



# **TI-89 Titanium**

## **Voyage™ 200 (航海者)**

### **图形计算器**

© 2004 Texas Instruments Incorporated (德州仪器公司) 版权所有  
Windows 和 Macintosh 为各自所有者之商标

## ***重要信息***

对于任何程序和书面材料，Texas Instruments(德州仪器公司)将不提供明示或默示保证，包括但不限于任何可销售性和特定用途适合性的默示保证，并将本材料以“现况如此，概不负责”的方式提供。无论任何情况下，Texas Instruments 将不向任何人承担由于购买或使用这些材料而引起的或相关的特别、附随、附带或间接伤害的责任，Texas Instruments 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，Texas Instruments 对任何地方使用这些材料而造成的任何种类的损害不负责任。

# 目录

<b>1 引言</b> .....	<b>v</b>
TI-89 Titanium 和 Voyage™ 200 图形计算器 .....	v
用户手册使用指南 .....	v
<b>1 快速入门</b> .....	<b>1</b>
首次启动计算器.....	1
TI-89 Titanium 和 Voyage™ 200 键盘 .....	6
模式设置 .....	14
从目录 (Catalog) 执行命令 .....	16
计算器主屏幕 (Home) .....	18
应用软件的使用.....	20
检查状态信息 .....	26
关闭 Apps 桌面 .....	28
时钟的使用 .....	29
菜单的使用.....	35
使用屏幕拆分 .....	41
应用软件和操作系统 (OS) 版本管理 .....	45
将 TI-89 Titanium 或 Voyage™ 200 连接到另一台设备.....	46
电池.....	48
<b>2 使用预览</b> .....	<b>51</b>
执行运算 .....	51
符号运算 .....	54
常数和度量单位.....	55
基本函数作图 I .....	57
基本函数作图 II .....	59
参变量作图.....	60
极坐标作图 .....	61
序列作图 .....	63
3D 作图 .....	64
微分方程作图 .....	67
添加作图 .....	69
表格.....	71
屏幕拆分 .....	72
数据 / 矩阵编辑器 (Data/Matrix Editor).....	73
统计图和数据图 .....	74
编程 .....	80
文本操作 .....	82
数值求解器 (Numeric Solver) .....	83
数基.....	85
内存及变量管理.....	87

<b>3 应用实例</b> .....	<b>91</b>
分析杆 - 角问题.....	91
推导一元二次方程求解公式.....	92
求解矩阵.....	94
求解 $\cos(x) = \sin(x)$ .....	94
求平行六面体的最小表面积.....	95
用文本编辑器 (Text Editor) 运行教学脚本.....	97
有理函数分解.....	98
统计研究: 数据分类筛选.....	100
TI-89 Titanium / Voyage™ 200 上的 CBL 2™ 程序.....	102
研究棒球的飞行轨迹.....	104
直观表示三次多项式的复零点.....	105
标准年金问题.....	107
计算货币的时间价值.....	108
有理数、实数和复数因式分解.....	110
不放回情况下的抽样模拟.....	110
用向量求解速度.....	111
<b>4 连接方法</b> .....	<b>115</b>
连接两台设备.....	115
变量、闪存 (Flash) 应用软件和文件夹的传送.....	117
程序控制下的变量传送.....	124
操作系统 (OS) 升级.....	126
采集及传送 ID 列表.....	129
TI-89 Titanium, Voyage™200, TI-89 和 TI-92 Plus 间的兼容性.....	130
<b>5 内存及变量管理</b> .....	<b>133</b>
内存检查和复位.....	133
显示 VAR-LINK 屏幕.....	134
通过 VAR-LINK 对变量和文件夹进行操作.....	136
将变量名粘贴到应用软件中.....	143
变量存档和解除存档.....	144
若屏幕上出现碎片收集 (Garbage Collection) 信息.....	145
查看已存档变量时的内存出错信息 (Memory Error).....	147
<b>A 附录 A: 函数和指令</b> .....	<b>149</b>
<b>B 附录 B: TI 产品和服务信息</b> .....	<b>275</b>
索引.....	<b>277</b>

# 引言

## TI-89 Titanium 和 Voyage™ 200 图形计算器

本使用手册为您介绍德州仪器公司提供的两款功能强大、技术先进的图形计算器：TI-89 Titanium 和 Voyage™ 200 图形计算器。

您所使用的 TI-89 Titanium 和 Voyage™ 200 图形计算器装有多种功能各异的应用软件（Apps），可以满足不同学科和课程的需要。

借助 TI-89 Titanium 和 Voyage™ 200 上配备的只读闪存（两者分别配备有 4 兆字节 [MB] 和 2.7 MB 容量），您可以在安装其它应用程序来增强计算器的功能。在计算器上安装应用程序和升级操作系统（OS）的过程类似于在计算机上安装软件。您只需要有 TI Connect™ 软件和 TI 连接电缆即可实现。

TI-89 Titanium 和 Voyage™ 200 计算器采用图形用户界面（GUI）和可配置的应用程序桌面，可以方便地将应用程序按您所创建的分类进行管理。

通过配备各种附件，如 Calculator-Based Laboratory™ (CBL 2™) 系统、Calculator-Based Ranger™ (CBR™) 系统、TI-Presenter™ 视频适配器、以及 TI ViewScreen™ 投影面板，TI-89 Titanium 和 Voyage™ 200 计算器的功能得到进一步的增强。

CBL2™ 和 CBR™ 系统提供静态数据和现实数据采集功能。使用 TI-Presenter™ 视频适配器，可以将 TI-89 Titanium 或 Voyage™ 200 计算器连接到电视机、录像机、摄像机或电脑投影仪等视频显示 / 录制设备上。您可以使用 TI ViewScreen™ 投影面板将 TI-89 Titanium 或 Voyage™ 200 计算器显示的图像放大并投影出来，让教师里所有人都能看清楚。

## 用户手册使用指南

以下说明将有助于您更方便、更有效地使用本用户手册。

- 按键符号将在整本用户手册中出现。
- 许多按键具有不止一个功能。要应用按键上方的第 2 功能，您必须首先按下 **2nd**、**alpha**、或 **◊**。这些增加的功能在本用户手册中写在括号内。

例如，为了显示特殊字符菜单，操作过程会包含以下按键次序：

按下 **2nd** [CHAR]。（按下并松开 **2nd** 键，然后按下 [CHAR] 键，该键为按键“**+**”的第 2 功能。）

- 短语“*按下并保持 (press and hold)*”表示该按键命令需要您同时按下且不释放 2 个按键。例如，将显示对比度调暗的指令为：

按下 **◊** 键并保持，并轻按“**+**”键。

- 您的图形计算器使用菜单来选取各种操作。通常可以使用一到两种方式来选取菜单选项。例如，

按下 **F2** **9:Trig**

表示您可以通过先按下 **F2** 键，然后按需要反复按下按键 “**9**” 或 “**☺**” 来选取 **Trig** 选项，之后再按下 **ENTER** 键。

本使用手册包括以下章节：

**快速入门** – 为所有课程的教师和学生提供有关 **TI-89 Titanium** 和 **Voyage™ 200** 计算器基本操作知识的快速入门。

**使用预览** – 一系列简短示例，包括循序渐进的使用过程、按键操作、以及简单的显示内容。

**应用实例** – 一系列详细实例，说明如何解决、分析和用图形方式显示实际的数学问题。

**连接方法** – 如何使用 **USB** 或 **I/O** 端口将您的图形计算器和其它计算器或计算机连接起来，以及如何传送变量及应用程序、如何升级操作系统的详细说明。

**内存及变量管理** – 如何管理存储在您的图形计算器内存及数据文档中的变量，这里的数据文档指的是 **RAM**（随机存取内存）之外受保护的内存区域。

**技术参考** – 包括操作系统中各功能及指令的句法及作用、按字母排序的操作列表、错误提示、以及其它参考信息。

其它产品信息均在电子文档中。**TI-89 Titanium** 和 **Voyage™ 200** 计算器配套的光盘中载有全套的电子文档。您也可以访问以下网址免费下载上述文档：  
[education.ti.com/guides](http://education.ti.com/guides)

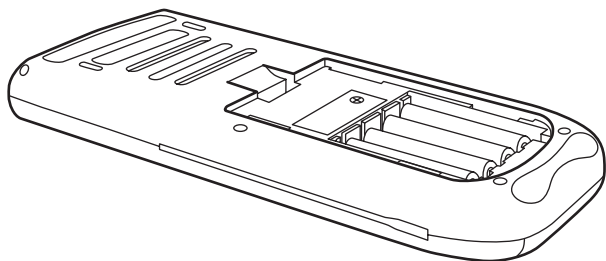
## 快速入门

### 首次启动计算器

#### 安装 AAA 电池

TI-89 Titanium 使用四节 AAA 碱性电池和一节备用氧化银电池 (SR44SW 或 303)。Voyage™ 200 使用四节 AAA 碱性电池和一节锂电池 (CR1616 或 CR1620)。备用电池已安装妥当，AAA 电池随产品附送。

1. 取下计算器背面的电池盖。
2. 根据电池槽上的正 (+) 负 (-) 极标记，将四节 AAA 电池装入电池槽内。



3. 盖上计算器的电池盖。注意：盖子要扣稳机身。

#### 第一次打开您的 TI-89 Titanium 或 Voyage™ 200





电池安装完毕后，按下 **[ON]**，屏幕上会出现 Apps (应用软件) 桌面。

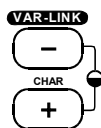
**注：**如果您的计算器需要对预装的应用软件进行初始化，那么屏幕上就不会出现 Apps 桌面，而出现一个进度条和一条提示信息 “Installation in progress . . . Do not interrupt! (正在安装，请勿中断!)”。为了避免应用软件丢失，请勿在初始化过程中取出电池。(您可从随机附带的光盘或 [education.ti.com](http://education.ti.com) 上重新安装应用软件)

## 进度条



## 对比度调整

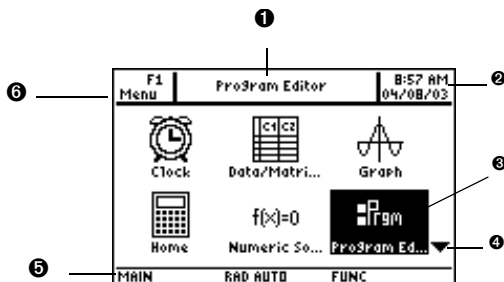
- 要使显示屏变亮，按下并按住  键，再按  键。
- 要使显示屏变暗，按下并按住  键，再按  键。




## Apps 桌面

Apps 桌面是进入 TI-89 Titanium 或 Voyage™ 200 的第一站。为便于访问，您所安装的应用软件将出现在 Apps 桌面上，并以不同的图标归类。从 Apps 桌面，您可以：

- 打开应用软件。
- 选择和编辑应用软件类别。
- 查看所有安装在您的计算器中的应用软件。
- 查看高亮的应用软件全称。
- 查看和编辑时间和日期。
- 检查状态行信息。
- 查看屏幕拆分模式信息



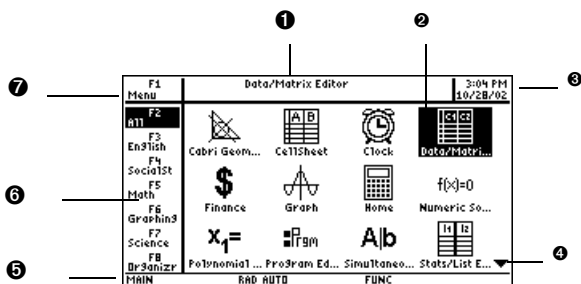
### TI-89 Titanium Apps 桌面

- ① 查看高亮的应用软件全称
- ② 查看时间和日期
- ③ 按下  键打开高亮的应用软件



- ④ 向下滚动来查看其他应用软件
- ⑤ 检查状态行信息
- ⑥ 编辑类别

**注意：**由于显示屏相对较小，TI-89 Titanium 的 Apps 桌面与 Voyage™ 200 的 Apps 桌面看上去有些细微差别。在 Voyage™ 200 的桌面左侧是应用软件分类表，而在 TI-89 Titanium 上却没有，但仍然可以按相同的方式选择软件类别。



### Voyage™ 200 Apps 桌面

- ① 查看高亮的应用软件全称
- ② 按下 **[ENTER]** 键打开高亮的应用软件
- ③ 查看时间和日期
- ④ 向下滚动来查看其他应用软件
- ⑤ 检查状态行信息
- ⑥ 选择应用软件类别
- ⑦ 编辑类别

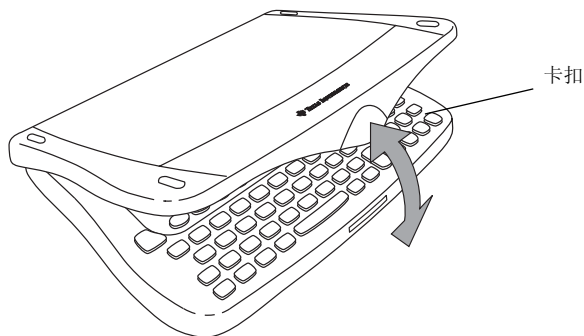
任何时候想返回 Apps 桌面，可按下 **[APPS]** 键。最后一次选定的类别将与最后一次打开的高亮应用软件一起出现。

### 取下和盖上机盖 (Voyage™ 200)

要取下机盖，应当：

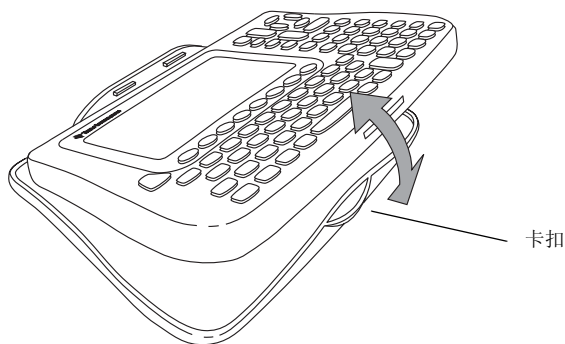
1. 一只手握住计算器。
2. 另一只手扣在机盖卡扣处。
3. 掰动卡扣，揭开机盖

要重新盖上机盖，可将机盖卡扣朝前放置于计算器上，并压下机盖与机身扣合。



## 反扣机盖

要反扣机盖，可将机盖卡扣朝前放置于计算器下方，压下机盖与机身扣合。







## 关闭计算器

按下 **[2nd] [OFF]**。当您下一次打开计算器时，**Apps** 桌面会显示所保存的与关机前相同的设置和内存信息。（如您关闭了 **Apps** 桌面，计算器主屏幕（**Home**）会出现。）

您可以采用以下任一种方式关闭 **TI-89 Titanium / Voyage™ 200**。

按键:	描述
<b>[2nd] [OFF]</b> （按下 <b>[2nd]</b> 然后按下 <b>[OFF]</b> ）	常驻内存功能（ <b>Constant Memory™</b> ）对设置和内存信息进行保存 <ul style="list-style-type: none"> <li>• 如果屏幕上显示有出错信息，则不能用 <b>[2nd] [OFF]</b> 关闭计算器。</li> <li>• 当您再次打开 <b>TI-89 Titanium / Voyage™ 200</b> 时，计算器将显示主屏幕或 <b>Apps</b> 桌面（无论您上一次使用哪个应用软件）</li> </ul>

按键:	描述
 [OFF] (按下  然后按下 [OFF])	除了以下两方面，其余内容与  [OFF] 方式相同 <ul style="list-style-type: none"> <li>即使屏幕上显示有出错信息，您也可以使用  [OFF] 关闭计算器。</li> <li>当您再次打开 TI-89 Titanium / Voyage™ 200 时，计算器将显示您关机时屏幕的显示状态。</li> </ul>

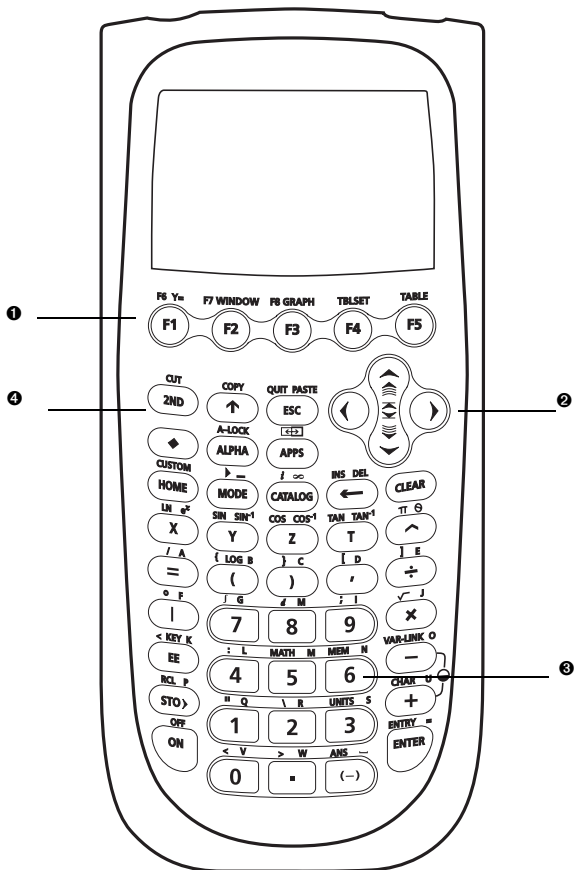
**注意:** [OFF] 是  键第二功能。

计算器的自动断电功能（Automatic Power Down™ (APD™)）让计算器在停止工作几分钟后自动关闭，这样能够延长电池寿命。当你在自动断电后打开计算器时：

- 屏幕显示、光标和任何出错信息都会与自动断电前相同。
- 所有设置和内存信息都将得到保存。

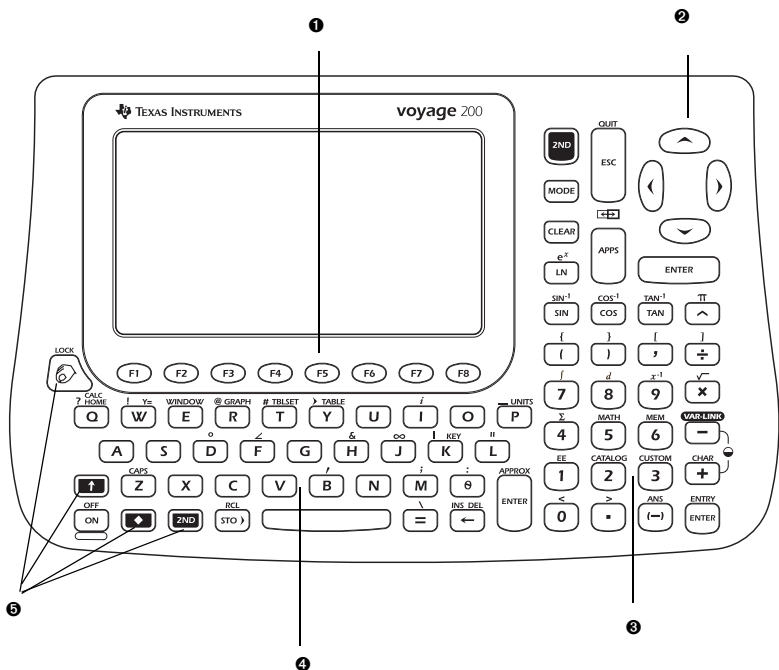
**注意:** 当计算器正处于计算或程序运行状态时，自动断电功能将不起作用，除非计算或程序运行过程被暂停。

## TI-89 Titanium 和 Voyage™ 200 键盘



### TI-89 Titanium 键盘

- ① 功能键 (F1)–(F8) 可打开工具菜单、访问应用程序和编辑应用程序类别。
- ② 光标键 (←, →, ↑, ↓) 用于移动光标。
- ③ 数字键区用于数学计算和函数运算。
- ④ 功能调节键 (2nd, ↕, ↑) 通过更多的按键命令，增加了更多的功能。



## Voyage™ 200 键盘

- ① 功能键 (F1)–(F8) 可打开工具菜单、访问应用程序和编辑应用程序类别。
- ② 光标键 (←, →, ↑, ↓) 用于移动光标。
- ③ 数字键区用于数计算数和函数运算。
- ④ 类似于计算机键盘的标准键盘 (QWERTY 键盘)。
- ⑤ 功能调节键 (2nd, ←, ↑, ↓) 通过更多的按键命令, 增加了更多功能。

## 标准键盘 (只有 Voyage™ 200 具备)

如果您熟悉计算机的键盘输入, 那么您对 Voyage™ 200 的标准键盘将不会感到茫然, 它们之间的差别只在于:

- 要输入大写字母需要先按下 [↑] 键然后再按相应的字母键。
- 要打开大写键盘, 按下 [2nd][CAPS]; 要关闭大写键盘, 按下 [2nd][CAPS]。




## 输入特殊字符

使用 CHAR (字符) 菜单和按键命令输入特殊字符。CHAR 菜单允许您输入希腊字符、数学符号、国际字符和其他特殊字符。屏幕上的软键盘会显示出输入其他常用字符的快捷键位置。

从 CHAR 菜单选择字符：

1. 按下 **[2nd][CHAR]**，出现 CHAR 菜单。
2. 使用光标键选择一个类别，子菜单中会列出该类别中的字符。
3. 使用光标键选择一个字符，然后按下 **[ENTER]** 键。





**例子：**在 Text Editor 中输入右箭头符号 (→)。

按键	结果
<b>[2nd][CHAR]</b>	
<b>4</b>	 <p>向下滚动显示更多字符。</p>
<b>9</b> - 或 - 多次单击 <b>⏏</b> 键选择 <b>9</b> ：→ 并按下 <b>[ENTER]</b> 键	 <p>符号显示在光标所在的位置。</p>


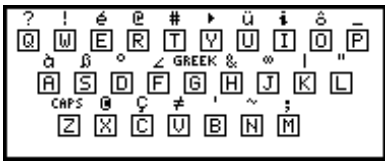

要打开软键盘，可按下 **[KEY]** 键。

要输入更多字符，可按下 **[2nd]** 以及相应的按键。按下 **[ESC]** 退出软键盘。

**TI-89 Titanium 例子:** 使用软键盘查找“不等号”(≠)的快捷键并在程序编辑器 (Program Editor) 中输入该符号。

Press	Result
 [KEY]	
 =	

**Voyage™ 200 例子:** 使用软键盘查找引号 (") 的快捷键并在 Program Editor 中输入该符号。

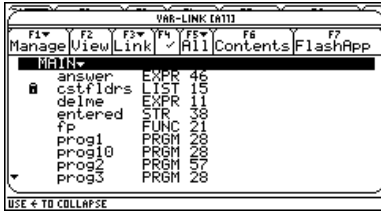
Press	Result
 [KEY]	
<b>2nd</b> L	

## 功能调节键

功能调节键通过增加按键操作的次数，从而带来了更多的功能。要实现调节键功能，按下功能调节键，然后按与操作相应的按键。

按键	描述
<b>[2nd]</b> (第二功能键)	访问应用软件，打开菜单选项和进行其它操作。 第二功能用与 <b>[2nd]</b> 键相同的颜色标记在相应按键的上方。
<b>[◆]</b> (钻石键)	访问应用软件，打开菜单选项和进行其它操作。 钻石键功能用与 <b>[◆]</b> 键相同的颜色标记在相应按键的上方。
<b>[↑]</b> (上档键)	用以输入大写字母。也可在编辑时与 <b>[⏏]</b> 和 <b>[⏏]</b> 配合使用来高亮字符。
<b>[alpha]</b> (Alpha: 只有 TI-89 Titanium 具备)	让您无需标准键盘也可输入字母。字母用与 <b>[alpha]</b> 键相同的颜色标记在相应按键的上方。
<b>[☞]</b> (手形键: 只有 Voyage 200 具备)	您能使用光标键实现对几何对象的操作。也可以用来绘制图形。

**例子:** 打开 VAR-LINK [All] 屏幕，对变量和应用软件进行管理。

Press	Result
<b>[2nd]</b> [VAR-LINK]	

## 功能键

使用功能键进行如下操作：

- 在 APPs 桌面打开应用软件，并选择或编辑应用软件类别。
- 在计算器主屏幕，打开工具条菜单，选择与数学有关的运算。
- 在应用软件里，打开工具条菜单选择应用软件选项。



## 光标键

按下  $\leftarrow$ ,  $\rightarrow$ ,  $\uparrow$ , 或  $\downarrow$  将光标朝相应的方向移动。根据不同的应用软件，以及  $\boxed{2nd}$  和  $\blacklozenge$  功能调节键使用与否，光标键可通过多种方式来移动光标。

- 使用  $\leftarrow$  或  $\rightarrow$  逐行上下移动光标。
- 使用  $\boxed{2nd} \leftarrow$  或  $\boxed{2nd} \rightarrow$  将光标移至行首或行尾。
- 使用  $\boxed{2nd} \uparrow$  或  $\boxed{2nd} \downarrow$  逐页上下移动光标。
- 使用  $\blacklozenge \leftarrow \blacklozenge \rightarrow$  将光标移至页首或页尾。
- 使用  $\leftarrow$  和  $\uparrow$ 、 $\leftarrow$  和  $\downarrow$ 、 $\rightarrow$  和  $\uparrow$ 、或  $\rightarrow$  和  $\downarrow$  时光标对角移动。（同时按下前述每对按键。）

## 数字键区

您可在数字键区输入正数和负数。

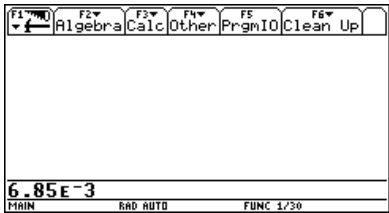
要输入负数，在输入数字前先按  $\boxed{-}$ 。

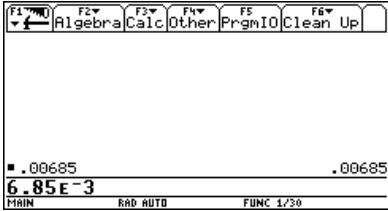
**注意：**不要将负数键 ( $\boxed{-}$ ) 与减法键 ( $\boxed{-}$ ) 相混淆。

以科学记数法输入数字：

1. 在指数前输入数字。（该数值可以是一个表达式）
2. 按  $\boxed{2nd} \boxed{EE}$  (TI-89 Titanium) 或  $\boxed{2nd} \boxed{EE}$  (Voyage™ 200)。指数符号 (E) 紧跟在输入的数字后面。
3. 以整数形式输入多达三位数的指数。（如下例所示，您可以输入负指数。）

**例子：**用科学记数法在计算器 Home 屏幕上输入 0.00685。

Press	Result
$0 \boxed{-} 85$	
TI-89 Titanium: $\boxed{EE}$ Voyage™ 200: $\boxed{2nd} \boxed{EE}$	
$\boxed{-} 3$	

Press	Result
[ENTER]	

## 其他重要按键

按键命令	描述
◆ [Y=]	显示 Y= Editor。
◆ [WINDOW]	显示窗口编辑器 (Window Editor)。
◆ [GRAPH]	显示绘图 (Graph) 屏幕。
◆ [TBLSET]	设置表 (Table) 屏幕。
◆ [TABLE]	显示 (Table) 屏幕。
TI-89 Titanium:	使用这些按键, 通过剪切、复制或粘贴操作来编辑已经输入到屏幕的信息。
◆ [CUT]	
◆ [COPY]	
◆ [PASTE]	
Voyage™ 200:	
◆ X (剪切)	
◆ C (复制)	
◆ V (粘贴)	
◆ S 只有 Voyage™ 200 具备	显示 “另存为 (SAVE COPY AS)” 对话框, 提示您选择一个文件夹并输入一个变量名来保存屏幕上的数据。
◆ N 只有 Voyage™ 200 具备	创建新文件。
◆ O 只有 Voyage™ 200 具备	打开制定的文件。 .

按键命令	描述
[♦] F 只有 Voyage™ 200 具备	显示格式（FORMATS）或图形格式（GRAPH FORMATS）对话框，设置活动应用程序的格式。
[APPS]	显示 Apps 桌面。
[♦] [APPS]	在 Apps 桌面关闭的情况下，显示 FLASH 应用程序（FLASH APPLICATIONS）菜单。
[2nd] [⇄]	在两个选中的应用程序之间切换。
[2nd] [CUSTOM]	打开和关闭用户自定义菜单。
[2nd] [▶]	转换度量单位。
TI-89 Titanium: [♦] [-]	指定度量单位。
Voyage™ 200: [2nd] [-]	
[←]	删除光标左边的字符（退格键）。
[♦] [DEL]	删除光标右边的字符。
[2nd] [INS]	在插入和改写模式之间切换。
[2nd] [MEM]	显示内存（MEMORY）屏幕。
TI-89 Titanium: [CATALOG]	显示命令表。
Voyage™ 200: [2nd] [CATALOG]	
[2nd] [RCL]	调用变量内容。
[STO▶]	保存变量值。
[2nd] [CHAR]	显示 CHAR 菜单，在此您可以选择希腊字母、国际重音符号。
[2nd] [QUIT]	<ul style="list-style-type: none"> <li>在全屏模式下显示 Apps 桌面。</li> <li>在屏幕拆分模式下，显示活动应用程序的全屏显示外观。</li> <li>在 Apps 桌面关闭的情况下，显示计算器主屏幕（Home）。</li> </ul>

## 模式设置

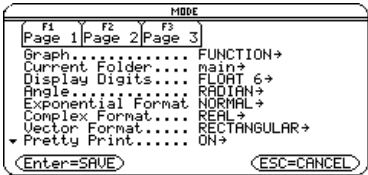
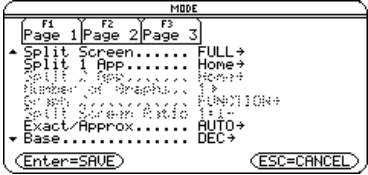
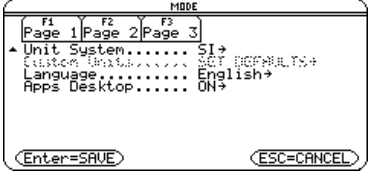
计算器模式控制着 TI-89 Titanium 或 Voyage™ 200 显示和解释信息的方式。包括矩阵和数组元素的所有数字会根据当前的模式设置进行显示。当 TI-89 Titanium 或 Voyage™ 200 关闭后，常驻内存（Constant Memory™）功能会保存您选定的所有模式设置。

要查看 TI-89 Titanium 或 Voyage™ 200 模式设置：

1. 按下 **[MODE]**，出现模式（MODE）对话框的第 1 页。
2. 按下 **[F2]** 或 **[F3]** 显示 屏幕上第 2 页或第 3 页的模式列表。

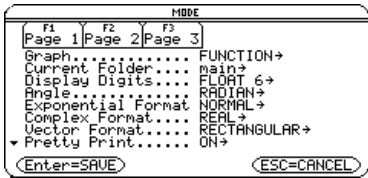
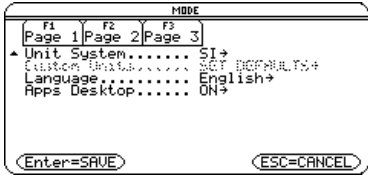
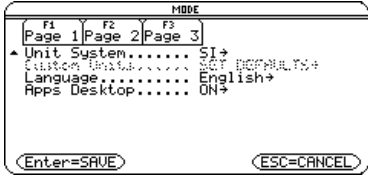


**注意：** 变灰状态的模式只有在选定了其要求的模式以后方能使用。例如，第 3 页列出的自定义单位（Custom Units）模式只有在单位制（Unit System）模式设定成 CUSTOM 后才可用。

### 查看模式设置

按键	结果
<b>[MODE]</b>	
<b>[F2]</b>	
<b>[F3]</b>	

## 改变模式设置

**例子:** 将语言 (Language) 模式设定为西班牙语语言 (*Español*)。

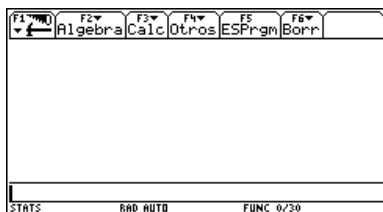
按键	结果
<b>MODE</b>	 The screenshot shows the MODE menu with options: Graph, Current Folder, Display Digits, Angle, Exponential Format, Complex Format, Vector Format, and Pretty Print. The 'Language' option is not yet visible.
<b>F3</b>	 The screenshot shows the MODE menu with 'Unit System' selected. The 'Language' option is now visible and set to 'English'.
向下滚动语言区域。 ⏴	 The screenshot shows the MODE menu with 'Language' selected. The list of languages is visible, including English, Deutsch, Español, and Français.
按 ⏴ 然后按下 ⏵, 使 <b>3:Español</b> 呈 高亮显示。	 The screenshot shows the MODE menu with 'Language' selected. The list of languages is visible, and '3:Español' is highlighted.
<b>ENTER</b>	 The screenshot shows the MODE menu with 'Language' selected. The list of languages is visible, and 'Español' is now the selected language.

---

按键

结果

**[ENTER]**



**注意：**上一次打开的应用软件出现（该例使用计算器主屏幕（Home））。

要将语言模式重新设定为英文，可重复以上操作步骤，在语言区域选择 **1:English**。

## 从目录（Catalog）执行命令

从目录访问 TI-89 Titanium 或 Voyage™ 200 命令表，包括函数、指令和用户自定义程序。命令按字母顺序列出，不是以字母（比如：&, /, +, -, etc.）开头的命令位于命令表的尾部。

目录帮助软件包含了所有命令的用法。

当前无效的选项将变灰。例如，如果您的 TI-89 Titanium 或 Voyage™ 200 没有安装 Flash 应用软件，Flash 应用软件（**[F3]**）菜单选项将是灰的；如果没有创建函数或程序，用户自定义（**[F4]**）菜单选项将是灰的。

**注意：**键入一个字母，以该字母为首字母的命令将出现在列表中最前面的位置。

---

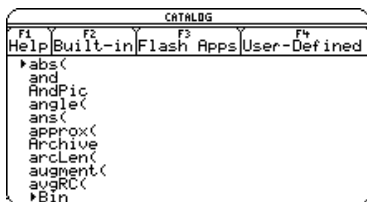
按键

结果

TI-89 Titanium: **[CATALOG]**

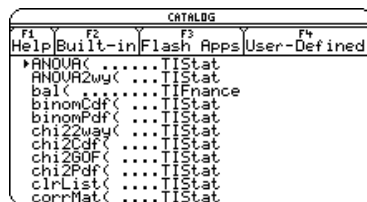
Voyage™ 200: **[2nd][CATALOG]**

（显示嵌入式（Built-in）命令）



**[F3]**

（如果有，显示 Flash 应用软件命令）



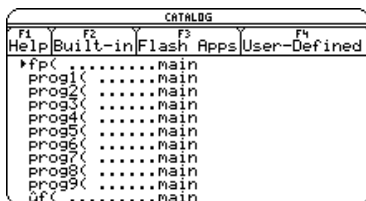
---

**按键****结果**

---

**[F4]**

(如果有, 显示用户自定义命令)



---

从目录选择命令并插入到计算器主屏幕输入行或粘贴至其他应用软件, 如: Y= Editor, Text Editor, 或 电子表格 (CellSheet™) 应用软件。

**例子:** 将 **comDenom**( 命令插入到计算器 主屏幕输入行。

**注意:** 选择命令前, 先将光标定位在希望命令出现的位置。

按下 **[2nd]** 主页向前翻动目录表。

---

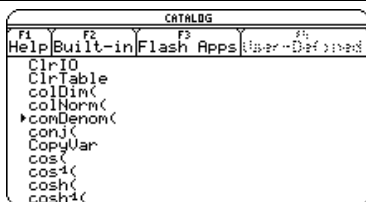
**按键****结果**

---

TI-89 Titanium: **[CATALOG]** **[alpha]** **C**

Voyage™ 200: **[2nd]** **[CATALOG]** **C**

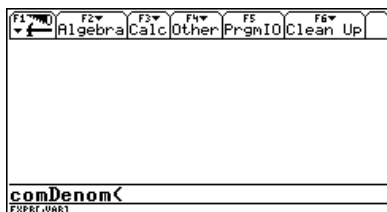
**[2nd]**



然后按下 直到指针移动到

**comDenom**( 函数。

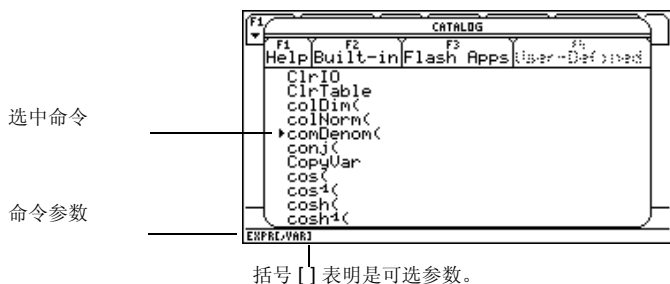
---

**[ENTER]**

---

状态行显示所选中命令的所有必选和可选参数。可选参数出现在方括号里面。

**注意:** 按下 **[F1]** 也可显示所选中命令的参数。



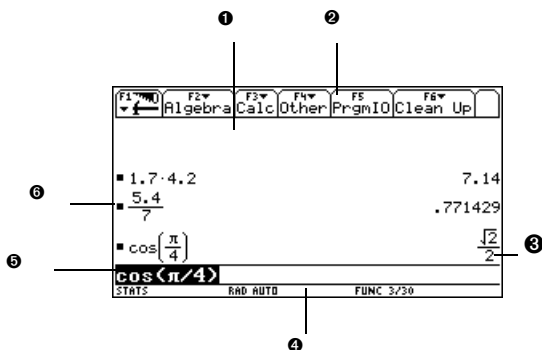
若要不选择任何命令而退出目录，按 **[ESC]** 键。

## 计算器主屏幕 (Home)

计算器主屏幕 (Home) 是数学运算的基本屏幕，包括执行指令、计算表达式和查看结果。

要显示计算器主屏幕，按下 **[HOME]**。

您也可以在 Apps 桌面上选择 Home 图标，并按下 **[ENTER]** 键来显示计算器主屏幕。



- ① 历史区域显示输入项 / 答案对。
- ② 包含相应运算列表的菜单项。按 **[F1]**, **[F2]**, 等显示菜单。
- ③ 上一次输入项结果在此显示。(注意：结果不显示在输入行中。)
- ④ 状态行显示计算器当前状态。
- ⑤ 输入行显示当前输入项。
- ⑥ 您的上一次输入项在此显示。

要从计算器主屏幕返回 Apps 桌面，请按 **[APPS]** 键。



## 关于历史区域

根据表达式的复杂性和表达式高度的不同，历史区域最多可显示八个输入项 / 答案对。当显示区满后，信息会滚出屏幕顶部。历史区域用于：

- 查看以前的输入项和答案。使用光标键查看滚出屏幕的输入项和答案。
- 在输入行调用或自动粘贴一个前面的输入项或答案以重新使用或编辑。（要了解更多信息，可参阅电子文档的 *计算器操作 (Operating the Calculator)* 一章。）

通常停留在输入行的光标可移至历史区域，以下表格演示了如何让光标在历史区域内移动。

要：	请如下操作：
查看滚出屏外的输入项 / 答案对。	按 $\leftarrow$ 键，光标从输入行移至历史区域并高亮最近一个答案项。 连续按 $\leftarrow$ 键，将光标依次从答案项移至输入项，并穿过整个历史区域。
转至最早或最新的输入项 / 答案对。	如果光标已经在历史区域，按下 $\leftarrow$ 或 $\rightarrow$ 键。
查看长度超出一行的输入项或答案（▶符号标记在该行的末尾）。	将光标移至输入项或答案项。使用 $\leftarrow$ 键或 $\rightarrow$ 键向左或者向右滚动，或按 $\left[\text{2nd}\right] \leftarrow$ 或 $\left[\text{2nd}\right] \rightarrow$ 将光标直接转至行首或行尾。
将光标返回输入行。	按 $\left[\text{ESC}\right]$ 键 或 $\rightarrow$ 键，直至光标回到输入行。

## 在状态行显示历史信息

通过状态行的历史标记了解输入项 / 答案对的信息，例如：

如果光标在输入行：

当前保存的总对数       $\text{8/30}$       可保存的最大对数

如果光标在历史区域：

高亮的输入项 / 答案对  
的位置       $\text{8/30}$       当前保存的总对数

## 修改历史区域

要改变可保存的输入项 / 答案对的容量：

1. 在计算器主屏幕按  $\left[\text{F1}\right]$  键，并选择 **9:Format**。
2. 按下  $\rightarrow$  键，并使用  $\leftarrow$  或  $\rightarrow$  选择新的值。

### 3. 按下 **[ENTER]** **[ENTER]**。

要清空历史区域并删除所保存的信息：

- 在计算器主屏幕按 **[F1]** 键，并选择 **8:Clear Home**。  
- 或 -
- 在计算器主屏幕的输入行中输入 **ClrHome**。

要删除某个输入项 / 答案对，可将光标移至该输入项或答案项，并按下 **[←]** 键或 **[CLEAR]** 键。

## 应用软件的使用

TI-89 Titanium 或 Voyage™ 200 在 **Apps** 桌面对应用软件分类管理。要选择一个类别，按下功能键 (**[F2]** 至 **[2nd][F8]**，或 **[F2]** 至 **[F8]**)，所选择类别中的所有图标会显示在 **Apps** 桌面。

**注：**如果某应用软件图标下面的名称没有完全显示出来，可使用光标使其高亮，这样就可以在 **Apps** 桌面最上面的显示栏看到该应用软件的全称。

### 打开应用软件


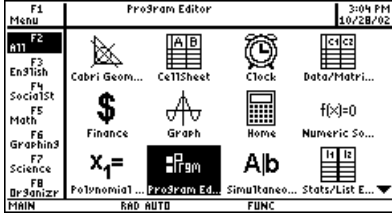




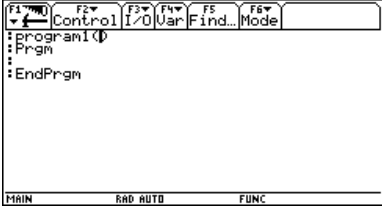
用光标键高亮 **Apps** 桌面的应用软件图标，并按下 **[ENTER]** 键，该应用软件会直接打开或出现一个对话框。在最常用的对话框列中，包含以下应用软件操作选项：

**注：**TI-89 Titanium 或 Voyage™ 200 采用以通用术语命名的 **变量** 来提示您要创建的该应用软件的数据文件。

选项	说明
当前	返回上次查看应用软件时的屏幕显示状态，如应用软件变量不存在，则会出现一个新的对话框。
打开	打开现有文件。
新建	在输入栏中输入文件名，创建新文件。

选定一个选项，输入所需信息，并按 **[ENTER]** 键，应用软件就会出现。

例子：使用程序编辑器 (Program Editor) 创建一个新程序。

按键	结果
用光标键将  Program Ed... 高亮显示	
[ENTER]	
3	
[ENTER]	
⏪ ⏩ <b>program 1</b>	
[ENTER] [ENTER]	

新创建的程序变量 *program1* 被保存在主 (Main) 文件夹中。

### 从当前应用软件返回 Apps 桌面

按 [APPS] 键，上次选定的应用软件类别的图标会出现在 Apps 桌面，上次打开的应用软件的图标呈高亮显示。

您也可以在全屏模式下按 **[2nd] [QUIT]** 返回 **Apps** 桌面；在屏幕拆分模式下按两次 **[2nd] [QUIT]**。

要从 **Apps** 桌面返回上一次打开的应用软件，按 **[2nd] [≡]**。

## 选择应用软件类别

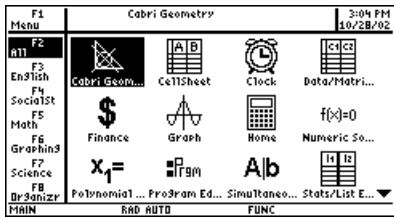
对于 **TI-89 Titanium**，应用软件类别的名称只出现在 **F1** 菜单中。要选择应用软件类别，按下 **[F1] 2:Select Category** 并移动光标键使应用软件类别高亮显示，然后按 **[ENTER]** 键选定该类别。您可以使用功能键快捷方式从键区选择类别（如需要，可使用 **[2nd]** 键），然后 **Apps** 桌面上将出现选定类别的应用软件图标。

对于 **Voyage™ 200**，应用软件类别名称靠 **Apps** 桌面的左侧显示。要选择应用软件类别，按相应的功能键（在 **Apps** 桌面上，显示在类别名称的上方）。

选定类别的应用软件图标出现在 **Apps** 桌面上。

按键	说明
<b>[F2] All</b>	显示所有已安装的应用软件图标。不允许自定义。
<b>[F3] English</b>	允许自定义类别。缺省为 <b>English</b> （英文）。
<b>[F4] SocialSt</b>	允许自定义类别。缺省为 <b>SocialSt</b> （社会学）。
<b>[F5] Math</b>	允许自定义类别。缺省为 <b>Math</b> （数学）。
<b>[2nd] [F6] Graphing</b> 或 <b>[F6] Graphing</b>	允许自定义类别。缺省为 <b>Graphing</b> （作图）。
<b>[2nd] [F7] Science</b> 或 <b>[F7] Science</b>	允许自定义类别。缺省为 <b>Science</b> （科学）。
<b>[2nd] [F8] Organizr</b> 或 <b>[F8] Organizr</b>	允许自定义类别。缺省为 <b>Organizr</b> （组织器）。

例子：选择所有类别。

按键	结果
<b>[F2]</b>	

如果选择了一个不含应用软件的空类别，会出现一条确认该类别为空的信息，并提示您按 **[F1] 1:Edit Categories** 菜单，这样您可以往类别中添加应用软件快捷键。（参见第 23 页的“自定义应用软件类别”）

按 **[ENTER]** 或 **[ESC]** 键清除信息并返回 Apps 桌面。

### 自定义应用软件类别

TI-89 Titanium 或 Voyage™ 200 将应用软件分为七大类，其中六个可通过自定义方式满足个人的要求。（All 类别包含所有已安装的应用软件，且不可编辑。）

要自定义从 **[F3]** 到 **[2nd] [F8] ([F8])** 的应用软件类别：

1. 选择 **[F1] 1:Edit Categories**。出现的子菜单显示了六个可自定义的应用软件类别名称。（All 类别不列出。）
2. 高亮某个应用软件类别并按下 **[ENTER]** 键，出现 **Edit Categories**（编辑类别）对话框，框中列出了所有已安装的应用软件，并在一个小框中标明了高亮类别的名称。
3. 要修改应用软件类别名称，可键入您想要的名称。

**注：**类别名称长度不超过 8 个字符，含字母（大写或小写）、数字标点符号和重音符号。

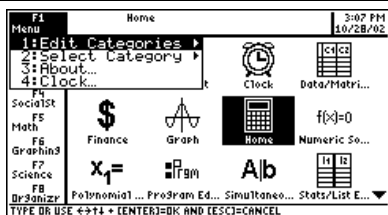
4. 要在类别中添加或删除应用软件快捷方式，可按 **⊖** 高亮该软件旁边的复选框，然后按 **⊕** 键来添加或去除选择标记 (✓)。
5. 要保存修改并返回 Apps 桌面，可按下 **[ENTER]** 键。

**例子:**用Business类替换Social Studies类, 并添加CellSheet™ 和 Finance的快捷方式。

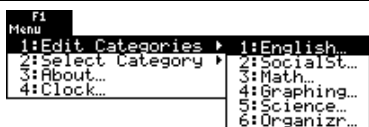
按键

结果

F1



⏴



2

- 或 -

⏴ ENTER



TI-89 Titanium: [2nd] [a-lock]

⏴ **Business**

Voyage™ 200:

⏴ **Business**

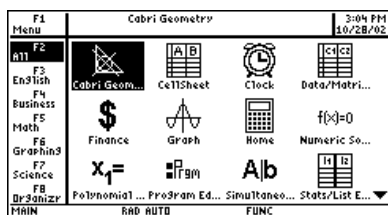
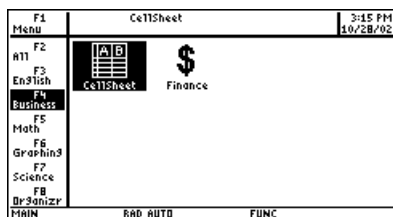


⏴

⋮

⏴



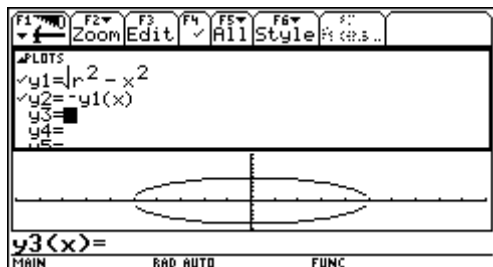
**按键****结果****[ENTER]****[F4]**

## 打开应用软件和屏幕拆分状态栏

TI-89 Titanium 或 Voyage™ 200 允许拆分屏幕来同时查看两个应用软件。例如，分别在两个半屏同时打开 **Y= Editor** 和 **Graph**，以查看函数列表及其如何进行作图。

在模式（MODE）屏幕的第 2 页选择屏幕拆分（Split Screen）模式。TI-89 Titanium 或 Voyage™ 200 按如下所示显示选定的应用软件。屏幕可被水平（左右）拆分或垂直（上下）拆分。

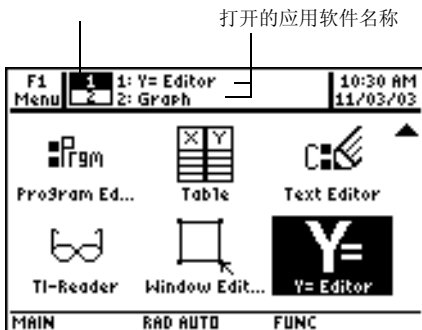
上下拆分模式



要返回 Apps 桌面，按 [APPS] 键。Apps 桌面顶部会显示屏幕拆分状态栏，标明了打开的应用软件名称及其所在屏幕。状态栏中高亮的数字提示您下一个应用软件将在哪个屏幕打开。

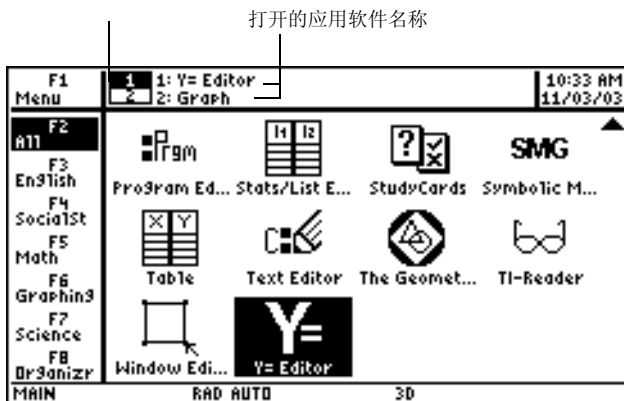
**注：** Apps 桌面总是以全屏形式出现。

屏幕拆分状态（高亮的数字提示您下一个应用软件在哪个屏幕打开。）



### TI-89 Titanium Apps 桌面的屏幕拆分状态标记

屏幕拆分状态（高亮的数字提示您下一个应用软件在哪个屏幕打开。）



### Voyage™ 200 Apps 桌面的屏幕拆分状态标记

您可以了解关于屏幕拆分的更多信息。（要了解更多信息，可参阅电子文档 *屏幕拆分 (Split Screens)* 一章）

## 检查状态信息

有关 TI-89 Titanium 或 Voyage™ 200 的当前状态信息，可查看屏幕底部的状态行。



MAIN	2ND	RAD	AUTO	GR#1	FUNC	22/30	BATT	BUSY
1	2	3	4	5	6	7	8	9

标记	含义
1 当前文件夹	当前选定文件夹的名称 (MAIN 为缺省文件夹)。
2 功能调节键	选定的功能调节键 (2ND, $\square$ , 1), 如有。
3 角度 (Angle) 模式	选定的角度单位 (RAD, DEG)。
4 精确 / 近似 (Exact/Approx) 模式	答案计算和结果显示模式 (AUTO, EXACT, APPROX)。
5 图形编号	在屏幕拆分模式下打开两个相互独立的图形 (GR#1, GR#2)。
6 作图 (Graph) 模式	选定的可绘制的图形类型。(FUNC, PAR, POL, SEQ, 3D, DE)。
7 输入项 / 答案对	22/30- 计算器主 (Home) 屏幕的历史区域的输入项 / 答案对的数量 (缺省为 30, 最大为 99)。
8 更换电池	当电池电量不足时出现 BATT。如果 BATT 以黑底突出显示 (BATT), 请尽快更换电池。
9 忙 / 暂停、锁定的 / 存档的变量	BUSY- 计算或作图进行中。 PAUSE- 暂停作图或程序。 ☒- 在当前编辑器中打开的变量状态为锁定或存档, 不能修改。

MAIN	2ND	$\square$	RAD	AUTO	GR#1	FUNC	22/30	BATT	BUSY
1	2	3	4	5	6	7	8	9	10

标记	含义
1 当前文件夹	当前选定文件夹的名称 (MAIN 为缺省文件夹)。

标记	含义
② 功能调节键	选定的功能调节键 ( $\text{2nd}$ , $\text{◀}$ , $\text{▶}$ ), 如有。
③ 手形 key	手形 $\text{☞}$ 功能调节键已被选定。
(只有 Voyage™ 200 具备)	
④ 角度模式	选定的角度单位 (RAD, DEG)。
⑤ 精确 / 近似 (Exact/Approx)	答案计算和结果显示的模式 (AUTO, EXACT, APPROX)。
⑥ 图形编号	在屏幕拆分模式下打开两个相互独立的图形 (GR#1, GR#2)。
⑦ 作图 (Graph) 模式	选定的可绘制的图形类型 (FUNC, PAR, POL, SEQ, 3D, DE)。
⑧ 输入项 / 答案对	22/30- 计算器主 (Home) 屏幕的历史区域的输入项 / 答案对的数量 (缺省为 30, 最大为 99)
⑨ 更换电池	当电池电量不足时出现 BATT。如果 BATT 以黑底突出显示 ( <b>BATT</b> ), 请尽快更换电池。
⑩ 忙 / 暂停、锁定的 / 存档的变量	BUSY- 计算或作图进行中。 PAUSE- 暂停作图或程序。 ■- 在当前编辑器中打开的变量状态为锁定或存档, 不能修改。

## 关闭 Apps 桌面

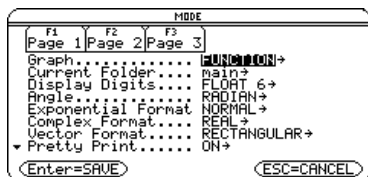
您可以在模式 (MODE) 对话框中关闭 Apps 桌面。这样, 您可从应用软件 (APPLICATIONS) 菜单打开应用软件。要打开 APPLICATIONS 菜单, 按 **[APPS]** 键。

**例子:** 关闭 Apps 桌面。

按键

**[MODE]**

结果



## 按键

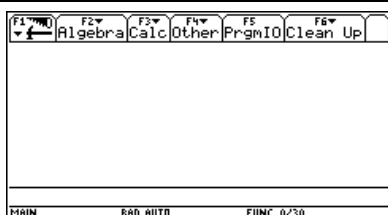
## 结果

[F3]



[ENTER] [ENTER]

