附件 A: 函数和指令

快速搜寻定位器.....96
运算的字母顺序列表.....100

本附件是叙述每一 TI-89 / TI-92 Plus 函数和指令的句法和动作。

函数或指令的名称。

输入名称的按键或菜单。您也可键入名称。

例子

Circle **CATALOG**

Circle x, y, r [, *drawMode*]

画出圆心位于窗口坐标(x, y)及半径为 r 的圆。

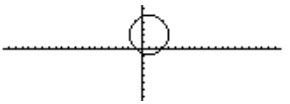
x、y 和 r 必须是实数值。

如 *drawMode* = 1, 画出圆(默认)。
如 *drawMode* = 0, 关闭圆。
如 *drawMode* = -1, 沿着圆颠倒象素。

备注: 重作图形会清除所有画出的项目。

在 ZoomSqr 观察窗内:

ZoomSqr:Circle 1,2,3 ENTER



函数或指令的展开。

变数是以*斜体*显示出来，置于中括号 [ ] 内的变数是选项。请勿键入中括号。

句法行是显示您所提供的变数的顺序和类型。应确认在多变数之间使用逗号(**,**)分隔开。

快速搜寻定位器

本章是列出 TI-89 / TI-92 Plus 功能组的函数和指令，并提供有本附件中叙述的页号。

代数

| ("**with**")	217	**cFactor()**	105	**comDenom()**	107
cSolve()	110	**cZeros**	114	**expand()**	127
factor()	129	**getDenom()**	135	**getNum()**	136
nSolve()	154	**propFrac()**	162	**randPoly()**	168
solve()	183	**tCollect()**	192	**tExpand()**	192
zeros()	199				

微积分

∫() (积分)	211	**Π()** (积)	213	**Σ()** (和)	213
arcLen()	102	**avgRC()**	103	***d()***	116
deSolve()	118	**fMax()**	131	**fMin()**	131
limit()	142	**nDeriv()**	151	**nInt()**	153
' (质数)	215	**seq()**	174	**taylor()**	191

作图

AndPic	101	**BldData**	104	**Circle**	105
ClrDraw	106	**ClrGraph**	106	**CyclePic**	114
DrawFunc	122	**DrawInv**	122	**DrawParm**	123
DrawPol	123	**DrawSlp**	123	**DrwCtour**	124
FnOff	131	**FnOn**	132	**Graph**	137
Line	142	**LineHorz**	143	**LineTan**	143
LineVert	143	**NewPic**	152	**PtChg**	162
PtOff	163	**PtOn**	163	**ptTest()**	163
PtText	163	**PxlChg**	163	**PxlCrcl**	163
PxlHorz	164	**PxlLine**	164	**PxlOff**	164
PxlOn	164	**pxlTest()**	164	**PxlText**	165
PxlVert	165	**RclGDB**	169	**RclPic**	169
RplcPic	173	**Shade**	178	**StoGDB**	186
StoPic	187	**Style**	187	**Trace**	195
XorPic	198	**ZoomBox**	201	**ZoomData**	202
ZoomDec	202	**ZoomFit**	203	**ZoomIn**	203
ZoomInt	203	**ZoomOut**	204	**ZoomPrev**	204
ZoomRcl	204	**ZoomSqr**	204	**ZoomStd**	205
ZoomSto	205	**ZoomTrig**	205		

数组

+ (加)	206	**–** (减)	206	**∗** (乘)	207
/ (除)	207	**⁻** (负号)	208	**^** (幂)	213
augment()	103	**crossP()**	110	**cumSum()**	113
dim()	120	**dotP()**	122	**exp▸list()**	127
left()	141	**list▸mat()**	144	**Δlist()**	144
mat▸list()	148	**max()**	148	**mid()**	149
min()	150	**newList()**	152	**polyEval()**	160
product()	162	**right()**	171	**rotate()**	171
shift()	179	**SortA**	185	**SortD**	185
sum()	188				

数学运算

+ (加)	206	− (减)	206	∗ (乘)	207
/ (除)	207	⁻ (负号)	208	**%** (百分数)	208
! (分数)	211	√**()** (平方根)	212	**^** (幂)	213
° (度)	214	∠ (角)	215	**°, ', "**	215
_ (下划线)	215	▸ (转换)	216	**10^()**	216
0b, 0h	218	**▸Bin**	103	**▸Cylind**	114
▸DD	116	**▸Dec**	116	**▸DMS**	122
▸Hex	137	**▸Polar**	159	**▸Rect**	169
▸Sphere	186	**abs()**	100	**and**	100
angle()	101	**approx()**	102	**ceiling()**	104
conj()	108	**cos**	108	**cos^{-1}()**	109
cosh()	109	**cosh^{-1}()**	110	**E**	124
e^()	124	**exact()**	126	**floor()**	130
fpart()	133	**gcd()**	133	**imag()**	139
int()	140	**intDiv()**	140	**iPart()**	140
isPrime()	141	**lcm()**	141	**ln()**	144
log()	146	**max()**	148	**min()**	150
mod()	150	**nCr()**	151	**nPr()**	154
P▸Rx()	156	**P▸Ry()**	156	**r** (弧度)	214
R▸Pθ()	167	**R▸Pr()**	168	**real()**	169
remain()	170	**rotate()**	171	**round()**	172
shift()	179	**sign()**	180	**sin()**	181
sin^{-1}()	181	**sinh()**	182	**sinh^{-1}()**	182
tan()	190	**tan^{-1}()**	190	**tanh()**	190
tanh^{-1}()	191	**tmpCnv()**	193	**ΔtmpCnv()**	194
x^{-1}	216				

矩阵

+ (加)	206	− (减)	206	∗ (乘)	207
/ (除)	207	⁻ (负号)	208	**.+** (点加)	210
.− (点减)	210	**.** (点乘)	211	**./** (点除)	211
.^ (点幂)	211	**^** (幂)	213	**augment()**	103
colDim()	107	**colNorm()**	107	**crossP()**	110
cumSum()	113	**det()**	120	**diag()**	120
dim()	120	**dotP()**	122	**eigVc()**	125
eigVl()	125	**Fill**	130	**identity()**	138
list▸mat()	144	**LU**	147	**mat▸list()**	148
max()	148	**mean()**	148	**median()**	148
min()	150	**mRow()**	150	**mRowAdd()**	150
newMat()	152	**norm()**	153	**product()**	162
QR	165	**randMat()**	168	**ref()**	170
rowAdd()	172	**rowDim()**	172	**rowNorm()**	173
rowSwap()	173	**rref()**	173	**simult()**	180
stdDev()	186	**subMat()**	188	**sum()**	188
T	189	**unitV()**	196	**variance()**	196
x^{-1}	216				

编程

=	209	**≠**	209	**<**	209
≤	210	**>**	210	**≥**	210
# (间接)	214	**→** (储存)	217	**◉** (注释)	218
and	100	**ans()**	102	**Archive**	102
ClrErr	106	**ClrGraph**	106	**ClrHome**	106
ClrIO	107	**ClrTable**	107	**CopyVar**	108
CustmOff	113	**CustmOn**	113	**Custom**	113
Cycle	114	**Define**	117	**DelFold**	118
DelVar	118	**Dialog**	120	**Disp**	121
DispG	121	**DispHome**	121	**DispTbl**	122
DropDown	123	**Else**	125	**ElseIf**	125
EndCustm	125	**EndDlog**	126	**EndFor**	126
EndFunc	126	**EndIf**	126	**EndLoop**	126
EndPrgm	126	**EndTBar**	126	**EndTry**	126
EndWhile	126	**entry()**	126	**Exec**	127
Exit	127	**For**	132	**format()**	132
Func	133	**Get**	133	**GetCalc**	134
getConfg()	134	**getFold()**	135	**getKey()**	135
getMode()	135	**getType()**	136	**getUnits()**	136
Goto	137	**If**	138	**Input**	139
InputStr	140	**Item**	141	**Lbl**	141
left()	141	**Local**	145	**Lock**	145
Loop	147	**MoveVar**	150	**NewFold**	152
NewProb	153	**not**	153	**or**	155
Output	156	**part()**	156	**PassErr**	158
Pause	159	**PopUp**	160	**Prgm**	161
Prompt	162	**Rename**	170	**Request**	170
Return	171	**right()**	171	**Send**	173
SendCalc	174	**SendChat**	174	**setFold()**	174
setGraph()	175	**setMode()**	176	**setTable()**	177
setUnits()	177	**Stop**	187	**Style**	187
switch()	188	**Table**	189	**Text**	192
Then	192	**Title**	193	**Toolbar**	194
Try	195	**Unarchiv**	196	**Unlock**	196
when()	197	**While**	198	**xor**	198

统计

! (阶乘)	211	**BldData**	104	**CubicReg**	112
cumSum()	113	**ExpReg**	129	**LinReg**	143
LnReg	145	**Logistic**	146	**mean()**	148
median()	148	**MedMed**	149	**nCr()**	151
NewData	151	**NewPlot**	152	**nPr()**	154
OneVar	155	**PlotsOff**	159	**PlotsOn**	159
PowerReg	161	**QuadReg**	166	**QuartReg**	167
rand()	168	**randNorm()**	168	**RandSeed**	168
ShowStat	180	**SinReg**	183	**SortA**	185
SortD	185	**stdDev()**	186	**TwoVar**	195
variance()	196				

字符串

& (附加)	211	# (间接)	214	**char()**	105
dim()	120	**expr()**	129	**format()**	132
inString()	140	**left()**	141	**mid()**	149
ord()	156	**right()**	171	**rotate()**	171
shift()	179	**string()**	187		

非字母名称(如 +、! 和 >)的运算列在本附件的尾部，具体在第 206 页开始。除非有特定的说明，本章中的所有例子均在默认重设模式下执行，所有变元均被假定为未定义的。此外，由于格式的限制，近似结果会被舍成三位小数(3.14159265359 会显示成 3.141...)。

abs() MATH/Number 菜单

abs(*表达式1***)** ⇒ *表达式*
abs(*数组1***)** ⇒ *数组*
abs(*矩阵1***)** ⇒ *矩阵*

对应变数的绝对值。

如变数是复数，则对应数模。

备注: 所有未定义的变元会作为实变元处理。

abs({π/2,⁻π/3}) ENTER $\{\frac{\pi}{2} \quad \frac{\pi}{3}\}$

abs(2−3*i*) ENTER $\sqrt{13}$

abs(z) ENTER $|z|$

abs(x+y*i*) ENTER $\sqrt{x^2+y^2}$

and MATH/Test and MATH/Base 菜单

布尔表达式1 **and** *表达式2* ⇒ *布尔表达式*
布尔数组1 **and** *数组2* ⇒ *布尔数组*
布尔矩阵1 **and** *矩阵2* ⇒ *布尔矩阵*

对应原始输入项的真实或虚假，以及简化的形式。

x≥3 and x≥4 ENTER x≥4

{x≥3,x≤0} and {x≥4,x≤⁻2} ENTER
{x ≥ 4 x ≤ ⁻2}

整数1 **and** *整数2* ⇒ *整数*

利用 **and** 运算来逐位比较两个实整数，就内部而言，两个实数都会被转换成 32 位的十进制数字。当相应的位得到比较后，如位是 1，那么结果就为 1；否则，结果就为 0。对应的数值表示位结果，并按照 Base 模式显示。

您可在任何数字基下输入整数。对于十进指或十六进制的输入项，您必须反复使用 0b 或 0h 前缀。如没有前缀，整数会被当成小数(基 10)处理。

如您输入一个远远超出 32 位十进制形式表示的小数整数，就会使用对称模运算来将数值带入合适的范围。

在十六进制基模式下:

0h7AC36 and 0h3D5F ENTER 0h2C16

重要事项: 是零，非字母 O。

在十进制基模式下:

0b100101 and 0b100 ENTER 0b100

在小数基模式:

37 and 0b100 ENTER 4

备注: 十进制输入项可有多达 32 个数位(前缀 0b 不计算在内)。十六进制输入项可有达 8 个数位。

AndPic CATALOG

AndPic *图形变元*[, *行*, *列*]

显示作图屏，并逻辑地在象素坐标(*行*, *列*)处将储存在*图形变元*内的图形和当前的作图屏 "加入"。

*图形变元*必须是图形类型。

默认坐标为(0,0)，就是屏幕的左上角。

在函数作图模式和 Y= 编辑器:

y1(x) = cos(x) ⊕

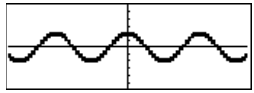
TI-89: [2nd][F6] Style = 3:Square

TI-92 Plus: [F6] Style = 3:Square

[F2] Zoom = 7:ZoomTrig

[F1] = 2:Save Copy As...

Type = Picture, Variable = PIC1



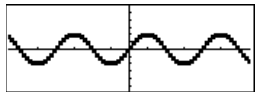
y2(x) = sin(x)

TI-89: [2nd][F6] Style = 3:Square

TI-92 Plus: [F6] Style = 3:Square

y1 = no checkmark (F4 to deselect)

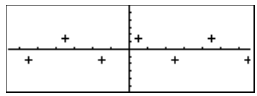
[F2] Zoom = 7:ZoomTrig



TI-89: [HOME]

TI-92 Plus: [◆][HOME]

AndPic PIC1 [ENTER] Done



angle() MATH/Complex 菜单

angle(*表达式1* **)** ⇒ *表达式*

对应*表达式1*的角，将*表达式1*当成整数演算。

备注: 所有未定义的变元会作实变元处理。

在 Degree 角模式下:

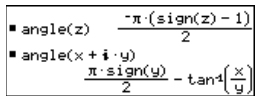
angle(0+2*i*) [ENTER] 90

在 Radian 角模式下:

angle(1+*i*) [ENTER] $\frac{\pi}{4}$

angle(z) [ENTER]

angle(x+ *i*y) [ENTER]



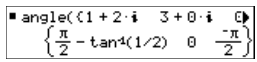
---

angle(*数组1* **)** ⇒ *数组*
angle(*矩阵1* **)** ⇒ *矩阵*

对应*数组1*或*矩阵1*中元素的角数组或矩阵数组，将每个元素当作表示两维直角坐标点的复数演算。

在 Radian 角模式下:

angle({1+2*i*,3+0*i*,0-4*i*}) [ENTER]



ans() 2nd [ANS] 键

ans() ⇒ *数值*
ans(*整数***)** ⇒ *数组*

对应主屏幕历史区域的前一答案。

如包含*整数*的话，指定要调用哪个前一答案。*整数*的有效范围是从 1 到 99，不得为表达式。默认值是最新的答案 1。

欲使用 **ans()** 来在主屏幕上生成斐波那奇数列，可按下:

1 ENTER 1
1 ENTER 1
2nd [ANS] + 2nd [ANS] ◀ ← 2 ENTER 2
ENTER 3
ENTER 5

approx() MATH/Algebra 菜单

approx(*表达式***)** ⇒ *数值*

当可能时，对应作为小数值的*表达式*计算，与当前的 Exact/Approx 模式无关。

这与主屏幕上输入*表达式*和按下 ◆ ENTER 等同。

approx(π) ENTER 3.141...

---

approx(*数组1***)** ⇒ *数组*
approx(*矩阵1***)** ⇒ *矩阵*

当可能时，对应数组或矩阵，其中每一元素都已被计算成小数值。

approx({sin(π),cos(π)}) ENTER
{0. -1.}

approx([√(2),√(3)]) ENTER
[1.414... 1.732...]

Archive CATALOG

Archive *变元1* [, *变元2*] [, *变元3*] ...

将指定的变元从 RAM 移到用户数据档案内存之中。

您可以象存取 RAM 中的变元一样存取一个已存档的变元。然而，由于该变元已自动锁定，您不能删除、重命名或储存一个已存档的变元。

欲解除变元存档，可使用 **Unarchiv**。

10→arctest ENTER 10
Archive arctest ENTER Done
5*arctest ENTER 50
15→arctest ENTER

ESC
Unarchiv arctest ENTER Done
15→arctest ENTER 15

arcLen() MATH/Calculus 菜单

arcLen(*表达式1***,***变元***,***开始***,***结束***)** ⇒ *表达式*

是对应*表达式1*从*开始*到*结束*的弧长，相对于*变元*。

与作图模式无关，弧长是被当作采取函数模式定义的积分计算的。

arcLen(cos(x),x,0,π) ENTER 3.820...

arcLen(f(x),x,a,b) ENTER

$$\int_a^b \sqrt{\left(\frac{d}{dx}(f(x))\right)^2+1}\ dx$$

---

arcLen(*数组1***,** *变元***,***开始***,***结束***)** ⇒ *数组*

对应*数组1* 的每一元素从*开始*到*结束*的弧长数组，相对于*变元*。

arcLen({sin(x),cos(x)},x,0,π)
{3.820... 3.820...}

augment() MATH/Matrix 菜单

augment(*数组1*, *数组2***)** ⇒ *数组*

对应*数组2* 附加在*数组1* 结尾的新数组。

augment({1,-3,2},{5,4}) ENTER

{1 -3 2 5 4}

---

augment(*矩阵1*, *矩阵2***)** ⇒ *矩阵*
augment(*矩阵1*; *矩阵2***)** ⇒ *矩阵*

对应*数组2*附加在*数组1*的新数组。当使用了“,”字符时，矩阵必须有等同的行维数，而*矩阵2*必须作为新的列附加在*矩阵1*中。当使用了“;”字符时，矩阵必须有等同列行维数，而*矩阵2*必须作为新的行附加在*矩阵1*中。请勿改变*矩阵1*或*矩阵2*。

[1,2;3,4]→M1 ENTER

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$$

[5;6]→M2 ENTER

$$\begin{bmatrix}5\\6\end{bmatrix}$$

augment(M1,M2) ENTER

$$\begin{bmatrix}1 & 2 & 5\\3 & 4 & 6\end{bmatrix}$$

[5,6]→M2 ENTER

[5 6]

augment(M1;M2) ENTER

$$\begin{bmatrix}1 & 2\\3 & 4\\5 & 6\end{bmatrix}$$

avgRC() CATALOG

avgRC(*表达式1*, *变元*[, *h*]**)** ⇒ *表达式*

对应前差商(变化平均率)。

*表达式1*可以是用户定义的函数名称(请参阅 **Func**)。

h 是分级值，如省略了 *h*，则它的默认值会是 0.001。

请注意，类似的函数 **nDeriv()** 使用中差商。

avgRC(f(x),x,h) ENTER

$$\frac{f(x+h) - f(x)}{h}$$

avgRC(sin(x),x,h)|x=2 ENTER

$$\frac{\sin(h+2) - \sin(2)}{h}$$

avgRC(x^2-x+2,x) ENTER

2.·(x - .4995)

avgRC(x^2-x+2,x,.1) ENTER

2.·(x - .45)

avgRC(x^2-x+2,x,3) ENTER 2·(x+1)

▸Bin MATH/Base 菜单

整数1 **▸Bin** ⇒ *整数*

将*整数1*转换成十进制数字。十进制或十六进制数字分别会带有 0b 或 0h 的前缀。

是零，非字母 O，随后有 b 或 h。
0b *十进制数字*
0h *十六进制数字*

十进制数字可有 32 个数位；十六进制数字可有 8 个数位。

不带前缀时，*整数1*会被当作小数(基 10)处理。结果会以十进制形式显示，与 Base 模式无关。

如您输入一个远远超出 32 位十进制形式表示的小数整数，就会使用对称模运算来将数值带入合适的范围。

256 ▸Bin ENTER 0b100000000

0h1F ▸Bin ENTER 0b11111

BldData CATALOG

BldData [*数据变元*]

基于画出当前图形所使用的信息来创建数据变元*数据变元*。**BldData** 在所有作图模式下均有效。

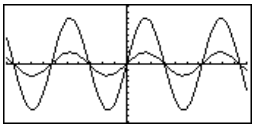
如遗漏了*数据变元*，数据将会被储存在系统变元 sysData 中。

备注: 在使用 **BldData**、*数据变元* 或 sysData (取决于与 **BldData** 一起使用的变数)后首次打开数据 / 矩阵编辑器时，会被设定成当前数据变元。

用于任何自变元(右边例子中的 x)的增量值会按照窗口变元值计算出来。

在函数作图模式和 Radian 角模式下:

```
8*sin(x)→y1(x) ENTER          Done
2*sin(x)→y2(x) ENTER          Done
ZoomStd ENTER
```



TI-89: [HOME]

TI-92 Plus: [♦] [HOME]

```
BldData ENTER                 Done
APPS 6 ENTER
```

DATA	x	y1	y2
	c1	c2	c3
1	-10.	4.3522	1.088
2	-9.832	3.168	.792
3	-9.664	1.8945	.47363
4	-9.496	.56769	.14192

3D 作图模式具有两个自变元。在右边的例子数据中，注意当 y 增加时，x 在其整个数值范围内都会保持为常数。

然后，x 增加到下一数值，而 y 会在整个数值范围内再次增加。此图案会继续下去，直至 x 在整个范围内增加。

备注: 以下的例子数据是来自一个 3D 图形的。

DATA	x	y	z1
	c1	c2	c3
1	-10.	-10.	0.
2	-10.	-8.571	5.8309
3	-10.	-7.143	8.9706
4	-10.	-5.714	9.8677

ceiling() MATH/Number 菜单

ceiling(*表达式1***)** ⇒ *整数*

```
ceiling(0.456) ENTER          1.
```

对应大于或等于变数的最接近整数。

变数可为实数或复数。

备注: 也请参阅 **floor()**。

---

ceiling(*数组1***)** ⇒ *数组*
ceiling(*矩阵1***)** ⇒ *矩阵*

```
ceiling({-3.1,1,2.5}) ENTER
                    {-3. 1 3.}
ceiling([0,-3.2i;1.3,4]) ENTER
                    [0   -3.·i
                     2.   4   ]
```

对应每一元素升限的数组或矩阵。

cFactor() MATH/Algebra/Complex 菜单

cFactor(*表达式1***[, *变元*])** ⇒ *表达式*
cFactor(*数组1***[, *变元*])** ⇒ *数组*
cFactor(*矩阵1***[, *变元*])** ⇒ *矩阵*

cFactor(*表达式1***)**是相对于所有变元在一个公分母之上的因式分解的*表达式1*。

表达式1 会尽最大可能接近地因式分解成一次关系因子，甚至如插入新的非实数。如您想相应于一个以上的变元进行因式分解，此替代方式是十分合适的。

```
cFactor(a^3*x^2+a*x^2+a^3+a) ENTER
```
$a\cdot(a+{}^{-}i)\cdot(a+i)\cdot(x+{}^{-}i)\cdot(x+i)$

```
cFactor(x^2+4/9) ENTER
```
$\frac{(3\cdot x+{}^{-}2\cdot i)\cdot(3\cdot x+2\cdot i)}{9}$

```
cFactor(x^2+3) ENTER
```
x^2+3

```
cFactor(x^2+a) ENTER
```
x^2+a

---

cFactor(*表达式1***,***变元***)**是相对于*变元*的因式分解的*表达式1*。

表达式1 会尽最大可能接近地因式分解成*变元*中也许带有非实数常数的一次因子，甚至会插入无理常数或在其它变元中是无理的子表达式。

因子和它们的项会作为主变元与*变元*一起储存。*变元*的类似幂会被收集在每一因子内。如因式分解仅需相应于该变元，以及您愿意接受任何其它变元中的无理表达式来增加相对于*变元*的因式分解，请包含*变元*。可能会有相对于其它变元的意想不到的因式分解出现。

对于 Exact/Approx 模式的 AUTO 设置，包含*变元*也允许带有浮点系数的近似值，其中无理系数是无法按照内建函数清晰地表达出。甚至当只有一个变元时，包含*变元*也许会产生更完整的因式分解。

备注: 也请参阅 **factor()**。

```
cFactor(a^3*x^2+a*x^2+a^3+a,x) ENTER
```
$a\cdot(a^2+1)\cdot(x+{}^{-}i)\cdot(x+i)$

```
cFactor(x^2+3,x) ENTER
```
$(x+\sqrt{3}\cdot i)\cdot(x+{}^{-}\sqrt{3}\cdot i)$

```
cFactor(x^2+a,x) ENTER
```
$(x+\sqrt{a}\cdot{}^{-}i)\cdot(x+\sqrt{a}\cdot i)$

```
cFactor(x^5+4x^4+5x^3-6x-3) ENTER
```
$x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3$

```
cFactor(ans(1),x) ENTER
```
$(x+1.11-1.07\cdot i)\cdot$

char() MATH/String 菜单

char(*整数***)** ⇒ *字符*

对应 TI-89 / TI-92 Plus 字符集中含有*整数*编号字符的字符串。

*整数*的有效范围是 0–255。

```
char(38) ENTER
```
"&"

```
char(65) ENTER
```
"A"

Circle CATALOG

Circle x, y, r [, *画图模式*]

画出圆心在窗口坐标(x, y)和半径为 r 的圆。

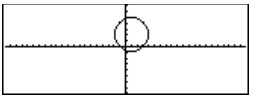
x、y 和 r 必须为实数值。

如*画图模式* = 1，画出圆(默认)。
如*画图模式* = 0，关闭圆。
如*画图模式* = -1，沿着圆颠倒象素。

备注: 重作图形会清除所有画出的项目。也请参阅 **PxlCrcl**。

在 ZoomSqr 观察窗:

```
ZoomSqr:Circle 1,2,3 ENTER
```



ClrDraw CATALOG

ClrDraw

清除作图屏并重设 Smart Graph 功能，以使下次作图屏显示后，图形能够重画。

在看到作图屏时，您按下 F4 (ReGraph)或:
TI-89: 2nd F6
TI-92 Plus: F6
并选择 1:ClrDraw 来清除所有画出的项目(点和线)。

ClrErr CATALOG

ClrErr

清除错误状态。设定 errornum 为零并清除内部错误变元。

程序中 **Try...EndTry** 的 **Else** 从句将使用 **ClrErr** 或 **PassErr**。如错误被执行或忽略，请使用 **ClrErr**。如对错误不知应怎么处理，请使用 **PassErr** 来将其发送到下一错误处理程序。如不存在未完的 **Try...EndTry** 错误处理程序，那么错误对话框就会以正常方式显示出来。

备注: 也请参阅 **PassErr** 和 **Try**。

程序表:

```
:clearerr()
:Prgm
:PlotsOff:FnOff:ZoomStd
:For i,0,238
:Δx*i+xmin→xcord
: Try
:  PtOn xcord,ln(xcord)
: Else
:  If errornum=800 or
     errornum=260 Then
:   ClrErr ◉clear the error
:  Else
:   PassErr ◉pass on any other
     error
:  EndIf
: EndTry
:EndFor
:EndPrgm
```

ClrGraph CATALOG

ClrGraph

清除利用 **Graph** 命令作出的或用 **Table** 命令创建的任何函数或表达式。(请参阅 **Graph** 或 **Table**)。

图形显示后，任何以前选定的 Y= 函数将会被重作出。

ClrHome CATALOG

ClrHome

清除储存在主屏幕历史区域 **entry()** 和 **ans()** 中的所有项目。请勿清除当前的输入项。

只要看到主屏幕，您可按下 F1 和选择 8:Clear Home 来清除历史区域。

对于函数，如对应任意常数或整数(@1、@2 等)的 **solve()**，可用 **ClrHome** 将后缀重设成 1。

ClrIO CATALOG

ClrIO

清除程序 I/O 屏。

ClrTable CATALOG

ClrTable

清除所有表格数值。仅应用于 Table Setup 对话框上的 ASK 设置。

只要看到表格屏处于 Ask 模式，您就可按下 F1 和选择 8:Clear Table 来清除数值。

colDim() MATH/Matrix/Dimensions 菜单

colDim(*矩阵***)** ⇒ *表达式*

对应*矩阵*中含有的列数。

备注: 也请参阅 **rowDim()**。

```
colDim([0,1,2;3,4,5]) ENTER        3
```

colNorm() MATH/Matrix/Norms 菜单

colNorm(*矩阵***)** ⇒ *表达式*

对应*矩阵*列中元素绝对值的最大和。

备注: 不允许未定义的矩阵元素。也请参阅 **rowNorm()**。

```
[1,-2,3;4,5,-6]→mat ENTER
```

$$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix}$$

```
colNorm(mat) ENTER                 9
```

comDenom() MATH/Algebra 菜单

comDenom(*表达式1*[,*变元*]**)** ⇒ *表达式*
comDenom(*数组1*[,*变元*]**)** ⇒ *数组*
comDenom(*矩阵1*[,*变元*]**)** ⇒ *矩阵*

comDenom(*表达式1***)**是对应完全扩展的分子在完全扩展的分母之上的化简比率。

```
comDenom((y^2+y)/(x+1)^2+y^2+y)
ENTER
```

$$\blacksquare\ \text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y\right)$$
$$\frac{x^2\cdot y^2+x^2\cdot y+2\cdot x\cdot y^2+2\cdot}{x^2+2\cdot x+1}$$

comDenom(*表达式1*,*变元***)**是分子和分母相对于*变元*扩展的化简比率。项及它们的因子会作为主变元与*变元*一起储存。类似*变元*的幂会被收集，收集到的系数也许会有一些伴随的因式分解。当需要令表达式更易于理解时，对*变元*遗漏状况作比较，有助于节省时间、内存和屏幕空间，也可使随后的运算更快得到结果和内存耗尽的可能性更低。

```
comDenom((y^2+y)/(x+1)
^2+y^2+y,x) ENTER
```

$$\blacksquare\ \text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y,\right.$$
$$\frac{x^2\cdot y\cdot(y+1)+2\cdot x\cdot y\cdot(y+1)}{x^2+2\cdot x+1}$$

```
comDenom((y^2+y)/(x+1)
^2+y^2+y,y) ENTER
```

$$\blacksquare\ \text{comDenom}\left(\frac{y^2+y}{(x+1)^2}+y^2+y,\right.$$
$$\frac{y^2\cdot(x^2+2\cdot x+2)+y\cdot(x^2+}{x^2+2\cdot x+1}$$

如*变元*没有在*表达式1*中出现，**comDenom(***表达式1*,*变元***)**是对应未完全扩展的分子在未完全扩展的分母之上的化简比率。该结果通常会节省更多的时间、内存和屏幕空间。此不完全的因式分解结果也可使随后的运算更快得到结果和内存耗尽的可能性更低。

```
comDenom(exprn,abc)→comden
(exprn) ENTER                      Done
comden((y^2+y)/(x+1)^2+y^2+y)
ENTER
```

$$\blacksquare\ \text{comden}\left(\frac{y^2+y}{(x+1)^2}+y^2+y\right)$$
$$\frac{(x^2+2\cdot x+2)\cdot y\cdot(y+1)}{(x+1)^2}$$

甚至当不存在分母，如 **factor()** 过慢或将内存耗用时，**comden** 函数通常是将不完全的因式分解存档的快捷途径。

提示: 输入此 **comden()** 函数定义并例行地将它作为 **comDenom()** 和 **factor()** 的替代来尝试。

```
comden(1234x^2*(y^3-y)+2468x
*(y^2-1)) ENTER
```

$1234 \cdot x \cdot (x \cdot y + 2) \cdot (y^2 - 1)$

conj() MATH/Complex 菜单

conj(*表达式1***)** ⇒ *表达式*
conj(*数组1***)** ⇒ *数组*
conj(*矩阵1***)** ⇒ *矩阵*

对应变数的复数共轭。

备注: 所有未定义的变元会被当作实数变元来处理。

```
conj(1+2i) ENTER
```

$1 - 2 \cdot i$

```
conj([2,1-3i;-i,-7]) ENTER
```

$$\begin{bmatrix} 2 & 1+3 \cdot i \\ i & -7 \end{bmatrix}$$

```
conj(z)
```

z

```
conj(x+iy)
```

$x + -i \cdot y$

CopyVar CATALOG

CopyVar *变元1*, *变元2*

复制*变元1* 的内容到*变元2*。如*变元2* 不存在的话，那么 **CopyVar** 可会创建之。

备注: 当您复制表达式、数组、矩阵或没有利用 **CopyVar** 进行化简的字符串以外的字符串时，**CopyVar** 类似储存指令(→)。您必须使用 **CopyVar** 连同非字母变元来键入，如 Pic 和 GDB 变元。

```
x+y→a ENTER          x + y
10→x ENTER           10
CopyVar a,b ENTER    Done
a→c ENTER            y + 10
DelVar x ENTER       Done
b ENTER              x + y
c ENTER              y + 10
```

cos() TI-89: 2nd COS 键 TI-92 Plus: COS 键

cos(*表达式1***)** ⇒ *表达式*
cos(*数组1***)** ⇒ *数组*

cos(*表达式1***)**对应作为表达式的变数的余弦。

cos(*数组1***)**对应*数组1*中所有元素的余弦数组。

备注: 变数会按照当前的角模式设置，被作为度或弧来演算。您可使用 ° 或 ʳ 来临时覆盖角模式。

在 Degree 角模式下:

```
cos((π/4)ʳ) ENTER
```

$\frac{\sqrt{2}}{2}$

```
cos(45) ENTER
```

$\frac{\sqrt{2}}{2}$

```
cos({0,60,90}) ENTER   {1   1/2   0}
```

在 Radian 角模式下:

```
cos(π/4) ENTER
```

$\frac{\sqrt{2}}{2}$

```
cos(45°) ENTER
```

$\frac{\sqrt{2}}{2}$

cos(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵余弦。这与计算每一元素的余弦*不*同。

当标量函数 f(A) 在*平方矩阵1* (A)中运算时，结果是由以下算法得出的:

1. 计算 A 的特征值 (λ_i) 和特征向量 (V_i)。

*平方矩阵1*必须为可对角的，不能有未赋值的符号变元。

2. 形成矩阵:

$$B = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \text{ 和 } X = [V_1, V_2, \dots, V_n]$$

3. 然后 $A = X\,B\,X^{-1}$ 和 $f(A) = X\,f(B)\,X^{-1}$。例如，$\cos(A) = X\cos(B)\,X^{-1}$，其中:

$$\cos(B) = \begin{bmatrix} \cos(\lambda_1) & 0 & \dots & 0 \\ 0 & \cos(\lambda_2) & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \cos(\lambda_n) \end{bmatrix}$$

所有运算均使用浮点算术进行。

在 Radian 角模式下:

cos([1,5,3;4,2,1;6,⁻2,1]) ENTER

$$\begin{bmatrix} .212\dots & .205\dots & .121\dots \\ .160\dots & .259\dots & .037\dots \\ .248\dots & ^-.090\dots & .218\dots \end{bmatrix}$$

cos⁻¹() TI-89: ◆[cos⁻¹] 键 TI-92 Plus: 2nd [cos⁻¹] 键

cos⁻¹(*表达式1***)** ⇒ *表达式*
cos⁻¹(*数组1***)** ⇒ *数组*

cos⁻¹(*表达式1***)**对应角度，该角度是作为表达式的*表达式1*的余弦。

cos⁻¹(*数组1***)**对应*数组1*中每一元素的反余弦数组。

备注: 按照当前的角模式设置，结果是对应度或弧度角。

在 Degree 角模式下:

cos⁻¹(1) ENTER 0

在 Radian 角模式下:

cos⁻¹({0,.2,.5}) ENTER

$\{\frac{\pi}{2}$ 1.369… 1.047…}

cos⁻¹(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵反余弦。这与计算每一元素的反余弦*不*同。有关计算方式，请参阅 **cos()**。

平方矩阵1 必须为可对角的，结果一直含有浮点数字。

在 Radian 角模式和 Rectangular 复数格式模式下:

cos⁻¹([1,5,3;4,2,1;6,⁻2,1]) ENTER

$$\begin{bmatrix} 1.734\dots+.064\dots\cdot i & ^-1.490\dots+2.105\dots\cdot i & \dots \\ ^-.725\dots+1.515\dots\cdot i & .623\dots+.778\dots\cdot i & \dots \\ ^-2.083\dots+2.632\dots\cdot i & 1.790\dots-1.271\dots\cdot i & \dots \end{bmatrix}$$

cosh() MATH/Hyperbolic 菜单

cosh(*表达式1***)** ⇒ *表达式*
cosh(*数组1***)** ⇒ *数组*

cosh (*表达式1***)**是对应作为表达式的变数的双曲余弦。

cosh (*数组1***)**是对应*数组1*每一元素的双曲余弦数组。

cosh(1.2) ENTER 1.810…

cosh({0,1.2}) ENTER {1 1.810…}

cosh(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵双曲余弦。这与计算每一元素的双曲余弦*不*同。有关计算方式，请参阅 **cos()**。

*平方矩阵1*必须为可对角的，结果一直含有浮点数字。

在 Radian 角模式下:

```
cosh([1,5,3;4,2,1;6,-2,1])
ENTER
```

$$\begin{bmatrix} 421.255 & 253.909 & 216.905 \\ 327.635 & 255.301 & 202.958 \\ 226.297 & 216.623 & 167.628 \end{bmatrix}$$

cosh^{-1}() MATH/Hyperbolic 菜单

cosh^{-1}(*表达式1***)** ⇒ *表达式*
cosh^{-1}(*数组1***)** ⇒ *数组*

cosh^{-1}(*表达式1***)**对应作为表达式的变数的反双曲余弦。

cosh^{-1}(*数组1***)**对应*数组1*每一元素的反双曲余弦数组。

```
cosh-1(1) ENTER                    0
cosh-1({1,2.1,3}) ENTER
          {0   1.372...   cosh-1(3)}
```

cosh^{-1}(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵反双曲余弦。这与计算每一元素的反双曲余弦*不*同。有关计算方式，请参阅 **cos()**。

*平方矩阵1*必须为可对角的，结果一直含有浮点数字。

在 Radian 角模式和 Rectangular 复数格式模式下:

```
cosh-1([1,5,3;4,2,1;6,-2,1])
ENTER
```

$$\begin{bmatrix} 2.525\ldots+1.734\ldots\cdot i & -.009\ldots-1.490\ldots\cdot i\ \ldots \\ .486\ldots-.725\ldots\cdot i & 1.662\ldots+.623\ldots\cdot i\ \ldots \\ -.322\ldots-2.083\ldots\cdot i & 1.267\ldots+1.790\ldots\cdot i\ \ldots \end{bmatrix}$$

crossP() MATH/Matrix/Vector ops 菜单

crossP(*数组1***,** *数组2***)** ⇒ *数组*

对应作为数组的*数组1*和*数组2*的交叉积。

*数组1*和*数组2*必须具有等同的维数，而且维数必须是 2 或 3。

```
crossP({a1,b1},{a2,b2}) ENTER
              {0  0  a1·b2-a2·b1}
crossP({0.1,2.2,-5},{1,-.5,0})
ENTER
                {-2.5  -5.  -2.25}
```

crossP(*向量1***,** *向量2***)** ⇒ *向量*

对应*向量1*和*向量2*的交叉积的行或列向量(取决于变数)。

*向量1*和*向量2*必须为行向量或两者必须为列向量，两向量的维数必须相等，维数是 2 或 3。

```
crossP([1,2,3],[4,5,6]) ENTER
                        [-3  6  -3]
crossP([1,2],[3,4]) ENTER
                          [0  0  -2]
```

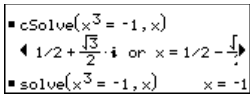
cSolve() MATH/Algebra/Complex 菜单

cSolve(*方程***,** *变元***)** ⇒ *布尔表达式*

对应一个*变元*方程的候选复数解。目的是为所有的实数或非实数解产生候选值。甚至如*方程*是实数，**cSolve()** 也允许在实数模式下的非实数结果。

尽管 TI-89 / TI-92 Plus 会将所有未定义变元作实数处理，**cSolve()** 可解出多项式方程的复数解。

```
cSolve(x^3=-1,x) ENTER
solve(x^3=-1,x) ENTER
```



在解法进行期间，**cSolve()** 会临时地将域设定成复数，甚至当前的域是实数时。在复数域中，带有奇数分母的分数幂是使用主分支而不用实数分支。这样一来，从 **solve()** 的解到含有该分数幂的方程就不需要来自 **cSolve()** 的它们的子集。

```
cSolve(x^(1/3)=-1,x) ENTER   false
solve(x^(1/3)=-1,x) ENTER    x = -1
```

cSolve() 以精确符号方式开始，除了在 EXACT 模式以外，如有需要，**cSolve()** 也可用交互近似复数多项式因式分解。

备注: 也请参阅 **cZeros()**、**solve()** 和 **zeros()**。

备注: 如*方程*为带有函数 **abs()**、**angle()**、**conj()**、**real()** 或 **imag()** 的非多项式，您则需要在*变元* 的结尾加入下划线 _ (**TI-89:** ◆ [_] **TI-92 Plus:** 2nd [_])。默认是将变元作为实数值处理。