注：上次打开的应用软件会出现。（本例所用屏幕为主屏幕（Home））。



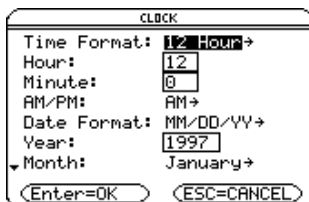
要打开 Apps 桌面，可重复以上操作，在 Apps Desktop 控制中选择 ON。要从计算器主屏幕（Home）返回 Apps 桌面，按 [APPS] 键。

## 时钟的使用

通过 **CLOCK** 对话框设置时间和日期，选择时钟显示格式，打开和关闭时钟。

时钟默认为“打开”。如果您关闭了时钟，那么除了 Clock ON/OFF（时钟开/关）以外，其它所有 Clock 对话框选项都会变灰。

▼ 符号表示您可以向下滚动以查看更多选项



## 显示时钟对话框

1. 用光标键将 Apps 桌面的时钟图标呈高亮显示。

- 按 **[ENTER]** 键，**CLOCK** 对话框将出现，时间格式 (**Time Format**) 区域呈高亮显示。

**注：**由于 **CLOCK** 对话框显示了打开对话框时的当前设置，因此，您可能需要在退出前更新时间。

## 时间设置

- 按 **[1]** 键打开时间格式列表。
- 按 **[<]** 或 **[>]** 键使高亮某一选项，然后按 **[ENTER]** 键，选定格式就出现在 **Time Format** 域中。
- 按 **[<]** 键高亮小时域 (**Hour**)。
- 键入小时值，按 **[<]** 键高亮分钟域 (**Minute**)。
- 键入分钟值。
- 如果采用 24 小时制，请转到第 9 步。  
— 或 —  
如果采用 12 小时制，按 **[<]** 键高亮 **AM/PM** 域。
- 按 **[1]** 键打开 **AM/PM** 选项。
- 按 **[<]** 或 **[>]** 键高亮某个 **AM/PM** 选项，然后按 **[ENTER]** 键，选定的 **AM/PM** 格式就会出现。
- 设置日期 (具体操作请参阅 *日期设置*)。  
— 或 —  
按下 **[ENTER]** 键，保存设置并退出。**Apps** 桌面右上角的时间会被更新。

## 日期设置

- 根据需要按 **[<]** 或 **[>]** 键使日期格式 (**Date Format**) 域突出显示。
- 按 **[1]** 键日期格式列表。
- 按 **[<]** 或 **[>]** 键高亮某一选项，然后按 **[ENTER]** 键，选定格式就出现在 **Date Format** 域中。
- 按 **[<]** 键高亮年份域 (**Year**)。
- 键入年份值，然后按 **[<]** 键高亮月份域 (**Month**)。
- 按 **[1]** 打开月份表。
- 按 **[<]** 或 **[>]** 键高亮某一选项，然后按 **[ENTER]** 键，选定月份就出现在月份域中。
- 按 **[<]** 键高亮日域 (**Day**)。

9. 键入日值，然后按下 **[ENTER]** **[ENTER]** 保存设置并退出。Apps 桌面右上角的日期会被更新。

**例子：** 将日期设定成 19/10/02 (October 19, 2002)，时间设定成 1:30 p.m.。

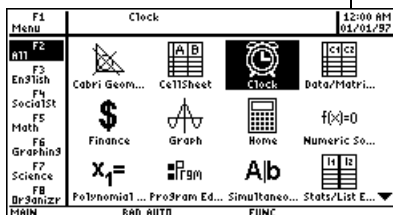
**按键**

**结果**

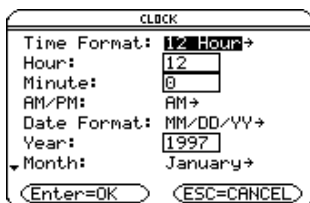
用光标键高亮以下图标：



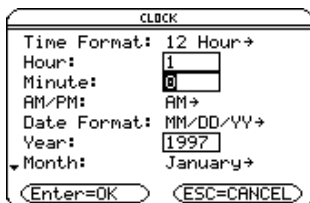
时间和日期



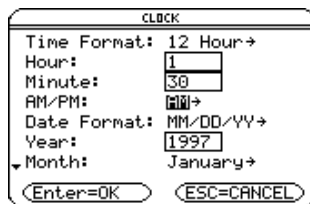
**[ENTER]**



**1**



**30**



按键

结果

↓ ↻

CLOCK

Time Format: 12 Hour→

Hour: 1

Minute: 30

AM/PM: 1:AM

Date Format: 23/01/YY→

Year: 1997

Month: January→

Enter=OK    ESC=CANCEL

ENTER ↻

CLOCK

Time Format: 12 Hour→

Hour: 1

Minute: 30

AM/PM: PM→

Date Format: MM/DD/YY→

Year: 1997

Month: January→

Enter=OK    ESC=CANCEL

↓ ↻

CLOCK

Time Format: 12 Hour→

Hour: 1

Minute: 30

AM/PM: 1:MM/DD/YY

Date Format: 23/01/YY

Year: 3:MM.DD.YY

Month: 4:DD.MM.YY

5:YY.MM.DD

6:MM-DD-YY

7:DD-MM-YY

8:YY-MM-DD

Enter=OK

ENTER ↻

CLOCK

Time Format: 12 Hour→

Hour: 1

Minute: 30

AM/PM: PM→

Date Format: DD/MM/YY→

Year: 1997

Month: January→

Enter=OK    ESC=CANCEL

2002

CLOCK

Time Format: 12 Hour→

Hour: 1

Minute: 30

AM/PM: PM→

Date Format: DD/MM/YY→

Year: 2002

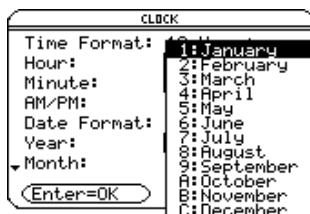
Month: January→

Enter=OK    ESC=CANCEL

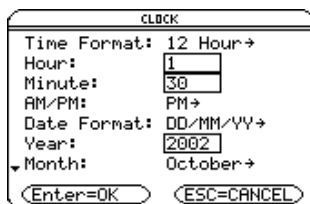
按键

结果

⏴ ⏵



光标移至 **October** ,  
然后按 **[ENTER]** 键

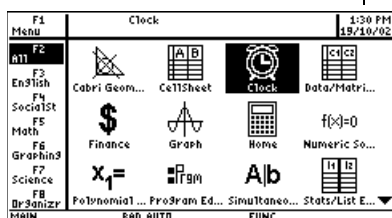


⏴ 19



**[ENTER]** **[ENTER]**

修改后的时间和日期



## 关闭时钟

从 Apps 桌面，打开 **CLOCK** 对话框，并在时钟 (Clock) 域选择 “关 (OFF)”。

例子: 关闭时钟。

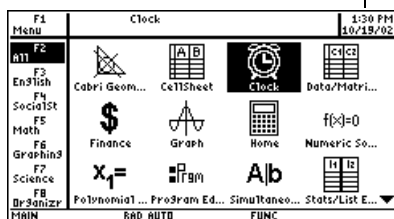
按键

结果

用光标键高亮以下图标:

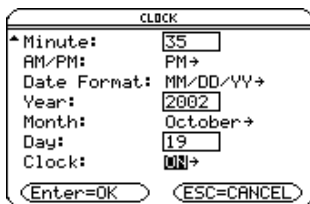


Clock on

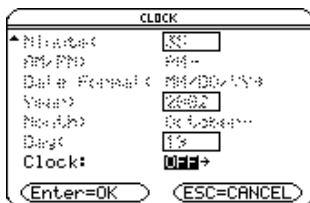


ENTER

光标移至 Clock 域

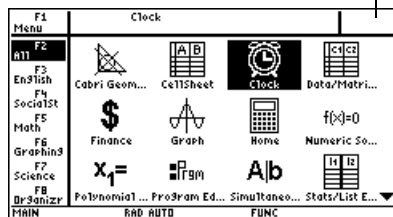


⬅️ ➡️ ENTER



ENTER

Clock off



要打开时钟, 可重复以上操作, 在 Clock 域选择“开 (ON)”。请记住重设时间和日期。



## 菜单的使用

TI-89 Titanium 或 Voyage™ 200 大多数菜单的选择，都可以通过功能键实现。功能键与计算器主 (Home) 屏幕和大多数应用软件屏幕顶部的工具条菜单相对应。使用按键命令选择其他菜单。





### 工具条菜单

作为 TI-89 Titanium 或 Voyage™ 200 数学运算的第一站，计算器主屏幕会显示工具条，让您选择与数学相关的选项。

工具条菜单也会在大多数应用软件屏幕出现，这些菜单列出了激活的应用软件的常用功能。

### 其他菜单



使用按键命令选择以下菜单。这些菜单具有相同的选项，与显示的屏幕或激活的应用软件无关。



按	来显示
 [CHAR]	字符 (CHAR) 菜单。列出键盘上没有的字符；字符按类别管理 (希腊、数学、标点符号、特殊字符和国际字符)。
 [MATH]	数学 (MATH) 菜单。按类别列出数学运算命令。
 [APPS]	应用软件 (APPLICATIONS) 菜单。列出已安装的应用软件。(菜单只有在 Apps 桌面关闭时才能使用；应用软件通常从 Apps 桌面访问。)
 [APPS]	FLASH APPLICATIONS 菜单。列出已安装的 Flash 应用软件。(菜单只有在 Apps 桌面关闭时才能使用；应用软件通常从 Apps 桌面访问。)

### 选择菜单选项

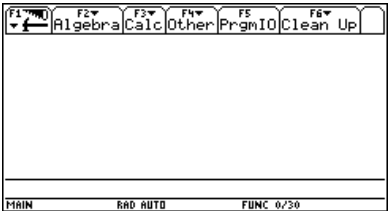
- 单击您要选择的选项左边的数字或字母。


— 或 —

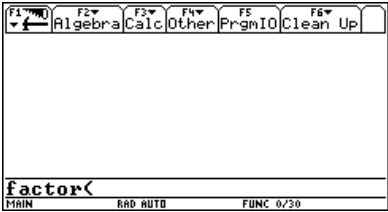
- 按  或  键选定选项，然后按 。

**注：**如果光标停留在菜单上的第一个选项，可以按  选择最后一个选项。如果光标停留在最后一个选项，可按  键选定第一个选项。

**例子：** 从计算器主屏幕代数 (Algebra) 菜单中选择 **factor(**。

按键	结果
按： TI-89 Titanium: <b>HOME</b> Voyage™ 200: <b>◀</b> <b>[CALC HOME]</b> - 或 - 在 Apps 桌面用光标键使  Home 呈高亮显示并按 <b>[ENTER]</b> 键	

<b>[F2]</b>	 <p>▼ 表示按下 <b>[F2]</b> 代数菜单会打开。</p>
-------------	--


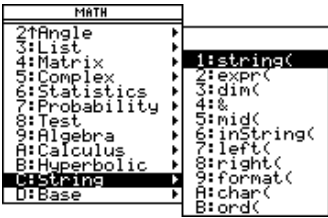
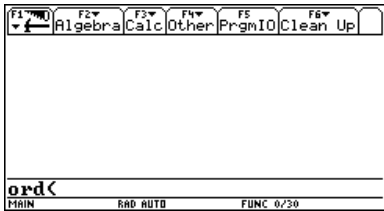
<b>2</b> - 或 - <b>◀</b> <b>[ENTER]</b>	
--	--

## 选择子菜单选项

菜单选项右侧的小箭头 (▶) 提示该菜单包含有子菜单。

<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">MATH</th> </tr> </thead> <tbody> <tr><td>1: Number</td><td>▶</td></tr> <tr><td>2: Angle</td><td>▶</td></tr> <tr><td>3: List</td><td>▶</td></tr> <tr><td>4: Matrix</td><td>▶</td></tr> <tr><td>5: Complex</td><td>▶</td></tr> <tr><td>6: Statistics</td><td>▶</td></tr> <tr><td>7: Probability</td><td>▶</td></tr> <tr><td>8: Test</td><td>▶</td></tr> <tr><td>9: Algebra</td><td>▶</td></tr> <tr><td>A: Calculus</td><td>▶</td></tr> <tr><td>B: Hyperbolic</td><td>▶</td></tr> <tr><td>C: String</td><td>▶</td></tr> </tbody> </table>	MATH		1: Number	▶	2: Angle	▶	3: List	▶	4: Matrix	▶	5: Complex	▶	6: Statistics	▶	7: Probability	▶	8: Test	▶	9: Algebra	▶	A: Calculus	▶	B: Hyperbolic	▶	C: String	▶	<table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr><td>1: seq(</td><td>▶</td></tr> <tr><td>2: min(</td><td>▶</td></tr> <tr><td>3: max(</td><td>▶</td></tr> <tr><td>4: SortA</td><td>▶</td></tr> <tr><td>5: SortD</td><td>▶</td></tr> <tr><td>6: sum(</td><td>▶</td></tr> <tr><td>7: cumSum(</td><td>▶</td></tr> <tr><td>8: product(</td><td>▶</td></tr> <tr><td>9: left(</td><td>▶</td></tr> <tr><td>A: mid(</td><td>▶</td></tr> <tr><td>B: right(</td><td>▶</td></tr> <tr><td>C: List(</td><td>▶</td></tr> </tbody> </table> <p>↓ 指向另外的选项。</p>	1: seq(	▶	2: min(	▶	3: max(	▶	4: SortA	▶	5: SortD	▶	6: sum(	▶	7: cumSum(	▶	8: product(	▶	9: left(	▶	A: mid(	▶	B: right(	▶	C: List(	▶
MATH																																																			
1: Number	▶																																																		
2: Angle	▶																																																		
3: List	▶																																																		
4: Matrix	▶																																																		
5: Complex	▶																																																		
6: Statistics	▶																																																		
7: Probability	▶																																																		
8: Test	▶																																																		
9: Algebra	▶																																																		
A: Calculus	▶																																																		
B: Hyperbolic	▶																																																		
C: String	▶																																																		
1: seq(	▶																																																		
2: min(	▶																																																		
3: max(	▶																																																		
4: SortA	▶																																																		
5: SortD	▶																																																		
6: sum(	▶																																																		
7: cumSum(	▶																																																		
8: product(	▶																																																		
9: left(	▶																																																		
A: mid(	▶																																																		
B: right(	▶																																																		
C: List(	▶																																																		

**例子：** 从计算器主屏幕数学 (MATH) 菜单中选择 **ord()**。


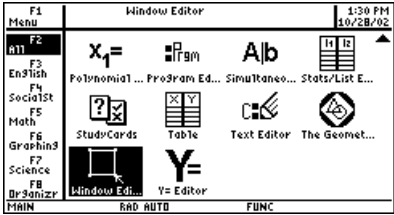
按键	结果
<b>[2nd] [MATH]</b>	
<b>C</b> - 或 - <b>[←] [→]</b>	
<b>B</b> - 或 - <b>[←] [ENTER]</b>	

## 使用对话框

菜单选项末尾的省略号 (...) 提示该选项包含有对话框。选定选项并按 **[ENTER]**。



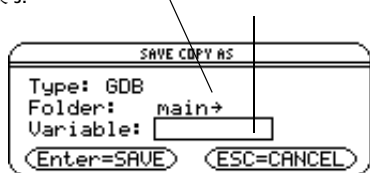
**例子：** 从窗口编辑器 (Window Editor) 选择 **SAVE COPY AS** 对话框。

按键	结果
<p>[APPS]</p> <p>用光标键使</p>  <p>Window Edi...</p> <p>呈高亮显示并按 [ENTER]</p>	

<p>[F1]</p>	
-------------	---

2  
- 或 -  
[ENTER] [ESC]

按 [F1] 键显示文件列表 输入变量名称表 s.



按两次 [ENTER] 保存，然后关闭对话框

**注：** 在大多数应用软件中按 [F1] S 快捷键也可以打开 **SAVE COPY AS** 对话框。

## 放弃菜单选择

如果要放弃选择某个菜单，可按 [ESC]。

## 在工具条菜单间移动

要在工具条菜单之间移动而不选择任何菜单选项：

- 按工具条菜单所示的功能键 (F1 到 F8)。
- 按一下功能键，然后按 [F1] 或 [F8] 将光标从一个工具条菜单移至下一个。在最后一个菜单上按 [F1] 可将光标移至第一个菜单，反之亦然。

**注：** 当含有子菜单的菜单选项被选定时，如果按 [F1]，就会显示其子菜单而不是下一个工具条菜单。再次按 [F1] 键可移至下一菜单。

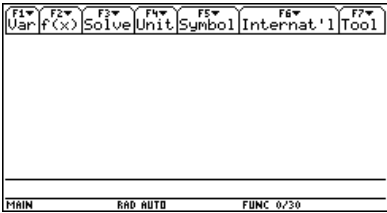
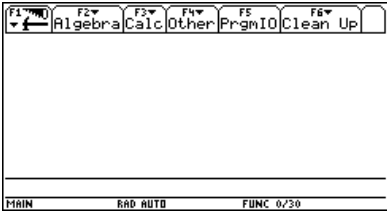
您可以了解更多关于菜单使用的更多信息。(请参阅电子文档中 *计算器操作 (Operating the Calculator)* 一章。)

### 自定义菜单:

自定义菜单可为常用选项提供快速访问途径。采用缺省自定义菜单或通过程序编辑器 (Program Editor) 来创建您的自定义菜单。您可将 TI-89 Titanium 或 Voyage™ 200 的任何可用命令和字符都添加到自定义菜单中。

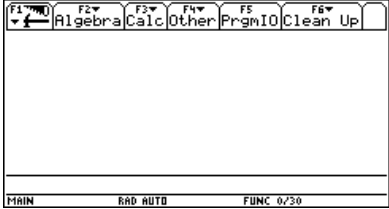

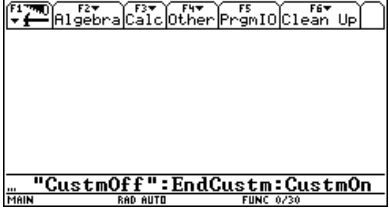
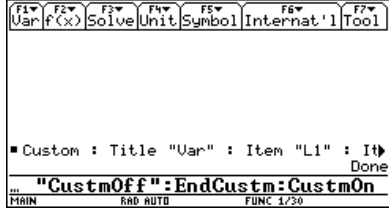
用自定义菜单代替计算器主屏幕上的标准工具条菜单 (有关创建自定义菜单的详细信息, 请参阅电子文档中 *编程 (Programming)* 一章)。您可以了解关于菜单使用的更多信息。(请参阅电子文档中 *计算器操作 (Operating the Calculator)* 一章。)

**例子:** 在计算器主屏幕上打开 / 关闭自定义菜单。

按键	结果
$\boxed{2nd}$ [CUSTOM]	缺省自定义菜单 
$\boxed{2nd}$ [CUSTOM]	标准工具条菜单 

**例子:** 还原缺省的自定义菜单。

**注：**还原缺省的自定义菜单会清除以前的自定义菜单。如果菜单是用程序创建的，那么您可以通过再次运行程序来重新恢复该菜单。

按键	结果
<p><b>2nd</b> [CUSTOM]</p> <p>(在标准工具条菜单上关闭和打开自定义菜单)</p>	
<p>TI-89 Titanium: <b>2nd</b> [F6]</p> <p>Voyage™ 200: [F6]</p>	
<p><b>3</b></p> <p>- 或 -</p> <p><b>ENTER</b></p>	
<p><b>ENTER</b></p>	

### 在 **Apps** 桌面关闭的情况下打开应用软件

如果您关闭了 **Apps** 桌面，可通过 **APPLICATIONS** 菜单打开应用软件。要在 **Apps** 桌面关闭的情况下打开 **APPLICATIONS** 菜单，按下 **[APPS]** 键。

**注：**如果在 **Apps** 桌面打开的情况下按下 **[APPS]** 键，出现的将是 **Apps** 桌面，而不是 **APPLICATIONS** 菜单。

**例子:** 在 Apps 桌面关闭的情况下, 从 APPLICATIONS 菜单打开 Window Editor。

按键	结果
<b>APPS</b>	
<b>3</b> - 或 - ← → <b>ENTER</b>	

要访问没有在 APPLICATIONS 菜单列出的应用软件, 请选择 **1:FlashApps** 。

## 使用屏幕拆分

TI-89 Titanium 或 Voyage™ 200 允许拆分屏幕来同时显示两个应用软件。例如, 同时显示 Y= Editor 和作图 (Graph) 屏幕, 以便将函数列表和作图方式进行比较。

### 设置屏幕拆分模式

您可以从模式 (MODE) 对话框将屏幕上下拆分或左右拆分。直至您改变屏幕拆分设置, 当前设置将一直保持有效。

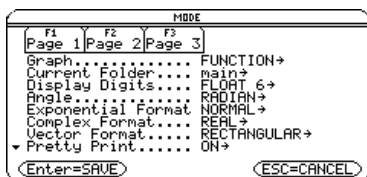
1. 按 **MODE** 键显示 MODE 对话框。
2. 按 **F2** 键显示屏幕拆分 (Split Screen) 模式设置。
3. 按 **↓** 键打开屏幕拆分 (Split Screen) 模式菜单。
4. 根据需要按 **↑** 键高亮 TOP-BOTTOM (上下) 或 LEFT-RIGHT (左右)。
5. 按 **ENTER** 键。Split Screen 模式设置显示出您选定的选项。

例子：将屏幕拆分模式设定为 TOP-BOTTOM。

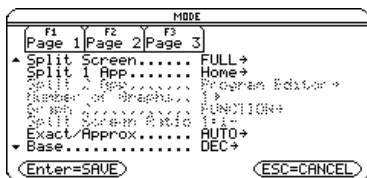
按键

结果

[MODE]



[F2]



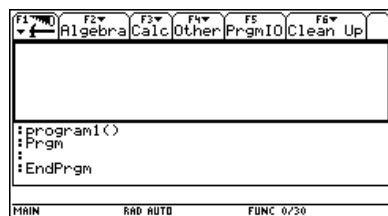
⏪ ⏩



[ENTER]



[ENTER]

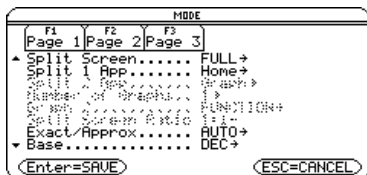




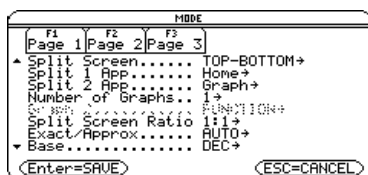
## 为拆分后的屏幕设定初始应用软件

在选择 TOP-BOTTOM 或 LEFT-RIGHT 屏幕拆分模式后，附带的一些模式设置变为可用。

全屏模式



屏幕拆分模式



模式	说明
屏幕 2 的应用软件	您可以为下半屏或右半屏（屏幕 2）指定应用软件，也可上半屏或左半屏（屏幕 1）指定应用软件。两个应用软件可同时运行。
图形数目	您可设定并显示两个互相独立的图形。
屏幕拆分比例	您可以调整两部分屏幕的相对比例。

要为每个屏幕设定初始应用软件：

1. 选择 **Split 1 App** 模式设置并按 **⏏** 键打开菜单，查看可用的应用软件。（参见第 41 页的“设置屏幕拆分模式”）
2. 按 **⏏** 和 **⏏** 使应用软件呈高亮显示，并按下 **[ENTER]** 键。
3. 重复上两步操作完成 **Split 2 App** 模式设置。

**例子：** 在上半屏显示 **Y= Editor**，在下半屏显示作图 (**Graph**) 应用软件。

按键	结果
<b>⏏ ⏏</b>	

2



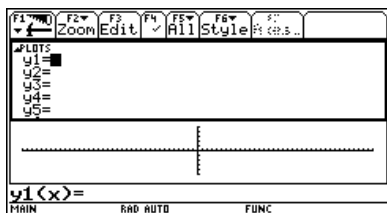
⏪ ⏩



4



ENTER



如果 Split 1 App 和 Split 2 App 设定为同一个非作图应用软件或 同一个图形数目 (Number of Graphs) 为 “1” 的作图应用软件，TI-89 Titanium 或 Voyage™ 200 将退出屏幕拆分模式，并以全屏模式显示该应用软件。

### 选择激活的应用软件

在屏幕拆分模式下，只允许有一个应用软件处于激活状态。

- 切换并激活另外一个应用软件，按 **2nd** [**☐**]。
- 要打开第三个应用软件，按下 **APPS** 键，并选择应用软件。该软件将取代拆分屏幕中激活的应用软件。

### 退出屏幕拆分模式

以下列任一方式退出屏幕拆分模式：

- 按 **[2nd] [QUIT]** 来关闭激活的应用软件，并以全屏模式显示其它打开的应用软件。
- 如果 Apps 桌面关闭，按 **[2nd] [QUIT]** 将处于激活状态的应用软件替换为计算器主 (Home) 屏幕。再次按 **[2nd] [QUIT]** 关闭屏幕拆分模式，并以全屏模式显示计算器 Home 屏幕。
- 在模式 (MODE) 对话框的第 2 页选择屏幕拆分 (Split Screen)，将屏幕拆分模式设定为 FULL (全屏)，按下 **[ENTER]** 键。
- 按 **[2nd] [QUIT]** 两次，显示 Apps 桌面。

您可以了解关于屏幕拆分的更多信息。(要了解更多信息，可参阅电子文档中 *屏幕拆分 (Split Screens)* 一章)

## 应用软件和操作系统 (OS) 版本管理

通过 TI-89 Titanium 或 Voyage™ 200 的连接功能，您可从通过如下方式下载应用软件：

- TI Educational & Productivity Solutions (E&PS) 网站：  
[education.ti.com/latest](http://education.ti.com/latest)
- TI-89 Titanium 或 Voyage™ 200 附带的光盘。
- 一台兼容的图形计算器。

在 TI-89 Titanium 或 Voyage™ 200 中安装新的应用软件就象在计算机里安装软件一样方便。您只要用 TI Connect™ 软件和 TI-89 Titanium 随机附带的 USB 电缆或 Voyage™ 200 随机附带的 USB Silver Edition 电缆即可实现安装。

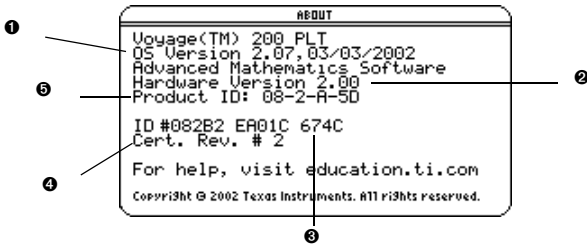
有关实现与兼容计算器相链接，以及下载 TI Connect 软件、应用软件和 OS 版本的系统要求和方法，请访问 TI E&PS 网站。

将应用软件下载至 TI-89 Titanium 或 Voyage™ 200 之前，请先阅读光盘或 TI 网站上的许可协议。

### 寻找 OS 版本和识别码 (ID 号)

如果您使从 TI E&PS 网站或致电客户支持中心，会要求提供您的 TI-89 Titanium 或 Voyage™ 200 的有关信息。您可以在 ABOUT 屏幕找到这些信息。

要显示 ABOUT 屏幕，可从 Apps 桌面按 **[F1] 3:About**。ABOUT 屏幕会显示您的 TI-89 Titanium 或 Voyage™ 200 的有关信息 (如下图所示)：



❶ OS 版本

❷ 硬件版本

❸ 设备 ID (须通过其获得安装所购买应用软件之证书)。类似于序列号。记下并妥善保管 ID 号, 以备计算器丢失或被盗后之用。

❹ 应用软件证书修订码 (Cert. Rev.)

❺ 产品识别码 (Product ID)。类似于型号。

请注意, 您的屏幕显示可能与上面显示有所差异。

## 删除应用软件

删除 TI-89 Titanium 或 Voyage™ 200 中不需要的应用软件能为其他软件腾出空间。删除应用软件之前, 建议将其保存到计算机里, 以备日后重新安装之用。

1. 退出应用软件。
2. 按 **[2nd][VAR-LINK]** 显示 VAR-LINK (All) 屏幕。
3. 按 **[2nd][F7]** (TI-89 Titanium) 或 **[F7]** 显示已安装的应用软件列表。
6. 按 **[ENTER]** 键删除应用软件。

**注:** 只有 Flash 应用软件才可被删除。

## 将 TI-89 Titanium 或 Voyage™ 200 连接到另一台设备

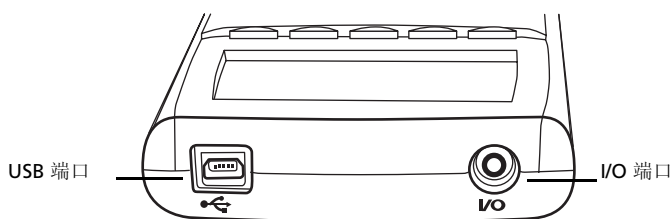
TI-89 Titanium 带有一个迷你 USB 端口和一个标准输入 / 输出 (I/O) 端口。Voyage™ 200 带有一个标准 I/O 端口。这些端口用于链接两台兼容的图形计算器或连接到计算器或外围设备。

另外, 教师专用 TI-89 Titanium 还带有一个附件端口, 而任何 Voyage™ 200 都带有一个附件端口。该端口用于视频数据输出, 以便让教室里所有人都能通过视频设备或投影仪来观看计算器屏幕的显示内容。

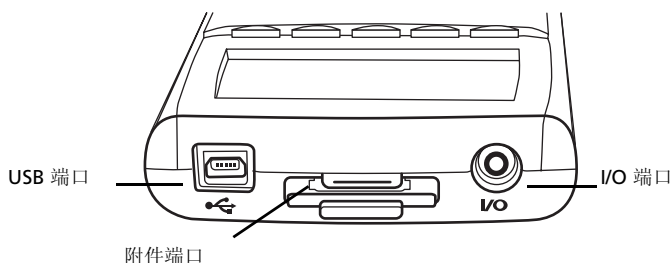
**将您的计算器连接到计算机** – 通过 USB 端口和附带的 USB 电缆实现 TI-89 Titanium 与计算机的连接。通过 I/O 端口和附带的 USB Silver Edition 电缆实现 Voyage™ 200 与计算机的连接。

**将您的计算器连接到另一台计算器** – 通过机对机 USB 连接电缆 (USB unit-to-unit cable) 或 机对机 I/O 连接电缆 (I/O unit-to-unit cable) 将一台 TI-89 Titanium 或 Voyage™ 200 连接到兼容的图形计算器或连接到计算器或外围设备，比如：TI-89 或 TI-92 Plus 图形计算器或 CBL 2™ 和 CBR™ 系统。

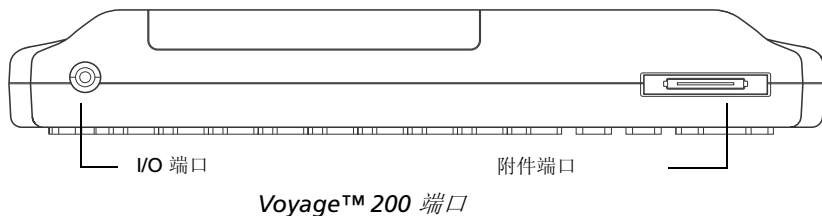
**向教室里所有人演示您的计算器屏幕内容** – 使用附件端口将 TI-Presenter™ 视频适配器连接到教师专用 TI-89 Titanium 或 Voyage™ 200。TI-Presenter™ 视频适配器在计算器和视频显示或录制设备之间提供视频接口。也可以使用附件端口将 TI ViewScreen™ 投影面板与计算器相连接。TI ViewScreen™ 投影面板将计算器屏幕显示放大并投影出来，让教师里所有人都能看清楚。要了解更多关于 TI-Presenter™ 适配器和 TI ViewScreen™ 投影面板的信息，请访问 TI E&PS 网站 [education.ti.com](http://education.ti.com)。



*TI-89 Titanium 端口*



*TI-89 Titanium 端口 (教师用机)*

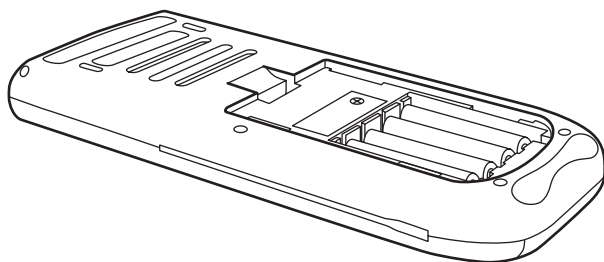


## 电池

TI-89 Titanium使用四节AAA碱性电池一节备用氧化银电池(SR445W 或303)。Voyage™ 200 使用四节 AAA 碱性电池和一节锂电池 (CR1616 或 CR1620)。备用电池已安装妥当，AAA 电池随产品提供。

### 安装 AAA 电池

1. 取下计算器背面的电池盖。
2. 根据电池槽上的正 (+) 负 (-) 极指示，将四节 AAA 电池装入电池槽内。



3. 盖上计算器的电池盖。注意：盖子要扣紧机身。

### 更换 AAA 碱性电池

由于电量不足，显示屏会开始变暗，尤其是在计算过程中。如果您发现需要不断调高对比度才能看清屏幕的话，请更换碱性电池。

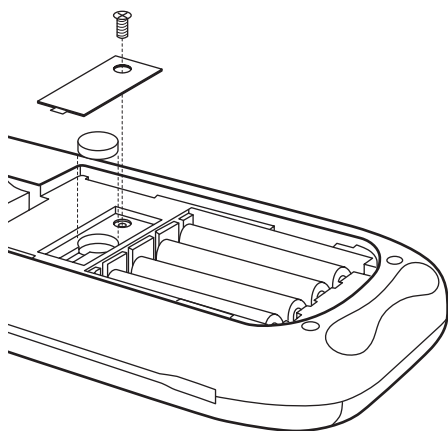
状态栏会显示电池信息。

标记	含义
<b>BATT</b>	电量不足。
<b>BATT</b>	尽快更换电池。

更换电池前，请按 **[2nd] [OFF]** 先关闭 TI-89 Titanium 或 Voyage™ 200，以免丢失储存在内存中的信息。请不要将备份电池与 AAA 碱性电池同时取下。

## 更换备份氧化银电池

1. 要更换备份氧化银电池，需取下电池盖，并拧下用于固定备用电池盖的小螺丝。



2. 取出旧电池并安装新的 SR445W 或 303 电池，正极 (+) 朝上。最后盖上备用电池盖并拧紧螺丝。

## OS 下载重要信息

开始下载 OS 之前先装上新电池。

在 OS 下载模式下，APD™ 功能不会启动。但是，如果您在正式开始下载前让计算器处于下载模式状态的时间过长，电池电量就可能被耗尽。这样，您就需要在开始下载前用新电池更换被耗尽的电池。

您也可以将 OS 通过机对机 USB 连接电缆 (USB unit-to-unit cable) 传送到另一台 TI-89 Titanium 或通过机对机 I/O 连接电缆 (unit-to-unit cable) 传送到另一台 Voyage™ 200。如果您在传送过程中无意间中断了传送，您就需要通过计算机来重装 OS 了。再次提醒：在开始下载前切记安装新电池。

如果您遇到问题，请根据服务和支持 (Service & Support) 信息联系德州仪器公司 (Texas Instruments)。

## 电池预防措施

更换电池时请采取如下预防措施：

- 请将电池放在小孩接触不到的地方。
- 请勿将新旧电池混合使用。请勿将不同品牌电池混合使用。请勿将不同型号的电池混合使用。

- 请勿将充电电池和非充电电池混合使用。
- 请按照电池正负极 (+/-) 标记安装电池。
- 请勿将非充电电池放在充电器中。
- 请采用正确的办法立刻处理用过的旧电池。
- 请勿焚烧或拆毁电池。



## 使用预览


### 执行运算


本节为您提供了几个在计算器主屏幕上进行操作的例子，来演示 TI-89 Titanium / Voyage™ 200 的计算功能。在每个例子执行之前，可按下按键 **F1** 并选择 **8:Clear Home** 清空屏幕，使之仅显示该例的键击结果。

### 运算演示

#### 步骤和键击

计算  $\sin(\pi/4)$  并以符号和数字格式显示结果。欲清空前次运算历史信息，可按下按键 **F1** 并选取 **8:Clear Home**。

 **2nd** **[SIN]** **2nd** **[π]** **÷** **4** **)** **ENTER** **♦** **[≈]**

 **[SIN]** **2nd** **[π]** **÷** **4** **)** **ENTER** **♦** **[≈]**

#### 显示

F1 Tools	F2 13abrd	F3 Calc	F4 Other	F5 Pr3rd	F6 Clean Up
■ $\sin\left(\frac{\pi}{4}\right)$					$\frac{\sqrt{2}}{2}$
■ $\sin\left(\frac{\pi}{4}\right)$					.707107
■ $\sin(\pi/4)$					
MAIN					RAD AUTO FUNC 2/30

### 求阶乘


#### 步骤和键击

通过计算几个数字的阶乘来演示 TI-89 Titanium / Voyage™ 200 如何处理超大整数的运算。欲选取阶乘算子 (!)，可按下 **2nd** **[MATH]**，选取

**7:Probability**，然后选取 **1:!**。

 **5** **2nd** **[MATH]** **7** **1** **ENTER** **20** **2nd** **[MATH]** **7** **1** **ENTER**

**30** **2nd** **[MATH]** **7** **1** **ENTER**

 **5** **2nd** **[!]** **ENTER** **20** **2nd** **[!]** **ENTER** **30** **2nd** **[!]** **ENTER**

#### 显示

F1 Tools	F2 13abrd	F3 Calc	F4 Other	F5 Pr3rd	F6 Clean Up
■ 5!					120
■ 20!					2432902008176640000
■ 30!					265252859812191058636308
30!					
MAIN					RAD AUTO FUNC 2/30

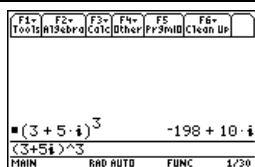
## 复数展开

### 步骤和键击

通过计算  $(3+5i)^3$  来演示 TI-89 Titanium / Voyage™ 200 如何处理涉及复数的运算。

按下  $\boxed{3} \boxed{+} \boxed{5} \boxed{2nd} \boxed{[i]} \boxed{)} \boxed{^} \boxed{3} \boxed{ENTER}$

### 显示



## 求质数因子

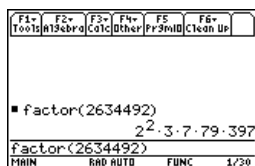
### 步骤和键击

计算有理数 2634492 的因子。您可以通过键盘在输入行键入 **FACTOR**，或是按下按键  $\boxed{F2}$  并选取 **2:factor()**。

按下  $\boxed{F2} \boxed{2} \boxed{2634492} \boxed{)} \boxed{ENTER}$

(可选操作) 您可以任意输入其它数字。

### 显示



## 展开表达式

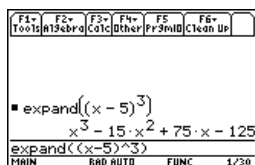
### 步骤和键击

展开表达式  $(x-5)^3$ 。您可以通过键盘在输入行键入 **EXPAND**，或是按下按键  $\boxed{F2}$  并选取 **3:expand()**。

按下  $\boxed{F2} \boxed{3} \boxed{(} \boxed{x} \boxed{-} \boxed{5} \boxed{)} \boxed{^} \boxed{3} \boxed{)} \boxed{ENTER}$

(可选操作) 您可以任意输入其它表达式。

### 显示



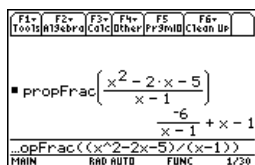
## 化简表达式

### 步骤和键击

将表达式  $(x^2-2x-5)/(x-1)$  化简为最简形式。您可以通过键盘在输入行键入 **PROPFrac**，或是按下按键  $\boxed{F2}$  并选取 **7:propFrac()**。

按下  $\boxed{F2} \boxed{7} \boxed{(} \boxed{x} \boxed{^} \boxed{2} \boxed{-} \boxed{2} \boxed{x} \boxed{-} \boxed{5} \boxed{)} \boxed{\div} \boxed{(} \boxed{x} \boxed{-} \boxed{1} \boxed{)} \boxed{)} \boxed{ENTER}$

### 显示



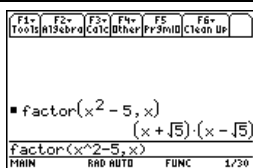
## 多项式的因式分解

### 步骤和键击

将多项式  $(x^2-5)$  按  $x$  进行因式分解。您可以通过键盘在输入行中键入 **FACTOR**，或是按下按键 **F2** 并选取 **2:factor()**。

按下 **F2**  $2 \times \wedge 2 - 5$   $,$   $x$  **ENTER**

### 显示



## 解方程

### 步骤和键击

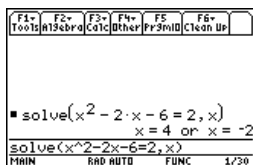
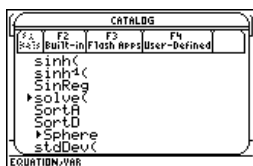
求解关于  $x$  的方程  $x^2-2x-6=2$ 。

您可以从目录菜单选取 **"solve("** 或通过键盘在输入行键入 **SOLVE(**，或是按下按键 **"F2"** 并选取 **1:solve()**。

状态行将显示 **Catalog** 菜单中标记项所需的句法。

按下 **F2**  $1 \times \wedge 2 - 2 \times - 6 = 2$   $,$   $x$  **ENTER**

### 显示



## 求解有定义域约束的方程

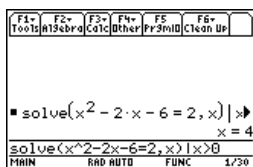
### 步骤和键击

求解关于  $x$  的方程  $x^2-2x-6=2$ ，其中  $x$  大于零。算子 **"with" (|)** 确定了定义域约束。

**F2**  $1 \times \wedge 2 - 2 \times - 6 = 2$   $,$   $x$  **|**  $1 \times$  **2nd** **[>]** **0** **ENTER**

**F2**  $1 \times \wedge 2 - 2 \times - 6 = 2$   $,$   $x$  **2nd** **[|]**  $1 \times$  **2nd** **[>]** **0** **ENTER**

### 显示



## 求函数的导数

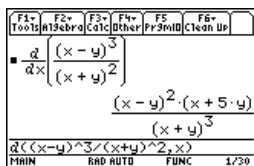
### 步骤和键击

求  $(x-y)^3/(x+y)^2$  关于  $x$  的导数。

本例演示了微积分中的微分函数运用，以及该函数是如何在历史区域中以“清楚显示 (Pretty Print)”方式显示。

按下  $\text{2nd}$   $[d]$   $(\text{X} - \text{Y}) \wedge 3 \div (\text{X} + \text{Y}) \wedge 2$   
 $\text{X}$   $\text{ENTER}$

### 显示



## 求函数的积分

### 步骤和键击

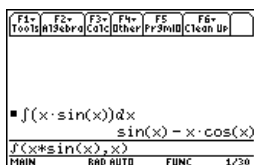
求  $x \cdot \sin(x)$  关于  $x$  的积分。

本例演示了微积分中积分函数的运用。

$\text{2nd}$   $[j]$   $\text{X}$   $\times$   $\text{2nd}$   $[\sin]$   $\text{X}$   $\text{X}$   $\text{X}$   $\text{ENTER}$

$\text{2nd}$   $[j]$   $\text{X}$   $\times$   $[\sin]$   $\text{X}$   $\text{X}$   $\text{X}$   $\text{ENTER}$

### 显示



## 符号运算

求解方程组  $2x - 3y = 4$  和  $-x + 7y = -12$ 。解第一个方程，将  $x$  表示成关于  $y$  的表达式。将  $x$  代入第二个方程，求出  $y$  值。然后将  $y$  值代回到第一个方程，从而求出  $x$  的值。

### 步骤和键击

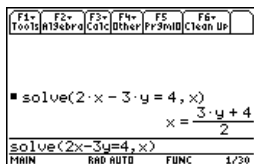
1. 显示主屏幕并清空输入行。求解方程  $2x - 3y = 4$  中的变量  $x$ 。

从 Algebra 菜单，按  $\text{F2}$  1 选取 **solve**。您也可以使用键盘直接键入或从 **Catalog** 中选取 **solve**。

$\text{HOME}$   $\text{CLEAR}$   $\text{CLEAR}$   $\text{F2}$  1 2  $\text{X}$   $-$  3  $\text{Y}$   $=$  4  
 $\text{X}$   $\text{ENTER}$

$\text{CALC HOME}$   $\text{CLEAR}$   $\text{CLEAR}$   $\text{F2}$  1 2  $\text{X}$   $-$   
3  $\text{Y}$   $=$  4  $\text{X}$   $\text{ENTER}$

### 显示



2. 开始求解方程  $-x + 7y = -12$  中的  $y$  值，但不要按下  $\text{ENTER}$ 。

按下  $\text{F2}$  1  $(-)$   $\text{X}$   $+$  7  $\text{Y}$   $=$   $(-)$  12  $\text{Y}$

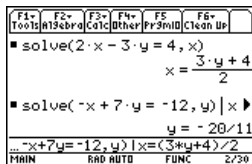
## 步骤和键击

## 显示

3. 使用“with”算子替代从第一个方程中解出的  $x$  的表达式。从而得到  $y$  值。

“with”算子在屏幕显示为 |。

使用自动粘贴功能突出显示历史区域中的前次运算答案，并将其粘贴到输入行中。

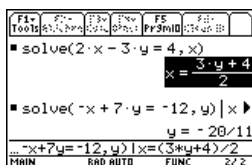


[1] [ENTER] [ENTER]

[2nd] [1] [ENTER] [ENTER]

4. 在历史区域中突出显示  $x$  的表达式。

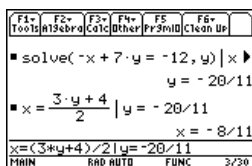
按下



5. 高亮的表达式被自动粘贴到输入行中。然后替换在第二个方程式中解出的  $y$  值。

[ENTER] [1] [ENTER] [ENTER]

[ENTER] [2nd] [1] [ENTER] [ENTER]



方程组的解为：

$x = -8/11$  和  $y = -20/11$

本例演示了如何进行符号运算。也可以运用一步求解功能来解方程组。

## 常数和度量单位

运用公式  $f = m \cdot a$ ，计算当  $m = 5$  千克和  $a = 20$  米/秒<sup>2</sup> 时的力的大小。问当  $a = 9.8$  米/秒<sup>2</sup> 时，力的大小又是多少？（9.8 米/秒<sup>2</sup> 是重力加速度，记作常数  $g$ ）。请将计算结果的单位从牛顿换算成千克力。

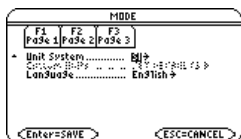
## 步骤和键击

## 显示

1. 显示屏幕上第 3 页上的 **MODE** 对话框。对于 **Unit System** 模式，选取 **SI** 作为公制度量单位。

结果显示采用这些默认的度量单位。

按下 [MODE] [F3] [1] [ENTER]

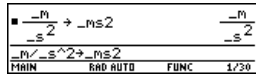


2. 创建以米 / 秒<sup>2</sup>表示的加速度单位，记作  $\_ms2$ 。

您可以使用 **UNITS** 对话框从按字母顺序排列的分类列表中选择度量单位。您可以使用  $\boxed{2nd} \odot$  和  $\boxed{2nd} \ominus$  在所有分类列表中一次滚动一个页面。



如果您使用 **UNITS** 对话框选取某一度量单位，前缀 “ $\_$ ” 将自动被输入。现在，您无须在每次都重新输入  $\_m/s^2$ ，可以用  $\_ms2$  表示。或者，您也可以使用 **UNITS** 对话框从 **Acceleration** 类中选取  $\_ms2$ 。

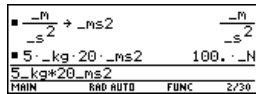


$\boxed{2nd} \boxed{[UNITS]} \odot \boxed{M} \boxed{ENTER} \boxed{\div} \boxed{2nd} \boxed{[UNITS]} \odot \ominus \ominus \ominus \ominus \boxed{S} \boxed{ENTER} \boxed{\wedge} \boxed{2} \boxed{STO} \boxed{\blacktriangleright} \boxed{\blacklozenge} \boxed{[-]} \boxed{2nd} \boxed{[a-lock]} \boxed{MS} \boxed{alpha} \boxed{2} \boxed{ENTER}$

$\boxed{2nd} \boxed{[UNITS]} \odot \boxed{M} \boxed{ENTER} \boxed{\div} \boxed{2nd} \boxed{[UNITS]} \odot \ominus \ominus \ominus \ominus \boxed{S} \boxed{ENTER} \boxed{\wedge} \boxed{2} \boxed{STO} \boxed{2nd} \boxed{[-]} \boxed{MS2} \boxed{ENTER}$

3. 计算当  $m = 5$  千克 ( $\_kg$ ) 且  $a = 20$  米 / 秒<sup>2</sup> ( $\_ms2$ ) 时的力的大小。

如果您知道某个度量单位的缩写，也可以直接从键盘上输入。

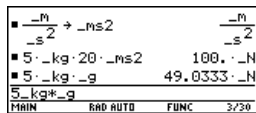


$\boxed{5} \boxed{\blacklozenge} \boxed{[-]} \boxed{2nd} \boxed{[a-lock]} \boxed{KG} \boxed{alpha} \boxed{\times} \boxed{20} \boxed{\blacklozenge} \boxed{[-]} \boxed{2nd} \boxed{[a-lock]} \boxed{MS} \boxed{alpha} \boxed{2} \boxed{ENTER}$

$\boxed{5} \boxed{2nd} \boxed{[-]} \boxed{KG} \boxed{\times} \boxed{20} \boxed{2nd} \boxed{[-]} \boxed{MS2} \boxed{ENTER}$

4. 使用相同的  $m$  值，计算产生重力加速度（常数  $\_g$ ）时的力的大小。

对于  $\_g$ ，您可以使用 **UNITS** 对话框中预先定义的常数或是键入  $\_g$ 。



$\boxed{5} \boxed{\blacklozenge} \boxed{[-]} \boxed{2nd} \boxed{[a-lock]} \boxed{KG} \boxed{alpha} \boxed{\times} \boxed{2nd} \boxed{[UNITS]} \odot \boxed{alpha} \boxed{G} \boxed{ENTER} \boxed{ENTER}$

$\boxed{5} \boxed{2nd} \boxed{[-]} \boxed{KG} \boxed{\times} \boxed{\blacklozenge} \boxed{[UNITS]} \odot \boxed{G} \boxed{ENTER} \boxed{ENTER}$

## 步骤和键击

## 显示

5. 将结果的度量单位换算成千克力 (kgf)。

[2nd] [▶] 可显示 ▶ 换算因子。

[▶] [2nd] [▶] [◀] [2nd] [a-lock] KGF [alpha]  
[ENTER]

[▶] [2nd] [▶] [2nd] [◀] KGF [ENTER]

■	$\frac{m}{s^2}$	$\rightarrow$	$\frac{m}{s^2}$	$\frac{m}{s^2}$
■	$\frac{kg}{s^2}$	$\rightarrow$	$\frac{kg}{s^2}$	$\frac{kg}{s^2}$
■	$5 \cdot \frac{kg}{s^2} \cdot 20 \cdot \frac{m}{s^2}$	$\rightarrow$	$100 \cdot \frac{N}{s^2}$	$\frac{N}{s^2}$
■	$5 \cdot \frac{kg}{s^2} \cdot g$	$\rightarrow$	$49.0333 \cdot \frac{N}{s^2}$	$\frac{N}{s^2}$
■	$5 \cdot \frac{kg}{s^2} \cdot g \rightarrow \frac{kgf}{s^2}$	$\rightarrow$	$5 \cdot \frac{kgf}{s^2}$	$\frac{kgf}{s^2}$
■	$5 \cdot \frac{kgf}{s^2} \cdot g$	$\rightarrow$	$\frac{kgf}{s^2}$	$\frac{kgf}{s^2}$
MAIN	RAD	AUTO	FUNC	4/20

## 基本函数作图 I

本例演示了 TI-89 Titanium / Voyage™ 200 的作图能力，教您如何使用 **Y= Editor** 进行函数作图。您将了解如何输入一个函数、生成函数图形、描绘曲线、找出最小值点，并将最小值点的坐标传送到主屏幕上。

通过绘制函数  $y=|x^2-3|-10/2$  的图形，掌握 TI-89 Titanium / Voyage™ 200 的作图能力。

## 步骤和键击

## 显示

1. 显示 **Y= Editor**。

按下 [▶] [Y=]

F1=	F2=	F3=	F4=	F5=	F6=	F7=
Tools	Zoom	Edit	✓	All	Style	Plot
+PLT5						
y1=						
y2=						
y3=						
y4=						
y5=						
y6=						
y7=						
y1(x)=						
y2=						
y3=						
y4=						
y5=						
y6=						
y7=						
MAIN	RAD	AUTO	FUNC			

2. 输入函数  $(\text{abs}(x^2-3)-10)/2$ 。

屏幕上以“清楚显示”方式显示 **y1=**。

[▶] [CATALOG] A [ENTER] X [^] 2 [ ] 3 [ ] - 10  
[ ] ÷ 2 [ENTER]

[▶] [2nd] [CATALOG] A [ENTER] X [^] 2 [ ] 3 [ ]  
- 10 [ ] ÷ 2 [ENTER]

F1=	F2=	F3=	F4=	F5=	F6=	F7=
Tools	Zoom	Edit	✓	All	Style	Plot
+PLT5						
Plot 1:						
y1=						
$ x^2-3 -10$						
2						
y2=						
y3=						
y4=						
y1(x)=(abs(x^2-3)-10)/2						
y2=						
y3=						
y4=						
y5=						
y6=						
y7=						
MAIN	RAD	AUTO	FUNC			

3. 显示函数图形。

按下 **6** 或是将光标移动到 **6:ZoomStd** 后按下 [ENTER] 来选取 **6:ZoomStd**。

按下 [F2] 6

F1=	F2=	F3=	F4=	F5=	F6=	F7=
Tools	Zoom	Trace	Graph	Math	Draw	Pen-C
+PLT5						
Plot 1:						
y1=						
$ x^2-3 -10$						
2						
y2=						
y3=						
y4=						
y5=						
y6=						
y7=						
y1(x)=(abs(x^2-3)-10)/2						
y2=						
y3=						
y4=						
y5=						
y6=						
y7=						
MAIN	RAD	AUTO	FUNC			

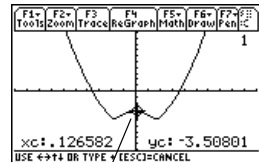
## 步骤和键击

## 显示

4. 开启 **Trace** 功能。

屏幕上出现显迹光标、以及  $x$  轴和  $y$  轴的坐标。

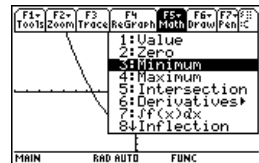
按下 **[F3]**



显迹光标

5. 打开 **MATH** 菜单并选取 **3:Minimum**。

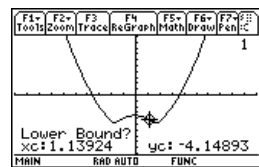
按下 **[F5]** **[ $\odot$ ]** **[ $\odot$ ]** **[ENTER]**



6. 设定下界。

在再次按下 **[ENTER]** 之前，按住 **[ $\odot$ ]** (右光标键) 将显迹光标移动到左下界，此时其  $x$  坐标位置正好在最小值结点的左侧。

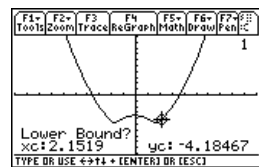
按下 **[ $\odot$ ]** ... **[ $\odot$ ]** **[ENTER]**



7. 设定上界。

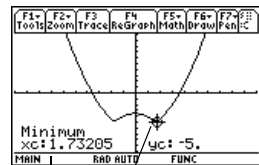
按住 **[ $\odot$ ]** 键将显迹光标移动到上界，此时其  $x$  坐标位置正好在最小值结点的右侧。

按下 **[ $\odot$ ]** ... **[ $\odot$ ]**



8. 在图形的下界和上界间找到最小值点。

按下 **[ENTER]**

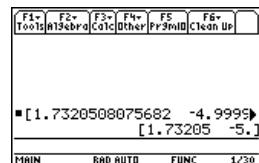


最小值点  
最小值点坐标

9. 将结果传送并显示在主屏幕上。

**[ $\odot$ ]** **[ $\blacklozenge$ ]** **[(-)]** **[HOME]**

**[ $\odot$ ]** **[H]** **[ $\blacklozenge$ ]** **[CALC HOME]**





## 基本函数作图 II

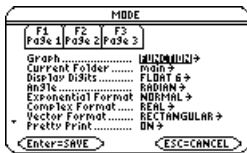
作出一个半径为 5、圆心位于坐标系原点的圆。使用标准观测窗 (**ZoomStd**) 来查看该圆。然后用 **ZoomSqr** 来调整观测窗。

### 步骤和键击

### 显示

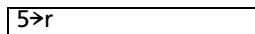
1. 显示 **MODE** 对话框。对于 **Graph** 模式，选取 **FUNCTION**。

按下 **[MODE]** **[1]** **[ENTER]**



2. 显示主屏幕。然后将半径值 5 存储在变量 r 中。

**[HOME]** **5** **[STO▶]** **[alpha]** **R** **[ENTER]**



**[CALC HOME]** **5** **[STO▶]** **R** **[ENTER]**

3. 显示并清空 **Y= Editor**。然后定义  $y_1(x) =$

$\sqrt{r^2 - x^2}$ ，即表示圆的上半部分的函数。

在进行函数作图时，必须分别定义上半圆和下半圆函数。

**[Y=]** **[F1]** **8** **[ENTER]** **[ENTER]** **[2nd]** **[√]** **[alpha]** **R**  
**[^]** **2** **[X]** **[^]** **2** **[)]** **[ENTER]**

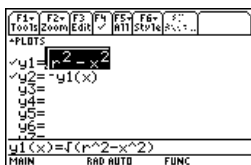
**[Y=]** **[F1]** **8** **[ENTER]** **[ENTER]** **[2nd]** **[√]** **R** **[^]** **2**  
**[X]** **[^]** **2** **[)]** **[ENTER]**

4. 定义  $y_2(x) = -\sqrt{r^2 - x^2}$ ，即表示下半圆函数。

下半圆函数是上半圆函数的负值，因此可以定义  $y_2(x) = -y_1(x)$ 。

请使用完整的函数名 **y1(x)**，而不是简单表示为 **y1**。

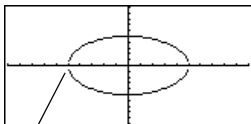
按下 **[ENTER]** **[(-)]** **Y 1** **[X]** **[ENTER]**



5. 选择 **ZoomStd** 观测窗，可以自动作出函数图形。

在标准观测窗中，**x** 和 **y** 轴的刻度范围都是从 -10 到 10。但是，此范围在 **x** 轴上的延伸度要比在 **y** 轴上大。因此，圆形看起来象椭圆。

按下 **[F2]** **6**



注意上下半圆结合处的微量间隙。

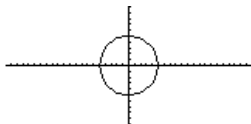
## 步骤和键击

## 显示

6. 选取 **ZoomSqr**。

**ZoomSqr** 可增大沿  $x$  轴的刻度范围，从而圆形和正方形都可以按正确的比例显示出来。

按下 **[F2] 5**



**注：**上半圆和下半圆之间存在间隙的原因是因为两个半圆函数各自独立。两个半圆的端点都是  $(-5,0)$  和  $(5,0)$ 。然而，由于观测窗的原因，每个半圆在屏幕上绘制出的端点及其精确点可能会有细微的差别。

## 参变量作图

根据表示以角  $(\theta)$  为  $60^\circ$ 、初速度  $(v_0)$  15米/秒踢出的球的轨迹的参数方程进行作图。重力常数  $g = 9.8$  米/秒<sup>2</sup>。若不计空气阻力和其它阻力，球可达到的最大高度及落地时间分别是多少？

## 步骤和键击

## 显示

1. 显示 **MODE** 对话框。对于 **Graph** 模式，选取 **PARAMETRIC**。

按下 **[MODE] 2 [ENTER]**



2. 显示并清空 **Y= Editor**。然后定义水平分量  $xt1(t) = v_0 t \cos \theta$ 。

输入  $v_0$  和  $\theta$  的值。

**[2nd] [Y=] [F1] 8 [ENTER] [ENTER] 15T [X] [2nd] [COS] 60 [2nd] [°] [ENTER]**

**[2nd] [Y=] [F1] 8 [ENTER] [ENTER] 15T [X] [COS] 60 [2nd] [°] [ENTER]**

$$xt1(t)=15t*\cos(60^\circ)$$

**TI-89 Titanium:** 键入 **T [X] [2nd] [COS]**，而不是 **T [2nd] [COS]**。

**Voyage™ 200:** 键入 **T [X] [COS]**，而不是 **T [COS]**。

键击 **[2nd] [°]** 或 **[2nd] [MATH] 2 1**，输入符号“°”。这样可确保无论是否在角度模式下，该数值均会被理解为角度值。

## 步骤和键击

3. 定义垂直分量  $yt_1(t) = v_0 t \sin \theta - (g/2)t^2$ 。

输入  $v_0$ 、 $\theta$  和  $g$  的值。

$\text{[ENTER]} 15 \text{[T]} \times \text{[2nd]} \text{[SIN]} 60 \text{[2nd]} \text{[°]} \text{[)]}$

$\text{[)]} 9.8 \text{[÷]} 2 \text{[)]} \text{[T]} \text{[^]} 2 \text{[ENTER]}$

$\text{[ENTER]} 15 \text{[T]} \times \text{[SIN]} 60 \text{[2nd]} \text{[°]} \text{[)]} \text{[-]} \text{[)]}$

$9.8 \text{[÷]} 2 \text{[)]} \text{[T]} \text{[^]} 2 \text{[ENTER]}$

## 显示



4. 显示 **Window Editor**。输入合适的窗口变量。

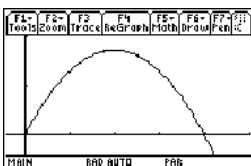
你可以按下  $\text{[◀]}$  或  $\text{[ENTER]}$  来输入数值并移动到下一个变量。

按下  $\text{[▼]} \text{[WINDOW]} 0 \text{[◀]} 3 \text{[◀]} .02 \text{[◀]} \text{[◀]} 2 \text{[◀]} 25 \text{[◀]} 5 \text{[◀]} \text{[◀]} 2 \text{[◀]} 10 \text{[◀]} 5$

```
tmin=0.
tmax=3.
tstep=.02
xmin=-2.
xmax=25.
xsc1=5.
ymin=-2.
ymax=10.
ysc1=5.
```

5. 根据表示球轨迹模型的参变量作图。

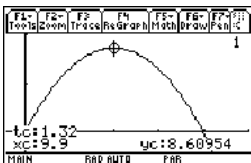
按下  $\text{[▼]} \text{[GRAPH]}$



6. 选取 **Trace**。然后沿轨迹移动光标来找到：

- 最大高度处的  $y$  值。
- 球落地时是  $t$  值。

在必要时按下  $\text{[F3]} \text{[▶]}$  或  $\text{[◀]}$



## 极坐标作图

极坐标方程  $r_1(\theta) = A \sin B \theta$  的图形构成玫瑰状。作出  $A=8$ ， $B=2.5$  时的玫瑰图形。然后改变  $A$  和  $B$  的值，观察玫瑰形的变化。

## 步骤和键击

1. 显示 **MODE** 对话框。对于 **Graph** 模式，选取 **POLAR**。对于 **Angle** 模式，选取 **RADIAN**。

按下  $\text{[MODE]} \text{[▶]} 3 \text{[▶]} \text{[▶]} \text{[▶]} \text{[▶]} 1 \text{[ENTER]}$

## 显示



2. 显示并清空 **Y= Editor**。然后定义极坐标方程  $r1(\theta) = A \sin B\theta$ 。

分别给 A 和 B 赋值 8 和 2.5。

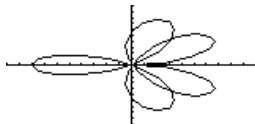
$\diamond$  [Y=] [F1] 8 [ENTER] [ENTER] 8 [2nd] [SIN] 2.5  $\diamond$  [ $\theta$ ] [ ] [ENTER]

$\diamond$  [Y=] [F1] 8 [ENTER] [ENTER] 8 [SIN] 2.5 [ $\theta$ ] [ ] [ENTER]



3. 选取 **ZoomStd** 观测窗，作出方程图形。

- 图形显示出仅有 5 片玫瑰花瓣。
  - 在标准观测窗中，Window 变量  $\theta_{max} = 2\pi$ 。其它花瓣的  $\theta$  值大于  $2\pi$ 。
- 玫瑰看上去并不对称。
  - 在标准观测窗中，x 和 y 轴的刻度范围都是从 -10 到 10。但是，此范围在 x 轴上的延伸度要比 y 轴大。因此，图形看起来象椭圆。



按下 [F2] 6

4. 显示 **Window Editor**，并将  $\theta_{max}$  改为  $4\pi$ 。

当退出 **Window Editor** 时， $4\pi$  将被替代其近似值。

按下  $\diamond$  [WINDOW]  $\odot$  4 [2nd] [ $\pi$ ]

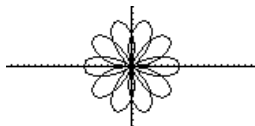
```

 $\theta_{min}=0.$ 
 $\theta_{max}=4\pi$ 
 $\theta_{step}=1.3089969389957$ 
 $x_{min}=-10.$ 
 $x_{max}=10.$ 
 $x_{scl}=1.$ 
 $y_{min}=-10.$ 
 $y_{max}=10.$ 
 $y_{scl}=1.$ 
    
```

5. 选取 **ZoomSqr**，作出方程的图形。

**ZoomSqr** 沿 x 轴增大刻度范围，从而图形可按正确的比例显示出来。

按下 [F2] 5



如有必要，可以改变 A 和 B 的值并重新绘制方程图形。

## 序列作图

一片树林有 4000 棵树。每年，有 20% 的树会被砍伐（其它 80% 保留），同时种植 1000 棵新树。使用序列方式来计算每年年底树林中树木的数量。树木数量能否保持稳定？

起始数	1 年后	2 年后	3 年后	...
4000	$.8 \times 4000 + 1000$	$.8 \times (.8 \times 4000 + 1000) + 1000$	$.8 \times (.8 \times (.8 \times 4000 + 1000) + 1000) + 1000$	...

### 步骤和键击

### 显示

1. 显示 **MODE** 对话框。对于 **Graph** 模式，选取 **SEQUENCE**。

按下 **[MODE]** **[4]** **[ENTER]**



2. 显示并清空 **Y= Editor**。然后定义序列为  $u1(n) = iPart(.8 * u1(n-1) + 1000)$ 。

使用 **iPart** 获取结果的整数部分。因为砍伐的树木数量不可能有小数。

要应用 **iPart**(，您可以使用 **[2nd]** **[MATH]**，并键入该函数即可，或是在 **CATALOG** 菜单上选取。

**[Y=]** **[F1]** **8** **[ENTER]** **[ENTER]** **[2nd]** **[MATH]** **14.8**  
**[alpha]** **U1** **[alpha]** **N** **[ ]** **1** **[ ]** **+** **1000** **[ ]**  
**[ENTER]**



**[Y=]** **[F1]** **8** **[ENTER]** **[ENTER]** **[2nd]** **[MATH]** **14.8**  
**U1** **[ ]** **N** **[ ]** **1** **[ ]** **+** **1000** **[ ]** **[ENTER]**

3. 将 **u1** 定义为初始值，作为数列的第一项。

按下 **[ENTER]** **4000** **[ENTER]**

4. 显示 **Window Editor**。设定 **n** 值和窗口变量。

**nmin=0** 和 **nmax=50** 用于计算 50 年后树林的规模。

按下 **[WINDOW]** **0** **[ ]** **50** **[ ]** **1** **[ ]** **1** **[ ]**

```
nmin=0
nmax=50
plotStart=1
plotStep=1
xmin=0
xmax=50
xsc1=10
ymin=0
ymax=6000
ysc1=1000
```

5. 设定与本例相对应的 **x**、**y** 窗口变量值。

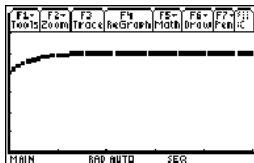
按下 **0** **[ ]** **50** **[ ]** **10** **[ ]** **0** **[ ]** **6000** **[ ]** **1000**

## 步骤和键击

## 显示

6. 显示 Graph 屏幕。

按下  $\blacktriangledown$  [GRAPH]



7. 选取 **Trace**。移动光标逐年进行轨迹跟踪。看看多少年 (nc) 后树木的数量 (yc) 能够稳定?

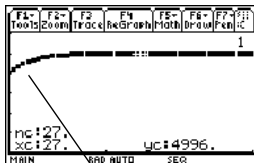
nc=0 时开始跟踪。

nc 为年数。

xc = nc, 因为 nc 已经在 x 轴上标出。

yc = u1(n), 即在第 n 年时的树木数量。

在必要时可按下  $\boxed{F3}$   $\blacktriangleright$  和  $\textcircled{1}$



缺省情况下, 序列图形采用小方块作为绘图样式。

## 3D 作图

作出 3 维方程  $z(x,y) = (x^3y - y^3x) / 390$  的图形。使用光标以交互方式改变控制视角的 eye 窗口变量值, 您可以使图形动态变化。然后可以用不同图形格式来观察该图形。

## 步骤和键击

## 显示

1. 显示 **MODE** 对话框。对于 **Graph** 模式, 选取 **3D**。

按下  $\boxed{\text{MODE}}$   $\blacktriangleright$  5  $\boxed{\text{ENTER}}$

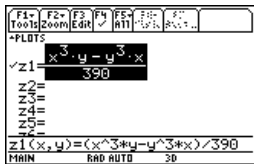


2. 显示并清空 **Y= Editor**。然后定义 3 维方程

$z1(x,y) = (x^3y - y^3x) / 390$ 。

注意在键击中使用的隐式乘法。

按下  $\blacktriangledown$  [Y=]  $\boxed{F1}$  8  $\boxed{\text{ENTER}}$   $\boxed{\text{ENTER}}$   $\boxed{\text{X}}$   $\wedge$  3  $\boxed{\text{Y}}$   $\boxed{-}$   $\boxed{\text{Y}}$   $\wedge$  3  $\boxed{\text{X}}$   $\boxed{/}$  390  $\boxed{\text{ENTER}}$

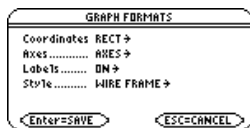


3. 改变图形格式以显示并标明坐标轴。并设定 **Style = WIRE FRAME**。

您可以尝试使用任意一种图形格式，但 **WIRE FRAME** 格式最快捷。

1 2 2 1 **[ENTER]**

F 2 2 1 **[ENTER]**



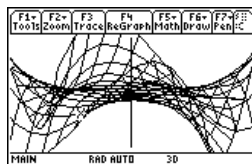
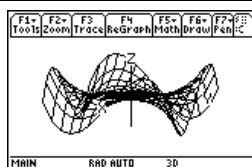
4. 选取 **ZoomStd** 观测立方体，它可以自动作出方程的图形。

在进行方程计算时（作图前），在屏幕的左上上方会显示出“evaluation percentages”（计算进度百分比）的字样。

按下 **[F2] 6**

**注：**如果您已经使用 3 维作图，该图形将以透视图显示。当您使图形动态变化时，屏幕自动恢复到正常视角。（除使图形动态变化外，您在正常视角和透视图情况下的操作是完全相同的。）

按下 **[X]**（按下 **[X]** 在扩张和正常视角间切换）

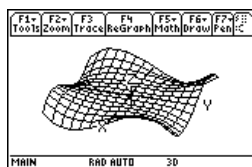


5. 改变 **eyeφ** 窗口变量值可使图形动态变化。

或 会影响 **eyeθ** 和 **eyeψ**，但其程度不及对 **eyeφ** 的影响。

欲使图形连续动态变化，可按下光标并保持 1 秒钟后再松开。欲停止变化，可按下 **[ENTER]**。

连续按 键 8 次。



6. 将图形恢复到初始状态。然后沿图形的“视角轨道”移动视角。

按下 **0**（数字零，而不是字母 **O**）



---

**步骤和键击****显示**

---

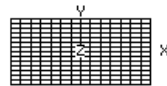
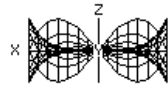
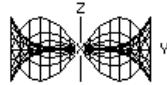
7. 沿  $x$  轴、 $y$  轴、和  $z$  轴观察该图形。

按下  $X$

该图形沿  $y$  轴和  $x$  轴有相同的形状。

按下  $Y$

按下  $Z$




---


8. 恢复到初始状态。

按下  $0$  (数字零)

---

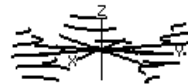
9. 以不同图形格式显示该图形。

  $\square$  (按下  $\square$  在各种格式间依次切换)

  $F$  (按下  $F$  在各种格式间依次切换)



不可见表面



轮廓  
(计算轮廓需要一定的时间)



线条和轮廓



线框

---

**注:** 您也使用 **GRAPH FORMATS** 对话框, 将该图形作为隐式图显示 (**TI-89 Titanium:**  $\square$  **Voyage™ 200:**  $F$ )。如果您按下 **TI-89 Titanium:**  $\square$  **Voyage™ 200:**  $F$  在格式间进行切换, 则隐式图不会显示出来。



## 微分方程作图

作出 Logistic 一阶微分方程  $y' = .001y*(100-y)$  的解的图形。先作出斜率场图。然后在图形屏幕上，以交互方式在 **Y= Editor** 中输入初始条件。

### 步骤和键击

### 显示

1. 显示 **MODE** 对话框。对于 **Graph** 模式下，选取 **DIFF EQUATIONS**。

按下 **[MODE]** **[6]** **[ENTER]**



2. 显示并清空 **Y= Editor**。然后定义一阶微分方程：

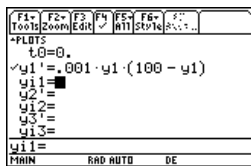
$$y1'(t) = .001y1*(100-y1)$$

按下 **[\*]** 以输入如上所示的 \*。在变量和括号间不要使用隐式乘法运算。否则，将会被当作函数调用。

将初始条件 **y11** 置空。

**注：** 由于选取了 **y1'**，计算器将作出 **y1** 的曲线，而不是导数 **y1'** 的曲线。

按下 **[Y=]** **[F1]** **[8]** **[ENTER]** **[ENTER]** **.001** **Y1** **[\*]** **[100]** **[\*]** **Y1** **[\*]** **[ENTER]**

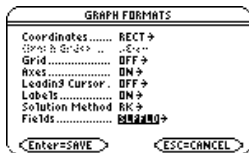


3. 显示 **GRAPH FORMATS** 对话框。然后设定 **Axes = ON**、**Labels = ON**、**Solution Method = RK**、及 **Fields = SLPFLD**。

**注：** 欲作出微分方程的图形，**Fields** 必须设定为 **SLPFLD** 或 **FLDOFF**。如果 **Fields=DIRFLD**，在作图时将出错。

**[F1]** **[1]** **[2]** **[2]** **[1]** **[1]** **[ENTER]**

**[F2]** **[2]** **[2]** **[1]** **[1]** **[ENTER]**



## 步骤和键击

## 显示

4. 显示 **Window Editor**，并如右边所示设定窗口变量。

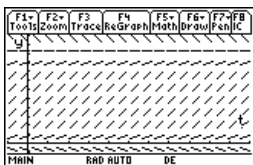
按下  $\blacktriangleright$  [WINDOW] 0  $\blacktriangleright$  10  $\blacktriangleright$  .1  $\blacktriangleright$  0  $\blacktriangleright$  [C] 10  $\blacktriangleright$  110  $\blacktriangleright$  10  $\blacktriangleright$  [C] 10  $\blacktriangleright$  120  $\blacktriangleright$  10  $\blacktriangleright$  0  $\blacktriangleright$  .001  $\blacktriangleright$  20

```
t0=0.
tmax=10.
tstep=.1
tplot=0.
xmin=-10.
xmax=110.
xsc1=10.
ymin=-10.
ymax=120.
ysc1=10.
ncurves=0.
diftol=.001
fldres=20.
```

5. 显示 Graph 屏幕。

由于您未指定初始条件，因此仅绘制出斜率场 (由 **GRAPH FORMATS** 对话框中的 **Fields=SLPFLD** 进行确定)。

按下  $\blacktriangleright$  [GRAPH]



6. 返回 **Y= Editor** 并输入初始条件:

**yi1=10**

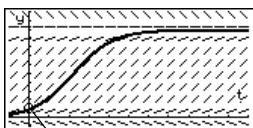
按下  $\blacktriangleright$  [Y=] [ENTER] 10 [ENTER]

```
F1- F2- F3- F4- F5- F6- F7-
Tools Zoom Trace RecGraph Math Draw Fn C
*PLOTS
t0=0.
y1'= .001·y1·(100 - y1)
yi1=10
```

7. 返回 Graph 屏幕。

在 **Y= Editor** 中输入的初始条件通常是  $t_0$  时刻的情形。图形从初始条件处起笔向右绘制。然后图形再在左边绘制。

按下  $\blacktriangleright$  [GRAPH]



用圆圈标记初始条件

8. 返回 **Y= Editor** 并以数列形式输入2个初始条件以改变 **yi1** 值:

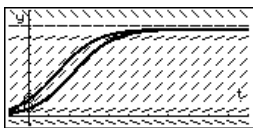
**yi1={10,20}**

按下  $\blacktriangleright$  [Y=]  $\ominus$  [ENTER] 2nd [ { ] 10 [ , ] 20 [ 2nd ] [ } ] [ENTER]

```
F1- F2- F3- F4- F5- F6- F7-
Tools Zoom Edit RTN Step S...
*PLOTS
t0=0.
y1'= .001·y1·(100 - y1)
yi1={10 20}
```

9. 返回 Graph 屏幕。


按下  $\blacktriangleright$  [GRAPH]



## 步骤和键击

## 显示

10. 要交互式地选取初始条件，按下：


 **2nd** **[F8]**


 **F8**

当出现提示时，输入  $t=40$  和  $y1=45$ 。

在交互式选取初始条件时，您可以在

**Y= Editor** 或 **Window Editor** 中输入不同于  $t_0$  值的  $t$  值。


若不想在按下  **2nd** **[F8]**

 **F8** 后输入  $t$  和  $y1$ ，

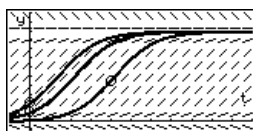
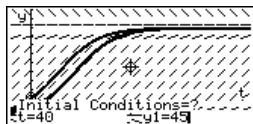
您可以移动光标至屏幕上某一点并按下

**[ENTER]**。

您可以使用 **[F3]** 跟踪由 **Y= Editor** 中初始条件所指定的曲线。然而，您不能跟踪通过交互方式选取初始条件的曲线。

 **2nd** **[F8]** **40** **[ENTER]** **45** **[ENTER]**

 **F8** **40** **[ENTER]** **45** **[ENTER]**



## 添加作图

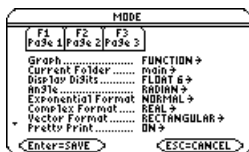
在主屏幕上，作出分段函数： $y = -x$  且  $x < 0$  和  $y = 5 \cos(x)$  且  $x \geq 0$  的图形。过余弦曲线的顶点作一条水平线，然后将所显示的图形存为图片。

## 步骤和键击

## 显示


1. 显示 **MODE** 对话框。对于 **Graph** 模式下，选取 **FUNCTION**。对于 **Angle** 模式下，选取 **RADIAN**。


按下 **[MODE]** **1** **[<]** **[>]** **[>]** **[>]** **1** **[ENTER]**



2. 显示主屏幕。使用 **Graph** 命令和 **when** 函数指定分段函数。

**[F4]** **2** 从 **Other** 工具条菜单中选取 **Graph** 并自动添加一个空格。

 **[HOME]** **[F4]** **2** **[2nd]** **[a-lock]** **WHEN** **[alpha]** **[ ]** **X**  
**[2nd]** **[<]** **0** **[,]** **[<]** **X** **[,]** **5** **[x]** **[2nd]** **[COS]** **X** **[ ]** **[ ]**

 **[<]** **[CALC HOME]** **[F4]** **2** **WHEN** **[ ]** **X** **[2nd]** **[<]**  
**0** **[,]** **[<]** **X** **[,]** **5** **[x]** **[COS]** **X** **[ ]** **[ ]**

Graph when( $x < 0$ ,  $-x$ ,  
 $5 * \cos(x)$ )

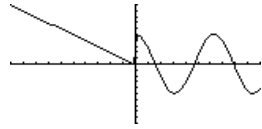
## 步骤和键击

## 显示

3. 执行 **Graph** 命令，它可以自动显示 **Graph** 屏幕。

该图形使用当前的 **Window** 变量，在本例中假定这些变量为其标准值 (**F2** **6**)。

按下 **[ENTER]**




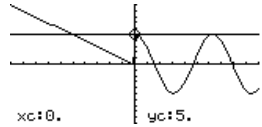
4. 在余弦曲线顶点作一条水平线。

在您选取其它运算或按下 **[ESC]** 之前，计算器保持“line”模式不变。

 **[2nd]** **[F7]** **5** **⇐** (直至水平线定位完毕)

**[ENTER]**

 **[F7]** **5** **⇐** (直至水平线定位完毕) **[ENTER]**



5. 将该图形存为图片。使用 **PIC1** 作为图片的变量名。

确认设定 **Type = Picture**。在缺省情况下，它被设定为 **GDB**。


 **[F1]** **2** **⇓** **2** **⇐** **[alpha]** **1** **[ENTER]** **[ENTER]**

 **[F1]** **2** **⇓** **2** **⇐** **[alpha]** **PIC1** **[ENTER]** **[ENTER]**

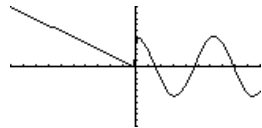


6. 清除所作的水平线。

您也可以按下 **[F4]** 以重新作图。

 **[2nd]** **[F6]** **1**

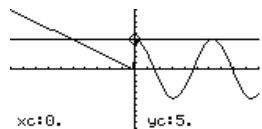
 **[F6]** **1**



7. 打开存储的图片变量并重新显示含上述水平线的图形。

确认设定 **Type = Picture**。在缺省情况下，它被设定为 **GDB**。

按下 **[F1]** **1** **⇓** **2** (若尚未显示，也可设定 **Variable = pic1**) **[ENTER]**



## 表格

计算在自变量取  $-10$  到  $10$  之间的整数时函数  $y=x^3-2x$  的值。有多少次函数值出现正负变化的情形？以及所对应的自变量是多少？

### 步骤和键击

### 显示

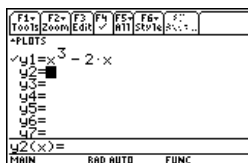
1. 显示 **MODE** 对话框。对于 **Graph** 模式，选取 **FUNCTION**。

按下  $\boxed{\text{MODE}}$   $\rightarrow$   $\boxed{1}$   $\boxed{\text{ENTER}}$



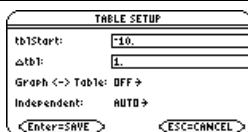
2. 显示并清空 **Y= Editor**。然后定义  $y_1(x) = x^3 - 2x$ 。

按下  $\boxed{\blacktriangleright}$   $\boxed{[Y=]}$   $\boxed{\text{F1}}$   $\boxed{8}$   $\boxed{\text{ENTER}}$   $\boxed{\text{ENTER}}$   $\boxed{X}$   $\boxed{\wedge}$   $\boxed{3}$   $\boxed{-}$   $\boxed{2}$   $\boxed{X}$   $\boxed{\text{ENTER}}$



3. 设定表格参数为：  
**tblStart = -10**  
**Δtbl = 1**  
**Graph <-> Table = OFF**  
**Independent = AUTO**

按下  $\boxed{\blacktriangleright}$   $\boxed{[\text{TBLSET}]}$   $\boxed{(-)}$   $\boxed{10}$   $\boxed{\odot}$   $\boxed{1}$   $\boxed{\odot}$   $\boxed{\blacktriangleright}$   $\boxed{1}$   $\boxed{\odot}$   $\boxed{\blacktriangleright}$   $\boxed{\blacktriangleright}$   $\boxed{\text{ENTER}}$



4. 显示表格屏幕。

按下  $\boxed{\blacktriangleright}$   $\boxed{[\text{TABLE}]}$

x	y1
-10.	-980.
-9.	-711.
-8.	-496.
-7.	-329.
-6.	-204.

5. 来回滚动表格。注意 **y1** 的正负发生变化的位置是  $x = -1$ 、 $1$  和  $2$ 。

要一次滚动一个页面，可使用  $\boxed{2\text{nd}}$   $\boxed{\odot}$  和  $\boxed{2\text{nd}}$   $\boxed{\odot}$ 。

在需要时按下  $\boxed{\odot}$  和  $\boxed{\blacktriangleleft}$

x	y1
-1.	1.
0.	0.
1.	-1.
2.	4.
3.	21.

6. 如下改变表格参数，以放大  $x = -2$  和  $x = -1$  间发生函数值正负变化的位置：

**tblStart = -2**  
**Δtbl = .1**

按下  $\boxed{\text{F2}}$   $\boxed{(-)}$   $\boxed{2}$   $\odot$   $\boxed{.1}$   $\boxed{\text{ENTER}}$   $\boxed{\text{ENTER}}$

x	y1
-2.	-4.
-1.9	-3.059
-1.8	-2.232
-1.7	-1.513
-1.6	-.896

## 屏幕拆分

拆分屏幕，使之同时显示 **Y= Editor** 和 Graph 屏幕。然后观察当系数改变时多项式的形态变化。

### 步骤和键击

### 显示

1. 显示 **MODE** 对话框。

对于 **Graph** 模式，选取 **FUNCTION**。

对于 **Split Screen** 模式，选取 **LEFT-RIGHT**。

对于 **Split 1 App**，选取 **Y= Editor**。

对于 **Split 2 App**，选取 **Graph**。

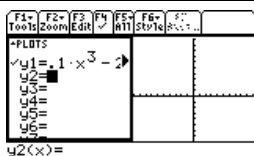
按下 **[MODE]** **[1]** **[F2]** **[3]** **[2]** **[4]** **[ENTER]**



2. 清空 **Y= Editor** 并关闭所有统计数据图。然后定义  $y1(x) = .1x^3 - 2x + 6$ 。

**Y= Editor** 的粗边框说明其处于激活状态。在激活状态下，其输入行可延伸至整个显示区的宽度。

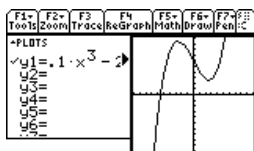
按下 **[F1]** **[8]** **[ENTER]** **[F5]** **[5]** **[ENTER]** **.1 X ^ 3 - 2 X + 6** **[ENTER]**



3. 选取 **ZoomStd** 观测窗，它可以切换到 Graph 屏幕并作出函数图形。

Graph 屏幕围呈现粗边框。

按下 **[F2]** **[6]**



4. 切换至 **Y= Editor**。然后编辑 **y1(x)** 来将  $.1x^3$  改变为  $.5x^3$ 。

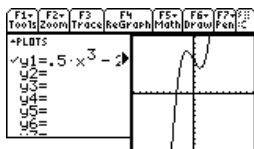
**[2nd]** **[=]** 是 **[APPS]** 的第二功能。**Y= Editor** 呈现粗边框。

按下 **[2nd]** **[=]** **[ENTER]** **[4]** **[5]** **[ENTER]**

5. 切换到 **Graph** 屏幕，它可以重新作出编辑过的函数的图形。

Graph 屏幕呈现粗边框。

按下 **[2nd]** **[=]**

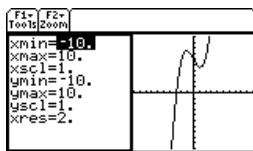


## 步骤和键击

## 显示

6. 切换至 **Y= Editor**。然后在其中打开 **Window Editor**。

按下 **[2nd] [F2]** **[WINDOW]**



7. 打开主屏幕。然后切换到全屏状态的主屏幕。

连续 2 次按下 **[2nd] [QUIT]**。

## 数据 / 矩阵编辑器 (Data/Matrix Editor)

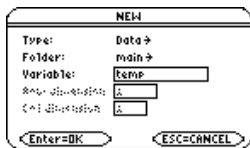
使用 **Data/Matrix Editor** 创建单列数组变量。然后填加第 2 列的信息。注意数组变量（可只有一列）会自动转换成数据变量（可为多列）。

## 步骤和键击

## 显示

1. 使用 **[APPS]** 来显示 **Data/Matrix Editor**, 创建名为 **TEMP** 的新数组变量。

按下 **3** **[>]** **3** **[<]** **TEMP** **[ENTER]** **[ENTER]**

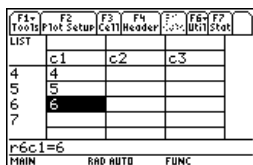


2. 输入一系列数字。然后将光标上移到某个单元格（以观察高亮单元格中的数值是否出现在输入行中）。

在左上角显示出 **LIST**，说明这是一个数组变量。

您可以使用 **[<]** 来代替 **[ENTER]** 在单元格中输入信息。

按下 **1** **[ENTER]** **2** **[ENTER]** **3** **[ENTER]** **4** **[ENTER]** **5** **[ENTER]** **6** **[ENTER]** **[<]**

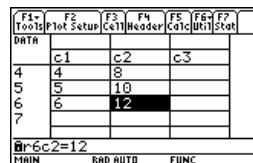


3. 移动光标到第 2 列，然后定义其列头使得第 2 列的值为第 1 列的 2 倍。

在左上角显示出 **DATA**，说明数组变量已经转换为数据变量。

**[<]** **[F4]** **2** **[x]** **[alpha]** **C 1** **[ENTER]**

**[<]** **[F4]** **2** **[x]** **C 1** **[ENTER]**



**[<]** 表示该单元格位于已定义的列中。

## 步骤和键击

## 显示

- 移动光标到第 2 列头的单元格中，将其定义显示在输入行中。

当光标位于列头单元格中时，您无须按下 **[F4]** 对其进行定义。只需输入表达式即可。

按下 **[2nd] [←]**

F1 [Tools]	F2 [Plot Setup]	F3 [CST]	F4 [Mode]	F5 [C/C]	F6 [Unit]	F7 [Stat]
DATA						
	c1	c2	c3			
1	1	2				
2	2	4				
3	3	6				
4	4	8				
c2=2*c1						
MAIN      RAD AUTO      FUNC						

- 清除变量内容。

仅仅清除数据并不能将数据变量重新转化为数组变量。

按下 **[F1] 8 [ENTER]**

**注：**若无需存储当前变量值，您可将其作为 *中间结果暂存器* 使用。当您需要某个变量来存放临时数据时，清除当前变量值即可重新使用。这使您不必每次都创建新的变量即可输入临时数据，从而最大限度地利用了内存。

## 统计图和数据图

以 7 个城市为样本，输入描述城市人口数量与 12 层以上楼宇的数量之间关系的数据。使用中位数 - 中位数法和线性回归，确定符合相关数据的方程并进行作图。对于每个回归方程，请您预测在一个拥有 300,000 居民的城市里可能有多少栋 12 层以上的楼宇。

## 步骤和键击

## 显示

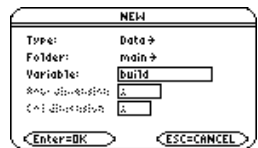
- 显示 **MODE** 对话框。对于 **Graph** 模式，选取 **FUNCTION**。

按下 **[MODE] [1] [ENTER]**



- 使用 **[APPS]** 以显示 **Data/Matrix Editor**。创建一个新的数据变量，命名为 **BUILD**。

按下 **3 [←] [←] BUILD [ENTER] [ENTER]**





## 步骤和键击

## 显示

3. 使用以下的样本数据，在第一列中输入人口数量。

人口 (千人)	楼宇 > 12 层
150	4
500	31
800	42
250	9
500	20
750	55
950	73

F1- Tools	F2 Plot Setup	F3 Cell	F4 Header	F5 Calc	F6- Util	F7 Stat
DATA						
	c1	c2	c3			
5	500					
6	750					
7	950					
8						
r8c1=						
MAIN		DEG AUTO		FUNC		

按下 150 [ENTER] 500 [ENTER] 800 [ENTER] 250 [ENTER] 500 [ENTER] 750 [ENTER] 950 [ENTER]

4. 将光标移动到第 2 列的第 1 行 (r1c2)。然后输入相应的楼宇数量。

将光标移动到页面顶部。在单元格中键入数据后，您可以按下 [ENTER] 或 输入数据并将光标下移到某个单元格上。按下 输入数据并将光标将上移到某个单元格上。

4 [ENTER] 31 [ENTER] 42 [ENTER] 9 [ENTER] 20 [ENTER] 55 [ENTER] 73 [ENTER]

[2nd] 4 [ENTER] 31 [ENTER] 42 [ENTER] 9 [ENTER] 20 [ENTER] 55 [ENTER] 73 [ENTER]

F1- Tools	F2 Plot Setup	F3 Cell	F4 Header	F5 Calc	F6- Util	F7 Stat
DATA						
	c1	c2	c3			
5	500	20				
6	750	55				
7	950	73				
8						
r8c2=						
MAIN		DEG AUTO		FUNC		

5. 将光标移动到第 1 列的第 1 行 (r1c1)。将人口数量按升序排列数据。

此操作将对第 1 列进行排序，然后调整所有其它列，以使它们保持与第 1 列相同的次序。此操作对于保持各列数据间的关系十分重要。

在对第 1 列进行排序时，光标可以停留在第 1 列的任何位置。在此例中您可以按下

[2nd]

即可以看见前 4 行。

[2nd] [F6] 4

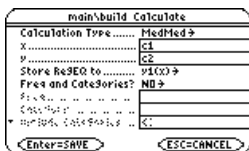
[2nd] [F6] 4

1: Insert	
2: Delete	
3: Sort Column	
4: Sort Col, adjust all	
5: Clear Column	
6: Clear Row	

F1- Tools	F2 Plot Setup	F3 Cell	F4 Header	F5 Calc	F6- Util	F7 Stat
DATA						
	c1	c2	c3			
1	150	4				
2	250	9				
3	500	31				
4	500	20				
r1c1=150						
MAIN		RAD AUTO		FUNC		

6. 显示 **Calculate** 对话框。设定 **Calculation Type = MedMed**  
**x = C1**  
**y = C2**  
**Store RegEQ to = y1(x)**

按下 **F5**  $\rightarrow$  **7**  $\rightarrow$  **C** **alpha** **1**  $\rightarrow$  **alpha** **C2**  $\rightarrow$   $\rightarrow$  **ENTER**  
 按下 **F5**  $\rightarrow$  **7**  $\rightarrow$  **C1**  $\rightarrow$  **C2**  $\rightarrow$   $\rightarrow$  **ENTER**



7. 执行运算并显示中位数 - 中位数回归方程。  
 如 **Calculate** 对话框所示，此方程存储于 **y1(x)** 中。

按下 **ENTER**

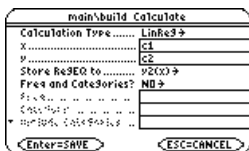


8. 关闭 **STAT VARS** 屏幕。 **Data/Matrix Editor** 出现。

按下 **ENTER**

9. 显示 **Calculate** 对话框。设定：  
**Calculation Type = LinReg**  
**x = C1**  
**y = C2**  
**Store RegEQ to = y2(x)**

按下 **F5**  $\rightarrow$  **5**  $\rightarrow$   $\rightarrow$   $\rightarrow$   $\rightarrow$  **ENTER**



10. 执行运算并显示 **LinReg** 回归方程。  
 此方程式储存在 **y2(x)** 内。

按下 **ENTER**

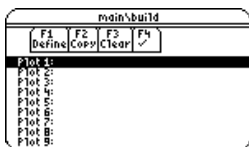


11. 关闭 **STAT VARS** 屏幕。 **Data/Matrix Editor** 出现。

按下 **ENTER**

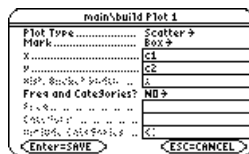
12. 显示 **Plot Setup** (绘图设置) 屏幕。  
**Plot 1** 在缺省情况下处于高亮状态。  
 您可以使用 **F3** 清除高亮显示的 **Plot** 设置。

按下 **F2**



13. 定义 **Plot 1** 为:

**Plot Type = Scatter**  
**Mark = Box**  
**x = C1**  
**y = C2**



请注意此屏幕和 **Calculate** 对话框间的相似性。

**F1** **1** **1** **1** **C2**  
**F1** **1** **1** **C1** **C2**

14. 保存图形定义并返回 **Plot Setup** 屏幕。

请注意 **Plot 1** 的定义中使用的简化符号。  
 连续 2 次按下 **ENTER**



15. 显示 **Y= Editor**。对于 **y1(x)**、以及中位数 - 中位数回归方程，设置显示风格为 **Dot**。

**注:** 视 **Y= Editor** 中已有的内容而定，您可能需要将光标移到 **y1**。



屏幕顶部的 **PLOTS 1** 表示 **Plot 1** 被选中。  
 请注意在保存回归方程时，**y1(x)** 和 **y2(x)** 同时被选中。

**[Y=]** **[F6]** **2**  
**[Y=]** **2**

16. 向上滚动以高亮 **Plot 1**。

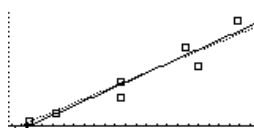
显示出的简化定义与 **Plot Setup** 屏幕上所显示的相同。  
 按下



17. 使用 **ZoomData** 绘制 **Plot 1** 和回归方程 **y1(x)** 和 **y2(x)**。

**ZoomData** 对所有选中的统计图进行数据检查，并调整观测窗使其包含所有点。

按下 **F2** **9**




18. 返回到 **Data/Matrix Editor** 的当前进程。

按下 **[2nd][⇐]**

19. 输入第 3 列的标题。将第 3 列列头定义为中位数 - 中位数线的预测值。

欲输入标题，光标必须高亮该列靠近顶端的列头单元格。

您可以使用 **[F4]** 在一列的任何位置定义列头。当光标停留在某个列头单元格上时，无须按下 **[F4]**。

 **[→][→][←][←][2nd][a-lock] MED [alpha] [ENTER]**  
**[F4] Y1 [⏏] [alpha] C1 [⏏] [ENTER]**

 **[→][→][←][←] MED [ENTER] [F4] Y1 [⏏] C1 [⏏]**  
**[ENTER]**

20. 输入第 4 列的标题。将第 4 列的列头定义为中位数 - 中位数残差（观测值和预测值之间的差值）。

 **[→][←][2nd][a-lock] RESID [alpha] [ENTER] [alpha]**  
**C2 [⏏] [alpha] C3 [ENTER]**

 **[→][←] RESID [ENTER] [F4] C2 [⏏] C3 [ENTER]**


F1- Tools	F2 Plot Setup	F3 c1	F4 Header	F5 Calc	F6 Util	F7 Stat
DATA		med		resid		
		c2	c3	c4		
1	4	3.3333	.66667			
2	9	10.889	-1.889			
3	31	29.778	1.2222			
4	20	29.778	-9.778			
c4=c2-c3						
MAIN		RAD AUTO		FUNC		

21. 输入第 5 列的标题。根据 **LinReg**（线性回归）线的预测值来定义第 5 列的列头。

 **[→][←][2nd][a-lock] LIN [alpha] [ENTER] [F4] Y2**  
**[⏏] [alpha] C1 [⏏] [ENTER]**

 **[→][←] LIN [ENTER] [F4] Y2 [⏏] C1 [⏏] [ENTER]**

22. 输入第 6 列的标题。根据 **LinReg**（线性回归）线的残差来定义第 6 列的列头。

 **[→][←][2nd][a-lock] RESID [alpha] [ENTER] [F4]**  
**[alpha] C2 [⏏] [alpha] C5 [ENTER]**

 **[→][←] RESID [ENTER] [F4] C2 [⏏] C5 [ENTER]**

F1- Tools	F2 Plot Setup	F3 c1	F4 Header	F5 Calc	F6 Util	F7 Stat
DATA		resid	lin	resid		
		c4	c5	c6		
1		.66667	22169	3.7783		
2		-1.889	8.3778	.62224		
3		1.2222	28.768	2.232		
4		-9.778	28.768	-8.768		
c6=c2-c5						
MAIN		RAD AUTO		FUNC		

23. 显示 **Plot Setup** 屏幕并取消对 **Plot 1** 的选取。

按下 **[F2][F4]**

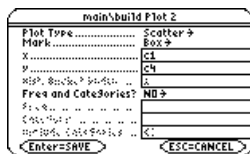
24. 高亮 **Plot 2** 并定义如下:

**Plot Type = Scatter**

**Mark = Box**

**x = C1**

**y = C4** (中位数 - 中位数残差)



$\rightarrow$  **F1**  $\rightarrow$   $\rightarrow$  **C** **alpha** **1**  $\rightarrow$  **alpha** **C4** **ENTER**  
**ENTER**

$\rightarrow$  **F1**  $\rightarrow$   $\rightarrow$  **C1**  $\rightarrow$  **C4** **ENTER** **ENTER**

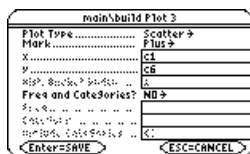
25. 高亮 **Plot 3** 并定义为:

**Plot Type = Scatter**

**Mark = Plus**

**x = C1**

**y = C6** (LinReg (线性回归) 残差)



$\rightarrow$  **F1**  $\rightarrow$   $\rightarrow$  **3**  $\rightarrow$  **C** **alpha** **1**  $\rightarrow$  **alpha** **C6**  
**ENTER** **ENTER**

$\rightarrow$  **F1**  $\rightarrow$   $\rightarrow$  **3**  $\rightarrow$  **C1**  $\rightarrow$  **C6** **ENTER** **ENTER**

26. 高亮 **Y= Editor** 并关闭所有 **y(x)** 函数。

从 **F5**, 选取 **3:Functions Off**, 而不是 **1:All Off**。

图 2 和图 3 仍然处于选中状态。

按下  $\blacklozenge$  **[Y=]** **F5** **3**

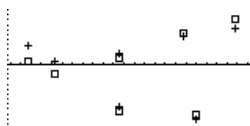


27. 使用 **ZoomData** 绘制残差。

$\square$  标出了中位数中位数 - 中位数残差;

$+$  标出了 LinReg (线形回归) 残差。

按下 **F2** **9**



28. 显示主屏幕。

**HOME**

$\blacklozenge$  **[CALC HOME]**

## 步骤和键击

## 显示

- 使用 **MedMed** ( $y_1(x)$ ) (中位数 - 中位数) 和 **LinReg** ( $y_2(x)$ ) 线性回归方程来计算  $x = 300$  (300,000 人口) 时的值。

**round** 函数 ( $\text{2nd}$  [MATH] 1 3) 确保楼宇数量的结果为整数。

在第一次计算后, 编辑输入行, 将 **y1** 改成 **y2**。

按下  $\text{2nd}$  [MATH] 1 3 Y1  $\left[ \square \right]$  300  $\left[ \right]$  , 0  $\left[ \right]$  [ENTER]

$\left[ \downarrow \right]$   $\left[ \downarrow \right]$  (连续 8 次)  $\left[ \leftarrow \right]$  2 [ENTER]

F1->	F2->	F3->	F4->	F5	F6->
[Tool]	[Control]	[Var]	[Find...]	[Mode]	
■ round(y1(300), 0)		15.			
■ round(y2(300), 0)		12.			
round(y2(300), 0)					
MAIN		RAD AUTO		FUNC	
				2/30	

## 编程

编写一个程序, 用以提示用户输入一个整数, 然后求从 1 到该整数之间的所有整数之和, 并显示结果。

## 步骤和键击

## 显示

- 使用 [APPS] 显示 **Program Editor**。创建一个新的程序。

按下 3



- 键入 **PROG1** (不能含空格) 作为新程序变量的名称。

$\left[ \text{key} \right]$   $\left[ \downarrow \right]$   $\left[ \downarrow \right]$  PROG  $\left[ \alpha \right]$  1

$\left[ \text{key} \right]$   $\left[ \downarrow \right]$   $\left[ \downarrow \right]$  PROG 1



- 为新程序打开“template” (模板)。程序名、**Prgm**、和 **EndPrgm** 会自动显现。

在输入框中键入如 **Variable** 等内容后, 您必须连续 2 次按下 [ENTER]。

连续 2 次按下 [ENTER]



## 步骤和键击

## 显示

4. 键入下列程序行。

**Request "Enter an integer",n**

显示出提示“Enter an integer”（输入一个整数）的对话框，等待用户输入数值，并将该数值（以字符串形式）赋给变量 **n**。

```
:prog1()  
:Prgrn  
:Request "Enter an integer"  
:n  
:expr(n)→n  
:0→temp  
:For i,1,n,1  
:temp+i→temp  
:EndFor  
:Disp temp  
:  
:EndPrgrn
```

**expr(n)→n**

将字符串转换成数字表达式。

**0→temp**

创建名为 **temp** 的变量，并赋初值 **0**。

**For i,1,n,1**

以 **i** 为循环变量启动一个 *For loop* 循环。在第一次循环时，**i = 1**。在每次循环结束时，**i** 将增加 1。循环直到 **i > n** 时才停止。

**temp+i→temp**

将 **i** 的当前数值增加给 **temp**。

**EndFor**

标明 *For loop* 循环结束。

**Disp temp**

显示 **temp** 的最终数值。 .