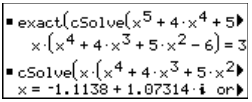
如您使用*变元_*，变元会被当作复数处理。

您可对带有非实数值的方程中的任何其它变元使用*变元_*。否则，您就会得到一个意想不到结果。

以 Fix 2 显示 Digits 模式:

exact(cSolve(x^5+4x^4+5x ^3-6x-3=0,x)) ENTER

cSolve(ans(1),x) ENTER



z 被当作实数处理:

cSolve(conj(z)=1+*i*,z) ENTER

z=1+*i*

z_ 被当作复数处理:

cSolve(conj(z_)=1+*i*,z_) ENTER

z_=1−*i*

---

cSolve(*方程1* **and** *方程2* [**and** ...],
{*变元或猜测值1*, *变元或猜测值2* [, ...]})
⇒ *布尔表达式*

对应联立代数方程的候选复数解，其中每一*变元或猜测值*是指定您要求解的变元。

作为任选项，您可以为变元指定一个起始猜测值，每一*变元或猜测值*必须具备以下形式:

变元
– 或 –
变元 = 实数或非实数

例如，x 是有效的，所以 x=3+*i*。

如所有方程为多项式或如您没有指定任何起始猜测值，**cSolve()** 就会使用词汇的 Gröbner /Buchberger 消除方式来尝试定义所有复数解。

复数解可包含实数和非实数解，如右图的例子所示。

备注: 以下例子是使用下划线 _ (**TI-89:** ◆ [_] **TI-92 Plus:** 2nd [_])，所以，变元会被当作复数处理。

cSolve(u_*v_−u_=v_ and v_^2=−u_,{u_,v_}) ENTER

$$u_=1/2+\frac{\sqrt{3}}{2}\cdot i \text{ and } v_=1/2-\frac{\sqrt{3}}{2}\cdot i$$

$$\text{or } u_=1/2-\frac{\sqrt{3}}{2}\cdot i \text{ and } v_=1/2+\frac{\sqrt{3}}{2}\cdot$$

or u_=0 and v_=0

联立*多项式*方程可有额外的无值变元，但它是代表稍后会被替换的给出数值。

cSolve(u_*v_−u_=c_*v_ and v_^2=−u_,{u_,v_}) ENTER

$$u_=\frac{-\left(\sqrt{1-4\cdot c_}+1\right)^2}{4} \text{ and } v_=\frac{\sqrt{1-4\cdot c_}+1}{2}$$

or

$$u_=\frac{-\left(\sqrt{1-4\cdot c_}-1\right)^2}{4} \text{ and } v_=\frac{-\left(\sqrt{1-4\cdot c_}-1\right)}{2}$$

or u_=0 and v_=0

您也可将方程中没有出现的解变元包含进去，这些解示出解族中多少会含有任意常数 @*k*，其中 *k* 是从 1 到 255 的整数后缀。当您使用 **ClrHome** 或 F1 8:Clear Home 时，后缀会重设到 1。

对于多项式组，计算的时间或内存的耗用很大程度是取决于您列出解变元的顺序。如您开始选择排放内存或由于您的忍耐程度有限，可尝试重新安排方程和 / 或*变元或猜测值*数组中的变元。

```
cSolve(u_*v_-u_=v_ and
v_^2=-u_,{u_,v_,w_}) ENTER
```

$$u_=1/2+\frac{\sqrt{3}}{2}\cdot i \text{ and } v_=1/2-\frac{\sqrt{3}}{2}\cdot i \text{ and } w_=@1$$

or

$$u_=1/2-\frac{\sqrt{3}}{2}\cdot i \text{ and } v_=1/2+\frac{\sqrt{3}}{2}\cdot i \text{ and } w_=@1$$

or $u_=0$ and $v_=0$ and $w_=@1$

如您没有将任何猜测值包含进去，或如任何方程在何变元中是非多项式，但所有方程在所有解变元中均为一次的话，**cSolve()** 会使用高斯消除方式来尝试定义所有解。

```
cSolve(u_+v_=e^(w_) and u_-v_=
i, {u_,v_}) ENTER
```

$$u_=\frac{e^{w_}}{2}+1/2\cdot i \text{ and } v_=\frac{e^{w_}-i}{2}$$

如方程组在其所有变元中既不是多项式，在解变元中也不是一次的话，**cSolve()** 会使用近似重复方式定义最多一个解。为了这样做，解变元数必须等于方程数，方程中的其它所有变元必须化简成数字。

```
cSolve(e^(z_)=w_ and w_=z_^2,
{w_,z_}) ENTER
```

$w_=.494\ldots$ and $z_=-.703\ldots$

一个非实数猜测值通常需要定义一个非实数的解。而对于收敛，猜测值则会是相当接近解的。

```
cSolve(e^(z_)=w_ and w_=z_^2,
{w_,z_=1+i}) ENTER
```

$w_=.149\ldots+4.891\ldots\cdot i$ and $z_=1.588\ldots+1.540\ldots\cdot i$

CubicReg MATH/Statistics/Regressions 菜单

CubicReg *数组1*, *数组2*[, [*数组3*] [, *数组4*, *数组5*]]

计算三次多项式回归和更新所有统计变元。

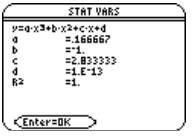
除*数组5*外，所有数组必须具有等同的维数，

数组1 代表 x 数组。
数组2 代表 y 数组。
数组3 代表频数。
数组4 代表类别代码。
数组5 代表类别包含数组。

备注: *数组1*到*数组4*必须为变元名称或 c1-c99 (上一数据变元中的列数是在数据 / 矩阵编辑器中示出)。*数组5*不能为变元名称，也不能是 c1-c99。

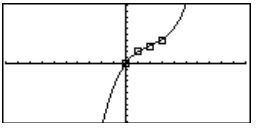
在函数作图模式。

```
{0,1,2,3}→L1 ENTER        {0 1 2 3}
{0,2,3,4}→L2 ENTER        {0 2 3 4}
CubicReg L1,L2 ENTER          Done
ShowStat ENTER
```




```
ENTER
regeq(x)→y1(x) ENTER          Done
NewPlot 1,1,L1,L2 ENTER       Done
```

◆ [GRAPH]



cumSum() MATH/List 菜单

cumSum(*数组1*) ⇒ *数组*

对应*数组1*中每一元素的累积和数组，有元素 1 开始。

```
cumSum({1,2,3,4}) ENTER
                {1 3 6 10}
```

cumSum(*矩阵1*) ⇒ *矩阵*

对应*矩阵1*中元素的累积和矩阵，每一元素是从顶到底的列的累进和。

```
[1,2;3,4;5,6]→m1 ENTER
```

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

```
cumSum(m1) ENTER
```

$$\begin{bmatrix} 1 & 2 \\ 4 & 6 \\ 9 & 12 \end{bmatrix}$$

CustmOff CATALOG

CustmOff

删除自定工具栏。

CustmOn 和 **CustmOff** 允许程序来控制工具栏。您可按下 2nd [CUSTOM] 来切换自定工具栏的开和关。此外，当您改变应用程序时，自定工具栏会被自动删除。

请参阅 **Custom** 程序列表的例子。

CustmOn CATALOG

CustmOn

激活已经在 **Custom...EndCustm** 块中设定的自定工具栏。

CustmOn 和 **CustmOff** 允许程序来控制工具栏。您可按下 2nd [CUSTOM] 来切换自定工具栏的开和关。

请参阅 **Custom** 程序列表的例子。

Custom 2nd [CUSTOM] 键

Custom
块
EndCustm

当您按下 2nd [CUSTOM] 时，设定已激活的工具栏。除了 Title 和 Item 陈述没有标记以外，它与 **ToolBar** 指令十分相似。

*块*可为单一陈述或一系列通过字符“:”分隔的陈述。

备注: 2nd [CUSTOM] 具有切换的作用。第一个实例是请求菜单，第二个实例是删除菜单。当您改变应用程序时，菜单也会被删除。

程序列表:

```
:Test()
:Prgm
:Custom
:Title  "Lists"
:Item   "List1"
:Item   "Scores"
:Item   "L3"
:Title  "Fractions"
:Item   "f(x)"
:Item   "h(x)"
:Title  "Graph"
:EndCustm
:EndPrgm
```

Cycle CATALOG

Cycle

立即转移程序控制到当前循环的下一迭代(**For**、**While** 或 **Loop**)。

Cycle 不允许超出三个循环结构(**For**、**While** 或 **Loop**)。

程序列表:

```
:◉ Sum the integers from 1 to
  100 skipping 50.
:0→temp
:For i,1,100,1
:If i=50
:Cycle
:temp+i→temp
:EndFor
:Disp temp
```

执行后 temp 的内容: 5000

CyclePic CATALOG

CyclePic *图形名称字符串*, *n* [, [*等待*] , [*循环*], [*方向*]]

按照指定的时间间隔显示所有已指定的 PIC 变元。在图象与图象之间，用户可有按照时间；循环次数；方向；循环、向前或向后的选项控制。

*方向*对于循环是 1，对于向前和向后是 ⁻1。默认值 = 1。

1. 保存三个名为 pic1、pic2 和 pic3 的图象。
2. 输入: CyclePic "pic",3,.5,4,⁻1
1. 三个图象(3)会在半秒(.5)间隔内自动显示出来，进行四次循环(4)并前进和后退(⁻1)。

▸Cylind MATH/Matrix/Vector ops 菜单

向量▸**Cylind**

以柱面形式 [r∠θ, z] 显示行和列向量。

*向量*必须为三个元素。它可为行或列。

[2,2,3] ▸Cylind ENTER

$[2\cdot\sqrt{2}\ \angle\frac{\pi}{4}\quad 3]$

cZeros() MATH/Algebra/Complex 菜单

cZeros(*表达式*, *变元***)** ⇒ *数组*

对应构成*表达式*=0 的*变元*的候选实数值和非实数值数组。**cZeros()** 透过计算 **exp▸list(cSolve(***表达式*=0,*变元***)**,*变元***)**来实现。否则，**cZeros()** 就会类似 **zeros()**。

备注: 也请参阅 **cSolve()**、**solve()** 和 **zeros()**。

备注: 如*表达式*不是带有函数 **abs()**、**angle()**、**conj()**、**real()** 或 **imag()** 的非多项式，您则需要在*变元*的结尾放置一个下划线 _ (**TI-89:** ◆ [_] **TI-92 Plus:** 2nd [_])。默认是将变元当成实数值处理。如您使用*变元*_，变元就会被当成复数处理。

您可对带有非实数值的*表达式*中的任何其它变元使用*变元*_。否则，您就会得到一个意想不到结果。

以 Fix 3 显示 Digits 模式:

cZeros(x^5+4x^4+5x^3−6x−3,x) ENTER

{⁻2.125 ⁻.612 .965 ⁻1.114−1.073·*i* ⁻1.114+1.073·*i*}

z 是被当成实数处理:

cZeros(conj(z)−1−*i*,z) ENTER

{1+*i*}

z_ 是被当成复数处理:

cZeros(conj(z_)−1−*i*,z_) ENTER

{1−*i*}

---

cZeros({*表达式1*, *表达式2* [, ...]},
{*变元或猜测值1*,*变元或猜测值2* [, ...]})** ⇒ *矩阵*

对应表达式同时为零的候选位置。每一*变元或猜测值*是指定一个您要寻找的未知值。

作为任选项，您可以为变元指定一个起始猜测值，每一*变元或猜测值*必须具备以下形式:

变元
– 或 –
变元 = 实数后非实数

例如，x 是有效的，所以 x=3+$\boldsymbol{i}$。

如所有表达式为多项式，或如您没有指定任何起始猜测值，**cZeros()** 就会使用词汇的 Gröbner /Buchberger 消除方式来尝试定义所有复数零。

备注: 以下例子是使用下划线 _ (**TI-89:** ◆ [_] **TI-92 Plus:** 2nd [_])，所以，变元会被当作复数处理。

复数零可包含实数和非实数零，如右图中的内容所示。

结果矩阵的每一行代表替代零，带有与*变元或猜测值*数组相同顺序的成分。欲提取一行，可以 [*行*] 指明矩阵。

```
cZeros({u_*v_-u_-v_,v_^2+u_},
{u_,v_}) ENTER
```

$$\begin{bmatrix} 1/2 - \frac{\sqrt{3}}{2}\cdot\boldsymbol{i} & 1/2 + \frac{\sqrt{3}}{2}\cdot\boldsymbol{i} \\ 1/2 + \frac{\sqrt{3}}{2}\cdot\boldsymbol{i} & 1/2 - \frac{\sqrt{3}}{2}\cdot\boldsymbol{i} \\ 0 & 0 \end{bmatrix}$$

提取行 2:

```
ans(1)[2] ENTER
```

$$\begin{bmatrix} 1/2 + \frac{\sqrt{3}}{2}\cdot\boldsymbol{i} & 1/2 - \frac{\sqrt{3}}{2}\cdot\boldsymbol{i} \end{bmatrix}$$

联立*多项式*可有额外的无值变元，但它是代表稍后会被替换的给出数值。

```
cZeros({u_*v_-u_-(c_*v_),
v_^2+u_},{u_,v_}) ENTER
```

$$\begin{bmatrix} \frac{-\left(\sqrt{1-4\cdot c_}+1\right)^2}{4} & \frac{\sqrt{1-4\cdot c_}+1}{2} \\ \frac{-\left(\sqrt{1-4\cdot c_}-1\right)^2}{4} & \frac{-\left(\sqrt{1-4\cdot c_}-1\right)}{2} \\ 0 & 0 \end{bmatrix}$$

您也可将表达式中没有出现的未知变元包含进去，这些零示出零族中多少会含有任意常数 @k，其中 k 是从 1 到 255 的整数后缀。当您使用 **ClrHome** 或 F1 8:Clear Home 时，后缀会重设到 1。

```
cZeros({u_*v_-u_-v_,v_^2+u_},
{u_,v_,w_}) ENTER
```

$$\begin{bmatrix} 1/2 - \frac{\sqrt{3}}{2}\cdot\boldsymbol{i} & 1/2 + \frac{\sqrt{3}}{2}\cdot\boldsymbol{i} & @1 \\ 1/2 + \frac{\sqrt{3}}{2}\cdot\boldsymbol{i} & 1/2 - \frac{\sqrt{3}}{2}\cdot\boldsymbol{i} & @1 \\ 0 & 0 & @1 \end{bmatrix}$$

对于多项式组，计算的时间或内存的耗用很大程度是取决于您列出的顺序。如您开始选择排放内存或由于您的忍耐程度有限，可尝试重新安排表达式和 / 或*变元或猜测值*数组中的变元。

如您没有包含任何猜测值，又或如果任何表达式中的任何变元是非多项式，而所有表达式的未知值均为一次的，那么，**cZeros()** 就会使用高斯消除方式来尝试定义所有零。

```
cZeros({u_+v_-e^(w_),u_-v_-i},
{u_,v_}) ENTER
```

$$\begin{bmatrix} \frac{e^{w_}}{2} + 1/2\cdot\boldsymbol{i} & \frac{e^{w_}-\boldsymbol{i}}{2} \end{bmatrix}$$

如系统的所有变元都不是多项式，未知值中也没有一次的话，**cZeros()** 就会使用近似交互法定义最多一个零。为此，未知值的数必须等于表达式的数，而表达式中的其它所有变元则必须简化成数字。

```
cZeros({e^(z_)-w_,w_-z_^2},
{w_,z_}) ENTER
```

$$\begin{bmatrix} .494\ldots & {}^{-}.703\ldots \end{bmatrix}$$

非实数猜测值通常需要定义一个非实数零。对于收敛，猜测值也许必须接近零。

cZeros({*e*^(z_)−w_,w_−z_^2},
{w_,z_=1+*i*}) ENTER

[.149…+4.89…·*i* 1.588…+1.540…·*i*

d()

2nd [*d*] 键或 **MATH/Calculus** 菜单

d*(表达式1*, *变元* [,*顺序*]) ⇒ *表达式***
d*(数组1*, *变元* [,*顺序*]) ⇒ *数组***
d*(矩阵1*, *变元* [,*顺序*]) ⇒ *矩阵***

对应*表达式1*相对于*变元*的第一导数。*表达式1*可为数组或矩阵。

如包含有*顺序*，则必须为整数，如该数少于零，那么结果就会是反导数。

d*()** 并非遵循全化简其变数，然后将函数定义应用到这些已化简的变数中去的正常计算途径。相反，d*()** 是按照以下步骤执行的:

1. 将第二变数化简成其长度刚好不至于导致出现非变元。
2. 将第一变数化简成其长度能够为步骤 1 中定义的变元调用任何储存的数值。
3. 相对于步骤 1 来的变元，定义步骤 2 结果的符号导数。
4. 如来自步骤 1 的变元带有已储存的数值或利用"with" (|)算子指定的数值，请将该数值代入步骤 3 中得出的结果。

d(3x^3−x+7,x) ENTER $9x^2-1$

d(3x^3−x+7,x,2) ENTER 18·x

d(f(x)*g(x),x) ENTER

$$\frac{d}{dx}(f(x))\cdot g(x)+\frac{d}{dx}(g(x))\cdot f(x)$$

d(sin(f(x)),x) ENTER

$$\cos(f(x))\frac{d}{dx}(f(x))$$

d(x^3,x)|x=5 ENTER 75

d(*d*(x^2*y^3,x),y) ENTER $6\cdot y^2\cdot x$

d(x^2,x,⁻1) ENTER $\frac{x^3}{3}$

d({x^2,x^3,x^4},x) ENTER
{2·x 3·x² 4·x³}

▸DD

MATH/Angle 菜单

数字▸**DD** ⇒ *数值*
数组1▸**DD** ⇒ *数组*
矩阵1▸**DD** ⇒ *矩阵*

是对应变数的等同小数。变数是一个已利用模式设置来演算成弧度或角的数字、数组或矩阵。

备注: ▸**DD** 同样接受以弧度的输入。

在 Degree 角模式下:

1.5°▸DD ENTER 1.5°

45° 22'14.3"▸DD ENTER 45.370…°

{45° 22'14.3",60° 0'0"}▸DD ENTER
{45.370… 60}°

在 Radian 角模式下:

1.5▸DD ENTER 85.9°

▸Dec

MATH/Base 菜单

整数1▸**Dec** ⇒ *整数*

将*整数1*转换成小数(基 10)。十进制或十六进制输入项分别必须带有 0b 或 0h 前缀。

0b10011▸Dec ENTER 19

0h1F▸Dec ENTER 31

┌ 是零，非字母 O，后面跟随 b 或 h。
0b *十进制数字*
0h *十六进制数字*
└ 十进制数字可有 32 位，十六进制数字可有 8 位。

不带前缀时，*整数1* 会被当作小数处理，结果会以小数显示，与 Base 模式无关。

Define CATALOG

Define *函数名称*(*变数1名称*, *变数2名称*, ...) = *表达式*

创建作为用户定义函数的*函数名称*，然后您就可以把*函数名称*()当作内建函数一样使用了。函数是利用所提供的变数计算*表达式*并得出结果的。

*函数名称*不得与系统变元或内建函数的名称相同。

变数名称是占位符号；当您使用函数时，您不能使用与变数相同的名称。

备注: Define 的此形式与执行表达式等同: *表达式*→*函数名称*(*变数1名称*,*变数2名称*)。
此命令也可用于定义简单变元；例如，Define a=3。

```
Define g(x,y)=2x-3y ENTER        Done
g(1,2) ENTER                      -4
1→a:2→b:g(a,b) ENTER              -4

Define h(x)=when(x<2,2x-3,
-2x+3) ENTER                      Done

h(-3) ENTER                       -9
h(4) ENTER                        -5

Define eigenvl(a)=
  cZeros(det(identity(dim(a)
  [1])-x*a),x) ENTER              Done
eigenvl([-1,2;4,3]) ENTER
```

$$\left\{\frac{2\cdot\sqrt{3}-1}{11}\quad\frac{-(2\cdot\sqrt{3}+1)}{11}\right\}$$

Define *函数名称*(*变数1名称*, *变数2名称*, ...) = **Func**
块
EndFunc

是与 **Define** 的前一形式完全相同，除此形式以外，用户定义的函数*函数名称*()可执行多陈述块。

*块*可为“:”分隔的单陈述或系列陈述。*块*也可包含表达式和指令(如 **If**、**Then**、**Else** 和 **For**)。这点允许函数 *函数名称*() 使用 **Return** 指令来得出指定的结果。

备注: 它通常是为了令程序编写人员在程序编辑器中更简便地创作和编辑函数的此形式而不用在输入行上进行。

```
Define g(x,y)=Func:If x>y Then
:Return x:Else:Return y:EndIf
:EndFunc ENTER                    Done

g(3,-7) ENTER                     3
```

Define *程序名称*(*变数1名称*, *变数2名称*, ...) = **Prgm**
块
EndPrgm

创建作为程序或子程序的*程序名称*，但不能利用 **Return** 得出结果。可执行多陈述块。

*块*可为“:”分隔的单陈述或系列陈述。*块*也可包含表达式和指令(如 **If**、**Then**、**Else** 和 **For**)，没有限制。

备注: 它通常是为了令程序编写人员在程序编辑器中更简便地编辑程序块而不用在输入行上进行。

```
Define listinpt()=prgm:Local
  n,i,str1,num:InputStr "Enter
  name of list",str1:Input "No.
  of elements",n:For
  i,1,n,1:Input "element
  "&string(i),num:
  num→#str1[i]:EndFor:EndPrgm
  ENTER
                                  Done

listinpt() ENTER Enter name of list
```

DelFold CATALOG

DelFold *文件夹名称1*[, *文件夹名称2*] [, *文件夹名称3*] ...

删除带有名称*文件夹名称1*、*文件夹名称2*等的用户定义文件夹。如文件夹内任何变元的话，会显示出一个错误信息。

备注: 您不能删除主 main 文件夹。

```
NewFold games ENTER          Done
```

(创建文件夹 games)

```
DelFold games ENTER          Done
```

(删除文件夹 games)

DelVar CATALOG

DelVar *变元1*[, *变元2*] [, *变元3*] ...

从内存中删除指定的变元。

```
2→a ENTER                    2
(a+2)^2 ENTER               16
DelVar a ENTER           Done
(a+2)^2 ENTER        (a + 2)^2
```

deSolve() MATH/Calculus 菜单

deSolve(*第一或第二顺序Ode*, *自变元*, *因变元***)** ⇒ *普通解*

对应明显或蕴含地指定第一或第二顺序普通微分方程(ODE)普通解的方程。在 ODE 下:

- 使用一个原始符号(', 按下 2nd [′])来表示相对于自变元的因变元第一导数。
- 使用两个原始符号表示对应的第二导数。

符号 ' 仅用于 **deSolve()** 内的导数。其它情况下请使用 *d*( )。

第一顺序方程的普通解含有 @*k* 的任意常数，其中 *k* 是 1 到 255 的整数前缀。当使用 **ClrHome** 或 F1 8: Clear Home 时，前缀会重设成 1。第二顺序方程的解含有两个任意常数。

如您要尝试将它转换成一个或多个等同的明显解时，您可将 **solve()** 应用到蕴含解中去。

当利用教科书或手动求解来比较您的结果时，应知道，利用不同方式在运算中不同的点上引出的任意常数，也许会产生不同的普通解。

备注: 欲键入原始符号(')，可按下 2nd [′]。

```
deSolve(y''+2y'+y=x^2,x,y) ENTER
```

$$y=(@1\cdot x+@2)\cdot e^{-x}+x^2-4\cdot x+6$$

```
right(ans(1))→temp ENTER
```

$$(@1\cdot x+@2)\cdot e^{-x}+x^2-4\cdot x+6$$

```
d(temp,x,2)+2*d(temp,x)+temp-x^
2 ENTER                      0
DelVar temp ENTER          Done
```

```
deSolve(y'=(cos(y))^2*x,x,y)
ENTER
```

$$\tan(y)=\frac{x^2}{2}+@3$$

```
solve(ans(1),y) ENTER
```

$$y=\tan^{-1}\left(\frac{x^2+2\cdot @3}{2}\right)+@n1\cdot\pi$$

备注: 欲键入符号 @，可按下:

TI-89: ◆ STO▸

TI-92 Plus: 2nd R

```
ans(1)|@3=c-1 and @n1=0 ENTER
```

$$y=\tan^{-1}\left(\frac{x^2+2\cdot(c-1)}{2}\right)$$

---

deSolve(*第一红第二顺序Ode* **and** *初始条件*, *自变元*,*因变元***)** ⇒ *特殊解*

对应能满足*第一顺序Ode*和*初始条件*的特殊解。这比起定义普通解，替换起始值，解出任意常数然后将该值代入普通解中较常用。

*起始条件*是以下形式的方程:

因变元 (*起始自变元值*) = *起始因变元值*

```
sin(y)=(y*e^(x)+cos(y))y'→ode
ENTER
```

$$\sin(y)=(e^x\cdot y+\cos(y))\cdot y'$$

```
deSolve(ode and
y(0)=0,x,y)→soln ENTER
```

$$\frac{-(2\cdot\sin(y)+y^2)}{2}=-(e^x-1)\cdot e^{-x}\cdot\sin(y)$$

*起始自变元值*和*起始因变元值*可为不带已储存值的变元 x0 和 y0。蕴含微分法对蕴含解的检查有帮助。

```
soln|x=0 and y=0 ENTER          true
d(right(eq)-left(eq),x)/
(d(left(eq)-right(eq),y))
→impdif(eq,x,y) ENTER
                                Done
ode|y'=impdif(soln,x,y) ENTER
                                true
DelVar ode,soln ENTER           Done
```

deSolve(*第二顺序Ode* **and** *初始条件1* **and** *初始条件2*, *自变元*, *因变元*) ⇒ *特殊解*

对应能满足*第二顺序Ode*，带有因变元的指定值，其第一导数在一个点上的特殊解。

deSolve(y''=y^(⁻1/2) and y(0)=0 and y'(0)=0,t,y) ENTER

$$\frac{2\cdot y^{3/4}}{3}=t$$

solve(ans(1),y) ENTER

$$y=\frac{2^{2/3}\cdot(3\cdot t)^{4/3}}{4} \text{ and } t\geq 0$$

对于*初始条件1*，请使用以下形式:

因变元 (*初始自变元值*) = *初始因变元值*

对于*初始条件2*，请使用以下形式:

因变元' (*初始自变元值*) = *初始导数值*

deSolve(*第二顺序Ode* **and** *边界条件1* **and** *边界条件2*, *自变元*, *因变元*) ⇒ *特殊解*

对应能满足*第二顺序Ode*，在两个不同点上有指定值的特殊解。

deSolve(w''-2w'/x+(9+2/x^2)w= x*e^(x) and w(π/6)=0 and w(π/3)=0,x,w) ENTER

$$w=\frac{e^{\frac{\pi}{3}}\cdot x\cdot\cos(3\cdot x)}{10}-\frac{e^{\frac{\pi}{6}}\cdot x\cdot\sin(3\cdot x)}{10}+\frac{x\cdot e^{x}}{10}$$

det() MATH/Matrix 菜单

det(*平方矩阵***[, ***误差***]) ⇒ ***表达式***

对应*平方矩阵*的行列式。

作为选项，任何矩阵元素会被当作零来处理，如果其绝对值小于*误差*的话。这种误差是用于如矩阵带有浮点输入项和不含任何未赋值的符号变元时。否则，*误差*就会被忽略掉。

- 如您使用 ◆ ENTER 或将模式设定成 Exact/Approx=APPROXIMATE，那么运算就会利用浮点算法进行。
- 如*误差*被省略或没有使用，默认的误差就会被计算成:

  5E⁻14 * **max(dim(***平方矩阵***)) * rowNorm(***平方矩阵***)**

```
det([a,b;c,d]) ENTER        a·d−b·c
det([1,2;3,4]) ENTER              -2
det(identity(3)−x*[1,-2,3;
-2,4,1;-6,-2,7]) ENTER
        -(98·x³−55·x²+12·x−1)
[1E20,1;0,1]→mat1          [1.E20 1; 0 1]
det(mat1) ENTER                    0
det(mat1,.1) ENTER           1.E20
```

diag() MATH/Matrix 菜单

diag(*数组***) ⇒ ***矩阵***
diag(*行矩阵***) ⇒ ***矩阵***
diag(*列矩阵***) ⇒ ***矩阵***

对应在自身的主对角线内的变数数组或矩阵带有数值的矩阵。

```
diag({2,4,6}) ENTER        [2 0 0; 0 4 0; 0 0 6]
```

diag(*平方矩阵***) ⇒ ***行矩阵***

对应含有*平方矩阵*主对角线元素行矩阵。

*行矩阵*必须为平方。

```
[4,6,8;1,2,3;5,7,9] ENTER  [4 6 8; 1 2 3; 5 7 9]
diag(ans(1)) ENTER              [4 2 9]
```

Dialog CATALOG

Dialog
块
EndDlog

当执行了程序后，生成对话框。

*块*可为“:”字符分隔的单陈述或系列陈述。程序编辑器菜单项 F3 I/O, 1:Dialog 中有效的*块*选项计有 1:Text、2:Request、4:DropDown 和 7:Title。

对话框中的变元可为给出的数值，这样就可作为默认值(或起始值)显示出来。如按下了 ENTER，变元就会从对话框中更新，变元 ok 就会设定成 1。如按下了 ESC，它的变元就不会得到更新，系统变元 ok 就会设定成零。

程序列表:

```
:Dlogtest()
:Prgm
:Dialog
:Title    "This is a dialog
box"
:Request  "Your name",Str1
:Dropdown "Month you were
born",
  seq(string(i),i,1,12),Var1
:EndDlog
:EndPrgm
```

dim() MATH/Matrix/Dimensions 菜单

dim(*数组***) ⇒ ***整数***

对应*数组*的维数。

```
dim({0,1,2}) ENTER                 3
```

dim(*矩阵***) ⇒ ***数组***

对应作为两个元素数组{行, 列}的*矩阵*维数。

```
dim([1,-1,2;-2,3,5]) ENTER  {2 3}
```

dim(*字符串***)** ⇒ *整数*

对应字符串*字符串*中含有的字符数。

```
dim("Hello") ENTER          5
dim("Hello"&" there") ENTER  11
```

Disp CATALOG

Disp [*表达式或字符串1*] [,*表达式或字符串2*] **...**

显示程序 I/O (输入 / 输出)屏的当前内容。如指定了一个或多个*表达式或字符串*，每一表达式或字符串就会显示在程序 I/O 屏的不同行之中。

表达式可包含换算操作，如 ▸**DD** 和 ▸**Rect**。您也可使用 ▸ 算子执行单位和数字基的换算。

如 Pretty Print = ON，表达式就会以工整印刷体显示出来。

从程序 I/O 屏上，您可按下 F5 来显示主屏幕，或在程序中按下 **DispHome** 来显示。

```
Disp "Hello" ENTER          Hello
Disp cos(2.3) ENTER         ⁻.666…
{1,2,3,4}→L1 ENTER
Disp L1 ENTER          {1  2  3  4}
Disp 180_min▸_hr ENTER     3.·_hr
```

备注: 欲键入下划线(_)，可按下:
TI-89: ◆ [_]
TI-92 Plus: 2nd [_]
欲键入 ▸，可按下 2nd [▸]。

DispG CATALOG

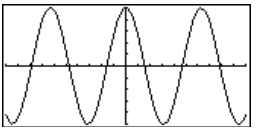
DispG

显示图形屏幕的当前内容。

在函数作图模式下:

程序段:

```
  ⋮
:5*cos(x)→y1(x)
:⁻10→xmin
:10→xmax
:⁻5→ymin
:5→ymax
:DispG
  ⋮
```



DispHome CATALOG

DispHome

显示主屏幕的当前内容。

程序段:

```
  ⋮
:Disp "The result is: ",xx
:Pause "Press Enter to quit"
:DispHome
:EndPrgm
```

DispTbl CATALOG

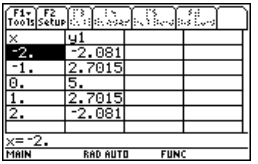
DispTbl

显示表格屏的当前内容。

备注: 光标控制盘可用于滚动。如在程序之中，可按下 ESC 或 ENTER 来假设执行。

5*cos(x)→y1(x) ENTER

DispTbl ENTER



▸DMS MATH/Angle 菜单

表达式▸**DMS**
数组▸**DMS**
矩阵▸**DMS**

将变数当成角演算并显示等同的 DMS (*DDDDDD°MM′SS.ss″*)数字。请参阅第 215 页 DMS (度、分、秒)格式中的 °、'、"。

备注: 当在弧度模式下使用时，▸**DMS** 会从弧度转换成角度。如输入后跟随有一个度符号 (°)，那么转换就不会发生。您只可在输入行的结尾使用▸**DMS**。

在 Degree 角模式下:

45.371 ▸DMS ENTER　　45° 22'15.6"

{45.371,60} ▸DMS ENTER
{45° 22'15.6"　60° }

dotP() MATH/Matrix/Vector 选项菜单

dotP(*数组1***,** *数组2***)** ⇒ *表达式*

对应两数组的 "点"积。

dotP({a,b,c},{d,e,f}) ENTER
a·d + b·e + c·f

dotP({1,2},{5,6}) ENTER　　17

---

dotP(*向量1***,** *向量2***)** ⇒ *表达式*

对应两向量的 "点"积。

两者均必须为行向量或列向量。

dotP([a,b,c],[d,e,f]) ENTER
a·d + b·e + c·f

dotP([1,2,3],[4,5,6]) ENTER　　32

DrawFunc CATALOG

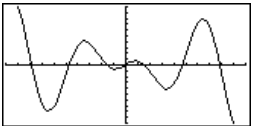
DrawFunc *表达式*

将*表达式*当函数画出，利用 x 作为自变元。

备注: 重作图形时会清除所有画出的项目。

在函数作图和 ZoomStd 窗口模式下:

DrawFunc 1.25x*cos(x) ENTER



DrawInv CATALOG

DrawInv *表达式*

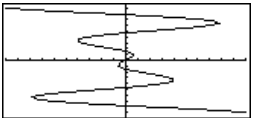
通过在 y 轴绘出 x 值和在 x 轴绘出 y 值来画出*表达式*的反函数。

x 是自变元。

备注: 重作图形时会清除所有画出的项目。

在函数作图和 ZoomStd 窗口模式下:

DrawInv 1.25x*cos(x) ENTER



DrawParm CATALOG

DrawParm *表达式1*, *表达式2* [, *t最小*] [, *t最大*] [, *t分级*]

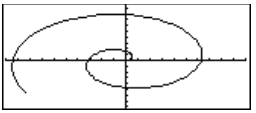
利用 t 作为自变元，画出参数方程有*表达式1*和*表达式2*。

t最小、*t最大*和 *t分级*的默认值是窗口变元 tmin、tmax 和 tstep 的当前设置。指定数值不会改变窗口设置。如当前的作图模式不是参数，那么就需要三个变数。

备注: 重作图形时会清除所有画出的项目。

在函数作图和 ZoomStd 窗口模式下:

```
DrawParm
t*cos(t),t*sin(t),0,10,.1 ENTER
```



DrawPol CATALOG

DrawPol *表达式*[, θ*最小*] [, θ*最大*] [, θ*分级*]

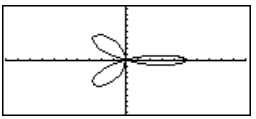
画出*表达式*的极坐标图，利用 θ 作为自变元。

θ*最小*、θ*最大* 和 θ*分级* 的默认值是窗口变元 θmin、θmax 和 θstep 的当前设置。指定数值不会改变窗口设置。如当前的作图模式不是参数，那么就需要三个变数。

备注: 重作图形时会清除所有画出的项目。

在函数作图和 ZoomStd 窗口模式下:

```
DrawPol 5*cos(3*θ),0,3.5,.1
ENTER
```



DrawSlp CATALOG

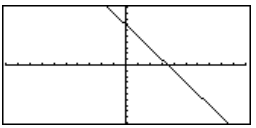
DrawSlp *x1*, *y1*, *斜率*

显示图形和利用程式 y-y1=slope·(x-x1)来进行画线。

备注: 重作图形时会清除所有画出的项目。

在函数作图和 ZoomStd 窗口模式下:

```
DrawSlp 2,3,-2 ENTER
```



DropDown CATALOG

DropDown *标题字符串*, **{***项目1字符串*, *项目2字符串*, **...}**, *变元名称*

显示带有*标题字符串*名称和含有项目 **1:***项目1字符串*、**2:***项目2字符串*等等的下拉菜单。**DropDown** 必须 **Dialog...EndDlog** 块之内。

如*变元名称* 已经存在并带有项目范围内的数值，关联的项目就会以默认选择显示出来。否则，菜单的第一项就是默认选择。

当您从菜单选择一个项时，该项所对应的数字就被存入*变元名称*中。(如有需要，**DropDown** 可创建*变元名称*。)

请参阅 **Dialog** 程序列表中的例子。

DrwCtour CATALOG

DrwCtour *表达式*
DrwCtour *数组*

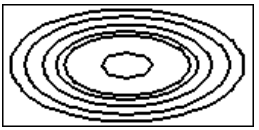
利用由*表达式*或*数组*指定的 **z** 值在 3D 图形上画轮廓线。3D 作图模式必须为已经设定的。**DrwCtour** 可自动将作图格式样式设定成 CONTOUR LEVELS。

作为默认，图形会自动包含由窗口变元 ncontour 指定的相等间隔数的轮廓线。**DrwCtour** 可画出默认值以外的轮廓线。

欲关闭默认的轮廓线，可将 ncontour 设定成零，也可利用窗口屏或将 0 存入 ncontour 系统变元来设定。

在 3D 作图模式下:

```
(1/5)x^2+(1/5)y^2-10→z1(x,y)
ENTER
                                Done
-10→xmin:10→xmax ENTER            10
-10→ymin:10→ymax ENTER            10
-10→zmin:10→zmax ENTER            10
0→ncontour ENTER                   0
DrwCtour {-9,-4.5,-3,0,4.5,9}
ENTER
```



- 利用光标来改变观察角度。按下 0 (零) 来返回原始外观。
- 欲在不同的图形格式样式之间切换，可按下:

TI-89: I **TI-92 Plus**: F

- 按下 X、Y 或 Z 来向下查看对应的坐标轴。

E **TI-89**: EE 键 **TI-92 Plus:** 2nd [EE] 键

*尾数***E***指数*

以科学记数法输入一个数字。数字会被当作*尾数* × 10*指数*演算。

提示: 如您想输入 10 的幂又不想造成小数结果结果，请使用 10^*整数*。

```
2.3E4 ENTER                    23000.
2.3E9+4.1E15 ENTER             4.1E15
3*10^4 ENTER                   30000
```

e^() **TI-89**: ◆ [e^x] 键 **TI-92 Plus:** 2nd [e^x] 键

e*^(表达式1***)** ⇒ *表达式*

对应增加成*表达式1*幂的 *e*。

备注: 在 TI-89 上，可按下 ◆ [e^x] 来显示 e^(与按下 alpha [E] 有所不同。在 TI-92 Plus 上，按下 2nd [e^x] 来显示 e^ 与存取字符和从 QWERTY 键盘进行有所不同。

您可以 $re^{i\theta}$ 极坐标形式输入一个复数。尽管如此，请仅在 Radian 角模式下使用此形式；它会在 Degree 角模式下造成 Domain error。

```
e^(1) ENTER                    e
e^(1.) ENTER                   2.718...
e^(3)^2 ENTER                  e^9
```

e*^(数组1***)** ⇒ *数组*

对应增加到*数组1*中每一元素的幂的 *e*。

```
e^({1,1.,0,.5}) ENTER
                 {e   2.718...   1   1.648...}
```

e^(*平方矩阵1* **)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵指数。这与计算增加到每一元素的幂的 *e* 不同。有关计算模式的信息，请参阅 **cos()**。

*平方矩阵1*必须为可对角的。结果一直会含有浮点数字。

e^([1,5,3;4,2,1;6,-2,1]) ENTER

$$\begin{bmatrix} 782.209 & 559.617 & 456.509 \\ 680.546 & 488.795 & 396.521 \\ 524.929 & 371.222 & 307.879 \end{bmatrix}$$

eigVc() MATH/Matrix 菜单

eigVc(*平方矩阵* **)** ⇒ *矩阵*

对应含有实数或复数*平方矩阵*特征向量的矩阵，其中结果里的每一列对应一个特征值。请记住，特征向量不是唯一的，它也许会受任何常数因子所调节。特征向量是标准化的，意思就是如 $V = [x_1, x_2, \ldots, x_n]$，那么：

$$\sqrt{x_1{}^2 + x_2{}^2 + \ldots + x_n{}^2} = 1$$

*平方矩阵*会首先利用类似的转移得到均衡，直至行和列规则最大限度地接近相同值位置。随后，*平方矩阵*会简化成上 Hessenberg 形式，并特征向量会通过 Schur 分解因子法计算出。