键入所示的程序行。在每行的最后按下 **[ENTER]**。

5. 回到主屏幕。输入程序名和一对括号。

prog1()

即使在程序不带自变量时，也不得省略上述 **()**。

程序将显示出一个对话框，其中内容为程序中所确定的提示信息。

 **[HOME]** **[2nd]** **[a-lock]** **PROG** **[alpha]** **1** **[ ( ]**  
**[ENTER]**

 **[CALC HOME]** **PROG1** **[ ( ]** **[ENTER]**

6. 在所显示的对话框中键入 5。

按下 5



## 步骤和键击

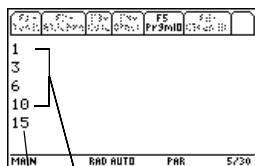
- 继续执行程序。**Disp** 命令使结果显示在 Program I/O 屏幕上。

结果为 1 到 5 的整数之和。

尽管 Program I/O 屏幕看起来和主屏幕类似，但它仅提供程序输入和输出功能。您无法在 Program I/O screen 进行运算。

连续按 2 次 **[ENTER]**

## 显示



其它程序的输入结果可能还留在屏幕上。

输入整数 5 时的结果

- 退出 Program I/O 屏幕并返回主屏幕。

您也可以按下 **[ESC]**, **[2nd]** **[QUIT]**, 或



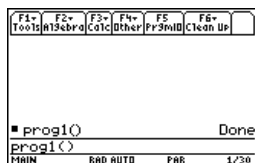
**[HOME]**



**[CALC HOME]**

来返回主屏幕。

按下 **[F5]**



## 文本操作

启动一个新的 **Text Editor** 进程。然后随意键入文本信息来练习使用 **Text Editor**。在您键入时，可练习移动文本光标并对所输入的文稿进行修改。

## 步骤和键击

- 启动一个新的 **Text Editor** 进程。

按下 **3**

## 显示

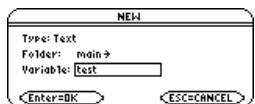


- 创建名为 **TEST** 的文本变量，它将自动存储您在新进程中输入的任何文本。

如 **NEW** 对话框中的缺省情况所示，使用 **MAIN** 文件夹。

在输入框中键入如 **Variable** 等内容，您必须连续 2 次按下 **[ENTER]**。

按下 **[C]** **[TEST]** **[ENTER]** **[ENTER]**













## 步骤和键击

## 显示



### 3. 输入文本。

- 欲键入一个大写字母，需按下  后再按下该字母键。

仅适用于 TI-89 Titanium:

- 欲输入一个空格，按下   ( 键的 alpha 功能)。
- 欲键入一个句号，按下  以关闭 alpha-lock，按下 ，然后再按下   将 alpha-lock 重新开启。

按以下步骤练习编辑您输入的文本：

- 用光标键移动文本光标。
- 用  或  [DEL] 可分别删除光标左边或右边的字母。



  [a-lock] 随意键入信息

 随意键入信息





### 4. 退出 **Text Editor** 并重新显示主屏幕。

您输入时，文本进程会自动进行保存。因此，您无须在退出 **Text Editor** 时手动存储该进程。

  [CALC HOME]

### 5. 返回 **Text Editor** 上的当前进程。观察所显示进程是否和您退出时完全一样。

按下  

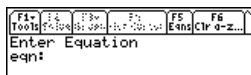
## 数值求解器 (Numeric Solver)

对于方程  $a = (m_2 - m_1) / (m_2 + m_1) * g$ ，已知  $m_2 = 10$  且  $g = 9.8$ 。若假设  $a = 1/3 g$ ，求  $m_1$  的值。

## 步骤和键击

## 显示

### 1. 采用 显示 **Numeric Solver**。

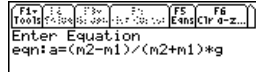
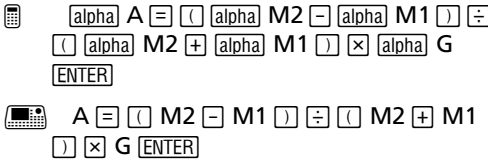


## 步骤和键击

## 显示

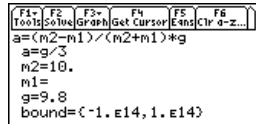
### 2. 输入方程。

当您按下 **ENTER** 或  $\odot$  时，屏幕上将列出方程中所使用的变量。



### 3. 除未知变量 m1 以外，为每个变量赋值。

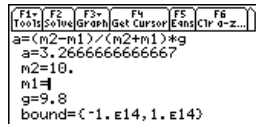
首先定义 m2 和 g。然后定义 a。(您必须在用 g 定义 a 前，先定义 g。)接受约束的缺省形式。若某个变量在此前曾定义过，其值将以缺省形式显示。



### 4. 将光标移动到未知变量 m1。

作为选择项，您可以为 m1 输入初始猜测值。即使您为所有变量赋了一个值，**Numeric Solver**（数值求解器）也只对光标标记的变量求解。

按下  $\odot$

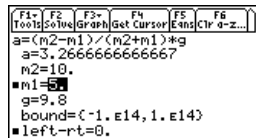


当您移动光标离开此行时， $g/3$  的值会被计算出来。

### 5. 求解未知变量。

要检查结果的准确性，可对方程式的左右两侧分别进行计算。其差值显示为 left-rt 的值。如果结果准确，则  $left-rt=0$ 。

按下 **F2**



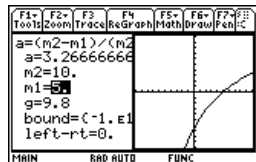
■ 标出了计算出的值。

### 6. 使用一个 ZoomStd 观测窗对结果作图。

该图形显示在一个拆分屏幕中。您可以通过跟踪、缩放等手段查看该图形。

光标标出的变量（未知变量 m1）在 x 轴上，而 left-rt 在 y 轴上。

按下 **F3** 3



7. 返回 **Numeric Solver** 并退出屏幕拆分状态。

您可以按下 **[ENTER]** 或  $\odot$  重新显示变量列表。

按下 **[2nd] [⊞] [F3] 2**

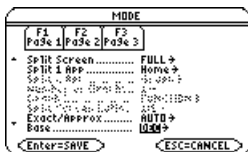
## 数基

计算二进制数字 10 (数基为 2) + 十六进制数字 F (数基为 16) + 十进制数字 10 (数基为 10)。然后, 使用  $\blacktriangleright$  算子将一个整数从某个数基转换到另一个数基。最后, 观察在改变 **Base** (数基) 模式时对所显示结果的影响。

1. 显示第 2 页上的 **MODE** 对话框。对于 **Base** 模式, 选取 **DEC** 作为缺省数基。

根据 **Base** (数基) 模式显示整数结果。分数及浮点数结果通常以十进制形式显示。

按下 **[MODE] [F2]** (使用  $\odot$  移动到 **Base** 模式)  $\odot$  **1 [ENTER]**



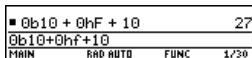
2. 计算  $0b10+0hF+10$ 。

要输入一个 2 进制或十六进制数字, 您必须使用 **0b** 或 **0h** 前缀 (零加上字母 **B** 或 **H**)。否则, 输入的数值将被视为十进制。

**注:** **0b** 或 **0h** 前缀中第一个字符是数字零, 而不是字母 **O**, 后面紧跟 **B** 或 **H**。

$\text{0 [alpha] B 10 [⊞] 0 [2nd] [a-lock] HF [alpha] [⊞] 10 [ENTER]}$

$\text{0 B 10 [⊞] 0 HF [⊞] 10 [ENTER]}$



3. 给结果加 1 并将其转换为二进制。

**[2nd] [▶]** 显示  $\blacktriangleright$  转换算子。

$\text{[+] 1 [2nd] [▶] [2nd] [a-lock] BIN [alpha] [ENTER]}$

$\text{[+] 1 [2nd] [▶] BIN [ENTER]}$

4. 给结果加 1 并将其转换为十六进制。

$\text{[+] 1 [2nd] [▶] [2nd] [a-lock] HEX [alpha] [ENTER]}$

$\text{[+] 1 [2nd] [▶] HEX [ENTER]}$

## 步骤和键击

## 显示

5. 给结果加 1 并将保持为缺省的十六进制。

结果应使用 **0b** 或 **0h** 前缀来辨别数基。

按下 **[+] 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

6. 将 **Base** 模式设为 **HEX**。

当 **Base = HEX** 或 **BIN**，结果的大小将被限定在一定范围内。

按下 **[MODE] [F2]** (使用 **↵** 移动到 **Base** 模式) **①**

**2 [ENTER]**

7. 计算 **0b10+0hF+10**。

**[📱] 0 [alpha] B 10 [+] 0 [2nd] [a-lock] HF [alpha] [+]**  
**10 [ENTER]**

**[📱] 0 B 10 [+] 0 HF [+] 10 [ENTER]**

0b10 + 0hF + 10	0h1B
0b10+0hf+10	
MAIN	RAD AUTO FUNC 1/30

8. 将 **Base** 模式设为 **BIN**。

按下 **[MODE] [F2]** (使用 **↵** 移动到 **Base** 模式) **①**

**3 [ENTER]**

9. 重新输入 **0b10+0hF+10**。

按下 **[ENTER]**

0b10 + 0hF + 10	0h1B
0b10 + 0hF + 10	0b11011
0b10+0hf+10	
MAIN	RAD AUTO FUNC 2/30

## 内存及变量管理

为各种变量数据类型赋值。使用 **VAR-LINK** 屏幕查看一组已定义好的变量。然后将某个变量移动到用户数据文档内存中，并尝试所有能成功读取和不能读取已存档变量的方法。（存档变量会被自动锁定。）最后，将该变量从文档中取出并删除未用的变量以避免其占用内存。

### 步骤和键击

### 显示


- 在主屏幕上，为下列类型的若干变量赋值。


表达式:  $5 \rightarrow x1$

函数:  $x^2 + 4 \rightarrow f(x)$

数组:  $\{5, 10\} \rightarrow I1$

矩阵:  $[30, 25] \rightarrow m1$

 **HOME** **CLEAR** 5 **STO** X1 **ENTER** X  $\wedge$  2 **+**  
4 **STO** alpha F **(** X **)** **ENTER** 2nd **(** { **)** 5 **.**  
10 **2nd** **(** } **)** **STO** alpha L1 **ENTER** 2nd **(** [ **)** 30  
**.** 25 **2nd** **(** ] **)** **STO** alpha M1 **ENTER**

 **(** **+** **]** **CALC HOME** **CLEAR** 5 **STO** X1 **ENTER** X  
 $\wedge$  2 **+** 4 **STO** F **(** X **)** **ENTER** 2nd **(** { **)** 5  
**.** 10 **2nd** **(** } **)** **STO** L1 **ENTER** 2nd **(** [ **)** 30 **.**  
25 **2nd** **(** ] **)** **STO** M1 **ENTER**

F1- Tools	F2- M3Cbrj	F3- Calc	F4- Other	F5- Pr3mD	F6- Clean Up
<ul style="list-style-type: none"> <li>■ <math>5 \rightarrow x1</math> 5</li> <li>■ <math>x^2 + 4 \rightarrow f(x)</math> Done</li> <li>■ <math>\{5, 10\} \rightarrow I1</math> {5 10}</li> <li>■ <math>[30, 25] \rightarrow m1</math> [30 25]</li> </ul>					
[30, 25] → m1					
MAIN		RAD AUTO		FUNC 4/30	

- 假如您开始使用某函数变量进行运算，但忘记了该函数的名称。

按下 5 **×**

5\*

- 显示 **VAR-LINK** 屏幕。

本例假定上述赋值变量为唯一定义的。

按下 **2nd** **[VAR-LINK]**

VAR-LINK (A11)						
F1- Mond3s	F2- ViewLink	F3- Link	F4- M1	F5- Contents	F6- FlashApp	F7- FlashApp
MAIN						
f	FUNC	19				
I1	MAT	12				
m1	MAT	12				
x1	EXPR	5				

- 改变屏幕视野使之仅显示函数变量。

尽管在一个只有 4 个变量的例子中此操作显得并非特别有用，但可以想象在涉及各种不同类型的多变量情形下，它将会非常有用。

按下 **F2** **(** **(** **)** 5 **ENTER**

VAR-LINK VIEW	
View.....	Variables →
Folder... A11	→
Var Type	Func(Expr) →
<b>ENTER</b> → BK	<b>ESC</b> → CANCEL

VAR-LINK (A11)						
F1- Mond3s	F2- ViewLink	F3- Link	F4- M1	F5- Contents	F6- FlashApp	F7- FlashApp
MAIN						
f		FUNC 19				

---

**步骤和键击****显示**

---

5. 高亮函数变量 **f**，并查看其内容。

注意该函数在赋值时使用 **f(x)** 的形式，但在屏幕列表中显示为 **f**。

  **2nd** [F6]

  **F6**

x<sup>2</sup>+4



- 
6. 关闭内容窗口。


按下 **ESC**

- 
7. 使变量 **f** 保持高亮状态，关闭 **VAR-LINK** 并将变量名称粘贴到输入行中。注意应粘贴 “(”。

按下 **ENTER**

5\*f(  


- 
8. 完成操作。

按下 **2**  **ENTER**

5\*f(2)  


---

## 存档变量

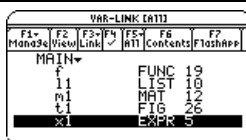
### 步骤和键击

### 显示

1. 重新显示 **VAR-LINK**，并高亮您打算存档的变量。

先前的视野改变不再有效。屏幕上会列出所有定义过的变量。

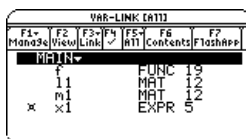
按下 **[2nd] [VAR-LINK]** (使用 **☺** 以点亮 **x1**)



2. 使用 **[F1] Manage** 工具菜单来存取变量。

**✖** 表示该变量已经存档。

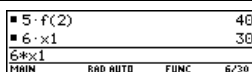
按下 **[F1] 8**



3. 返回主屏幕并在计算中使用已存档的变量。

**[HOME] 6 [X] X1 [ENTER]**

**[CALC HOME] 6 [X] X1 [ENTER]**



4. 尝试给已存档的变量赋新值。

按下 **10 [STO>] X1 [ENTER]**



5. 关闭错误信息。

按下 **[ESC]**

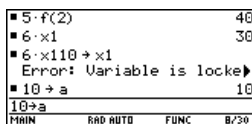
6. 使用 **VAR-LINK** 取消变量存档。

按下 **[2nd] [VAR-LINK]** (用 **☺** 高亮 **x1**) **[F1] 9**

7. 返回主屏幕并将一个新值存入已取消存档的变量中。

**[HOME] [ENTER]**

**[CALC HOME] [ENTER]**



## 删除变量

### 步骤和键击

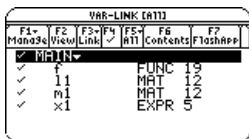
### 显示

1. 显示 **VAR-LINK**，并使用 **[F5] All** 工具条菜单以选取所有变量。

“✓”标记表示该项目已被选取。注意 **MAIN** 文件夹也同时被选中。

**注：**除使用 **[F5]** 外（若您不想删除所有变量），您可以选取单个变量。高亮某个需要删除的变量并按下 **[F4]**。

按下 **[F5] 1**



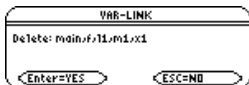
2. 使用 **[F1]** 进行删除。

**注：**您可以按下 **[←]**（而不是 **[F1] 1**）删除已标记的变量。

按下 **[F1] 1**

3. 确认删除。

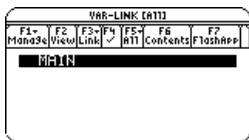
按下 **[ENTER]**



4. 由于 **[F5] 1** 命令同时也选定了 **MAIN** 文件夹，因此会出现一条错误信息警告您不可删除 **MAIN** 文件夹。确认此消息。

当 **VAR-LINK** 重新显示时，被删除的变量将不再列出。

按下 **[ENTER]**



5. 关闭 **VAR-LINK** 并返回到当前应用软件（在本例中即主屏幕）。

当您使用 **[ESC]**（而不是 **[ENTER]**）来关闭 **VAR-LINK** 时，呈高亮显示的变量名称将不会被粘贴到输入行中。

按下 **[ESC]**



## 应用实例

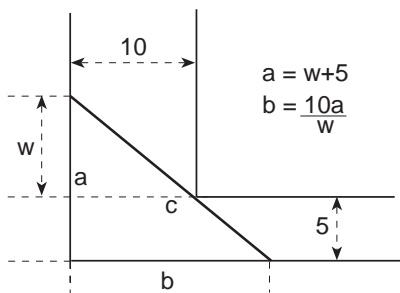
### 分析杆-角问题

一条 10 英尺宽的走廊与一条 5 英尺宽的走廊相交于建筑物一角。若要一根杆在不倾斜的条件下通过该角，请问该杆最大长度是多少？

#### 走廊中长杆的最大长度

杆的最大长度  $c$  是过内拐角并与两过道外侧接触的线段的最短值。如下图所示。

利用相似三角形性质和勾股定理求得长度  $c$  关于  $w$  的表达式。然后确定  $c(w)$  的一阶导数的零点。  $c(w)$  的最小值即杆的最大长度。



1. 定义边长  $a$  关于  $w$  的表达式，并存放在  $a(w)$  中。

**注：**当定义一个函数时，请在定义中使用多字母函数名称。

Define a(w)=w+5	Done
Define a(w)=w+5	
MAIN	RAD AUTO FUNC 1/30

2. 定义边长  $b$  关于  $w$  的表达式，并存放在  $b(w)$  中。

Define a(w)=w+5	Done
Define b(w)= $\frac{10 \cdot a(w)}{w}$	Done
Define b(w)= $10 \cdot a(w) / w$	Done
MAIN	RAD AUTO FUNC 2/30

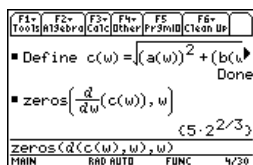
3. 定义边长  $c$  关于  $w$  的表达式，并存放在  $c(w)$  中。

输入：**Define c(w)= $\sqrt{(a(w)^2+b(w)^2)}$**

Define b(w)= $\frac{10 \cdot a(w)}{w}$	Done
Define c(w)= $\sqrt{(a(w))^2+(b(w))^2}$	Done
Define c(w)= $\sqrt{(a(w))^2+b(w)^2}$	Done
MAIN	RAD AUTO FUNC 3/30

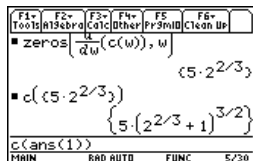
4. 使用 **zeros()** 函数计算 **c(w)** 的一阶导数的零点，从而得到 **c(w)** 的最小值点。

注：杆的最大长度即 **c(w)** 的最小值。



5. 计算杆的最大长度的精确数值。

输入：**c** (**2nd** **[ANS]**)

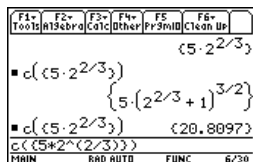


6. 计算杆的最大长度的近似值。

结果：近似为 **20.8097** 英尺。

注：使用自动粘贴功能将计算结果从步骤 4 起中复制到输入行里中 **c()** 的括号内并按下 **◆**

**[ENTER]**。



## 推导一元二次方程求解公式

本例为您说明如何推导出一元二次方程的求解公式：

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

欲了解在本例中所使用之函数的详细信息，可参考符号运算一节。

## 执行运算来推导二次方程求解公式

执行以下步骤对一般形式的一元二次方程进行配方，以推导出一元二次方程的求解公式。

1. 清除当前文件夹中的所有单字符变量。

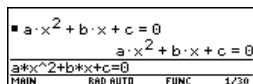
**◆** **[2nd]** **[F6]**

**◆** **[F6]**

选择 **1:Clear a-z** 并按 **[ENTER]** 键确认。

2. 在主屏幕上，输入一般形式的一元二次方程：

$$ax^2 + bx + c = 0.$$



3. 方程两边同时减去  $c$ 。

  $2\text{nd}$   $[\text{ANS}]$   $\square$   $[\alpha]$   $\text{C}$

  $2\text{nd}$   $[\text{ANS}]$   $\square$   $\text{C}$

**注意：**本例使用上面的结果在 TI-89 Titanium / Voyage™ 200 上进行运算。该功能可以减少键击次数和错误出现的几率。

$$\begin{aligned} & \blacksquare (a \cdot x^2 + b \cdot x + c = 0) - c \\ & \qquad \qquad \qquad a \cdot x^2 + b \cdot x = -c \\ \text{ans}(1) - c & \\ \text{MAIN} & \quad \text{RND AUTO} \quad \text{FUNC} \quad 2/20 \end{aligned}$$

4. 方程两边同时除以二次项系数  $a$ 。

**注意：**从步骤 4 到步骤 9 应一直使用步骤 3 中得到的结果 ( $2\text{nd}$   $[\text{ANS}]$ )。

$$\begin{aligned} & \blacksquare \frac{a \cdot x^2 + b \cdot x = -c}{a} \\ & \qquad \qquad \qquad x \cdot (a \cdot x + b) = \frac{-c}{a} \\ \text{ans}(1)/a & \\ \text{MAIN} & \quad \text{RND AUTO} \quad \text{FUNC} \quad 3/20 \end{aligned}$$

5. 使用 **expand()** 函数展开上面的结果。

$$\begin{aligned} & \blacksquare \text{expand}\left(\frac{x \cdot (a \cdot x + b) = -c}{a}\right) \\ & \qquad \qquad \qquad x^2 + \frac{b \cdot x}{a} = \frac{-c}{a} \\ \text{expand}(\text{ans}(1)) & \\ \text{MAIN} & \quad \text{RND AUTO} \quad \text{FUNC} \quad 4/20 \end{aligned}$$

6. 方程两边同时加上  $((b/a)/2)^2$  以完成配方。

$$\begin{aligned} & \blacksquare \left(x^2 + \frac{b \cdot x}{a} = \frac{-c}{a}\right) + \left(\frac{b}{2}\right)^2 \\ & \qquad \qquad \qquad x^2 + \frac{b \cdot x}{a} + \frac{b^2}{4 \cdot a^2} = \frac{b^2}{4 \cdot a^2} - \frac{c}{a} \\ \text{ans}(1) + ((b/a)/2)^2 & \\ \text{MAIN} & \quad \text{RND AUTO} \quad \text{FUNC} \quad 5/20 \end{aligned}$$

7. 使用 **factor()** 函数进行因式分解。

$$\begin{aligned} & \blacksquare \text{factor}\left(x^2 + \frac{b \cdot x}{a} + \frac{b^2}{4 \cdot a^2} = \frac{-c}{a}\right) \\ & \qquad \qquad \qquad \left(\frac{2 \cdot a \cdot x + b}{2 \cdot a}\right)^2 = \frac{-(4 \cdot a \cdot c - b^2)}{4 \cdot a^2} \\ \text{factor}(\text{ans}(1)) & \\ \text{MAIN} & \quad \text{RND AUTO} \quad \text{FUNC} \quad 6/20 \end{aligned}$$

8. 方程两边同时乘以  $4a^2$ 。

$$\begin{aligned} & \blacksquare 4 \cdot a^2 \cdot \left(\frac{2 \cdot a \cdot x + b}{2 \cdot a}\right)^2 = \frac{-(4 \cdot a \cdot c - b^2)}{4} \\ & \qquad \qquad \qquad (2 \cdot a \cdot x + b)^2 = -(4 \cdot a \cdot c - b^2) \\ 4a^2 * \text{ans}(1) & \\ \text{MAIN} & \quad \text{RND AUTO} \quad \text{FUNC} \quad 7/20 \end{aligned}$$

9. 在满足  $a > 0$  且  $b > 0$  且  $x > 0$  的约束条件下对方程两边求平方根。

$$\begin{aligned} & \blacksquare \sqrt{(2 \cdot a \cdot x + b)^2} = \sqrt{-(4 \cdot a \cdot c - b^2)} \\ & \qquad \qquad \qquad \sqrt{(2 \cdot a \cdot x + b)^2} = \sqrt{-(4 \cdot a \cdot c - b^2)} \\ & \qquad \qquad \qquad 2 \cdot a \cdot x + b = \sqrt{b^2 - 4 \cdot a \cdot c} \\ & \qquad \qquad \qquad (1) | a > 0 \text{ and } b > 0 \text{ and } x > 0 \\ \text{MAIN} & \quad \text{RND AUTO} \quad \text{FUNC} \quad 8/20 \end{aligned}$$

10. 方程两边同时减去  $b$  后再除以  $2a$ ，从而解出  $x$ 。

$$\begin{aligned} & \blacksquare (2 \cdot a \cdot x + b = \sqrt{b^2 - 4 \cdot a \cdot c}) - b \\ & \qquad \qquad \qquad 2 \cdot a \cdot x = \sqrt{b^2 - 4 \cdot a \cdot c} - b \\ \text{ans}(1) - b & \\ \text{MAIN} & \quad \text{RND AUTO} \quad \text{FUNC} \quad 9/20 \end{aligned}$$

**注：**由于步骤 9 中的条件约束，本例仅解出二次方程的 2 个通解中的 1 个。

$$2 \cdot a \cdot x = \frac{\sqrt{b^2 - 4 \cdot a \cdot c - b}}{2 \cdot a}$$

$$x = \frac{\sqrt{b^2 - 4 \cdot a \cdot c - b}}{2 \cdot a}$$

ans(1)/(2a)

## 探索矩阵

本例将演示几种矩阵运算。

### 探索一个 3x3 的矩阵

按以下步骤生成一个随机矩阵，进行增广运算生成单位阵，并找出使其逆阵无效的一个元素值。

1. 在主屏幕上，使用 **RandSeed** 将随机数发生器的种子设定为出厂缺省值，然后使用 **randMat()** 生成一个 3x3 的随机矩阵并存放到 **a** 中。

RandSeed 0 Done  
 randMat(3,3) → a

9	-3	-9
4	-2	0
-7	8	8

randmat(3,3)→a

2. 使用变量 **x** 来替换矩阵中的 **[2,3]** 元素，然后使用 **augment()** 函数，将一个 3x3 的单位阵添加到 **a** 的右侧，并将其存放到 **b** 中。

x → a[2,3] x  
 augment(a, identity(3)) → b

9	-3	-9	1	0	0
4	-2	x	0	1	0
-7	8	8	0	0	1

augment(a, identity(3))→b

3. 使用 **rref()** 对矩阵 **b** 进行“行初等变换”：运算结果的前 3 列即为单位阵，而后 3 列为 **a<sup>-1</sup>**。

$$\begin{bmatrix} 1 & 0 & 0 & 8/51 & -17/17 \cdot x + 96 \\ 0 & 1 & 0 & 18/17 \cdot (17 \cdot x + 70) + 7 \\ 0 & 0 & 1 & -6 \end{bmatrix}$$

rref(b)

**注：**可使用光标在历史区域中滚动屏幕以查看结果。

4. 求解使矩阵不可逆的 **x** 值。

输入：

**solve(getDenom( [2nd] [ANS] [1,4] )=0,x)**

结果：**x = -70/17**

solve getDenom

1	0	0	ε
0	1	0	→
0	0	1	-
			x = -70/17

getDenom(ans(1)[1,4])=0,x

**注：**可用光标在历史区域中滚动屏幕以查看结果。

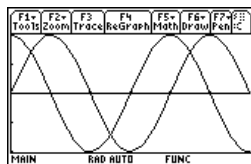
## 求解 $\cos(x) = \sin(x)$

本例采用 2 种方法求方程 **cos(x) = sin(x)** 在 0 到  $3\pi$  间的解 **x**。

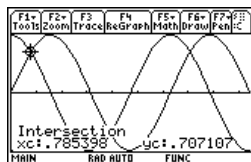
## 方法 1：作图法

按以下步骤找出函数图形  $y_1(x)=\cos(x)$  和  $y_2(x)=\sin(x)$  的交点。

1. 在 **Y= Editor** 中，设  $y_1(x)=\cos(x)$  和  $y_2(x)=\sin(x)$ 。
2. 在 **Window Editor** 中，设  $x_{\min}=0$  和  $x_{\max}=3\pi$ 。
3. 按下 **[F2]** 并选择 **A:ZoomFit**。
4. 找出两函数图形的交点。



**注：**按下 **[F5]** 并选取 **5:Intersection**。根据屏幕上的提示选择两曲线，以及交点 **A** 的上下界限。

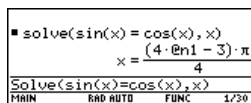


5. 标出 **x** 和 **y** 坐标。(重复步骤 4 和 5 以找出其它交点。)

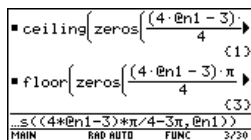
## 方法 2：符号操作

按下列步骤求方程  $\sin(x)=\cos(x)$  的解 **x**。

1. 在主屏幕上，输入 **solve(sin(x)=cos(x),x)**。  
求解当 **@n1** 为任意整数时 **x** 的值。



2. 使用 **ceiling()** 和 **floor()** 函数，分别求出大于等于和小于等于所示交点数值的最近整数。



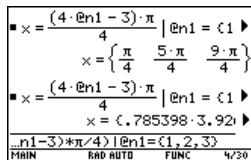
**注：**将光标移动到历史区域中，使最后一个答案处于高亮。按 **[ENTER]** 复制通解。

3. 输入 **x** 的通解并对 **@n1** 采用如图所示的约束条件。

将该结果与 方法 1 所求结果进行比较。

**注：**使用 **with** 算子的方法如下：

**[2nd] [I]; [2nd] [K].**



## 求平行六面体的最小表面积

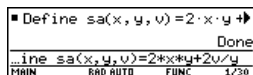
本例演示如何求出体积为常量 **V** 的平行六面体的最小表面积。本例中相关步骤的详细说明可参见 **符号操作** 和 **3D 作图**。

## 绘制平行六面体表面积函数的 3D 图形

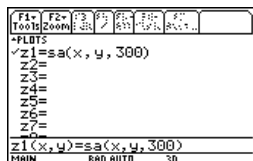
按下列步骤定义平行六面体的表面积，作出 3D 图形，并使用 **Trace** 工具找出与最小表面积处最接近的点。

1. 在主屏幕上，定义函数 **sa(x,y,v)** 表示平行六面体的表面积。

输入：**define sa(x,y,v)=2\*x\*y + 2v/x+2v/y**

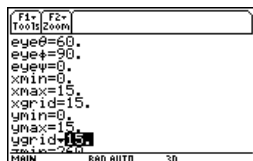


2. 选取 3D 图形模式。然后输入函数 **z1(x,y)**，并按本例所示选取体积 **v=300**。

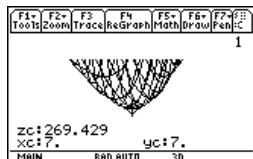


3. 设定 Window 变量为：

**eye= [60,90,0]**  
**x= [0,15,15]**  
**y= [0,15,15]**  
**z= [260,300]**  
**ncontour= [5]**



4. 作出函数图形，并使用 **Trace** 获取与表面积函数最小值最接近的点。

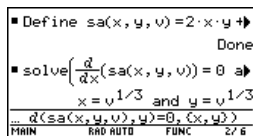


## 使用解析法求出最小表面积

按以下步骤用解析法在主屏幕上求解该问题。

1. 根据 **v** 解出 **x** 和 **y**。

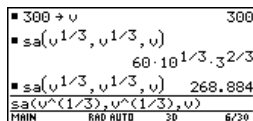
输入：**solve(d(sa(x,y,v),x)=0 and d(sa(x,y,v),y)=0,{x,y})**



2. 求当 **v** 等于 300 时的最小表面积。

输入：**300>v**

输入：**sa(v^(1/3), v^(1/3),v)**



**注：**按下 **[ENTER]** 获取以符号形式表示的精确值。按下 **[◊][ENTER]** 获取以十进制形式表示的近似值。

## 用 **Text Editor** (文本编辑器) 运行教学脚本

本例演示如何使用 **Text Editor** 来运行教学脚本。

### 运行教学脚本

按下列步骤使用 **Text Editor** 编写脚本，试运行每一行脚本，并在主屏幕的历史区域中观察运行结果。

1. 打开 **Text Editor**，并创建一个新变量 **demo1**。



注：命令符 **C** 可通过 **F2 1:Command** 工具条菜单获取。

2. 在 **Text Editor** 中键入以下内容：

: Compute the maximum value of  $f$  on the closed interval  $[a,b]$  (计算闭区间  $[a,b]$  内  $f$  的最大值)

: assume that  $f$  is differentiable on  $[a,b]$  (假设  $f$  在  $[a,b]$  内可微)

: define  $f(x)=x^3-2x^2+x-7$

C : 1→a:3.22→b

C : d(f(x),x)→df(x)

C : zeros(df(x),x)

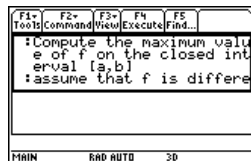
C : f(ans(1))

C : f([a,b])

C : The largest number from the previous two commands is the maximum value of the function. The smallest number is the minimum value (前 2 条命令得到的最大数值即为函数的最大值。同样，得到的最小数值即为函数的最小值)。

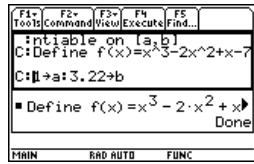


3. 按下 **F3** 并选取 **1:Script view** 在拆分式屏幕上显示 **Text Editor** 和主屏幕。将光标移动到 **Text Editor** 的第一行。



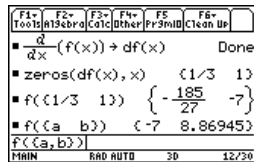
4. 重复按下 **[F4]**，使得每次只执行脚本中的一行。

**注：**按下 **[F4]** 并选取 **2:Clear split** 返回到显示整个 **Text Editor** 的屏幕。



5. 要查看在全屏模式下的脚本结果，应返回主屏幕。

**注：**连按 **[2nd]** **[QUIT]** 2 次以显示主屏幕。



## 有理函数分解

本例演示了将有理函数分解成商式和余式的形式。与本例中所含各步骤有关的详细说明可参见 **基本函数作图** 和 **符号运算**。

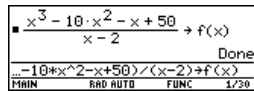
### 有理函数分解

通过图形验证对有理函数  $f(x)=(x^3-10x^2-x+50)/(x-2)$  的分解。

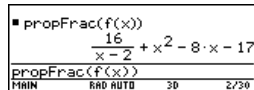
1. 在主屏幕上，输入如下所示的有理函数并将其存为函数 **f(x)**。

输入： $(x^3-10x^2-x+50)/(x-2) \rightarrow f(x)$

**注：**实际输入会在本例屏幕上以反转字体显示出来。



2. 使用真分数函数 (**propFrac**) 将该函数分解为商式和余式的形式。

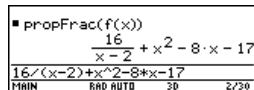


3. 将上面的结果复制到输入行中。

– 或者 –

输入： $16/(x-2)+x^2-8x-17$

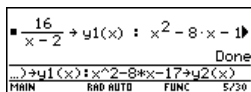
**注：**将光标移动到历史区域中，使上面的结果以高亮方式显示。按下 **[ENTER]** 将最后答案复制到输入行中。



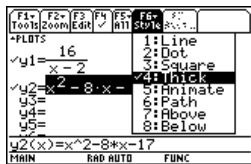


4. 在输入行中编辑上面的结果。如图所示，将余式存放到  $y_1(x)$  中，商式存放到  $y_2(x)$  中。

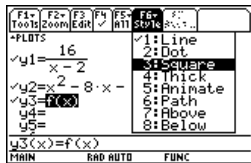
输入： $16/(x-2) \rightarrow y_1(x)$ ； $x^2-8*x-17 \rightarrow y_2(x)$



5. 在 **Y=Editor** 中，为  $y_2(x)$  选取粗线条绘图样式。

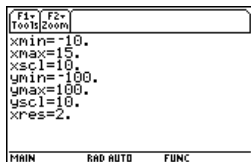


6. 将原函数  $f(x)$  定义为  $y_3(x)$ ，并选取方块绘图样式。



7. 在 **Window Editor** 中，将窗体变量设为：

$x = [-10, 15, 10]$   
 $y = [-100, 100, 10]$

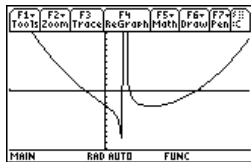
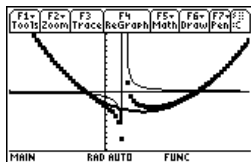


8. 绘制图形。

注：必须将绘图模式设为“函数”。

观察： $f(x)$  的图形特征基本上可以用二次商式  $y_2(x)$  的图形特征来表示。当  $x$  值在正、负两个方向上很大时，该有理表达式基本上可以看成是一个二次函

下方的图形为使用线段样式单独作出的  $y_3(x)=f(x)$  的图形。



数。

## 统计研究：数据分类筛选

本例演示了如何使用数据分类筛选的方法来对高中学生的体重统计数据进行研究。

### 分类过滤数据

每位学生均根据其性别和年级（一年级、二年级、三年级、四年级）归入 8 个类别中的某一个。各相关数据（体重，单位为磅）及对应的分类均输入到 **Data/Matrix Editor** 中。

表 1：分类及说明

分类 (C2)	学年和性别
1	一年级男生
2	一年级女生
3	二年级男生
4	二年级女生
5	三年级男生
6	三年级女生
7	四年级男生
8	四年级女生

表 2：C1（各学生体重，单位为磅）— C2（分类）

C1	C2	C1	C2	C1	C2	C1	C2
110	1	115	3	130	5	145	7
125	1	135	3	145	5	160	7
105	1	110	3	140	5	165	7
120	1	130	3	145	5	170	7
140	1	150	3	165	5	190	7
85	2	90	4	100	6	110	8
80	2	95	4	105	6	115	8
90	2	85	4	115	6	125	8
80	2	100	4	110	6	120	8
95	2	95	4	120	6	125	8

按以下步骤对高中学生的体重根据不同学年进行比较。

1. 启动 **Data/Matrix Editor**，并创建新 Data 变量 **students**。

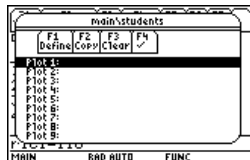


2. 将表2中的数据和分类分别输入到列 **c1** 和 **c2**。

F1 Tool	F2 Plot Setup	F3 C1	F4 Header	F5 Col Cont	F6 F7 Plot
DATA		c1	c2	c3	
4	120	1			
5	140	1			
6	85	2			
7	80	2			
r7c2=2					
MAIN RAD AUTO FUNC					

3. 打开 **[F2] Plot Setup** 工具菜单。

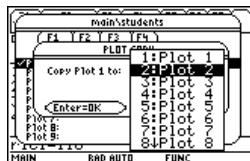
注：采用若干箱形图来比较整个体重数据集的不同子集。



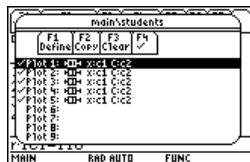
4. 如本屏幕所示，给 **Plot 1** 定义绘图和过滤器参数。



5. 将 **Plot 1** 复制到 **Plot 2**。



6. 重复步骤 5 将 **Plot 1** 复制到 **Plot 3, Plot 4,** 和 **Plot 5**。



7. 按下 **[F1]**，将 **Plot 2** 到 **Plot 5** 的 **Include Categories** 项进行如下修改：

**Plot 2: {1,2}**

(一年级男生、女生)

**Plot 3: {7,8}**

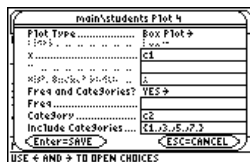
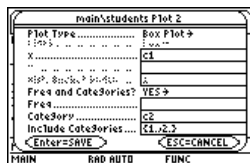
(四年级男生、女生)

**Plot 4: {1,3,5,7}**

(所有男生)

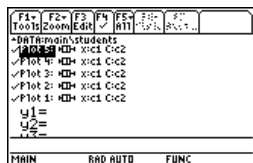
**Plot 5: {2,4,6,8}**

(所有女生)

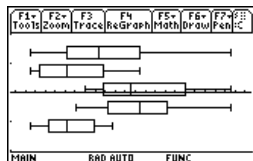


8. 在 **Y= Editor** 中，取消在前一示例中选取的任何功能。

注：只有 **Plot 1** 到 **Plot 5** 可供选取。

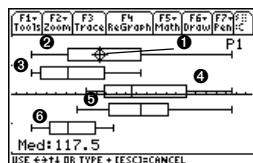


9. 按下 **[F2]** 并选取 **9:Zoomdata** 来显示条状图。



10. 使用 **Trace** 工具来比较不同子集中学生的平均体重。

- ① 平均，所有学生
- ② 所有学生
- ③ 所有一年级学生
- ④ 所有四年级学生
- ⑤ 所有男生
- ⑥ 所有女生



## TI-89 Titanium / Voyage™ 200 上的 CBL 2™ 程序

本例提供的程序可以在 TI-89 Titanium / Voyage™ 200 与某个 Calculator-Based Laboratory™ (CBL 2™) 系统连接时使用。该程序用于“Newton’s Law of Cooling”（牛顿冷却法则）实验中，与 *CBL System Experiment Workbook* 中的“Coffee To Go”实验非常相似。您可以用计算机键盘输入一定长度的文字，然后使用 TI Connect™ 软件将其发送给计算器。在 TI 的网页 [education.ti.com](http://education.ti.com) 上可以找到更多的 CBL 2™ 程序。

程序指令	说明
:cooltemp()	程序名
:Prgm	
:Local i	声明本地变量；仅在运行时可用。
:setMode("Graph", "FUNCTION")	将 TI-89 Titanium / Voyage™ 200 设定为函数作图模式。
:PlotsOff	关闭此前作的任何图形。
:FnOff	关闭此前定义的任何函数。
:ClrDraw	清除此前在屏幕上绘制的任何项目。

程序指令	说明
:ClrGraph	清除此前绘制的任何图形。
:ClrIO	清空 TI-89 Titanium / Voyage™ 200 的程序 IO (输入 / 输出) 屏幕。
:-10→xmin:99→xmax:10→xscl	设定 Window 变量。
:-20→ymin:100→ymax:10→yscl	
:{0}→data	创建和 / 或清除 名称为 data 的条目。
:{0}→time	创建和 / 或清除 名称为 time 的条目。
:Send{1,0}	发送命令以清空 CBL 2™ 系统。
:Send{1,2,1}	将 CBL 2™ 中的 Chan. 2 设定为 AutoID, 并记录到 temp 中。
:Disp "Press ENTER to start"	提示用户按下 [ENTER]。
:Disp "graphingTemperature."	
:Pause	等待至用户准备好开始启动。
:PtText "TEMP(C)",2,99	在图形上标出 y 轴。
:PtText "T(S)",80,-5	在图形上标出 x 轴。
:Send{3,1,-1,0}	向 CBL 2™ 发送 Trigger 指令; 实时采集数据。
:For i,1,99	重复后面 2 条读取温度值的指令共 99 次。
:Get data[i]	从 CBL 2™ 中获取温度值并存入一个列表中。
:PtOn i,data[i]	将温度数据绘制在图上。
:EndFor	
:seq(i,i,1,99,1)→time	创建一个条目, 代表时间或数据的采样次数。
:NewPlot 1,1,time,data,,,,4	使用 NewPlot 和 Trace 工具绘制时间和数据。
:DispG	显示图形。
:PtText "TEMP(C)",2,99	重新标出坐标轴。
:PtText "T(S)",80,-5	

:EndPrgm

停止程序运行。

您也可以使用 Calculator-Based Ranger™ 系统 (CBRTM) 求解距离、速度、加速度的数学和物理关系，并了解您运行该例的用时。

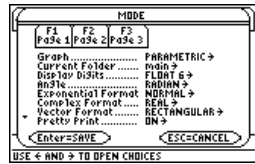
## 研究棒球的飞行轨迹

本例使用屏幕拆分来同时显示参数图形和表格，以便研究棒球飞行轨迹。

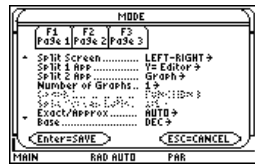
### 设定参数图形和表格

按下列步骤对被击中的棒球飞行轨迹进行研究，条件是球的初速度为 95 英尺 / 秒，起始角为 32 度。

1. 如本屏幕所示，设定 **Page 1** 的模式。



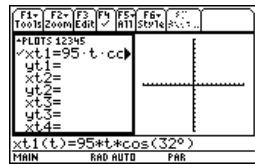
2. 如本屏幕所示，设定 **Page 2** 的模式。



3. 在屏幕左侧的 **Y= Editor** 中，输入描述当飞行时间为 **t** 时，球的飞行距离的方程，用 **xt1(t)** 表示。

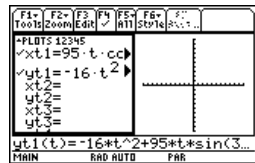
$$xt1(t) = 95 * t * \cos(32^\circ)$$

注：按下  $\boxed{2nd} \boxed{[^\circ]}$  可获得角度符号。



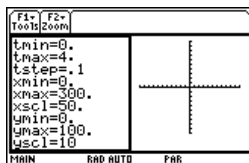
4. 在 **Y= Editor** 中，输入描述当飞行时间为 **t** 时，球的飞行高度的方程，用 **yt1(t)** 表示。

$$yt1(t) = -16 * t^2 + 95 * t * \sin(32^\circ)$$



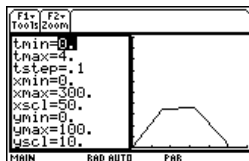
5. 将 Window 变量设定为:

**t values= [0,4,.1]**  
**x values= [0,300,50]**  
**y values= [0,100,10]**



6. 切换至屏幕右侧并显示图形。

注: 按下  $\text{2nd}$   $[\text{ZOOM}]$ 。



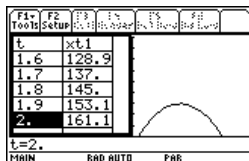
7. 显示 TABLE SETUP 对话框, 将 **tblStart** 改为 **0**, 将  **$\Delta$ tbl** 改为 **0.1**。

注: 按下  $\blacklozenge$   $[\text{TBLSET}]$ 。



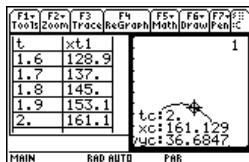
8. 显示屏幕左侧的表格, 并按下  $\odot$  高亮 **t=2**。

注: 按下  $\blacklozenge$   $[\text{TABLE}]$ 。



9. 切换至右侧屏幕。按下  $\text{F3}$ , 对图形进行追踪, 找到当 **tc=2** 时的 **xc** 和 **yc** 值。

注: 当您将追踪光标从 **tc=0.0** 移到 **tc=3.1** 时, 您会看到在 **tc** 时刻球的位置。



## 备选练习

假设初速度还是 95 英尺 / 秒, 求出使球能飞行最大距离的起始角度。

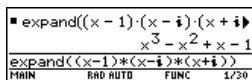
## 直观表示三次多项式的复零点

本例演示了如何用图形表示三次多项式的复零点。

## 直观表示复数根

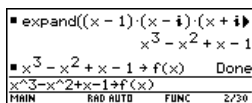
按下列步骤展开三次多项式  $(x-1)(x-i)(x+i)$ ，求出函数的绝对值，作出模面图，并使用 **Trace** 工具对图形进行研究。

1. 在主屏幕上，用 **expand( )** 函数展开三次多项式  $(x-1)(x-i)(x+i)$ ，并观察其一次项。



2. 将最近一次计算结果复制并粘贴到输入行中，并存放于函数 **f(x)** 中。

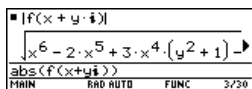
**注：**将光标移动到历史区域中，使最近一次计算结果呈高度显示，然后按下 **[ENTER]**，将其复制到输入行。



3. 使用 **abs( )** 函数求出 **f(x+yi)** 的绝对值。

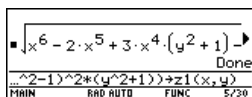
(此运算可能需要 2 分钟左右。)

**注：**一元函数的绝对值可使其根值点正好与 **x** 轴相切。同样，二元函数的绝对值也会使根值点正好与 **xy** 平面相切。



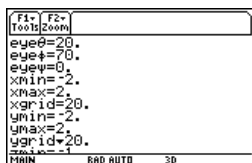
4. 将最近一次计算结果复制并粘贴到输入行中，并存放于函数 **z1(x,y)** 中。

**注：****z1(x,y)** 的图形即为模面。



5. 将设备设定为 3D 图形模式，开启坐标轴的图形格式，并设定 **Window** 变量为：

**eye= [20,70,0]**  
**x= [-2,2,20]**  
**y= [-2,2,20]**  
**z= [-1,2]**  
**ncontour= [5]**

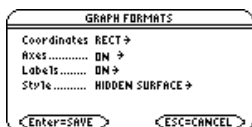


6. 在 **Y=Editor** 中，按下：



并设定 **Graph Format** 变量为：

**Axes= ON**  
**Labels= ON**  
**Style= HIDDEN SURFACE**

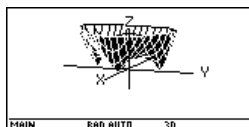


**注：**计算并绘制图形需要约 3 分钟。

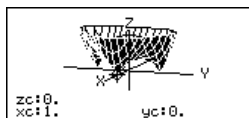


7. 作出模面的图形。

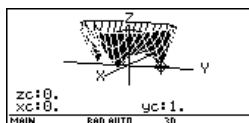
利用 3D 图形可以直观地显示出该面与  $xy$  平面相切处即为根值点。



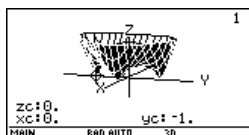
8. 使用 Trace 工具求出当  $x=1$  且  $y=0$  时的函数值。



9. 使用 Trace 工具求出当  $x=0$  且  $y=1$  时的函数值。



10. 使用 Trace 工具求出当  $x=0$  且  $y=-1$  时的函数值。



## 小结

请注意  $zc$  代表步骤 7 到步骤 9 中各函数的零值。因此，多项式  $x^3 - x^2 + x - 1$  的复零点  $1, -i, i$  可以用 3 个点直观表示，这些点正好是模面与  $xy$  平面相切的位置。

## 标准年金问题

本例用于求解年金利率、初始本金、复利期数、以及终值。

### 计算年金利率

按以下步骤求解年金的利率 ( $i$ )，条件是初始本金 ( $p$ ) 为 1,000，复利期数 ( $n$ ) 为 6，而终值 ( $s$ ) 为 2,000。

1. 在主屏幕上，输入方程式以求解  $p$ 。

$\begin{aligned} \blacksquare \text{ solve}(s = p \cdot (1+i)^n, p) \\ p = (i+1)^{-n} \cdot s \\ \text{solve}(s = p \cdot (1+i)^n, p) \end{aligned}$
MAIN RAD AUTO FUNC 1/30

2. 输入方程式以求解  $n$ 。

$\begin{aligned} \blacksquare \text{ solve}(s = p \cdot (1+i)^n, n) \\ n = \frac{\ln(\frac{s}{p})}{\ln(1+i)} \text{ and } \frac{s}{p} > 0 \\ \text{solve}(s = p \cdot (1+i)^n, n) \end{aligned}$
MAIN RAD AUTO FUNC 2/30

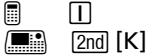
3. 使用 **with** 算子，输入方程式以求解  $i$ 。


**solve(s=p\*(1+i)^n,i) | s=2000 and p=1000  
and n=6**

结果：利率为 12.246%。

注：

- 输入 “with” (|) 算子的方法：



- 按下  **ENTER** 可获得浮点计算结果。

■ solve(s = p · (1 + i) <sup>n</sup> , i)   s = 2000			
i = .122462 or i = -2.12246			
solve(s = p · (1 + i) <sup>n</sup> , i)   s = 2000			
MAIN	RND AUTO	FUNC	B/20

### 计算年金终值

对于上面的例子，若利率为 14%，计算年金终值。

输入方程式以求解  $s$ 。

**solve(s=p\*(1+i)^n,s) | i=.14 and p=1000 and  
n=6**

结果：利率为 14% 时的终值为 2,194.97。

■ solve(s = p · (1 + i) <sup>n</sup> , s)   i = .14			
s = 2194.97			
... i = .14 and p = 1000 and n = 6			
MAIN	RND AUTO	FUNC	9/20

### 计算货币的时间价值

本例中创建了一个函数，可用于计算融资成本。与本例中各步骤相关的详细说明可参见编程。

## 货币的时间价值函数

在 Program Editor 中，定义货币的时间价值函数 (**tvm**) 如下：**temp1** = 付款次数，**temp2** = 年利率，**temp3** = 现值，**temp4** = 月付款额，**temp5** = 终值，且 **temp6** = 支付期头或尾 (1 = 月头，0 = 月尾)。

```
:tvm(temp1,temp2,temp3,temp4,temp5,temp6)
:Func
:Local temp1,tempfunc,tempstr1
:-temp3+(1+temp2/1200 temp6) temp4 ((1-(1+temp2/1200)^
(-temp1))/(temp2/1200))-temp5 (1+temp2/1200)^(-temp1)
>tempfunc
:For temp1,1,5,1
:"temp" &exact(string(temp1))>tempstr1
:If when(#tempstr1=0,false,false,true) Then
:If temp1=2
:Return approx(nsolve(tempfunc=0,#tempstr1) | #tempstr1>0
and
#tempstr1<100)
:Return approx(nsolve(tempfunc=0,#tempstr1))
:EndIf
:EndFor
:Return "parameter error"
:EndFunc
```

**注：**您可以使用键盘在计算机上输入一段较长的文字，然后用 TI Connect™ 软件将其发送到 TI-89 Titanium / Voyage™ 200。

### 求解月付款额

若您按每年 10% 的利率分 48 次进行付款，求解总付款额为 10,000 时的月付款额。

在主屏幕上，输入 **tvm** 值以求解 **pmt**。

结果：月付款额为 251.53。

■ tvn(48, 10, 10000, pmt, 0, 1)			
			251.53
tvm(48, 10, 10000, pmt, 0, 1)			
MAIN	RND AUTO	FUNC	2/30

### 求解付款次数

若您每月支付金额为 300，求解需要多少次才能付清贷款。

在主屏幕上，输入 **tvm** 值以求解 **n**。

结果：付款次数为 38.8308。

■ tvn(n, 10, 10000, 300, 0, 1)			
			38.8308
tvm(n, 10, 10000, 300, 0, 1)			
MAIN	RND AUTO	FUNC	2/30

## 有理数、实数和复数因式分解

本例演示了如何求表达式的有理数、实数和复数因式。与本例中各步骤相关的详细说明可参见符号运算。

### 求解因子

主屏幕上输入如下表达式。

1. **factor(x<sup>3</sup>-5x)** [ENTER] 显示有理数分解结果。

A calculator screen showing the result of the factor function. The top line displays the input expression:  $\text{factor}(x^3 - 5 \cdot x)$ . The second line shows the result:  $x \cdot (x^2 - 5)$ . The bottom status bar shows "MAIN", "RAD AUTO", "FUNC", and "1/30".

2. **factor(x<sup>3</sup>+5x)** [ENTER] 显示有理数分解结果。

A calculator screen showing the result of the factor function. The top line displays the input expression:  $\text{factor}(x^3 + 5 \cdot x)$ . The second line shows the result:  $x \cdot (x^2 + 5)$ . The bottom status bar shows "MAIN", "RAD AUTO", "FUNC", and "2/30".

3. **factor(x<sup>3</sup>-5x,x)** [ENTER] 显示实数分解结果。

A calculator screen showing the result of the factor function with the variable x. The top line displays the input expression:  $\text{factor}(x^3 - 5 \cdot x, x)$ . The second line shows the result:  $x \cdot (x + \sqrt{5}) \cdot (x - \sqrt{5})$ . The bottom status bar shows "MAIN", "RAD AUTO", "FUNC", and "3/30".

4. **cfactor(x<sup>3</sup>+5x,x)** [ENTER] 显示复数分解结果。

A calculator screen showing the result of the cfactor function with the variable x. The top line displays the input expression:  $\text{cFactor}(x^3 + 5 \cdot x, x)$ . The second line shows the result:  $x \cdot (x + \sqrt{5} \cdot i) \cdot (x - \sqrt{5} \cdot i)$ . The bottom status bar shows "MAIN", "RAD AUTO", "FUNC", and "4/30".

## 不放回情况下的抽样模拟

本例模拟从一口缸中取出不同颜色小球，且不放回情形。与本例中各步骤相关的详细说明可参见编程。

## 不放回情况下的抽样函数

在 **Program Editor** 中, 定义 **drawball()** 为双参数函数。第一个参数是一个数列, 数列的每个元素表示某一种颜色小球的数量。第二个参数为选取的小球数量。该函数的返回值为不同颜色小球被选取数量值所组成的列表。

```
:drawball(urnlist,drawnum)           :For j,1,colordim,1
:Func                                  :cumSum(templist)→urncum
:Local templist,drawlist,colordim,   :If pick ≤ urncum[j] Then
    numballs,i,pick,urncum,j         :drawlist[j]+1→drawlist[j]
:If drawnum>sum(urnlist)             :templist[j]-1→templist[j]
:Return "too few balls"              :Exit
:dim(urnlist)→colordim              :EndIf
:urnlist→templist                   :EndFor
:newlist(colordim)→drawlist         :EndFor
:For i,1,drawnum,1                   :Return drawlist
:sum(templist)→numballs             :EndFunc
:rand(numballs)→pick
```

(接右列)

## 不放回条件下的抽样

假设缸中有 **n1** 个小球具有第 1 种颜色、**n2** 个小球具有第 2 种颜色、**n3** 个小球具有第 3 种颜色, 如此类推。模拟在不放回条件下取出小球的情况。

1. 使用 **RandSeed** 命令输入随机数种子。

■ RandSeed 1147	Done
randseed 1147	
MAIN	RND AUTO FUNC 1/30

2. 假设缸中有 10 个红球和 25 个白球, 仿真在不放回条件下从缸中随机取出 5 个球的情况。

输入 **drawball({10,25},5)**。

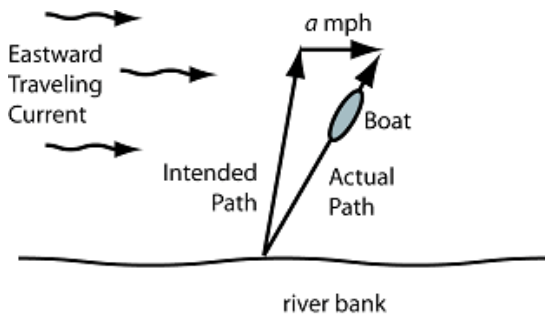
■ drawball({10 25},5)	
	{2 3}
drawball({10,25},5)	
MAIN	RND AUTO FUNC 2/30

结果: 2 个红球 和 3 个白球。

## 用向量求解速度

一只小渔船从 Allegheny 河南岸出发, 航向为  $80^\circ$  角, 航行速度为 20 节 (海里 / 小时)。然而, 由于河水向东流动, 使得船的实际航向与河岸成  $60^\circ$  角。

问: 河水流动速度是多少, 船的实际航行速度又是多少?



- 按该屏幕所示设定 **Page 1** 的模式。(角度以度数而不是以弧度表示,且所有数字均采用浮点十进制表示。)

按下:  $\text{MODE}$   $\downarrow$   $\downarrow$   $\downarrow$ 。在 **Angle** 选项上,选取 **2:DEGREE**。在 **Display Digits** 选项上,选取 **E:FLOAT**。



- 输入向量,以描述船的初始路径、水流、以及船的最终路径。

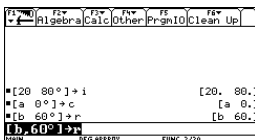
将这些向量分别存为 **i**、**c** 和 **r**。用值 **a** 表示未知的水流速度。用值 **b** 表示船速。

输入:

$[20,80^\circ] \rightarrow i$

$[a,0^\circ] \rightarrow c$

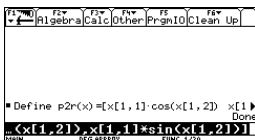
$[b,60^\circ] \rightarrow r$



向量通常可以用极坐标或直角坐标的形式表示,因此将极坐标转换为直角坐标形式是非常有用。

- 定义函数 **p2r**。

输入: **Define p2r(x)=[x[1,1]\*cos(x[1,2]), x[1,1]\*sin(x[1,2])]**



当转换为直角坐标形式时，向量 **i** 和 **c** 的和等于合向量 **r**。

- 使用函数 **p2r**，将 **i**、**c**、和 **r** 转化为直角坐标格式。

输入：

**p2r(i)>i**

**p2r(c)>c**

**p2r(r)>r**

因为上述向量相等，则 **i+c** 的 **x** 坐标值必须等于合向量 **r** 的 **x** 坐标值。同样，**i+c** 的 **y** 坐标值必须等于合向量 **r** 的 **y** 坐标值。

- 建立由向量 **i+c** 和 **r** 构成的 2 个方程式。

- 方程 1 表示它们的 **x** 坐标相等。
- 方程 2 表示它们的 **y** 坐标相等。

将上述方程式分别保存为 **eq1** 和 **eq2**。输入：

**i[1,1]+c[1,1]=r[1,1]>eq1**

**i[1,2]+c[1,2]=r[1,2]>eq2**

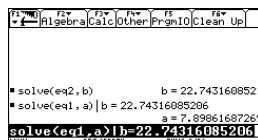
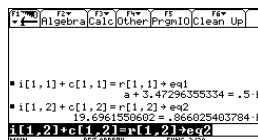
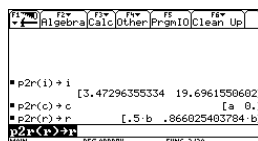
- 求解 **eq2** 得到 **b**，以计算船的实际速度。

**solve(eq2,b)**

- 将已知的 **b** 值代入 **eq1**，从 **eq1** 求解 **a** 以确定河水向东的流速。

**solve(eq1,a) | b=22.74316085206**

船的实际航速是 22.7 节，河水流速约为 7.9 节。







## 连接方法

### 连接两台设备

TI-89 Titanium 和 Voyage™ 200 图形计算器各自配有一根用于与另一台设备相连接的电缆。一旦为两台设备建立起连接，您就可以在其间传递信息。

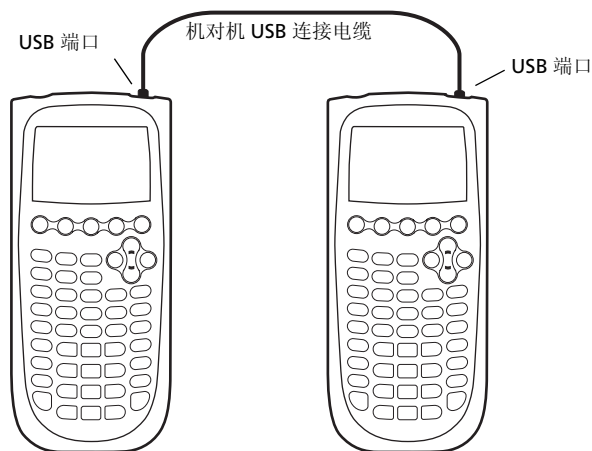
TI-89 Titanium 配有一根机对机 USB 连接电缆；该电缆可以与计算器的 USB 端口配合使用。Voyage™ 200 配有一根机对机 I/O 连接电缆；该电缆可以与计算器的 I/O 端口配合使用。

**注：**TI-89 Titanium 配有一个 USB 端口和一个 I/O 端口，因此您可以将其与配有这两类接口中任何一种的 TI 图形计算器相连接。当然，若使用 I/O 端口，需要配备机对机 I/O 连接电缆（需单独购买）或是用于连接计算机的 USB Silver Edition 电缆（也需单独购买）。

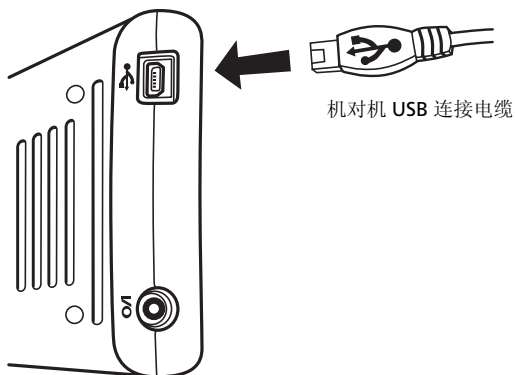
### 发送或接收信息前的连接工作

均匀用力，分别将电缆的两端插入两台计算器的连接端口里。通过在 **VAR-LINK** 屏幕上进行设置，每台计算器均可执行收发。

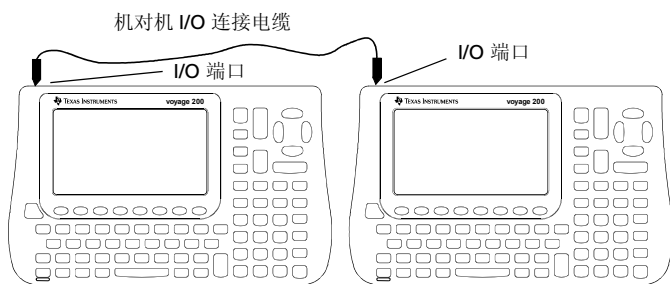
您可以将一台 TI-89 Titanium 或 Voyage™ 200 与另一台 TI-89 Titanium、Voyage™ 200、TI-89、或 TI-92 Plus 相连。



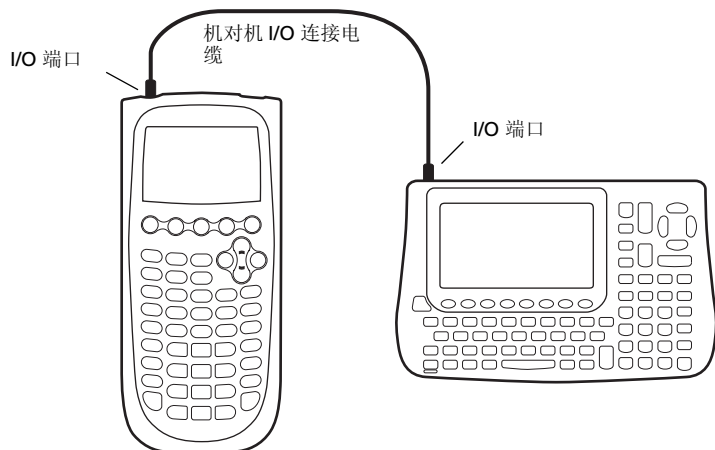
连接在一起的两台 TI-89 Titanium 计算器



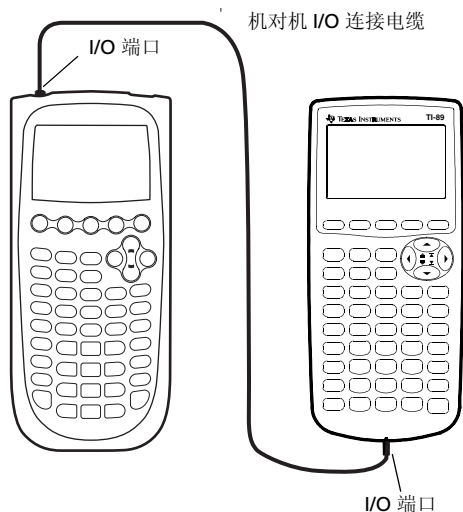
调整相对位置使接头和端口的 USB 符号处于同一面，然后将接头插入。



连接在一起的两台 Two Voyage™ 200 计算器



连接在一起的一台 TI-89 Titanium 与一台 Voyage™ 200



连接在一起的一台 TI-89 Titanium 和一台 TI-89

## 变量、闪存 (Flash) 应用软件和文件夹的传送

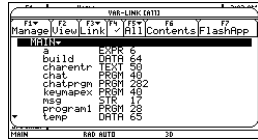
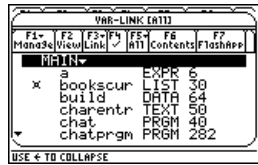
变量传送使您可以方便地共享在 **VAR-LINK** 屏幕上列出的任何变量 — 包括函数、程序和其它变量。您也可以传送闪存应用软件 (Apps) 和文件夹。

### 计算器的配置

闪存应用软件只能在符合条件的计算器之间传送。例如，您可以将闪存应用软件从一台 TI-89 Titanium 传送到另一台 TI-89 Titanium，或是从一台 TI-89 Titanium 传送到一台 TI-89。您可以将应用软件从一台 Voyage™ 200 传送到另一台 Voyage™ 200，或是从一台 Voyage™ 200 到一台 TI-92 Plus。

1. 使用合适的电缆将两台图形计算器连接起来。

2. 在发送设备上，按下 **[2nd][VAR-LINK]** 以显示 **VAR-LINK** 屏幕。

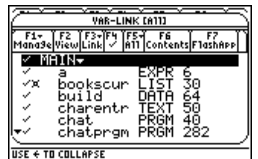


3. 在发送设备上，选择需要发送的变量、文件夹、或是闪存应用软件。

- 如果您需要选择某个变量、闪存应用软件、或文件夹，可移动光标使选择对象呈突出显示，并按下 **[F4]** 在其旁边作一个复选标志 (✓)。



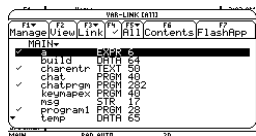
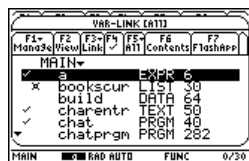
- 在缺省的 **VAR-LINK** 屏幕上，此操作将选择文件夹和其中的内容。多层折叠式文件夹在被选中时将展开。



- 如果您选择了某个闪存应用软件（从表 F7 中），则选择了该应用软件的文件夹以及文件夹中的内容。在文件夹旁边会出现一个复选标志，但在具体内容旁边不会出现。此时多层折叠式文件夹不会自动展开。



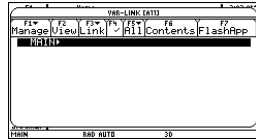
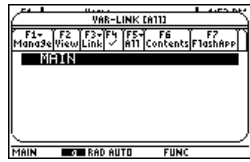
- 如果您需要选择多个变量、闪存应用软件、或文件夹，可使选择对象呈高亮显示，并按下 **[F4]** 在其旁边作一个复选标志 (✓)。如果您决定取消该次传送，可再次使用 **[F4]** 取消选择。



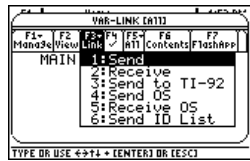
- 如果您需要选择多个变量、闪存应用软件、或文件夹，可使用 **[F5]** **All 1:Select All**。



4. 在接收设备上，按下 **[2nd] [VAR-LINK]** 以显示 **VAR-LINK** 屏幕。（**VAR-LINK** 屏幕上将继续显示出发送设备）。



5. 在接收设备和发送设备上，按下 **[F3] Link** 以显示菜单选项。



6. 在接收设备上，选择 **2:Receive**。

提示信息 **VAR-LINK: WAITING TO RECEIVE** 与 **BUSY** 指示标志将同时显示在接收设备的状态行上。

7. 在发送设备上，选择 **1:Send**

此操作后将开始传送。

在传送过程中，在接受设备的状态行上将显示一个进度条。当传送完成时，接受设备的 **VAR-LINK** 屏幕上的信息将更新。

**注：**在传送某个购买的应用软件时，如果需要，则接收设备必须具备相应的使用许可。该使用许可通常是 TI 提供的文件。免费或演示版的应用软件均无需使用许可。

## 传送变量、闪存应用软件、或文件夹的规则

如果在发送设备和接收设备上有同名的变量，且该变量未经过加锁或存档处理，则接收设备上的变量会被传送过来的同名变量所覆盖。

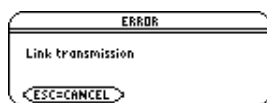
若发送设备和接收设备上有同名的变量，且该变量经过加锁处理，则接收设备上的变量只有当被解锁后，才可以被发送设备传送过来的变量所覆盖。若发送设备和接收设备上有同名的变量，且该变量经过存档处理，则在需要进行变量覆盖时，会有提示信息要求您确认此操作。

若您选择：	后果：
解锁的变量	变量将被传送到当前文件夹，且在接收设备上保持解锁状态。
加锁的变量	变量将被传送到当前文件夹，且在接收设备上保持加锁状态。
存档的变量	变量将被传送到当前文件夹，且在接收设备上保持存档状态。
解锁的闪存应用软件	若接收设备具有相应的使用许可，则闪存应用软件得以被传送。该应用软件在接收设备上保持解锁状态。
加锁的闪存应用软件	若接收设备具有相应的使用许可，则闪存应用软件得以被传送。该应用软件在接收设备上保持加锁状态。
解锁的文件夹	该文件夹和文件夹中被选中的内容会被传送。该文件夹在接收设备上保持解锁状态。
加锁的文件夹	该文件夹和文件夹中被选中的内容会被传送。该文件夹在接收设备上保持解锁状态。

## 取消传送

无论是对于发送设备还是接收设备：

1. 按下 **[ON]**。  
屏幕上显示出错误提示信息。
2. 按下 **[ESC]** 或 **[ENTER]**。

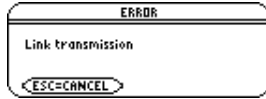


## 常见出错及提示信息

显示在:

信息及其说明:

发送设备



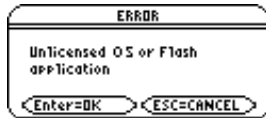
在以下情况出现数秒钟后，屏幕上将会显示上述信息：

- 电缆未正确插入发送设备的连接端口。  
- 或者 -
- 接收设备未正确连接到电缆的另一端。  
- 或者 -
- 接收设备未正确设置为准备接受的状态。

按下 **[ESC]** 或 **[ENTER]** 以取消此次传送。

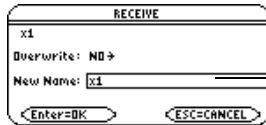
**注：**作为传送设备可能不会一直显示该信息。在您取消传送之前，它仍然会保持显示 **BUSY**。

发送设备



接收设备不具有对所发送的操作系统（OS）或闪存应用程序的相应使用许可。

接收设备



除非您将 **Overwrite** 的值改为 **NO**，<sup>7</sup> 则 **New Name** 将一直处于激活状态。

接收设备上有一个与所传送来的变量同名的变量。

- 欲覆盖现有变量，按下 **[ENTER]**。（缺省情况下，**Overwrite = YES**。）
- 欲将该变量另存为不同的名字，应设置 **Overwrite = NO**。请在 **New Name** 输入框中，键入一个在接收设备里不存在的变量名称。然后连续 2 次按下 **[ENTER]**。
- 欲跳过该变量而对下一个进行处理，可设置 **Overwrite = SKIP** 并按下 **[ENTER]**。
- 欲取消传送，可按下 **[ESC]**。

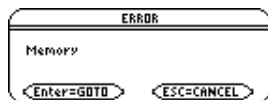


---

**显示在:**                      **信息及其说明:**

---

接收设备



接收设备没有足够的内存用于接受发送来的内容。请按下 **[ESC]** 或 **[ENTER]** 以取消此次传送。

---

## 删除变量、闪存应用软件、或文件夹

1. 按下 **[2nd] [VAR-LINK] to** 以显示 **VAR-LINK** 屏幕。
2. 选择需要删除的变量、文件夹、或闪存应用软件。
  - 欲选择某个变量、闪存应用软件、或文件夹，可移动光标使其呈高亮显示，然后按下 **[F4]** 在其旁边作一个复选标志 (✓)。
    - 若在缺省的 **VAR-LINK** 屏幕上，此操作将选择该文件夹和其中的内容。在选择时，多层可折叠文件夹会自动展开。
    - 若选择了某个闪存应用软件 (从表 **F7** 中)，此操作将选择该应用软件所在的文件夹和其中的内容。在文件夹旁将会出现一个复选标志，但不会出现在具体的内容旁边。多层可折叠的闪存应用软件文件夹不会自动展开。
  - 注:** 您无权删除 **Main** 文件夹。
  - 欲删除多个变量、闪存应用软件、或文件夹，可依次使它们呈高亮显示，并按下 **[F4]** 在其旁边作一个复选标志 (✓)。再次使用 **[F4]** 可以取消您已经选择但不想发送的项目。
  - 欲选择所有变量、闪存应用软件、或文件夹，可采用 **[F5] All 1:Select All**。
3. 按下 **[F1]** 并选择 **1:Delete**。
  - 或 -
  - 按下 **[←]**。屏幕上出现一条确认信息。
4. 按下 **[ENTER]** 以确认该次删除。

## 获取闪存应用软件 (Apps) 的途径

欲了解与可用闪存应用软件相关的最新信息，请访问德州仪器公司的网页 [education.ti.com](http://education.ti.com)。

许多应用软件不再需要使用许可证。若您在将某个应用软件从一台计算器传送到另一台时遇到提示信息 **Unlicensed OS or Flash application**，请重新从德州仪器公司的网页 [education.ti.com](http://education.ti.com) 上下载该应用软件。

---

您可从德州仪器公司的网页上将某个闪存应用软件和 / 或使用许可下载到一台计算机上，并使用带 USB 接口的计算机连接电缆或 USB Silver Edition 连接电缆将该应用或使用许可安装到您的 TI-89 Titanium / Voyage™ 200 图形计算器上。

欲了解与闪存应用软件安装相关的信息，可参见 [education.ti.com/guides](http://education.ti.com/guides)。

## 程序控制下的变量传送

您可以使用包含 **GetCalc** 和 **SendCalc** 命令的程序将变量从一台设备传到另一台。

**SendCalc** 将变量发送到连接端口，与端口相连接的设备可接收该变量。该连接设备必须显示在主屏幕上，或必须从某个程序中执行 **GetCalc**。

您可以使用与 **SendCalc** 或 **GetCalc** 命令相关的可选参数来指定 USB 端口或 I/O 端口。(详细说明请参见附录 A。)若您未使用这些相关参数，TI-89 Titanium 将通过 USB 端口进行通讯。

## “会话 (Chat)” 程序

以下程序使用了 **GetCalc** 和 **SendCalc**。该程序通过建立双重循环，使得相互连接的两台设备能够交替发送和接收名为 **msg** 的变量。**InputStr** 允许所有使用者将消息输入到变量 **msg** 中。

```

:Chat ()
:Prgm
:ClrIO
:Disp "On first unit to send",""
  enter 1;","On first to receive,"
:InputStr " enter 0",msg
:If msg="0" Then
  :   While true
  ①  :   GetCalc msg
  ②  :   Disp msg
  :   InputStr msg
  :   SendCalc msg
  :   EndWhile
  :Else
  ④  :   While true
  :   InputStr msg
  :   SendCalc msg
  ⑤  :   GetCalc msg
  :   Disp msg
  :   EndWhile
  :EndIf
:EndPrgm

```

注:

- ① 设置此设备来接收和发送变量 **msg**。
- ② 然后让该用户在 **msg** 中输入一条信息并发送出去。
- ③ 接收首条信息的设备执行循环。
- ④ 让此用户在 **msg** 中输入一条信息并发送出去。
- ⑤ 然后设置此设备来接收和显示信息。
- ⑥ 发送首条信息的设备执行循环。

为使 **GetCalc** 和 **SendCalc** 同步，循环被设置成当作为发送方的设备等待用户输入信息时，接收设备将执行 **GetCalc**。

## 运行程序

该过程假设:

- 两台设备通过连接电缆相连接。
- 会话（Chat）程序同时加载到两台设备上。

- 使用各台设备上的 **Program Editor**（程序编辑器）分别输入该程序。
- 或 -
- 在一台设备上输入该程序，然后使用 **VAR-LINK** 将程序变量传送到另一台设备上。

要在两台设备上同时运行该程序：

1. 在各台设备的主屏幕上，输入 **chat()**。
2. 当各台设备屏幕上显示初始提示时，按如下方式进行回复。

显示于：	键入：
将要发送第一个消息的设备。	<b>1</b> 并按下 <b>[ENTER]</b> 。
将要接受第一个消息的设备。	<b>0</b> 并按下 <b>[ENTER]</b> 。

3. 依次键入消息，并按下 **[ENTER]** 将变量 **msg** 发送到另一台设备。

### 停止程序执行

由于 **Chat** 程序在两台设备上同时设置了无限循环，所以必须按下 **[ON]**（在两台设备上）中断程序。若您按下 **[ESC]** 对出错提示信息进行确认，则程序将停在 **Program I/O** 屏幕上。按下 **[F5]** 或 **[ESC]** 返回主屏幕。

## 操作系统 (OS) 升级

您可以使用计算机将 **TI-89 Titanium** 或 **Voyage™ 200** 上的操作系统进行升级。您也可以将操作系统从一台设备上传送到同型号的另一台上（例如，从一台 **TI-89 Titanium** 到另一台 **TI-89 Titanium** 或从一台 **Voyage™ 200** 到另一台 **Voyage™ 200**）。

安装操作系统软件时，设备上的所有内存均会恢复到出厂时的设置状态。即所有的用户自定义变量（存放在 **RAM** 和用户数据文档中）、函数、程序、列表以及文件夹（除 **Main** 文件夹外）均将被删除。闪存应用软件也可能被删除。您需要使用 **TI Connect™** 软件在向计算器上安装新的操作系统前将您的数据备份到计算机上。

在进行操作系统升级之前请参参与电池有关的重要信息。

### 与操作系统下载有关的重要信息

在开始下载操作系统前应该安装新的电池。

在操作系统下载模式下，**Automatic Power Down™**（自动电源切断，**APD™**）功能将不起作用。若您在开始下载过程之前将设备置于下载模式过长，电池将可能会耗尽。这样您在开始下载之前必须更换电池。

若您不小心在传送完成前时其中断，您需要重新安装操作系统。同时，勿忘在开始下载前安装上新的电池。

## 在安装操作系统前对您的设备进行备份

操作系统升级时的安装过程如下：

- 删除所有用户自定义数据（同时从 RAM 和用户数据文档中）、函数、程序、以及文件夹。
- 可以删除所有的闪存应用软件。
- 将所有的系统变量和模式复位到其原始的出厂设置。此操作等同与使用 **MEMORY** 屏幕复位所有内存。

要保留某些现有的变量或闪存应用软件，应在安装升级程序前执行下列步骤：

- **重要：** 安装新电池。
- 将需要保留的变量或闪存应用软件传送到另一台设备上。  
– 或 –
- 使用 USB 电缆 或 TI Connectivity Cable USB 电缆，以及 TI Connect™ 软件 ([education.ti.com/downloadticonnect](http://education.ti.com/downloadticonnect)) 将变量和 / 或闪存应用软件发送给计算机。

## 获取操作系统升级软件的途径

欲了解与现有系统升级相关的最新信息，请访问德州仪器的网页 [education.ti.com/downloadticonnect](http://education.ti.com/downloadticonnect)。

您可以从德州仪器的网页上将操作系统升级软件或闪存应用软件下载到一台计算机上，并使用 USB 计算机电缆 或 USB Silver Edition 电缆 将操作系统或应用软件安装到 TI-89 Titanium / Voyage™ 200 上。

欲了解所有的信息，请参见网页上的指导内容。

## 操作系统的传送

操作系统软件仅可从一台 TI-89 Titanium 传送到另一台 TI-89 Titanium，或从一台 TI-89 传送到另一台 TI-89，或从一台 Voyage™ 200 传送到另一台 Voyage™ 200，或从一台 TI-92 Plus 传送到另一台 TI-92 Plus。

将操作系统 (OS) 从一台设备传送到另一台设备：

1. 连接两台相同的设备，如，一台 TI-89 Titanium 连接到另一台 TI-89 Titanium；或一台 Voyage™ 200 连接到另一台 Voyage™ 200。
2. 在接收设备和发送设备上，按下 **[2nd]** [VAR-LINK] 以显示 **VAR-LINK** 屏幕。
3. 在接收设备和发送设备上，按下 **[F3]** **Link** 以显示屏幕选项。
4. 在接收设备上，选取 **5:Receive OS**。

屏幕上显示出警告信息。按下 **[ESC]** 停止传送，或按下 **[ENTER]** 继续进行。按下 **[ENTER]**，在接收设备的状态行上显示出 **VAR-LINK: WAITING TO RECEIVE** 和 **BUSY**。

5. 在发送设备上，选取 **4:Send OS**。

屏幕上显示出告警信息。按下 **[ESC]** 停止传送，或按下 **[ENTER]** 开始传送。

### 重要说明：

- 对每台接收设备，切记在必要时进行备份，并安装新电池。
- 确认接收设备和发送设备均显示为 **VAR-LINK** 屏幕。

在传送过程中，接收设备显示出传送进度。当传送完成时，

- 发送设备返回 **VAR-LINK** 屏幕。
- 接收设备返回应用软件桌面或主屏幕。您可能需要使用 **[◀] [▶]** (变亮) 或 **[◀] [▲] [▼] [▶]** (变暗) 来调节对比度。

### 请勿尝试在操作系统传送过程中将其取消

在传送开始后，接收设备的现有操作系统会被有效地删除掉。若您在传送完成前将其中断，接收设备将无法正常工作。此时您需要重新安装操作系统的软件。

### 若您同时升级多台设备的操作系统

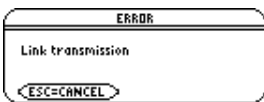
欲在多台设备上进行操作系统的升级，可将操作系统下载并安装到某一台设备上，并将操作系统升级软件从一台设备传送到另外一台。此方法比通过计算机为各台设备分别安装操作系统要快。操作系统的升级软件是免费发放的，因此您无需在下载或安装前获得使用许可。

### 出错提示信息

大部分出错提示信息会显示在发送设备上。根据出错提示信息在传送过程中的发生时间，您可以在接收设备上看到出错提示信息提示。

#### 出错提示信息

#### 说明



发送设备和接收设备未能正确地连接，或是接收设备未被设置为接收模式。

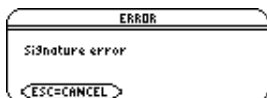


属于接收设备的使用许可并不适用于发送设备上的操作系统或应用软件。您必须获取和安装有效的使用许可。

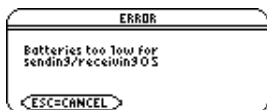
若某个应用软件不再需要使用许可，您可以再次从德州仪器的网页 [education.ti.com](http://education.ti.com) 进行下载，然后在计算器上重新安装应用程序。

## 出错提示信息

## 说明



在传送过程中产生错误。接收设备上的操作系统崩溃。您必须从计算机上重新安装产品软件。



若某台设备显示该信息，则应为其更换电池。

## 采集及传送 ID 列表

**VAR-LINK** 屏幕 **[F3] 6:Send ID List** 菜单选项可让用户从不同的 TI-89 Titanium、TI-89、Voyage™ 200、或 TI-92 Plus 计算器上采集电子 ID 数据。

### ID 列表与团体使用许可

ID 列表特性有助于方便地为团购商业应用采集设备的 ID 号。在 ID 号采集完毕后，将其传送到德州仪器，以便发放一份团体使用许可证。

团体使用许可证允许将购买的软件分发到多台 TI-89 Titanium、TI-89、Voyage™ 200、或 TI-92 Plus 计算器上。只要该软件包含在团体使用许可中，则可以随时将其加载、删除或重载到计算器上。

### 采集 ID 列表

您可以使用一台计算器采集所有的 ID 号，或是使用多台计算器进行采集，然后将采集到的 ID 号统一集中到一台计算器上。

要将 ID 号从一台计算器传送到另一台，需首先使用机对机 USB 连接电缆或机对机 I/O 连接电缆将两台设备连接起来。

### 步骤： 执行位置：

### 执行措施：

1. 采集设备  
(接收设备)

显示主屏幕。按下：  
 **[HOME]**  
 **[CALC HOME]**

2. 发送设备

- a. 按下 **[2nd] [VAR-LINK]** 以显示 **VAR-LINK** 屏幕。
- b. 按下 **[F3] Link** 并选取 **6:Send ID List**。



步骤： 执行位置：	执行措施：
	发送设备将其独有的 ID 号复制到接收设备的 ID 列表中。发送设备一直使用固定的 ID 号，且不允许从设备中删除。
3. 其它设备	重复执行步骤 1 和 2，直到所有的 ID 号都被采集和集中到一台设备上。 如果采集设备上有足够的内存，它可以采集超过 4,000 条 ID 号。

**注：**

- 用户无法在查看发送设备或采集设备上的 ID 列表。
- 每次当 ID 列表被成功地从一台设备传送到另一台上，ID 列表将自动地从发送设备上被删除。
- 若某个 ID 在同一台设备上被重复采集 2 次，则重复的 ID 号将自动从列表中被删除。

**清空 ID 列表**

在将 ID 列表上载到设备之后，采集设备上仍然留有原始信息。因此您可以从该采集设备上将列表上载到其它设备上。

从采集设备上清空 ID 列表：

1. 按下 **[2nd]** [VAR-LINK]，显示 **VAR-LINK** 屏幕。
2. 按下 **[F1]** **Manage** 并选取 **A:Clear ID List**。



**TI-89 Titanium、Voyage™ 200、TI-89 和 TI-92 Plus 间的兼容性**

总体而言，TI-89 Titanium、TI-89、Voyage™ 200 和 TI-92 Plus 的数据和程序互相兼容，仅有少数例外。

TI-89 Titanium 的大部分功能都与 TI-89、Voyage™ 200、和 TI-92 Plus 兼容。TI-89 Titanium 和 TI-89 非常相似，区别是 TI-89 Titanium 内存更大 (可以存放更多应用软件和用户文档)，而 TI-89 Titanium 配备有一个 USB 端口。Voyage™ 200 与 TI-92 Plus 几乎完全相同，只是其内存更大，因此可以存放更多的应用程序 (Apps)。



TI-89 Titanium、TI-89、Voyage™ 200、和 TI-92 Plus 上的所有数据都能兼容，但考虑到各种计算器的屏幕尺寸和键盘布置有差异，且 TI-89 Titanium 上配备有 USB 端口，因此某些专用于一种型号计算器的程序可能在其它型号上不能运行或是不能正常运行。

操作系统的不同也会造成其它不兼容的情况。要下载最新的操作系统版本，可访问德州仪器的网页 [education.ti.com/downloadticonnect](http://education.ti.com/downloadticonnect)。参见第 126 页“操作系统 (OS) 升级”。

### 连接传送表

至 → 从 ↓	TI-89 Titanium	TI-89	Voyage™ 200	TI-92 Plus
<b>TI-89 Titanium</b>	操作系统 应用软件 变量	应用软件 变量	变量	变量
<b>TI-89</b>	应用软件 变量	操作系统 应用软件 变量	变量	变量
<b>Voyage™ 200</b>	变量	变量	操作系统 应用软件 变量	应用软件 变量
<b>TI-92 Plus</b>	变量	变量	应用软件 变量	操作系统 应用软件 变量



## 内存及变量管理

### 内存检查和复位

内存 (**MEMORY**) 屏幕上显示出各种数据类型的所有变量所占内存的大小 (以 bytes 为单位), 无论这些变量是存储在 **RAM** 或是用户数据文档。您也可以使用此屏幕对内存进行复位。

### 显示 **MEMORY** 屏幕

按下 **[2nd] [MEM]**。下图是一台 **Voyage™ 200** 的屏幕拷贝信息。(在您的计算器 **MEMORY** 屏幕上显示出的实际数字可能与之有所不同。)

MEMORY			
Flw RESET			
Expr	61	Text	74
List	90	6DB	0
Matrix	238	Data	0
Function	0	Other	154
Prgm/Asn	269	History	1872
Picture	2241	System	125062
String	0	FlashApp	894315
		Archive	219
Enter=OK		RAM free	131312
		Flash ROM free	1923452

**Prgm/Asn:** 包括为 TI-89 Titanium / Voyage™ 200 设计的程序, 以及您已经载入的汇编语言程序。

**History:** 存放在主屏幕历史区域中的历史数据占用内存的大小。

**FlashApp:** 闪存应用软件的大小。

**RAM free:** 剩余 RAM 的大小。

**Flash ROM free:** 闪存剩余空间的大小。

**注:** 欲显示各变量的大小并确定其是否存放在用户数据文件夹里, 可使用 **VAR-LINK** 屏幕。

要关闭屏幕, 可按下 **[ENTER]**。欲对内存进行复位, 可采用以下步骤。

## 内存复位

在 **MEMORY** 屏幕上:

1. 按下 **[F1]**。
2. 选取合适的项目。



项目	说明
RAM	<b>1:All RAM:</b> 复位 RAM 将从 RAM 中删除所有数据和程序。 <b>2:Default:</b> 将所有系统变量和模式复位到出厂时的初始状态。此操作不会影响到任何用户自定义的变量、函数、或文件夹。
Flash ROM	<b>1:Archive:</b> 复位文档将从闪存中删除所有数据和程序。 <b>2:Flash Apps:</b> 复位闪存应用软件将从闪存中删除所有闪存应用软件。 <b>3:Both:</b> 同时复位文件夹和闪存应用软件将从闪存中删除所有数据、程序和闪存应用软件。
所有内存	复位所有内存将从 <b>RAM</b> 和闪存中删除所有数据、程序和闪存应用软件。

**重要信息:** 要删除单个 (而非所有) 变量, 可使用 **VAR-LINK**。

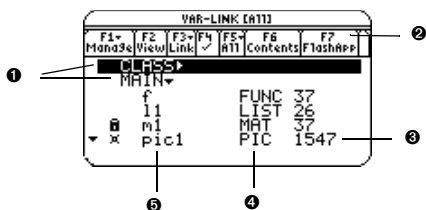
3. 当出现需要进行确认的提示信息时, 按下 **[ENTER]**。  
TI-89 Titanium / Voyage™ 200 在复位完成时均会显示提示信息。  
**注:** 欲取消复位操作, 应按下 **[ESC]** 而不是 **[ENTER]**。
4. 按下 **[ENTER]** 对 ([ENTER]) 提示信息进行确认。

## 显示 **VAR-LINK** 屏幕

**VAR-LINK** 屏幕列出了当前所定义的变量和文件夹。在显示该屏幕后, 您可以对这些变量和文件夹进行相关操作。

### 显示 **VAR-LINK** 屏幕

按下 **[2nd] [VAR-LINK]**。在缺省情况下, **VAR-LINK** 屏幕会列出所有文件夹中所有数据类型的用户自定义变量。



- ❶ 文件夹名称 (按字母表排序)
- ❷ 显示出已安装的闪存应用
- ❸ 以 bytes 为单位的文件大小
- ❹ 数据类型
- ❺ 变量名称 (按字母表排序)

以下 ...	表示 ...
▶	多层可折叠文件夹视图 (在文件夹名的右侧)。
▼	可展开的文件夹视图 (在文件夹名的右侧)。
▼	您可以通过滚动列表而看到更多的变量和 / 或文件夹 (在屏幕的左下角)。
✓	若选取了 [F4]。
🔒	加锁
🗑️	存档

要滚动列表:

- 按下  $\ominus$  或  $\omin�$ 。(使用  $\text{[2nd] } \omin�$  或  $\text{[2nd] } \omin�$  每次滚动一个页面。)
- 或 -
- 键入一个字母。若存在以该字母开头的变量名, 光标将移动至这些变量的第一个变量, 并使其呈高亮显示。

**注:** 重复键入某个字母, 从而对以该字母开头的变量进行遍历。

## 在 VAR-LINK 屏幕上列出的变量类型

类型	说明
<b>ASM</b>	汇编语言程序
<b>DATA</b>	数据
<b>EXPR</b>	表达式（含数值）
<b>FUNC</b>	函数
<b>GDB</b>	图形数据
<b>LIST</b>	数列
<b>MAT</b>	矩阵
<b>PIC</b>	图形
<b>PRGM</b>	程序
<b>STR</b>	字符串
<b>TEXT</b>	文本

在以上未列出的类型均为软件应用自身所用的其它数据类型。

## 关闭 VAR-LINK 屏幕

要关闭 VAR-LINK 屏幕并返回至当前应用程序，可按以下说明使用 **ENTER** 或 **ESC**。

按下：	以便：
<b>ENTER</b>	将高亮变量或文件夹的名称粘贴到当前应用程序的光标位置。
<b>ESC</b>	返回至当前应用程序的光标位置，但不粘贴高亮变量的名称。

## 通过 VAR-LINK 对变量和文件夹进行操作

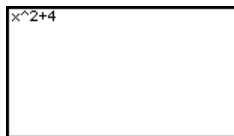
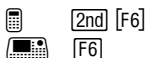
在 VAR-LINK 屏幕上，可以显示变量的内容。您也可以选定一个或多个列表中的项目并按本节提供的方法对其进行操作。

### 显示变量的内容

您可以显示除 **ASM, DATA, GDB, and variables created by Flash Apps** 外的所有变量类型。例如，您必须在 Data/Matrix Editor 中打开一个 **DATA** 变量。

1. 在 **VAR-LINK** 屏幕上，移动鼠标以使该变量呈高亮显示。

2. 按下：



若您高亮某个文件夹，屏幕上将显示出该文件夹中的变量数量。

3. 欲返回 **VAR-LINK**，可按下任意键。

**注：**您无权对本屏幕中的内容进行编辑。

## 从列表选取项目

对于其它操作，选取一个或多个变量和 / 或文件夹。

---

### 欲选取：

### 具体方法：

单个变量或文件夹

移动光标以使该项目呈突出显示，然后按下 **[F4]**。

一组变量或文件夹

使逐个高亮您要选取的项目，然后按下 **[F4]**。✓ 标记出现在该项目的左侧（若您选定了某个文件夹，则也同时选中了该文件中的所有变量。）使用 **[F4]** 选中或取消选取该项目。

所有文件夹和所有变量

按下 **⏴** 以将文件夹展开，然后按下 **[F5]** **All** 并选取 **1:Select All**。



选择 **3:Select Current** 以选取在当前 **VAR-LINK** 会话中传送到您的计算器上的最后一组变量。

选择 **4:Expand All** 或 **5:Collapse All** 以打开或关闭所有文件夹或闪存应用程序。

**注：**当您高亮某个文件夹以后，按下 **⏴** 或 **⏵** 可在打开视图或关闭视图之间进行切换。

## 文件夹和变量

采用文件夹形式，您可以将变量放在相关的群组中，从而方便地进行管理。

TI-89 Titanium / Voyage™ 200 均有一个名称为 **MAIN** 的内建文件夹。除非您另行创建其它文件夹并指定某个用户自定义的文件夹作为当前文件夹，否则所有的变量都会以省略方式存放在 **MAIN** 文件夹中。系统变量或是具有保留名称的变量只能被存储在 **MAIN** 文件夹中。

只能被存放在 **MAIN** 文件夹中的  
变量举例

Window variables

(**xmin**, **xmax**, etc.)

Table setup variables

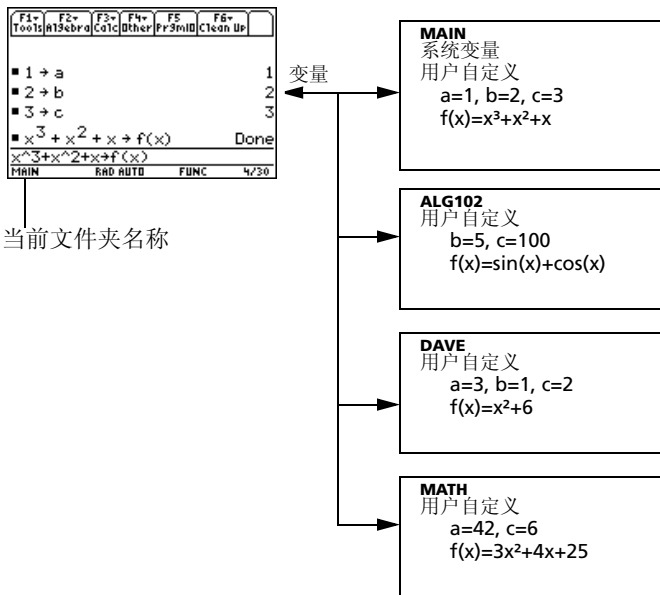
(**TblStart**, **ΔTbl**, etc.)

Y= Editor functions

(**y1(x)**, etc.)

通过创建其它文件夹，您可以存储独立的用户自定义变量（包括用户自定义函数）。例如，你可以为不同的 TI-89 Titanium / Voyage™ 200 应用软件（数学、Text Editor 等）或应用软件分类创建独立的文件夹。您可以在任何现有文件夹中存储用户自定义函数。

一个文件夹中的用户自定义函数与其它任何文件夹中的都是相互独立的。因此，在不同文件夹中可以存储多组同名不同值的变量。



您无权在某个文件夹中创建另一个文件夹。



**MAIN** 文件夹中的系统变量通常都是可以直接取用的，无论其是否存储在当前文件夹中。

**注：**除非您特别指定，否则用户定义的变量存储在“当前文件夹”中。

### 在 **VAR-LINK** 屏幕上创建新的文件夹

1. 按下 **[2nd][VAR-LINK]**。
2. 按下 **[F1] Manage** 并选取 **5:Create Folder**。



3. 键入一个长度最多为8个字母的特有文件夹名称，并连续2次按下 **[ENTER]**。

在您从 **VAR-LINK** 上创建新的文件夹后，该文件夹不会被自动地设置为当前文件夹。

### 在主屏幕上创建文件夹

在主屏幕上输入 **NewFold** 命令。

**NewFold** 文件夹名称

└─ 创建的文件夹名称。此新的文件夹将自动被设置为当前文件夹。

### 在主屏幕上设定当前文件夹

在主屏幕上输入 **setFold** 函数。

**setFold** (文件夹名称)

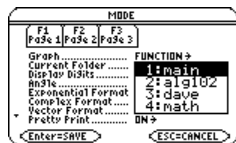
└─ **setFold** 为需要您在括号中输入文件夹名称的函数。

当您执行 **setFold** 时，它将返回前面被设为当前文件夹的文件夹的名称。

### 在 **MODE** 对话框中设定当前文件夹

1. 按下 **[MODE]**。
2. 高亮 **Current Folder** 。
3. 按下 **[↓]** 以显示现有文件夹中的菜单。

**注：**要关闭菜单或退出对话框，且不保存所作的任何修改，可按下 **[ESC]**。



4. 选取适合的文件夹。可以：
  - 高亮某文件夹名并按下 **[ENTER]**。

- 或 -

- 按下与该文件夹对应的数字或字母。

5. 按下 **[ENTER]** 以存储您所作的改变并关闭对话框。

### 重命名变量或文件夹

切记，若您使用 **[F4]** 选取一个文件夹，该文件夹中的变量也自动被选中。如果需要，可使用 **[F4]** 来取消对特定变量的选取。

1. 在 **VAR-LINK** 上，选取变量和 / 或文件夹。
2. 按下 **[F1] Manage** 并选取 **3:Rename**。
3. 键入特有的文件名，并连续 2 次按下 **[ENTER]**。

若您选取了多个项目，您会看到要求您为每个项目输入新名称的提示。



### 使用不同文件夹中的变量

您可以取用未存储在当前文件夹中的用户自定义变量或函数。应指明完整的路径名称，而不能仅仅是变量名称。

路径名称的格式为：

文件夹名 (**folderName**) \ 变量名称

- 或 -

文件夹名 (**folderName**) \ 函数名称

例如：

若当前文件夹 = **MAIN**

文件夹和变量

■ 1 → a	1
■ $x^3 + x^2 + x \rightarrow f(x)$	Done
■ 42 → math\ a	42
■ $3 \cdot x^2 + 4 \cdot x + 25 \rightarrow math \setminus f(x)$	Done
$3 \cdot x^2 + 4 \cdot x + 25 \rightarrow math \setminus f(x)$	
MAIN	RAD AUTO FUNC 4/30

<b>MAIN</b>
a=1
$f(x)=x^3+x^2+x$

■ 4 · a	4
■ 4 · math\ a	168
■ f(5)	155
■ math\ f(5)	120
math\ f(5)	
MAIN	RAD AUTO FUNC 4/30

<b>MATH</b>
a=42
$f(x)=3x^2+4x+25$

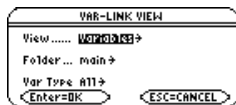
要查看现有的文件夹和变量，可按下 **[2nd] [VAR-LINK]**。在 **VAR-LINK** 屏幕上，您可以使某个变量呈突出显示，并按下 **[ENTER]** 将该变量名粘贴到处于开启状态的应用程序输入行中。若您要粘贴了某个不在当前文件夹中的变量的名称，则路径名称 ( 文件夹名称 \ 变量名称 ) 也同时被粘贴。

## 仅列出某种特定的文件夹和 / 或变量类型，或闪存应用软件

若您的计算器里存有大量的变量、文件夹、或闪存应用软件，则定位某一特定变量会比较困难。通过改变 **VAR-LINK** 屏幕的视图，您可以选取希望看到的特定信息。

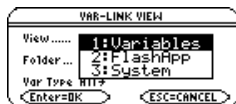
在 **VAR-LINK** 屏幕上：

1. 按下 **[F2] View**。
2. 高亮您要改动的部分，并按下 **[↓]**。此操作将显示出包含有效选项的菜单。（要退出某个菜单，请按下 **[ESC]**。）

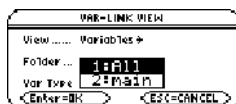


**View**（视图）— 使您可以按需要选择查看变量、闪存应用软件、或系统变量。

注：要列出系统变量（窗口变量等），选取 **3: System**



**Folder**（文件夹）— 一般情况下列出 **1: All** 和 **2: main**，同时也会列出已创建的其它文件夹。



**Var Type**（变量类型）— 列出有效的变量类型。

↓ — 表示您可以滚动屏幕以查看其它变量类型。



3. 选取新的设置。
4. 当您退回到 **VAR-LINK VIEW** 屏幕后，按下 **[ENTER]**。

**VAR-LINK** 屏幕将被刷新，从而仅仅显示出您指定的文件夹、变量类型、或闪存应用软件。

## 将变量从一个文件夹复制或移动到另一个文件夹

除 **MAIN** 之外，您必须建立至少一个文件夹。您无权使用 **VAR-LINK** 在同一个文件夹中复制变量。

1. 在 **VAR-LINK** 屏幕上，选取变量。
2. 按下 **[F1] Manage** 并选取 **2: Copy** 或 **4: Move**。
3. 选取目标文件夹。



4. 按下 **[ENTER]**。被复制或移动的变量会保留原来的名称。

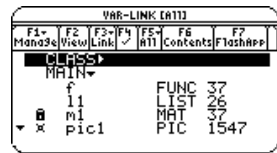
**注：**欲将某个变量以另外的名字复制到同一文件夹里，可使用 **[STO▶]**（例如 **a1→a2**）或在主屏幕中使用 **CopyVar** 命令。

## 对变量、文件夹、或闪存应用软件进行加锁或解锁

当某个变量被加锁时，您无法将其删除、重命名、或赋予新值。然而，您可以复制、移动、或显示其内容。当某个文件夹被加锁时，您可以对该文件夹中的变量进行操作（前提是这些变量未被加锁），但您不能删除该文件夹。当某个闪存应用软件被加锁时，您不能将其删除。

1. 在 **VAR-LINK** 屏幕上，选取变量、文件夹、或闪存应用软件。
2. 按下 **[F1] Manage** 并选取 **6:Lock** 或 **7:UnLock**。

- ☑ 指定 RAM 中加锁的变量或文件夹。
- ☒ 指定某个被存档的变量，其自动被加锁。



## 从 VAR-LINK 屏幕上删除文件夹

当您从 **VAR-LINK** 屏幕上删除某个文件夹时，该文件夹中的所有变量均会被删除。您无权删除 **MAIN** 文件夹。

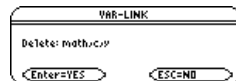
1. 按下 **[2nd] [VAR-LINK]**。
2. 按下 **[F4]** 以选取需要删除的文件夹。（文件夹里的变量被自动选中。）



3. 按下 **[F1] 1:Delete** 或 **[←]**。



4. 按下 **[ENTER]** 以确认删除该文件夹和其中的所有变量。



## 从主屏幕上删除变量或文件夹

在从主屏幕上删除某个文件夹之前，您必须先删除该文件夹中的所有变量。

- 要删除某个变量，在计算器的主屏幕上输入 **DelVar** 命令。

**DelVar** *var1* [, *var2*] [, *var3*] ...

- 要删除某个空文件夹，在计算器的主屏幕上输入 **DelFold** 命令。

**DelFold** *folder1* [, *folder2*] [, *folder3*] ...

**注：**您无权删除 **MAIN** 文件夹。

## 将变量名粘贴至应用软件中

如果您在主屏幕上键入了某个表达式且记不起使用了哪个变量，您可以打开 **VAR-LINK** 屏幕，从列表中选择一个变量，并将该变量名直接粘贴到主屏幕的输入行中。

### 您可使用哪些应用软件？

在下列应用软件中，您可以将某个变量名粘贴到当前光标所在的位置。

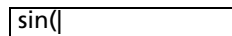
- Home screen、Y= Editor、Table Editor、或 Data/Matrix Editor — 光标必须停留在输入行中。
- Text Editor、Window Editor、Numeric Solver、或 Program Editor — 光标可停留在屏幕上任何位置。

您也可将变量名粘贴到光标多个闪存应用软件中的当前位置。

### 步骤

从上面所列举的某个应用软件开始：

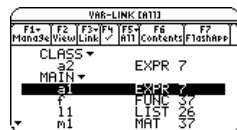
- 将光标定位到需要插入变量名的位置。



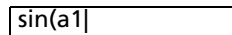
- 按下 **[2nd]** [**VAR-LINK**]。

- 高亮相关变量。

**注：**您也可以高亮文件夹名称并进行粘贴。

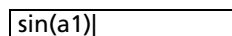


- 按下 **[ENTER]** 以粘贴变量名。



**注：**此操作将粘贴变量名称，而不是其内容。  
使用 **[2nd]** [**RCL**] 而不是 **[2nd]** [**VAR-LINK**] 来调出变量内容。

- 完成表达式的键入。



若您要粘贴某个不在当前文件夹中的变量名，则该变量的路径名称也会被粘贴。

sin(classa2

假设 CLASS 不是当前文件夹，若您高亮了 CLASS 中的 a2 变量，此操作将完成粘贴。

## 变量存档和解除存档

要以交互方式对一个或多个变量进行存档或解除存档，可使用 **VAR-LINK** 屏幕来完成。您可以在主屏幕或某个程序中执行这些操作。

### 为什么需要对变量进行存档？

对用户数据进行存档可以帮助您：

- 将数据、程序、或任何其它变量存储到安全的位置，以避免被不小心编辑或删除。
  - 通过对变量进行存档，可以腾出更多的 **RAM** 空间。例如：
    - 您可以将那些需要查看但不需要编辑或改变的变量进行存档，或是将那些目前不用但需要保留以备今后使用的变量进行存档。
- 注：**您不能对那些使用保留名称的变量或系统变量进行存档。
- 如果您要在 **TI-89 Titanium / Voyage™ 200** 上运行其它的程序，特别是当这些程序很大时，您需要在安装程序前腾出更多的 **RAM** 空间。

更多的 **RAM** 空间可以在某些类型的运算中成倍提高效率。

### 从 **VAR-LINK** 屏幕

进行存档或解除存档：

1. 按下 **[2nd]** [**VAR-LINK**] 以显示 **VAR-LINK** 屏幕。
2. 选取一个或多个变量，这些变量可以位于不同的文件夹。（您可以选择某个文件夹名以选取该文件夹中的所有变量。）

**注：**欲选取单个变量，应高亮该变量。欲选取多个变量，应分别高亮各个变量并按下 **[F4]**  以选择这些变量。

3. 按下 **[F1]** 并选取：

**8:Archive Variable**

– 或 –

**9:Unarchive Variable**



若选取 **8:Archive Variable**，变量将会移动到用户数据文档中。



x = 存档的变量

您可以象查看加锁变量一样查看存档变量。而且，存档变量仍处于原来的文件夹中，只不过是存储在用户数据文档而不是 RAM 中。

**注：**存档变量可以自动加锁。您可以查看该变量，但您不能对其进行编辑或删除。

## 在主屏幕或某个程序中

使用 **Archive** 和 **Unarchiv** 命令：

**Archive** variable1, variable2, ...

**Unarchiv** variable1, variable2, ...

## 若屏幕上出现碎片收集 (Garbage Collection) 信息

若您频繁使用用户数据文档，您可能会遇到碎片收集信息。这通常发生在您试图对某个变量进行存档而存储空间不足时。但是，TI-89 Titanium / Voyage™ 200 会设法重新整理存档变量以腾出存储空间。

## 对碎片收集信息的处理

当您看到右侧的信息时：

- 继续存档操作，可按下 **[ENTER]**。  
- 或 -
- 取消存档，按下 **[ESC]**。



在完成碎片收集后，根据腾出的空间大小，变量可能被存档。若仍然不能存档，您可以将某些变量解除存档后再行尝试。

## 为何不自动进行碎片收集而无需给出提示信息？

信息如下：

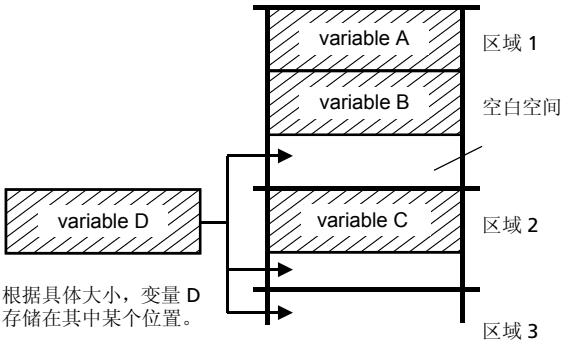
- 让您知道为何文件存档比普通情况需要更长时间。同时警示您在内存不足时存档可能失败。
- 在发现某个程序在循环过程中重复填写用户数据文档时，将给出警示。同时取消此次存档并 查询原因。

## 为什么需要碎片收集？

用户数据文档可分为若干区域。当您首次开始存档时，变量将被顺序存入区域 1，直至该区域全部被存满。若该区域中没有足够的内存，则下一个变量将会存到下一区域的开始位置。通常，这会在上一区域的最后部分留出一段空白。