在 Rectangular 复数格式模式下：

[-1,2,5;3,-6,9;2,-5,7]→m1 ENTER

$$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix}$$

eigVc(m1) ENTER

$$\begin{bmatrix} -.800\ldots & .767\ldots & .767\ldots \\ .484\ldots & .573\ldots+.052\ldots\cdot i & .573\ldots-.052\ldots\cdot i \\ .352\ldots & .262\ldots+.096\ldots\cdot i & .262\ldots-.096\ldots\cdot i \end{bmatrix}$$

eigVl() MATH/Matrix 菜单

eigVl(*平方矩阵* **)** ⇒ *数组*

对应实数或复数*平方矩阵*的特征值数组。

*平方矩阵*会首先利用类似的转移得到均衡，直至行和列规则最大限度地接近相同值位置。随后，*平方矩阵*会简化成上 Hessenberg 形式，并特征向量会通过 Hessenberg 矩阵计算出来。

在 Rectangular 复数格式模式下：

[-1,2,5;3,-6,9;2,-5,7]→m1 ENTER

$$\begin{bmatrix} -1 & 2 & 5 \\ 3 & -6 & 9 \\ 2 & -5 & 7 \end{bmatrix}$$

eigVl(m1) ENTER

{-4.409… 2.204…+.763…·*i* 2.204…-.763…·*i*}

Else 请参阅第 138 页的 **If**。

ElseIf CATALOG 也请参阅第 138 页的 **If**。

If *布尔表达式1* **Then**
  块1
ElseIf *布尔表达式2* **Then**
  块2
  ⋮
ElseIf *布尔表达式N* **Then**
  块N
EndIf
  ⋮

ElseIf 可在程序分支中作程序指令之用。

程序段：

```
  ⋮
:If choice=1 Then
:  Goto option1
:  ElseIf choice=2 Then
:  Goto option2
:  ElseIf choice=3 Then
:  Goto option3
:  ElseIf choice=4 Then
:  Disp "Exiting Program"
:  Return
:EndIf
  ⋮
```

EndCustm 请参阅第 113 页的 **Custom**。

EndDlog

请参阅第 120 页的 **Dialog**。

EndFor

请参阅第 132 页的 **For**。

EndFunc

请参阅第 133 页的 **Func**。

EndIf

请参阅第 138 页的 **If**。

EndLoop

请参阅第 147 页的 **Loop**。

EndPrgm

请参阅第 161 页的 **Prgm**。

EndTBar

请参阅第 194 页的 **ToolBar** 。

EndTry

请参阅第 195 页的 **Try**。

EndWhile

请参阅第 198 页的 **While**。

entry()

CATALOG

entry() ⇒ *表达式*
entry(*整数***)** ⇒ *表达式*

对应主屏幕历史区域的输入行输入项。

如包含有*整数*，就是指明历史区域中有哪种输入项表达式。默认值是 1，也就是最新计算出的输入项。有效范围是从 1 到 99 且不能为表达式。

备注: 如上一输入项在主屏幕上仍然为突出显示，按下 ENTER 就等于执行 **entry(1)**。

在主屏幕上:

1+1/x ENTER $\frac{1}{x}+1$

1+1/entry(1) ENTER $2-\frac{1}{x+1}$

ENTER $\frac{1}{2\cdot(2\cdot x+1)}+3/2$

ENTER $5/3-\frac{1}{3\cdot(3\cdot x+2)}$

entry(4) ENTER $\frac{1}{x}+1$

exact()

MATH/Number 菜单

exact(*表达式1* **[,** *误差***])** ⇒ *表达式*
exact(*数组1* **[,** *误差***])** ⇒ *数组*
exact(*矩阵1* **[,** *误差***])** ⇒ *矩阵*

使用 Exact 模式算术与 Exact/Approx 模式设置无关，在有可能的时候，有理数与变数是等同的。

*误差*是指定换算的误差，默认值是 0 (零)。

exact(.25) ENTER 1/4

exact(.333333) ENTER $\frac{333333}{1000000}$

exact(.33333,.001) 1/3

exact(3.5x+y) ENTER $\frac{7\cdot x}{2}+y$

exact({.2,.33,4.125}) ENTER

$\{1/5\ \frac{33}{100}\ 33/8\}$

Exec CATALOG

Exec *字符串* [, *表达式1*] [, *表达式2*] **...**

执行由系列 Motorola 68000 运算码构成的*字符串*，这些编码象汇编语言程序的其它形式一样作用。如有需要，任选的*表达式*可让您通过一或多个变数到达程序。

更为详尽的信息，请访问 TI 网址: **education.ti.com**

警告: Exec 可给您存取微处理机的所有功能。请记住，您很容易会出现错误而造成计算器锁死和数据丢失。我们建议您在尝试使用 **Exec** 命令之前，请先备份计算器中储存的内容。

Exit CATALOG

Exit

退出当前的 **For**、**While** 或 **Loop** 块。

Exit 不允许超过三次的循环结构(**For**、**While** 或 **Loop**)。

程序列表:

```
:0→temp
:For i,1,100,1
:  temp+i→temp
:  If temp>20
:  Exit
:EndFor
:Disp temp
```

执行后 **temp** 的内容: 21

exp▸list() CATALOG

exp▸list(*表达式,变元* ⇒ *数组*

为被“or,” 分隔的且对应含有*变元*=*表达式*右边的方程的方程检查表达式。这点为您提取嵌在 **solve()**、**cSolve()**、**fMin()** 和 **fMax()** 函数结果内的解值提供了简便的途径。

备注: exp▸list() 不需要与 **zeros** 和 **cZeros()** 函数一起使用，这是因为它们直接对应解值的数组。

```
solve(x^2-x-2=0,x) ENTER   x=2 or
x=-1

exp▸list(solve(x^2-x-2=0,x),x)
ENTER
                          {-1  2}
```

expand() MATH/Algebra 菜单

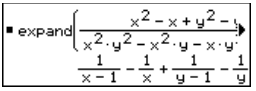
expand(*表达式1* [,*变元*]) ⇒ *表达式*
expand(*数组1* [,*变元*]) ⇒ *数组*
expand(*矩阵1* [,*变元*]) ⇒ *矩阵*

expand(*表达式1***)**对应所有变元展开后的*表达式1*。展开是多项式展开和有理表达式的部分分数展开。

expand() 的目的是将*表达式1*转移入一个和和 / 或不同的简单项中。而 **factor()** 目的是将*表*
*owa*dd172 简单因子的商中。

```
expand((x+y+1)^2) ENTER
      x²+2·x·y+2·x+y²+2·y+1

expand((x^2-x+y^2-y)/(x^2*y^2
  -x^2*y-x*y^2+x*y)) ENTER
```



expand(*表达式1,变元***)**对应*变元*展开后的*表达式*。类似*变元*的幂会被收集，项及它们的因子会作为主变元与*变元*一起储存。收集到的系数也许会有一些伴随的因式分解或展开。需要令表达式更易于理解时，对*变元*遗漏状况作出比较，有助于节省时间、内存和屏幕空间。

expand((x+y+1)^2,y) ENTER

$$y^2+2\cdot y\cdot(x+1)+(x+1)^2$$

expand((x+y+1)^2,x) ENTER

$$x^2+2\cdot x\cdot(y+1)+(y+1)^2$$

expand((x^2-x+y^2-y)/(x^2*y^2
-x^2*y-x*y^2+x*y),y) ENTER

$$\blacksquare\ \text{expand}\left(\frac{x^2-x+y^2-y}{x^2\cdot y^2-x^2\cdot y-x\cdot y^2}\right)\blacktriangleright \frac{1}{y-1}-\frac{1}{y}+\frac{1}{x\cdot(x-1)}$$

expand(ans(1),x) ENTER

$$\blacksquare\ \text{expand}\left(\frac{1}{y-1}-\frac{1}{y}+\frac{1}{x\cdot(x-1)}\right)\blacktriangleright \frac{1}{x-1}-\frac{1}{x}+\frac{1}{y\cdot(y-1)}$$

甚至当只有一个变元时，利用*变元*可使用于部分分数展开的分母因式分解更完整。

提示: 对于有理表达式，**propFrac()** 是快速但对于 **expand()** 的极端替代较少。

备注: 对于一个展开的分子在一个展开的分母上，也请参阅 **comDenom()**。

expand((x^3+x^2-2)/(x^2-2))
ENTER

$$\frac{2\cdot x}{x^2-2}+x+1$$

expand(ans(1),x) ENTER

$$\frac{1}{x-\sqrt{2}}+\frac{1}{x+\sqrt{2}}+x+1$$

expand(*表达式1,*[*变元*]**)** 也可分布对数和分数幂，与*变元*无关。对于对数和分数幂增加的分布，可能需要不等式强制，以保证某些因子为非负数。

expand(*表达式1,* [*变元*]**)** 也可分布绝对值 **sign()** 和指数，与*变元*无关。

备注: 对于三角形的角和以及多角展开，也请参阅 **tExpand()**。

ln(2x*y)+√(2x*y) ENTER

$$\ln(2\cdot x\cdot y)+\sqrt{(2\cdot x\cdot y)}$$

expand(ans(1)) ENTER

$$\ln(x\cdot y)+\sqrt{2}\cdot\sqrt{(x\cdot y)}+\ln(2)$$

expand(ans(1))|y>=0 ENTER

$$\ln(x)+\sqrt{2}\cdot\sqrt{x}\cdot\sqrt{y}+\ln(y)+\ln(2)$$

sign(x*y)+abs(x*y)+e^(2x+y)
ENTER

$$e^{2\cdot x+y}+\text{sign}(x\cdot y)+|x\cdot y|$$

expand(ans(1)) ENTER

$$\text{sign}(x)\cdot\text{sign}(y)+|x|\cdot|y|+(e^x)^2\cdot e^y$$

expr() MATH/String 菜单

expr(*字符串***)** ⇒ *表达式*

对应作为一个表达式的*字符串*内含有的字符串，并立即执行之。

```
expr("1+2+x^2+x") ENTER        x²+x+3
expr("expand((1+x)^2)") ENTER
                           x²+2·x+1
"Define cube(x)=x^3"→funcstr
ENTER
              "Define cube(x)=x^3"
expr(funcstr) ENTER            Done
cube(2) ENTER                     8
```

ExpReg MATH/Statistics/Regressions 菜单

ExpReg *数组1*, *数组2* [, [*数组3*] [, *数组4*, *数组5*]]

计算指数回归并更新所有系统统计变元。

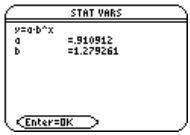
所有数组必须具有等同的维数，*数组5*除外。

数组1 代表 x 数组。
数组2 代表 y 数组。
数组3 代表频数。
数组4 代表类别代码。
数组5 代表类别包含数组。

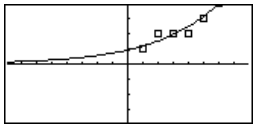
备注: *数组1*到*数组4*必须为变元名称或 c1-c99 (上一数据变元中的列数是在数据 / 矩阵编辑器中示出)。*数组5*不能为变元名称，也不能是 c1-c99。

在函数作图模式下:

```
{1,2,3,4,5,6,7,8}→L1 ENTER
                          {1 2 ...}
{1,2,2,2,3,4,5,7}→L2 ENTER
                          {1 2 ...}
ExpReg L1,L2 ENTER            Done
ShowStat ENTER
```




```
ENTER
Regeq(x)→y1(x) ENTER          Done
NewPlot 1,1,L1,L2 ENTER       Done
◆ [GRAPH]
```



factor() MATH/Algebra 菜单

factor(*表达式1*[, *变元*]**)** ⇒ *表达式*
factor(*数组1*[, *变元*]**)** ⇒ *数组*
factor(*矩阵1*[, *变元*]**)** ⇒ *矩阵*

factor(*表达式1***)**是相对于所有变元在一个公分母之上的因式分解的*表达式1*。

*表达式1*会尽最大可能接近地因式分解成一次关系因子，甚至如插入新的非实数。如您想相应于一个以上的变元进行因式分解，此替代方式是十分合适的。

```
factor(a^3*x^2-a*x^2-a^3+a)
ENTER
  a·(a-1)·(a+1)·(x-1)·(x+1)
factor(x^2+1) ENTER           x²+1
factor(x^2-4) ENTER (x-2)·(x+2)
factor(x^2-3) ENTER           x²-3
factor(x^2-a) ENTER           x²-a
```

factor(*表达式1,变元***)** 是相对于*变元*因式分解出的*表达式1*。

*表达式1*会尽最大可能接近地因式分解成*变元*中也许带有非实数常数的一次因子，甚至会插入无理常数或在其它变元中是无理的子表达式。

因子和它们的项会作为主变元与*变元*一起储存。*变元*的类似幂会被收集在每一因子内。如因式分解仅需相应于该变元，以及您愿意接受任何其它变元中的无理表达式来增加相对于*变元*的因式分解，请包含*变元*。可能会有相对于其它变元的意想不到的因式分解出现。

```
factor(a^3*x^2-a*x^2-a^3+a,x) ENTER
```

$$a\cdot(a^2-1)\cdot(x-1)\cdot(x+1)$$

```
factor(x^2-3,x) ENTER
```

$$(x+\sqrt{3})\cdot(x-\sqrt{3})$$

```
factor(x^2-a,x) ENTER
```

$$(x+\sqrt{a})\cdot(x-\sqrt{a})$$

对于 Exact/Approx 模式的 AUTO 设置，包括*变元*，允许带有浮点系数的近似值，其中无理系数不能简明地按照内建函数明显表示出来。甚至当只有一个变元时，包括*变元* 在内也许会产生更为完整的因式分解。

```
factor(x^5+4x^4+5x^3-6x-3) ENTER
```

$$x^5+4\cdot x^4+5\cdot x^3-6\cdot x-3$$

备注: 当 **factor()** 过慢或将内存耗用时将不完全的因式分解存档的快捷途径，也请参阅 **comden()**。

```
factor(ans(1),x) ENTER
```

$$(x-.964\ldots)\cdot(x+.611\ldots)\cdot(x+2.125\ldots)\cdot(x^2+2.227\ldots\cdot x+2.392\ldots)$$

备注: 对于复数系数追踪一次因子的所有因式分解途径，也请参阅 **cFactor()**。

---

factor(*有理数***)**对应因式分解成质数的有理数。对于复合数，计算时间与第二大因子的数位成指数增长。例如，对一个 30 位的整数进行因式分解会需时一天以上，对一个 100 位的数进行因式分解会需时一个纪元。

```
factor(152417172689) ENTER
```

123457·1234577

备注: 欲停止(中断)运算，可按下 ON。

```
isPrime(152417172689) ENTER false
```

如您只想定义一个数字为质数，可使用 **isPrime()** 替代。尤其是在*有理数*不是质数和第二大因子有超过五个数位时会特别快速。

Fill　　MATH/Matrix 菜单

Fill *表达式, 矩阵变元* ⇒ *矩阵*

用*表达式*替换变元*矩阵变元*中的每一元素。

矩阵变元 必须为已经存在的。

```
[1,2;3,4]→amatrix ENTER
```

$$\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$$

```
Fill 1.01,amatrix ENTER
```

Done

```
amatrix ENTER
```

$$\begin{bmatrix}1.01 & 1.01\\1.01 & 1.01\end{bmatrix}$$

Fill *表达式, 数组变元* ⇒ *数组*

用*表达式*替换变元*数组变元*中的每一元素。

数组变元 必须为已经存在的。

```
{1,2,3,4,5}→alist ENTER
```

{1 2 3 4 5}

```
Fill 1.01,alist ENTER
```

Done

```
alist ENTER
```

{1.01 1.01 1.01 1.01 1.01}

floor()　　MATH/Number 菜单

floor(*表达式***)** ⇒ *整数*

对应小于变数的最大整数，次函数与 **int()** 完全相同。

变数可为实数或复数。

```
floor(⁻2.14) ENTER
```

⁻3.

floor(*数组1***)** ⇒ *数组*
floor(*矩阵1***)** ⇒ *矩阵*

对应每一元素的底的矩阵或数组。

备注: 也请参阅 **ceiling()** 和 **int()**。

```
floor({3/2,0,⁻5.3}) ENTER
                    {1 0 ⁻6.}
floor([1.2,3.4;2.5,4.8]) ENTER
                    [1. 3.
                     2. 4.]
```

fMax() MATH/Calculus 菜单

fMax(*表达式, 变元***)** ⇒ *布尔表达式*

对应为最大化*表达式*或定位其最小上限的*变元*指定候选值的布尔表达式。

利用"|" 算子解来限制区间和 / 或指定其它未定义变元的符号。

对于 Exact/Approx 模式的 PPROX 设置，**fMax()** 会反复搜索一个接近的本地最小值，这样做通常可加快速度，尤其是如您使用 "|" 算子来强制搜索准确地含有一个本地最小值的相应小区间。

备注: 也请参阅 **fMin()** 和 **max()**。

```
fMax(1-(x-a)^2-(x-b)^2,x)
ENTER
                    x = (a+b)/2
fMax(.5x^3-x-2,x) ENTER     x = ∞
fMax(.5x^3-x-2,x)|x≤1 ENTER
                    x = ⁻.816...
fMax(a*x^2,x) ENTER
  x = ∞ or x = ⁻∞ or x = 0 or a = 0
fMax(a*x^2,x)|a<0 ENTER     x = 0
```

fMin() MATH/Calculus 菜单

fMin(*表达式, 变元***)** ⇒ *布尔表达式*

对应为最小化*表达式*或定位其最大下限的*变元*指定候选值的布尔表达式。

利用"|" 算子解来限制区间和 / 或指定其它未定义变元的符号。

对于 Exact/Approx 模式的 APPROX 设置，**fMin()** 会反复搜索一个接近的本地最小值，这样做通常可加快速度，尤其是如您使用"|"算子来强制搜索准确地含有一个本地最小值的相应小区间。

备注: 也请参阅 **fMax()** 和 **min()**。

```
fMin(1-(x-a)^2-(x-b)^2,x)
ENTER
                    x = ∞ or x = ⁻∞
fMin(.5x^3-x-2,x)|x≥1 ENTER  x = 1
fMin(a*x^2,x) ENTER
  x = ∞ or x = ⁻∞ or x = 0 or a = 0
fMin(a*x^2,x)|a>0 and x>1 ENTER
                    x = 1.
fMin(a*x^2,x)|a>0 ENTER     x = 0
```

FnOff CATALOG

FnOff

去除当前图形模式的所有 Y= 函数选择。

在拆分屏幕的双图模式下，**FnOff** 只应用于活动图形上。

FnOff [1] [, 2] ... [,99]

去除当前图形模式的指定 Y= 函数选择。

在函数作图模式下:
FnOff 1,3 ENTER 可去除 y1(x) 和 y3(x) 的选择。

在参数作图模式下:
FnOff 1,3 ENTER 可去除 xt1(t)、yt1(t) xt3(t) 和 yt3(t) 的选择。

FnOn CATALOG

FnOn

选择为当前图形模式定义的所有 Y= 函数。

在拆分屏幕的双图模式下，**FnOn** 只应用于活动图形上。

FnOn [1] [, 2] ... [,99]

选择为当前图形模式指定的 Y= 函数。

备注: 在 3D 作图模式下，每次只能选择一个函数。FnOn 2 可选择 z2(x,y) 和去除任何以前选定的函数。在其它作图模式下，以前选定的函数不会受影响。

For CATALOG

For *变元*, *低*, *高*[, *分级*]
块
EndFor

为*变元*的每一数值反复执行*块*中的陈述，从*低*到*高*，按*分级*增加。

*变元*不能为系统变元。

*分级*可为正数或负数。默认值是 1。

*块*可为由“:”字符分隔的单陈述或系列陈述。

程序段:

```
  ⋮
:0→tempsum : 1→step
:For i,1,100,step
:  tempsum+i→tempsum
:EndFor
:Disp tempsum
  ⋮
```

知心后 tempsum 的内容: 5050

当 step 改变成 2 时，tempsum 所含的内容: 2500

format() MATH/String 菜单

format(*表达式*[, *格式字符串*]**)** ⇒ *字符串*

对应作为字符串(基于规格模板)的*表达式*。

*表达式*必须化简成数字。*格式字符串*是字符串并必须为这些形式: “F[*n*]”、“S[*n*]”、“E[*n*]”、“G[*n*][*c*]”，其中 [ ] 表示选项部分。

F[*n*]: 固定格式。*n* 是小数点后要显示出来的数位。

S[*n*]: 科学格式。*n* 是小数点后要显示出来的数位。

E[*n*]: 工程格式。*n* 是第一个有效数字后的位数。指数会调整成三的倍数，小数点会右移零、一或两个数位。

G[*n*][*c*]: 与固定格式相同，但也会将记数根左方的数位分隔入三的组内。*c* 是指定组的分隔字符并默认成逗号。如 *c* 是句号，记数根会被显示成一个逗号。

[R*c*]: 上述的任何特定符会有 R*c* 记数根旗作为后缀，其中 *c* 是一个指定替换记数根点的单字符。

```
format(1.234567,"f3") ENTER
                    "1.235"
format(1.234567,"s2") ENTER
                  "1.23E0"
format(1.234567,"e3") ENTER
                 "1.235E0"
format(1.234567,"g3") ENTER
                    "1.235"
format(1234.567,"g3") ENTER
                "1,234.567"
format(1.234567,"g3,r:") ENTER
                    "1:235"
```

fpart() MATH/Number 菜单

fpart(*表达式1***)** ⇒ *表达式*
fpart(*数组1***)** ⇒ *数组*
fpart(*矩阵1***)** ⇒ *矩阵*

对应变数的分数部分。

对于数组或矩阵，是对应元素的分数部分。

变数可为实数或复数。

```
fpart(⁻1.234) ENTER          ⁻.234

fpart({1, ⁻2.3, 7.003}) ENTER
                    {0  ⁻.3  .003}
```

Func CATALOG

Func
块
EndFunc

在多陈述定义中要求作为第一陈述。

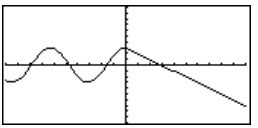
*块*可为由字符“:”分隔的单或系列陈述。

备注: when() 也可用于定义作图分段函数。

在函数作图模式下，定义一个分段函数:

```
Define g(x)=Func:If x<0 Then
  :Return 3*cos(x):Else:Return
  3-x:EndIf:EndFunc ENTER     Done

Graph g(x) ENTER
```



gcd() MATH/Number 菜单

gcd(*数字1***,** *数字2***)** ⇒ *表达式*

对应两变数的最大公约数。两分数的 **gcd** 是它们的分子除以它们的分母的 **gcd**。

在自动模式或近似模式下，分数浮点数的 **gcd** 为 1.0。

```
gcd(18,33) ENTER                3
```

---

gcd(*数组1***,** *数组2***)** ⇒ *数组*

对应*数组1*和*数组2*中相应元素的最大公分约数。

```
gcd({12,14,16},{9,7,5}) ENTER
                        {3 7 1}
```

---

gcd(*矩阵1***,** *矩阵2***)** ⇒ *矩阵*

对应*矩阵1*和*矩阵2*中相应元素的最大公分约数。

```
gcd([2,4;6,8],[4,8;12,16])
ENTER
                      [2  4]
                      [6  8]
```

Get CATALOG

Get *变元*

从链接端口取回 CBL™ (Calculator-Based Laboratory™) 或 CBR™ (Calculator-Based Ranger™)值并将其存入*变元*中。

程序段:

```
  ⋮
:Send {3,1,⁻1,0}
:For i,1,99
:  Get data[i]
:  PtOn i,data[i]
:EndFor
  ⋮
```

GetCalc CATALOG

GetCalc *变元*

从链接端口取回数值并将其存入*变元*中。这是单元对单元链接。

备注: 欲从其它单元为链接端口获取一个变元，可使用其它单元上的 2nd [VAR-LINK] 来选择并发送变元，或者在其它单元上执行 **SendCalc**。

程序段:

```
  ⋮
:Disp "Press Enter when ready"
:Pause
:GetCalc L1
:Disp "List L1 received"
  ⋮
```

getConfg() CATALOG

getConfg() ⇒ *数组对*

对应计算器属性表。属性名称显示在先，数值跟随其后。

TI-89:

```
getConfg() ENTER
        {"Product Name" "Advanced
            Mathematics Software"
    "Version" "2.00, 09/25/1999"
        "Product ID" "03-1-4-68"
        "ID #" "01012 34567 ABCD"
                "Cert. Rev. #" 0
              "Screen Width" 160
            "Screen Height" 100
              "Window Width" 160
              "Window Height" 67
               "RAM Size" 262132
               "Free RAM" 197178
           "Archive Size" 655360
          "Free Archive" 655340}
```

TI-92 Plus:

```
getConfg() ENTER
        {"Product Name" "Advanced
            Mathematics Software"
    "Version" "2.00, 09/25/1999"
        "Product ID" "01-1-4-80"
        "ID #" "01012 34567 ABCD"
                "Cert. Rev. #" 0
              "Screen Width" 240
            "Screen Height" 120
              "Window Width" 240
              "Window Height" 91
               "RAM Size" 262144
               "Free RAM" 192988
           "Archive Size" 720896
          "Free Archive" 720874}
```

备注: 您的屏幕也许会显示不同的属性值。只有您为计算器购买并安装了附加的软件后，Cert. Rev. # 属性方会显示出来。

getDenom() MATH/Algebra/Extract 菜单

getDenom(*表达式1***)** ⇒ *表达式*

将*表达式1*转入带有化简公分母的表达式内并对应其分母。

```
getDenom((x+2)/(y-3)) ENTER   y-3
getDenom(2/7) ENTER            7
getDenom(1/x+(y^2+y)/y^2) ENTER
                              x·y
```

getFold() CATALOG

getFold() ⇒ *名称字符串*

对应作为字符串的当前文件夹名称。

```
getFold() ENTER              "main"
getFold()→oldfoldr ENTER     "main"
oldfoldr ENTER               "main"
```

getKey() CATALOG

getKey() ⇒ *整数*

对应按下的键的键码。如没有按下任何按键，则对应 0。

前缀键 (shift 键 ⇧、第二功能键 2nd、选项键 ◆、字母键 alpha 和拖动手键 ⊡)不为它们本身所识别；然而，它们会改变跟随在后面的按键的键码。例如: ◆ X ≠ X ≠ 2nd X。

程序段:

```
:Disp
:Loop
:  getKey()→key
:  while key=0
:    getKey()→key
:  EndWhile
:  Disp key
:  If key = ord("a")
:  Stop
:EndLoop
```

getMode() CATALOG

getMode(*模式名称字符串***)** ⇒ *字符串*
getMode("ALL") ⇒ *数组字符串对*

如变数是特定模式名，则对应含有该模式的当前设置的字符串。

如变数是**"ALL"**，则对应含有所有模式设置的字符串对表。如您想在以后重新储存模式设置，您必须将 **getMode("ALL")** 的结果存入变元并利用 **setMode()** 来重新储存模式。

对应模式名称列表和可能的设置，请参阅 **setMode()**。

备注: 欲设置或返回有关 Unit System 模式的信息，请使用 **setUnits()** 或 **getUnits()** 而不是 **setMode()** 或 **getMode()**。

```
getMode("angle") ENTER   "RADIAN"
getMode("graph") ENTER "FUNCTION"
getMode("all") ENTER
              {"Graph" "FUNCTION"
      "Display Digits" "FLOAT 6"
                 "Angle" "RADIAN"
   "Exponential Format" "NORMAL"
         "Complex Format" "REAL"
   "Vector Format" "RECTANGULAR"
             "Pretty Print" "ON"
           "Split Screen" "FULL"
            "Split 1 App" "Home"
           "Split 2 App" "Graph"
         "Number of Graphs" "1"
            "Graph 2" "FUNCTION"
     "Split Screen Ratio"  "1,1"
          "Exact/Approx" "AUTO"
                  "Base" "DEC"}
```

备注: 您的屏幕也许会显示出不同的模式设置。

getNum() MATH/Algebra/Extract 菜单

getNum(*表达式1***)** ⇒ *表达式*

将*表达式1* 转入带有化简公分母的表达式内并对应其分子。

```
getNum((x+2)/(y-3)) ENTER      x + 2
getNum(2/7) ENTER                  2
getNum(1/x+1/y) ENTER          x + y
```

getType() CATALOG

getType(*变元***)** ⇒ *字符串*

对应指明*变元*数据类型的字符串。

如*变元*还没有定义，则对应字符串 "NONE"。

```
{1,2,3}→temp ENTER        {1 2 3}
getType(temp) ENTER       "LIST"
2+3i→temp ENTER             2 + 3i
getType(temp) ENTER       "EXPR"
DelVar temp ENTER           Done
getType(temp) ENTER       "NONE"
```

数据类型	变元内容
"ASM"	汇编语言程序
"DATA"	数据类型
"EXPR"	表达式(包括复数 / 任意值、未定义的、∞、⁻∞、TRUE、FALSE、pi、*e*)
"FUNC"	函数
"GDB"	作图数据库
"LIST"	数组
"MAT"	矩阵
"NONE"	不存在的变元
"NUM"	数目
"OTHER"	软件应用程序日后使用的各种数据类型
"PIC"	图象
"PRGM"	程序
"STR"	字符串
"TEXT"	文本类型
"VAR"	其它变元名称

getUnits() CATALOG

getUnits() ⇒ *数组*

对应含有所有类别的当前默认单位的字符串表，常数、温度、物质的量、光照密度和加速度除外。*表格*具有以下形式:

{"*系统*" "*类别1*" "*单位1*" "*类别2*" "*单位2*" ...}

第一个字符串给出系统(SI、ENG/US 或 CUSTOM)。字符串的随后对给出类别(如长度)及其默认单位(如米的 _m)。

欲设定默认单位，可使用 **setUnits()**。

```
getUnits() ENTER
{"SI" "Area" "NONE"
 "Capacitance" "_F"
 "Charge" "_coul"
 … }
```

备注: 您的屏幕也许会显示出不同的默认单位。

Goto　　CATALOG

Goto *标记名称*

将程序控制转移到*标记名称*。

*标记名称*必须在同一程序内利用 **Lbl** 指令定义好的。

程序段:

```
  ⋮
:0→temp
:1→i
:Lbl TOP
:  temp+i→temp
:  If i<10 Then
:   i+1→i
:   Goto TOP
:  EndIf
:Disp temp
  ⋮
```

Graph　　CATALOG

Graph *表达式1*[, *表达式2*] [, *变元1*] [, *变元2*]

Smart Graph 功能可利用当前的作图模式作出要求的表达式 / 函数。

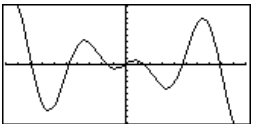
利用 **Graph** 或 **Table** 命令输入的表达式是指定从 1 开始增加函数的数量。利用按下 F4 Header 并显示出表格时可用的编辑功能，可修改或单独删除它们。当前选定的 Y= 函数会被忽略。

如您遗漏一个任选*变元*变数，**Graph** 就会使用当前作图模式的自变元。

备注: 并非所有任选的变数在所有模式下都是有效的，这是因为您永远不可能同时获得四个变数的原因。

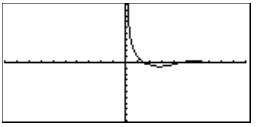
在函数作图模式和 ZoomStd 窗口下:

```
Graph 1.25a*cos(a),a ENTER
```



在参数作图模式和 ZoomStd 窗口下:

```
Graph
time,2cos(time)/time,time ENTER
```



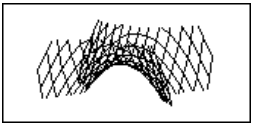
此指令的一些有效变动为:

函数作图	**Graph** *表达式*、*x*
参数作图	**Graph** *x表达式*、*y表达式*、*t*
极坐标作图	**Graph** *表达式*、θ
序列作图	不允许。
3D 作图	**Graph** *表达式*、*x*、*y*
微分方程	不允许。

备注: 利用 **ClrGraph** 来清除这些函数，或转到 Y= 编辑器或重新许用系统 Y= 函数。

在 3D 作图模式下:

```
Graph (v^2-w^2)/4,v,w ENTER
```



▸Hex　　MATH/Base 菜单

整数1 **▸Hex** ⇒ *整数*

将*整数1*转换成十六进制数字。十进制或十六进制数字一直分别带有 0b 或 0h 前缀。

```
256▸Hex ENTER                    0h100
0b111100001111▸Hex ENTER         0hF0F
```

┌ 是零，非字母 O ，后面跟随 b 或 h。
0b *十进制数字*
0h *十六进制数字*
└── 十进制数字可有多达 32 个数位，
十六进制有 8 个数位。

不带前缀时，*整数1*会被当作小数(基 10)处理，结果会以十六进制显示出，与 Base 模式无关。

如您输入一个远远超出 32 位十进制形式表示的小数整数，就会使用对称模运算来将数值带入合适的范围。

identity() MATH/Matrix 菜单

identity(*表达式***)** ⇒ *矩阵*

对应带有*表达式*维数的识别矩阵。

*表达式*必须计算成一个正整数。

```
identity(4) ENTER
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

If CATALOG

If *布尔表达式*
陈述
EndIf

If *布尔表达式* **Then**
块

如 *布尔表达式*计算为真实的，在继续执行之前，会先执行单陈述的*陈述*或陈述*块*。

如*布尔表达式*计算为假的，就会继续执行而不会执行陈述或陈述块。

*块*可为被字符“:”分隔的单陈述或陈述序列。

程序段:

```
  ⋮
:If x<0
:Disp "x is negative"
  ⋮
 —or—
  ⋮
:If x<0 Then
:  Disp "x is negative"
:  abs(x)→x
:EndIf
  ⋮
```

If *布尔表达式* **Then** *块1*
Else *块2* **EndIf**

如*布尔表达式*计算为真实的，就会执行*块1*并随后跳过*块2*。

如*布尔表达式*计算为假的，就会跳过*块1*但执行*块2*。

*块1*和*块2*可为单陈述。

程序段:

```
  ⋮
:If x<0 Then
:  Disp "x is negative"
:    Else
:  Disp "x is positive or
zero"
:EndIf
  ⋮
```

If *布尔表达式1* **Then**
块1
ElseIf *布尔表达式2* **Then**
块2
⋮
ElseIf *布尔表达式N* **Then**
块N
EndIf

允许程序分支。如*布尔表达式1*计算是真实的，就会执行*块1*。如*布尔表达式1*计算是假的，就会执行*布尔表达式2*等。

程序段:

```
  ⋮
:If choice=1 Then
:  Goto option1
:  ElseIf choice=2 Then
:    Goto option2
:  ElseIf choice=3 Then
:    Goto option3
:  ElseIf choice=4 Then
:    Disp "Exiting Program"
:    Return
:EndIf
  ⋮
```

imag() MATH/Complex 菜单

imag(*表达式1***)** ⇒ *表达式*

imag(*表达式1***)**对应变数的虚数部分。

备注: 所有未定义的变元均会被当作实数变元处理。也请参阅 **real()**。

```
imag(1+2i) ENTER                2
imag(z) ENTER                   0
imag(x+iy) ENTER                y
```

imag(*数组1***)** ⇒ *数组*

对应元素虚数部分的数组。

```
imag({-3,4-i,i}) ENTER   {0 -1 1}
```

imag(*矩阵1***)** ⇒ *矩阵*

对应元素虚数部分的矩阵。

```
imag([a,b;ic,id]) ENTER   [0 0; c d]
```

Input CATALOG

Input

暂停程序，显示当前的作图模式并让您通过定位作图光标来更新变元 *xc* 和 *yc* (极坐标模式下也有 *rc* 和 θ*c*)。

当您按下 ENTER 时，程序会恢复。

程序段:

```
  ⋮
:◉Get 10 points from the Graph
  Screen
:For i,1,10
:  Input
:  xc→XLIST[i]
:  yc→YLIST[i]
:EndFor
  ⋮
```

Input [*提示字符串*,] *变元*

Input [*提示字符串*], *变元*可暂停程序，在程序 I/O 屏显示*提示字符串*，等待您输入表达式并将表达式存入*变元*中。

如您遗漏*提示字符串*，“?”就会被当作提示一样显示出来。

程序段:

```
  ⋮
:For i,1,9,1
:  "Enter x" & string(i)→str1
:  Input str1,#(right(str1,2))
:EndFor
  ⋮
```

InputStr CATALOG

InputStr [*提示字符串*,] *变元*

暂停程序，在程序 I/O 屏显示*提示字符串*，等待您输入回答，并将回答作为字符串存入*变元*中。

如您遗漏*提示字符串*，“?”就会被当作提示一样显示出来。

备注: Input 和 **InputStr** 之间的差别在于 **InputStr** 一直会将结果当成字符串储存，所以不需要符号“ ”。

程序段:

```
  ⋮
:InputStr "Enter Your Name",str1
  ⋮
```

inString() MATH/String 菜单

inString(*搜寻字符串*, *子字符串*[, *开始*]**)** ⇒ *整数*

对应*搜寻字符串*中字符的位置，由第一次出现字符串*子字符串*算起。

如含有*开始*，是指定搜寻开始的*搜寻字符串*内的字符位置。默认值 = 1 (*搜寻字符串*的第一个字符)。

如*搜寻字符串*不含*子字符串*或*开始*大于*搜寻字符串*的长度，就对应零。

```
inString("Hello there","the")
ENTER                                  7

"ABCEFG"→s1:If inString(s1,
"D")=0:Disp "D not found."
ENTER
                          D not found.
```

int() CATALOG

int(*表达式***)** ⇒ *整数*
int(*数组1***)** ⇒ *数组*
int(*矩阵1***)** ⇒ *矩阵*

对应小于或等于变数的最大整数。此函数与 **floor()** 完全相同。

变数可为实数或复数。

对于数组或矩阵，是对应每一元素的最大整数。

```
int(-2.5) ENTER                       -3.

int([-1.234,0,0.37]) ENTER
                           [-2. 0 0.]
```

intDiv() CATALOG

intDiv(*数字1*, *数字2***)** ⇒ *整数*
intDiv(*数组1*, *数组2***)** ⇒ *数组*
intDiv(*矩阵1*, *矩阵2***)** ⇒ *矩阵*

对应变数 1 标记的整数部分被变数 2 除。

对于数组和矩阵，是对应每一元素对的变数 1 标记的整数部分被变数 2 除。

```
intDiv(-7,2) ENTER                    -3

intDiv(4,5) ENTER                      0

intDiv({12,-14,-16},{5,4,-3})
ENTER
                            {2 -3 5}
```

integrate 请参阅第 211 页的 ∫()。

iPart() MATH/Number 菜单

iPart(*数字***)** ⇒ *整数*
iPart(*数组1***)** ⇒ *数组*
iPart(*矩阵1***)** ⇒ *矩阵*

对应变数的整数部分。

对于数组和矩阵，是对应每一个变元的整数部分。

变数可为实数或复数。

```
iPart(-1.234) ENTER                   -1.

iPart({3/2,-2.3,7.003}) ENTER
                          {1 -2. 7.}
```

isPrime() MATH/Test 菜单

isPrime(*数字***)** ⇒ *布尔常数表达式*

当*数字*是 ≥ 2 并只能被自己和 1 除时，指明真实和虚假。

如*数字*超过大约 306 数位并没有 ≤ 1021 的因子，**isPrime(***数字***)** 会显示一个错误信息。

如您只想定义*数字*为质数，请使用 **isPrime()** 而不要用 **factor()**。尤其是在*数字*不是质数和第二大因子有超过五个数位时会特别快速。

```
IsPrime(5) ENTER                    true
IsPrime(6) ENTER                   false
```

指定数字后求下一质数的函数:

```
Define nextPrim(n)=Func:Loop:
n+1→n:if isPrime(n):return n:
EndLoop:EndFunc ENTER              Done

nextPrim(7) ENTER                    11
```

Item CATALOG

Item *项目名称字符串*
Item *项目名称字符串*, *标记*

只有在 **Custom...EndCustm** 或 **ToolBar...EndTBar** 块内才有效。设置一个下拉菜单，让您将文本粘贴到光标的位置(**Custom**)上或分支成一个标记(**ToolBar**)。

备注: 在 **Custom** 块内不允许分支标记。

请参阅 **Custom** 的例子。

Lbl CATALOG

Lbl *标记名称*

在程序中用名称*标记名称*定义标记。

您可使用 **Goto** *标记名称*指令来马上将程序控制在标记后转移到指令中。

*标记名称*必须与变元名称一样符合相同的命名要求。

程序段:

```
  ⋮
:Lbl lbl1
:InputStr "Enter password",
str1
:If str1≠password
:  Goto lbl1
:Disp "Welcome to ..."
  ⋮
```

lcm() MATH/Number 菜单

lcm(*数字1***,** *数字2***)** ⇒ *表达式*
lcm(*数组1***,** *数组2***)** ⇒ *数组*
lcm(*矩阵1***,** *矩阵2***)** ⇒ *矩阵*

对应两变数的最小公倍数。两分数的 **lcm** 是它们分子的 **lcm** 被它们分母的 **gcd** 除。分数浮点数的 **lcm** 是它们的积。

对应两个数组或矩阵，是对应相应元素的最小公倍数。

```
lcm(6,9) ENTER                       18

lcm({1/3,-14,16},{2/15,7,5})
ENTER
                        {2/3 14 80}
```

left() MATH/String 菜单

left(*源字符串***[,** *数字***])** ⇒ *字符串*

对应串*源字符串*中含有的最左方的*数字*字符。

如您省略*数字*，则对应所有*源字符串*。

```
left("Hello",2) ENTER              "He"
```

---

left(*数组1***[,** *数字***])** ⇒ *数组*

对应*数组1*中含有的最左方的*数字*元素。

如您省略*数字*，则对应所有*数组1*。

```
left({1,3,-2,4},3) ENTER
                          {1 3 -2}
```

left(*比较***)** ⇒ *表达式*

对应方程或不等式的左方。

```
left(x<3) ENTER                x
```

limit() MATH/Calculus 菜单

limit(*表达式1***, *变元*, *点*[,*方向*])** ⇒ *表达式*
limit(*数组1***, *变元*, *点*[,*方向*])** ⇒ *数组*
limit(*矩阵1***, *变元*, *点*[, *方向*])** ⇒ *矩阵*

对应所要求的限制。

方向: 负数=从左起，正数=从右起，否则=两者。(如省略之，*方向*的默认是两者。)

```
limit(2x+3,x,5) ENTER          13
limit(1/x,x,0,1) ENTER         ∞
limit(sin(x)/x,x,0) ENTER      1
limit((sin(x+h)-sin(x))/h,h,0)
ENTER
                               cos(x)
limit((1+1/n)^n,n,∞) ENTER     e
```

在正数 ∞ 和在负数 ∞ 中的限制会一直转换成从有限方起的单方限制。

视情形而定，当 **limit()** 不能定义唯一的限制时，**limit()** 是对应自己或 undef。这并不一定是说唯一的限制不存在。undef 意味着结果会是带有有穷或无穷大小的未知数，或它是该数的输入项集。

limit() 是使用如 L'Hopital' 的规则，所以，这是不能定义的唯一限制。如*表达式1* 含有*变元*以外的未定义变元，您也许要强制它们获得更简明的结果。

```
limit(a^x,x,∞) ENTER           undef
limit(a^x,x,∞)|a>1 ENTER       ∞
limit(a^x,x,∞)|a>0 and a<1
ENTER                          0
```

限制会对舍入错误十分敏感。当计算限制时，在有可能情形下，应避免 Exact/Approx 模式的 APPROX 设置和近似数字。否则，原应是零或有穷大小的限制也许不然，以及原应是有穷非零大小的限制也不然。

Line CATALOG

Line *x开始*, *y开始*, *x结束*, *y结束*[, *画图模式*]

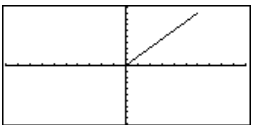
显示作图屏并在窗口坐标(*x开始*, *y开始*)和(*x结束*, *y结束*)之间画出、清除或反转线段，包括两个结束点。

如*画图模式* = 1，画线段(默认)。
如*画图模式* = 0，关闭线段。
如*画图模式* = ⁻1，将线段从打开到关闭或从关闭到打开(沿线段翻转象素)。

备注: 重作图形时会清除所有项目。也请参阅 **PxlLine**。

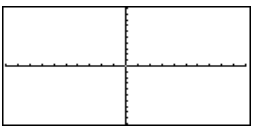
在 ZoomStd 窗口下画出线段并清除之。

```
Line 0,0,6,9 ENTER
```



TI-89: HOME
TI-92 Plus: ♦ [HOME]

```
Line 0,0,6,9,0 ENTER
```



LineHorz CATALOG

LineHorz *y* [, *画图模式*]

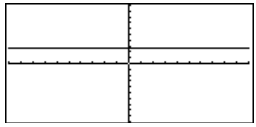
显示作图屏并在窗口位置 *y* 处画出、清除或反转水平线。

如*画图模式* = 1，画线段(默认)。
如*画图模式* = 0，关闭线段。
如*画图模式* = ⁻1，将线段从打开到关闭或从关闭到打开(沿线段翻转象素)。

备注: 重作图形时会清除所有项目。也请参阅 **PxlHorz**。

在 ZoomStd 窗口下:

LineHorz 2.5 [ENTER]



LineTan CATALOG

LineTan *表达式1*, *表达式2*

显示作图屏，并在指定的点处为*表达式1*画出切线。

*表达式1*是表达式或函数的名称，其中 x 被假定成自变元，*表达式2*是相切点的 x 值。

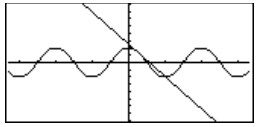
备注: 在示出的例子中，*表达式1*是分开作图的，**LineTan** 不能作出*表达式1*。

在函数作图模式和 ZoomTrig 窗口下:

Graph cos(x)

TI-89: [HOME]
TI-92 Plus: [◆] [HOME]

LineTan cos(x),π/4 [ENTER]



LineVert CATALOG

LineVert *x* [, *画图模式*]

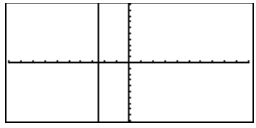
显示作图屏并在窗口位置 *x* 处画出、清除或反转垂线。

如*画图模式* = 1，画线段(默认)。
如*画图模式* = 0，关闭线段。
如*画图模式* = ⁻1，将线段从打开到关闭或从关闭到打开(沿线段翻转象素)。

备注: 重作图形时会清除所有项目。也请参阅 **PxlVert**。

在 ZoomStd 窗口下:

LineVert ⁻2.5 [ENTER]



LinReg MATH/Statistics/Regressions 菜单

LinReg *数组1*, *数组2*[, [*数组3*] [, *数组4*, *数组5*]]

计算线性回归并更新所有系统统计变元。

所有数组必须有相等的维数，*数组5*除外。

数组1 代表 x 数组。
数组2 代表 y 数组。
数组3 代表频数。
数组4 代表类别代码。
数组5 代表类别包含数组。

在函数作图模式下:

{0,1,2,3,4,5,6}→L1 [ENTER]

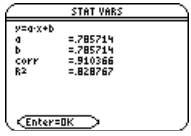
{0 1 2 ...}

{0,2,3,4,3,4,6}→L2 [ENTER]

{0 2 3 ...}

LinReg L1,L2 [ENTER] Done

ShowStat [ENTER]

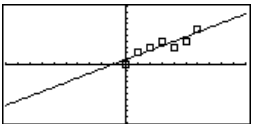


备注: *数组1*到*数组4*必须为变元名称或 c1-c99 (数据 / 矩阵编辑器显示的上一数据变元的列)。*数组5*不必为变元名称并不能为 c1-c99。

ENTER

```
Regeq(x)→y1(x) ENTER          Done
NewPlot 1,1,L1,L2 ENTER       Done
```

◆ [GRAPH]



list▸mat() MATH/List 菜单

list▸mat(*数组* **[,** *每行元素***])** ⇒ *矩阵*

对应以*数组*的元素逐行填充的矩阵。

如包含*每行元素*，是指定每行的元素数量。默认值是*数组* (一行)中的元素数量。

如*数组*不能填充结果矩阵，就会被加入零。

```
list▸mat({1,2,3}) ENTER      [1 2 3]
```

list▸mat({1,2,3,4,5},2) ENTER

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 0 \end{bmatrix}$$

Δlist() MATH/List 菜单

list(*数组1***)** ⇒ *数组*

对应含有*数组1*中连续元素之间差别的数组。*数组1*的每一元素被*数组1*的下一元素所减。结果数组一直是较原始*数组1*为短的一个元素。

```
Δlist({20,30,45,70}) ENTER
                    {10,15,25}
```

ln() TI-89: 2nd [LN] 键 TI-92 Plus: [LN] 键

ln(*表达式1***)** ⇒ *表达式*
ln(*数组1***)** ⇒ *数组*

对应变数的自然对数。

对于数组，是对应元素的自然对数。

```
ln(2.0) ENTER                .693...
```

如复数格式是 REAL:

```
ln({⁻3,1.2,5}) ENTER
          Error: Non-real result
```

如复数格式是 RECTANGULAR:

```
ln({⁻3,1.2,5}) ENTER
     {ln(3) + π·i   .182...   ln(5)}
```

---

ln(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵自然对数。这与计算每一元素的自然对数*不*同。有关计算方式的信息请参阅 **cos()**。

平方矩阵1 必须为可对角的。结果一直会含有浮点数字。

在 Radian 角模式和 Rectangular 复数格式模式下:

```
ln([1,5,3;4,2,1;6,⁻2,1]) ENTER
```

$$\begin{bmatrix} 1.831\ldots+1.734\ldots\cdot i & .009\ldots-1.490\ldots\cdot i & \ldots \\ .448\ldots-.725\ldots\cdot i & 1.064\ldots+.623\cdot i & \ldots \\ -.266\ldots-2.083\ldots\cdot i & 1.124\ldots+1.790\ldots\cdot i & \ldots \end{bmatrix}$$

LnReg　MATH/Statistics/Regressions 菜单

LnReg *数组1*, *数组2*[, [*数组3*] [, *数组4*, *数组5*]]

计算对数回归并更新所有系统统计变元。

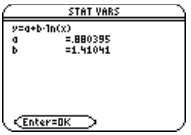
所有数组必须有相等的维数，*数组5*除外。

数组1 代表 x 数组。
数组2 代表 y 数组。
数组3 代表频数。
数组4 代表类别代码。
数组5 代表类别包含数组。

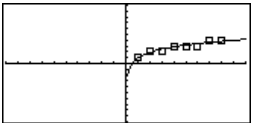
备注: *数组1*到*数组4*必须为变元名称或 c1-c99 (数据 / 矩阵编辑器显示的上一数据变元的列)。*数组5*不必为变元名称并不能为 c1-c99。

在函数作图模式下:

```
{1,2,3,4,5,6,7,8}→L1 ENTER
                          {1 2 3 ...}
{1,2,2,3,3,3,4,4}→L2 ENTER
                          {1 2 2 ...}
LnReg L1,L2 ENTER              Done
ShowStat ENTER
```




```
ENTER
Regeq(x)→y1(x) ENTER           Done
NewPlot 1,1,L1,L2 ENTER        Done
◆ [GRAPH]
```



Local　CATALOG

Local *变元1*[, *变元2*] [, *变元3*] ...

表明作为本地变元的指定*变元*。这些变元只有在程序或函数运算进行期间才退出，并当程序或函数完成执行时被删除。

备注: 本地变元可节省内存，这是由于它们只临时退出的原因。此外，它们也不影响现存的综合变元值。本地变元必须用于 **For** 循环和临时在多行函数中保存数值，这是由于在函数中修改综合函数是不允许的。

程序列表:

```
:prgmname()
:Prgm
:Local x,y
:Input "Enter x",x
:Input "Enter y",y
:Disp x*y
:EndPrgm
```

备注: x 和 y 在程序执行后不会退出。

Lock　CATALOG

Lock *变元1*[, *变元2*] ...

锁定指定的变元。这样可避免在该变元上没有首先使用松锁指令之前，意外地删除或改变该变元。

在右边的例子中，变元 L1 是被锁定的，故不能修改或删除之。

备注: 变元可使用 **Unlock** 指令松锁。

```
{1,2,3,4}→L1 ENTER        {1,2,3,4}
Lock L1 ENTER                  Done
DelVar L1 ENTER
Error: Variable is locked or protected
```

log() CATALOG

log(*表达式1***)** ⇒ *表达式*
log(*数组1***)** ⇒ *数组*

对应变数的基 10 对数。

对于数组，是对应元素的基 10 对数。

log(2.0) ENTER .301...

如复数格式模式是 REAL:

log({-3,1.2,5}) ENTER

Error: Non-real result

如复数格式模式是 RECTANGULAR:

log({-3,1.2,5}) ENTER

$$\left\{\frac{\ln(3)}{\ln(10)}+\frac{\pi}{\ln(10)}\cdot i \quad .079\ldots \quad \frac{\ln(5)}{\ln(10)}\right\}$$

---

log(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵基 10 对数。这与计算每一元素的基 10 对数*不*同。有关计算方式的信息请参阅 **cos()**。

平方矩阵1 必须为可对角的。结果一直含有浮点数字。

在 Radian 角模式和 Rectangular 复数格式模式下:

log([1,5,3;4,2,1;6,-2,1]) ENTER

$$\begin{bmatrix} .795\ldots+.753\ldots\cdot i & .003\ldots-.647\ldots\cdot i & \ldots \\ .194\ldots-.315\ldots\cdot i & .462\ldots+.270\cdot i & \ldots \\ -.115\ldots-.904\ldots\cdot i & .488\ldots+.777\ldots\cdot i & \ldots \end{bmatrix}$$

Logistic MATH/Statistics/Regressions 菜单

Logistic *数组1*, *数组2* [, [*重复*], [*数组3*] [, *数组4*, *数组5*]]

计算逻辑回归并更新所有系统统计变元。

所有数组必须有相等的维数，*数组5*除外。

数组1 代表 x 数组。
数组2 代表 y 数组。
数组3 代表频数。
数组4 代表类别代码。
数组5 代表类别包含数组。

*重复*是指定尝试解的最大时间数。如省略之，就会使用 64。传统而言，较大数值的结果会更精确但耗费的时间较长，反之也成立。

备注: *数组1*到*数组4*必须为变元名称或 c1-c99 (数据 / 矩阵编辑器显示的上一数据变元的列)。*数组5*不必为变元名称并不能为 c1-c99。

在函数作图模式下:

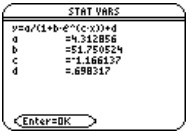
{1,2,3,4,5,6}→L1 ENTER {1 2 3 ...}

{1,1.3,2.5,3.5,4.5,4.8}→L2 ENTER

{1 1.3 2.5 ...}

Logistic L1,L2 ENTER Done

ShowStat ENTER



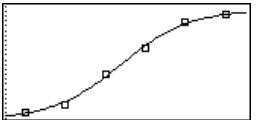
ENTER

regeq(x)→y1(x) ENTER Done

NewPlot 1,1,L1,L2 ENTER Done

◆ [GRAPH]

F2 9



Loop　　CATALOG

Loop
　块
EndLoop

重复执行*块*中的陈述。请记住，循环将会是无穷执行的，除非在*块*内执行了 **Goto** 或 **Exit** 指令。

*块*是由字符“:”分隔的陈述序列。

程序段:

```
  ⋮
:1→i
:Loop
:  Rand(6)→die1
:  Rand(6)→die2
:  If die1=6 and die2=6
:    Goto End
:  i+1→i
:EndLoop
:Lbl End
:Disp "The number of rolls is", i
  ⋮
```

LU　　MATH/Matrix 菜单

LU *矩阵*, *l矩阵名称*, *u矩阵名称*, *p矩阵名称*[, *tol*]

计算实数或复数*矩阵*的 Doolittle LU (上下)分解。下三角矩阵是储存在*l矩阵名称* 中，上三角矩阵是储存在*u矩阵名称* 中，置换矩阵(说明计算中交换的行)是储存在*p矩阵名称* 中。

l矩阵名称 ∗ *u矩阵名称* = *p矩阵名称* ∗ *矩阵*

作为选项，任何矩阵元素如其绝对值小于*误差*，可会被当作零来处理。此误差只有在矩阵带有浮点输入项且不含有任何未赋值的符号变元时才用到。否则，*误差*会被忽略。

- 如您使用 ◆ ENTER 或将模式设定成 Exact/Approx=APPROXIMATE，那么运算就会使用浮点算术进行。
- 如省略或没有使用*误差*，默认误差是按以下方式计算出来的:

  5E - 14 ∗ **max(dim(***矩阵***))** ∗ **rowNorm(***矩阵***)**

LU 因式分解算法使用带行互换的部分中枢。

[6,12,18;5,14,31;3,8,18]→m1 ENTER

$$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$$

LU m1,lower,upper,perm ENTER Done

lower ENTER

$$\begin{bmatrix} 1 & 0 & 0 \\ 5/6 & 1 & 0 \\ 1/2 & 1/2 & 1 \end{bmatrix}$$

upper ENTER

$$\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$$

perm ENTER

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

[m,n;o,p]→m1 ENTER

$$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$$

LU m1,lower,upper,perm ENTER Done

lower ENTER

$$\begin{bmatrix} 1 & 0 \\ \frac{m}{o} & 1 \end{bmatrix}$$

upper ENTER

$$\begin{bmatrix} o & p \\ 0 & n - \frac{m \cdot p}{o} \end{bmatrix}$$

perm ENTER

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

mat▸list() MATH/List 菜单

mat▸list(*矩阵***)** ⇒ *数组*

对应用*矩阵*中的元素填充的数组。元素是从 *矩阵*逐行复制的。

```
mat▸list([1,2,3]) ENTER      {1 2 3}
[1,2,3;4,5,6]→M1 ENTER
                          [1 2 3
                           4 5 6]
mat▸list(M1) ENTER   {1 2 3 4 5 6}
```

max() MATH/List 菜单

max(*表达式1***, *表达式2***)** ⇒ *表达式*
max(*数组1***, *数组2***)** ⇒ *数组*
max(*矩阵1***, *矩阵2***)** ⇒ *矩阵*

对应变数的两个最大值。如变数是两个数组或矩阵，则对应含有每一相应元素对最大值的数组或矩阵。

```
max(2.3,1.4) ENTER          2.3
max({1,2},{⁻4,3}) ENTER     {1 3}
```

---

max(*数组***)** ⇒ *表达式*

对应*数组*中的最大元素。

```
max({0,1,⁻7,1.3,.5}) ENTER    1.3
```

---

max(*矩阵1***)** ⇒ *矩阵*

对应含有*矩阵1*中每列最大值的行向量。

备注: 也请参阅 fMax() 和 min()。

```
max([1,⁻3,7;⁻4,0,.3]) ENTER
                          [1 0 7]
```

mean() MATH/Statistics 菜单

mean(*数组***[, *频数数组*])** ⇒ *表达式*

对应*数组*中元素的平均值。

每个*频数数组*元素是计算*数组*中相应元素的连续出现次数。

```
mean({.2,0,1,⁻.3,.4}) ENTER    .26
mean({1,2,3},{3,2,1}) ENTER    5/3
```

---

mean(*矩阵1***[, *频数矩阵*])** ⇒ *矩阵*

对应*矩阵1*中所有列的平均值的行向量。

每个*频数矩阵*元素是计算*矩阵1*中相应元素的连续出现次数。

在向量格式直角模式下:

```
mean([.2,0;-1,3;.4,-.5]) ENTER
                 [-.133... .833...]
mean([1/5,0;-1,3;2/5,-1/2])
ENTER
                   [⁻2/15  5/6]
mean([1,2;3,4;5,6],[5,3;4,1;
6,2]) ENTER        [47/15, 11/3]
```

median() MATH/Statistics 菜单

median(*数组***)** ⇒ *表达式*

对应*数组1*中元素的中位数。

```
median({.2,0,1,⁻.3,.4}) ENTER   .2
```

---

median(*矩阵1***)** ⇒ *矩阵*

对应*矩阵1*中含有列中位数的行向量。

备注: 数组或矩阵中的所有输入项必须化简成数字。

```
median([.2,0;1,⁻.3;.4,⁻.5])
ENTER
                       [.4 ⁻.3]
```

MedMed　MATH/Statistics/Regressions 菜单

MedMed *数组1*, *数组2*[, [*数组3*] [, *数组4*, *数组5*]]

计算中位数-中位数线回归并更新所有系统统计变元。

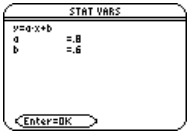
所有数组必须有相等的维数，*数组5*除外。

数组1 代表 x 数组。
数组2 代表 y 数组。
数组3 代表频数。
数组4 代表类别代码。
数组5 代表类别包含数组。

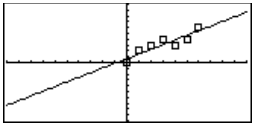
备注: *数组1*到*数组4*必须为变元名称或 c1-c99 (数据 / 矩阵编辑器显示的上一数据变元的列)。*数组5*不必为变元名称并不能为 c1-c99。

在函数作图模式下:

```
{0,1,2,3,4,5,6}→L1 ENTER {0 1 2 …}
{0,2,3,4,3,4,6}→L2 ENTER {0 2 3 …}
MedMed L1,L2 ENTER              Done
ShowStat ENTER
```




```
ENTER
Regeq(x)→y1(x) ENTER            Done
NewPlot 1,1,L1,L2 ENTER         Done
◆ [GRAPH]
```



mid()　MATH/String 菜单

mid(*源字符串*, *开始*[, *记数*]**)** ⇒ *字符串*

对应来自*源字符串*的*记数*字符，以字符数*开始*开头。

如省略了*记数*或大于*源字符串*的维数，对应来自*源字符串* 的所有字符，以字符数*开始*开头。

*记数*必须 ≥ 0，如*记数* = 0，则对应一个空的字符串。

```
mid("Hello there",2) ENTER
                    "ello there"
mid("Hello there",7,3) ENTER
                           "the"
mid("Hello there",1,5) ENTER
                         "Hello"
mid("Hello there",1,0) ENTER
                              ""
```

---

mid(*源数组*, *开始* [, *记数*]**)** ⇒ *数组*

对应来自*源数组*的*记数*元素，以元素数*开始*开头。

如省略了*记数*或大于*源数组*的维数，对应来自*源数组* 的所有元素，以元素数*开始*开头。

*记数*必须 ≥ 0，如*记数* = 0，则是对应一个空的数组。

```
mid({9,8,7,6},3) ENTER      {7 6}
mid({9,8,7,6},2,2) ENTER    {8 7}
mid({9,8,7,6},1,2) ENTER    {9 8}
mid({9,8,7,6},1,0) ENTER       {}
```

---

mid(*源字符串数组*, *开始*[, *记数*]**)** ⇒ *数组*

对应来自*源字符串数组*的字符数组的*记数*字符，以元素数*开始*。

```
mid({"A","B","C","D"},2,2)
ENTER
                  {"B" "C"}
```

min() MATH/List 菜单

min(*表达式1*, *表达式2***)** ⇒ *表达式*
min(*数组1*, *数组2***)** ⇒ *数组*
min(*矩阵1*, *矩阵2***)** ⇒ *矩阵*

对应变数的两个最小值。如变数是两个数组或矩阵，则对应含有每一相应元素对最小值的数组或矩阵。

```
min(2.3,1.4) ENTER                    1.4
min({1,2},{-4,3}) ENTER          {-4 2}
```

---

min(*数组***)** ⇒ *表达式*

对应*数组*中的最小元素。

```
min({0,1,-7,1.3,.5}) ENTER            -7
```

---

min(*矩阵1***)** ⇒ *矩阵*

对应含有*矩阵1*中每列最小值的行向量。

备注: 也请参阅 fMin() 和 max()。

```
min([1,-3,7;-4,0,.3]) ENTER
                        [-4 -3 .3]
```

mod() MATH/Number 菜单

mod(*表达式1*, *表达式2***)** ⇒ *表达式*
mod(*数组1*, *数组2***)** ⇒ *数组*
mod(*矩阵1*, *矩阵2***)** ⇒ *矩阵*

是对应作为恒等式定义的第一变数对模第二变数:

mod(x,0) ≡ x
mod(x,y) ≡ x− y floor(x/y)

当第二变数是非零时，结果在该变数中是周期的。结果会是零或带有象第二变数相同的标记。

如变数是两个数组或两个矩阵，则对应含有每一相应元素对对模的数组或矩阵。

备注: 也请参阅 remain()。

```
mod(7,0) ENTER                        7
mod(7,3) ENTER                        1
mod(-7,3) ENTER                       2
mod(7,-3) ENTER                      -2
mod(-7,-3) ENTER                     -1
mod({12,-14,16},{9,7,-5}) ENTER
                          {3 0 -4}
```

MoveVar CATALOG

MoveVar *变元*, *旧文件夹*, *新文件夹*

将*变元*从*旧文件夹*移动到*新文件夹*。如*新文件夹*不存在，**MoveVar** 就会创建之。

```
{1,2,3,4}→L1 ENTER          {1 2 3 4}
MoveVar L1,Main,Games ENTER   Done
```

mRow() MATH/Matrix/Row ops 菜单

mRow(*表达式*, *矩阵1*, *索引***)** ⇒ *矩阵*

对应带有*矩阵1*的行*索引*中每一元素乘以*表达式*的*矩阵1*副本。

```
mRow(-1/3,[1,2;3,4],2) ENTER
                    [ 1   2
                     -1  -4/3 ]
```

mRowAdd() MATH/Matrix/Row ops 菜单

mRowAdd(*表达式*, *矩阵1*, *索引1*, *索引2***)** ⇒ *矩阵*

对应带有*矩阵1*的行*索引2*中每一元素被以下项目替换的*矩阵1*副本:

表达式 × 行*索引1* + 行*索引2*

```
mRowAdd(-3,[1,2;3,4],1,2) ENTER
                    [ 1   2
                      0  -2 ]
mRowAdd(n,[a,b;c,d],1,2) ENTER
                    [ a        b
                      a·n+c    b·n+d ]
```

nCr() MATH/Probability 菜单

nCr(*表达式1*, *表达式2***)** ⇒ *表达式*

对于 *表达式1* ≥ *表达式2* ≥ 0 的整数 *表达式1*和 *表达式2*，**nCr()** 是 *表达式1*在 *表达式2*每次取得的组合数。(这也成为二项系数) 。两变数可为整数或符号表达式。

nCr(*表达式*, 0**)** ⇒ 1

nCr(*表达式*, *负整数***)** ⇒ 0

nCr(*表达式*, *正整数***)** ⇒ *表达式*·(*表达式*−1)...(*表达式*−*正整数*+1)/ *正整数*!

nCr(*表达式*, *非整数***)** ⇒ *表达式*!/ ((*表达式*−*非整数*)!·*非整数*!)

```
nCr(z,3)
```

$$\frac{z\cdot(z-2)\cdot(z-1)}{6}$$

```
ans(1)|z=5
```

10

```
nCr(z,c)
```

$$\frac{z!}{c!(z-c)!}$$

```
ans(1)/nPr(z,c)
```

$$\frac{1}{c!}$$

---

nCr(*数组1*, *数组2***)** ⇒ *数组*

对应基于两数组相应元素对的组合数组。变数必须为相同大小的数组。

```
nCr({5,4,3},{2,4,2}) ENTER
{10  1  3}
```

---

nCr(*矩阵1*, *矩阵2***)** ⇒ *矩阵*

对应基于两矩阵相应元素对的组合矩阵。变数必须为相同大小的矩阵。

```
nCr([6,5;4,3],[2,2;2,2]) ENTER
```

$$\begin{bmatrix} 15 & 10 \\ 6 & 3 \end{bmatrix}$$

nDeriv() MATH/Calculus 菜单

nDeriv(*表达式1*, *变元*[, *h*]**)** ⇒ *表达式*
nDeriv(*表达式1*, *变元*, *数组***)** ⇒ *数组*
nDeriv(*数组*, *变元*[, *h*]**)** ⇒ *数组*
nDeriv(*矩阵*, *变元*[, *h*]**)** ⇒ *矩阵*

是对应作为表达式的数字导数，使用中差商程式。

h 是分级值，如 *h* 被省略，它就会被默认成 0.001。

当使用*数组* 或 *矩阵* 时，运算会映射在数组的数值或矩阵元素上。

备注: 也请参阅 avgRC() 和 *d*()。

```
nDeriv(cos(x),x,h) ENTER
```

$$\frac{-(\cos(x-h)-\cos(x+h))}{2\cdot h}$$

```
limit(nDeriv(cos(x),x,h),h,0)
ENTER
```

-sin(x)

```
nDeriv(x^3,x,0.01) ENTER
```

$3.\cdot(x^2+.000033)$

```
nDeriv(cos(x),x)|x=π/2 ENTER
```

-1.

```
nDeriv(x^2,x,{.01,.1}) ENTER
{2.·x  2.·x}
```

NewData CATALOG

NewData *数据变元*, *数组1*[, *数组2*] [, *数组3*]...

创建*数据变元*，其中列是顺序中的数组。

至少要有一个数组。

如右表所示，*数组1*, *数组2*, ..., *数组n* 可为数组，就是说表达式解数组或数组变元名称。

NewData 可使新变元在数据 / 矩阵编辑器中成为当前的项目。

```
NewData mydata,{1,2,3},{4,5,6}
ENTER
```

Done

(转到数据 / 矩阵编辑器并打开*变元* mydata 来显示以下的数据变元。)

DATA	c1	c2	c3
1	1	4	
2	2	5	
3	3	6	
4			

NewData *数据变元*, *矩阵*

基于*矩阵*创建*数据变元*。

NewData sysData, *矩阵*

是将*矩阵*的内容加载入系统数据变元 sysData 之中。

NewFold CATALOG

NewFold *文件夹名*

用*文件夹名*创建用户定义文件夹，然后将当前文件夹设定成该文件夹。执行此指令后，您就会处于新文件夹内了。

```
NewFold games ENTER          Done
```

newList() CATALOG

newList(*数字元素***)** ⇒ *数组*

对应维数是*数字元素*的数组。每一个元素均为零。

```
newList(4) ENTER          {0 0 0 0}
```

newMat() CATALOG also Math/Matrix 菜单

newMat(*数字行*, *数字列***)** ⇒ *矩阵*

对应零矩阵，数是*数字行*乘*数字列*。

```
newMat(2,3) ENTER          [0 0 0
                           0 0 0]
```

NewPic CATALOG

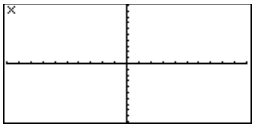
NewPic *矩阵*, *图形变元* [, *最大行*][, *最大列*]

基于*矩阵*创建*图形变元*。*矩阵*必须是 $n \times 2$ 矩阵，每一行代表一个象素，象素坐标从 0,0 开始。如果*图形变元*已经存在的话，**NewPic** 会取代之。

图形变元 的默认值是矩阵值所要求的最小区域。任选变数*最大行*和*最大列*是定义*图形变元*的最大边界限制。

```
NewPic [1,1;2,2;3,3;4,4;5,5;
  5,1;4,2;2,4;1,5],xpic ENTER Done

RclPic xpic ENTER
```



NewPlot CATALOG

NewPlot *n*, *类型*, *x 数组* [,[*y 数组*], [*频数数组*], [*类别数组*], [*包含类别数组数组*], [*标记*] [, *桶大小*]]

为图号 *n* 创建新的图形定义。

*类型*是指定作图的类型。
1 = 散射图
2 = xy 线图
3 = 箱线图
4 = 直方图
5 = 修饰的直方图

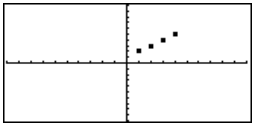
*标记*是指定标记的显示类型。
1 = ▫ (箱线)
2 = × (交叉)
3 = + (加)
4 = ▪ (平方)
5 = ・(点)

*桶大小*是每一"桶" (*类型* = 4)直方图的宽度，而且会基于窗口变元 xmin 和 xmax 变动。*桶大小*必须 >0。默认值 = 1。

备注: *n* 可为 1–9。数组必须为变元名称或 c1–c99 (数据 / 矩阵编辑器显示的上一数据变元中的列)，不一定要求为变元名称和不能是 c1–c99 的*包含类别数组*除外。

```
FnOff ENTER                  Done
PlotsOff ENTER               Done
{1,2,3,4}→L1 ENTER        {1 2 3 4}
{2,3,4,5}→L2 ENTER        {2 3 4 5}
NewPlot 1,1,L1,L2,,,,4 ENTER Done
```

按下 ◆[GRAPH] 来显示:



NewProb CATALOG

NewProb

NewProb [ENTER] Done

执行大量的运算，让您在一个清空的状态且无须重设内存的前提下开始一个新的习题。

- 是在当前文件夹中清除所有单字符变元名称(Clear a-z)，除非变元为已锁定的或已存档的。
- 在当前的作图模式下关闭所有函数和统计图(**FnOff** 和 **PlotsOff**)。
- 执行 **ClrDraw**、**ClrErr**、**ClrGraph**、**ClrHome**、**ClrIO** 和 **ClrTable**。

nInt() MATH/Calculus 菜单

nInt(*表达式1, 变元, 下, 上***)** ⇒ *表达式*

nInt(*e*^(⁻x^2),x,⁻1,1) [ENTER]

1.493...

如被积函数*表达式1*含有的变元与*变元*相同，并如*下*和*上*含有正数 ∞ 或负数 ∞，那么 **nInt()** 对应 ∫(*表达式1, 变元, 下, 上*)的近似值。此近似值是*下*<*变元*<*上*区间中被积函数的某些样本值的称重平均数。

目标是六个有效数字，适配算法与已达到的目标可能相似时就终止，或当其看起来与附加的样本将要产生的合算改良不可能相似时即终止。

nInt(cos(x),x,⁻π,π+1E⁻12) [ENTER]

⁻1.041...E⁻12

∫(cos(x),x,⁻π,π+10^(⁻12)) [ENTER]

$$-\sin\left(\frac{1}{1000000000000}\right)$$

当其看起来与未达到的目标相似时，就会显示出一个警告信息("Questionable accuracy")。

ans(1)[◆] [ENTER] ⁻1.E⁻12

套入 **nInt()** 来执行倍数数字积分法。积分法限制会依靠它们以外的积分变元。

nInt(nInt(*e*^(⁻x*y)/√(x^2−y^2), y,⁻x,x),x,0,1) [ENTER] 3.304...

备注: 也请参阅 ∫() 。

norm() MATH/Matrix/Norms 菜单

norm(*矩阵***)** ⇒ *表达式*

norm([a,b;c,d]) [ENTER]

$$\sqrt{a^2+b^2+c^2+d^2}$$

对应弗洛尼斯规则。

norm([1,2;3,4]) [ENTER] $\sqrt{30}$

not MATH/Test 菜单

not *布尔表达式1* ⇒ *布尔表达式*

not 2>=3 [ENTER] true

not x<2 [ENTER] x ≥ 2

对应真的、假的或已化简的 *布尔表达式1*。

not not innocent [ENTER] innocent

not *整数1* ⇒ *整数*

对应实数整数的一的余数。在内部，*整数1*被转换成标记的 32-位十进制数字，每位的值为一的余数会被翻转(0 变成 1 或相反)。结果会按照 Base 模式显示出来。

您可一任何数字基输入整数。对于十进制或十六进制输入项，您必须相应使用 0b 或 0h 前缀。不带前缀的话，整数会被当作小数(基 10)处理。

如您输入了对于标记的 32-位十进制形式过大的小数整数，就会采用对称模运算来将数值带入近似的范围内。

在十六进制基模式下:

```
not 0h7AC36 ENTER        0hFFF853C9
```

重要事项: 是零，不是字母 O。

在十进制基模式下:

```
0b100101▸dec ENTER                37

not 0b100101 ENTER
0b11111111111111111111111111101101(

ans(1)▸dec ENTER                 -38
```

备注: 一个十进制输入项可有多达 32 位 (0b 前缀不计算在内)；一个十六进制输入项可有 8 位。

备注: 欲输入▸换算算子，可按下 2nd [▸]。您也可从 MATH/Base 菜单选择基础换算形式。

nPr() MATH/Probability 菜单

nPr(*表达式1***,** *表达式2***)** ⇒ *表达式*

对 *表达式1* ≥ *表达式2* ≥ 0 的整数 *表达式1*和*表达式2*，**nPr()** 是 *表达式1*每次在 *表达式2*取得的元素排列数。两变数可为整数或符号表达式。

nPr(*表达式***, 0)** ⇒ 1

nPr(*表达式***,** *负整数***)** ⇒ 1/((*表达式*+1)·(*表达式*+2)...(*表达式*− *负整数*))

nPr(*表达式***,** *正整数***)** ⇒ *表达式*·(*表达式*− 1)...(*表达式*− *正整数*+1)

nPr(*表达式***,** *非整数***)** ⇒ *表达式*!/ (*表达式*− *非整数*)!

```
nPr(z,3) ENTER        z·(z−2)·(z−1)

ans(1)|z=5 ENTER                  60

nPr(z,-3) ENTER        1/((z+1)·(z+2)·(z+3))

nPr(z,c) ENTER                 z!/((z−c)!)

ans(1)*nPr(z−c,-c) ENTER           1
```

nPr(*数组1***,** *数组2***)** ⇒ *数组*

对应两数组中基于相应元素对的排列数组。变数必须为相同大小的数组。

```
nPr({5,4,3},{2,4,2}) ENTER
                        {20 24 6}
```

nPr(*矩阵1***,** *矩阵2***)** ⇒ *矩阵*

对应两矩阵中基于相应元素对的排列矩阵。变数必须为相同大小的矩阵。

```
nPr([6,5;4,3],[2,2;2,2]) ENTER
                        [30 20; 12 6]
```

nSolve() MATH/Algebra 菜单

nSolve(*方程***,** *变元或猜测值***)** ⇒ *数字或错误_字符串*

为一个变元重复搜寻*方程*的近似实数数字解。指定*变元或猜测值*为:

变元
– 或 –
变元 = *实数*

例如，x 是有效的，所以 x=3。

```
nSolve(x^2+5x−25=9,x) ENTER
                          3.844...

nSolve(x^2=4,x=-1) ENTER        -2.

nSolve(x^2=4,x=1) ENTER          2.
```

备注: 如有多个解，您可使用猜测来帮助求出一个特殊的解。

nSolve() 通常比 **solve()** 或 **zeros()** 快些，尤其是使用了算子来限制搜寻准确含有样本解的小区间时。

```
nSolve(x^2+5x-25=9,x)|x<0 ENTER
                          -8.844...
```

nSolve() 是尝试定义离差为零的一点或两相应的接近点，在那里，离差有相反的符号且离差的大小不致于过大。如利用样本点的数字无法达到目的，它则对应字符串 "no solution found"。

```
nSolve(((1+r)^24-1)/r=26,r)|r>
0 and r<.25 ENTER          .0068...
```

如您在程序中用到 **nSolve()**，可在用于代数表达式中之前使用 **getType()** 来检查数字结果。

```
nSolve(x^2=-1,x) ENTER
              "no solution found"
```

备注: 也请参阅 **cSolve()**、**cZeros()**、**solve()** 和 **zeros()**。

OneVar　MATH/Statistics 菜单

OneVar *数组1* [[, *数组2*] [, *数组3*] [, *数组4*]]

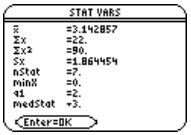
计算一统计并更新所有系统统计变元。

所有数组必须为等维数，*数组4* 除外。

数组1 代表 x 数组。
数组2 代表频数。
数组3 代表类别代码。
数组4 代表类别包含数组。

备注: *数组1* 到 *数组3* 必须为变元名称或 c1-c99 (数据 / 矩阵编辑器显示的上一数据变元的列)。*数组4* 不必为变元名称并不能为 c1-c99。

```
{0,2,3,4,3,4,6}→L1 ENTER
OneVar L1 ENTER                Done
ShowStat ENTER
```



or　MATH/Test 菜单

布尔表达式1 **or** *布尔表达式2* ⇒ *布尔表达式*

对应原始输入项真实的、虚假的或已化简的形式。

如表达式之一或两者化简成真实的，就对应真实；只有当两表达式计算成虚假的时候才对应虚假。

备注: 请参阅 **xor**。

```
x≥3 or x≥4 ENTER                x ≥ 3
```

程序段:

```
  ⋮
If x<0 or x≥5
  Goto END
  ⋮
If choice=1 or choice=2
  Disp "Wrong choice"
  ⋮
```

整数1 **or** *整数2* ⇒ *整数*

利用 **or** 运算逐位比较两个实数整数。在内部，两个整数会被转换成标记的 32-位十进制数字。当对应的位得到比较后，如位是 1，那么结果就是 1；结果只有在两个位均是 0 时才是 0。对应的值表示位结果，并按照 Base 模式显示出来。

您可一任何数字基输入整数。对于十进制或十六进制输入项，您必须相应使用 0b 或 0h 前缀。不带前缀的话，整数会被当作小数(基 10)处理。

如您输入了对于标记的 32-位十进制形式过大的小数整数，就会采用对称模运算来将数值带入近似的范围内。

备注: 请参阅 **xor**。

在十六进制基模式下:

```
0h7AC36 or 0h3D5F ENTER   0h7BD7F
```

└ **重要事项:** 是零，不是字母 O。

在十进制基模式下:

```
0b100101 or 0b100 ENTER 0b100101
```

备注: 一个十进制输入项可有多达 32 位 (0b 前缀不计算在内)；一个十六进制输入项可有 8 位。

ord() MATH/String 菜单

ord(*字符串***)** ⇒ *整数*
ord(*数组1***)** ⇒ *数组*

对应字符串*字符串*中第一个字符的数字码，或每一元素数组的第一个字符的数组。

```
ord("hello") ENTER                 104
char(104) ENTER                    "h"
ord(char(24)) ENTER                 24
ord({"alpha","beta"}) ENTER
                            {97  98}
```

Output CATALOG

Output *行, 列, 表达式或字符串*

在程序 I/O 屏的文本坐标(*行, 列*)显示*表达式或字符串* (表达式或字符串)。

一条表达式可包含换算运算，如 **▸DD** 和 **▸Rect**。您也可使用 ▸ 算子来执行单位和数字基的换算。

如 Pretty Print = ON，*表达式或字符串* 是 "工整印刷体的"显示。

在程序 I/O 屏上，您可按下 F5 来显示主屏幕，或在程序中用 **DispHome** 来显示。

程序段:

```
   ⋮
:RandSeed 1147
:ClrIO
:For i,1,90,10
:  Output i, rand(100),"Hello"
:EndFor
   ⋮
```

执行后的结果:

```
        Hello
   Hello
           Hello
      Hello
               Hello
    Hello
                 Hello
```

P▸Rx() MATH/Angle 菜单

P▸Rx(*r表达式***, θ***表达式***)** ⇒ *表达式*
P▸Rx(*r数组***, θ***数组***)** ⇒ *数组*
P▸Rx(*r矩阵***, θ***矩阵***)** ⇒ *矩阵*

对应(r, θ)对的等同 x-坐标。

备注: 按照当前的角模式，θ 变数会被当作度或弧角演算。如变数是表达式，您可使用 ° 或 ʳ 临时覆盖角模式设置。

在 Radian 角模式下:

```
P▸Rx(r,θ) ENTER                cos(θ)·r
P▸Rx(4,60°) ENTER                     2
P▸Rx({-3,10,1.3},{π/3,-π/4,0})
ENTER
                 {-3/2   5·√2   1.3}
```

P▸Ry() MATH/Angle 菜单

P▸Ry(*r表达式***, θ***表达式***)** ⇒ *表达式*
P▸Ry(*r数组***, θ***数组***)** ⇒ *数组*
P▸Ry(*r矩阵***, θ***矩阵***)** ⇒ *矩阵*

对应(r, θ)对的等同 y-坐标。

备注: 按照当前的角模式，θ 变数会被当作度或弧角演算。如变数是表达式，您可使用 ° 或 ʳ 临时覆盖角模式设置。

在 Radian 角模式下:

```
P▸Ry(r,θ) ENTER                sin(θ)·r
P▸Ry(4,60°) ENTER                2·√3
P▸Ry({-3,10,1.3},{π/3,-π/4,0})
ENTER
                 {-3·√3/2   -5·√2   0.}
```

part() CATALOG

part(*表达式1***[,***非负整数***])**

此高级编程功能可让您在*表达式1*化简的结果中识别和提取所有子表达式。

例如，如*表达式1* 化简成 cos(π∗ x+3):

- **cos()** 函数有一个变数: (π∗ x+3)。
- (π∗ x+3)的和有两个运算域: π∗ x 和 3。
- 数字 3 没有变数或运算域。
- π∗ x 的积有两个运算域: π 和 x。
- 变元 x 和符号常数 π 没有变数或运算域。

如 x 有一个数值，而您又按下了 ⊡ ENTER，π∗ x 的数值就得到计算，结果会加 3，随后，余弦也得到计算。**cos()** 是**顶级**算子，这是因为它已应用了**上一个**。

---

part(*表达式1***)** ⇒ *数字*

化简*表达式1* 和对应顶级变数或运算域的数字。如*表达式1*是数字、符号常数如 π、*e*、***i*** 或 ∞，这是对应 0。

```
part(cos(π∗ x+3)) ENTER          1
```

备注: cos(π∗ x+3)没有变数。

part(*表达式1***, 0)** ⇒ *字符串*

化简*表达式1*和对应含有顶级函数名称或运算域的字符串。如*表达式1*是数字、符号常数如 π、*e*、***i*** 或 ∞，这是对应 **string(***表达式1***)**。

```
part(cos(π∗ x+3),0) ENTER    "cos"
```

part(*表达式1***,** *n***)** ⇒ *表达式*

化简*表达式1*并对应 n^{th} 变数或运算域，其中 *n* 是 > 0 和 ≤ 由 **part(***表达式1***)** 对应的顶级变数或运算域数。否则就会对应错误。

```
part(cos(π∗ x+3),1) ENTER    3+π·x
```

备注: 化简回改变数的顺序。

通过 **part()** 的组合变分，您可从*表达式1*化简的结果中提取所有的子表达式。如右边的例子所示，您可储存变数或运算域，然后利用 **part()** 来进一步提取子表达式。

备注: 当使用 **part()** 时，切勿依赖和与积中的任何特殊顺序。

```
part(cos(π∗ x+3)) ENTER          1
part(cos(π∗ x+3),0) ENTER    "cos"
part(cos(π∗ x+3),1)→temp ENTER
                             3+π·x
temp ENTER                   π·x+3
part(temp,0) ENTER             "+"
part(temp) ENTER                 2
part(temp,2) ENTER               3
part(temp,1)→temp ENTER        π·x
part(temp,0) ENTER             "∗"
part(temp) ENTER                 2
part(temp,1) ENTER               π
part(temp,2) ENTER               x
```

(x+y+z) 和 (x− y− z)表达式在内部被表示成 (x+y)+z 和 (x− y)− z，这样就会影响对应第一和第二变数数值的数值。这些是技术上的原因，因为 **part(x+y+z,1)** 是对应 y+x 而不是 x+y。

```
part(x+y+z) ENTER                2
part(x+y+z,2) ENTER              z
part(x+y+z,1) ENTER            y+x
```

同样道理，x∗ y∗ z 在内部也被表示成 (x∗ y)∗ z。这些也是技术上的原因，因为第一变数是对应 y·x 而不是 x·y。

```
part(x∗ y∗ z) ENTER              2
part(x∗ y∗ z,2) ENTER            z
part(x∗ y∗ z,1) ENTER          y·x
```

当您从矩阵中提取子表达式时，请记住，矩阵是当作数组的数组被储存起来的，如右边的例子所示。

```
part([a,b,c;x,y,z],0) ENTER        "{"
part([a,b,c;x,y,z]) ENTER            2
part([a,b,c;x,y,z],2)→temp
ENTER
                          {x   y   z}
part(temp,0) ENTER                 "{"
part(temp) ENTER                     3
part(temp,3) ENTER                   z
delVar temp ENTER                 Done
```

右边的例子程序编辑器函数是使用 **getType()** 和 **part()** 来将符号微分法部分完成的。分析和完成此函数有助于教导您如何手动微分的。您甚至可以包含 TI-89 / TI-92 Plus 不能微分的函数，如 Bessel 函数。

```
:d(y,x)
:Func
:Local f
:If getType(y)="VAR"
:  Return when(y=x,1,0,0)
:If part(y)=0
:  Return 0 ◉ y=π,∞,i,numbers
:part(y,0)→f
:If f="-" ◉ if negate
:  Return ⁻d(part(y,1),x)
:If f="–" ◉ if minus
:  Return d(part(y,1),x)
     -d(part(y,2),x)
:If f="+"
:  Return d(part(y,1),x)
     +d(part(y,2),x)
:If f="*"
:  Return
part(y,1)*d(part(y,2),x)
     +part(y,2)*d(part(y,1),x)
:If f="{"
:  Return seq(d(part(y,k),x),
     k,1,part(y))
:Return undef
:EndFunc
```

PassErr CATALOG

PassErr

将错误过渡到下一层次。

如 "errornum" 是零，**PassErr** 将会不起任何作用。

程序中的 **Else** 分句将使用 **ClrErr** 或 **PassErr**。如错误要被执行或被忽略，请使用 **ClrErr**。如对于错误还不知如何处理，请使用 **PassErr** 来将错误过渡到下一错误处理程序。(也请参阅 **ClrErr**。)

请参阅 **ClrErr** 程序表例子。

Pause CATALOG

Pause [*表达式*]

将程序的执行挂起。如您包含*表达式*，*表达式*就会在程序 I/O 屏显示出来。

*表达式*可包含换算运算，如 **▸DD** 和 **▸Rect**。您也可使用 ▸ 算子来执行单位和数字基换算。

如*表达式*的结果对于单屏显示过大，您可用光标控制盘来滚屏。

当您按下 ENTER 时，程序执行就会恢复。

程序段:

```
  ⋮
:ClrIO
:DelVar temp
:1→temp[1]
:1→temp[2]
:Disp temp[2]
:◉ Guess the Pattern
:For i,3,20
:  temp[i-2]+temp[i-1]→temp[i]
:  Disp temp[i]
:  Disp temp,"Can you guess
the
     next","number?"
:  Pause
:EndFor
  ⋮
```

PlotsOff CATALOG

PlotsOff [**1**] [, **2**] [, **3**] **...** [, **9**]

为作图而关闭指定的图象。当处于双图模式时，仅影响活动的图象。

如没有参数，会关闭所有图象。

```
PlotsOff 1,2,5 ENTER          Done
PlotsOff ENTER                Done
```

PlotsOn CATALOG

PlotsOn [**1**] [, **2**] [, **3**] **...** [, **9**]

为作图而打开指定的图象。当处于双图模式时，仅影响活动的图象。

如您不包含任何变数，就会打开所有图象。

```
PlotsOn 2,4,5 ENTER           Done
PlotsOn ENTER                 Done
```

▸Polar MATH/Matrix/Vector ops 菜单

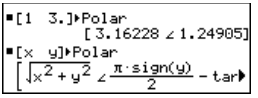
向量▸**Polar**

以极坐标形式 [r ∠θ] 显示*向量*。向量必须是维数 2 且可为行或列。

备注: **▸Polar** 是一个显示格式指令，不是一个换算函数。您只可在输入行的结尾使用它，它不会更新 ans。

备注: 也请参阅 **▸Rect**。

```
[1,3.]▸Polar ENTER
[x,y]▸Polar ENTER
```



复数值▸**Polar**

以极坐标形式显示 *复数向量*。

- Degree 角模式对应(r∠θ)。
- Radian 角模式对应 $re^{i\theta}$。

复数值 可有任何复数形式，尽管如此，$re^{i\theta}$ 输入项会在 Degree 角模式中造成错误。

备注: 对于一个(r∠θ)极坐标输入项，您必须使用括号。

在 Radian 角模式下:

3+4*i* ▸Polar ENTER $e^{i\cdot\left(\frac{\pi}{2}-\tan^{-1}(3/4)\right)}$

(4∠π/3)▸Polar ENTER $e^{\frac{i\cdot\pi}{3}}\cdot 4$

在 Degree 角模式下:

3+4*i*▸Polar ENTER (5∠90−tan⁻¹(3/4))

polyEval() MATH/List 菜单

polyEval(*数组1*, *表达式1***)** ⇒ *表达式*
polyEval(*数组1*, *数组2***)** ⇒ *表达式*

将第一变数当作多项式下降度系数来演算，对应第二变数值计算的多项式。

polyEval({a,b,c},x) ENTER

$a\cdot x^2+b\cdot x+c$

polyEval({1,2,3,4},2) ENTER 26

polyEval({1,2,3,4},{2,⁻7}) ENTER {26 ⁻262}

PopUp CATALOG

PopUp *项目数组*, *变元*

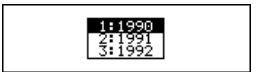
显示含有自*项目数组* 的字符串的弹出菜单，等待您选择一个项并将您所选择的号码存入*变元*中。

项目数组 的元素必须为字符串: {*项目1 字符串*, *项目2 字符串*, *项目3 字符串*, ...}

如*变元* 已存在并带有有效的项目号码，那么它们就会被当成默认选择显示出来。

*项目数组*必须含有最少一个选择。

PopUp {"1990","1991","1992"},var1 ENTER



PowerReg MATH/Statistics/Regressions 菜单

PowerReg *数组1*, *数组2*[, [*数组3*] [, *数组4*, *数组5*]]

计算幂回归并更新所有系统统计变元。

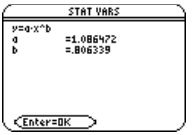
所有数组必须带有等同的维数，*数组5*除外。

数组1 代表 x 数组。
数组2 代表 y 数组。
数组3 代表频数。
数组4 代表类别代码。
数组5 代表类别包含数组。

备注: *数组1*到*数组4*必须为变元名称或 c1-c99 (数据 / 矩阵编辑器显示的上一数据变元的列)。*数组5*不必为变元名称并不能为 c1-c99。

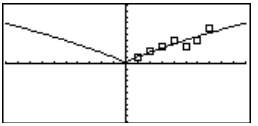
在函数作图模式下:

```
{1,2,3,4,5,6,7}→L1 [ENTER]
                          {1 2 3 ...}
{1,2,3,4,3,4,6}→L2 [ENTER]
                          {1 2 3 ...}
PowerReg L1,L2 [ENTER]          Done
ShowStat [ENTER]
```




```
[ENTER]
Regeq(x)→y1(x) [ENTER]          Done
NewPlot 1,1,L1,L2 [ENTER]       Done
```

[◆] [GRAPH]



Prgm CATALOG

Prgm
⋮
EndPrgm

识别程序开头所需的指令。程序的最后一行必须为 **EndPrgm**。

程序段:

```
:prgmname()
:Prgm
:
:EndPrgm
```

product() MATH/List 菜单

product(*数组***[, ***开始***[, ***结束***]]) ⇒** *表达式*

对应*数组*中含有的元素的积。*开始*和*结束*是任选项，它们是用于指定元素类别的。

```
product({1,2,3,4}) ENTER          24
product({2,x,y}) ENTER          2·x·y
product({4,5,8,9},2,3) ENTER    40
```

---

product(*矩阵1***[, ***开始***[, ***结束***]]) ⇒** *矩阵*

是对应含有*矩阵1*的列中元素的积的行向量。*开始*和*结束*均是任选项，它们是用于指定行范围的。

```
product([1,2,3;4,5,6;7,8,9])
ENTER                  [28 80 162]
product([1,2,3;4,5,6;7,8,9],
1,2) ENTER               [4,10,18]
```

Prompt CATALOG

Prompt *变元1*[, *变元2*] [, *变元3*] **...**

在程序 I/O 屏上利用提示 var1? 为变数数组中的每一变元显示提示。在相应的变元中储存输入的表达式。

Prompt 起码要有一个变数。

程序段:

```
  ⋮
Prompt A,B,C
  ⋮
EndPrgm
```

propFrac() MATH/Algebra 菜单

propFrac(*表达式1***[, ***变元***]) ⇒** *表达式*

propFrac(*有理_数***)** 对应带有相同标记的整数和分数，以及分母数量较分子数量为大的*有理_数*和。

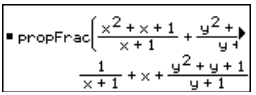
```
propFrac(4/3) ENTER          1 + 1/3
propFrac(⁻4/3) ENTER         ⁻1−1/3
```

propFrac(*有理_表达式,变元***)** 对应正常比与相对于*变元*的多项式的和。在每一正常比中，分母中*变元*度超过分子中的*变元*度。相似的*变元*幂会被收集，项和因子会被当作主变元与*变元*一起储存。

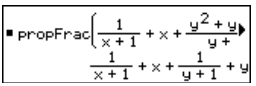
如省略了*变元*，正常分数会相对于大多数主变元扩展。多项式部分的系数随后会按照它们的大多数主变元首先得到转换等等。

对于有理表达式，**propFrac()** 是快速但对于 **expand()** 的极端替代较少。

```
propFrac((x^2+x+1)/(x+1)+
  (y^2+y+1)/(y+1),x) ENTER
```




```
propFrac(ans(1))
```



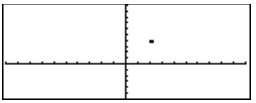
PtChg CATALOG

PtChg x, y
PtChg x*数组*, y*数组*

显示作图屏并将最接近窗口坐标(x, y)的屏幕象素颠倒。

备注: PtChg 到 **PtText** 是显示继续类似的例子。

```
PtChg 2,4 ENTER
```

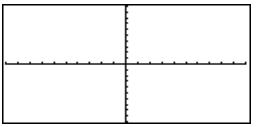


PtOff CATALOG

PtOff x, y
PtOff x*数组*, y*数组*

显示作图屏并将最接近窗口坐标(x, y)的屏幕象素关闭。

PtOff 2,4 ENTER

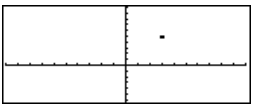


PtOn CATALOG

PtOn x, y
PtOn x*数组*, y*数组*

显示作图屏并将最接近窗口坐标(x, y)的屏幕象素打开。

PtOn 3,5 ENTER



ptTest() CATALOG

ptTest (x, y) ⇒ *布尔常数表达式*
ptTest (x*数组*, y*数组*) ⇒ *布尔常数表达式*

对应 true 或 false。只有在最接近窗口坐标(x, y)的屏幕象素打开时才对应 true。

ptTest(3,5) ENTER true

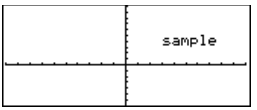
PtText CATALOG

PtText *字符串*, x, y

显示作图屏并将字符串 *字符串* 放在指定的最靠近窗口坐标(x, y)的象素处。

字符串 是定位在其坐标的第一个字符的左上角处。

PtText "sample",3,5 ENTER



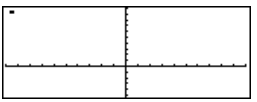
PxlChg CATALOG

PxlChg *行*, *列*
PxlChg *行数组*, *列数组*

显示作图屏并将坐标(*行*, *列*)处的象素颠倒。

备注: 重新作图时会清除所有画出的项。

PxlChg 2,4 ENTER



PxlCrcl CATALOG

PxlCrcl *行*, *列*, r [, *画图模式*]

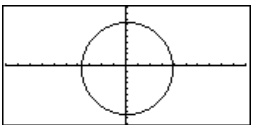
显示作图屏并画出圆心在象素坐标(*行*, *列*)，半径为 r 象素的圆。

如*画图模式* = 1，画圆(默认)。
如*画图模式* = 0，关闭圆。
如*画图模式* = -1，延着圆翻转象素。

备注: 重新作图时会清除所有画出的项。也请参阅 **Circle**。

TI-89: PxlCrcl 40,80,30,1 ENTER

TI-92 Plus: PxlCrcl 50,125,40,1 ENTER



PxlHorz CATALOG

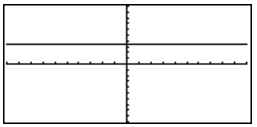
PxlHorz *row* [, *画图模式*]

显示作图屏并在象素位置*行*处画线。

如*画图模式* = 1，画线(默认)。
如*画图模式* = 0，关闭线。
如*画图模式* = -1，将打开的线关闭或将关闭的线打开(延线段翻转象素)。

备注: 重新作图时会清除所有画出的项。也请参阅 **LineHorz**。

PxlHorz 25,1 ENTER



PxlLine CATALOG

PxlLine *行开始*, *列开始*, *行结束*, *列结束* [, *画图模式*]

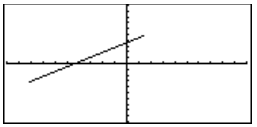
显示作图屏并在象素坐标(*行开始*, *列开始*)和(*行结束*, *列结束*)处画出线段，包括结束点在内。

如*画图模式* = 1，画出线段(默认)。
如*画图模式* = 0，关闭线段。
如*画图模式* = -1，打开的线关闭或将关闭的线打开(延线段翻转象素)。

备注: 重新作图时会清除所有画出的项。也请参阅 **Line**。

TI-89: PxlLine 50,15,20,90,1 ENTER

TI-92 Plus: PxlLine 80,20,30,150,1 ENTER



PxlOff CATALOG

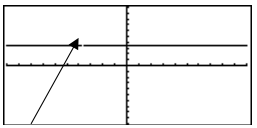
PxlOff *行*, *列*
PxlOff *行数组*, *列数组*

显示作图屏，并将象素坐标(*行*, *列*)处的象素关闭。

备注: 重新作图时会清除所有画出的项。

PxlHorz 25,1 ENTER

PxlOff 25,50 ENTER



25,50

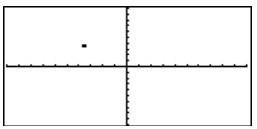
PxlOn CATALOG

PxlOn *行*, *列*
PxlOn *行数组*, *列数组*

显示作图屏并将象素坐标(*行*, *列*)处的象素打开

备注: 重新作图时会清除所有画出的项。

PxlOn 25,50 ENTER



pxlTest() CATALOG

pxlTest (*行*, *列***)** ⇒ *布尔表达式*
pxlTest (*行数组*, *列数组***)** ⇒ *布尔表达式*

如象素坐标(*行*, *列*)处的象素是打开的话就对应 true；象素关闭的话就对应 false。

备注: 重新作图时会清除所有画出的项。

PxlOn 25,50 ENTER

TI-89: HOME

TI-92 Plus: ◆ [HOME]

PxlTest(25,50) ENTER true

PxlOff 25,50 ENTER

TI-89: HOME

TI-92 Plus: ◆ [HOME]

PxlTest(25,50) ENTER false

PxlText CATALOG

PxlText *字符串*, *行*, *列*

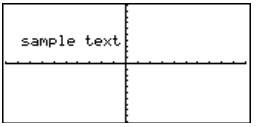
显示作图屏并将*字符串*放在屏幕上，从象素坐标(*行*, *列*)开始。

*字符串*是定位在其坐标的第一个字符的左上角位置。

备注: 重新作图时会清除所有画出的项。

TI-89: `PxlText "sample text",20,10` ENTER

TI-92 Plus: `PxlText "sample text",20,50` ENTER



PxlVert CATALOG

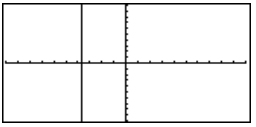
PxlVert *列* [, *画图模式*]

在象素位置*列*处画垂线。

如*画图模式* = 1，画线(默认)。
如*画图模式* = 0，关闭线。
如*画图模式* = -1，将打开的线关闭或将关闭的线打开(延线段翻转象素)。

备注: 重新作图时会清除所有画出的项。也请参阅 **LineVert**。

`PxlVert 50,1` ENTER



QR MATH/Matrix 菜单

QR *矩阵*, *q矩阵名称*, *r矩阵名称*[, *误差*]

计算实数或复数矩阵的 Householder QR 因式分解。结果 Q 和 R 矩阵会存入指定的*矩阵名称*。Q 矩阵单式的，R 矩阵是上三角形的。

作为任选项，任何矩阵元素如其绝对值小于*误差*，可会被当作零来处理。此误差只有在矩阵带有浮点输入项且不含有任何未赋值的符号变元时才用到。否则，*误差*会被忽略。

- 如您使用 ◆ ENTER 或将模式设定成 Exact/Approx=APPROXIMATE，那么运算就会使用浮点算术进行。
- 如省略或没有使用*误差*，默认误差是按以下方式计算出来的:

  5E⁻14 * **max(dim(***矩阵***))** * **rowNorm(***矩阵***)**

在 m1 中的浮点数字(9.)造成以浮点模式计算结果。

`[1,2,3;4,5,6;7,8,9.]→m1` ENTER

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9. \end{bmatrix}$$

`QR m1,qm,rm` ENTER Done

qm ENTER

$$\begin{bmatrix} .123\ldots & .904\ldots & .408\ldots \\ .492\ldots & .301\ldots & -.816\ldots \\ .861\ldots & -.301\ldots & .408\ldots \end{bmatrix}$$

rm ENTER

$$\begin{bmatrix} .123\ldots & .904\ldots & .408\ldots \\ .492\ldots & .301\ldots & -.816\ldots \\ .861\ldots & -.301\ldots & .408\ldots \end{bmatrix}$$

`[m,n;o,p]→m1` ENTER

$$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$$

QR 因式分解是利用 Householder 转移来进行数字计算的。符号解是使用 Gram-Schmidt 计算的。*q矩阵名称* 中的列是估量 *矩阵*指定的空间的标准正交基向量。

QR m1,qm,rm [ENTER] Done

qm [ENTER]

$$\begin{bmatrix} \frac{m}{\sqrt{m^2+o^2}} & \frac{-\text{sign}(m\cdot p-n\cdot o)\cdot o}{\sqrt{m^2+o^2}} \\ \frac{o}{\sqrt{m^2+o^2}} & \frac{m\cdot\text{sign}(m\cdot p-n\cdot o)}{\sqrt{m^2+o^2}} \end{bmatrix}$$

rm [ENTER]

$$\begin{bmatrix} \sqrt{m^2+o^2} & \frac{m\cdot n+o\cdot p}{\sqrt{m^2+o^2}} \\ 0 & \frac{|m\cdot p-n\cdot o|}{\sqrt{m^2+o^2}} \end{bmatrix}$$

QuadReg MATH/Statistics/Regressions 菜单

QuadReg *数组1*, *数组2*[, [*数组3*] [, *数组4*, *数组5*]]

计算二次多项式回归并更新系统统计变元。

所有数组必须有等同维数，*数组5*除外。

数组1 代表 x 数组。
数组2 代表 y 数组。
数组3 代表频数。
数组4 代表类别代码。
数组5 代表类别包含数组。

备注: *数组1*到*数组4*必须为变元名称或 c1-c99 (数据 / 矩阵编辑器显示的上一数据变元的列)。*数组5*不必为变元名称并不能为 c1-c99。

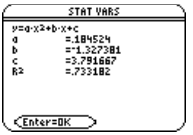
在函数作图模式下:

{0,1,2,3,4,5,6,7}→L1 [ENTER]
{1 2 3 ...}

{4,3,1,1,2,2,3,3}→L2 [ENTER]
{4 3 1 ...}

QuadReg L1,L2 [ENTER] Done

ShowStat [ENTER]

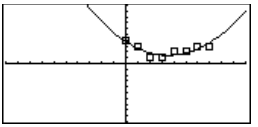


[ENTER]

Regeq(x)→y1(x) [ENTER] Done

NewPlot 1,1,L1,L2 [ENTER] Done

◆ [GRAPH]



QuartReg MATH/Statistics/Regressions 菜单

QuartReg *数组1*, *数组2*[, [*数组3*] [, *数组4*, *数组5*]]

计算四次多项式回归并更新系统统计变元。

所有数组必须有等同维数，*数组5*除外。

数组1 代表 x 数组。
数组2 代表 y 数组。
数组3 代表频数。
数组4 代表类别代码。
数组5 代表类别包含数组。

备注: *数组1*到*数组4*必须为变元名称或 c1-c99 (数据 / 矩阵编辑器显示的上一数据变元的列)。*数组5*不必为变元名称并不能为 c1-c99。

在函数作图模式下:

{-2,-1,0,1,2,3,4,5,6}→L1 [ENTER]

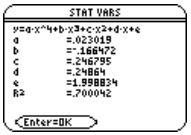
{-2 -1 0 ...}

{4,3,1,2,4,2,1,4,6}→L2 [ENTER]

{4 3 1 ...}

QuartReg L1,L2 [ENTER] Done

ShowStat [ENTER]

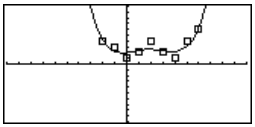


[ENTER]

Regeq(x)→y1(x) [ENTER] Done

NewPlot 1,1,L1,L2 [ENTER] Done

[◆] [GRAPH]



R▸Pθ() MATH/Angle 菜单

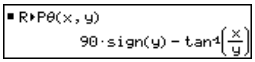
R▸Pθ (*x表达式*, *y表达式*) ⇒ *表达式*
R▸Pθ (*x数组*, *y数组*) ⇒ *数组*
R▸Pθ (*x矩阵*, *y矩阵*) ⇒ *矩阵*

对应(*x*,*y*)对变数的等同 θ-坐标。

备注: 按照当前的角模式，结果会对应角或弧度角。

在 Degree 角模式下:

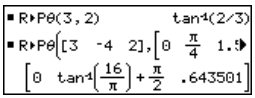
R▸Pθ(x,y) [ENTER]



在 Radian 角模式下:

R▸Pθ(3,2) [ENTER]

R▸Pθ([3,-4,2],[0,π/4,1.5]) [ENTER]



R▸Pr() MATH/Angle 菜单

R▸Pr (*x表达式*, *y表达式*) ⇒ *表达式*
R▸Pr (*x数组*, *y数组*) ⇒ *数组*
R▸Pr (*x矩阵*, *y矩阵*) ⇒ *矩阵*

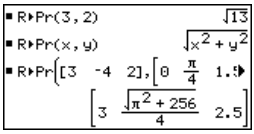
对应(x,y)对变数的等同 r-坐标。

在 Radian 角模式下:

R▸Pr(3,2) ENTER

R▸Pr(x,y) ENTER

R▸Pr([3,-4,2],[0,π/4,1.5]) ENTER



rand() MATH/Probability 菜单

rand([*n*]) ⇒ *表达式*

n 是 ≠ 零的整数。

没有参数时，是对应序列中 0 和 1 之间的下一随机数。当变数是正数时，是对应区间 [1, *n*] 中的随机整数。
当变数是负数时，是对应区间 [-*n*, -1] 中的随机整数。

RandSeed 1147 ENTER Done

(设定随机数种子。)

rand() ENTER .158...

rand(6) ENTER 5

rand(-100) ENTER -49

randMat() MATH/Probability 菜单

randMat(*数字行*, *数字列*) ⇒ *矩阵*

对应指定维数的 -9 到 9 之间的整数矩阵。

两变数必须化简成整数。

RandSeed 1147 ENTER Done

randMat(3,3) ENTER

$$\begin{bmatrix} 8 & -3 & 6 \\ -2 & 3 & -6 \\ 0 & 4 & -6 \end{bmatrix}$$

备注: 每次当您按下 ENTER 后，此矩阵中的数值也会变化。

randNorm() MATH/Probability 菜单

randNorm(*平均值*, *sd*) ⇒ *表达式*

对应指定正态分布的小数。它可为任何实数，但要在区间 [*平均值*-3∗ *sd*, *平均值*+3∗ *sd*] 更集中。

RandSeed 1147 ENTER Done

randNorm(0,1) ENTER .492...

randNorm(3,4.5) ENTER -3.543...

randPoly() MATH/Probability 菜单

randPoly(*变元*, *顺序*) ⇒ *表达式*

以*变元*对应指定顺序的多项式。系数是 -9 到 9 之间的随机整数。带头的系数不得为零。

*顺序*必须是 0–99。

RandSeed 1147 ENTER Done

randPoly(x,5) ENTER

$-2\cdot x^5+3\cdot x^4-6\cdot x^3+4\cdot x-6$

RandSeed MATH/Probability 菜单

RandSeed *数字*

如*数字* = 0，设定种子成随机数生成器的工厂默认值。如*数字* ≠ 0，是用于生成两个种子，储存在系统变元 seed1 和 seed2 中。

RandSeed 1147 ENTER Done

rand() ENTER .158...

RclGDB CATALOG

RclGDB *GDB变元*

将储存在图形数据库变元 GDBvar 中的所有设置值重新储存。

设置值的列表请参阅 **StoGDB**。

备注: 重新储存前，变元 GDBvar 内保存必须有一些东西。

```
RclGDB GDBvar ENTER          Done
```

RclPic CATALOG

RclPic *图形变元* [, *行*, *列*]

显示作图屏，并利用 OR 逻辑将储存在*图形变元*中的图象添加到象素坐标(*行*, *列*)的左上角位置。

*图形变元*必须为图象数据类型。

默认坐标为(0, 0)。

real() MATH/Complex 菜单

real(*表达式1***)** ⇒ *表达式*

对应变数的实数部分。

备注: 所有未定义的变元会被当作实数变元处理。也请参阅 **imag()**。

```
real(2+3i) ENTER             2
real(z) ENTER                z
real(x+iy) ENTER             x
```

real(*数组1***)** ⇒ *数组*

对应所有元素的实数部分。

```
real({a+i*b,3,i}) ENTER  {a  3  0}
```

real(*矩阵1***)** ⇒ *矩阵*

对应所有元素的实数部分。

```
real([a+i*b,3;c,i]) ENTER  [a  3
                            c  0]
```

▸Rect MATH/Matrix/Vector ops 菜单

向量▸**Rect**

以直角形式 [x, y, z] 显示*向量*。向量必须是维数 2 或 3，可为行或列。

备注: ▸**Rect** 是显示格式指令，不是一个换算函数。您只可在输入行的结尾使用它，它不会更新 ans。

备注: 也请参阅 ▸**Polar**。

```
[3,∠π/4,∠π/6]▸Rect ENTER
```

$$\left[\frac{3\cdot\sqrt{2}}{4}\quad\frac{3\cdot\sqrt{2}}{4}\quad\frac{3\cdot\sqrt{3}}{2}\right]$$

```
[a,∠b,∠c] ENTER [a·cos(b)·sin(c)
     a·sin(b)·sin(c)  a·cos(c)]
```

复数值▸**Rect**

以直角形式 a+b*i* 显示*复数值*。*复数值*可为任何复数形式。尽管如此，在 Degree 角模式下，一个 $re^{i\theta}$ 输入项会造成错误。

备注: 对应(r∠θ)极坐标输入项，您必须使用括号。

在 Radian 角模式下:

```
4e^(π/3)▸Rect ENTER
```

$4\cdot e^{\frac{\pi}{3}}$

```
(4∠π/3)▸Rect ENTER
```

$2+2\cdot\sqrt{3}\cdot i$

在 Degree 角模式下:

```
(4∠60)▸Rect ENTER
```

$2+2\cdot\sqrt{3}\cdot i$

备注: 欲从键盘上键入 ▸**Rect**，可按下 2nd [▸] 以获得 ▸ 算子；欲键入 ∠，可按下 2nd [∠]。

ref() MATH/Matrix 菜单

ref(*矩阵1*[, *误差*]**)** ⇒ *矩阵*

对应*矩阵1*的行梯形式。

作为任选项，任何矩阵元素如其绝对值小于*误差*，可会被当作零来处理。此误差只有在矩阵带有浮点输入项且不含有任何未赋值的符号变元时才用到。否则，*误差*会被忽略。

- 如您使用 ◆ ENTER 或将模式设定成 Exact/Approx=APPROXIMATE，那么运算就会使用浮点算术进行。
- 如省略或没有使用*误差*，默认误差是按以下方式计算出来的:

  5E⁻14 ∗ **max(dim(***矩阵1***))** ∗ **rowNorm(***矩阵1***)**

备注: 也请参阅 rref()。

```
ref([-2,-2,0,-6;1,-1,9,-9;-5,
2,4,-4]) ENTER
```

$$\begin{bmatrix} 1 & -2/5 & -4/5 & 4/5 \\ 0 & 1 & 4/7 & 11/7 \\ 0 & 0 & 1 & -62/71 \end{bmatrix}$$

```
[a,b,c;e,f,g]→m1 ENTER
```

$$\begin{bmatrix} a & b & c \\ e & f & g \end{bmatrix}$$

```
ref(m1) ENTER
```

$$\begin{bmatrix} 1 & \frac{f}{e} & \frac{g}{e} \\ 0 & 1 & \frac{a\cdot g-c\cdot e}{a\cdot f-b\cdot e} \end{bmatrix}$$

remain() MATH/Number 菜单

remain(*表达式1*, *表达式2***)** ⇒ *表达式*
remain(*数组1*, *数组2***)** ⇒ *数组*
remain(*矩阵1*, *矩阵2***)** ⇒ *矩阵*

第一变数相对于第二变数的，作为特性定义的余数:

remain(x,0) ≡ x
remain(x,y) ≡ x− y∗iPart(x/y)

作为后果，请注意 **remain(**− x,y**)** ≡ − **remain(**x,y**)**。此结果为零或带有类似第一变数的标记。

备注: 也请参阅 **mod()**。

```
remain(7,0) ENTER                    7
remain(7,3) ENTER                    1
remain(-7,3) ENTER                  -1
remain(7,-3) ENTER                   1
remain(-7,-3) ENTER                 -1
remain({12,-14,16},{9,7,-5})
ENTER
                          {3  0  1}
remain([9,-7;6,4],[4,3;4,-3])
ENTER
```

$$\begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}$$

Rename CATALOG

Rename *旧变元名称*, *新变元名称*

将*旧变元名称*重新命名成*新变元名称*。

```
{1,2,3,4}→L1 ENTER        {1,2,3,4}
Rename L1, list1 ENTER          Done
list1 ENTER               {1,2,3,4}
```

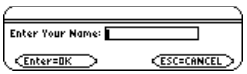
Request CATALOG

Request *提示字符串*, *变元*

如在 **Dialog**...**EndDlog** 结构里带有 **Request**，就会创建一个对话框，让用户键入数据。如是一个单独的指令，就会为此输入创建一个对话框。在两种情形下，如*变元*含有字符串，它会在输入框中作为默认选择而显示和突出显示。*提示字符串*必须为 ≤ 20 字符。

此指令可单独成立，或作为对话框结构的一部分。

```
Request "Enter Your Name",str1
ENTER
```



Return CATALOG

Return [*表达式***]**

返回作为函数结果的*表达式*。用在 **Func**...**EndFunc** 块或 **Prgm**...**EndPrgm** 块内。

备注: 可使用无变数的 **Return** 来退出程序。

备注: 请输入作为主屏幕长行的文本(不带分行符)。

```
Define factoral(nn)=Func
 :local answer,count:1→answer
 :For count,1,nn
 :answer*count→answer:EndFor
 :Return answer:EndFunc ENTER Done
factoral(3) ENTER                    6
```

right() MATH/List 菜单

right(*数组1***[, *数字*])** ⇒ *数组*

对应*数组1*中含有的最右方*数字*元素。

如您省略了*数字*，就对应*数组1*的全部。

```
right({1,3,-2,4},3) ENTER
                          {3 -2 4}
```

right(*源字符串***[, *数字*])** ⇒ *字符串*

对应*源字符串*中含有的最右方的*数字*字符。

如您省略了*数字*，就对应*源字符串*的全部。

```
right("Hello",2) ENTER        "lo"
```

right(*比较***)** ⇒ *表达式*

对应等式或不等式的右边。

```
right(x<3) ENTER                 3
```

rotate() MATH/Base 菜单

rotate(*整数1***[,*旋转#*])** ⇒ *整数*

旋转十进制整数的位数。您可以任何数字基输入*整数1*；它会自动转换成一个标记的 32 位十进制形式。如*整数1* 的数量对于此形式过大，就会采用对称模运算来将数值带入近似的范围内。

如*旋转#*是正数，就会向左转。如*旋转#*是负数，就会向右转。默认值是 -1 (即向右转一个位)。

例如，在一个右转中:

每位向右转。

0b00000000000000111101011000011010 1

最右的位转到最左。

产生:

0b10000000000000111101011000011010

结果会按照 Base 模式显示出来。

在十进制基模式下:

```
rotate(0b11110101100001101 01)
ENTER
0b10000000000000111101011000011010
rotate(256,1) ENTER 0b1000000000
```

在十六进制基模式下:

```
rotate(0h78E) ENTER          0h3C7
rotate(0h78E,-2) ENTER 0h800001E3
rotate(0h78E,2) ENTER       0h1E38
```

重要事项: 欲输入十进制或十六进制数字，请一直使用 0b 或 0h 前缀(是零，并非字母 O)。

rotate(*数组1*[,*旋转#*]) ⇒ *数组*

对应由*旋转#*元素转到左边或右边的*数组1*的复制件，不改变*数组1*。

如*旋转#*是正数，就会向左转。如*旋转#*是负数，就会向右转。默认是 ⁻1 (即向右转一个元素)。

在小数基模式下:

```
rotate({1,2,3,4}) ENTER
                    {4 1 2 3}
rotate({1,2,3,4},⁻2) ENTER
                    {3 4 1 2}
rotate({1,2,3,4},1) ENTER
                    {2 3 4 1}
```

rotate(*字符串1*[,*旋转#*]) ⇒ *字符串*

对应由*旋转#*元素转到左边或右边的*字符串1*的复制件，不改变*字符串1*。

如*旋转#*是正数，就会向左转。如*旋转#*是负数，就会向右转。默认是 ⁻1 (即向右转一个字符)。