您存档的每个变量都存放在最近一个具有足够空间的位置。

**注：**每个存档变量存放在单一区域的连续空间里；不得跨区域存放。

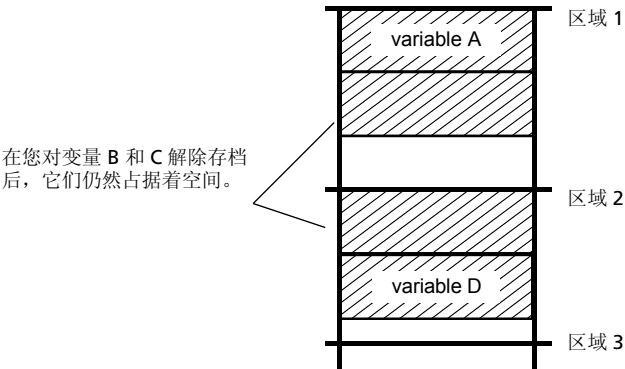


此过程将持续到最后一个区域的末尾。根据变量的具体大小，空白空间可提供大量的可用区域。

**注：**若当您正在存档的变量的大小超过了现有的空白空间，则执行碎片收集。

## 解除变量存档对过程的影响

当您解除变量存档时，该变量被复制到 RAM 中，但未真正从用户数据文档内存中删除。





未存档的变量为“已标记删除”，表示在下一片碎片收集过程中会将它们删除。

## 若 MEMORY 屏幕显示存在足够大的空间

即使 MEMORY 屏幕显示有足够大的空间用于变量存档，您也可能会遇到碎片整理信息。

此 TI-89 Titanium 的 Memory 屏幕显示在删除所有“已标记删除”的变量后将得到的可用空间。

Variable	Size	Address
Exp1	6	008
List	404	004
Matrix	6484	004
Function	23	History
Program	1040	System
Picture	3087	FlashMem
Strings	773	Archive
		RAM free
		Flash RAM free

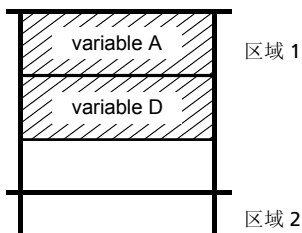
当您解除某变量的存档时，闪存的可用空间立即增加，但在下一次碎片收集完成前该空间并不实际可用。

Voyage™ 200 为用户提供 2.7 MB 的可用闪存。全部 2.7 MB 均可用于闪存应用程序，但其中仅有大约 1 MB 可以用于用户数据存档。

## 碎片收集过程

碎片收集过程如下：

- 从用户数据文档中删除未存档的变量。
- 将剩下的变量重新整理到连续的空间中。



## 查看已存档变量时的内存出错信息 (Memory Error)

对已存档变量的操作与对加锁变量相同。您可以查看该变量，但不能对其进行编辑或删除。某些时候，在您试图查看已存档变量时会遇到 **Memory Error**。

### 出现内存出错信息的原因是什么？

若在查看已存档变量时内存不足，则屏幕上会显示出内存出错信息。这也许会使您困惑，“如果变量在用户数据文档中，又和内存空间有什么关系呢？”原因是仅当变量存放在内存中时，以下操作才可能执行。

- 在 Text Editor 中打开一个文本变量。
- 在 Data/Matrix Editor 中打开一个数据变量、列表、或矩阵。
- 在 Program Editor 中打开一个程序或函数。

- 执行程序或调用函数。

**注：**临时复本能让您打开或执行一个存档变量，但是您不能对该变量作任何修改。

因此您无需对变量解除存档，TI-89 Titanium / Voyage™ 200 能进行“后台”复制。例如，若您执行存放在用户数据文档中的程序，TI-89 Titanium / Voyage™ 200 将：

1. 将程序复制到内存。
2. 运行程序。
3. 在程序运行结束后从内存中删除复本。

若内存空间不足以进行临时复制时，屏幕上将出现出错信息。

**注：**与程序和函数不同，调用某个存档变量时并不会对其进行复制。若变量 **ab** 已存档，在您执行 **6\*ab** 不会有复制的行为。

## 纠正错误

要腾出足够的内存空间来查看变量：

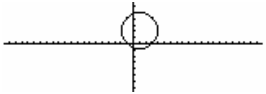
1. 使用 **VAR-LINK** 屏幕 (**2nd** [VAR-LINK]) 来确定您需要查看的存档变量的大小。
2. 使用 **MEMORY** 屏幕 (**2nd** [MEM]) 来查看可用的内存空间大小。
3. 通过以下方法腾出所需的内存量：
  - 从内存中删除不需要的变量。
  - 将较大的变量或程序存档 ( 将其从内存中移动到用户数据文档中 )。

**注：**通常，可用内存空间必须大于需存档的变量的大小。

## 附录 A: 函数和指令

快速查找定位器.....	150
按字母顺序排列的运算列表.....	154

本节提供了有关TI-89 Titanium/ Voyage™ 200计算器的操作系统(OS)中每一条函数和指令的语法及其功能的说明。可参与与计算器应用软件(Apps) 相关的模块以了解这些应用软件中涉及的具体函数和指令。

<p>函数或指令的名称</p>	<p>用于输入名称的按键或菜单。您也可以输入名称。</p>	<p>示例</p>
<p><b>Circle</b>      <b>CATALOG</b></p>		
<p><b>Circle</b> <i>x</i>, <i>y</i>, <i>r</i> [, <i>drawMode</i>]</p> <p>以窗口坐标(<i>x</i>, <i>y</i>) 为圆心、半径为<i>r</i>作圆。 <i>x</i>、<i>y</i>、和<i>r</i>必须是实数。 若 <i>drawMode</i> = 1, 作圆(缺省情况)。 若 <i>drawMode</i> = 0, 关闭已作的圆。 若 <i>drawMode</i> = -1, 沿着圆进行像素反转。</p> <p><b>注目</b> 重新作图将关闭所有已画出的项目</p> <p>自变量显示为 <i>斜体</i>。处于 [ ] 括号中的自变量为可选。切勿键入圆括号.....</p>	<p>在ZoomSqr 视窗中。 ZoomSqr:Circle 1,2,3 [ENTER]</p>	
<p>对函数或指令的说明。</p>		
<p>语法行显示出您所提供的自变量的顺序和类型。请注意用逗号(,)将多个自变量分隔开.....</p>		

# 快速查找定位器

本节列出了以函数群分类的TI-89 Titanium / Voyage™ 200的函数以及其出现的页码。

## 代数

("with")	270	<b>cFactor()</b>	158	<b>comDenom()</b>	161
<b>cSolve()</b>	166	<b>cZeros</b>	169	<b>expand()</b>	182
<b>factor()</b>	184	<b>getDenom()</b>	189	<b>getNum()</b>	190
<b>nSolve()</b>	209	<b>propFrac()</b>	216	<b>randPoly()</b>	222
<b>solve()</b>	237	<b>tCollect()</b>	246	<b>tExpand()</b>	246
<b>zeros()</b>	253				

## 微积分

<b>f()</b> (积分)	265	<b>∏()</b> (乘积)	266	<b>Σ()</b> (求和)	267
<b>arcLen()</b>	156	<b>avgRC()</b>	157	<b>d()</b>	171
<b>deSolve()</b>	173	<b>fMax()</b>	185	<b>fMin()</b>	185
<b>limit()</b>	197	<b>nDeriv()</b>	206	<b>nInt()</b>	208
' (微分符号)	268	<b>seq()</b>	228	<b>taylor()</b>	246

## 绘图

<b>AndPic</b>	155	<b>BldData</b>	158	<b>Circle</b>	159
<b>ClrDraw</b>	160	<b>ClrGraph</b>	160	<b>CyclePic</b>	169
<b>DrawFunc</b>	177	<b>DrawInv</b>	177	<b>DrawParm</b>	178
<b>DrawPol</b>	178	<b>DrawSlp</b>	178	<b>DrwCtour</b>	179
<b>FnOff</b>	185	<b>FnOn</b>	186	<b>Graph</b>	192
<b>Line</b>	197	<b>LineHorz</b>	198	<b>LineTan</b>	198
<b>LineVert</b>	198	<b>NewPic</b>	207	<b>PtChg</b>	216
<b>PtOff</b>	216	<b>PtOn</b>	217	<b>ptTest()</b>	217
<b>PtText</b>	217	<b>PxlChg</b>	217	<b>PxlCrcl</b>	217
<b>PxlHorz</b>	217	<b>PxlLine</b>	218	<b>PxlOff</b>	218
<b>PxlOn</b>	218	<b>pxlTest()</b>	218	<b>PxlText</b>	218
<b>PxlVert</b>	219	<b>RclGDB</b>	222	<b>RclPic</b>	222
<b>RplcPic</b>	226	<b>Shade</b>	233	<b>StoGDB</b>	241
<b>StoPic</b>	241	<b>Style</b>	242	<b>Trace</b>	248
<b>XorPic</b>	252	<b>ZoomBox</b>	255	<b>ZoomData</b>	256
<b>ZoomDec</b>	256	<b>ZoomFit</b>	257	<b>ZoomIn</b>	257
<b>ZoomInt</b>	257	<b>ZoomOut</b>	258	<b>ZoomPrev</b>	258
<b>ZoomRcl</b>	258	<b>ZoomSqr</b>	258	<b>ZoomStd</b>	259
<b>ZoomSto</b>	259	<b>ZoomTrig</b>	259		

## 数组

<b>+</b> (加)	259	<b>-</b> (减)	260	<b>*</b> (乘)	260
<b>/</b> (除)	261	<b>-</b> (求负)	263	<b>^</b> (幂)	261
<b>augment()</b>	156	<b>crossP()</b>	165	<b>cumSum()</b>	168
<b>dim()</b>	176	<b>dotP()</b>	177	<b>expList()</b>	182
<b>left()</b>	196	<b>ΔList()</b>	199	<b>ListMat()</b>	199
<b>matList()</b>	203	<b>max()</b>	203	<b>mid()</b>	204
<b>min()</b>	204	<b>newList()</b>	207	<b>polyEval()</b>	215
<b>product()</b>	216	<b>right()</b>	224	<b>rotate()</b>	224
<b>shift()</b>	233	<b>SortA</b>	240	<b>SortD</b>	240
<b>sum()</b>	242				

## 数学

+	(加)	259	-	(减)	260	*	(乘)	260
/	(除)	261	-	(求负)	263	%	(百分比)	263
!	(阶乘)	265	$\sqrt{()}$	(平方根)	266	^	(幂)	261
°	(度)	268	∠	(角)	268	°, ', "		268
_	(下划线)	269	↔	(转换)	269	10^()		269
Ob, Oh		271	↔Bin		157	↔Cylind		169
↔DD		172	↔Dec		172	↔DMS		177
↔Hex		193	↔Polar		214	↔Rect		222
↔Sphere		240	abs()		154	and		154
angle()		155	approx()		156	ceiling()		158
conj()		162	cos		162	cos <sup>-1</sup> ()		163
cosh()		163	cosh <sup>-1</sup> ()		164	cot()		164
cot <sup>-1</sup> ()		164	coth()		164	coth <sup>-1</sup> ()		164
csc()		165	csc <sup>-1</sup> ()		165	csch()		165
cosh <sup>-1</sup> ()		165	E		179	e^()		179
exact()		181	floor()		185	fPart()		187
gcd()		187	imag()		194	int()		195
intDiv()		195	iPart()		195	isPrime()		196
lcm()		196	ln()		199	log()		201
max()		203	min()		204	mod()		205
nCr()		206	nPr()		209	↔RrX()		211
↔RrY()		211	r	(弧度)	267	↔Rrθ()		221
↔RrPr()		221	real()		222	remain()		223
rotate()		224	round()		225	sec()		226
sec <sup>-1</sup> ()		227	sech()		227	sech <sup>-1</sup> ()		227
shift()		233	sign()		234	sin()		235
sin <sup>-1</sup> ()		236	sinh()		236	sinh <sup>-1</sup> ()		236
tan()		244	tan <sup>-1</sup> ()		244	tanh()		245
tanh <sup>-1</sup> ()		245	tmpCnv()		247	ΔtmpCnv()		248
x <sup>-1</sup>		270						

## 矩阵

+	(加)	259	-	(减)	260	*	(乘)	260
/	(除)	261	-	(求负)	263	+	(点和)	262
.	(点差)	262	.	(点积)	262	./	(点商)	262
.	^ (点幂)	263	^	(幂)	261	augment()		156
colDim()		161	colNorm()		161	crossP()		165
cumSum()		168	det()		175	diag()		175
dim()		176	dotP()		177	eigVc()		180
eigVI()		180	Fill		184	identity()		193
ListMat()		199	LU		202	matList()		203
max()		203	mean()		203	median()		203
min()		204	mRow()		205	mRowAdd()		205
newMat()		207	norm()		208	product()		216
QR		219	randMat()		221	ref()		223
rowAdd()		225	rowDim()		225	rowNorm()		226
rowSwap()		226	rref()		226	simult()		235
stdDev()		241	subMat()		242	sum()		242
T		243	unitV()		249	variance()		250
x <sup>-1</sup>		270						

=	264	≠	264	<	264
≤	265	>	265	≥	265
# (间接引用)	267	> (存储)	271	⊙ (注释)	271
and	154	ans()	156	Archive	156
checkTmr()	159	ClockOff	159	ClockOn	159
ClrErr	160	ClrGraph	160	ClrHome	160
ClrIO	160	ClrTable	161	CopyVar	162
CustmOff	168	CustmOn	168	Custom	169
Cycle	169	dayOfWk()	171	Define	172
DelFold	173	DelVar	173	Dialog	175
Disp	176	DispG	176	DispHome	176
DispTbl	177	DropDown	178	Else	180
Elseif	180	EndCustm	180	EndDlog	180
EndFor	180	EndFunc	180	EndIf	180
EndLoop	180	EndPrgm	180	EndTBar	181
EndTry	181	EndWhile	181	entry()	181
Exec	181	Exit	182	For	186
format()	186	Func	187	Get	187
GetCalc	188	getConfig()	188	getDate()	188
getDtFmt()	189	getDtStr()	189	getFold()	189
getKey()	190	getMode()	190	getTime()	190
getTmFmt()	190	getTmStr()	191	getTmZn()	191
getType()	191	getUnits()	192	Goto	192
if	193	Input	194	InputStr	195
isClkOn()	196	Item	196	Lbl	196
left()	196	Local	200	Lock	200
Loop	202	MoveVar	205	NewFold	207
NewProb	208	not	208	or	210
Output	211	part()	211	PassErr	213
Pause	214	PopUp	215	Prgm	215
Prompt	216	Rename	223	Request	224
Return	224	right()	224	Send	227
SendCalc	227	SendChat	228	setDate()	228
setDtFmt()	228	setFold()	228	setGraph()	228
setMode()	230	setTable()	231	setTime()	231
setTmFmt()	232	setTmZn()	232	startTmr()	240
setUnits()	232	Stop	241	Style	242
switch()	243	Table	244	Text	247
Then	247	timeCnv()	247	Title	247
Toolbar	248	Try	249	Unarchiv	249
Unlock	250	when()	250	While	251
xor	251				

## 统计

!(阶乘)	265	<b>BldData</b>	158	<b>CubicReg</b>	168
<b>cumSum()</b>	168	<b>ExpReg</b>	183	<b>LinReg</b>	198
<b>LnReg</b>	200	<b>Logistic</b>	201	<b>mean()</b>	203
<b>median()</b>	203	<b>MedMed</b>	204	<b>nCr()</b>	206
<b>NewData</b>	206	<b>NewPlot</b>	207	<b>nPr()</b>	209
<b>OneVar</b>	210	<b>PlotsOff</b>	214	<b>PlotsOn</b>	214
<b>PowerReg</b>	215	<b>QuadReg</b>	220	<b>QuartReg</b>	220
<b>rand()</b>	221	<b>randNorm()</b>	221	<b>RandSeed</b>	222
<b>ShowStat</b>	234	<b>SinReg</b>	237	<b>SortA</b>	240
<b>SortD</b>	240	<b>stdDev()</b>	241	<b>TwoVar</b>	249
<b>variance()</b>	250				

## 字符串

& (添加)	265	# (间接引用)	267	<b>char()</b>	159
<b>dim()</b>	176	<b>expr()</b>	183	<b>format()</b>	186
<b>inString()</b>	195	<b>left()</b>	196	<b>mid()</b>	204
<b>ord()</b>	211	<b>right()</b>	224	<b>rotate()</b>	224
<b>shift()</b>	233	<b>string()</b>	242		

# 按字母顺序排列的运算列表

以非字母方式命名的运算(例如 +、!、和 >) 列在本附录的最后, 起始页码为259。除非另行指明, 否则本节中所有例子均按缺省复位模式执行, 且所有变量均假定为未定义变量。此外, 由于格式的限制, 近似值通常保留3位小数(3.14159265359 显示为 3.141.....)。

## abs() MATH/Number 菜单

abs(表达式) ⇒ 表达式

abs(数组) ⇒ 数组

abs(矩阵) ⇒ 矩阵

返回自变量的绝对值。

若自变量为复数, 则返回该复数的模。

**注:** 所有未定义变量均视为实变量处理。

abs( $\{\pi/2, -\pi/3\}$ ) [ENTER]

$\{\frac{\pi}{2} \frac{\pi}{3}\}$

abs( $2-3j$ ) [ENTER]

$\sqrt{13}$

abs(z) [ENTER]

|z|

abs(x+yj) [ENTER]

$\sqrt{x^2+y^2}$

## and MATH/Test 和 MATH/Base 菜单

布尔表达式1 and 布尔表达式2 ⇒ 布尔表达式

布尔数组1 and 布尔数组2 ⇒ 布尔数组

布尔矩阵1 and 布尔矩阵2 ⇒ 布尔矩阵

返回值“真”或“假”, 或是原输入的化简形式。

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

用and 运算逐位比较2个实整数。在内部运算中, 2个整数均会被转化为带符号的32位二进制数。当相应位进行比较时, 若2个位值都为1, 则结果为1; 否则, 结果为0。返回值代表了逐位的结果, 并以所选基数的模式显示出来。

您可以按任何数基输入整数。对于按2进制或16进制输入的整数, 您必须分别使用0b或0h作为前缀。若不带前缀, 任何整数均会被视为10进制数(以10为基数)。

若您输入的十进制整数超过了带符号的32位二进制数字的表示范围, 对称求模运算将使该值落入正确的范围内。

对于16进制:

0h7AC36 和 0h3D5F [ENTER]

0h2C16

**重要信息:** 是零, 而不是字母O。

对于2进制:

0b100101 和 0b100 [ENTER]

0b100

对于十进制:

37 和 0b100 [ENTER]

4

**注:** 输入的2进制数可以长达32位数字(不包括0b前缀)。输入的16进制数最多可以有8位数字。



## AndPic CATALOG

**AndPic** *picVar*, 行, 列

显示出图形屏幕, 并使用逻辑运算“ANDS”对存储在*picVar*里的图形和当前屏幕上的图形在像素坐标(行,列)处进行处理。

*picVar* 必须为图形类型。

缺省情况下的坐标取(0,0), 即屏幕的左上角。

在函数作图模式和Y= Editor中:

$y1(x) = \cos(x)$   $\ominus$

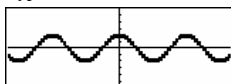
[2nd] [F6] Style = 3:Square

[F6] Style = 3:Square

[F2] Zoom = 7:ZoomTrig

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

Type = Picture, Variable = PIC1



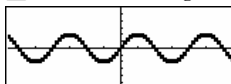
$y2(x) = \sin(x)$

[2nd] [F6] Style = 3:Square

[F6] Style = 3:Square

$y1 = \text{no checkmark}$  (F4 to deselect)

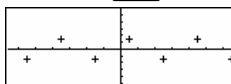
[F2] Zoom = 7:ZoomTrig



[HOME]

[CALC HOME]

AndPic PIC1 [ENTER]



Done

## angle() MATH/Complex 菜单

**angle**(表达式*n*)  $\Rightarrow$  表达式

返回表达式*n*的角度值, 将表达式*n*视为复数处理。

**注:** 所有未定义变量均应作为实变量处理。

在Degree (角度) 模式下:

$\text{angle}(0+2i)$  [ENTER]

90

在Radian (弧度) 模式下:

$\text{angle}(1+i)$  [ENTER]

$\frac{\pi}{4}$

$\text{angle}(z)$  [ENTER]

$\text{angle}(x+iy)$  [ENTER]

$$\begin{aligned} \blacksquare \text{angle}(z) &= \frac{-\pi \cdot (\text{sign}(z) - 1)}{2} \\ \blacksquare \text{angle}(x + i \cdot y) &= \frac{\pi \cdot \text{sign}(y)}{2} - \tan^{-1}\left(\frac{x}{y}\right) \end{aligned}$$

**angle**(数组*n*)  $\Rightarrow$  数组

**angle**(矩阵*n*)  $\Rightarrow$  矩阵

返回数组*n* 或 矩阵*n*中元素的角度所构成的数组或矩阵, 将每个元素均视为代表二维直角坐标点的复数来处理。

在Radian (弧度) 模式下:

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

$$\blacksquare \text{angle}(\{1+2i \quad 3+0i \quad 0-4i\}) = \left\{ \frac{\pi}{2} - \tan^{-1}(1/2) \quad 0 \quad -\frac{\pi}{2} \right\}$$

## ans() 2nd ANS 键

ans()  $\Rightarrow$  值  
ans(整数)  $\Rightarrow$  值

返回主屏幕历史区域中的一个答案项。

如果带整数参数, 则指明了希望调用前面特定的答案项。该整数参数的有效范围为1到99, 且不能为表达式形式。其缺省值为1, 此时返回最近一个答案。

欲使用 ans() 在主屏幕上生成一个菲波那其 (Fibonacci) 数列, 可按下:

1 ENTER 1  
1 ENTER 1  
2nd ANS + 2nd ANS  $\downarrow$   $\leftarrow$  2 ENTER 2  
ENTER 3  
ENTER 5

## approx() MATH/Algebra 菜单

approx(表达式)  $\Rightarrow$  值

在可能的情形下, 无论当前是Exact模式还是Approx模式, 都以十进制形式返回表达式的估计值。

此运算等同于在主屏幕上输入表达式并按下  $\downarrow$  ENTER。

approx( $\pi$ ) ENTER 3.141...

approx(数组*n*)  $\Rightarrow$  数组  
approx(矩阵*n*)  $\Rightarrow$  矩阵

在可能的情形下, 返回一个其元素均以十进制数表示的数组或矩阵。

approx({sin( $\pi$ ),cos( $\pi$ )}) ENTER {0. -1.}

approx(| $\sqrt{2}$ ), $\sqrt{3}$ |) ENTER [1.414... 1.732...]

## Archive CATALOG

Archive 变量1[, 变量2[, 变量3]...

将指定的变量从内存中移动到用户数据文档存储区里。

您可以象存取内存变量一样来存取存档变量。然而, 由于存档变量被自动加锁, 您不能对其进行删除、重命名或存储。

欲对变量解除存档, 可以使用Unarchiv。

10  $\rightarrow$  arctest ENTER 10  
Archive arctest ENTER Done  
5 \* arctest ENTER 50  
15  $\rightarrow$  arctest ENTER



ESC  
Unarchiv arctest ENTER Done  
15  $\rightarrow$  arctest ENTER 15

## arcLen() MATH/Calculus 菜单

arcLen(表达式1, var, 起点, 终点)  $\Rightarrow$  表达式

返回用变量var表示的表达式1从起点到终点的弧长。

无论何种作图模式, 弧长将用按函数模式定义的积分运算来计算。

arcLen(cos(x), x, 0,  $\pi$ ) 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, var, 起点, 终点)  $\Rightarrow$  数组

返回一个数组, 其元素为数组1中各元素关于变量var从起点到终点的弧长。

arcLen({sin(x),cos(x)}, x, 0,  $\pi$ )  
{3.820... 3.820...}

## augment() MATH/Matrix 菜单

augment(数组1, 数组2)  $\Rightarrow$  数组

返回一个新数组, 其组成为将数组2添加到数组1的末尾处。

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

<b>augment</b> ( <i>矩阵1</i> , <i>矩阵2</i> ) ⇒ <i>矩阵</i>	<b>[1,2;3,4]►M1</b> [ENTER]	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
<b>augment</b> ( <i>矩阵1</i> ; <i>矩阵2</i> ) ⇒ <i>矩阵</i>	<b>[5;6]►M2</b> [ENTER]	$\begin{bmatrix} 5 \\ 6 \end{bmatrix}$
返回一个新矩阵，其组成为将 <i>矩阵2</i> 添加到 <i>矩阵1</i> 。当使用符号“;”时，上述矩阵必须有相同的行数，则 <i>矩阵2</i> 作为新的列添加到 <i>矩阵1</i> 。当使用符号“,”时，上述矩阵必须有相同的列数，则 <i>矩阵2</i> 作为新的行添加到 <i>矩阵1</i> 。此运算不会改变 <i>矩阵1</i> 或 <i>矩阵2</i> 。	<b>augment(M1,M2)</b> [ENTER]	$\begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix}$
	<b>[5,6]►M2</b> [ENTER]	$\begin{bmatrix} 5 & 6 \\ 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$
	<b>augment(M1;M2)</b> [ENTER]	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$

## avgRC() CATALOG

<b>avgRC</b> ( <i>表达式1</i> , <i>var</i> [, <i>h</i> ]) ⇒ <i>表达式</i>	<b>avgRC(f(x),x,h)</b> [ENTER]	$\frac{f(x+h) - f(x)}{h}$
返回前项差商(平均变化率)。		
<i>表达式1</i> 可以为用户自定义的函数名(参见 <b>Func</b> )。	<b>avgRC(sin(x),x,h) x=2</b> [ENTER]	$\frac{\sin(h+2) - \sin(2)}{h}$
<i>h</i> 为步长值。若 <i>h</i> 被省略，则其缺省值为0.001。	<b>avgRC(x^2-x+2,x)</b> [ENTER]	2. · (x - .4995)
注意函数 <b>nDeriv()</b> 功能与之相似，只是使用中央差商。	<b>avgRC(x^2-x+2,x,1)</b> [ENTER]	2. · (x - .45)
	<b>avgRC(x^2-x+2,x,3)</b> [ENTER]	2. · (x+1)

## ►Bin MATH/Base 菜单

<b>整数1</b> ►Bin ⇒ <b>整数</b>	<b>256</b> ►Bin [ENTER]	<b>0b10000000</b>
将 <b>整数1</b> 转换为一个二进制数。二进制或十六进制数通常分别带 <b>0b</b> 或 <b>0h</b> 前缀。	<b>0h1F</b> ►Bin [ENTER]	<b>0b11111</b>
┌ 零，而不是字母O，后面为 <b>b</b> 或 <b>h</b> 。		
0b <b>二进制数</b>		
0h <b>十六进制数</b>		
└ 二进制数可长达32位。十六进制数可长达8位。		

若不带前缀，整数1将视为十进制数处理(数基为10)。无论何种Base模式，结果均以二进制数显示。

若您输入的十进制整数超过了带符号的32位二进制数字的表示范围，对称求模运算将使该值落入正确的范围内。

## BldData CATALOG

### BldData [dataVar]

根据绘制当前图形所用的信息来创建数据变量 *dataVar*。BldData 在所有的作图模式中均有效。

若 *dataVar* 省略，则数据将存储在系统变量 *sysData* 中。

**注：**当您在使用 BldData 后第一次启动数据/矩阵编辑器(Data/Matrix Editor)时，*dataVar* 或 *sysData* (取决于您使用 BldData 时用到的自变量) 将被设置为当前的数据变量。

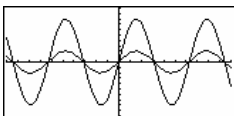
任何自变量(右面例子中为 *x*)的增量均根据 Window 变量值进行计算。

要了解与图形数值化所用增量相关的信息，请参考说明作图模式的章节。

在函数作图模式和弧角度模式下：

$8*\sin(x) \rightarrow y1(x)$  [ENTER]  
 $2*\sin(x) \rightarrow y2(x)$  [ENTER]  
 ZoomStd [ENTER]

Done  
Done



[HOME]  
 [CALC HOME]

BldData [ENTER]  
 APPS 6 [ENTER]

Done

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作图模式使用2个自变量。在右面给出的采样数据中，请注意当 *y* 在其取值范围内增加时，*x* 保持为常数。

然后，*x* 增加至下一取值，而 *y* 重新在其取值范围内增加。该过程持续进行，直到 *x* 遍历其取值范围内的所有值为止。

**注：**下列采样数据来自某3D图形。

DATA	x	y	z
	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*n*)  $\Rightarrow$  整数

返回大于等于自变量的最接近的整数。

该自变量可以为实数或复数。

**注：**可参见 floor()。

ceiling(0.456) [ENTER]

1.

ceiling(数组*n*)  $\Rightarrow$  数组

ceiling(矩阵*n*)  $\Rightarrow$  矩阵

返回一个数组或矩阵，其各元素为原数组或矩阵相应元素的 ceiling 函数值。

ceiling({ -3.1,1,2.5}) [ENTER]

{ -3.1 3.}

ceiling([0, -3.2; 1.3, 4]) [ENTER]

$\begin{bmatrix} 0 & -3. \\ 2. & 4 \end{bmatrix}$

## cFactor() MATH/Algebra/Complex 菜单

cFactor(表达式*n*, var)  $\Rightarrow$  表达式

cFactor(数组*n*, var)  $\Rightarrow$  数组

cFactor(矩阵*n*, var)  $\Rightarrow$  矩阵

cFactor(表达式*n*) 返回一个关于所有变量的因式分解并带有公分母的表达式*n*。

表达式*n* 应尽可能地分解为线性有理因式，甚至可以引入新的非实型常数。如果您想进行关于2个以上的变量的因式分解，则此方法适用。

cFactor(a^3\*x^2+a\*x^2+a^3+a) [ENTER]

$a \cdot (a + -j) \cdot (a + j) \cdot (x + -j) \cdot (x + j)$

cFactor(x^2+4/9) [ENTER]

$\frac{(3 \cdot x + -2 \cdot j) \cdot (3 \cdot x + 2 \cdot j)}{9}$

cFactor(x^2+3) [ENTER]

$x^2 + 3$

cFactor(x^2+2a) [ENTER]

$x^2 + a$

**cFactor**(*表达式1, var*) 返回按变量 *var* 进行因式分解的表达式 *1*。

表达式 *1* 应尽量分解为关于变量 *var* 的线形有理因式，可以包含非实数型常数，甚至可以引入无理常数或关于其它变量的无理子表达式。

因式及其相关项将按主变量 *var* 进行分类。各因式中 *var* 的同次幂将汇集在一起。如果只进行关于变量 *var* 的因式分解，并且您允许在因式分解中存在关于其它变量的无理表达式，以进一步开展关于 *var* 的因式分解，则变量 *var* 必须包含在函数命令中。结果中可能出现关于其它变量的伴随因式分解。

如果 Exact/Approx 模式设置为 AUTO，该函数命令中包含变量 *var* 就意味着在无理系数不能采用内建函数进行简要清楚地表达时，可以采用浮点系数来的近似计算。即使只有一个变量，包含变量 *var* 也可能产生更加完全的因式分解式。

**注：** 参见 **factor()**。

$$\mathbf{cFactor(a^3*x^2+a*x^2+a^3+a,x)} \quad \text{[ENTER]}$$

$$a \cdot (a^2 + 1) \cdot (x + \sqrt{3} \cdot \sqrt{a}) \cdot (x + \sqrt{3} \cdot \sqrt{a})$$

$$\mathbf{cFactor(x^2+3,x)} \quad \text{[ENTER]}$$

$$(x + \sqrt{3} \cdot \sqrt{3}) \cdot (x + \sqrt{3} \cdot \sqrt{3})$$

$$\mathbf{cFactor(x^2+a,x)} \quad \text{[ENTER]}$$

$$(x + \sqrt{a} \cdot \sqrt{a}) \cdot (x + \sqrt{a} \cdot \sqrt{a})$$

$$\mathbf{cFactor(x^5+4x^4+5x^3-6x-3)} \quad \text{[ENTER]}$$

$$x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3$$

$$\mathbf{cFactor(ans(1),x)} \quad \text{[ENTER]}$$

$$(x - .965) \cdot (x + .612) \cdot (x + 2.13) \cdot (x + 1.11 - 1.07 \cdot \sqrt{3}) \cdot (x + 1.11 + 1.07 \cdot \sqrt{3})$$

## char() MATH/String 菜单

**char**(*integer*) ⇒ 字符

返回一个字符串，其内容包括 Ti-89 Titanium/Voyage™ 200 字符集中的编号为整数 *integer* 的字符。可参见附录 B 中字符编码列表。*integer* 的有效范围是 0–255。

**char**(38) [ENTER] "&"

**char**(65) [ENTER] "A"

## checkTmr() CATALOG

**checkTmr**(*starttime*) ⇒ 整数

返回一个整数，表示在计时器启动后已历经的秒数。*starttime* 是一个由 **startTmr()** 函数返回的整数。

您可以使用一个数组或矩阵来表示 *starttime* 整数。有效的 *starttime* 整数值一定会落入 0 时刻和时钟的当前时刻之间。您可以同时运行多个计时器。

**注：** 可参见 **startTmr()** 和 **timeCnv()**。

**startTmr**() [ENTER] 148083315

**checkTmr**(148083315) 34

**startTmr**()>Timer1

⋮

**startTmr**()>Timer2

⋮

**checkTmr**(Timer1)>Timer1 的

⋮

**checkTmr**(Timer2)>Timer2 的

## Circle CATALOG

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

以窗口坐标 (*x, y*) 为圆心、半径为 *r* 作圆。

*x, y* 和 *r* 必须是实数。

若 *drawMode* = 1，作圆(缺省情况)。

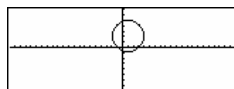
若 *drawMode* = 0，关闭已作的圆。

若 *drawMode* = -1，沿着圆进行像素反转。

**注：** 重新作图将清除所有已经画出的项目。可参见 **PxlCrcl**。

在 ZoomSqr 视窗中：

**ZoomSqr:Circle 1,2,3** [ENTER]



## ClockOff CATALOG

**ClockOff**

关闭时钟。

## ClockOn CATALOG

### ClockOn

开启时钟。

## ClrDraw CATALOG

### ClrDraw

清空作图屏幕，并将Smart Graph 属性复位，以便当再次显示Graph屏幕时，图形将被重画。

在查看图形屏幕时，您可以按下 **F4** (ReGraph)清空已经画出的所有项目(例如线和点)或按下：

 **2nd** **F6**

 **F6**

以选定 1:ClrDraw。

## ClrErr CATALOG

### ClrErr

清空错误状态。将错误代码设置为0，并清除内部错误关联变量。

程序中**Try...EndTry** 中的 **Else** 子句应使用 **ClrErr** 或 **PassErr**。若需要处理或省略某个错误，则应该使用 **ClrErr**。若不清楚如何处理该错误，应使用 **PassErr** 将其送至下一个错误处理句柄。若没有其它未完成的 **Try...EndTry** 错误处理句柄，则错误对话框将正常显示。

**注：**可参见 **PassErr** 或 **Try**。

程序清单：

```

: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

清除用**Graph**命令绘图或使用**Table**命令创建的所有函数或表达式。(参见 **Graph** 或 **Table**。)

任何此前选中的Y= functions 将会在下一次图形显示时绘制。

## ClrHome CATALOG

### ClrHome

清空存储在主屏幕历史区域的**entry()** 和 **ans()**中的所有项目。但不清空当前的输入行。

在查看主屏幕时，您可以按下 **F1** 并选定 8:Clear Home清空主屏幕。

对于那些返回任意常数或整数(@1、@2等)的函数，如 **solve()**，**ClrHome** 将后缀复位为1。

## ClrO CATALOG

### ClrO

清空程序输入/输出屏幕。

## ClrTable CATALOG

### ClrTable

清除所有表格数值。仅适用于Table Setup对话框的ASK设置。

在Ask模式下查看Table屏幕时，您可以按下[F1]并选定8:Clear Table清除数值。

## colDim() MATH/Matrix/Dimensions 菜单

colDim(矩阵) ⇒ 表达式

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

3

返回矩阵所包含的列数。

注：可参见rowDim()。

## colNorm() MATH/Matrix /Norms 菜单

colNorm(矩阵) ⇒ 表达式

[1, -2,3;4,5, -6] > mat [ENTER]

返回矩阵各列元素的绝对值之和的最大值。

注：不可用于未定义的矩阵元素。可参见rowNorm()。

colNorm(mat) [ENTER]

$\begin{bmatrix} 1 & -2 & 3 \\ 4 & 5 & -6 \end{bmatrix}$   
9

## comDenom() MATH/Algebra 菜单

comDenom(表达式 $l, var$ ) ⇒ 表达式

comDenom(数组 $l, var$ ) ⇒ 数组

comDenom(矩阵 $l, var$ ) ⇒ 矩阵

comDenom(表达式 $l$ ) 返回一个分子和分母完全展开的化简分子式。

comDenom(表达式 $l, var$ ) 返回一个分子和分母关于变量 $var$ 展开的化简分子式。各项和因式按主变量 $var$ 进行分类。各因式中 $var$ 的同次幂汇集在一起。结果中可能包含系数的因式分解。与省略 $var$ 相比，在使表达式的表述更全面的的同时，上述操作通常可以能够节省时间、内存、和屏幕空间。该函数同时有助于更快地通过后续步骤得出结果，对内存的占用也更少。

若表达式中不含变量 $var$ ，comDenom(表达式 $l, var$ ) 返回分子和分母均未展开的化简分子式。上述结果通常可以节省更多时间、内存、和屏幕空间。这些部分因式分解的结果同时有助于更快地通过后续步骤得出结果，对内存的占用也更少。

comDenom(( $y^2+2y$ )/( $(x+1)^2+y^2+2y$ )) [ENTER]

$$\text{comDenom}\left(\frac{y^2+y}{(x+1)^2} + y^2 + y, y\right) \\ \frac{x^2 \cdot y^2 + x^2 \cdot y + 2 \cdot x \cdot y^2 + 2 \cdot y^3}{x^2 + 2 \cdot x + 1}$$

comDenom(( $y^2+2y$ )/( $(x+1)^2+y^2+y, x$ )) [ENTER]

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

comDenom(( $y^2+2y$ )/( $(x+1)^2+y^2+y, y$ )) [ENTER]

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

comDenom(exprn,abc)⇒comden(exprn) [ENTER]

Done

comden(( $y^2+2y$ )/( $(x+1)^2+y^2+2y$ )) [ENTER]

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

即使没有分母，若 **factor()** 执行过慢或是占用过多内存时，**comden** 函数通常有助于完成部分因式分解。

**comden(1234x^2\*(y^3-y)+2468x\*(y^2-1))** **[ENTER]**  
 $1234 \cdot x \cdot (x \cdot y + 2) \cdot (y^2 - 1)$

**提示：** 输入 **comden()** 函数的定义，并通常可将其当作 **comDenom()** 和 **factor()** 的替代函数试用。

## conj() MATH/Complex 菜单

**conj(表达式)**  $\Rightarrow$  表达式  
**conj(数组)**  $\Rightarrow$  数组  
**conj(矩阵)**  $\Rightarrow$  矩阵

返回自变量的共轭复数。

**注：** 所有未定义变量均作为实变量处理。

**conj(1+2i)** **[ENTER]**  $1 - 2 \cdot i$   
**conj([2,1-3i-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** *var1, var2*

将变量 *var1* 的内容复制到 *var2*。若 *var2* 不存在，**CopyVar** 将创建此变量。

**注：** 在您复制表达式、数组、矩阵或字符串时，**CopyVar** 类似于存储指令 (**\(\rightarrow\)**)，唯一的不同之处是在使用 **CopyVar** 时不会对被处理对象进行化简。您必须在使用 **CopyVar** 时采用非代数变量类型，如 Pic 和 GDB 类型的变量。

**x+y \(\rightarrow\) a** **[ENTER]**  $x + y$   
**10 \(\rightarrow\) 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() [2nd][COS] 键

**cos(表达式)**  $\Rightarrow$  表达式  
**cos(数组)**  $\Rightarrow$  数组

**cos(表达式)** 以表达式形式返回自变量的余弦值。

**cos(数组)** 返回一个数组，其元素为数组 *l* 中所对应元素的余弦值。

**注：** 根据当前的角度模式设置，自变量可以是角度或弧度形式。您可以使用  $^\circ$  或  $r$  来临时改变角度模式。

在 Degree (角度) 模式下:

**cos( $\pi/4$ )<sup>r</sup>** **[ENTER]**  $\frac{\sqrt{2}}{2}$   
**cos(45)** **[ENTER]**  $\frac{\sqrt{2}}{2}$   
**cos({0,60,90})** **[ENTER]**  $\{1 \ 1/2 \ 0\}$

在 Radian (弧度) 模式下:

**cos( $\pi/4$ )** **[ENTER]**  $\frac{\sqrt{2}}{2}$   
**cos(45 $^\circ$ )** **[ENTER]**  $\frac{\sqrt{2}}{2}$



$\cos(\text{squareMatrix1}) \Rightarrow \text{squareMatrix1}$  (方阵)

返回方阵  $T$  的矩阵余弦。该运算不同于求方阵  $T$  各元素的余弦值。

当使用标量函数  $f(A)$  对方阵  $T$  (A) 进行运算时, 结果按代数方式进行计算:

1. 计算特征值( $\lambda_i$ ) 和A的特征向量 ( $V_i$ )。

方阵  $T$  必须可对角化。同时, 不得包含未赋值的符号变量。

2. 构建矩阵:

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

3. 然后令  $A = X B X^{-1}$  且  $f(A) = X f(B) X^{-1}$ 。例如, 满足以下条件时,  $\cos(A) = X \cos(B) X^{-1}$ :

$\cos(B) =$

所有计算均以浮点运算进行。

在Radian (弧度) 模式下:

$\cos([1,5,3;4,2,1;6,^{-}2,1])$  [ENTER]

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

**$\cos^{-1}()$**    [COS<sup>-1</sup>] 键   [2nd] [COS<sup>-1</sup>] 键

$\cos^{-1}(\text{表达式}) \Rightarrow$  表达式

$\cos^{-1}(\text{数组}) \Rightarrow$  数组

$\cos^{-1}(\text{表达式})$  以表达式形式返回一个角度值, 其余弦值为表达式  $T$ 。

$\cos^{-1}(\text{数组})$  返回一个数组, 其元素为数组  $T$  中所对应元素的反余弦值。

**注:** 根据当前的角度模式设置, 返回结果可以是角度或弧度形式。

在Degree (角度) 模式下:

$\cos^{-1}(1)$  [ENTER]

0

在Radian (弧度) 模式下:

$\cos^{-1}(\{0,^{-}2,5\})$  [ENTER]

$\left\{ \frac{\pi}{2}, 1.369\dots, 1.047\dots \right\}$

$\cos^{-1}(\text{方阵}) \Rightarrow$  方阵

返回方阵  $T$  的矩阵反余弦。该运算不同于求方阵  $T$  各元素的反余弦值。欲了解与计算方法相关的信息, 可参见  $\cos()$ 。

方阵  $T$  必须可对角线化。结果中总包含有浮点数值。

在Radian (弧度) 模式和Rectangular (直角) 复数格式模式下:

$\cos^{-1}([1,5,3;4,2,1;6,^{-}2,1])$

[ENTER]

$\begin{bmatrix} 1.734\dots+0.064\dots\cdot i & -1.490\dots+2.105\dots\cdot i\dots \\ ^{-}.725\dots+1.515\dots\cdot i & .623\dots+0.778\dots\cdot i\dots \\ ^{-}2.083\dots+2.632\dots\cdot i & 1.790\dots-1.271\dots\cdot i\dots \end{bmatrix}$

**$\cosh()$**  **MATH/Hyperbolic 菜单**

$\cosh(\text{表达式}) \Rightarrow$  表达式

$\cosh(\text{数组}) \Rightarrow$  数组

$\cosh(\text{表达式})$  返回以表达式形式表示的自变量的双曲余弦值。

$\cosh(\text{数组})$  返回一个数组, 其元素为数组  $T$  所对应元素的双曲余弦值。

$\cosh(1.2)$  [ENTER]

1.810...

$\cosh(\{0,1.2\})$  [ENTER]

{1 1.810...}

**cosh(方阵*n*)** ⇒ 方阵

返回方阵*n*的矩阵双曲余弦。该运算不同于求方阵*n*各元素的双曲余弦。要了解与计算方法相关的信息，可参见**cos()**。

方阵*n*必须可对角线化。结果中总包含有浮点数。

在Radian (弧度) 模式下:

**cosh({1,5,3;4,2,1;6,-2,1})**  
[ENTER]

```
421.255 253.909 216.905
327.635 255.301 202.958
226.297 216.623 167.628
```

## **cosh<sup>-1</sup>()** MATH/Hyperbolic 菜单

**cosh<sup>-1</sup>(表达式*n*)** ⇒ 表达式

**cosh<sup>-1</sup>(数组*n*)** ⇒ 数组

**cosh<sup>-1</sup>(表达式*n*)** 以表达式形式返回自变量的反双曲余弦值。

**cosh<sup>-1</sup>(数组*n*)** 返回一个数组，其元素为数组*n*所对应元素的反双曲余弦值。

**cosh<sup>-1</sup>(1)** [ENTER]

0

**cosh<sup>-1</sup>({1,2,1,3})** [ENTER]

{0 1.372... cosh<sup>-1</sup>(3)}

**cosh<sup>-1</sup>(方阵*n*)** ⇒ 方阵

返回方阵*n*的矩阵反双曲余弦。该运算不同于对各元素求反双曲余弦的计算。欲了解与计算方法相关的信息，可参见**cos()**。

方阵*n*必须可对角线化。结果中总包含有浮点数。

在Radian (弧度) 模式和Rectangular (直角) 复数格式模式下:

**cosh<sup>-1</sup>({1,5,3;4,2,1;6,-2,1})** [ENTER]

```
2.525...+1.734...·i -.009...-1.490...·i ...
.486...-.725...·i 1.662...+.623...·i ...
-.322...-2.083...·i 1.267...+1.790...·i ...
```

## **cot()** MATH/Trig 菜单

**cot(表达式*n*)** ⇒ 表达式

**cot(数组*n*)** ⇒ 数组

返回表达式*n*的余切值或返回一个数组，其元素为数组*n*中所对应元素的余切值。

注: 根据当前的角度模式设置，自变量可以是角度或弧度形式。

在Degree (角度) 模式下:

**cot(45)** [ENTER]

1

在Radian (弧度) 模式下:

**cot({1,2,1,3})** [ENTER]

$$\frac{1}{\tan(1)} \quad -.584... \quad \frac{1}{\tan(3)}$$

## **cot<sup>-1</sup>()** MATH/Trig 菜单

**cot<sup>-1</sup>(表达式*n*)** ⇒ 表达式

**cot<sup>-1</sup>(数组*n*)** ⇒ 数组

返回余切值为表达式*n*的角度，或返回一个数组，其元素为数组*n*中所对应元素的反余切值。

注: 根据当前的角度模式设置，返回结果可以是角度或弧度形式。

在Degree (角度) 模式下:

**cot<sup>-1</sup>(1)** [ENTER]

45

在Radian (弧度) 模式下:

**cot<sup>-1</sup>(1)** [ENTER]

$\frac{\pi}{4}$

## **coth()** MATH/Hyperbolic 菜单

**coth(表达式*n*)** ⇒ 表达式

**coth(数组*n*)** ⇒ 数组

返回表达式*n*的双曲余切值，或返回一个数组，其元素为数组*n*中所对应元素的双曲余切值。

**coth(1.2)** [ENTER]

1.199...

**coth({1,3,2})** [ENTER]

$$\frac{1}{\tanh(1)} \quad 1.003...$$

## **coth<sup>-1</sup>()** MATH/Hyperbolic 菜单

**coth<sup>-1</sup>(表达式*n*)** ⇒ 表达式

**coth<sup>-1</sup>(数组*n*)** ⇒ 数组

返回表达式*n*的反双曲余切值，或返回一个数组，其元素为数组*n*中所对应元素的反双曲余切值。

**coth<sup>-1</sup>(3.5)** [ENTER]

.293...

**coth<sup>-1</sup>({-2,2,1,6})** [ENTER]

$$\frac{-\ln(3)}{2} \quad .518... \quad \frac{\ln(7/5)}{2}$$

**CROSSP()** MATH/Matrix/Vector ops 菜单**CROSSP**(数组1, 数组2) ⇒ 数组

以数组形式返回数组1和数组2的叉积。

数组1和数组2必须具有相同的维数, 且该维数必须为2或3。

**CROSSP**({a1,b1},{a2,b2}) **[ENTER]**  
 $\{0 \ 0 \ a1 \cdot b2 - a2 \cdot b1\}$ **CROSSP**({0.1,2.2,-5},{1,-5.0}) **[ENTER]**  
 $\{-2.5 \ -5. \ -2.25\}$ **CROSSP**(向量1, 向量2) ⇒ 向量

返回一个行向量或列向量(根据自变量的不同), 其值为向量1和向量2的叉积。

向量1和向量2必须同时为行向量或同时为列向量。这2个向量必须具有相同的维数, 且该维数必须为2或3。

**CROSSP**({1,2,3},{4,5,6}) **[ENTER]**  
 $[-3 \ 6 \ -3]$ **CROSSP**({1,2},{3,4}) **[ENTER]**  
 $[0 \ 0 \ -2]$ **CSC()** MATH/Trig 菜单**CSC**(表达式1) ⇒ 表达式**CSC**(数组1) ⇒ 数组

返回表达式1的余割值, 或返回一个数组, 其元素为数组1中所对应元素的余割值。

在Degree (角度) 模式下:

**CSC**( $\pi/4$ ) **[ENTER]**  
 $\frac{1}{\sin(\frac{\pi}{4})}$ 

在Radian (弧度) 模式下:

**CSC**({1, $\pi/2$ , $\pi/3$ }) **[ENTER]**  
 $\frac{1}{\sin(1)} \ 1 \ \frac{2\sqrt{3}}{3}$ **CSC<sup>-1</sup>()** MATH/Trig 菜单**CSC<sup>-1</sup>**(表达式1) ⇒ 表达式**CSC<sup>-1</sup>**(数组1) ⇒ 数组

返回余割值为表达式1的角度值, 或返回一个数组, 其元素为数组1中所对应元素的反余割值。

**注:** 根据当前的角度模式设置, 返回结果可以是角度或是弧度形式。

在Degree (角度) 模式下:

**CSC<sup>-1</sup>**(1) **[ENTER]** 90

在Radian (弧度) 模式下:

**CSC<sup>-1</sup>**({1,4,6}) **[ENTER]**  
 $\frac{\pi}{2} \ \sin^{-1}(1/4) \ \sin^{-1}(1/6)$ **CSCH()** MATH/Hyperbolic 菜单**CSCH**(表达式1) ⇒ 表达式**CSCH**(数组1) ⇒ 数组

返回表达式1的双曲余割值, 或返回一个数组, 其元素为数组1中所对应元素的双曲余割值。

**CSCH**(3) **[ENTER]**  
 $\frac{1}{\sinh(3)}$ **CSCH**({1,2,1,4}) **[ENTER]**  
 $\frac{1}{\sinh(1)} \ .248\dots \ \frac{1}{\sinh(4)}$ **CSCH<sup>-1</sup>()** MATH/Hyperbolic 菜单**CSCH<sup>-1</sup>**(表达式1) ⇒ 表达式**CSCH<sup>-1</sup>**(数组1) ⇒ 数组

返回表达式1的反双曲余割值, 或返回一个数组, 其元素为数组1中所对应元素的反双曲余割值。

**CSCH<sup>-1</sup>**(1) **[ENTER]**  $\sinh^{-1}(1)$ **CSCH<sup>-1</sup>**({1,2,1,3}) **[ENTER]**  
 $\sinh^{-1}(1) \ .459\dots \ \sinh^{-1}(1/3)$

## cSolve() MATH/Algebra/Complex 菜单

**cSolve**(方程, var) ⇒ 布尔表达式

返回方程中变量 *var* 的候选复数解。该函数的功能是产生所有实数或非实数候选解。即使该方程为实数类型, **cSolve()** 也允许在实数模式下产生非实数解。





尽管 TI-89 Titanium/Voyage™ 200 按实数方式处理所有不以下划线(\_)形式结尾的未定义变量, **cSolve()** 可以解出多项式方程的复数解。

在求解时, 即使当前为实数域, **cSolve()** 也会临时性地将其改设为复数域。在复数域中, 奇分母分数幂将使用主支而不是实数支。因此, 对于涉及上述分数幂的方程, 由 **solve()** 得到的解不一定是由 **cSolve()** 得到的解的子集。

**cSolve()** 在开始使用时均采用精确符号法则。除 EXACT 模式外, 在需要时, **cSolve()** 也可采用迭代法进行近似复数多项式因式分解。


**注:** 可参见 **cZeros()**、**solve()**、和 **zeros()**。


**注:** 若方程式为含有如 **abs()**、**angle()**、**conj()**、**real()** 或 **imag()** 等函数的非多项式, 您需要在 *var* 的末尾加上下划线\_

  [-]   [2nd] [-]。缺省情况下, 变量视为实数值处理。


若您使用 *var\_*, 则变量视为复数处理。

您也需要对方程中可能是非实数值的其它变量使用 *var\_*。否则, 您可能会得到意想不到的结果。


**cSolve**( $x^3 = -1, x$ ) 

**solve**( $x^3 = -1, x$ ) 

■ **cSolve**( $x^3 = -1, x$ )  
 $\leftarrow 1/2 + \frac{\sqrt{3}}{2} \cdot i$  or  $x = 1/2 - \frac{\sqrt{3}}{2} \cdot i$   
 ■ **solve**( $x^3 = -1, x$ )      $x = -1$


**cSolve**( $x^{1/3} = -1, x$ ) 


false

**solve**( $x^{1/3} = -1, x$ ) 

$x = -1$


在Fix 2中显示Digits模式:

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

**cSolve**(**ans**(1), x) 


■ **exact**(**cSolve**( $x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3 = 0, x$ ))  
 $x \cdot (x^4 + 4 \cdot x^3 + 5 \cdot x^2 - 6) = 3$   
 ■ **cSolve**( $x \cdot (x^4 + 4 \cdot x^3 + 5 \cdot x^2 - 6) = 3$ )  
 $x = -1.1138 + 1.07314 \cdot i$  or  $x = -1.1138 - 1.07314 \cdot i$

*z* 视为实数处理:

**cSolve**(**conj**(*z*) = 1 + *iz*) 

$z = 1 + i$

*z\_* 视为复数处理:

**cSolve**(**conj**(*z\_*) = 1 + *iz\_*) 

$z = 1 - i$

**cSolve**(方程1 and 方程2 [and ... ],  
 {varOrGuess1, varOrGuess2[, ... ]})

⇒ 布尔表达式

返回联立代数方程组的候选复数解, 其中各 *varOrGuess* 指定了您希望求解的变量。

作为可选项, 您可以为变量指定一个初始估计值。每个 *varOrGuess* 的形式必须为:

变量

- 或 -



变量 = 实数 或 虚数

例如, *x* 有效, 则  $x = 3 + i$ 。

若所有方程均为多项式, 且您未指定任何初始估计值, 则 **cSolve()** 使用 Gröbner/Buchberger 词法消元法来求得全部复数解。

**注:** 下例使用了下划线\_

  [-]

  [2nd] [-], 因此其中的变量视为复数处理。

复数解可同时包括实数和虚数解，如右例所述。

$$\begin{aligned} & \mathbf{cSolve(u_*v_-u=v\_and} \\ & \mathbf{v\_^2=-u\_}\{u\_v\_}\} \text{ [ENTER]} \\ & u\_ = 1/2 + \frac{\sqrt{3}}{2} \cdot j \text{ and } v\_ = 1/2 - \frac{\sqrt{3}}{2} \cdot j \\ & \text{or } u\_ = 1/2 - \frac{\sqrt{3}}{2} \cdot j \text{ and } v\_ = 1/2 + \frac{\sqrt{3}}{2} \cdot j \\ & \text{or } u\_ = 0 \text{ and } v\_ = 0 \end{aligned}$$

联立多项式方程可以包含无具体数值的其他变量，但在解中用给定数值进行替换。

$$\begin{aligned} & \mathbf{cSolve(u_*v_-u=c_*v\_and} \\ & \mathbf{v\_^2=-u\_}\{u\_v\_}\} \text{ [ENTER]} \\ & u\_ = \frac{\sqrt{(1-4 \cdot c + 1)^2}}{4} \text{ and } v\_ = \frac{\sqrt{1-4 \cdot c + 1}}{2} \\ & \text{or} \\ & u\_ = \frac{\sqrt{(1-4 \cdot c - 1)^2}}{4} \text{ and } v\_ = \frac{\sqrt{1-4 \cdot c - 1}}{2} \\ & \text{or } u\_ = 0 \text{ and } v\_ = 0 \end{aligned}$$

解中也可以包含未出现在方程式中的求解变量。这些解说明解系可能包含形式为@k的任意常数，此处k是1到255之间的整数后缀。当您使用**ClrHome**或[F1] 8:Clear Home时下标复位为1。

$$\begin{aligned} & \mathbf{cSolve(u_*v_-u=v\_and} \\ & \mathbf{v\_^2=-u\_}\{u\_v\_w\_}\} \text{ [ENTER]} \\ & u\_ = 1/2 + \frac{\sqrt{3}}{2} \cdot j \text{ and } v\_ = 1/2 - \frac{\sqrt{3}}{2} \cdot j \\ & \text{and } w\_ = @1 \\ & \text{or} \\ & u\_ = 1/2 - \frac{\sqrt{3}}{2} \cdot j \text{ and } v\_ = 1/2 + \frac{\sqrt{3}}{2} \cdot j \\ & \text{and } w\_ = @1 \\ & \text{or } u\_ = 0 \text{ and } v\_ = 0 \text{ and } w\_ = @1 \end{aligned}$$

对于多项式方程组，计算时间或内存占用很大程度上取决于求解变量的排列次序。若您的初始选择会占用过多内存或时间，您可以尝试重新排列方程中和或varOrGuess列表中变量的次序。

$$\begin{aligned} & \mathbf{cSolve(u_+v_ = e^w\_and u_-v_ = i} \\ & \mathbf{\{u\_v\_}\} \text{ [ENTER]} \\ & u\_ = \frac{e^{w\_}}{2} + 1/2 \cdot j \text{ and } v\_ = \frac{e^{w\_} - j}{2} \end{aligned}$$

若不含任何估计值，且任何方程都不是任何变量的多项式，而所有方程都是求解变量的线性表达式，则**cSolve()**使用高斯消元法来求得全部的解。

$$\begin{aligned} & \mathbf{cSolve(e^z\_ = w\_and w_ = z\_^2, \{w\_z\_}\} \text{ [ENTER]} \\ & w\_ = .494... \text{ and } z\_ = ^-.703... \end{aligned}$$

若一个方程组既不是其任何变量的多项式，也不是求解变量的线性表达式，则通过近似叠代法**cSolve()**最多只能求得一个解。为此，求解变量的数量必须等于方程的数量，且方程中的所有其它变量必须化简为数字形式。

$$\begin{aligned} & \mathbf{cSolve(e^z\_ = w\_and w_ = z\_^2, \{w\_z\_ = 1 +} \\ & \mathbf{j}\} \text{ [ENTER]} \\ & w\_ = .149... + 4.891... \cdot j \text{ and } \\ & z\_ = 1.588... + 1.540... \cdot j \end{aligned}$$

非实数估计值对于确定非实数解而言通常是必不可少的。为了满足收敛，估计值应该尽可能的靠近解值。

## 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, 并更新所有统计变量。

除数组5外, 所有数组必须具有相同维数。

数组1代表x数组。

数组2代表y数组。

数组3代表频率。

数组4代表分类代码。

数组5代表分类所包含的数组。

注: 数组1到数组4必须有变量名或取名为c1-c99 (数据/矩阵编辑器中最末位的数据变量所代表的列)。数组5不一定要有变量名, 也可以不取名为c1-c99。

在函数作图模式下。

{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]

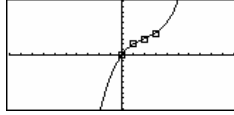
regeq(x)→y1(x) [ENTER]

Done

NewPlot 1,1,L1,L2 [ENTER]

Done

• [GRAPH]



## cumSum() MATH/List 菜单

**cumSum**(数组*n*) ⇒ 数组

返回一个数组, 其组成为数组*n*中从元素1开始的元素的累加和。

cumSum({1,2,3,4}) [ENTER]

{1 3 6 10}

**cumSum**(矩阵*n*) ⇒ 矩阵

返回一个矩阵, 其组成为矩阵*n*中元素的累加和。即其各元素为矩阵*n*中每列从顶到底的累加和。

[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**

清除定制工具条。

参见 **Custom** 程序清单的示例。

**CustmOn** 和 **CustmOff** 使用程序来控制定制工具条。通过手动方式, 您可以按下 [2nd][CUSTOM] 对某个定制工具条在开启和关闭状态间进行切换。同样, 当您切换应用软件时, 定制工具条会自动清除。

## CustmOn CATALOG

**CustmOn**

激活在 **Custom...EndCustm** 部分中已经设定的定制工具条。

参见 **Custom** 程序清单的示例。

**CustmOn** 和 **CustmOff** 使用程序来控制定制工具条。通过手动方式, 您可以按下 [2nd][CUSTOM] 对某个定制工具条在开启和关闭状态间进行切换。

## Custom 2nd [CUSTOM] 键

**Custom**  
*block*  
**EndCustm**

设定工具条, 使您在按下 2nd [CUSTOM] 时可以激活它。除 Title 和 Item 中的说明外, 该函数功能类似于 **ToolBar** 指令。

*block* 中可以是单条说明, 也可以是用字符“.”分隔的多条说明。

**注:** 2nd [CUSTOM] 功能是切换。第一次使用它将激活菜单, 而第二次使用则使菜单消失。当您改变应用软件时, 菜单也会消失。

程序清单:

```
: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**

立即将程序控制转入当前循环的下一轮循环 (**For**、**While**、或 **Loop**)。

**Cycle** 不得在3种循环结构外使用(**For**、**While**、或 **Loop**)。

程序清单:

```
:Ⓞ 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
```

程序执行后的 temp 内容: 5000

## CyclePic CATALOG

**CyclePic** *picNameString, n*, [*wait*], [*cycles*], [*direction*]

以指定间隔显示所有指定 PIC 变量。用户可以选择性地控制图片间隔时间、遍历图片的循环次数、以及遍历的方向、采用循环、前向还是后向。

*direction* 在循环时取 1 或在采用前向和后向时取 -1。缺省值 = 1。

- 将 3 张图片存为 pic1、pic2 和 pic3。
- 输入: CyclePic "pic",3,5,4,-1
- 自动显示三 (3) 张图片—图片间隔为半秒钟 (.5), 共四 (4) 次循环, 采用前向和后向 (-1)。

## ►Cylind MATH/Matrix/Vector ops 菜单

向量►Cylind

以圆柱坐标  $[r, \theta, z]$  格式显示行或列向量。

向量必须恰好包含 3 个元素。它可以是行向量或列向量。

[2,2,3] ►Cylind ENTER

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

## cZeros() MATH/Algebra/Complex 菜单

**cZeros**(*表达式, var*) ⇒ 数组

返回一个数组, 其元素为使得表达式=0 的 *var* 的实数和非实数候选值。**cZeros()** 通过计算 **expList(cSolve(表达式=0, var), var)** 完成其运算。否则, **cZeros()** 类似于 **zeros()**。

**注:** 参见 **cSolve()**、**solve()**、和 **zeros()**。

**注:** 若表达式不是带 **abs()**、**angle()**、**conj()**、**real()**、或 **imag()** 等函数的多项式, 您需要加上下划线 (  )

显示 Fix 3 中的 Digits 模式:

```
cZeros(x^5+4x^4+5x^3-6x-3,x) ENTER
{ -2.125 - .612 .965
-1.114 - 1.073 .j
-1.114 + 1.073 .j }
```

*z* 视为实数处理:

**cZeros(conj(z)-1-iz)** ENTER

( $\text{[ ]}$   $\text{[ ]}$   $\text{[ ]}$ ), ( $\text{[ ]}$   $\text{[ ]}$   $\text{[ ]}$ ) 位于 *var* 的末尾。缺省情况下, 变量视为实数处理。若您使用 *var\_*, 则变量视为复数处理。

您也需要对表达式中可能为虚数值的其它变量使用 *var\_*。否则, 您会得到意想不到的结果。

{1+*i*}

*z\_* 视为复数处理:

**cZeros(conj(*z\_*)-1-*i**z\_*)** **[ENTER]**

{1-*i*}

**cZeros**({表达式1, 表达式2[, ... ]},  
{*varOrGuess1*, *varOrGuess2* [, ... ]})  $\Rightarrow$  矩阵

返回所有表达式同时为零的候选值。各 *varOrGuess* 给定了一个您要寻找的数值。

作为可选项, 您可以为变量指定初始估计值。各 *varOrGuess* 的格式必须为:

*variable*  
- 或 -  
*variable* = 实数或虚数

例如, *x* 有效且 *x*=3+*i*。

若所有表达式均为多项式, 且您未指定任何初始估计值, **cZeros**() 使用 Gröbner/Buchberger 词法消元法来求得全部复零点。

复零点可以包括实数和非实数零点, 正如右侧的例子所示。

所生成矩阵的每一行都代表一个零点, 其元素的顺序与 *varOrGuess* 数组中元素的顺序相同。为了便于提取某行, 可以将矩阵按 [row] 添加索引。

**注:** 下例使用了下划线 ( $\text{[ ]}$   $\text{[ ]}$   $\text{[ ]}$ ) 从而所有变量均视为复数处理。

**cZeros**({*u\_\*v\_*-*u\_*-*v\_**v\_*<sup>2</sup>+*u\_*},  
{*u\_*, *v\_*}) **[ENTER]**

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

提取第2行:

**ans(1)|2** **[ENTER]**

$$\left[ 1/2 + \frac{\sqrt{3}}{2} \cdot i \quad 1/2 - \frac{\sqrt{3}}{2} \cdot i \right]$$

联立多项式可以包含无具体数值的其他变量, 但它们应该可以在解中用特定数值予以替换。

**cZeros**({*u\_\*v\_*-*u\_*-(*c\_\*v\_*),  
*v\_*<sup>2</sup>+*u\_*}, {*u\_*, *v\_*}) **[ENTER]**

$$\begin{bmatrix} \frac{-\sqrt{1-4 \cdot c} + 1}{4} & \frac{\sqrt{1-4 \cdot c} + 1}{2} \\ \frac{-\sqrt{1-4 \cdot c} - 1}{4} & \frac{-\sqrt{1-4 \cdot c} - 1}{2} \\ 0 & 0 \end{bmatrix}$$

零值中也可以包含未出现在表达式中的未知数。这些零值表明在零值系中可能包含形式为 *k* 的任意常数, 其中 *k* 是从 1 到 255 的整数后缀。当您使用 **ClrHome** 或 **[F1] 8:Clear Home** 时后缀复位为 1。

**cZeros**({*u\_\*v\_*-*u\_*-*v\_**v\_*<sup>2</sup>+*u\_*},  
{*u\_*, *v\_*, *w\_*}) **[ENTER]**

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

对于多项式系, 计算时间或内存占用很大程度上取决于未知数的排序。若您的初始选择占用过多内存或时间, 您可以尝试重新排列表达式中和 *i* 或 *varOrGuess* 列表中变量的次序。

若您不使用估计值, 且所有表达式都不是某个变量的多项式, 而只是未知数的一次式, 则 **cZeros**() 使用高斯消元法来求得所有零值。

**cZeros**({*u\_*+*v\_*-*e*<sup>(*w\_*)</sup>, *u\_*-*v\_*-*i*, {*u\_*, *v\_*})  
**[ENTER]**

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



若多项式系既不是其所有变量的多项式，也不是未知数的线性表达式，则使用近似迭代法 **cZeros()** 最多只能确定一个零值。为此，未知数的数量必须等于表达式的数量，且表达式中所有其它变量都必须化为数字形式。

非实数估计值通常需要确定非实数零值。为了满足收敛，估计值必须尽可能地靠近零点。

$$\text{cZeros}(\{e^{\wedge}(z_{-} - w_{-}w_{-} - z_{-}^{\wedge}2), \{w_{-}z_{-}\}\}) \text{ [ENTER]}$$

[.494... - .703...]

$$\text{cZeros}(\{e^{\wedge}(z_{-} - w_{-}w_{-} - z_{-}^{\wedge}2), \{w_{-}z_{-}=1 + j\}\}) \text{ [ENTER]}$$

[.149...+4.89...·j 1.588...+1.540...·j]

**d()** [2nd] [d] 键或 MATH/Calculus 菜单

**d**(表达式1, var[, 阶数]) ⇒ 表达式

**d**数组1, var[, 阶数] ⇒ 数组

**d**矩阵1, var[, 阶数] ⇒ 矩阵

返回表达式1 关于变量 var 的一阶导数。表达式1 可以为数组或矩阵。

若上式中包含阶数(Order)，则它必须为整数。若阶数小于零，结果将为反导数。

**d()** 并非按照通常的求解方式来完全化简其自变量然后将函数定义应用于已完全化简的自变量。相反，**d()** 采用下列步骤：

1. 仅将第一自变量化简到不导致常量的程度即可。
2. 仅将第一自变量化简至它能为步骤1中所确定的变量调用任何已保存数值的程度即可。
3. 确定步骤2的计算结果关于步骤1中的变量的符号导数。
4. 若步骤1 得到的变量被赋以已保存数值或由“with” (|) 算子确定的数值，将该数值代入步骤3得到的结果。

$$d(3x^3 - x + 7, x) \text{ [ENTER]} \quad 9x^2 - 1$$

$$d(3x^3 - x + 7, x, 2) \text{ [ENTER]} \quad 18 \cdot x$$

$$d(f(x) \cdot g(x), x) \text{ [ENTER]}$$

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

$$d(\sin(f(x)), x) \text{ [ENTER]}$$

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

$$d(x^3, x) | x=5 \text{ [ENTER]} \quad 75$$

$$d(d(x^2 \cdot y^3, x), y) \text{ [ENTER]} \quad 6 \cdot y^2 \cdot x$$

$$d(x^2, x, -1) \text{ [ENTER]} \quad \frac{x^3}{3}$$

$$d(\{x^2, x^3, x^4\}, x) \text{ [ENTER]} \quad \{2 \cdot x \quad 3 \cdot x^2 \quad 4 \cdot x^3\}$$

**dayOfWk()** CATALOG

**dayOfWk**(year, month, day) ⇒ 整数

返回1到7间的一个整数，其意义为一周中的某一天。用**dayOfWk()**确定某特定日期为周几。

**注：**不能给出1583年前的准确结果（早于格里高利(Gregorian)历）。

输入4位整数表示的年份。月份和日期可以为1位或2位整数。

$$\text{dayOfWk}(1948, 9, 6) \quad 2$$

整数值：

1 = 星期日

2 = 星期一

3 = 星期二

4 = 星期三

5 = 星期四

6 = 星期五

7 = 星期六

## ►DD MATH/Angle 菜单

数字►DD ⇒ 数值  
数组1►DD ⇒ 数组  
矩阵1►DD ⇒ 矩阵

返回自变量的十进制等效值。自变量可以是设定为弧度或是角度模式的数字、数组、或矩阵。

注：►DD 也可以接受弧度形式的输入。

在Degree(角度)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角度模式下：

1.5 ►DD [ENTER] 85.9°

## ►Dec MATH/Base 菜单

整数1►Dec ⇒ 整数

将整数1 转化为十进制数（以10为基）。二进制或十六进制形式的输入必须分别采用0b或0h前缀。

└─ 零，而不是字母O，后面为b或h。

0b 二进制数

0h 十六进制数

└─ 二进制数可多达32位。十六进制数可多达8位。

若不带前缀，整数1 视为十进制数处理。无论何种Base模式，结果均显示为十进制。

0b10011►Dec [ENTER] 19

0h1F►Dec [ENTER] 31

## Define CATALOG

Define funcName(arg1Name, arg2Name, ...) = 表达式

创建名称为 funcName 的用户自定义函数。您此后可以使用 funcName(), 使用方法同内建函数。该函数使用所提供的自变量求表达式的值并返回结果。

funcName 不能取系统变量或内建函数的名称。

自变量名称为占位符；您在使用该函数时不得使用相同名称的自变量。

注：此形式的Define 指令等同于执行以下表达式：表达式►funcName(arg1Name, arg2Name)。此命令也可用于定义简单变量；例如，定义 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] D

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]

{ $\frac{2 \cdot \sqrt{3} - 1}{11}$  ;  $\frac{-(2 \cdot \sqrt{3} + 1)}{11}$ }

Define funcName(arg1Name, arg2Name, ...) = Func  
block  
EndFunc

与上面的Define指令相同，只是采用这种形式，用户自定义函数funcName() 可以执行多条语句。

block 可以包含单条或一系列用字符"."分隔的语句。block 也可以包括表达式和指令（如 If、Then、Else、和 For）。这使得函数 funcName() 可以使用Return 指令返回特定的结果。

注：该函数形式的使得在程序编辑器中创建和编辑程序模块要比在输入行中方便。

Define g(x,y)=Func:If x>y Then

:Return x:Else:Return y:EndIf

:EndFunc [ENTER] Done

g(3, -7) [ENTER] 3

**Define** *progName*(*arg1Name*, *arg2Name*, ...) = **Prgm**  
*block*  
**EndPrgm**

创建名称为 *progName* 的程序或子程序，但不使用 **Return** 返回结果。可以执行多条语句。

*block* 可以包含单条或一系列用字符“:”分隔的语句。*block* 也可以不受限制地包括表达式和指令 (如 **If**、**Then**、**Else**、和 **For**)。

**注:** 该函数形式的使得在程序编辑器中创建和编辑程序模块要比在输入行中方便。

**Define** Listpnt()**=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**

Listpnt() **[ENTER]** Enter name of List

## DelFold CATALOG

**DelFold** *folderName*1, *folderName*2 [, *folderName*3] ...

删除名称为 *folderName*1、*folderName*2 等用户自定义文件夹。若这些文件夹中包含任何变量均会显示出出错提示。

**注:** 您不能删除 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** *var*1, *var*2 [, *var*3] ...

从内存中删除指定的变量。

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**(*1stOr2ndOrderOde*, *independentVar*, *dependentVar*) ⇒ 通解

返回一个方程式，它显式或隐式地给出了一个1阶或2阶常微分方程(ODE)的通解。在 ODE 中：

- 使用1个微分符号 ('，按下 **[2nd]** **[t]**) 表示因变量关于自变量的1阶导数。
- 使用2个微分符号表示出对应的2阶导数。

上述“'”符号仅用于 **deSolve()** 中的导数。在其它情况下，使用 **d()**。

1阶方程的通解包含形式为 **@k** 的任意常数，此处 **k** 是从1到255取值的整数后缀。当您使用 **ClrHome** 或 **[F1] 8: Clear Home** 时该后缀复位为1。2阶方程式的解包含两个这样的常数。

**注:** 欲键入1阶符号(')，可按下 **[2nd]** **[t]**。

**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**

如果您希望将某个隐解转化为一个或多个等同的显解，可使用**solve()**。

在将您的结果与说明书或使用手册中的解进行比较时，请注意不同的方法会在计算中采用不同的任意常数，从而产生不同的通解。

**deSolve(y'=(cos(y))^2 \* x,x,y)** [ENTER]

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

**solve(ans(1),y)** [ENTER]

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

注：欲键入@ symbol，可按下：

**ans(1)@3=c-1 and @n1=0** [ENTER]

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

**deSolve(1stOrderOde and initialCondition, independentVar, dependentVar)**

⇒ 特解

返回满足 *1stOrderOde* (1阶常微分方程) 和 *initialCondition* (初始条件) 的特解。这通常比确定通解、代入初始数值、取任意常数求解、然后将数值代入通解要简单。

*initialCondition* 以如下方形式给出：

*dependentVar* (初始自变量数值) = 初始因变量数值

初始自变量数值和 初始因变量数值可以没有无已保存数值的变量，比如 *x0* 和 *y0*。隐函数微分法可以帮助验证隐解。

**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]

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

**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(2ndOrderOde and initialCondition1 and initialCondition2, independentVar, dependentVar)** ⇒ 特解

返回满足 *2ndOrderOde* (2阶常微分方程) 的特解，并给定因变量及其1阶导数在某一点的值。

对于 *initialCondition1*，使用以下形式：

*dependentVar* (初始自变量数值) = 初始因变量数值

对于 *initialCondition2*，使用以下形式：

*dependentVar*<sup>1</sup> (初始自变量数值) = 初始1阶导数数值

**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]

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

**deSolve(2ndOrderOde and boundaryCondition1 and boundaryCondition2, independentVar, dependentVar) ⇒ 特解**

返回满足2ndOrderOde (2阶常微分方程) 的特解, 并给定其在2个不同点的值。

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

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

## det() MATH/Matrix 菜单

**det(方阵, tol) ⇒ 表达式**

返回方阵的行列式。

作为可选项, 如果矩阵的任何元素的绝对值小于 *tol*, 它将按零值处理。仅当矩阵具有浮点输入项且不含任何未赋值的符号变量时使用此容许值。否则, *tol* 会被忽略。

- 若您使用  $\blacktriangledown$  [ENTER] 或设定模式为 Exact/Approx=APPROXIMATE, 则计算过程会使用浮点算法完成。
- 若 *tol* 被省略或未使用, 缺省的容许值计算如下:

$$5E-14 * \max(\dim(\text{方阵}) * \text{rowNorm}(\text{方阵}))$$

**det([a,b;c,d]) [ENTER]**  $a \cdot d - b \cdot c$

**det([1,2;3,4]) [ENTER]**  $-2$

**det(identity(3) - x \* [1, -2, 3; -2, 4, 1; -6, -2, 7]) [ENTER]**  
 $-(98 \cdot x^3 - 55 \cdot x^2 + 12 \cdot x - 1)$

**[1E20,1;0,1]→mat1**  $\begin{bmatrix} 1.E20 & 1 \\ 0 & 1 \end{bmatrix}$

**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**

返回一个矩阵, 其主对角线上为自变量数组或矩阵中的数值。

**diag({2,4,6}) [ENTER]**  $\begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix}$

**diag(方阵) ⇒ 行矩阵**

返回一个行矩阵, 包含方阵的主对角线上的元素。

方阵必须为矩形。

**[4,6,8;1,2,3;5,7,9] [ENTER]**  $\begin{bmatrix} 4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$

**diag(ans(1)) [ENTER]**  $[4 \ 2 \ 9]$

## Dialog CATALOG

**Dialog**  
*block*  
**EndDlog**

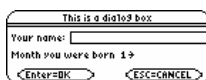
在程序执行时生成一个对话框。

*block* 可以包含单条或一系列用字符“.”分隔的语句。在程序编辑器的 [F3] I/O, 1:Dialog 菜单项中的有效的 *block* 选项为 1:Text、2:Request、4:DropDown、和 7:Title。

对话框里的变量可以被赋值, 这些值将显示为缺省值(或初始值)。若按下 [ENTER], 则根据对话框内容更新变量, 且变量 *ok* 被设置为 1。若按下 [ESC], 则对话框里的变量不会被更新, 且系统变量 *ok* 被设置为 0。

程序清单:

```
: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

<b>dim(数组)</b> ⇒ 整数 返回数组的维数。	<b>dim({0,1,2})</b> [ENTER]	3
<b>dim(矩阵)</b> ⇒ 数组 以二维数组 {行,列} 的形式返回矩阵的维数。	<b>dim({1, -1,2; -2,3,5})</b> [ENTER]	{2 3}
<b>dim(string)</b> ⇒ 整数 返回字符串 <i>string</i> 中包含的字符数量。	<b>dim("Hello")</b> [ENTER]	5
	<b>dim("Hello"&amp;" there")</b> [ENTER]	11

## Disp CATALOG

**Disp** [exprOrString1] [, exprOrString2] ...

显示当前程序输入/输出屏幕上的内容。若指定了一个或多个 *exprOrString*, 则各表达式或字符串显示在程序输入/输出屏幕的不同的行中。

一个表达式可以包含转换运算, 如 **►DD** 和 **►Rect** 等。您也可以使用 **►** 算子进行单位或数基的转换。

若 **Pretty Print = ON**, 则表达式以“清楚显示”方式显示。

在程序输入/输出屏幕上, 您可以按 **[F5]** 显示主屏幕, 也可以在程序中使用 **DispHome** 命令。

**Disp "Hello"** [ENTER] Hello

**Disp cos(2.3)** [ENTER] .666...


**{1,2,3,4}►L1** [ENTER] {1 2 3 4}

**Disp L1** [ENTER] {1 2 3 4}

**Disp 180\_min►\_hr** [ENTER] 3.◦\_hr

**注:** 欲键入下划线 (  ), 可按下:

 **[ ]** **[\_]**

 **[2nd]** **[\_]**

欲键入 **►**, 可按下 **[2nd]** **[►]**。

## DispG CATALOG

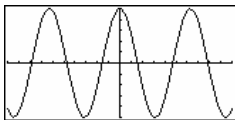
**DispG**

显示Graph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Table屏幕的当前内容。

**注：**光标键(cursor pad)处于滚动激活状态。如果在一个程序当中，可按下[ESC]或[ENTER]继续执行程序。

5 \* cos(x) → y1(x) [ENTER]

DispTbl [ENTER]

Flt	Fix	Eng	Pol	Norm	Table
x	u1				
-2.	-2.681				
-1.	2.7015				
0.	5.				
1.	2.7015				
2.	-2.681				
x=-2.					
MIN	END AUTO				FUNC

## ►DMS MATH/Angle 菜单

表达式 ►DMS

数组 ►DMS

矩阵 ►DMS

以角度形式表示自变量并显示等效的DMS (DDDD° MM SS.ss") 值。关于DMS (度、分、秒) 格式可以参见268上的°, ', ''。

**注：**►DMS 在弧度模式下使用时会将弧度转换为角度。若输入值后跟着角度符号(°), 则不会进行转换。您可以仅在每输入行的末尾使用►DMS。

在Degree(角度)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 ops 菜单

dotP(数组1, 数组2) ⇒ 表达式

返回两个数组的“点”积。

dotP({a,b,c},{d,e,f}) [ENTER]

$a \cdot d + b \cdot e + c \cdot f$

dotP({1,2},{5,6}) [ENTER]

17

dotP(向量1, 向量2) ⇒ 表达式

返回两个向量的“点”积。

两个向量必须同为行向量, 或同为列向量。

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

$a \cdot d + b \cdot e + c \cdot f$

dotP([1,2,3],[4,5,6]) [ENTER]

32

## DrawFunc CATALOG

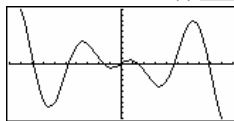
DrawFunc 表达式

将表达式视为函数进行作图, 使用x作为自变量。

**注：**重新作图将清除所有已经画出的项目。

在函数作图模式下的ZoomStd 窗口中:

DrawFunc 1.25x\*cos(x) [ENTER]



## DrawInv CATALOG

DrawInv 表达式

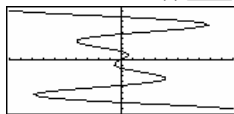
通过在y轴上绘制x值和和x轴上绘制y值, 从而作出表达式的反转图。

x 为自变量。

**注：**重新作图将清除所有已经画出的项目。

在函数作图模式下的ZoomStd 窗口中:

DrawInv 1.25x\*cos(x) [ENTER]



## DrawParm CATALOG

**DrawParm** 表达式1, 表达式2  
[, tmin] [, tmax] [, tstep]

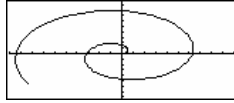
画出参数方程表达式1 和表达式2的图形, t 为自变量。

其中 tmin、tmax、和 tstep 的缺省值为窗体变量 tmin、tmax、和 tstep 的当前设定值。指定数值不会改变窗体的设定。若当前作图模式不是参数模式, 则这3个自变量均需要。

**注:** 重新作图将清除所有已经画出的项目。

在函数作图模式下的ZoomStd 窗口中:

**DrawParm** t\* cos(t),t\* sin(t),0,10,.1 [ENTER]



## DrawPol CATALOG

**DrawPol** 表达式[, θmin] [, θmax] [, θstep]

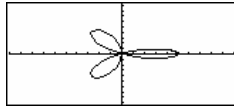
用θ作为自变量, 画出表达式的极坐标图形。

其中 θmin、θmax、和θstep 的缺省值为窗体变量 θmin、θmax、和θstep 的当前设定值。指定数值不会改变窗体的设定。若当前作图模式不是极坐标模式, 则这3个自变量均需要。

**注:** 重新作图将清除所有已经画出的项目。

在函数作图模式下的ZoomStd 窗口中:

**DrawPol** 5\* cos(3\* θ),0,3.5,.1 [ENTER]



## DrawSlp CATALOG

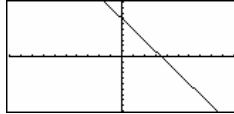
**DrawSlp** x1, y1, slope

显示图形并根据公式  $y - y_1 = \text{slope} \cdot (x - x_1)$  画一条线。

**注:** 重新作图将清除所有已经画出的项目。

在函数作图模式下的ZoomStd 窗口中:

**DrawSlp** 2,3, - 2 [ENTER]



## DropDown CATALOG

**DropDown** titleString, {item1String, item2String, ...},  
varName

显示名称为 titleString 的下拉菜单, 其中包含项目为 1:item1String, 2:item2String, 诸如此类。**DropDown**必须位于 **Dialog...EndDialog** 区间内。

若 varName 已经存在, 并有一个包含在项目范围内的值, 则被引用的项目将作为缺省选项显示出来。否则, 菜单的第1个选项为缺省选择。

当您从菜单中选择项目时, 该项目的相应编号存入变量 varName中 (若需要, **DropDown** 会创建 varName。)

参见 Dialog 示例的程序清单:



## DrwCtour CATALOG

**DrwCtour 表达式**

**DrwCtour 数组**

根据 **表达式** 或 **数组** 指定的  $z$  值在当前3维图形上作出等高线。3维作图模式必须已经设定。**DrwCtour** 自动将图形格式类型设置为 CONTOUR LEVELS (等高线)。

缺省状态下, 图形会自动包含由窗体变量 `ncontour` 确定的等距等高线的数量。

**DrwCtour** 可画出缺省情形以外的等高线。

要关闭缺省等高线, 需将设定 `ncontour` 为零, 可通过 **Window** 屏幕或直接给系统变量 `ncontour` 赋0值。

在3维作图模式下:

```
(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]
```



- 使用光标来改变视角。按下0 (零)恢复初始视角。

欲在不同图形格式类型间进行切换, 可按下:



F

- 按下 X、Y、或 Z 来查看相应的坐标。

**E** **EE** 键 **2nd** **EE** 键

**尾数E指数**

输入以科学计数法表示的数字。该数字表示 **尾数** × **10 指数**。

**提示:** 若您希望输入10的幂而不出现小数结果, 可使用  $10^{\wedge}$  整数。

```
2.3E 4 [ENTER] 23000.
2.3E 9+4.1E 15 [ENTER] 4.1E 15
3* 10^4 [ENTER] 30000
```

**e^()** **[e^x]** 键 **2nd** **[e^x]** 键

**e^(表达式)** ⇒ **表达式**

返回以  $e$  为底, 以 **表达式** 为幂的指数值。

**注:** 在 TI-89 Titanium 上, 按下 **[e^x]** 来显示  $e^x$  (是与按下 **[alpha]** **[E]** 有区别的)。在 Voyage™ 200 上, 使用 **[e^x]** 组合键来显示  $e^x$  与使用 QWERTY 键盘输入字符  $e$  是有区别的。

您可以输入形式为  $re^{i\theta}$  的极坐标复数。然而, 仅可以在 Radian (弧度) 模式下可以使用此形式; 而在 Degree (角度) 模式下会造成 Domain error 错误。

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

**e^(数组)** ⇒ **数组**

返回以  $e$  为底, 以 **数组** 中各元素为幂的指数值。

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

**e^(方阵)** ⇒ **方阵**

返回 **方阵** 的矩阵指数。该运算不同于计算矩阵各元素的  $e$  的幂。欲了解相关计算方法的具体说明, 请参见 **cos()**。

**方阵** 必须可对角化, 结果中总包含有浮点数。

```
e^({1,5,3;4,2,1;6,-2,1}) [ENTER]
782.209 559.617 456.509
680.546 488.795 396.521
524.929 371.222 307.879
```

**eigVc()** MATH/Matrix 菜单

eigVc(方阵) ⇒ 矩阵

返回一个矩阵，其中包含实数或复数方阵的特征向量，矩阵中各列对应于一个特征值。应注意特征向量并不是唯一的；改变常数因子可得到不同的特征向量。特征向量应规范化，即若  $V = [x_1, x_2, \dots, x_n]$ ，则：

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

方阵首先要通过相似变换进行平衡，直到行范数和列范数最大程度地接近。然后将方阵化简为上Hessenberg形式，并通过Schur因式分解来计算特征向量。

在直角坐标形式的复数格式模式下：

[-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]

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

**eigVl()** MATH/Matrix 菜单

eigVl(方阵) ⇒ 数组

返回一个矩阵，其中包含实数或复数类型方阵的特征值。

方阵首先要通过相似变换进行平衡，直至行范数和列范数最大程度地接近。然后将方阵化简为上Hessenberg形式，并通过上Hessenberg矩阵来计算特征值。

在直角坐标形式的复数格式模式下：

[-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]

$$\{-4.409\dots & 2.204\dots+.763\dots\cdot i \\ & 2.204\dots-.763\dots\cdot i\}$$

**Else** 参见193页上的**If**。**Elseif** CATALOG 也可参见193页上的**If**。**If** 布尔表达式1 Then

block1

**Elseif** 布尔表达式2 Then

block2

⋮

**Elseif** 布尔表达式N Then

blockN

**Endif**

⋮

**Elseif** 可以在程序分支里作为程序指令来使用。

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** 参见169页上的**Custom**。**EndDlog** 参见9页上的**Dialog**。**EndFor** 参见169页上的**For**。**EndFunc** 参见169页上的**Func**。**EndIf** 参见169页上的**If**。**EndLoop** 参见169页上的**Loop**。**EndPrgm** 参见169页上的**Prgm**。

**EndTBar** 参见169页上的**ToolBar**。

**EndTry** 参见169页上的**Try**。

**EndWhile** 参见169页上的**While**。

## entry() CATALOG

**entry()** ⇒ 表达式

在主屏幕上:

**entry(整数)** ⇒ 表达式

从主屏幕的历史区域中返回一个前面的输入项。

$1+1/x$  [ENTER]

$$\frac{1}{x} + 1$$

如果命令中包含了**整数**, 则返回历史区域中指定的输入表达式。缺省值为1, 指向最新的输入项。**整数**的有效范围为从1到99, 并且不能是表达式。

$1+1/\text{entry}(1)$  [ENTER]

$$2 - \frac{1}{x+1}$$

[ENTER]

$$\frac{1}{2 \cdot (2 \cdot x + 1)} + 3/2$$

**注:** 如果最近一个输入仍然在主屏幕上处于高亮显示状态, 则按下[ENTER]相当于执行**entry(1)**。

[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 [, tol])** ⇒ 表达式

**exact(.25)** [ENTER]

1/4

**exact(数组1 [, tol])** ⇒ 数组

**exact(矩阵1 [, tol])** ⇒ 矩阵

**exact(.333333)** [ENTER]

$$\frac{333333}{1000000}$$

无论Exact/Approx模式是否设定为返回与自变量等效的有理数, 应尽量使用Exact模式下的算法。

**exact(.33333,.001)**

1/3

*tol*指明了转换时的容许值, 缺省值为0(零)。

**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 string [, 表达式1] [, 表达式2] ...**

执行包含一系列Motorola 68000操作代码的string。这些代码以汇编语言程序的形式运行。如有必要, 可选表达式允许您将一个或多个自变量传递给程序。

要了解更多信息, 请登陆德州仪器的网页:  
<http://www.ti.com/calc>

**注意:** Exec能最大限度发挥微处理器的能力。因此, 请注意您可能因误操作而导致计算器锁死, 并造成数据丢失。我们建议您在**Exec**命令前对计算器进行备份。

## Exit CATALOG

### Exit

退出当前的 **For**、**While**、或 **Loop** 程序段。

**Exit** 不允许在3种循环结构外使用 (**For**、**While**、或 **Loop**)。

程序清单:

```

:0→temp
:For i,1,100,1
: temp+i→temp
: If temp>20
: Exit
:EndFor
:Disp temp

```

执行后的 **temp** 内容为:

21

## expList() CATALOG

**expList(表达式, var) ⇒ 数组**

检查用“or”分隔的方程式中含的表达式，并返回一个数组，其元素是形式为  $var=表达式$  的方程式的右侧。这为您提供了从 **solve()**、**cSolve()**、**fMin()** 和 **fMax()** 函数的结果中提取所含解值的简便方法。

**注:** 由于 **zeros** 和 **cZeros()** 函数直接返回包含解值的数组，因此 **expList()** 无需同它们配合使用。

$\text{solve}(x^2 - x - 2=0, x)$  [ENTER]  $x=2$  or  $x=-1$

$\text{expList}(\text{solve}(x^2 - x - 2=0, x), x)$  [ENTER]  $\{-1\ 2\}$

## expand() MATH/Algebra 菜单

**expand(表达式 $t$ , var) ⇒ 表达式**

**expand(数组 $t$ , var) ⇒ 数组**

**expand(矩阵 $t$ , var) ⇒ 矩阵**

**expand(表达式 $t$ )** 返回按其所有变量展开的表达式 $t$ 。对于多项式而言该展开为多项式展开，而对于有理表达式则为部分分式展开。

使用 **expand()** 的目的是将表达式 $t$  转换为简单项的和与 $t$  或差的形式。反之，使用 **factor()** 的目的是将表达式 $t$  转换为简单因子的积与/或商的形式。

**expand(表达式 $t$ , var)** 返回按  $var$  展开的表达式。各因式中  $var$  的同次幂将汇集在一起。各项及其因子按主变量  $var$  进行排列。可能出现对系数进行的伴随因式分解或伴随展开。与省略  $var$  的情况相比，该方式通常更节省时间、内存和屏幕空间，也使表达式更容易理解。

$\text{expand}((x+y+1)^2)$  [ENTER]  $x^2 + 2 \cdot x \cdot y + 2 \cdot x + y^2 + 2 \cdot y + 1$

$\text{expand}((x^2 - x + y^2 - y)/(x^2 \cdot y^2 - x^2 \cdot y - x \cdot y^2 + x \cdot y))$  [ENTER]

$$\text{expand}\left(\frac{x^2 - x + y^2 - y}{x^2 \cdot y^2 - x^2 \cdot y - x \cdot y^2 + x \cdot y}\right)$$

$\text{expand}((x+y+1)^2, y)$  [ENTER]  $y^2 + 2 \cdot y \cdot (x+1) + (x+1)^2$

$\text{expand}((x+y+1)^2, x)$  [ENTER]  $x^2 + 2 \cdot x \cdot (y+1) + (y+1)^2$

$\text{expand}((x^2 - x + y^2 - y)/(x^2 \cdot y^2 - x^2 \cdot y - x \cdot y^2 + x \cdot y), y)$  [ENTER]

$$\text{expand}\left(\frac{x^2 - x + y^2 - y}{x^2 \cdot y^2 - x^2 \cdot y - x \cdot y^2 + x \cdot y}, y\right)$$

$\text{expand}(\text{ans}(1), x)$  [ENTER]

$$\text{expand}\left(\frac{1}{y-1} - \frac{1}{y} + \frac{1}{x \cdot (x-1)}\right), x$$

即使在只有一个变量的情况下，在命令中使用 *var* 可以使为部分分式展开而进行的公分母因式分解更为完全。

**提示：**对于有理表达式，**propFrac()** 比 **expand()** 要快，但不能完全替代之。

**注：**参见 **comDenom()** 以便了解分子和分母均完全展开的形式。

无论是否带 *var*，**expand(表达式1,[var])** 也会将结果按对数项和分数幂项分开排列。对于升序排列的对数项和分数幂项，需要用不等式限制条件来保证某些因子为非负。

无论是否带 *var*，**expand(表达式1,[var])** 也会将结果按绝对值项、**sign()**项、以及指数项排列。

**注：**可参见 **tExpand()** 了解三角求和和多角度展开。

$$\text{expand}((x^3+x^2-2)/(x^2-2)) \text{ [ENTER]} \\ \frac{2 \cdot x}{x^2-2} + x+1$$

$$\text{expand}(\text{ans}(1),x) \text{ [ENTER]} \\ \frac{1}{x-\sqrt{2}} + \frac{1}{x+\sqrt{2}} + x+1$$

$$\ln(2x \cdot y) + \sqrt{(2x \cdot y)} \text{ [ENTER]} \\ \ln(2 \cdot x \cdot y) + \sqrt{(2 \cdot x \cdot y)}$$

$$\text{expand}(\text{ans}(1)) \text{ [ENTER]} \\ \ln(x \cdot y) + \sqrt{2} \cdot \sqrt{(x \cdot y)} + \ln(2)$$

$$\text{expand}(\text{ans}(1))|y>=0 \text{ [ENTER]} \\ \ln(x) + \sqrt{2} \cdot \sqrt{x} \cdot \sqrt{y} + \ln(2)$$

$$\text{sign}(x \cdot y) + \text{abs}(x \cdot y) + e^{(2x+y)} \text{ [ENTER]} \\ e^{2 \cdot x+y} + \text{sign}(x \cdot y) + |x \cdot y|$$

$$\text{expand}(\text{ans}(1)) \text{ [ENTER]} \\ \text{sign}(x) \cdot \text{sign}(y) + |x \cdot |y| + (e^x)^2 \cdot e^y$$

## expr() MATH/String 菜单

**expr(string)** ⇒ 表达式

以表达式形式返回 *string* 所包含的字符串并立即执行表达式。

$$\text{expr}("1+2+x^2+x") \text{ [ENTER]} \quad x^2 + x + 3$$

$$\text{expr}("\text{expand}((1+x)^2)") \text{ [ENTER]} \\ x^2 + 2 \cdot x + 1$$

"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]

进行指数回归并更新所有系统统计变量。

除数组5外，所有的数组必须具有相同的维数。

数组1代表x数组。

数组2代表y数组。

数组3代表频率。

数组4代表分类代码。

数组5代表分类所包含的数组。

**注：**数组1到数组4必须有变量名或取名为 c1-c99 (数据/矩阵编辑器中最末位的数据变量所代表的列)。数组5不一定要有变量名，也可以不取名为 c1-c99。

在函数图形模式下：

{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

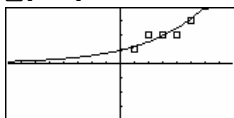
**ShowStat** [ENTER]



[ENTER] **Regeq(x)→y1(x)** [ENTER] Done

**NewPlot 1,1,L1,L2** [ENTER] Done

◻ [GRAPH]



## factor() MATH/Algebra 菜单

**factor**(表达式 $\ell$ , var)  $\Rightarrow$  表达式

**factor**(数组 $\ell$ , var)  $\Rightarrow$  数组

**factor**(矩阵 $\ell$ , var)  $\Rightarrow$  矩阵

**factor**(表达式 $\ell$ ) 返回一个关于所有变量的因式分解并带有公分母的表达式 $\ell$ 。

表达式 $\ell$ 应该尽量分解为线性有理因子，而不引入新的非实型表达式。如果您想进行关于多个变量的因式分解时，此方法是一种合理的选择。

**factor**(表达式 $\ell$ , var) 返回按变量 var 进行因式分解的表达式 $\ell$ 。

表达式 $\ell$ 应该尽量分解为关于变量 var 的线性有理因式，甚至引入了无理常数或关于其它变量的无理子表达式。

因式及其相关项将按主变量 var 进行分类。各因式中 var 的将汇集在一起。如果只进行关于变量 var 的因式分解，并且您允许在因式分解中存在关于其它变量的无理表达式，以进一步开展关于 var 的因式分解，则变量 var 必须包含在函数命令中。结果中可能出现关于其它变量的伴随因式分解。

如果 Exact/Approx 模式设置为 AUTO，该函数命令中包含变量 var 就意味着在无理系数不能采用内建函数进行简要清楚地表达时，可以采用浮点系数的近似计算。即使只有一个变量，包含变量 var 也可能产生更加完全的因式分解式。

**注：**参见 comDenom() 可了解在在 factor() 不够快或占用过多内存时，如何能更快地进行部分因式分解。

**注：**参见 cFactor() 可了解如何最大程度将复杂系数按一次因子进行因式分解。

**factor**(有理数) 返回有理数的素数分解。对于合数，计算时间按第二大因数的位数呈指数增长。例如，分解一个 30 位的整数需要一天以上的时间，而分解一个 100 位的数则需要一个世纪。

**注：**要停止（中断）计算，可以按下 **[ON]**。

如果您只是想确认某个数字是否为素数，使用 isPrime() 即可。这样速度更快，特别是当有理数不是素数且其第二大因数不超过 5 位时更加高效。

**factor**( $a^3 \cdot x^2 - a \cdot x^2 - a^3 + a$ ) **[ENTER]**

$a \cdot (a - 1) \cdot (a + 1) \cdot (x - 1) \cdot (x + 1)$

**factor**( $x^2 + 1$ ) **[ENTER]**

$x^2 + 1$

**factor**( $x^2 - 4$ ) **[ENTER]**

$(x - 2) \cdot (x + 2)$

**factor**( $x^2 - 3$ ) **[ENTER]**

$x^2 - 3$

**factor**( $x^2 - a$ ) **[ENTER]**

$x^2 - a$

**factor**( $a^3 \cdot x^2 - a \cdot x^2 - a^3 + a, x$ ) **[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]**

$(x + \sqrt{a}) \cdot (x - \sqrt{a})$

**factor**( $x^5 + 4x^4 + 5x^3 - 6x - 3$ ) **[ENTER]**

$x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3$

**factor**(ans(1), x) **[ENTER]**

$(x - .964\dots) \cdot (x + .611\dots) \cdot (x + 2.125\dots) \cdot (x^2 + 2.227\dots \cdot x + 2.392\dots)$

**factor**(152417172689) **[ENTER]**

123457 · 1234577

**isPrime**(152417172689) **[ENTER]**

false

## Fill MATH/Matrix 菜单

**Fill** 表达式 矩阵变量  $\Rightarrow$  矩阵

用表达式来替换变量 矩阵变量 中的各元素。

矩阵变量必须已经存在。

**[1,2;3,4]**  $\rightarrow$  **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 数组变量  $\Rightarrow$  数组

用表达式来替换变量 数组变量 中的各元素。

数组变量必须已经存在。

**{1,2,3,4,5}**  $\rightarrow$  **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

<b>floor(表达式)</b> ⇒ <i>integer</i>	<b>floor(-2.14)</b> [ENTER]	-3.
返回小于等于自变量的最接近的整数。该函数类似于 <b>int()</b> 。		
自变量可以是实数或复数。		
<b>floor(数组)</b> ⇒ <i>数组</i>	<b>floor({3/2,0,-5.3})</b> [ENTER]	{1 0 -6.}
<b>floor(矩阵)</b> ⇒ <i>矩阵</i>		
返回一个数组或矩阵，其组成为自变量各元素的floor函数值。	<b>floor([1.2,3.4;2.5,4.8])</b> [ENTER]	$\begin{bmatrix} 1.3. \\ 2.4. \end{bmatrix}$
<b>注：</b> 参加 <b>ceiling()</b> 和 <b>int()</b> 。		

## fMax() MATH/Calculus 菜单

<b>fMax(表达式 var)</b> ⇒ <i>布尔表达式</i>	<b>fMax(1-(x-a)^2-(x-b)^2,x)</b> [ENTER]	$x = \frac{a+b}{2}$
返回一个布尔表达式，它指定了变量 <i>var</i> 的候选值，该数值是表达式的最大值点或确定了表达式的最小上界。	<b>fMax(.5x^3-x-2,x)</b> [ENTER]	$x = \infty$
使用“ ”算子来限定求解区间和/或指定其它未定义变量的符号。	<b>fMax(.5x^3-x-2,x) x≤1</b> [ENTER]	$x = -.816\dots$
如果Exact/Approx模式设置为APPROX， <b>fMax()</b> 通过反复搜索来确定近似的局部最大值。这通常能提高运算速度，特别是当您使用“ ”算子将搜索范围限制在仅包含一个精确局部最大值的相对较小区间内时。	<b>fMax(a*x^2,x)</b> [ENTER]	$x = \infty$ or $x = -\infty$ or $x = 0$ or $a = 0$
<b>注：</b> 参见 <b>fMin()</b> 和 <b>max()</b> 。	<b>fMax(a*x^2,x) a&lt;0</b> [ENTER]	$x = 0$

## fMin() MATH/Calculus 菜单

<b>fMin(表达式 var)</b> ⇒ <i>布尔表达式</i>	<b>fMin(1-(x-a)^2-(x-b)^2,x)</b> [ENTER]	$x = \infty$ or $x = -\infty$
返回一个布尔表达式，它指定了变量 <i>var</i> 的候选值，该数值是表达式的最小值点或确定了表达式的最大下界。	<b>fMin(.5x^3-x-2,x) x≥1</b> [ENTER]	$x = 1$
使用“ ”算子来限定解的间隔和/或指定其它未定义变量的符号。	<b>fMin(a*x^2,x)</b> [ENTER]	$x = \infty$ or $x = -\infty$ or $x = 0$ or $a = 0$
如果Exact/Approx模式设置为APPROX， <b>fMin()</b> 通过反复搜索来确定近似的局部最小值。这通常能提高运算速度，特别是当您使用“ ”算子将搜索范围限制在仅包含一个精确局部最小值的相对较小区间内时。	<b>fMin(a*x^2,x) a&gt;0 and x&gt;1</b> [ENTER]	$x = 1.$
<b>注：</b> 参见 <b>fMax()</b> 和 <b>min()</b> 。	<b>fMin(a*x^2,x) a&gt;0</b> [ENTER]	$x = 0$

## FnOff CATALOG

<b>FnOff</b>
取消选择当前作图模式下的所有Y=functions。
在屏幕拆分的双图模式下， <b>FnOff</b> 仅对处于激活状态的图形起作用。

### FnOff [1] [, 2] ... [,99]

取消选择当前作图模式下指定的Y= functions。

在函数作图模式下:

FnOff 1,3 [ENTER] 取消选择 y1(x) 和 y3(x)。

在参变量作图模式下:

FnOff 1,3 [ENTER] 取消选择xt1(t)、yt1(t)、xt3(t)、和yt3(t)。

## FnOn CATALOG

### FnOn

选定当前作图模式下所定义的所有Y= functions。

在屏幕拆分的双图模式下, FnOn仅对处于激活状态的图形起作用。

### FnOn [1] [, 2] ... [,99]

选定当前作图模式下指定的Y= functions。

**注:** 在三维作图模式, 每次只能选定一个函数。FnOn 2 选定 z2(x,y) 并取消此前选定的任何函数。在其它图形模式下, 则不会影响到此前选定的函数。

## For CATALOG

For *var, low, high [, step]*  
*block*

### EndFor

从*low*到*high*, 每次增加值为*step*, 反复执行 *block* 中的语句。

*var* 不得为系统变量。

*step* 可以为正数也可以为负数。其缺省值为 1。

*block* 可以为单条语句, 也可以是"以字符:"分隔的一系列语句。

程序段:

```
⋮  
:0→tempsum : 1→step  
:For i,1,100,step  
: tempsum+i→tempsum  
:EndFor  
:Disp tempsum  
⋮
```

执行后的 tempsum 内容: 5050

当step变为2时的tempsum 内容: 2500

## format() MATH/String 菜单

format(*表达式*, *formatString*) ⇒ *string*

以格式模板形式返回字符串的*表达式*。

*表达式* 必须简化为数字。*formatString* 必须是如下形式的字符串: "F[n]"、"S[n]"、"E[n]"、"G[n][d]", 其中 [] 代表可选的比例。

F[n]: 固定格式。n为小数点后显示的数位。

S[n]: 科学计数格式。n为小数点后显示的数位。

E[n]: 工程格式。Engineering format。n为第一个有效数字后的数位。指数调整为3的倍数, 且小数点向右移0、1或2位。

format(1.234567,"f3") [ENTER] "1.235"

format(1.234567,"s2") [ENTER] "1.23E 0"

format(1.234567,"e3") [ENTER] "1.235E 0"

format(1.234567,"g3") [ENTER] "1.235"

format(1234.567,"g3") [ENTER] "1,234.567"

format(1.234567,"g3,r:") [ENTER] "1:235"



G[n]d: 与固定格式相同, 但也将小数点左边的数位每3个分为一组。c指定了分隔符, 其缺省值为逗号。若c为句号, 则小数点将显示为逗号。

[Rd]: 上述指定符可以加上一个以Rc小数点标记的后缀, 其中c为单个字符, 指明替代小数点的符号。

## fPart() MATH/Number 菜单

fPart(表达式) ⇒ 表达式  
fPart(数组) ⇒ 数组  
fPart(矩阵) ⇒ 矩阵

fPart(-1.234) [ENTER] - .234  
fPart({1, -2.3, 7.003}) [ENTER] {0 - .3 .003}

返回自变量的分数部分。

对于数组或矩阵, 返回其各元素的分数部分。

自变量可以为实数或复数。

## Func CATALOG

Func  
block  
EndFunc

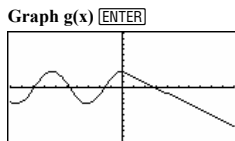
在函数作图模式下, 定义分段函数:

```
Define g(x)=Func:If x<0 Then
:Return 3*cos(x):Else:Return
3-x:EndIf:EndFunc [ENTER] Done
```

必须在多语句函数定义中作为第一条语句使用。

block 可以为单条语句, 也可以是以字符"."分隔的一系列语句。

注: when() 也可用来定义分段函数和进行分段函数作图。



## gcd() MATH/Number 菜单

gcd(number1, number2) ⇒ 表达式

gcd(18,33) [ENTER] 3

返回2个自变量的最大公约数。2个分数的gcd 值为其分子的gcd值除以其分母的lcm 值。

在Auto (自动) 或Approximate (近似) 模式下, 浮点分数的gcd 值是1.0。

gcd(数组1, 数组2) ⇒ 数组

gcd({12,14,16},{9,7,5}) [ENTER] {3 7 1}

返回数组1和数组2中对应元素的最大公约数。

gcd(矩阵1, 矩阵2) ⇒ 矩阵

gcd({2,4;6,8},{4,8;12,16}) [ENTER]  $\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$

返回矩阵1和矩阵2中相应元素的最大公约数。

## Get CATALOG

Get var

程序段:

从连接端口中获取一个CBL 2™/CBL™  
(Calculator-Based Laboratory™)(基于计算器的  
实验室™) 或 CBR™ (Calculator-Based Ranger™)  
(基于计算器的测距仪™)数值并存放在变量  
var中。