```
rotate("abcd") ENTER      "dabc"
rotate("abcd",⁻2) ENTER   "cdab"
rotate("abcd",1) ENTER    "bcda"
```

round() MATH/Number 菜单

round(*表达式1*[,*数位*]) ⇒ *表达式*

对应小数点后舍成指定位数的变数。

*数位*必须是范围 0–12 的整数。如不包含*数位*，是对应舍成 12 个有效数位的变数。

备注: 显示数位模式会影响数位显示的效果。

```
round(1.234567,3) ENTER    1.235
```

round(*数组1*[, *数位*]) ⇒ *数组*

对应舍成指定位数的元素数组。

```
round({π,√(2),ln(2)},4) ENTER
                {3.1416 1.4142 .6931}
```

round(*矩阵1*[, *数位*]) ⇒ *矩阵*

对应舍成指定位数的元素矩阵。

```
round([ln(5),ln(3);π,e^(1)],1)
ENTER
                  [1.6  1.1]
                  [3.1  2.7]
```

rowAdd() MATH/Matrix/Row ops 菜单

rowAdd(*矩阵1*, *r索引1*, *r索引2*) ⇒ *矩阵*

对应行*r索引2*被行*r索引1*与行*r索引2*的和取代的*矩阵1*的复制件。

```
rowAdd([3,4;⁻3,⁻2],1,2) ENTER
                    [3 4]
                    [0 2]
rowAdd([a,b;c,d],1,2) ENTER
                  [a     b]
                  [a+c   b+d]
```

rowDim() MATH/Matrix/Dimensions 菜单

rowDim(*矩阵*) ⇒ *表达式*

对应*矩阵*中的行数。

备注: 也请参阅 **colDim()**。

```
[1,2;3,4;5,6]→M1 ENTER
```

$$\begin{bmatrix}1 & 2\\3 & 4\\5 & 6\end{bmatrix}$$

```
rowdim(M1) ENTER
                    3
```

rowNorm() MATH/Matrix/Norms 菜单

rowNorm(*矩阵***)** ⇒ *表达式*

对应*矩阵*中行元素绝对值和的最大值。

备注: 所有矩阵元素必须为数字。也请参阅 **colNorm()**。

```
rowNorm([-5,6,-7;3,4,9;9,-9,-7])
ENTER                                25
```

rowSwap() MATH/Matrix/Row ops 菜单

rowSwap(*矩阵1***,** *r索引1***,** *r索引2***)** ⇒ *矩阵*

对应行*r索引1*与行*r索引2*互换的*矩阵1*。

```
[1,2;3,4;5,6]→Mat ENTER
```

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

```
rowSwap(Mat,1,3) ENTER
```

$$\begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix}$$

RplcPic CATALOG

RplcPic *图形变元*[, *行*][, *列*]

清除作图屏并在象素坐标(*行*, *列*)处放置图象*图形变元*。如您不想清除屏幕，可使用 **RclPic**。

图形变元 必须是图象数据变元。如包含有*行*和*列*，是指定图象左上角的象素坐标。默认坐标是 (0, 0)。

备注: 对于小于全屏的图象，只有受新图象影响的区域才会被清除。

rref() MATH/Matrix 菜单

rref(*矩阵1***[,** *误差***])** ⇒ *矩阵*

对应*矩阵1*化简的行梯形式。

```
rref([-2,-2,0,-6;1,-1,9,-9;
-5,2,4,-4]) ENTER
```

$$\begin{bmatrix} 1 & 0 & 0 & 66/71 \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & -62/71 \end{bmatrix}$$

作为任选项，任何矩阵元素如其绝对值小于*误差*，可会被当作零来处理。此误差只有在矩阵带有浮点输入项且不含有任何未赋值的符号变元时才用到。否则，*误差*会被忽略。

- 如您使用 ◆ ENTER 或将模式设定成 Exact/Approx=APPROXIMATE，那么运算就会使用浮点算术进行。
- 如省略或没有使用*误差*，默认误差是按以下方式计算出来的:

  5E-14 * **max(dim(***矩阵1***))** * **rowNorm(***矩阵1***)**

```
rref([a,b,x;c,d,y]) ENTER
```

$$\begin{bmatrix} 1 & 0 & \frac{d \cdot x - b \cdot y}{a \cdot d - b \cdot c} \\ 0 & 1 & \frac{-(c \cdot x - a \cdot y)}{a \cdot d - b \cdot c} \end{bmatrix}$$

备注: 也请参阅 **ref()**。

Send CATALOG

Send *数组*

CBL™ (Calculator-Based Laboratory™) 或 CBR™ (Calculator-Based Ranger™)指令。将*数组*发送到链接端口。

程序段:

```
  ⋮
:Send {1,0}
:Send {1,2,1}
  ⋮
```

SendCalc CATALOG

SendCalc *变元*

发送*变元*到链接端口，该端口上链接的其它单元就可接收变元值。接收单元必须在主屏幕上或必须从程序中执行 **GetCalc**。

如您从一台 TI-89 或 TI-92 Plus 向一台 TI-92 发送的话，如 TI-92 在程序中执行 **GetCalc**，就会出错。此情形下，发送单元必须使用 **SendChat** 指令来替代。

程序段:

```
  ⋮
:a+b→x
:SendCalc x
  ⋮
```

SendChat CATALOG

SendChat *变元*

SendCalc 的一个常用的替代指令，如接收单元是 TI-92 (或是允许 TI-92 或 TI-92 Plus 使用的普通"聊天"程序)时很有用。更为详尽的信息，请参阅 **SendCalc**。

SendChat 只有在变元与"聊天"程序中典型真实的 TI-92 兼容时才将其发送出去。尽管如此，**SendChat** 不能发送已存档的变元、TI-89 图形数据库等。

程序段:

```
  ⋮
:a+b→x
:SendChat x
  ⋮
```

seq() MATH/List 菜单

seq(*表达式*,*变元*, *低*, *高*[, *分级*]**)** ⇒ *数组*

利用*分级*的增量从*低*到*高*增加*变元*，计算*表达式*和对应结果成数组。**seq()** 完成后，*变元*的原始内容仍然会保留下来。

*变元*不能为系统变元。

*分级*的默认值是 = 1。

```
seq(n^2,n,1,6) ENTER
                {1 4 9 16 25 36}
seq(1/n,n,1,10,2) ENTER
              {1 1/3 1/5 1/7 1/9}
sum(seq(1/n^2,n,1,10,1)) ENTER
                              196...
                              127...
```

或按下 ◆ENTER 来取得: 1.549...

setFold() CATALOG

setFold(*新文件夹名***)** ⇒ *旧文件夹字符串*

对应作为一个字符串的当前文件夹名并设定*新文件夹名*作为当前文件夹。

*新文件夹名*必须已存在。

```
newFold chris ENTER          Done
setFold(main) ENTER        "chris"
setFold(chris)→oldfoldr ENTER
                           "main"
1→a ENTER                      1
setFold(#oldfoldr) ENTER   "chris"
a ENTER                        a
chris\a ENTER                  1
```

setGraph() CATALOG

setGraph(*模式名称字符串*, *设置字符串*) ⇒ *字符串*

把 Graph 模式*模式名称字符串*设定为*设置字符串*，并对应模式的前一设置，储存前一设置可让您稍后再重新储存之。

*模式名称字符串*是指定您要设定哪种模式的字符串。该字符串必须是以下表格中的模式名称之一。

*设置字符串*是指定新的模式设置的字符串。它必须是下面列出的设置之一。

```
setGraph("Graph Order","Seq")
ENTER                          "SEQ"
setGraph("Coordinates","Off")
ENTER                          "RECT"
```

备注: 当输入模式名称时，大写字母开头和空白是选项。

模式名称	设置	
"Coordinates"	"Rect", "Polar", "Off"	
"Graph Order"	"Seq", "Simul" [1]	
"Grid"	"Off", "On" [2]	
"Axes"	"Off", "On" "Off", "Axes", "Box"	(非 3D 作图模式) (3D 作图模式)
"Leading Cursor"	"Off", "On" [2]	
"Labels"	"Off", "On"	
"Style"	"Wire Frame", "Hidden Surface", "Contour Levels", "Wire and Contour", "Implicit Plot" [3]	
"Seq Axes"	"Time", "Web", "U1-vs-U2" [4]	
"DE Axes"	"Time", "t-vs-y' ", "y-vs-y' ", "y1-vs-y2", "y1-vs-y2' ", "y1'-vs-y2' " [5] **提示:** 欲键入质数符号(')，可按下 2nd [′]。	
"Solution Method"	"RK", "Euler" [5]	
"Fields"	"SlpFld", "DirFld", "FldOff" [5]	

[1] 在序列、3D 或微分方程作图模式下不可用。
[2] 在 3D 作图模式下不可用。
[3] 只应用于 3D 作图模式。
[4] 只应用于序列作图模式。
[5] 只应用于微分方程作图模式。

setMode() CATALOG

setMode(*模式名称字符串***, *设置字符串*)** ⇒ *字符串*
setMode(*数组***)** ⇒ *字符串数组*

将模式*模式名称字符串*设定成新设置*设置字符串*并对应该模式的当前设置。

*模式名称字符串*是您要设定哪种模式的字符串。它必须是以下表格中的模式名称之一。

*设置字符串*是指定新的模式设置的字符串。它必须是下面列出的设置之一。

*数组*含有关键字符串对并将设定它们一次全部设定。这是建议给多模式更改时使用的。例子中示出了如每对按下面的顺序输入了分隔的 **setMode()** 是不工作的。

使用 **setMode(***变元***)** 来储存用 **getMode("ALL")→** *变元* 保存的设置。

备注: 有关 Unit System 模式的设定或返回信息，请使用 **setUnits()** 或 **getUnits()** 而不要使用 **setMode()** 或 **getMode()**。

```
setMode("Angle","Degree")
ENTER                          "RADIAN"

sin(45) ENTER                   √2/2

setMode("Angle","Radian")
ENTER                          "DEGREE"

sin(π/4) ENTER                  √2/2

setMode("Display Digits",
"Fix 2") ENTER                 "FLOAT"

π ◆ ENTER                        3.14

setMode ("Display Digits",
"Float") ENTER                 "FIX 2"

π ◆ ENTER                       3.141...

setMode ({"Split Screen",
"Left-Right","Split 1 App",
"Graph","Split 2
App","Table"})
ENTER
          {"Split 2 App" "Graph"
            "Split 1 App" "Home"
          "Split Screen" "FULL"}
```

备注: 当输入模式名称时，大写字母开头和空白是选项。此外，这些例子的结果也许会与您的单元所显示的不同。

模式名称	设置
"Graph"	"Function", "Parametric", "Polar", "Sequence", "3D", "Diff Equations"
"Display Digits"	"Fix 0", "Fix 1", ..., "Fix 12", "Float", "Float 1", ..., "Float 12"
"Angle"	"Radian", "Degree"
"Exponential Format"	"Normal", "Scientific", "Engineering"
"Complex Format"	"Real", "Rectangular", "Polar"
"Vector Format"	"Rectangular", "Cylindrical", "Spherical"
"Pretty Print"	"Off", "On"
"Split Screen"	"Full", "Top-Bottom", "Left-Right"
"Split 1 App"	"Home", "Y= Editor", "Window Editor", "Graph", "Table", "Data/Matrix Editor", "Program Editor", "Text Editor", "Numeric Solver", "*Flash App*"
"Split 2 App"	"Home", "Y= Editor", "Window Editor", "Graph", "Table", "Data/Matrix Editor", "Program Editor", "Text Editor", "Numeric Solver", "*Flash App*"
"Number of Graphs"	"1", "2"
"Graph2"	"Function", "Parametric", "Polar", "Sequence", "3D", "Diff Equations"
"Split Screen Ratio"	"1:1", "1:2", "2:1" (仅对 TI-92 Plus)
"Exact/Approx"	"Auto", "Exact", "Approximate"
"Base"	"Dec", "Hex", "Bin"
"Language"	"English", "*Alternate Language*"

setTable() CATALOG

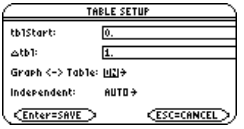
setTable(*模式名称字符串*, *设置字符串*) ⇒ *字符串*

设定表格参数*模式名称字符串*或*设置字符串*并对应参数的前一设置，储存前一设置可让您稍后再重新储存之。

*模式名称字符串*是指定哪个参数要设定的字符串。它必须是下表中的参数之一。

*设置字符串*是指定参数新设置的字符串。它必须是下表中您要的参数设置中的一个。

```
setTable("Graph <->
Table","ON")
ENTER                    "OFF"
setTable("Independent","AUTO")
ENTER                    "ASK"
◆ [TblSet]
```



备注: 当输入参数时，大写字母开头和空白是选项。

参数名称	设置
"Graph <-> Table"	"Off", "On"
"Independent"	"Auto", "Ask"

setUnits() CATALOG

setUnits(*数组1***)** ⇒ *数组*

将默认单位设定成*数组1*中指定的值，并对应前一默认值的数组。

- 欲指定内建的 SI (公制) 或 ENG/US 制，*数组1* 是使用如下形式的:

  {"SI"} 或 {"ENG/US"}

- 欲指定自默认单位的定义集，*数组1* 是使用以下形式的:

  {"CUSTOM", "*类别1*", "*单位1*"[, "*类别2*", "*单位2*", ...}

  其中没一*类别*和*单位*对是指定类别极其默认单位的。(您只可指定内建单位，不能指定用户定义单位)。任何没有指定的类别将会使用它的前一自定单位。

- 欲返回前一自定默认单位，*数组1*是利用以下形式的:

  {"CUSTOM"}

如您想要不同的默认值，视情形而定，您可创建单独的数组并将它们保存成唯一的数组名称。欲使用默认值集，请在 **setUnits()** 指定该数组的名称。

您可使用 **setUnits()** 来将先前用 **setUnits()** → *变元*或 **getUnits()** → *变元*保存的设置值重新储存起来。

所有单位名称必须以下划线 _ 开始。

TI-89: ◆ [_]
TI-92 Plus: 2nd [_]

您也可按下以下按键来从菜单上选择单位:

TI-89: 2nd [UNITS]
TI-92 Plus: ◆ [UNITS]

```
setUnits({"SI"}) ENTER
          {"SI"  "Area"  "NONE"
      "Capacitance"  "_F"  ...}

setUnits({"CUSTOM","Length",
"_cm","Mass","_gm"}) ENTER
          {"SI"  "Length"  "_m"
            "Mass"  "_kg"  ...}
```

备注: 您的屏幕也许会显示不同的单位。

Shade CATALOG

Shade *表达式1*, *表达式2*, [*x低*], [*x高*], [*图案*], [*图案分别率*]

显示作图屏，作出*表达式1*和*表达式2*的图并遮挡*表达式1*小于*表达式2*的区域。(*表达式1*和*表达式2*必须为使用 x 作为自变元的表达式)。

如含有*x低*和*x高*，是指定阴影的左或右边界。有效的输入是 xmin 和 xmax 之间。默认值是 xmin 和 xmax。

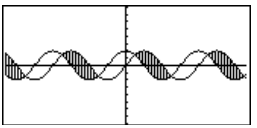
*图案*是指定四中阴影图案之一:
1 = 垂直(默认)
2 = 水平
3 = 反向倾斜 45°
4 = 正向倾斜 45°

*图案分别率*是指定阴影图案的分别率:
1= 普通阴影
2= 1 个象素间隔(默认)
3= 2 个象素间隔
⋮
10= 9 个象素间隔

备注: 交互阴影在作图屏上通过 **Shade** 指令可用。一个特定函数的自动阴影通过 **Style** 指令可用。**Shade** 在 3D 作图模式下不可用。

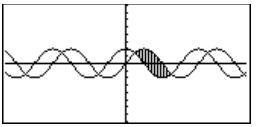
在 ZoomTrig 观察窗下:

```
Shade cos(x),sin(x) ENTER
```



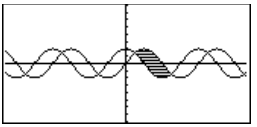
TI-89: HOME
TI-92 Plus: ◆ [HOME]

```
ClrDraw ENTER                    Done
Shade cos(x),sin(x),0,5 ENTER
```



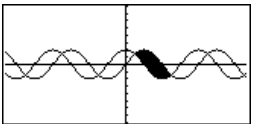
TI-89: HOME
TI-92 Plus: ◆ [HOME]

```
ClrDraw ENTER                    Done
Shade cos(x),sin(x),0,5,2 ENTER
```



TI-89: HOME
TI-92 Plus: ◆ [HOME]

```
ClrDraw ENTER                    Done
Shade cos(x),sin(x),0,5,2,1
ENTER
```



shift(*整数1*[,*转换#*]) ⇒ *整数*

将位转换成十进制整数。您可以任何数字基输入*整数1*；它会自动转换成标记的 32 位十进制形式。如*整数1*的数量对于此形式过大，就会采用对称模运算来将数值带入近似的范围内。

如*转换#*是正数，就会向左转换；如*转换#*是负数，就会向右转换。默认是 ⁻1 (向右转换一个位)。

在向右转换时，最右边的位会下沉并插入 0 或 1 来匹配最左边的位。在向左转换时，最左边的位会下沉并想最右边的位一样插入 0。

例如，在向右转换时:

每个位向右转换。

0b00000000000000111101011000011010 1

如最左边的位是 0 就插入 0，或如果是 1 就插入 1。

下沉

产生:

0b00000000000000111101011000011010

结果会按照 Base 模式显示出来。带头的零不会显示出。

在十进制基模式下:

```
shift(0b1111010110000110101) ENTER
                0b111101011000011010
shift(256,1) ENTER
                          0b1000000000
```

在十六进制基模式下:

```
shift(0h78E) ENTER          0h3C7
shift(0h78E,⁻2) ENTER       0h1E3
shift(0h78E,2) ENTER       0h1E38
```

重要事项: 欲输入十进制或十六进制数字，请一直使用 0b 或 0h 前缀(是零，并非字 母 O)。

---

shift(*数组1* [,*转换#*]) ⇒ *数组*

对应由*转换#*元素转到左边或右边的*数组1*的复制件，不改变*数组1*。

如*转换#*是正数，就会向左转。如*转换#*是负数，就会向右转。默认是 ⁻1 (即向右转一个元素)。

由转换插入在*数组* 的开头或结尾的元素会设定成符号 "undef"。

在小数基模式下:

```
shift({1,2,3,4}) ENTER
                    {undef 1 2 3}
shift({1,2,3,4},⁻2) ENTER
                 {undef undef 1 2}
shift({1,2,3,4},1) ENTER
                    {2 3 4 undef}
```

---

shift(*字符串1* [,*转换#*]) ⇒ *字符串*

对应由*转换#*字符转到左边或右边的*字符串1*的复制件，不改变 *字符串1*。

如*转换#*是正数，就会向左转。如*转换#*是负数，就会向右转。默认是 ⁻1 (即向右转一个字符)。

由转换插入在*字符串*的开头或结尾的元素会设定成一个空格。

```
shift("abcd") ENTER          " abc"
shift("abcd",⁻2) ENTER       "  ab"
shift("abcd",1) ENTER        "bcd "
```

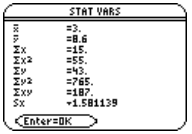
ShowStat CATALOG

ShowStat

显示含有上次计算出的统计结果(如仍然有效的话)的对话框。如计算它们的数据已改变，统计结果会自动清除。

在统计运算后使用此指令，如 **LinReg**。

```
{1,2,3,4,5}→L1 ENTER {1 2 3 4 5}
{0,2,6,10,25}→L2 ENTER
                    {0 2 6 10 25}
TwoVar L1,L2 ENTER
ShowStat ENTER
```



sign() MATH/Number 菜单

sign(*表达式1***)** ⇒ *表达式*
sign(*数组1***)** ⇒ *数组*
sign(*矩阵1***)** ⇒ *矩阵*

对于实数和复数*表达式1*，当*表达式1*≠ 0 时，是对应*表达式1*/**abs(***表达式1***)**。

如*表达式1*是正数则对应 1。
如*表达式1*是负数则对应 ⁻1。
如复数格式模式是 REAL，**sign(0)** 则对应 ±1。否则是对应自己。
sign(0) 代表复数域中的单位圆。

对于数组或矩阵，是对应所有元素的符号。

```
sign(⁻3.2) ENTER                ⁻1.
sign({2,3,4,⁻5}) ENTER
                      {1  1  1  ⁻1}
sign(1+abs(x)) ENTER              1
```

如复数格式模式是 REAL:

```
sign([⁻3,0,3]) ENTER    [⁻1 ±1 1]
```

simult() MATH/Matrix 菜单

simult(*系数矩阵***,** *常数向量***[,** *误差***])** ⇒ *矩阵*

对应含有线性方程组的解的列向量。

*系数矩阵*必须是含有方程系数的平方矩阵。

常数向量 必须具有与*系数矩阵*相同的行数 (相同维数)并含有常数。

作为选项，任何矩阵元素如其绝对值小于*误差*，可会被当作零来处理。此误差只有在矩阵带有浮点输入项且不含有任何未赋值的符号变元时才用到。否则，*误差*会被忽略。

- 如您使用 ◆ ENTER 或将模式设定成 Exact/Approx=APPROXIMATE，运算就会利用浮点算法进行。
- 如省略了*误差*或不使用之，默认误差是计算成:

  5E⁻14 ∗ **max(dim(***系数矩阵***))** ∗ **rowNorm(***系数矩阵***)**

解 x 和 y:

$$x + 2y = 1$$
$$3x + 4y = {}^{-}1$$

```
simult([1,2;3,4],[1;⁻1]) ENTER
                          [⁻3 ; 2]
```

解为 x=⁻3 和 y=2。

解:

$$ax + by = 1$$
$$cx + dy = 2$$

```
[a,b;c,d]→matx1 ENTER        [a b ; c d]
simult(matx1,[1;2]) ENTER
```

$$\begin{bmatrix} \dfrac{-(2\cdot b-d)}{a\cdot d-b\cdot c} \\ \dfrac{2\cdot a-c}{a\cdot d-b\cdot c} \end{bmatrix}$$

simult(*系数矩阵*, *常数矩阵*[, *误差*]**)** ⇒ *矩阵*

解多线性方程组，其中每组具有相同的方程系数但常数不同。

*常数矩阵*中的每一列必须含有方程组的常数。结果矩阵中的每一列含有相应方程组的解。

解: $x + 2y = 1$, $3x + 4y = -1$; $x + 2y = 2$, $3x + 4y = -3$

simult([1,2;3,4],[1,2;-1,-3]) ENTER

$$\begin{bmatrix} -3 & -7 \\ 2 & 9/2 \end{bmatrix}$$

对于第一组，x=-3 和 y=2。对于第二组，x=-7 和 y=9/2。

sin()

TI-89: 2nd [SIN] 键 **TI-92 Plus:** SIN 键

sin(*表达式1***)** ⇒ *表达式*
sin(*数组1***)** ⇒ *数组*

sin(*表达式1***)**对应作为表达式的变数的正弦。

sin(*数组1***)**对应*数组1*中所有元素的正弦数组。

备注: 变数会按照当前的角模式当作度或弧度计算。您可使用 ° 或 ʳ 来临时覆盖角模式设置。

在 Degree 角模式下:

sin((π/4)ʳ) ENTER $\frac{\sqrt{2}}{2}$

sin(45) ENTER $\frac{\sqrt{2}}{2}$

sin({0,60,90}) ENTER $\{0 \quad \frac{\sqrt{3}}{2} \quad 1\}$

在 Radian 角模式下:

sin(π/4) ENTER $\frac{\sqrt{2}}{2}$

sin(45°) ENTER $\frac{\sqrt{2}}{2}$

sin(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵正弦。这与计算每一元素的正弦不同。有关计算方式的信息，请参阅 **cos()**。

*平方矩阵1*必须为可对角的。结果一直含有浮点数字。

在 Radian 角模式下:

sin([1,5,3;4,2,1;6,-2,1]) ENTER

$$\begin{bmatrix} .942\ldots & -.045\ldots & -.031\ldots \\ -.045\ldots & .949\ldots & -.020\ldots \\ -.048\ldots & -.005\ldots & .961\ldots \end{bmatrix}$$

$\sin^{-1}()$

TI-89: ◆ [SIN⁻¹] 键 **TI-92 Plus:** 2nd [SIN⁻¹] 键

sin⁻¹(*表达式1***)** ⇒ *表达式*
sin⁻¹(*数组1***)** ⇒ *数组*

sin⁻¹(*表达式1***)** 对应正弦是作为表达式的*表达式1*的角。

sin⁻¹(*数组1***)** 对应*数组1*每一元素的反正弦数组。

备注: 结果会按当前模式设置对应角或弧度。

在 Degree 角模式下:

sin⁻¹(1) ENTER 90

在 Radian 角模式下:

sin⁻¹({0,.2,.5}) ENTER

{0 .201… .523…}

sin⁻¹(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵反正弦。这与计算每一元素的反正弦不同。有关计算方式的信息，请参阅 **cos()**。

*平方矩阵1*必须为可对角的。结果一直含有浮点数字。

在 Radian 角模式和 Rectangular 复数格式模式下:

sin⁻¹([1,5,3;4,2,1;6,-2,1]) ENTER

$$\begin{bmatrix} -.164\ldots-.064\ldots\cdot i & 1.490\ldots-2.105\ldots\cdot i & \ldots \\ .725\ldots-1.515\ldots\cdot i & .947\ldots-.778\ldots\cdot i & \ldots \\ 2.083\ldots-2.632\ldots\cdot i & -1.790\ldots+1.271\ldots\cdot i & \ldots \end{bmatrix}$$

sinh() MATH/Hyperbolic 菜单

sinh(*表达式1***)** ⇒ *表达式*
sinh(*数组1***)** ⇒ *数组*

sinh (*表达式1***)** 对应作为表达式的变数的双曲正弦。

sinh (*数组***)** 是对应*数组1*每一元素的双曲正弦数组。

```
sinh(1.2) ENTER                    1.509...

sinh({0,1.2,3.}) ENTER
                    {0   1.509...   10.017...}
```

---

sinh(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵双曲正弦。这与计算每一元素的双曲正弦*不*同。有关计算方式的信息，请参阅 **cos()**。

平方矩阵1 必须为可对角的。结果一直含有浮点数字。

在 Radian 角模式下:

```
sinh([1,5,3;4,2,1;6,-2,1])
ENTER
```

$$\begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix}$$

$\sinh^{-1}()$ MATH/Hyperbolic 菜单

$\sinh^{-1}$(*表达式1***)** ⇒ *表达式*
$\sinh^{-1}$(*数组1***)** ⇒ *数组*

$\sinh^{-1}$(*表达式1***)** 对应作为表达式的变数的双曲反正弦。

$\sinh^{-1}$(*数组1***)** 是对应*数组1*每一元素的双曲反正弦数组。

```
sinh⁻¹(0) ENTER                    0

sinh⁻¹({0,2.1,3}) ENTER
                    {0   1.487...   sinh⁻¹(3)}
```

$\sinh^{-1}$(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵双曲反正弦。这与计算每一元素的双曲正弦*不*同。有关计算方式的信息，请参阅 **cos()**。

*平方矩阵1*必须为可对角的。结果一直含有浮点数字。

在 Radian 角模式下:

```
sinh⁻¹([1,5,3;4,2,1;6,-2,1])
ENTER
```

$$\begin{bmatrix} .041... & 2.155... & 1.158... \\ 1.463... & .926... & .112... \\ 2.750... & -1.528... & .572... \end{bmatrix}$$

SinReg　MATH/Statistics/Regressions 菜单

SinReg *数组1*, *数组2* [, [*重复*], [*周期*] [, *数组3*, *数组4*]]

计算正弦回归并更新所有系统统计变元。

所有数组必须具有等同的维数，*数组4*除外。

数组1 代表 x 数组。
数组2 代表 y 数组。
数组3 代表类别代码。
数组4 代表类别包含数组。

*重复*是指定尝试求解的最大时间数(1 到 16)。如省略之，会使用 8。传统而言，较大数值的结果会更精确但耗费的时间较长，反之也成立。

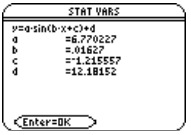
*周期*是指定估计的周期。如省略之，*数组1*中数值之间的差别必须等同，并是序列的排序。如您指定*周期*，x 值之间的差别可为不等同的。

备注: *数组1*到*数组3*必须为变元名称或 c1-c99 (数据 / 矩阵编辑器显示的上一数据变元的列)。*数组4*不必为变元名称并不能为 c1-c99。

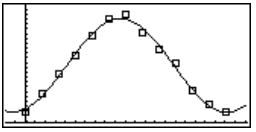
SinReg 的输出一直是弧度，与角模式的设置无关。

在函数作图模式下:

```
seq(x,x,1,361,30)→L1 [ENTER]
                          {1 31 61 …
{5.5,8,11,13.5,16.5,19,19.5,17,
14.5,12.5,8.5,6.5,5.5}→L2 [ENTER]
                          {5.5 8 11 …
SinReg L1,L2 [ENTER]            Done
ShowStat [ENTER]
```




```
[ENTER]
regeq(x)→y1(x) [ENTER]          Done
NewPlot 1,1,L1,L2 [ENTER]       Done
[◆][GRAPH]
[F2] 9
```



solve()　MATH/Algebra 菜单

solve(*方程*, *变元***)** ⇒ *布尔表达式*
solve(*不等式*, *变元***)** ⇒ *布尔表达式*

对应*变元*的方程或不等式的候选实数解。目的是对应所有解的候选值。尽管如此，这些也许是解的数量无穷的方程或不等式。

```
solve(a*x^2+b*x+c=0,x) [ENTER]
```

$$x = \frac{\sqrt{b^2-4\cdot a\cdot c}-b}{2\cdot a}$$

$$\text{or } x = \frac{-(\sqrt{b^2-4\cdot a\cdot c}+b)}{2\cdot a}$$

对于未定义变元的某些数值组合，解的候选值不能是实数有穷解。

```
ans(1)| a=1 and b=1 and c=1
[ENTER]
          Error: Non-real result
```

对于 Exact/Approx 模式的 AUTO 设置，目的是在解是余弦，并由近似算法的交重复寻得到补充，当精确解不切合实际时产生精确解。

```
solve((x-a)e^(x)=-x*(x-a),x)
[ENTER]
                x = a or x =-.567…
```

鉴于比率分子与分母的最大公约数默认相约，也许仅限制在一或两边的解。

```
(x+1)(x-1)/(x-1)+x-3 [ENTER]
                          2·x-2
solve(entry(1)=0,x) [ENTER]   x = 1
entry(2)|ans(1) [ENTER]      undef
limit(entry(3),x,1) [ENTER]      0
```

对于 ≥、≤、< 或 > 之类的不等式，除非不等式为线性并只含有*变元*，明显的解是不可靠的。

```
solve(5x-2 ≥ 2x,x) [ENTER]   x ≥ 2/3
```

对于 Exact/Approx 模式的 EXACT 设置，不能求解的部分是对应成含蓄方程或不等式。

```
exact(solve((x-a)e^(x)=-x*
  (x-a),x)) ENTER
```

$e^x + x = 0$ or $x = a$

使用 "|" 算子限制解的区间和或方程或不等式中出现的其它变元。当您在一个区间中求解时，您可使用不等算子来在以后的搜寻之中免除该区间。

在 Radian 角模式下:

```
solve(tan(x)=1/x,x)|x>0 and
x<1 ENTER
```

$x = .860...$

当求不到实数解时是对应 false。当 **solve()** 可定义满足方程或不等式的*变元*的任何有穷实数值时是对应 true。

```
solve(x=x+1,x) ENTER
```

false

```
solve(x=x,x) ENTER
```

true

由于 **solve()** 一直是对应一个布尔结果，您可使用 "and,"、"or," 和 "not" 来组合 **solve()** 彼此间的或其它布尔表达式的结果。

```
2x-1≤1 and solve(x^2≠9,x) ENTER
```

$x \le 1$ and $x \ne -3$

解也许会含有形式 @n*j* 的唯一新未定义变元，*j* 开始一个区间在 1–255 的整数，该变元是指明一个任意整数。

在 Radian 角模式下:

```
solve(sin(x)=0,x) ENTER
```

$x = @n1 \cdot \pi$

在实数模式下，分数幂具有奇数分母，仅表示实数分支。相反，多分支的表达式如分数幂、对数和反三角函数只是表示主分支。因此，**solve()** 只产生对应该实数或主分支的解。

```
solve(x^(1/3)=-1,x) ENTER
```

$x = -1$

```
solve(√(x)=-2,x) ENTER
```

false

备注: 也请参阅 **cSolve()**、**cZeros()**、**nSolve()** 和 **zeros()**。

```
solve(-√(x)=-2,x) ENTER
```

$x = 4$

---

solve(*方程1* **and** *方程2* [**and** ...], **{***变元或猜测值1*, *变元或猜测值2* [, ...]**})** ⇒ *布尔表达式*

```
solve(y=x^2-2 and
x+2y=-1,{x,y}) ENTER
```

$x=1$ and $y=-1$ or $x=-3/2$ and $y=1/4$

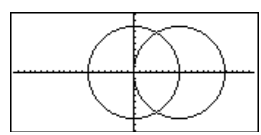
对应联立代数方程的候选实数解，其中每一*变元或猜测值*是指定您要求解的变元。

作为选项，您可为变元指定一个起始的猜测值。每一*变元或猜测值*必须具有如下形式:

变元
– 或 –
变元 = 实数或非实数

例如，x 是有效的所以 x=3。

如所有方程均为多项式及没指定一个起始的猜测值，**solve()** 就会利用词汇的 Gröbner /Buchberger 消除方式来尝试定义所有的实 数解。



例如，假定您有一个位于坐标原点、半径为 **r** 的圆和另一个半径为 **r**，圆心位于第一个圆与正 **x** 轴相交处的圆，利用 **solve()** 求相交点。

如右图例子中的 **r** 所示，联立*多项式方程*可有无值的其它变元，但它是代表稍后要被替换的给出数值。

```
solve(x^2+y^2=r^2 and
(x-r)^2+y^2=r^2,{x,y}) ENTER
```

$x=\frac{r}{2}$ and $y=\frac{\sqrt{3}\cdot r}{2}$ or $x=\frac{r}{2}$ and $y=\frac{-\sqrt{3}\cdot r}{2}$

您也可(或不)包含方程中没有出现的解变元。例如，您可将作为解变元的 **z** 包含进去，以将前一例子扩展成具有两个平行交点的半径 **r** 柱面。

```
solve(x^2+y^2=r^2 and
(x-r)^2+y^2=r^2,{x,y,z}) ENTER
```

柱面解示出多少解族含有形式 @k 的任意常数，其中 k 是一个从 1 到 255 的整数前缀。当您使用 **ClrHome** 或 F1 8:Clear Home 时，前缀会重设到 1。

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ and } z=@1$$

$$\text{or } x=\frac{r}{2} \text{ and } y=\frac{-\sqrt{3}\cdot r}{2} \text{ and } z=@1$$

对于多项式组，计算时间或内存耗费很大程度取决于您列出解变元的顺序。如您的起始选择耗用内存或花费您的耐性，可尝试将方程和 / 或*变元或猜测值*数组中的变元重新整理一下。

如您没有将任何猜测值包含进去，或如任何方程在何变元中是非多项式，但所有方程在所有解变元中均为一次的话，**Solve()** 会使用高斯消除方式来尝试定义所有解。

```
solve(x+e^(z)*y=1 and
x-y=sin(z),{x,y}) ENTER
```

$$x=\frac{e^z\cdot\sin(z)+1}{e^z+1} \text{ and } y=\frac{-(\sin(z)-1}{e^z+1}$$

如方程组在其所有变元中既不是多项式，在解变元中也不是一次的话，**Solve()** 会使用近似重复方式定义最多一个解。为了这样做，解变元数必须等于方程数，方程中的其它所有变元必须化简成数字。

```
solve(e^(z)*y=1 and
-y=sin(z),{y,z}) ENTER
            y=.041… and z=3.183…
```

如仅存在一个的话，每一解变元会从其猜测的值开始；否则会从 0.0 开始。

```
solve(e^(z)*y=1 and
-y=sin(z),{y,z=2π}) ENTER
            y=.001… and z=6.281…
```

利用猜测值来逐个寻找其它的解。对于收敛，猜测值反而会包含在解内。

SortA　MATH/List 菜单

SortA *数组名称1*[, *数组名称2*] [, *数组名称3*] ...
SortA *向量名称1*[, *向量名称2*] [, *向量名称3*] ...

将第一变数的元素按递增顺序分类。

如您包含额外的变数，是将每一变数的元素分类，使它们的新位置匹配第一变数的元素的新位置。

所有变数必须是数组或向量的名称。所有变数必须具有等同的维数。

```
{2,1,4,3}→list1 ENTER    {2,1,4,3}
SortA list1 ENTER              Done

list1 ENTER              {1 2 3 4}
{4,3,2,1}→list2 ENTER    {4 3 2 1}
SortA list2,list1 ENTER        Done

list2 ENTER              {1 2 3 4}
list1 ENTER              {4 3 2 1}
```

SortD　MATH/List 菜单

SortD *数组名称1*[, *数组名称2*] [, *数组名称3*] ...
SortD *向量名称1*[,*向量名称2*] [,*向量名称3*] ...

除了 **SortD** 是将元素按递减顺序分类外，与 **SortA** 完全一样。

```
{2,1,4,3}→list1 ENTER    {2 1 4 3}
{1,2,3,4}→list2 ENTER    {1 2 3 4}
SortD list1,list2 ENTER        Done
list1 ENTER              {4 3 2 1}
list2 ENTER              {3 4 1 2}
```

▸Sphere　MATH/Matrix/Vector ops 菜单

向量▸**Sphere**

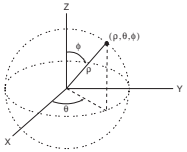
以球面形式 [ρ ∠θ ∠ϕ] 显示行或列向量。

向量 必须是维数 3，可以是行或列向量。

备注: ▸**Sphere** 是一个显示格式指令，并非一个换算函数。您只可在输入行结尾使用它。

```
[1,2,3]▸Sphere
◆ ENTER [3.741… ∠1.107… ∠.640…]
[2,∠π/4,3]▸Sphere
◆ ENTER [3.605… ∠.785… ∠.588…]
```

ENTER $\left[\sqrt{13}\ \angle\frac{\pi}{4}\ \angle\cos^{-1}\left(\frac{3\cdot\sqrt{13}}{13}\right)\right]$



stdDev()　MATH/Statistics 菜单

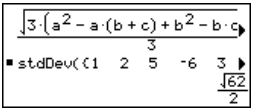
stdDev(*数组***[,** *频数数组***])** ⇒ *表达式*

对应*数组*中元素的标准差。

每一*频数数组*元素是计算*数组*中相应元素连续出现的数目。

备注: *数组*起码要有两个元素。

```
stdDev({a,b,c}) ENTER
stdDev({1,2,5,⁻6,3,⁻2}) ENTER
```




```
stdDev({1.3,2.5,-6.4},{3,2,5})
ENTER                    4.33345
```

---

stdDev(*矩阵1***[,** *频数矩阵***])** ⇒ *矩阵*

对应*矩阵1*中列的标准差的行向量。

每一*频数矩阵*元素是计算*矩阵1*中相应元素连续出现的数目。

备注: *矩阵1*起码要有两行。

```
stdDev([1,2,5;⁻3,0,1;.5,.7,3])
ENTER
          [2.179…  1.014…  2]
stdDev([-1.2,5.3;2.5,7.3;6,-4],
[4,2;3,3;1,7]) ENTER
          [2.7005,5.44695]
```

StoGDB　CATALOG

StoGDB *GDB变元*

创建含有以下当前项目的作图数据库(GDB)变元:

* 图形模式
* Y= 函数
* 窗口变元
* 作图格式设置
  1- 或 2-Graph 设置(如过是 2-Graph 模式，是拆屏和比率设置)
  角模式
  实数 / 复数模式
* 如处于序列或微分方程模式则是起始条件
* 表格旗
* tbl*开始*, Δtbl, tblInput

您可使用 **RclGDB** *GDB变元*来重新储存作图的环境。

*__备注:__ 在 2-Graph 作图模式下，此项是为两个图形而保存的。

Stop CATALOG

Stop

作为程序指令来停止程序执行。

程序段:

```
  ⋮
For i,1,10,1
  If i=5
  Stop
EndFor
  ⋮
```

StoPic CATALOG

StoPic *图形变元* [, *象素行*, *象素列*] [, *宽*, *高*]

显示作图屏并将显示的直角区域复制到*图形变元*中。

如包含*象素行*和*象素列*，是指定要复制区域的左上角(默认是 0, 0)。

如包含*宽*和*高*，是以象素指定区域的维数。默认值是当前作图屏的宽度和高度，单位是象素。

Store

请参阅第 217 页的 → (储存)。

string() MATH/String 菜单

string(*表达式***)** ⇒ *字符串*

化简*表达式*并对应作为字符串的结果。

```
string(1.2345) ENTER        "1.2345"
string(1+2) ENTER                "3"
string(cos(x)+√(3)) ENTER
                      "cos(x)+√(3)"
```

Style CATALOG

Style *方程数*, *样式属性字符串*

设定当前作图模式中的系统作图函数*方程数*使用图形属性*样式属性字符串*。

*方程数*必须是 1–99 的整数，且函数必须已经存在。

*styleProperty字符串*必须是以下项目之一: "Line"、"Dot"、"Square"、"Thick"、"Animate"、"Path"、"Above" 或 "Below"。

请注意，在参数作图中，只有对的 *xt* 一半含有样式信息。

Vali有效样式名称对作图模式:

函数:	所有模式
参数 / 极坐标:	线、点、方块、厚边、动画、路径
序列:	线、点、方块、厚边、
3D:	无
微分方程:	线、点、方块、厚边、动画、路径

备注: 当输入*样式属性字符串*名称时，大写字母开头和空白是选项。

```
Style 1,"thick" ENTER        Done
Style 10,"path" ENTER        Done
```

备注: 在函数作图模式下，这些例子是将 y1(x) 的样式设定成 "Thick" 和将 y10(x) 设定成 "Path"。

subMat() CATALOG

subMat(*矩阵1***[,** *开始行***] [,** *开始列***] [,** *结束行***] [,** *结束列***])** ⇒ *矩阵*

对应*矩阵1*指定的子矩阵。

默认值: *开始行*=1、*开始列*=1、*结束行*=最后的行、*结束列*=最后的列。

```
[1,2,3;4,5,6;7,8,9]→m1 ENTER
```

$$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$$

```
subMat(m1,2,1,3,2) ENTER
```

$$\begin{bmatrix}4 & 5\\7 & 8\end{bmatrix}$$

```
subMat(m1,2,2) ENTER
```

$$\begin{bmatrix}5 & 6\\8 & 9\end{bmatrix}$$

sum() MATH/List 菜单

sum(*数组***[,** *开始***[,** *结束***]])** ⇒ *表达式*

对应*数组*中元素的和。

*开始*和*结束*均为任选项，它们是指定元素的范围的。

```
sum({1,2,3,4,5}) ENTER          15
sum({a,2a,3a}) ENTER          6·a
sum(seq(n,n,1,10)) ENTER        55
sum({1,3,5,7,9},3) ENTER        21
```

---

sum(*矩阵1***[,** *开始***[,** *结束***]])** ⇒ *矩阵*

对应含有*矩阵1*列中的元素和的行向量。

*开始*和*结束*为任选项，它们是指定行范围的。

```
sum([1,2,3;4,5,6]) ENTER [5   7   9]
sum([1,2,3;4,5,6;7,8,9]) ENTER
                        [12 15 18]
sum([1,2,3;4,5,6;7,8,9],2,3)
ENTER
                        [11,13,15]
```

switch() CATALOG

switch([*整数1***])** ⇒ *整数*

对应活动窗口数，也可设定活动窗口。

备注: Window 1 是左边或顶部；Window 2 是右边或底部。

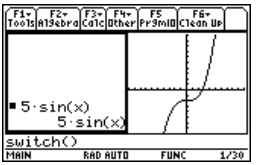
如*整数1* = 0，对应活动窗口数。

如*整数1* = 1，是使窗口 1 变成活动并对应前一活动窗口数。

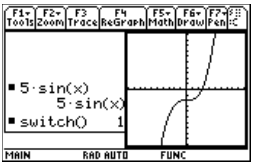
如*整数1* = 2，是使窗口 2 变成活动并对应前一活动窗口数。

如*整数1* 被省略，是切换窗口并对应前一活动窗口数。

如 TI-89 / TI-92 Plus 不在显示拆分屏幕，那么*整数1*会被忽略。




```
switch() ENTER
```



T (转置) MATH/Matrix 菜单

*矩阵1*T ⇒ *矩阵*

对应*矩阵1*的复数共轭转置。

[1,2,3;4,5,6;7,8,9]→mat1 ENTER

$$\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix}$$

mat1T ENTER

$$\begin{bmatrix}1&4&7\\2&5&8\\3&6&9\end{bmatrix}$$

[a,b;c,d]→mat2 ENTER $\begin{bmatrix}a&b\\c&d\end{bmatrix}$

mat2T ENTER $\begin{bmatrix}a&c\\b&d\end{bmatrix}$

[1+*i*,2+*i*;3+*i*,4+*i*]→mat3 ENTER

$$\begin{bmatrix}1+i&2+i\\3+i&4+i\end{bmatrix}$$

mat3T ENTER $\begin{bmatrix}1-i&3-i\\2-i&4-i\end{bmatrix}$

Table CATALOG

Table *表达式1*[, *表达式2*] [, *变元1*]

建立指定的表达式或函数的表格。

表格中的表达式也可作出图形。利用 **Table** 或 **Graph** 命令输入的表达式会被指定成以 1 开头的增量函数。在按下 F4 Header 并显示出表格时，表达式可通过可用的编辑功能来修改或单独删除。在 Y=编辑器中当前选择的函数会暂时忽略。

欲清除由 **Table** 或 **Graph** 创建的函数，可执行 **ClrGraph** 命令或显示 Y=编辑器。

如*变元*参数被省略，当前的作图模式自变元会被假定。此指令的某些有效变动为:

函数作图:	**Table** *表达式*, *x*
参数作图:	**Table** *x表达式*, *y表达式*, *t*
极坐标作图:	**Table** *表达式*, θ

备注: Table 命令对于 3D、序列或微分方程作图无效。作为替代，您可使用 **BldData**。

在函数作图模式下.

Table 1.25x*cos(x) ENTER

x	1		
0.	0.		
1.	.67538		
2.	-1.04		
3.	-3.712		
4.	-3.268		

Table cos(time),time ENTER

x	1	2	3
0.	0.	1.	
1.	.67538	.5403	
2.	-1.04	-.4161	
3.	-3.712	-.99	
4.	-3.268	-.6536	

tan()

TI-89: 2nd [TAN] 键 **TI-92 Plus:** TAN 键

tan(*表达式1***)** ⇒ *表达式*
tan(*数组1***)** ⇒ *数组*

tan(*表达式1***)** 对应作为表达式的变数正切。

tan(*数组1***)** 对应*数组1*中所有元素的正切数组。

备注: 变数会按当前的角模式作为度或弧度计算出。您可用 ° 或 ʳ 临时覆盖角模式。

在 Degree 角模式下:

tan((π/4)ʳ) ENTER 1

tan(45) ENTER 1

tan({0,60,90}) ENTER
{0 √3 undef}

在 Radian 角模式下:

tan(π/4) ENTER 1

tan(45°) ENTER 1

tan({π,π/3,⁻π,π/4}) ENTER
{0 √3 0 1}

---

tan(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵正切。这与计算每一元素的正切*不*同。有关计算方式的信息，请参阅 **cos()**。

平方矩阵1 必须是可对角的。结果一直含有浮点数字。

在 Radian 角模式下:

tan([1,5,3;4,2,1;6,⁻2,1]) ENTER

$$\begin{bmatrix} -28.291\ldots & 26.088\ldots & 11.114\ldots \\ 12.117\ldots & -7.835\ldots & -5.481\ldots \\ 36.818\ldots & -32.806\ldots & -10.459\ldots \end{bmatrix}$$

$\tan^{-1}()$

TI-89: ◆ [TAN⁻¹] 键 **TI-92 Plus:** 2nd [TAN⁻¹] 键

tan⁻¹(*表达式1***)** ⇒ *表达式*
tan⁻¹(*数组1***)** ⇒ *数组*

tan⁻¹(*表达式1***)** 对应正切是作为表达式的*表达式1*的角。

tan⁻¹(*数组1***)** 是对应*数组1*中每一元素的反正切数组。

备注: 按照当前角模式的设置，结果会对应角度或弧度。

在 Degree 角模式下:

tan⁻¹(1) ENTER 45

在 Radian 角模式下:

tan⁻¹({0,.2,.5}) ENTER
{0 .197… .463…}

---

tan⁻¹(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵反正切。这与计算每一元素的反正切*不*同。有关计算方式的信息，请参阅 **cos()**。

*平方矩阵1*必须是可对角的。结果一直含有浮点数字。

在 Radian 角模式下:

tan⁻¹([1,5,3;4,2,1;6,⁻2,1]) ENTER

$$\begin{bmatrix} -.083\ldots & 1.266\ldots & .622\ldots \\ .748\ldots & .630\ldots & -.070\ldots \\ 1.686\ldots & -1.182\ldots & .455\ldots \end{bmatrix}$$

tanh()

MATH/Hyperbolic 菜单

tanh(*表达式1***)** ⇒ *表达式*
tanh(*数组1***)** ⇒ *数组*

tanh(*表达式1***)** 是对应作为表达式的变数的双曲正切。

tanh(*数组***)** 对应*数组1*中的所有元素的双曲正切数组。

tanh(1.2) ENTER .833…

tanh({0,1}) ENTER {0 tanh(1)}

tanh(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵双曲正切。这与计算每一元素的双曲正切*不*同。有关计算方式的信息，请参阅 **cos()**。

平方矩阵1 必须是可对角的。结果一直含有浮点数字。

在 Radian 角模式下:

```
tanh([1,5,3;4,2,1;6,⁻2,1])
ENTER
```

$$\begin{bmatrix} -.097\ldots & .933\ldots & .425\ldots \\ .488\ldots & .538\ldots & -.129\ldots \\ 1.282\ldots & -1.034\ldots & .428\ldots \end{bmatrix}$$

tanh^{-1}() MATH/Hyperbolic 菜单

tanh^{-1}(*表达式1***)** ⇒ *表达式*
tanh^{-1}(*数组1***)** ⇒ *数组*

tanh^{-1}(*表达式1***)** 对应作为表达式的变数的双曲反正切。

tanh^{-1}(*数组1***)** 对应*数组1*中所有元素的双曲反正切数组。

在直角复数格式模式下:

```
tanh⁻¹(0) ENTER                    0
tanh⁻¹({1,2.1,3}) ENTER
```

$$\{\infty \quad .518\ldots-1.570\ldots\cdot i \quad \frac{\ln(2)}{2}-\frac{\pi}{2}\cdot i\}$$

tanh^{-1}(*平方矩阵1***)** ⇒ *平方矩阵*

对应*平方矩阵1*的矩阵双曲反正切。这与计算每一元素的双曲反正切*不*同。有关计算方式的信息，请参阅 **cos()**。

平方矩阵1 必须是可对角的。结果一直含有浮点数字。

在 Radian 角模式和 Rectangular 复数格式模式下:

```
tanh⁻¹([1,5,3;4,2,1;6,⁻2,1])
ENTER
```

$$\begin{bmatrix} -.099\ldots+.164\ldots\cdot i & .267\ldots-1.490\ldots\cdot i & \ldots \\ -.087\ldots-.725\ldots\cdot i & .479\ldots-.947\ldots\cdot i & \ldots \\ .511\ldots-2.083\ldots\cdot i & -.878\ldots+1.790\ldots\cdot i & \ldots \end{bmatrix}$$

taylor() MATH/Calculus 菜单

taylor(*表达式1***,** *变元***,** *顺序***[,** *点***]** ⇒ *表达式*

对应请求的泰勒多项式。多项式包括从零到*顺序*的整数度的非零项，以*变元*减号*点*表示。如此顺序没有缩短的幂系列，或如要求负数或分数指数时，**taylor()** 则对应自己。使用替代和 / 或用(*变元*减号*点*)的幂临时乘法来定义更多普通幂系列。

*点*会默认成零，也是指数点。

```
taylor(e^(√(x)),x,2) ENTER
taylor(e^(t),t,4)|t=√(x) ENTER
```

$$\begin{aligned} &\blacksquare\ \mathrm{taylor}\left(e^{\sqrt{x}},x,2\right) \\ &\qquad \mathrm{taylor}\left(e^{\sqrt{x}},x,2,0\right) \\ &\blacksquare\ \mathrm{taylor}\left(e^{t},t,4\right)\Big|\,t=\sqrt{x} \\ &\qquad \frac{x^2}{24}+\frac{x^{3/2}}{6}+\frac{x}{2}+\sqrt{x}+1 \end{aligned}$$

```
taylor(1/(x*(x-1)),x,3) ENTER
```

$$\begin{aligned} &\blacksquare\ \mathrm{taylor}\left(\frac{1}{x\cdot(x-1)},x,3\right) \\ &\qquad \mathrm{taylor}\left(\frac{1}{x\cdot(x-1)},x,3,0\right) \end{aligned}$$

```
expand(taylor(x/(x*(x-1)),
x,4)/x,x) ENTER
```

$$\begin{aligned} &\blacksquare\ \mathrm{expand}\left(\frac{\mathrm{taylor}\left(\frac{x}{x\cdot(x-1)},x\right.}{x}\right. \\ &\qquad -x^3-x^2-x-\frac{1}{x}-1 \end{aligned}$$

tCollect() MATH\Algebra\Trig 菜单

tCollect(*表达式1***)** ⇒ *表达式*

对应一个表达式，其中正弦和余弦的积及整数幂已转换成倍数角、角和以及角差的正弦和余弦线性组合。转换是将三角多项式换算成它们的调和函数的线性组合。

当默认的三角函数化简不能进行时，某些情形下，**tCollect()** 将可满足您的需求。**tCollect()** 倾向于将 **tExpand()** 进行的转换颠倒。有时，**tExpand()** 将应用到 **tCollect()** 的结果，或者相反，通过单独的两个步骤化简表达式。

```
tCollect((cos(α))^2) ENTER
```

$$\frac{\cos(2\cdot\alpha)+1}{2}$$

```
tCollect(sin(α)cos(β)) ENTER
```

$$\frac{\sin(\alpha-\beta)+\sin(\alpha+\beta)}{2}$$

tExpand() MATH\Algebra\Trig 菜单

tExpand(*表达式1***)** ⇒ *表达式*

对应一个表达式，其中整数倍数角、角和以及角差的正弦和余弦都已被扩展了。由于识别 $(\sin(x))^2+(\cos(x))^2=1$ 的原因，会有很多可能等同的结果。因此，结果也有可能与其它出版物中示出的不同。

当默认的三角函数化简不能进行时，某些情形下，**tCollect()** 将可满足您的需求。**tCollect()** 倾向于将 **tExpand()** 进行的转换颠倒。有时，**tExpand()** 将应用到 **tCollect()** 的结果，或者相反，通过单独的两个步骤化简表达式。

备注: 在 Degree 模式下的显示格式会影响 **tExpand()** 识别待扩展形式的能力。为了获得最理想的结果，**tExpand()** 要用在 Radian 模式下。

```
tExpand(sin(3ϕ)) ENTER
```

$$4\cdot\sin(\phi)\cdot(\cos(\phi))^2-\sin(\phi)$$

```
tExpand(cos(α-β)) ENTER
```

$$\cos(\alpha)\cdot\cos(\beta)+\sin(\alpha)\cdot\sin(\beta)$$

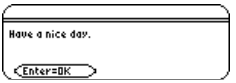
Text CATALOG

Text *提示字符串*

显示*提示字符串*对话框。

如作为 **Dialog...EndDlog block** 的一部分使用，*提示字符串* 会显示在对话框内。如作为单独的指令使用，**Text** 是创建显示字符串的对话框。

```
Text "Have a nice day." ENTER
                                Done
```



Then

请参阅第 138 页的 **If**。

Title CATALOG

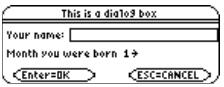
Title *标题字符串*, [*Lbl*]

当使用在 **Toolbar** 或 **Custom** 指令或 **Dialog...EndDlog** 块内时，是创建下拉菜单或对话框的标题。

备注: *Lbl* 只在 **Toolbar** 指令中有效。当起存在时，允许菜单选择程序内的一个指定标签的分支。

程序段:

```
  ⋮
:Dialog
:Title    "This is a dialog
box"
:Request  "Your name",Str1
:Dropdown "Month you were
born",
  seq(string(i),i,1,12),Var1
:EndDlog
  ⋮
```



tmpCnv() CATALOG

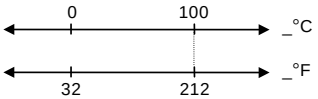
tmpCnv(*表达式1_°温度单位1*, *_°温度单位2***)**
⇒ *表达式_°温度单位2*

将*表达式1*指定的温度值从一种单位换算成另一种。有效的温度单位是:

_°C	摄氏
_°F	华氏
_°K	开氏
_°R	兰金

└─ 对于 °，可按下 [2nd] [°]。
TI-89: 对于 _，可按下 [◆] [_]。
TI-92 Plus: 对于 _，可按下 [2nd] [_]。

例如，100_°C 换算成 212_°F:



欲转换温度范围，可使用 **ΔtmpCnv()** 替代。

```
tmpCnv(100_°c,_°f) ENTER 212.•_°F

tmpCnv(32_°f,_°c) ENTER    0.•_°C

tmpCnv(0_°c,_°k) ENTER 273.15•_°K

tmpCnv(0_°f,_°r) ENTER 459.67•_°R
```

备注: 欲在菜单选一种温度单位，可按下:

TI-89: [2nd] [UNITS]
TI-92 Plus: [◆] [UNITS]

ΔtmpCnv() CATALOG

ΔtmpCnv(*表达式1_°温度单位1*, *_°温度单位2***)**
⇒ *表达式_°温度单位2*

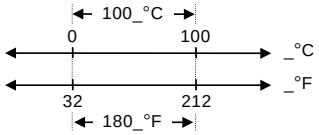
将*表达式1*指定的温度范围(两个温度值之间的差别)从一种单位换算成另一种。有效的温度单位是:

_°C	摄氏
_°F	华氏
_°K	开氏
_°R	兰金

└─对于 °，可按下 [2nd] [°]。
TI-89: 对于 _，可按下 [◆] [_]。
TI-92 Plus: 对于 _，可按下 [2nd] [_]。

1_°C 和 1_°K 具有相同的量值，1_°F 和 1_°R 也一样。然而，1_°C 是 1_°F 的大小的 9/5。

例如，100_°C 范围(从 0_°C 到 100_°C)与 180_°F 范围等同:



欲转换一个具体的温度值而不是范围，可使用 **tmpCnv()** 来替代。

欲获得 Δ，您可按下 [◆] [(] [↑] [D] (或 [2nd] [CHAR] 1 5)。

```
ΔtmpCnv(100_°c,_°f) ENTER
                    180.·_°F

ΔtmpCnv(180_°f,_°c) ENTER
                    100.·_°C

ΔtmpCnv(100_°c,_°k) ENTER
                    100.·_°K

ΔtmpCnv(100_°f,_°r) ENTER
                    100.·_°R

ΔtmpCnv(1_°c,_°f) ENTER
                    1.8·_°F
```

备注: 欲在菜单选一种温度单位，可按下:

TI-89: [2nd] [UNITS]
TI-92 Plus: [◆] [UNITS]

Toolbar CATALOG

Toolbar
块
EndTBar

创建工具栏菜单。

*块*可为单一陈述或是由字符“:”分隔的陈述序列。陈述可为 Title 或 Item。

Items 必须带有标签。如没有项目的话，Title 也应该有标签。

程序段:

```
  ⋮
:Toolbar
:  Title "Examples"
:  Item "Trig", t
:  Item "Calc", c
:  Item "Stop", Pexit
:EndTbar
  ⋮
```

备注: 在程序中运行时，此段是创建分支到程序中三个位置的三个选择的菜单。

Trace　CATALOG

Trace

画一个 Smart Graph 并将追踪光标放在上次定义的光标位置中第一个定义的 Y=函数上，或如需要重作图形时，放在重设的位置上。

当编辑坐标值时，允许光标和大多数按键的操作。系列按键如功能键、[APPS] 和 [MODE] 在追踪期间是不活动的。

备注: 按下 [ENTER] 恢复操作。

Try　CATALOG

Try
　块1
Else
　块2
EndTry

执行*块1*，除非出现错误。如*块1*中出现错误，程序的执行就会转移到*块2*。变元 errornum 含有错误号码，以允许程序执行错误恢复。

*块1*和*块2*可为单一陈述或是由字符“:”分隔的陈述系列。

程序段:

```
   ⋮
:Try
:  NewFold(temp)
:  Else
:  ◉Already exists
:  ClrErr
:EndTry
   ⋮
```

备注: 请参阅 **ClrErr** 和 **PassErr**。

TwoVar　MATH/Statistics 菜单

TwoVar *数组1*, *数组2*[, [*数组3*] [, *数组4*, *数组5*]]

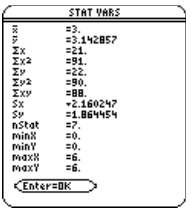
计算 **TwoVar** 统计并更新所有系统统计变元。

所有数组必须有等同维数，*数组5*除外。

数组1 代表 x 数组。
数组2 代表 y 数组。
数组3 代表频数。
数组4 代表类别代码。
数组5 代表类别包含数组。

备注: *数组1*到*数组4*必须为变元名称或 c1-c99 (数据 / 矩阵编辑器显示的上一数据变元的列)。*数组5*不必为变元名称并不能为 c1-c99。

```
{0,1,2,3,4,5,6}→L1 [ENTER]
                          {0 1 2 ...}
{0,2,3,4,3,4,6}→L2 [ENTER]
                          {0 2 3 ...}
TwoVar L1,L2 [ENTER]            Done
ShowStat [ENTER]
```



Unarchiv CATALOG

Unarchiv *变元1* [, *变元2*] [, *变元3*] …

将指定的变元从用户数据储存内存移动到 RAM。

您可象存取 RAM 中的变元一样存取已存档的变元。尽管如此，您不能删除、重命名或储存一个已存档的变元，这是因为它是自动锁定的。

欲将变元存档，可使用 **Archive**。

```
10→arctest ENTER                    10
Archive arctest ENTER             Done
5* arctest ENTER                    50
15→arctest ENTER
```

```
ESC
Unarchiv arctest ENTER            Done
15→arctest ENTER                    15
```

unitV() MATH/Matrix/Vector ops 菜单

unitV(*向量1***)** ⇒ *向量*

对应行或列单位向量，取决于*向量1*的形式。

*向量1*必须是单行矩阵或单列矩阵。

unitV([a,b,c]) ENTER

$$\left[\frac{a}{\sqrt{a^2+b^2+c^2}}\quad\frac{b}{\sqrt{a^2+b^2+c^2}}\quad\frac{c}{\sqrt{a^2+b^2+c^2}}\right.$$

unitV([1,2,1]) ENTER

$$\left[\frac{\sqrt{6}}{6}\quad\frac{\sqrt{6}}{3}\quad\frac{\sqrt{6}}{6}\right]$$

unitV([1;2;3]) ENTER

$$\begin{bmatrix}\frac{\sqrt{14}}{14}\\ \frac{\sqrt{14}}{7}\\ \frac{3\cdot\sqrt{14}}{14}\end{bmatrix}$$

Unlock CATALOG

Unlock *变元1*[, *变元2*][, *变元3*]...

将指定的变元松锁。

备注: 变元可通过 **Lock** 命令锁定。

variance() MATH/Statistics 菜单

variance(*数组***[**, *频数数组***])** ⇒ *表达式*

对应*数组* 的方差。

每一*频数数组*元素是计算*数组*相应元素连续出现的次数。

备注:*数组* 必须含有最少两个元素。

variance({a,b,c}) ENTER

$$\frac{a^2-a\cdot(b+c)+b^2-b\cdot c+c^2}{3}$$

variance({1,2,5,⁻6,3,⁻2}) ENTER

31/2

variance({1,3,5},{4,6,2}) ENTER

68/33

---

variance(*矩阵1***[**, *频数矩阵***])** ⇒ *矩阵*

对应含有*矩阵1*中每一列的方差的行向量。

每一*频数矩阵*元素是计算*矩阵1*相应元素连续出现的次数。

备注: *矩阵1*必须含有最少两个行。

variance([1,2,5;⁻3,0,1;
.5,.7,3]) ENTER [4.75 1.03 4]

variance([-1.1,2.2;3.4,5.1;
-2.3,4.3],[6,3;2,4;5,1]) ENTER
[3.91731,2.08411]

when() CATALOG

when(*条件*, *真实结果* [, *虚假结果*] [, *未知结果*]**)** ⇒ *表达式*

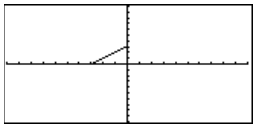
对应*真实结果*、*虚假结果*或*未知结果*，取决于 *条件*是真实、虚假的或是未知的。如指定合适结果的变数太少，是对应输入。

省略*虚假结果*和*未知结果*来形成一条表达式只定义*条件*是真实的范围。

```
when(x<0,x+3)|x=5 ENTER
                  when(x<0,3+x)
```

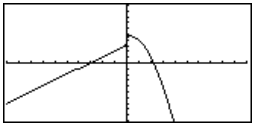
使用 undef *虚假结果*来定义只在一个区间作图的表达式。

```
ClrGraph ENTER
Graph when(x≥⁻π and
x<0,x+3,undef) ENTER
```



仅省略*未知结果*来定义两段表达式。

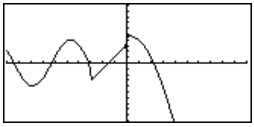
```
Graph when(x<0,x+3,5-x^2) ENTER
```



套入 **when()** 来定义超过两段的表达式。

TI-89: HOME
TI-92 Plus: ♦ [HOME]

```
ClrGraph ENTER                Done
Graph when(x<0,when(x<⁻π,
4*sin(x),2x+3),5-x^2) ENTER
```



when() 对于定义递归函数是有帮助的。

```
when(n>0,n*factoral(n-1),1)
→factoral(n) ENTER            Done
factoral(3) ENTER                6
3! ENTER                         6
```

While CATALOG

While *条件*
块
EndWhile

执行*块*中的陈述，*条件*就是真实的。

*块*可为单一陈述，或是由字符“:”分隔的陈述系列。

程序段:

```
  ⋮
:1→i
:0→temp
:While i<=20
:  temp+1/i→temp
:  i+1→i
:EndWhile
:Disp "sum of reciprocals up
to                    20",temp
  ⋮
```

"With" 请参阅第 217 页中的 |。

xor MATH/Test 菜单

布尔表达式1 **xor** *布尔表达式2* ⇒ *布尔表达式*

如*布尔表达式1*是 **true** 及*布尔表达式2*是假的话，则对应 true，反之也成立。
如*布尔表达式1*和*布尔表达式2*均为真的或均为假的话则对应 false。如原始布尔表达式不能解成真实或虚假的，对应化简的布尔表达式。

备注: 请参阅 **or**。

```
true xor true ENTER          false
(5>3) xor (3>5) ENTER         true
```

整数1 **xor** *整数2* ⇒ *整数*

利用 **xor** 运算逐位比较两个实数整数。在内部，两个整数会被转换成标记的 32-位十进制数字。当对应的位得到比较后，如位是 1，那么结果就是 1；结果只有在两个位均是 0 时才是 0。对应的值表示位结果，并按照 Base 模式显示出来。

您可一任何数字基输入整数。对于十进制或十六进制输入项，您必须相应使用 0b 或 0h 前缀。不带前缀的话，整数会被当作小数(基 10)处理。

如您输入了对于标记的 32-位十进制形式过大的小数整数，就会采用对称模运算来将数值带入近似的范围内。

备注: 请参阅 **or**。

在十六进制基模式下:

```
0h7AC36 xor 0h3D5F ENTER 0h79169
```

重要事项: 是零，非字母 O。

在十进制基模式下:

```
0b100101 xor 0b100 ENTER0b100001
```

备注: 一个十进制输入项可有多达 32 位 (0b 前缀不计算在内)；一个十六进制输入项可有 8 位。

XorPic CATALOG

XorPic *pic变元*[, *行*] [, *列*]

在当前作图屏上显示储存于*图形变元*的图象。

对每一象素使用 **xor** 逻辑。只有那些屏幕或图象专用的象素位置会被打开。此指令可将两图象中打开的象素关闭。

图形变元 必须含有图象数据类型。

如含有*行*和*列*，是指定图象左上角的象素坐标。默认值是(0, 0)。

zeros() MATH/Algebra 菜单

zeros(*表达式***,** *变元***)** ⇒ *数组*

对应使*表达式*=0 的*变元*的候选实数值数组。**zeros()** 是通过计算 **exp▸list(solve(***表达式***=0,***变元***),***变元***)** 进行的。

```
zeros(a*x^2+b*x+c,x) ENTER
```

$$\left\{\frac{-\left(\sqrt{b^2-4\cdot a\cdot c}+b\right)}{2\cdot a}\ \frac{\sqrt{b^2-4\cdot a\cdot c}-b}{2\cdot a}\right\}$$

```
a*x^2+b*x+c|x=ans(1)[2] ENTER   0
```

处于某些目的，**zeros()** 的结果形式较 **solve()** 更合算。尽管如此，**zeros()** 的结果仍不能表示含蓄解，该解要求是不等式或该解不包含*变元*。

```
exact(zeros(a*(e^(x)+x)
  (sign (x)-1),x)) ENTER        {}
```

备注: 也请参阅 **cSolve()**、**cZeros()** 和 **solve()**。

```
exact(solve(a*(e^(x)+x)
  (sign (x)-1)=0,x)) ENTER
```

$e^x + x = 0$ or $x>0$ or $a = 0$

zeros({*表达式1***, *表达式2***}, {***变元或猜测值1***,**
变元或猜测值2 [, ...]**})** ⇒ *矩阵*

对应联立代数*表达式*的候选实数零，其中每一*变元或猜测值*是指定您要找的一个未知整数。

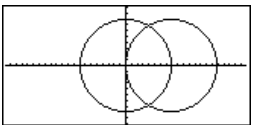
作为选项，您可为变元指定起始猜测值。每一*变元或猜测值*必须具有如下形式:

变元
– 或 –
变元 = 实数或非实数

例如，x 是有效的，所以 x=3。

如所有方程均为多项式及没指定一个起始的猜测值，**zeros()** 就会利用词汇的 Gröbner /Buchberger 消除方式来尝试定义所有的实 数零。

例如，假定您有一个位于坐标原点、半径为 r 的圆和另一个半径为 r，圆心位于第一个圆与正 x 轴相交处的圆。利用 **zeros()** 求相交点。



如右图例子中的 r 所示，联立*多项式*方程可有无值的其它变元，但它是代表稍后要被替换的给出数值。

zeros({x^2+y^2-r^2,
(x-r)^2+y^2-r^2},{x,y}) ENTER

$$\begin{bmatrix} \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} \\ \frac{r}{2} & \frac{-\sqrt{3}\cdot r}{2} \end{bmatrix}$$

结果矩阵的每行是表示一个替代的零，组成部分与*变元或猜测值*数组的顺序相同。欲提取一行，可用 [*行*] 来指示矩阵。

提取行 2:

ans(1)[2] ENTER

$$\begin{bmatrix} \frac{r}{2} & \frac{-\sqrt{3}\cdot r}{2} \end{bmatrix}$$

您也可(或不)包含方程中没有出现的未知数。例如，您可将作为未知数的 z 包含进去，以将前一例子扩展成具有两个平行交点的半径 r 柱面。柱面零示出多少零族含有形式 @*k* 的任意常数，其中 *k* 是一个从 1 到 255 的整数前缀。当您使用 **ClrHome** 或 F1 8:Clear Home 时，前缀会重设到 1。

zeros({x^2+y^2-r^2,
(x-r)^2+y^2-r^2},{x,y,z})
ENTER

$$\begin{bmatrix} \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} & @1 \\ \frac{r}{2} & \frac{-\sqrt{3}\cdot r}{2} & @1 \end{bmatrix}$$

对于多项式组，计算时间或内存耗费很大程度取决于您列出未知数的顺序。如您的起始选择耗用内存或花费您的耐性，可尝试将方程和 / 或*变元红猜测值*数组中的变元重新整理一下。

如您没有将任何猜测值包含进去，或如任何方程在何变元中是非多项式，但所有方程在未知数中均为一次的话，**zeros()** 会使用高斯消除方式来尝试定义所有实数零。

```
zeros({x+e^(z)*y-1,x-y-sin(z)}
,{x,y}) ENTER
```

$$\left[\frac{e^z\cdot\sin(z)+1}{e^z+1}\quad\frac{^-(\sin(z)-1)}{e^z+1}\right]$$

如方程组在其所有变元中既不是多项式，在未知数中也不是一次的话，**zeros()** 会使用近似重复方式定义最多一个零。为了这样做，未知数必须等于方程数，方程中的其它所有变元必须化简成数字。

```
zeros({e^(z)*y-1,-y-sin(z)},
{y,z}) ENTER
```

[.041… 3.183…]

如只有一个，那么每一未知数就会以其猜测的数值开始，否则就在 0.0 开始。

利用猜测值来逐个寻找其它的零。对于收敛，猜测值反而会包含在零内。

```
zeros({e^(z)*y-1,-y-sin(z)},
{y,z=2π}) ENTER
```

[.001… 6.281…]

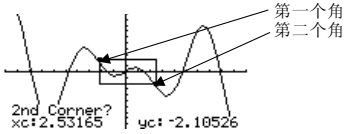
ZoomBox CATALOG

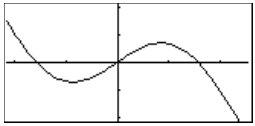
ZoomBox

显示作图 屏，让您画出一个定义新观察窗的框并更新窗口。

在函数作图模式下:

```
1.25x*cos(x)→y1(x) ENTER        Done
ZoomStd:ZoomBox ENTER
```





第二次按下ENTER 后定义 ZoomBox 的显示。

ZoomData CATALOG

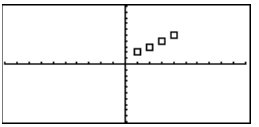
ZoomData

基于当前的图(和数据)调整窗口设置，使所有的统计数据点得到取样并显示作图屏。

备注: 对于直方图是不调整 ymin 和 ymax 的。

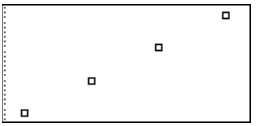
在函数作图模式下:

```
{1,2,3,4}→L1 ENTER        {1 2 3 4}
{2,3,4,5}→L2 ENTER        {2 3 4 5}
newPlot 1,1,L1,L2 ENTER        Done
ZoomStd ENTER
```



TI-89: HOME
TI-92 Plus: ◆ [HOME]

```
ZoomData ENTER
```



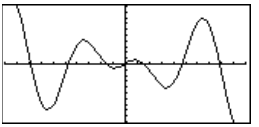
ZoomDec CATALOG

ZoomDec

调整观察窗，使 Δx 和 Δy = 0.1 并显示原点位于中央的作图屏。

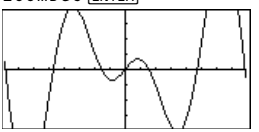
在函数作图模式下:

```
1.25x*cos(x)→y1(x) ENTER        Done
ZoomStd ENTER
```



TI-89: HOME
TI-92 Plus: ◆ [HOME]

```
ZoomDec ENTER
```



ZoomFit CATALOG

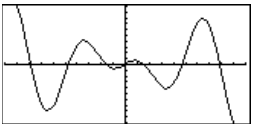
ZoomFit

显示作图屏并计算因变元所需的窗口维数，以查看当前自变元设置的所有图象。

在函数作图模式下:

1.25x*cos(x)→y1(x) [ENTER] Done

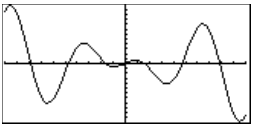
ZoomStd [ENTER]



TI-89: [HOME]

TI-92 Plus: [◆] [HOME]

ZoomFit [ENTER]



ZoomIn CATALOG

ZoomIn

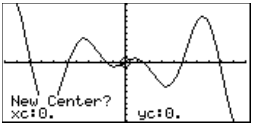
显示作图屏，让您设定放大的中心点并更新观察窗。

缩放的大小取决于 Zoom 因子 xFact 和 yFact。在 3D 作图模式下，大小是取决于 xFact、yFact 和 zFact。

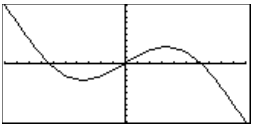
在函数作图模式下:

1.25x*cos(x)→y1(x) [ENTER] Done

ZoomStd:ZoomIn [ENTER]



[ENTER]



ZoomInt CATALOG

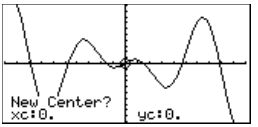
ZoomInt

显示作图屏，让您设定放大的中心点并调整窗口设置，使每一象素在所有方向为整数。

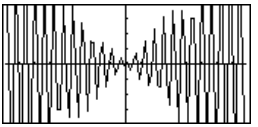
在函数作图模式下:

1.25x*cos(x)→y1(x) [ENTER] Done

ZoomStd:ZoomInt [ENTER]



[ENTER]



ZoomOut CATALOG

ZoomOut

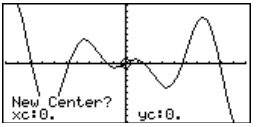
显示作图屏，让您设定缩小的中心点并更新观察窗。

缩放的大小取决于 Zoom 因子 xFact 和 yFact。在 3D 作图模式下，大小是取决于 xFact、yFact 和 zFact。

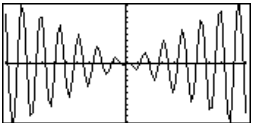
在函数作图模式下:

1.25x*cos(x)→y1(x) ENTER Done

ZoomStd:ZoomOut ENTER



ENTER



ZoomPrev CATALOG

ZoomPrev

显示作图屏并在上一缩放前更新使用中的观察窗设置。

ZoomRcl CATALOG

ZoomRcl

显示作图屏并利用 **ZoomSto** 指令储存的设置更新观察窗。

ZoomSqr CATALOG

ZoomSqr

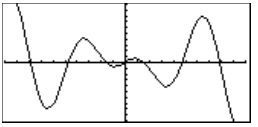
显示作图屏，调整 x 或 y 窗口设置，使每一象素的长和宽在坐标系中相等，并更新观察窗口。

在 3D Graph 模式下，**ZoomSqr** 会将两条最短的坐标轴延长成与两条最长的一样。

在函数作图模式下:

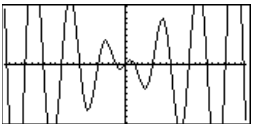
1.25x*cos(x)→y1(x) ENTER Done

ZoomStd ENTER



HOME

ZoomSqr ENTER



ZoomStd CATALOG

ZoomStd

将窗口变元设定成以下的标准值，然后更新观察窗。

函数作图:
x: [⁻10, 10, 1], y: [⁻10, 10, 1] 和 xres=2

参数作图:
t: [0, 2π, π/24], x: [⁻10, 10, 1], y:[⁻10, 10, 1]

极坐标作图:
θ: [0, 2π, π/24], x: [⁻10, 10, 1], y: [⁻10, 10, 1]

系列作图:
nmin=1, nmax=10, plotStrt=1, plotStep=1,
x: [⁻10, 10, 1], y: [⁻10, 10, 1]

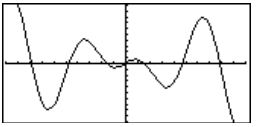
3D 作图:
eyeθ°=20, eyeφ°=70, eyeψ°=0
x: [⁻10, 10, 14], y: [⁻10, 10, 14],
z: [⁻10, 10], ncontour=5

微分方程作图:
t: [0, 10, .1, 0], x: [⁻1, 10, 1], y: [⁻10, 10, 1],
ncurves=0, Estep=1, diftol=.001, fldres=14,
dtime=0

在函数作图模式下:

1.25x∗cos(x)→y1(x) [ENTER] Done

ZoomStd [ENTER]



ZoomSto CATALOG

ZoomSto

将当前的窗口设置储存在 Zoom 内存中。您可使用 **ZoomRcl** 重新储存设置。

ZoomTrig CATALOG

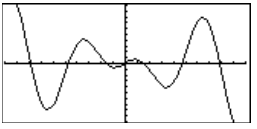
ZoomTrig

显示作图屏，将 Δx 设定成 π/24 和将 xscl 设定成 π/2，把原点对中，设定 y 设置值为 [⁻4, 4, .5] 并更新观察窗。

在函数作图模式下:

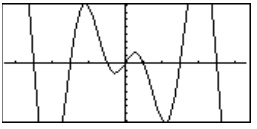
1.25x∗cos(x)→y1(x) [ENTER] Done

ZoomStd [ENTER]



TI-89: [HOME]
TI-92 Plus: [◆] [HOME]

ZoomTrig [ENTER]



+ (加)　　⊞ 键

表达式1 + *表达式2* ⇒ *表达式*

对应*表达式1*和*表达式2*之和。

```
56 ENTER                    56
ans(1)+4 ENTER              60
ans(1)+4 ENTER              64
ans(1)+4 ENTER              68
ans(1)+4 ENTER              72
```

数组1 + *数组2* ⇒ *数组*
矩阵1 + *矩阵2* ⇒ *矩阵*

对应含有*数组1*和*数组2* (或*矩阵1*和*矩阵2*)中相应元素的和的数组(或矩阵)。

变数的维数必须等同。

```
{22,π,π/2}→L1 ENTER        {22 π π/2}
{10,5,π/2}→L2 ENTER        {10 5 π/2}
L1+L2 ENTER                {32 π+5 π}
ans(1)+{π,⁻5,⁻π} ENTER
                           {π+32 π 0}
[a,b;c,d]+[1,0;0,1] ENTER
                           [a+1 b; c d+1]
```

表达式 + *数组1* ⇒ *数组*
数组1 + *表达式* ⇒ *数组*

对应*表达式*与*数组1*中每一元素的和的数组。

```
15+{10,15,20} ENTER        {25 30 35}
{10,15,20}+15 ENTER        {25 30 35}
```

表达式 + *矩阵1* ⇒ *矩阵*
矩阵1 + *表达式* ⇒ *矩阵*

对应*表达式*加到*矩阵1*对角线的每一元素的矩阵。*矩阵1*必须为平方。

备注: 使用 **.+** (点加)来将表达式加到每一元素中。

```
20+[1,2;3,4] ENTER
                           [21 2; 3 24]
```

− (减)　　⊟ 键

表达式1 - *表达式2* ⇒ *表达式*

对应*表达式1*减号*表达式2*。

```
6-2 ENTER                   4
π-π/6 ENTER                 5·π/6
```

数组1 - *数组2* ⇒ *数组*
矩阵1 - *矩阵2* ⇒ *矩阵*

从*数组1* (或*矩阵1*)相应元素减去*数组2* (或*矩阵2*)的每一元素，并对应结果。

变数的维数必须等同。

```
{22,π,π/2}-{10,5,π/2} ENTER
                           {12 π-5 0}
[3,4]-[1,2] ENTER           [2 2]
```

表达式 - *数组1* ⇒ *数组*
数组1 - *表达式* ⇒ *数组*

从*表达式*中减去每一*数组1*元素，或从每一*数组1*元素中减去*表达式*，并对应结果数组。

```
15-{10,15,20} ENTER        {5 0 -5}
{10,15,20}-15 ENTER        {-5 0 5}
```

表达式 - *矩阵1* ⇒ *矩阵*
矩阵1 - *表达式* ⇒ *矩阵*

表达式 - *矩阵1*是对应*表达式*乘识别矩阵减*矩阵1*的矩阵。*矩阵1*必须是平方。

矩阵1 - *表达式*是对应*表达式*乘以*矩阵1*中减出来的识别矩阵的矩阵。*矩阵1*必须是平方。

备注: 请使用 **.-** (点减)来从每一元素中减去表达式。

20-[1,2;3,4] ENTER

$$\begin{bmatrix} 19 & -2 \\ -3 & 16 \end{bmatrix}$$

* (乘) ⊠ 键

表达式1 * *表达式2* ⇒ *表达式*

对应*表达式1*与*表达式2*的积。

2*3.45 ENTER 6.9

x*y*x ENTER $x^2 \cdot y$

数组1 * *数组2* ⇒ *数组*

是对应含有*数组1*和*数组2*之中相应元素的积的数组。

数组中的维数必须等同。

{1.0,2,3}*{4,5,6} ENTER {4. 10 18}

{2/a,3/2}*{a²,b/3} ENTER $\{2 \cdot a \quad \frac{b}{2}\}$

矩阵1 * *矩阵2* ⇒ *矩阵*

对应*矩阵1*和*矩阵2*的矩阵积。

*矩阵1*中的行数必须与*矩阵2*中的列数相等。

[1,2,3;4,5,6]*[a,d;b,e;c,f] ENTER

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}$$

$$\begin{bmatrix} a+2\cdot b+3\cdot c & d+2\cdot e+3 \\ 4\cdot a+5\cdot b+6\cdot c & 4\cdot d+5\cdot e \end{bmatrix}$$

表达式 * *数组1* ⇒ *数组*
数组1 * *表达式* ⇒ *数组*

对应含有*表达式*与*数组1*之中每一元素的积的数组。

π*{4,5,6} ENTER {4·π 5·π 6·π}

表达式 * *矩阵1* ⇒ *矩阵*
矩阵1 * *表达式* ⇒ *矩阵*

对应含有*表达式*与*矩阵1*之中每一元素的积的矩阵。

备注: 请使用 **.*** (点乘)来将每一元素乘以表达式。

[1,2;3,4]*.01 ENTER

$$\begin{bmatrix} .01 & .02 \\ .03 & .04 \end{bmatrix}$$

λ*identity(3) ENTER

$$\begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}$$

/ (除) ÷ 键

表达式1 / *表达式2* ⇒ *表达式*

对应*表达式1*除以*表达式2*的商。

2/3.45 ENTER .57971

x^3/x ENTER x^2

数组1 / *数组2* ⇒ *数组*

对应含有*数组1*除以*数组2*的商的数组。

数组中的维数必须等同。

{1.0,2,3}/{4,5,6} ENTER

{.25 2/5 1/2}

表达式 / 数组1 ⇒ *数组*
数组1 / 表达式 ⇒ *数组*

对应含有*表达式*除以*数组1*或*数组1*除以*表达式*的商的数组。

a/{3,a,√(a)} ENTER

$$\left\{\frac{a}{3}\ 1\ \sqrt{a}\right\}$$

{a,b,c}/(a∗b∗c) ENTER

$$\left\{\frac{1}{b\cdot c}\ \frac{1}{a\cdot c}\ \frac{1}{a\cdot b}\right\}$$

矩阵1 / 表达式 ⇒ *矩阵*

对应含有*矩阵1 / 表达式*的商的矩阵。

备注: 使用 **.*/*** (点除)使表达式除以每一元素。

[a,b,c]/(a∗b∗c) ENTER

$$\left[\frac{1}{b\cdot c}\ \frac{1}{a\cdot c}\ \frac{1}{a\cdot b}\right]$$

⁻ (负) (-) 键 and MATH/Base 菜单

⁻*表达式1* ⇒ *表达式*
⁻*数组1* ⇒ *数组*
⁻*矩阵1* ⇒ *矩阵*

对应变数的负号。

对于数组或矩阵，对应所有负的元素。

如*表达式1*是十进制或十六进制整数，负号是给二的余数。

⁻2.43 ENTER ⁻2.43

⁻{⁻1,0.4,1.2E19} ENTER
{1 ⁻.4 ⁻1.2E19}

⁻a∗⁻b ENTER a·b

在十进制基模式下:

0b100101▸dec ENTER 37
└─ **重要事项:** 零，非字母 O。

⁻0b100101 ENTER
0b11111111111111111111111111011011

ans(1)▸dec ENTER ⁻37

备注: 欲键入 ▸，可按下 2nd [▸]。

% (百分数) CHAR/Punctuation 菜单

表达式1 % ⇒ *表达式*
数组1 % ⇒ *数组*
矩阵1 % ⇒ *矩阵*

对应 $\frac{变数}{100}$。

对于数组或矩阵，是对应每一元素除以 100。

13% ◆ ENTER .13

{1, 10, 100}% ◆ ENTER
{.01 .1 1.}

= (等于)　　[=] 键

表达式1 = *表达式2* ⇒ *布尔表达式*
数组1 = *数组2* ⇒ *布尔数组*
矩阵1 = *矩阵2* ⇒ *布尔矩阵*

如*表达式1*是定义成要等于*表达式2*，是对应 true。

如*表达式1*是定义成不等于*表达式2*，是对应 false。

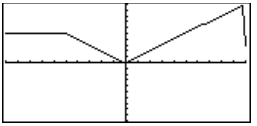
任何其它的是对应方程化简的形式。

对于数组和矩阵，对应元素与元素的比较。

例子函数表使用匹配的测试符号: =, ≠, <, ≤, >, ≥

```
:g(x)
:Func
:If x≤⁻5 Then
:  Return 5
:  ElseIf x>⁻5 and x<0 Then
:  Return ⁻x
:  ElseIf x≥0 and x≠10 Then
:  Return x
:  ElseIf x=10 Then
:  Return 3
:EndIf
:EndFunc
```

Graph g(x) [ENTER]



≠　　[◆] [=] 键

表达式1 ≠ *表达式2* ⇒ *布尔表达式*
数组1 ≠ *数组2* ⇒ *布尔数组*
矩阵1 ≠ *矩阵2* ⇒ *布尔矩阵*

如*表达式1*是定义成不等于*表达式2*的话，是对应 true。

如*表达式1*是定义成要等于*表达式2*的话，是对应 false。

任何其它的是对应方程化简的形式。

对于数组和矩阵，对应元素与元素的比较。

请参阅 "=" (等于)的例子。

<　　[2nd] [<] 键

表达式1 < *表达式2* ⇒ *布尔表达式*
数组1 < *数组2* ⇒ *布尔数组*
矩阵1 < *矩阵2* ⇒ *布尔矩阵*

如*表达式1*是定义成要小于*表达式2*的话，是对应 true。

如*表达式1*是定义成大于或等于*表达式2*，是对应 false。

任何其它的是对应方程化简的形式。

对于数组和矩阵，对应元素与元素的比较。

请参阅 "=" (等于)的例子。

≤ [◆][0] 键

表达式1 ≤ *表达式2* ⇒ *布尔表达式*
数组1 ≤ *数组2* ⇒ *布尔数组*
矩阵1 ≤ *矩阵2* ⇒ *布尔矩阵*

如*表达式1*是定义成要小于或等于*表达式2*，是对应 true。

如*表达式1*是定义成要大于*表达式2*的话，是对应 false。

任何其它的是对应方程化简的形式。

对于数组和矩阵，对应元素与元素的比较。

请参阅 "=" (等于)的例子。

> [2nd][>] 键

表达式1 > *表达式2* ⇒ *布尔表达式*
数组1 > *数组2* ⇒ *布尔数组*
矩阵1 > *矩阵2* ⇒ *布尔矩阵*

如*表达式1*是定义成要大于*表达式2*，是对应 true。

如*表达式1*是定义成要小于或等于*表达式2*，是对应 false。

任何其它的是对应方程化简的形式。

对于数组和矩阵，对应元素与元素的比较。

请参阅 "=" (等于)的例子。

≥ [◆][.] 键

表达式1 ≥ *表达式2* ⇒ *布尔表达式*
数组1 ≥ *数组2* ⇒ *布尔数组*
矩阵1 ≥ *矩阵2* ⇒ *布尔矩阵*

如*表达式1*是定义成要大于或等于*表达式2*，是对应 true。

如*表达式1*是定义成要小于*表达式2*的话，是对应 false。

任何其它的是对应方程化简的形式。

对于数组和矩阵，对应元素与元素的比较。

请参阅 "=" (等于)的例子。

.+ (点加) [.][+] 键

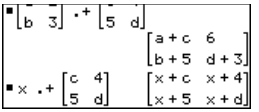
矩阵1 .+ *矩阵2* ⇒ *矩阵*
表达式 .+ *矩阵1* ⇒ *矩阵*

矩阵1 .+ *矩阵2*对应*矩阵1*和*矩阵2*中每一相应元素对的和的矩阵。

表达式 .+ *矩阵1*对应*表达式*与*矩阵1*中每一元素的和的矩阵。

[a,2;b,3].+[c,4;5,d] [ENTER]

x.+[c,4;5,d] [ENTER]



.- (点减) [.][−] 键

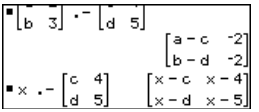
矩阵1 .- *矩阵2* ⇒ *矩阵*
表达式 .- *矩阵1* ⇒ *矩阵*

矩阵1 .- *矩阵2*对应*矩阵1*和*矩阵2*中每一相应元素对之间的差的矩阵。

表达式 .- *矩阵1*对应*表达式*与*矩阵1*中每一元素的差的矩阵。

[a,2;b,3].-[c,4;d,5] [ENTER]

x.-[c,4;d,5] [ENTER]



.* (点乘)　[.] [×] 键

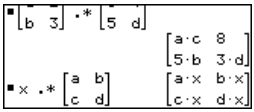
矩阵1 .* *矩阵2* ⇒ *矩阵*
表达式 .* *矩阵1* ⇒ *矩阵*

矩阵1 .* *矩阵2* 对应*矩阵1*和*矩阵2*中每一相应元素对的积的矩阵。

表达式 .* *矩阵1*对应含有*表达式*与*矩阵1*中每一元素的积的矩阵。

[a,2;b,3].*[c,4;5,d] [ENTER]

x.*[a,b;c,d] [ENTER]



./ (点除)　[.] [÷] 键

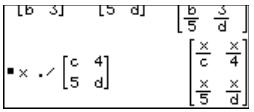
矩阵1 ./ *矩阵2* ⇒ *矩阵*
表达式 ./ *矩阵1* ⇒ *矩阵*

矩阵1 ./ *矩阵2*对应*矩阵1*和*矩阵2*中每一相应元素对的商的矩阵。

表达式 ./ *矩阵1*对应*表达式*与*矩阵1*中每一元素的商的矩阵。

[a,2;b,3]./[c,4;5,d] [ENTER]

x./[c,4;5,d] [ENTER]



.^ (点幂)　[.] [^] 键

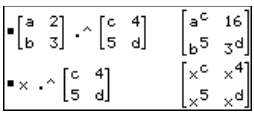
矩阵1 .^ *矩阵2* ⇒ *矩阵*
表达式 .^ *矩阵1* ⇒ *矩阵*

矩阵1 .^ *矩阵2*对应*矩阵2*中每一元素是*矩阵1*中相应元素的指数的矩阵。

表达式 .^ *矩阵1*对应*矩阵1*中每一元素是*表达式*的指数的矩阵。

[a,2;b,3].^[c,4;5,d] [ENTER]

x.^[c,4;5,d] [ENTER]



! (阶乘)　TI-89: [◆] [÷] 键　TI-92 Plus: [2nd] W 键

表达式1! ⇒ *表达式*
数组1! ⇒ *数组*
矩阵1! ⇒ *矩阵*

对应变数的阶乘。

对于数组或矩阵，是对应元素的阶乘数组或矩阵。

TI-89 只为非负数整数值计算数值。

5! [ENTER]　120

{5,4,3}! [ENTER]　{120　24　6}

[1,2;3,4]! [ENTER]　$\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

& (附加)　TI-89: [◆] [×] 键　TI-92 Plus: [2nd] H 键

字符串1 & *字符串2* ⇒ *字符串*

对应*字符串2*附加到*字符串1*的文本字符串。

"Hello " & "Nick" [ENTER]

"Hello Nick"

∫() (积分)　[2nd] [∫] 键

∫(*表达式1*, *变元*[, *下*] [,*上*]) ⇒ *表达式*
∫(*数组1*, *变元* [,*顺序*]) ⇒ *数组*
∫(*矩阵1*, *变元* [,*顺序*]) ⇒ *矩阵*

对应*表达式1*的积分，相对于*变元*从*下*到*上*。

∫(x^2,x,a,b) [ENTER]　$\frac{b^3}{3} - \frac{a^3}{3}$

如省略了 *下*和 *上*，是对应一个反导数。积分的符号常数如 C 被省略。

∫(x^2,x) ENTER $\frac{x^3}{3}$

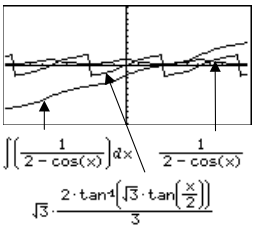
然而，如只有 *上*被省略，*下*会被作为积分常数加入。

∫(a∗x^2,x,c) ENTER $\frac{a \cdot x^3}{3} + c$

相等有效的反导数也许会有数字常数的不同。这样的一个常数尤其在反导数含有对数或反三角函数时，有可能会隐藏起来。再者，分段常数表达式有时回被加入，以使反导数在更大的区间内比普通程式更为有效。

∫(1/(2−cos(x)),x)→tmp(x) ENTER

ClrGraph:Graph tmp(x):Graph 1/(2−cos(x)):Graph √(3)(2tan⁻¹(√(3)(tan(x/2)))/3) ENTER



对于 *表达式1*不能定义成它的内建函数和算子的明显有穷组合的段，∫() 是对应自己。

∫(b∗*e*^(⁻x^2)+a/(x^2+a^2),x) ENTER

当 *下*和 *上*均存在时，会进行一个尝试来在 *下*< *变元*< *上*区间寻找不连续点或不连续导数，以及在它们的位置上将区间分成几部分。

$$\blacksquare \int\left(b\cdot e^{-x^2}+\frac{a}{x^2+a^2}\right)dx$$
$$b\cdot\int\left(e^{-x^2}\right)dx+\tan^{-1}\left(\frac{x}{a}\right)$$

对于 Exact/Approx 模式的 AUTO 设置，当反导数或限制不能定义时，会在可应用的地方使用数字积分法。

对于 APPROX 设置，如可应用的话，会首先进行数字积分法。只有在该数字积分法不能应用或失败时，才会搜寻反导数。

∫(*e*^(⁻x^2),x,⁻1,1)◆ ENTER 1.493...

可套入 ∫() 来进行倍数积分。积分法限制会取决于它们外部的积分法变元。

备注: 也请参阅 **nInt()**。

∫(∫(ln(x+y),y,0,x),x,0,a) ENTER

$$\blacksquare \int_0^a\int_0^x \ln(x+y)\,dy\,dx$$
$$\frac{a^2\cdot\ln(a)}{2}+a^2\cdot(\ln(2)-3/4)$$

√() (平方根)　2nd [√] 键

√ (*表达式1*) ⇒ *表达式*
√ (*数组1*) ⇒ *数组*

对应变数的平方根。

对于数组，是对应*数组1*中所有元素的平方根。

√(4) ENTER　2

√({9,a,4}) ENTER　{3　√a　2}

Π() (积) MATH/Calculus 菜单

Π(*表达式1, 变元, 低, 高***)** ⇒ *表达式*

为*变元*的每个值从*低*到*高*计算*表达式1*，并对应结果的积。

Π(1/n,n,1,5) ENTER $\frac{1}{120}$

Π(k^2,k,1,n) ENTER $(n!)^2$

Π({1/n,n,2},n,1,5) ENTER

$\{\frac{1}{120}\ 120\ 32\}$

Π(*表达式1, 变元, 低, 低*− 1**)** ⇒ 1

Π(k,k,4,3) ENTER 1

Π(*表达式1, 变元, 低, 高***)** ⇒ 1/Π(*表达式1, 变元, 高*+1, *低*− 1) if *高* < *低*− 1

Π(1/k,k,4,1) ENTER 6

Π(1/k,k,4,1)*Π(1/k,k,2,4) ENTER

1/4

Σ() (和) MATH/Calculus 菜单

Σ(*表达式1, 变元, 低, 高***)** ⇒ *表达式*

为*变元*的每个值从*低*到*高*计算*表达式1*，并对应结果的和。

Σ(1/n,n,1,5) ENTER $\frac{137}{60}$

Σ(k^2,k,1,n) ENTER

$\frac{n\cdot(n+1)\cdot(2\cdot n+1)}{6}$

Σ(1/n^2,n,1,∞) ENTER $\frac{\pi^2}{6}$

Σ(*表达式1, 变元, 低, 低*− 1**)** ⇒ 0

Σ(k,k,4,3) ENTER 0

Σ(*表达式1, 变元, 低, 高***)** ⇒ −Σ(*表达式1, 变元, 高*+1, *低*− 1) if *高* < *低*− 1

Σ(k,k,4,1) ENTER −5

Σ(k,k,4,1)+Σ(k,k,2,4) ENTER 4

^ (幂) ^ 键

表达式1 ^ *表达式2* ⇒ *表达式*

数组1 ^ *数组2* ⇒ *数组*

对应自乘到第二变数的幂的第一变数。

对于数组，是对应自乘到*数组2*相应元素的幂的*数组1*元素。

在实数幂中，带有化简指数和奇数分母的分数幂是使用实数分支对复数模式的主分支。

4^2 ENTER 16

{a,2,c}^{1,b,3} ENTER $\{a\ \ 2^b\ \ c^3\}$

表达式 ^ *数组1* ⇒ *数组*

对应自乘到*数组1*元素的幂的*表达式*。

p^{a,2,−3} ENTER $\{p^a\ \ p^2\ \ \frac{1}{p^3}\}$

数组1 ^ *表达式* ⇒ *数组*

对应自乘到*表达式*元素的幂的*数组1*。

{1,2,3,4}^−2 ENTER

{1 1/4 1/9 1/16}

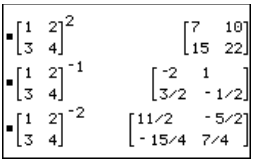
平方矩阵1 ^ *整数* ⇒ *矩阵*

对应自乘到*整数*幂的*平方矩阵1*。

*平方矩阵1*必须是平方矩阵。

如*整数* = ⁻1，计算反矩阵。
如*整数* < ⁻1，将反矩阵计算成合适正数幂。

```
[1,2;3,4]^2 ENTER
[1,2;3,4]^⁻1 ENTER
[1,2;3,4]^⁻2 ENTER
```



(间接) CATALOG

 # 变元名称字符串

请参阅带有名称*变元名称字符串*的变元。这可让您利用字符串在程序中创建和修改变元。

程序段:

```
  ⋮
:Request "Enter Your
Name",str1
:NewFold #str1
  ⋮
  ⋮
:For i,1,5,1
:  ClrGraph
:  Graph i*x
:  StoPic #("pic" & string(i))
:EndFor
  ⋮
```

ʳ (弧度) MATH/Angle 菜单

*表达式1*ʳ ⇒ *表达式*
*数组1*ʳ ⇒ *数组*
*矩阵1*ʳ ⇒ *矩阵*

在 Degree 角模式下，用 180/π 乘*表达式1*。在 Radian 角模式下，对应没有变化的*表达式1*。

此函数给您在 Degree 模式下使用弧度角提供了途径。(在 Degree 角模式下，**sin()**、**cos()**、**tan()** 和极坐标到直角坐标的换算必须以度进行，角变数除外)。

提示: 当使用了函数或程序时，如您想在函数或程度定义中强制弧度而不理会优胜的模式，可使用ʳ。

在 Degree 或 Radian 角模式下:

```
cos((π/4)ʳ) ENTER
```

$$\frac{\sqrt{2}}{2}$$

```
cos({0ʳ,(π/12)ʳ,⁻πʳ}) ENTER
```

$$\{1\ \ \frac{(\sqrt{3}+1)\cdot\sqrt{2}}{4}\ \ -1\}$$

° (度) 2nd [°] 键

表达式° ⇒ *数值*
数组1° ⇒ *数组*
矩阵1° ⇒ *矩阵*

在 Radian 角模式下，用 180/π 乘*表达式1*。在 Degree 角模式下，是对应没变化的*表达式*。

此函数给您在 Radian 模式下使用度角提供了途径。(在 Radian 角模式下，**sin()**、**cos()**、**tan()** 和极坐标到直角坐标的换算必须以弧度进行，角变数除外)。

在 Radian 角模式下:

```
cos(45°) ENTER
```

$$\frac{\sqrt{2}}{2}$$

```
cos({0,π/4,90°,30.12°}) ◆ ENTER
```

{1 .707… 0 .864…}

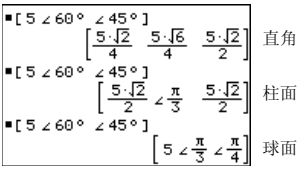
∠ (角)　　2nd [∠] 键

[*半径*,∠θ_*角*] ⇒ *向量* (极坐标输入)
[*半径*,∠θ_*角*,Z_*坐标*] ⇒ *向量*
(柱面输入)
[*半径*,∠θ_*角*,∠φ_*角*] ⇒ *向量*
(球面输入)

对应作为向量的坐标，取决于 Vector Format 模式的设置: 直角、柱面或球面。

[5,∠60°,∠45°] ENTER

在 Radian 模式及向量格式设定成:



(*数量* ∠ *角*) ⇒ *复数值* (极坐标输入)

以(r∠θ)极坐标形式输入复数值。*角*是按照当前的 Angle 模式设置计算。

在 Radian 角模式和 Rectangular 复数格式模式下:

5+3*i*−(10∠π/4) ENTER

5−5·√2+(3−5·√2)·*i*

◆ ENTER　　⁻2.071…−4.071…·*i*

°, ', "　　2nd [°] 键 (°), 2nd [′] 键 (′), 2nd ["] 键 (")

dd°*mm*'*ss.ss*"　⇒　*表达式*

度　一个正数或负数。
分分　一个非负数
秒秒.秒秒　一个非负数

对应 *度度*+(*分分*/60)+(*秒秒.秒秒*/3600)。

这种基-60 输入项格式让您:

- 以度 / 分 / 秒输入角，不理会当前的角模式设置。
- 将时间输入成时 / 分 / 秒。

在 Degree 角模式下:

25°13'17.5" ENTER　　25.221…

25°30' ENTER　　51/2

' (质数)　　2nd [′] 键

变元'
变元"

在微分方程中输入质数符号。单质数符号表示一阶微分方程，两个单质数符号表示二阶微分方程，等等。

deSolve(y''=y^(⁻1/2) and
y(0)=0 and y'(0)=0,t,y) ENTER

$\frac{2\cdot y^{3/4}}{3}=t$

_ (下划线)　　TI-89: ◆ [_] 键　TI-92 Plus: 2nd [_] 键

*表达式*_*单位*

为*表达式*指明单位。所有单位名称必须以下划线开头。

您可使用预定义的单位或创建您自有的单位。您可按下:

TI-89: 2nd [UNITS]
TI-92 Plus: ◆ [UNITS]
来从菜单上选择，或者可以直接键入单位名称。

3_m▸_ft ENTER　　9.842…·_ft

备注: 欲输入 ▸，可按下 2nd [▸]。

*变元_*

当*变元*无值时，它会被当作存在复数处理。默认值是当不带 _ 时，变元会被当作实数处理。

如*变元*已赋值，_ 就会被忽略并*变元*保持为原始的数据类型。

备注: 您可不使用 _ 来将复数存入变元之中。然而，对于 **cSolve()** 和 **cZeros()** 运算中的最佳结果，建议使用 _。

假定 z 是未定义的:

```
real(z) ENTER                    z
real(z_) ENTER            real(z_)

imag(z) ENTER                    0
imag(z_) ENTER            imag(z_)
```

▶ (转换)　　2nd [▶] 键

表达式_单位1 ▶ *_单位2* ⇒ *表达式_单位2*

将表达式从一种单位转换成另一种。单位必须同一类别之内。

_ 下划线字符是指明单位的。对于有效的预定义单位，请参阅本手册中有关常数和计量单位的章节。您可按下:

TI-89: 2nd [UNITS]
TI-92 Plus: ◆ [UNITS]
来从菜单上选择，或可直接键入单位名称。

直接键入单位时，欲获得 _ 下划线，可按下:

TI-89: ◆ [_]
TI-92 Plus: 2nd [_]

备注: 换算算子 ▶ 不能处理温度单位。请使用 **tmpCnv()** 和 **ΔtmpCnv()** 代替。

```
3_m▶_ft ENTER            9.842…·_ft
```

10^()　　CATALOG

10^ (*表达式1***)** ⇒ *表达式*
10^ (*数组1***)** ⇒ *数组*

对应自乘到变数的冪的 10。

对于数组来说，是对应自乘到*数组1*中元素的冪的 10。

```
10^(1.5) ENTER                31.622…

10^{0,-2,2,a} ENTER
```

$$\{1\ \ \frac{1}{100}\ \ 100\ \ 10^{a}\}$$

10^(*平方矩阵1***)** ⇒ *平方矩阵*

对应自乘到*平方矩阵1*的冪的 10。这与计算自乘到每一元素的冪的 10 *不*同。有关计算模式的信息，请参阅 **cos()**。

*平方矩阵1*必须为可对角的，结果一直含有浮点数字。

```
10^([1,5,3;4,2,1;6,-2,1]) ENTER
```

$$\begin{bmatrix} 1.143_E7 & 8.171_E6 & 6.675_E6 \\ 9.956_E6 & 7.115_E6 & 5.813_E6 \\ 7.652_E6 & 5.469_E6 & 4.468_E6 \end{bmatrix}$$

x^{-1}　　CATALOG (^-1)

表达式1 x^{-1} ⇒ *表达式*
数组1 x^{-1} ⇒ *数组*

对应变数的反商。

对于数组，是对应*数组1*中元素的反商。

```
3.1^-1 ENTER                  .322581

{a,4,-.1,x-2}^-1 ENTER
```

$$\{\frac{1}{a}\ \ \frac{1}{4}\ \ -10.\ \ \frac{1}{x-2}\}$$

平方矩阵1 **x**$^{-1}$ ⇒ *平方矩阵*

对应 *平方矩阵1*的逆。

*平方矩阵1*必须为非单一平方矩阵。

```
[1,2;3,4]^-1 ENTER
[1,2;a,4]^-1 ENTER
```

$$\blacksquare \begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}^{-1} \qquad \begin{bmatrix}-2 & 1\\3/2 & -1/2\end{bmatrix}$$

$$\blacksquare \begin{bmatrix}1 & 2\\a & 4\end{bmatrix}^{-1} \qquad \begin{bmatrix}\frac{-2}{a-2} & \frac{1}{a-2}\\ \frac{a}{2\cdot(a-2)} & \frac{-1}{2\cdot(a-2)}\end{bmatrix}$$

| ("With") TI-89: I 键 TI-92 Plus: 2nd [|] 键

表达式 | *布尔表达式1* [*and 布尔表达式2*]...[*and 布尔表达式N*]

"With" (|) 符号用作十进制算子。| 左边的运算域是一条表达式。| 右边的运算域是指定一或多影响表达式化简的关系。| 后面的倍数关系必须加入逻辑 "and"。

"W ith" 算子提供三种基本类型的功能: 代入、区间限制和免除。

代入是在等式中进行，如 x=3 或 y=sin(x)。要达到最高效率，左边应是简单变元。*表达式* | *变元* = *数值* 将代入*表达式*中每个出现的*变元*的*数值*。

区间限制是形成一或多个加入了逻辑算子 "and"的不等式。区间限制也允许化简，如没有区间限制，化简会变成无效或无法计算。

免除是使用"不等" (/= 或 ≠)关系算子来从考虑因素中免除一个特定的数值。当使用 **cSolve()**、**cZeros()**、**fMax()**、**fMin()**、**solve()** 和 **zeros()** 等时，它们主要是用于免除一个精确解的。

```
x+1 | x=3 ENTER                          4
x+y | x=sin(y) ENTER                 sin(y)+y
x+y | sin(y)=x ENTER                     x+y
x^3-2x+7→f(x) ENTER                     Done
f(x) | x=√(3) ENTER                     √3+7
(sin(x))^2+2sin(x)-6 | sin(x)=d
ENTER
                                      d²+2d-6
solve(x^2-1=0,x)|x>0 and x<2
ENTER
                                        x=1
√(x)*√(1/x)|x>0 ENTER                     1
√(x)*√(1/x) ENTER                    √(1/x)·√x
solve(x^2-1=0,x) | x≠1 ENTER x=-1
```

→ (储存) STO▸ 键

表达式 → *变元*
数组 → *变元*
矩阵 → *变元*
表达式 → *函数_名称(参数1,...)*
数组 → *函数_名称(参数1,...)*
矩阵 → *函数_名称(参数1,...)*

如不存在*变元*，是创建*变元*和将它初始化成*表达式*、*数组*或*矩阵*。

如*变元*已存在及未锁定或受保护的，会讲其内容替换成*表达式*、*数组*或*矩阵*。

提示: 如您计划利用未定义变元进行符号运算，应避免将任何内容存入常用的单字母变元如 a、b、c、x、y 和 z 等之中。

```
π/4→myvar ENTER                          π/4
2cos(x)→Y1(x) ENTER                     Done
{1,2,3,4}→Lst5 ENTER            {1 2 3 4}
[1,2,3;4,5,6]→MatG ENTER        [1 2 3; 4 5 6]
"Hello"→str1 ENTER                 "Hello"
```

◉ (注释)

Program Editor/Control 菜单 或
TI-89: [◆] [)] 键
TI-92 Plus: [2nd] X 键

◉ [*文本*]

◉ 是将*文本*当作可用在其它程序指令中的注释行来执行。

◉ 可在行的开头或其它任何位置。◉ 右边至行尾的任何内容为注释。

程序段:

```
  ⋮
:◉ Get 10 points from the Graph
  screen
:For i,1,10 ◉ This loops 10
times
  ⋮
```

0b, 0h

TI-89: [0] [alpha] [B] 键 **TI-92 Plus:** [0] B 键
TI-89: [0] [alpha] [H] 键 **TI-92 Plus:** [0] H 键

0b *十进制数字*
0h *十六进制数字*

相应表示十进制或十六进制数字。欲输入一个十进制或十六进制数字，您必须输入与 Base 模式无关的 0b 或 0h 前缀。不带前缀时，数字会被当作小数(基 10)处理。

结果会按照 Base 模式显示出来。

在小数基模式下:

```
0b10+0hF+10 [ENTER]          27
```

在十进制基模式下:

```
0b10+0hF+10 [ENTER]      0b11011
```

在十六进制基模式下:

```
0b10+0hF+10 [ENTER]        0h1B
```

遇到困难时

如您在操作 TI-89 / TI-92 Plus 时遇到困难，以下的建议有助于您圆满地解决问题。

建议

如:	建议采取的行动:
在显示器上看不到任何东西。	按下 [◆][+] 或 [◆][−] 来分别使显示对比度变暗或变亮。
BATT 指示符显示出来。	更换电池。如 BATT 显示成保留文本 (BATT)，请尽快更换电池。
BUSY 指示符显示出来。	表明一项运算正在进行。如您想停止运算，可按下 [ON]。
PAUSE 指示符显示出来。	一个图形或程序被暂停了， TI-89 / TI-92 Plus正等待输入； 按下 [ENTER]。
显示出一个错误信息。	按下 [ESC] 来清除之。
TI-89 / TI-92 Plus 无法正常工作。	按下 [ESC] 多次来退出任何菜单或对话框并使光标返回输入行中。 — 或 — 确认电池的安装是否正确，以及装上的电池是否是新的。
TI-89 呈“锁定”状且不响应键盘的输入。	1. 取走四节 AAA电池之一； 2. 象您重新装上电池一样按下并按住 [(-)] 和 [)]； 3. 在松开之前继续按住 [(-)] 和 [)] 五秒钟；
TI-92 Plus 呈“锁定”状且不响应键盘的输入。	按下并按住 [2nd] 和 [⊘]，然后按下并松开 [ON]。 — 或 — 如 [2nd] [⊘] 和 [ON] 不能校正问题: 1. 取走四节 AAA电池之一； 2. 象您重新装上电池一样按下并按住 [(-)] 和 [)]； 3. 在松开之前继续按住 [(-)] 和 [)] 五秒钟；

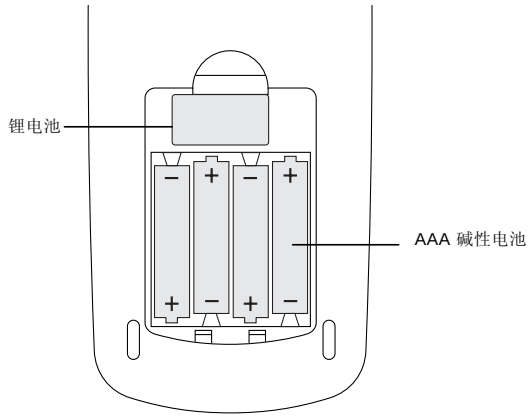
备注: *校正“锁定”状将会重设您的 TI-89 / TI-92 Plus 和清除内存。*

更换 TI-89 中的碱性电池

1. 如 TI-89 正开着，请将其关闭(按下 2nd [OFF])，以防止丢失储存在内存中的信息。
2. 在键盘上滑动护盖。
3. 将计算器持直，向下推电池盒盖的扣并移除电池盒盖。
4. 移除四节已耗尽的 AAA电池。

备注: 为防止内存中储存的信息丢失，TI-89 / TI-92 Plus必须处于关闭状态。请勿同时移除碱性电池和锂电池。

5. 装上四节崭新的 AAA 碱性电池，并按照电池盒内的极性(+ 和 -)图示将它们布置好。



6. 装上电池盒盖。将电池盒盖的两个突角插入电池盒底部的两条槽内，推动盖子直至扣紧为止。

更换 TI-89 中的锂电池

欲更换锂备份电池，可移除电池盒盖，握定 BACK UP BATTERY 盖并拧松小螺丝。

移除旧电池并装上崭新的 CR1616 或 CR1620 电池，正极(+)应面向上。装回盖子和螺丝。

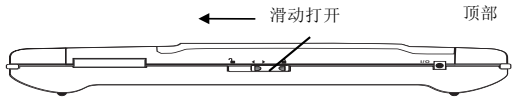
电池预防措施

更换电池时，请遵守以下的注意事项:

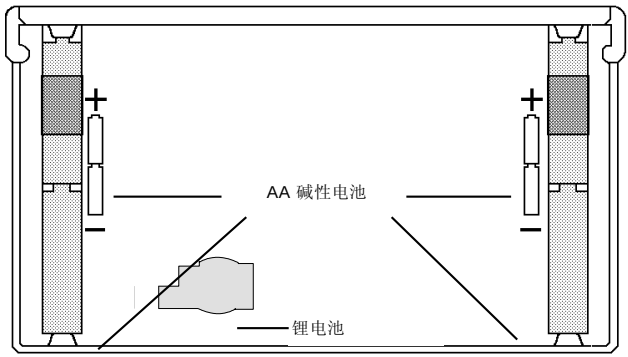
- 请勿将电池留在小孩能够触及的地方。
- 请勿将新旧电池混合，请勿混合牌子(或同一牌子不同型号)使用。
- 请勿混合可充电电池和非可充电电池。
- 按照极性(+ 和 -)图示安装电池。
- 请勿将非可充电电池放入电池充电器内。
- 以适当的方式立刻弃置用过的电池。
- 请勿焚烧或拆开电池。

更换 TI-92 Plus 中的碱性电池

1. 如 TI-92 Plus 正开着，请将其关闭(按下 2nd [OFF])，以防止丢失储存在内存中的信息。
2. 将 TI-92 Plus 计算器持直，用手指将顶部的闩推向左方的松锁位置；向下滑动后盖八分之一英寸并从单元上将其拆下。



3. 移除四节已耗尽的 AA电池。
4. 按照电池盒内的极性图示装上四节新 AA 电池。



5. 重新装上后盖，并将 TI-92 Plus 顶部的闩推向锁定位置，以把后盖在位锁定。
6. 如有需要，可打开 TI-92 Plus 并调整显示对比度。

更换 TI-92 Plus 中的锂电池

欲更换锂备份电池，可从单元上拆除后盖，握定电池盒盖并拧松小螺丝。

除旧电池并装上崭新的 CR2032 电池，正极(+)应面向上。装回盖子和螺丝。

支持和服务信息

关于 TI 的支持、服务和产品的其它信息，请参阅以下内容。

产品支持

美国、加拿大、波多黎各和维尔京群岛的客户

对于一般问题，请联系 Texas Instruments Customer Support:

电话: **1-800-TI-CARES (1-800-842-2737)**
电子邮件: **ti-cares@ti.com**

对于技术问题，请联系 Programming Assistance Group of Customer Support:

电话: **1-972-917-8324**

美国、加拿大、波多黎各和维尔京群岛以外的客户

请通过电子邮件或访问 TI 计算器的国际互联网主页来与 TI 取得联系。

电子邮件: **ti-cares@ti.com**
国际互联网网址: **education.ti.com**

产品服务

仅对美国和加拿大的客户

在把产品送回维修之前，请联系 Texas Instruments Customer Support。

美国和加拿大以外的客户

请参阅包含在本产品内的单张印件或联系您当地的 Texas Instruments 的零售商 / 经销商。

其它 TI 产品和服务

请访问国际互联网上的 TI 产品主页。

education.ti.com