```
⋮
:Send {3,1, -1,0}
:For i,1,99
: Get data[i]
: PtOn i,data[i]
:EndFor
⋮
```

## GetCalc CATALOG

### GetCalc var

从连接端口获取一个数值并存放在变量 *var* 中。这用于机对机连接。

**注：**要通过连接端口从另一台设备获取变量，可使用另一台设备上的 **[2nd][VAR-LINK]** 选定并发送一个变量，或在另一台设备上使用 **SendCalc**。

程序段：

```
:
:Disp "Press Enter when ready"
:Pause
:GetCalc L1
:Disp "List L1 received"
:
```

### GetCalc var[,port]

从连接端口获取一个数值并存放在接收该数值的 TI-89 Titanium 计算器的变量 *var* 中。

若未指定端口，或指定 *port = 0*，则 TI-89 Titanium 将等待从任一端口获取数据。

若 *port = 1*，TI-89 Titanium 等待从 USB 端口获取数据。

若 *port = 2*，TI-89 Titanium 从 I/O 端口获取数据。

## getConfig() CATALOG

### getConfig() ⇒ 列表对

返回一个描述计算器属性的列表。在列表中，属性名称放在前面，随后为其数值。

 :

```
getConfig() [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}
```



```
getConfig() [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}
```

**注：**您的屏可能会显示不同的属性值。只有当您购买并安装了另外的程序后，才会显示 Cert. Rev. # 属性。

## getDate() CATALOG

### getDate() ⇒ 数组

getDate() [ENTER]

{2002 2 22}

返回一个数组，它包含的数据与当前时钟的数值相关。数组格式为{年、月、日}。

### getDenom() MATH/Algebra/Extract 菜单

getDenom(表达式*n*) ⇒ 表达式

将表达式*n*转化为具有化简公分母的形式，然后返回其公分母。

getDenom((x+2)/(y-3)) [ENTER] y - 3

getDenom(2/7) [ENTER] 7

getDenom(1/(x+(y^2+y)/y^2)) [ENTER] x \* y

### getDtFmt() CATALOG

getDtFmt() ⇒ 整数

返回代表设备当前日期格式的整数。

整数:

1 = MM/DD/YY

2 = DD/MM/YY

3 = MM.DD.YY

4 = DD.MM.YY

5 = YY.MM.DD

6 = MM-DD-YY

7 = DD-MM-YY

8 = YY-MM-DD

### getDtStr() CATALOG

getDtStr([整数]) ⇒ 字符串

返回以当前日期格式表示的描述当前日期的字符串。例如，返回的字符串为28/09/02代表2002年9月的第28天（当日期格式设定为DD/MM/YY时）。

如果您输入了与某个日期格式相对应的可选整数，字符串将以指定格式返回当前日期。

可选整数:

1 = MM/DD/YY

2 = DD/MM/YY

3 = MM.DD.YY

4 = DD.MM.YY

5 = YY.MM.DD

6 = MM-DD-YY

7 = DD-MM-YY

8 = YY-MM-DD

### getFold() CATALOG

getFold() ⇒ 名称字符串

以字符串形式返回当前文件夹的名称。

getFold() [ENTER] "main"

getFold()⇒ oldfoldr [ENTER] "main"

oldfoldr [ENTER] "main"

## getKey() CATALOG

getKey() ⇒ 整数

返回按键的键代码。若没有按下任何键，则返回0。

前缀键（上档键 **[1]**、第二功能键 **[2nd]**、钻石键 **[α]**、alpha键 **[alpha]**、和手形键 **[☞]**）自身没有键代码；但是，它们会改变紧随其后的键的代码。例如：**[☞][X] ≠ [X] ≠ [2nd][X]**。

要了解键代码的完整列表，可参见附录B。

程序清单：

```

: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

若自变量为指定的模式名称，则返回该模式当前设置情况的字符串。

若自变量为“ALL”，返回包含所有模式的设置情况的一组字符串对。若您希望在此后恢复模式设定，您必须将getMode("ALL")得到的结果存储在某个变量中，并使用setMode()来恢复原来的模式。

要了解模式名称和可能的设置情况的详细说明，可参见setMode()。

**注：**于设定或返回与Unit System模式有关的信息，使用setUnits()或getUnits()，而不要使用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"
  "Victor 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"}

```

**注：**您的屏幕可以显示不同的模式设置情况。

## getNum() MATH/Algebra/Extract 菜单

getNum(表达式) ⇒ 表达式

将表达式 $i$ 转化为具有化简公分母的形式，然后返回其分子。

```

getNum((x+2)/(y-3)) [ENTER] x + 2
getNum(2/7) [ENTER] 2
getNum(1/(x+1/y)) [ENTER] x + y

```

## getTime() CATALOG

getTime() ⇒ 数组

返回一个数组，它根据当前的时钟数值给出时间值。数组格式为{时、分、秒}。返回的时间值采用24小时制。

## getTmFmt() CATALOG

getTmFmt() ⇒ 整数

返回表示当前时钟制式的整数。

整数：

12 = 12小时制

24 = 24小时制

## getTimeStr() CATALOG

getTimeStr([integer]) ⇒ string

返回以当前时间格式表示的描述当前时间的字符串。

若您输入了与某个时钟格式相对应的可选整数，字符串将以指定格式返回当前时间。

可选整数：

12 = 12小时制

24 = 24小时制

## getTimeZn() CATALOG

getTimeZn() ⇒ 整数

返回表示计算器当前所设时区的整数。

返回的整数代表当前所设时区相对以格林尼治标准时间 (GMT) 的偏移分钟数。例如，若该时区相对与GMT的偏移为2小时，则计算器返回值为120 (分钟)。

代表GMT以西的时区的整数为负值。

代表GMT以东的时区的整数为正值。

若格林尼治标准时间为14:07:07，则：

在科罗拉多州丹佛为8:07:07 a.m. (山地白昼时间)  
(相对GMT为-360分钟)

在比利时布鲁塞尔为16:07:07 p.m. (中欧标准时间)  
(相对GMT为+120分钟)

## getType() CATALOG

getType(var) ⇒ 字符串

返回说明变量var数据类型的字符串。

若var未被定义，则返回字符串"NONE"。

{1,2,3} >	temp [ENTER]	{1 2 3}
getTime(temp)	[ENTER]	"LIST"
2+3 >	temp [ENTER]	2 + 3/
getTime(temp)	[ENTER]	"EXPR"
DelVar temp	[ENTER]	Done
getTime(temp)	[ENTER]	"NONE"

数据类型	变量内容
"ASM"	汇编语言程序
"DATA"	数据类型
"EXPR"	表达式 (包括复数/任意/未定义值、∞、-∞、TRUE、FALSE、pi、e)
"FUNC"	函数
"GDB"	图形数据库
"List"	数组
"MAT"	矩阵
"NONE"	变量不存在
"NUM"	实数
"OTHER"	用于今后软件应用的其它数据类型
"PIC"	图形
"PRGM"	程序
"STR"	字符串
"TEXT"	文本类型
"VAR"	另一个变量名称

## getUnits() CATALOG

getUnits() ⇒ 列表

返回一个字符串列表，其内容为除常数、温度、物质的量、光照强度、和加速度以外的所有类别的当前缺省单位。列表形式为：

```
{ "单位制" "类别1" "单位1" "类别2" "单位2" ... }
```

第一个字符串说明了单位制（SI、英/美、或自定义）。后续的字符串对说明了类别（如长度）和其缺省单位（如用于米的\_m）。

欲设置缺省单位，可使用 setUnits()。

getUnits() [ENTER]

```
{ "SI" "Area" "NONE"
  "Capacitance" "_F"
  "Charge" "_coul"
  ... }
```

注：您的屏幕可以显示不同的缺省单位。

## Goto CATALOG

Goto 标签名称

将程序控制转移至标签名称处。

标签名称必须在同一程序中使用Lbl指令进行定义。


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<sub>1</sub>, 表达式<sub>2</sub> [, var1] [, var2]

使用当前作图模式，Smart Graph 功能可绘制表达式/函数的图形。

使用Graph 或 Table 命令输入的表达式从1开始按升序进行函数编号。在按下  Header 显示表格时，可以使用所提供的编辑函数对上述函数分别进行修改或删除。当前选定的Y= functions 将不予考虑。

若省略了可选自变量 var，Graph 将使用当前作图模式的自变量。

注：由于您不能同时使用4个自变量，因此并非所有的可选自变量在所有模式中都有效。

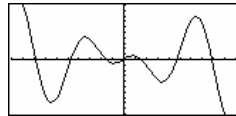
此指令的某些有效变化形式为：

- 函数作图                    **Graph** *expr*, *x*
- 参变量作图                **Graph** *xExpr*, *yExpr*, *t*
- 极坐标作图                **Graph** *expr*,  $\theta$
- 序列作图                   不允许。
- 3D作图                     **Graph** *expr*, *x*, *y*
- 微分方程作图              不允许。

注：使用 ClrGraph 可清除这些函数，或进入Y= 编辑器以重新激活Y= 函数。

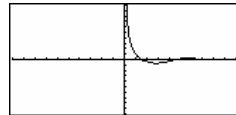
在函数作图模式和ZoomStd窗口：

Graph 1.25a\*cos(a),a [ENTER]



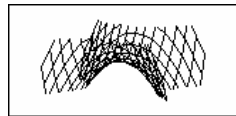
在参变量作图模式和ZoomStd窗口：

Graph time,2cos(time)/time,time [ENTER]



在三维作图模式下：

Graph (v^2 - w^2)/4,v,w [ENTER]



## Hex MATH/Base 菜单

整数  $I$  Hex  $\Rightarrow$  整数

256 Hex [ENTER]

0h100

将整数  $I$  转化为十六进制数。二进制或十六进制数通常分别具有形式为 0b 或 0h 的前缀。

0b111100001111 Hex [ENTER]

0hF0F

零，而不是字母 O，后面为 b 或 h。

0b 二进制数

0h 十六进制数

二进制数可长达 32 位。十六进制数可长达 8 位。

若不带前缀，整数  $I$  将视为十进制数处理（基于 10）。无论 Base 模式如何，结果均以十六进制显示。

若您输入的十进制整数超过了带符号的 32 位二进制数字的表示范围，对称求模运算将使该值落入正确的范围内。

## identity() MATH/Matrix 菜单

identity(表达式)  $\Rightarrow$  矩阵

identity(4) [ENTER]

返回维数为表达式的单位矩阵。

表达式必须为正整数。

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

## If CATALOG

If 布尔表达式  
statement

If 布尔表达式 Then  
block  
Endif

程序段:

若布尔表达式值为 true，在继续执行程序前先执行 statement 中的单条语句或 block 中的多条语句。

若布尔表达式的值为 false，继续执行程序，而无须执行 statement 中的单条语句或 block 中的多条语句。

block 内可以包含单条语句或一系列用字符“.”分隔的语句。

```
⋮  
: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  
block1

Else  
block2

Endif

程序段:

若布尔表达式的值为 true，执行 block1 并跳过 block2。

若布尔表达式的值为 false，跳过 block1 但执行 block2。

block1 和 block2 可以是单条语句。

```
⋮  
:If x<0 Then  
: Disp "x is negative"  
: Else  
: Disp "x is positive or zero"  
:EndIf  
⋮
```

<b>If</b> 布尔表达式1 Then block1 <b>Elseif</b> 布尔表达式2 Then block2 : <b>Elseif</b> 布尔表达式N Then blockN <b>Endif</b>	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 : :
允许程序有分支。若布尔表达式1的值为true, 执行 block1。若布尔表达式1的值为false, 则计算布尔表达式2的值, 依次类推。	

## imag() MATH/Complex 菜单

<b>imag(表达式<i>n</i>)</b> ⇒ 表达式 <b>imag(表达式<i>n</i>)</b> 返回自变量的虚部。 <b>注:</b> 所有未定义的变量均作为实变量处理。参见 <b>real()</b> 。	<b>imag(1+2<i>i</i>)</b> [ENTER]      2 <b>imag(<i>z</i>)</b> [ENTER]      0 <b>imag(<i>x</i>+<i>i</i>)</b> [ENTER] <i>y</i>
<b>imag(数组<i>n</i>)</b> ⇒ 数组 返回一个数组, 其元素为自变量数组中各元素的虚部。	<b>imag({-3,4-<i>i</i>})</b> [ENTER]      {0 -1 1}
<b>imag(矩阵<i>n</i>)</b> ⇒ 矩阵 返回一个矩阵, 其元素为自变量矩阵中各元素的虚部。	<b>imag({a,b;<i>i</i>,<i>i</i>})</b> [ENTER] $\begin{bmatrix} 0 & 0 \\ c & d \end{bmatrix}$

## Indirection See #(), page 267.

## Input CATALOG

<b>Input</b> 暂停程序执行, 显示当前的Graph屏幕, 并让您通过定位图形光标来更新变量 <i>xc</i> 和 <i>yc</i> (若极坐标模式下的 <i>rc</i> 和 <i>θd</i> )。 当您按下[ENTER]时, 程序将恢复执行。	程序段: : : :● Get 10 points from the Graph Screen :For i,1,10 : Input : xc→XList[i] : yc→YList[i] :EndFor : :
<b>Input [提示字符串] var</b> <b>Input [提示字符串], var</b> 会暂停程序执行, 并在程序I/O(输入/输出)屏幕上显示提示字符串, 等待您输入表达式, 并将此表达式存储在变量 <i>var</i> 中。 若您省略提示字符串, 将显示“?”作为提示符。	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,] *var*

暂停程序执行,并在程序I/O(输入/输出)屏幕上显示提示字符串,等待您的输入,并输入内容以字符串形式存储在变量*var*中。

若您省略提示字符串,将显示“?”作为提示符。

**注:** **Input** 和 **InputStr** 之间的区别是 **InputStr** 通常将结果以字符串形式存储,因此不需要使用“ ”。

程序段:

```
:  
:  
:InputStr "Enter Your Name",str1  
:
```

## inString() MATH/String 菜单

**inString**(*srcString*, *subString*, *start*) ⇒ 整数

返回在字符串 *srcString* 中首次出现 *subString* 的起始位置。

若指令中含 *start*, 则指定了在 *srcString* 中进行搜索的起始字符位置。缺省值 = 1 (*srcString* 的第一个字符)。

若 *srcString* 不包含 *subString* 或 *start* 比 *srcString* 的长度还大, 则返回0。

**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*n*) ⇒ 数组

**int**(矩阵*n*) ⇒ 矩阵

返回小于或等于自变量的最大整数。此功能与**floor()**类似。

自变量可以是实数或复数。

对于数组或矩阵, 返回其各元素的最大对应整数。

**int**(- 2.5) [ENTER]

- 3.

**int**([-1.234,0,0.37]) [ENTER]

[-2. 0 0.]

## intDiv() CATALOG

**intDiv**(数1, 数2) ⇒ 整数

**intDiv**(数组1, 数组2) ⇒ 数组

**intDiv**(矩阵1, 矩阵2) ⇒ 矩阵

返回自变量1除以自变量2的带符号的整数部分。

对于数组和矩阵, 其元素为自变量1中各元素除以自变量2中对应元素的带符号的整数部分。

**intDiv**(- 7,2) [ENTER]

- 3

**intDiv**(4,5) [ENTER]

0

**intDiv**({12, - 14, - 16},{5,4, - 3}) [ENTER]  
{2 - 3 5}

**integrate** 参见265页上的 **∫()**。

## iPart() MATH/Number 菜单

**iPart**(*number*) ⇒ 整数

**iPart**(数组*n*) ⇒ 数组

**iPart**(矩阵*n*) ⇒ 矩阵

返回自变量的整数部分。

对于数组或矩阵, 返回其各元素的整数部分。

自变量可以是实数或复数。

**iPart**(- 1.234) [ENTER]

- 1.

**iPart**({3/2, - 2.3,7.003}) [ENTER]

{1 - 2. 7.}

## isClkOn() CATALOG

isClkOn() ⇒ true,false

确认时钟处于开启或关闭状态。若时钟开启则返回true。若时钟关闭则返回OFF。

## isPrime() MATH/Test 菜单

isPrime(数字) ⇒ 布尔常数表达式

若数字为大于2的整数，则返回true或false来说明该整数是否仅能被它自己和1整除。

若number超过306位，且没有大于1021的因数，则isPrime(数字)显示出错误信息。

若您只是希望确认该数字是否为素数，可以使用isPrime()来代替factor()。这样运行速度更快，特别是当数字不是素数且其第二大因数超过5位数时更加高效。

isPrime(5) [ENTER] true  
isPrime(6) [ENTER] false

用于找出某个指定数字后的最近素数:

```
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, 标签

参见 Custom 中的示例。

只有在Custom...EndCustm 或ToolBar...EndTBar 程序段中, 该指令才有效。设置下拉式菜单项, 您可通过菜单项将文本粘贴至光标所在的位置(Custom)或分支转移至某个标签(ToolBar)。

注: 在 Custom 程序段中不允许分支转移至标签的指令。

## Lbl CATALOG

Lbl 标签名称

在程序中定义名称为标签名称的标签。

您可以使用 Goto 标签名称指令来将程序控制转移至紧接在标签后面的指令。

标签名称的必须符合与变量相同的命名要求。

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

返回2个自变量的最小公倍数。2个分数的lcm值等于其分子除以其分母的gcd值。浮点小数的lcm等于它们的乘积。

对于2个数组或矩阵, 返回其对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(sourceString, num) ⇒ string

返回字符串sourceString中最左边的num个字符。

若您省略num, 则返回整个sourceString。

left("Hello",2) [ENTER] "He"

<b>left(数组<i>l</i>, <i>num</i>)</b> ⇒ 数组	<b>left({1,3, -2,4},3)</b> <input type="button" value="ENTER"/>	{1 3 -2}
返回数组 <i>l</i> 中最左边的 <i>num</i> 个元素。 若您省略 <i>num</i> , 则返回整个数组 <i>l</i> 。		
<b>left(比较式)</b> ⇒ 表达式	<b>left(x&lt;3)</b> <input type="button" value="ENTER"/>	x
返回等式或不等式左侧部分。		

## limit() MATH/Calculus 菜单

<b>limit(表达式<i>l</i>, <i>var</i>, <i>point</i>, <i>direction</i>)</b> ⇒ 表达式	<b>limit(2x+3,x,5)</b> <input type="button" value="ENTER"/>	13
<b>limit(数组<i>l</i>, <i>var</i>, <i>point</i>, <i>direction</i>)</b> ⇒ 数组	<b>limit(1/x,x,0,1)</b> <input type="button" value="ENTER"/>	∞
<b>limit(矩阵<i>l</i>, <i>var</i>, <i>point</i>, <i>direction</i>)</b> ⇒ 矩阵	<b>limit(sin(x)/x,x,0)</b> <input type="button" value="ENTER"/>	1
返回所求极限。	<b>limit((sin(x+h)-sin(x))/h,h,0)</b> <input type="button" value="ENTER"/>	cos(x)
方向: 负值=左起, 正值=右起, 其它情况=两边。(若省略, 则方向的缺省值为两边)	<b>limit((1+1/n)^n,n,∞)</b> <input type="button" value="ENTER"/>	e
在正 ∞ 和负 ∞ 处的极限总会被转换为从有限侧趋近的单侧极限。		
视不同情况而定, <b>limit()</b> 在不能确定唯一极限的情况下, 返回自身或undef。但这并不意味着唯一极限不存在。undef表示结果是有限或无穷大的一个未知数, 或是此类数的集合。	<b>limit(a^x,x,∞)</b> <input type="button" value="ENTER"/>	undef
<b>limit()</b> 采用了如 L'Hopital (洛毕塔) 法则等方法, 因此某些唯一极限将无法确定。如果表达式 <i>l</i> 含除 <i>var</i> 以外的未定义变量, 您可以加上约束条件, 以得到更精确的结果。	<b>limit(a^x,x,∞) a&gt;1</b> <input type="button" value="ENTER"/>	∞
极限对舍入误差十分敏感。只要可能, 在计算极限时应避免采用Exact/Approx 模式的 APPROX (近似) 设置以及使用近似值。否则, 本应为0或无穷大的极限值将不会产生, 而有限非0极限值可能不会产生。	<b>limit(a^x,x,∞) a&gt;0 and a&lt;1</b> <input type="button" value="ENTER"/>	0

## Line CATALOG

**Line** *xStart*, *yStart*, *xEnd*, *yEnd*, *drawMode*

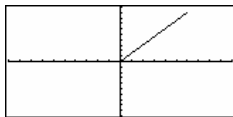
显示Graph (图形) 屏幕并画出、关闭、或反转位于窗口坐标(*xStart*, *yStart*) 和 (*xEnd*, *yEnd*) 间的线段, 包括2个端点。

若 *drawMode* = 1, 画出线段 (缺省情况)。  
若 *drawMode* = 0, 关闭线段。  
若 *drawMode* = -1, 将画出的线段关闭或将关闭的线段画出 (沿线段进行像素反转)。

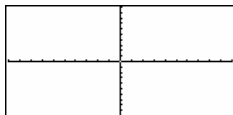
**注:** 重新作图会清除所有已画出的项目。  
参见 **PxlLine**。

在ZoomStd窗口中, 画出一条线段然后关闭。

**Line 0,0,6,9**



**Line 0,0,6,9,0**



## LineHorz CATALOG

**LineHorz**  $y$ ,  $drawMode$

显示Graph (图形) 屏幕并画出、关闭、或反转位于窗口位置 $y$ 的水平线。

若  $drawMode = 1$ , 画出线段 (缺省情况)。

若  $drawMode = 0$ , 关闭线段。

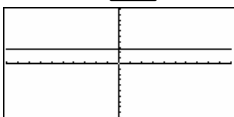
若  $drawMode = -1$ , 将画出的线段关闭或将关闭的线段画出 (沿线段进行像素反转)。

**注:** 重新作图会清除所有已画出的项目。

参见 **PxlHorz**。

在ZoomStd窗口下:

**LineHorz 2.5** [ENTER]



## LineTan CATALOG

**LineTan** 表达式1, 表达式2

显示Graph (图形) 屏幕并画出在指定点与表达式1的图形相切的直线。

表达式1为表达式或函数名, 其中 $x$ 为自变量, 且表达式2为相切点的 $x$ 值。

**注:** 在所例子中, 表达式1会单独画出。

**LineTan** 并不绘制表达式1的图形。

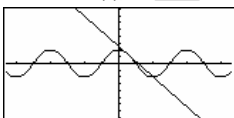
在函数作图模式和ZoomTrig窗口下:

**Graph cos(x)**

[HOME]

[CALC HOME]

**LineTan cos(x), $\pi/4$**  [ENTER]



## LineVert CATALOG

**LineVert**  $x$ ,  $drawMode$

显示Graph (图形) 屏幕并画出、关闭、或反转位于窗口位置 $x$ 的垂直线。

若  $drawMode = 1$ , 画出线段 (缺省情况)。

若  $drawMode = 0$ , 关闭线段。

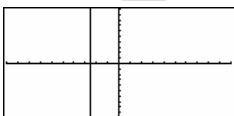
若  $drawMode = -1$ , 将画出的线段关闭或将关闭的线段画出 (沿线段进行像素反转)。

**注:** 重新作图会清除所有已画出的项目。

参见 **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的系统统计变量。

除数组5外, 所有的数组必须具有相同的维数。

数组1代表 $x$ 数组。

数组2代表 $y$ 数组。

数组3代表频率。

数组4代表分类代码。

数组5代表分类所包含的数组。

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]

**ShowStat** [ENTER]

Done

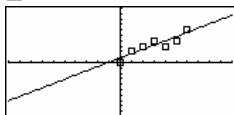


注：数组1到数组4必须有变量名或取名为c1-c99 (数据/矩阵编辑器中最末位的数据变量所代表的列)。数组5不一定要有变量名，也可以不取名为c1-c99。

ENTER  
Regeq(x)→y1(x) ENTER  
NewPlot 1,1,L1,L2 ENTER

Done  
Done

GRAPH



## ΔList() MATH/List 菜单

List(数组1) ⇒ 数组

ΔList({20,30,45,70}) ENTER

{10,15,25}

返回一个数组，其内容为数组1中相邻元素的差。数组1中各元素均被数组1中的下一个元素相减。所得数组通常比原来的数组1少一个元素。

## List→mat() MATH/List 菜单

List→mat(数组[, elementsPerRow]) ⇒ 矩阵

List→mat({1,2,3}) ENTER

[1 2 3]

返回一个矩阵，其内容为逐行将数组中的各元素填入。

List→mat({1,2,3,4,5},2) ENTER

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 0 \end{bmatrix}$

若指令中包含 elementsPerRow，则指明了每行的元素个数。缺省值为数组的元素个数（单行）。

若数组不能填满目标矩阵，则用0补入。

## ln() [LN] 键 [LN] 键

ln(表达式) ⇒ 表达式

ln(2.0) ENTER

.693...

ln(数组) ⇒ 数组

返回自变量的自然对数。

若复数格式模式为 REAL:

ln({-3,1,2,5}) ENTER

Error: Non-real result

对于数组，返回其各元素的自然对数值。

若复数格式模式为 RECTANGULAR:

ln({-3,1,2,5}) ENTER

{ln(3) + π · i .182... ln(5)}

ln(方阵) ⇒ 方阵

In Radian angle mode and Rectangular complex format mode:

返回的矩阵为方阵1的自然对数。这不同于计算矩阵中各元素的自然对数。欲了解有关计算方法的信息，可参见cos()。

ln({1,5,3;4,2,1;6,-2,1}) ENTER

方阵1必须可对角化。结果中总包含有浮点数。

$\begin{bmatrix} 1.831...+1.734... \cdot i & .009...-1.490... \cdot i & ... \\ .448...- .725... \cdot i & 1.064...+.623 \cdot i & ... \\ -.266...- 2.083... \cdot i & 1.124...+1.790... \cdot i & ... \end{bmatrix}$

## LnReg MATH/Statistics/Regressions 菜单

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

进行对数回归并更新所有的系统统计变量。

除数组5外, 所有的数组必须具有相同的维数。

数组1 代表x数组。

数组2代表y数组。

数组3代表频率。

数组4代表分类代码。

数组5代表分类所包含的数组。

**注:** 数组1 到 数组4 必须有变量名或取名为c1-c99 (数据/矩阵编辑器中最末位的数据变量所代表的列)。数组5 不一定要有变量名, 也可以不取名为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]

ShowStat [ENTER]



Done

[ENTER]

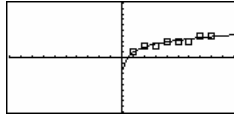
Regeq(x)→y1(x) [ENTER]

Done

NewPlot 1,1,L1,L2 [ENTER]

Done

[GRAPH]



## Local CATALOG

**Local** var1 [, var2] [, var3] ...

指定的 vars 为局部变量。这些变量仅在程序或函数求值过程中存在, 而当程序或函数执行结束后即被删除。

**注:** 由于局部变量是临时存在的, 因此可以节省内存。同时, 它们不会影响任何现有的全局变量值。局部变量必须用于For 循环及在多行函数中用于临时保存数值, 因为在函数中不得变更全局变量的值。

程序清单:

```
:prgname()
:Prgm
:Local x,y
:Input "Enter x",x
:Input "Enter y",y
:Disp x*y
:EndPrgm
```

**注:** x 和 y 在程序执行后不再存在。

## Lock CATALOG

**Lock** var1 [, var2] ...

为指定变量加锁。这使您不会在没有先对变量使用解锁指令的情况下就不小心删除或修改该变量值。

在右边的示例中, 变量L1 被加锁, 而且不能被删除或修改。

**注:** 可以使用Unlock 命令对变量进行解锁。

{1,2,3,4} > L1 [ENTER]

{1,2,3,4}

Lock L1 [ENTER]

Done

DelVar L1 [ENTER]

Error:Variableislockedorprotected

## log() CATALOG

**log(表达式*n*)** ⇒ 表达式

**log(数组*n*)** ⇒ 数组

返回自变量以10为底的对数值。

对于数组，返回其各元素以10为底的对数值。

**log(2.0)** [ENTER] .301...

若复数格式模式为 REAL:

**log({ - 3,1.2,5})** [ENTER]  
Error: Non-real result

若复数格式模式为 RECTANGULAR:

**log({ - 3,1.2,5})** [ENTER]  
$$\left\{ \frac{\ln(3)}{\ln(10)} + \frac{\pi}{\ln(10)} \cdot i \cdot .079 \dots \frac{\ln(5)}{\ln(10)} \right\}$$

**log(方阵*n*)** ⇒ 方阵

返回一个矩阵，其组成为方阵*n*的以10为底的对数。这不同于计算矩阵中各元素的以10为底的对数。欲了解其计算方法，可参见 **cos()**。

方阵*n*必须可对角化。结果中总包含有浮点数值。

在Radian 模式和直角坐标形式的复数格式模式下:

**log(1,5,3;4,2,1;6, - 2,1)** [ENTER]  
$$\begin{bmatrix} .795\dots + .753\dots \cdot i & .003\dots - .647\dots \cdot i & \dots \\ .194\dots - .315\dots \cdot i & .462\dots + .270\dots \cdot i & \dots \\ -.115\dots - .904\dots \cdot i & .488\dots + .777\dots \cdot i & \dots \end{bmatrix}$$

## Logistic MATH/Statistics/Regressions 菜单

**Logistic** 数组1, 数组2[, [重复次数], [数组3] [, 数组4, 数组5]

进行逻辑回归并更新所有的系统统计变量。

除数组5外，所有的数组必须具有相同的维数。

数组1代表*x*数组。

数组2代表*y*数组。

数组3代表频率。

数组4代表分类代码。

数组5代表分类所包含的数组。

重复次数指明了求解所需的最大次数。若省略该值，则缺省值为64。通常，取较大值时获得的精度更高，但需要更长的执行时间。反之亦然。

**注:** 数组1 到 数组4 必须有变量名或取名为 c1-c99 (数据/矩阵编辑器中最末位的数据变量所代表的列)。数组5 不一定要有变量名，也可以不取名为 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]

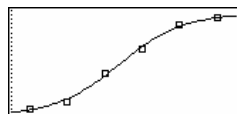
Done

NewPlot 1,1,L1,L2 [ENTER]

Done

[GRAPH]

F2 9



## Loop CATALOG

### Loop

*block*

### EndLoop

重复执行 *block* 中的语句。注意必须在 *block* 中执行 **Goto** 或 **Exit** 指令，否则会造成死循环。

*block* 为用字符“.”分隔的一系列语句。

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, *lMatName*, *uMatName*, *pMatName*, *tol*

计算实数或复数矩阵的Doolittle LU (下-上) 分解值。下三角矩阵存放在 *lMatName* 中, 上三角矩阵存放在 *uMatName* 中, 且置换矩阵 (描述在计算中完成的行交换) 存放在 *pMatName* 中。

*lMatName* \* *uMatName* = *pMatName* \* 矩阵

作为可选项, 如果矩阵的任何元素的绝对值小于 *tol*, 它将按零值处理。仅当矩阵具有浮点输入项且不含任何未赋值的符号变量时使用此容许值。否则, *tol* 会被忽略。

- 若您使用  $\square$  [ENTER] 或设置模式为 Exact/Approx=APPROXIMATE, 则计算过程会使用浮点算法完成。
- 若 *tol* 被省略或未使用, 缺省的容许值计算如下:

$$5E^{-14} * \max(\dim(\text{矩阵})) * \text{rowNorm}(\text{矩阵})$$

**LU** 的因式分解算法使用行交换的部分回转法。

[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;0,p]➤m1 [ENTER]

$$\begin{bmatrix} m & n \\ 0 & p \end{bmatrix}$$

LU m1,lower,upper,perm [ENTER]

Done

lower [ENTER]

$$\begin{bmatrix} 1 & 0 \\ m & 0 \\ 0 & 1 \end{bmatrix}$$

upper [ENTER]

$$\begin{bmatrix} 0 & p \\ 0 & n - \frac{m \cdot p}{0} \end{bmatrix}$$

perm [ENTER]

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$



## matList() MATH/List 菜单

<b>matlist(矩阵)</b> ⇒ 数组	<b>matlist({1,2,3})</b> <input type="button" value="ENTER"/>	{1 2 3}
返回一个数组，其组成元素为矩阵的元素。 这些元素从矩阵中逐行复制出来。	<b>[1,2,3;4,5,6] &gt; M1</b> <input type="button" value="ENTER"/>	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
	<b>matlist(M1)</b> <input type="button" value="ENTER"/>	{1 2 3 4 5 6}

## max() MATH/List 菜单

<b>max(表达式1, 表达式2)</b> ⇒ 表达式	<b>max(2,3,1,4)</b> <input type="button" value="ENTER"/>	2.3
<b>max(数组1, 数组2)</b> ⇒ 数组		
<b>max(矩阵1, 矩阵2)</b> ⇒ 矩阵	<b>max({1,2},{-4,3})</b> <input type="button" value="ENTER"/>	{1 3}

返回2个自变量中的最大值。若自变量为2个数组或矩阵，返回一个数组或矩阵组，其元素为这2个数组或矩阵的两个对应元素中的最大值。

<b>max(数组)</b> ⇒ 表达式	<b>max({0,1,-7,1.3,.5})</b> <input type="button" value="ENTER"/>	1.3
----------------------	--	-----

返回数组中的最大元素。

<b>max(矩阵<math>T</math>)</b> ⇒ 矩阵	<b>max({1,-3,7;-4,0,3})</b> <input type="button" value="ENTER"/>	[1 0 7]
-----------------------------------	--	---------

返回一个行向量，其元素为矩阵 $T$ 中各对应列中的最大元素。

注：参见 **fMax()** 和 **min()**。

## mean() MATH/Statistics 菜单

<b>mean(数组, 频率数组)</b> ⇒ 表达式	<b>mean({.2,0,1,-.3,.4})</b> <input type="button" value="ENTER"/>	.26
-----------------------------	---	-----

返回数组中各元素的平均值。

频率数组中的各元素为数组中对应元素出现的次数。

<b>mean({1,2,3},{3,2,1})</b> <input type="button" value="ENTER"/>	5/3
---	-----

<b>mean(矩阵<math>T</math>, 频率矩阵)</b> ⇒ 矩阵	在向量格式的直角坐标模式下：	
--	----------------	--

返回一个行向量，其元素为矩阵 $T$ 中各对应列元素的平均值。

<b>mean({1,2,0;-1,3;4,-.5})</b> <input type="button" value="ENTER"/>	[-.133... .833...]
--	--------------------

频率数组中的各元素为对矩阵 $T$ 中对应元素出现的次数。

<b>mean({1/5,0;-1,3;2/5,-1/2})</b> <input type="button" value="ENTER"/>	[-2/15 5/6]
---	-------------

<b>mean({1,2;3,4;5,6},{5,3;4,1,6,2})</b> <input type="button" value="ENTER"/>	[47/15, 11/3]
---	---------------

## median() MATH/Statistics 菜单

<b>median(数组)</b> ⇒ 表达式	<b>median({.2,0,1,-.3,.4})</b> <input type="button" value="ENTER"/>	.2
-------------------------	---	----

返回数组 $T$ 中元素的中值。

<b>median(矩阵<math>T</math>)</b> ⇒ 矩阵	<b>median({.2,0,1,-.3;4,-.5})</b> <input type="button" value="ENTER"/>	[.4 -.3]
--------------------------------------	--	----------

返回组成为矩阵 $T$ 各列中值的行向量。

注：数组或矩阵的所有输入必须简化为数字。

## MedMed MATH/Statistics/Regressions 菜单

MedMed 数组1, 数组2, [数组3], 数组4, 数组5]

进行中位数-中位数并更新所有的系统统计变量。

除数组5外, 所有的数组必须具有相同的维数。

数组1 代表x数组。

数组2代表y数组。

数组3代表频率。

数组4代表分类代码。

数组5代表分类所包含的数组。

注: 数组1 到 数组4 必须有变量名或取名为c1-c99 (数据/矩阵编辑器中最末位的数据变量所代表的列)。数组5 不一定要有变量名, 也可以不取名为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,L1,L2 [ENTER]

Done

ShowStat [ENTER]



[ENTER]

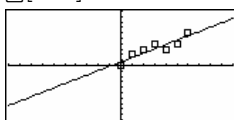
Regeq(x) > y1(x) [ENTER]

Done

NewPlot 1,1,L1,L2 [ENTER]

Done

GRAPH



## mid() MATH/String 菜单

mid(源字符串, 起点, 计数) ⇒ 字符串

返回源字符串中自第起点个字符开始的计数个字符所组成的字符串。

若计数被省略或大于源字符串的长度, 则返回从第起点个字符开始到源字符串末尾的所有字符所组成的字符串。

计数必须大于0。若计数 = 0, 则返回空字符串。

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, 起点, 计数) ⇒ 数组

返回源数组中自第起点个元素开始的计数个元素所组成的数组。

若计数被省略或大于源数组的长度, 则返回从第起点个元素开始到源数组末尾的所有元素所组成的数组。

计数必须大于0。若计数 = 0, 则返回空数组。

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, 起点, 计数) ⇒ 数组

返回源字符串数组中自第起点个元素开始的计数个字符串元素所组成的字符串数组。

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

返回2个自变量中的最小值。若自变量为2个数组或矩阵, 返回一个数组或矩阵组, 其元素为这2个数组或矩阵的两个对应元素中的最小值。

min(2.3,1.4) [ENTER]

1.4

min({1,2},{-4,3}) [ENTER]

{-4 2}

<b>min(数组)</b> ⇒ 表达式	<b>min({0,1,-7,1.3,..5})</b> [ENTER]	- 7
返回数组中的最小元素。		
<b>min(矩阵<i>n</i>)</b> ⇒ 矩阵	<b>min([1,-3,7;-4,0,3])</b> [ENTER]	[ - 4 - 3.3]
返回一个行向量，其元素为矩阵 <i>n</i> 中各对应列中的最小元素。		

注：参见 **fMin()** 和 **max()**。

### **mod()** MATH/Number 菜单

<b>mod(表达式1, 表达式2)</b> ⇒ 表达式	<b>mod(7,0)</b> [ENTER]	7
<b>mod(数组1, 数组2)</b> ⇒ 数组	<b>mod(7,3)</b> [ENTER]	1
<b>mod(矩阵1, 矩阵2)</b> ⇒ 矩阵	<b>mod(-7,3)</b> [ENTER]	2
按如下恒等式所定义，返回第1个自变量对第2个自变量取的模：	<b>mod(7,-3)</b> [ENTER]	- 2
$\text{mod}(x,0) = x$	<b>mod(-7,-3)</b> [ENTER]	- 1
$\text{mod}(x,y) = x - y \text{ floor}(x/y)$	<b>mod({12,-14,16},{9,7,-5})</b> [ENTER]	{3 0 -4}
当第2个自变量为非零时，其结果随该自变量呈周期性变化。结果要不为0，要不与第2个自变量具有相同的符号。		

若自变量为2个数组或矩阵，返回一个数组或矩阵，其元素为这2个数组或矩阵的两个对应元素的模值。

注：参见 **remain()**。

### **MoveVar** CATALOG

<b>MoveVar</b> <i>var</i> , 旧文件夹, 新文件夹	<b>{1,2,3,4} → L1</b> [ENTER]	{1 2 3 4}
将变量 <i>var</i> 从旧文件夹移动到 新文件夹。若新文件夹不存在，则 <b>MoveVar</b> 会创建之。	<b>MoveVar L1,Main,Games</b> [ENTER]	Done

### **mRow()** MATH/Matrix/Row ops 菜单

<b>mRow(表达式, 矩阵1, 索引)</b> ⇒ 矩阵	<b>mRow(-1/3,[1,2;3,4],2)</b> [ENTER]	$\begin{bmatrix} 1 & 2 \\ -1 & -4/3 \end{bmatrix}$
返回矩阵 <i>n</i> 的一个副本，其中第索引行的元素乘上表达式。		

### **mRowAdd()** MATH/Matrix/Row ops 菜单

<b>mRowAdd(表达式, 矩阵1, 索引1, 索引2)</b> ⇒ 矩阵	<b>mRowAdd(-3,[1,2;3,4],1,2)</b> [ENTER]	$\begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix}$
返回矩阵 <i>n</i> 的一个副本，其中第索引2行被替换为：	<b>mRowAdd(n,[a,b;c,d],1,2)</b> [ENTER]	$\begin{bmatrix} a & b \\ a \cdot n+c & b \cdot n+d \end{bmatrix}$
表达式 × 行索引1 + 行索引2		

## nCr() MATH/Probability 菜单

**nCr(表达式1, 表达式2)**  $\Rightarrow$  表达式

对于整数 表达式1 和 表达式2 且 表达式1  $\geq$  表达式2  $\geq 0$ , **nCr()** 表示从表达式1 件不同的东西中每次取出表达式2 件时可能的不同组合数 (这也称为二项式系数)。2 个自变量均可 为整数或符号表达式。

**nCr(表达式, 0)**  $\Rightarrow 1$

**nCr(表达式, 负整数)**  $\Rightarrow 0$

**nCr(表达式, 正整数)**  $\Rightarrow$  表达式  $\cdot$  (表达式 - 1) ...  
(表达式 - 正整数 + 1) 正整数!

**nCr(表达式, 非整数)**  $\Rightarrow$  表达式! / ((表达式 - 非整数)!  $\cdot$  非整数!)

**nCr(数组1, 数组2)**  $\Rightarrow$  数组

返回一个数组, 其元素是基于2 个数组中对应元素对的组合值。自变量必须是维数相同的数组。

**nCr(矩阵1, 矩阵2)**  $\Rightarrow$  矩阵

返回一个矩阵, 其元素是基于2 个矩阵中对应元素对的组合值。自变量必须是维数相同的矩阵。

$$\mathbf{nCr}(z, 3) \quad \frac{z \cdot (z-2) \cdot (z-1)}{6}$$

$$\mathbf{ans}(1)z=5 \quad 10$$

$$\mathbf{nCr}(z, c) \quad \frac{z!}{c!(z-c)!}$$

$$\mathbf{ans}(1)\mathbf{nPr}(z, c) \quad \frac{1}{c!}$$

$$\mathbf{nCr}(\{5,4,3\}, \{2,4,2\}) \text{ [ENTER]}$$
$$\{10 \ 1 \ 3\}$$

$$\mathbf{nCr}(\{6,5;4,3\}, \{2,2;2,2\}) \text{ [ENTER]}$$
$$\begin{bmatrix} 15 & 10 \\ 6 & 3 \end{bmatrix}$$

## nDeriv() MATH/Calculus 菜单

**nDeriv(表达式1, var, h)**  $\Rightarrow$  表达式

**nDeriv(表达式1, var, 数组)**  $\Rightarrow$  数组

**nDeriv(数组, var, h)**  $\Rightarrow$  数组

**nDeriv(矩阵, var, h)**  $\Rightarrow$  矩阵

以表达式形式返回数值导数。使用中央差商公式。

$h$  为步长数值。若  $h$  被省略, 则其缺省值为 0.001。

在使用 数组 或 矩阵 时, 该运算相当于求数组元素或矩阵元素的数值导数。

注: 参见 **avgRC()** 和 **d()**。

**nDeriv(cos(x), x, h)** [ENTER]

$$\frac{-(\cos(x-h) - \cos(x+h))}{2h}$$

**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 =  $\pi/2$**  [ENTER]

$$-1.$$

**nDeriv(x^2, x, {0.1, .1})** [ENTER]

$$\{2 \cdot x \ 2 \cdot x\}$$

## NewData CATALOG

**NewData dataVar, 数组1, 数组2 [, 数组3]...**

创建数据变量 **dataVar**, 其各列为依次排列列的各个数组。

必须至少有一个数组。

数组1, 数组2, ..., 数组  $n$  可以是如右图所示的列表, 也可以是能分解成数组的表达式, 或数组变量名。

**NewData** 创建的新变量是在数据/矩阵编辑器中的当前变量。

**NewData mydata, {1,2,3}, {4,5,6}** [ENTER]

Done

(进入数据/矩阵编辑器并打开 **var mydata** 来显示下面的数据变量。)

DATA			
1	c1	c2	c3
2	1	4	
3	2	5	
4	3	6	

**NewData dataVar, 矩阵**

创建基于 矩阵 的数据变量 **dataVar**。

**NewData sysData, 矩阵**

将 矩阵 内容放入系统数据变量 **sysData** 中。

## NewFold CATALOG

**NewFold** 文件夹名称

创建名称为文件夹名称的用户自定义文件夹，并将该文件夹设置为当前文件夹。在执行此指令后，您将在这个新文件夹操作。

**NewFold games** [ENTER]

Done

## newList() CATALOG

**newList(numElements)** ⇒ 数组

返回一个维数为 numElements 的数组。其元素均为 0。

**newList(4)** [ENTER]

{0 0 0 0}

## newMat() CATALOG 或 MATH/Matrix 菜单

**newMat(numRows, numColumns)** ⇒ 矩阵

返回全零矩阵，其行数为 numRows，列数为 numColumns。

**newMat(2,3)** [ENTER]

$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

## NewPic CATALOG

**NewPic** 矩阵, picVar[, maxRow][, maxCol]

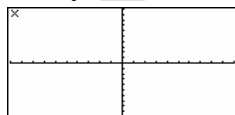
创建基于矩阵的图形变量 picVar。矩阵必须为  $n \times 2$  形式，其中各行代表一个像素。像素坐坐标起始点为 0,0。若 picVar 已经存在，NewPic 会将其替换。

picVar 的缺省值为矩阵值所需的最小面积。可选自变量 maxRow 和 maxCol 确定了 picVar 的最大边界。

**NewPic [1,1;2,2;3,3;4,4;5,5;  
5,1;4,2;2,4;1,5]**xpic [ENTER]

Done

**RelPic xpPic** [ENTER]



## NewPlot CATALOG

**NewPlot** *n*, 类型, *x*数组 [, *y*数组], [频率数组], [分类数组 (catlist)], [包含分类数组 (includeCatlist)], [标记] [, 柱状图尺寸]

为绘图编号 *n* 创建新的绘图定义。

类型指明了绘图的定义：

- 1 = 散点图
- 2 = xy线图
- 3 = 箱形图
- 4 = 柱状图
- 5 = 改良的箱形图

标记指定了标记符号的显示类型：

- 1 = □ (空心方块)
- 2 = × (十字)
- 3 = + (加号)
- 4 = ■ (实心方块)
- 5 = · (点状)

柱状图尺寸为每个柱形的宽度(类型 = 4)，且因窗口变量 xmin 和 xmax 的值而异。柱状图尺寸必须大于 0。缺省值 = 1。

注：*n* 可以为 1-9。数组必须有变量名或取值 c1-c99 (数据/矩阵编辑器中最末位的数据变量所代表的列)，包含分类数组除外，因为其没有变量名，也不能取值 c1-c99。

**FnOff** [ENTER]

Done

**PlotsOff** [ENTER]

Done

{1,2,3,4} → L1 [ENTER]

{1 2 3 4}

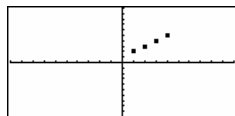
{2,3,4,5} → L2 [ENTER]

{2 3 4 5}

**NewPlot 1,1,L1,L2,,,4** [ENTER]

Done

Press  $\square$  [GRAPH] to display:



## NewProb CATALOG

NewProb	NewProb [ENTER]	Done
<p>执行若干运算，使您在已清空状态下开始求解一个新问题，而无需将内存复位。</p> <ul style="list-style-type: none"> <li>清除当前文件夹中的所有单字符变量 (Clear a-z)，除非该变量处于加锁或存档状态。</li> <li>关闭当前作图模式下的所有函数和统计图 (FnOff 和 PlotsOff)。</li> <li>执行 ClrDraw、ClrErr、ClrGraph、ClrHome、ClrIO、和 ClrTable。</li> </ul>		

## nInt() MATH/Calculus 菜单

<p><math>nInt(\text{表达式}1, \text{var}, \text{下限}, \text{上限}) \Rightarrow \text{表达式}</math></p> <p>如果被积函数表达式1未包含除var外的其它变量，且下限和上限为常数、<math>+\infty</math>或<math>-\infty</math>，则nInt()返回<math>\int(\text{表达式}1, \text{var}, \text{下限}, \text{上限})</math>的近似值。此近似值为被积函数在区间<math>[\text{下限}, \text{上限}]</math>上的一些样本值的加权平均。</p> <p>运算目标是获得6位有效数字。当目标实现后，或是在更多样本也不能对结果产生有意义的改善时，则所使用的算法将会终止。</p> <p>当目标难以实现时，将显示出警示信息 (“精度有问题(Questionable accuracy)”)。</p> <p>可嵌套nInt()来计算多重数值积分。积分极限可能取决于积分函数外部的积分变量。</p> <p>注：参见 <math>\int 0</math>。</p>	<p><math>nInt(e^{-(x^2)}, x, -1, 1)</math> [ENTER]</p> <p>1.493...</p> <p><math>nInt(\cos(x), x, -\pi, \pi + 1E-12)</math> [ENTER]</p> <p>- 1.041...E - 12</p> <p><math>\int(\cos(x), x, -\pi, \pi + 10^{-(12)})</math> [ENTER]</p> <p>- <math>\sin\left(\frac{1}{1000000000000}\right)</math></p> <p>ans(1) [ENTER]</p> <p>- 1.E - 12</p> <p><math>nInt(nInt(e^{-(x*y)}\sqrt{(x^2-y^2)}, y, -x, x), x, 0, 1)</math> [ENTER]</p> <p>3.304...</p>
--	--

## norm() MATH/Matrix/Norms 菜单

<p><math>norm(\text{矩阵}) \Rightarrow \text{表达式}</math></p> <p>返回Frobenius范数。</p>	<p><math>norm(\{a,b;c,d\})</math> [ENTER]</p> <p><math>\sqrt{a^2 + b^2 + c^2 + d^2}</math></p> <p><math>norm(\{1,2;3,4\})</math> [ENTER]</p> <p><math>\sqrt{30}</math></p>
--	--

## not MATH/Test 菜单

<p><math>not \text{布尔表达式}1 \Rightarrow \text{布尔表达式}</math></p> <p>返回值为 true、false、或化简的布尔表达式1。</p>	<p><math>not 2 &gt;= 3</math> [ENTER]</p> <p>true</p> <p><math>not x &lt; 2</math> [ENTER]</p> <p><math>x \geq 2</math></p> <p><math>not not innocent</math> [ENTER]</p> <p>innocent</p>
---	--

**not** 整数1  $\Rightarrow$  整数

返回实整数的补数。在计算器内部运算中，整数1被转换为一个带符号的32位二进制数。各位上的数值进行反转(0变成1，反之亦然)从而得到其补数。结果将按Base模式显示。

您可以按任何数基输入整数。对于按2进制或16进制输入的整数，您必须分别使用0b或0h作为前缀。若不带前缀，任何整数均会被视为10进制数(以10为基数)。

若您输入的十进制整数超过了带符号的32位二进制数字的表示范围，对称求模运算将使该值落入正确的范围内。

在十六进制模式下：

not 0h7AC36 [ENTER]      0hFFF853C9  
 └─ **重要提示：**是零，而不是字母O。

在二进制模式下：

0b100101►dec [ENTER]      37  
 not 0b100101 [ENTER]  
 0b111111111111111111111111111111110100  
 ans(1)►dec [ENTER]      - 38

**注：**二进制输入数可以达32位(不计前缀0b)。十六进制输入数可以达8位。

**注：**欲键入►转换算子，可按下[2nd][►]。您也可以从MATH/Base 菜单中选定基数转换。

**nPr()** MATH/Probability 菜单

nPr(表达式1, 表达式2)  $\Rightarrow$  表达式

对于整数 表达式1 和 表达式2 且 表达式1  $\geq$  表达式2  $\geq 0$ ，nPr()表示从表达式1件不同的东西中每次取出表达式2件时可能的不同排列数。2个自变量均可作为整数或符号表达式。

nPr(表达式, 0)  $\Rightarrow$  1

nPr(表达式, 负整数)  $\Rightarrow$  1/((表达式+1)·(表达式+2)·...·(表达式-负整数))

nPr(表达式, 正整数)  $\Rightarrow$  表达式·(表达式-1)·...·(表达式-正整数+1)

nPr(表达式, 非整数)  $\Rightarrow$  表达式!/(表达式-非整数)!

nPr(数组1, 数组2)  $\Rightarrow$  数组

返回一个数组，其元素是基于2个数组中对应元素对的排列值。自变量必须是维数相同的数组。

nPr(矩阵1, 矩阵2)  $\Rightarrow$  矩阵

返回一个矩阵，其元素是基于2个矩阵中对应元素对的排列值。自变量必须是维数相同的矩阵。

nPr(z,3) [ENTER]       $z \cdot (z-2) \cdot (z-1)$

ans(1)|z=5 [ENTER]      60

nPr(z, -3) [ENTER]       $\frac{1}{(z+1) \cdot (z+2) \cdot (z+3)}$

nPr(z,c) [ENTER]       $\frac{z!}{(z-c)!}$

ans(1)\*nPr(z-c, -c) [ENTER]      1

nPr({5,4,3},{2,4,2}) [ENTER]      {20 24 6}

nPr({6,5;4,3},{2,2;2,2}) [ENTER]       $\begin{bmatrix} 30 & 20 \\ 12 & 6 \end{bmatrix}$

**nSolve()** MATH/Algebra 菜单

nSolve(方程式, varOrGuess)  $\Rightarrow$  数字或出错\_字符串

对方程式的某个变量反复查找其真实解的近似值。指定 varOrGuess 为：

变量  
 - 或 -  
 变量 = 实数

例如，x 有效且 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.

**注：**若存在多个解，您可以使用估计值来帮助找到特定解。

**nSolve()** 通常比 **solve()** 或 **zeros()** 的快得多，特别是当您使用“|”算子将搜索范围限制在仅包含一个精确简单解的小区间内时。

**nSolve()** 将确定一个残差为0的点，或是确定两个比较靠近的点，其残差具有相反的符号并且数值不大。如果不能使用适当数量的样本点来得到以上结果，则返回字符串“no solution found（无解）”。

如果您在程序中使用 **nSolve()**，在其用在代数表达式中之前，您可以使用 **getType()** 来检查数值结果。

注：参见 **cSolve()**、**cZeros()**、**solve()**、和 **zeros()**。

$nSolve(x^2+5x-25=9,x)|x<0$  [ENTER] - 8.844...

$nSolve(((1+r)^24-1)/r=26,r)|r>0$  and  $r<25$  [ENTER] .0068...

$nSolve(x^2=-1,x)$  [ENTER] "no solution found"

## OneVar MATH/Statistics 菜单

**OneVar** 数组1 [, 数组2] [, 数组3] [, 数组4]

进行单统计变量并更新所有的系统统计变量。

除数组4外，所有的数组必须具有相同的维数。

数组1代表x数组。

数组2代表频率。

数组3代表分类代码。

数组4代表分类所包含的数组。

注：数组1到数组3必须有变量名或取名为c1-c99（数据/矩阵编辑器中最末位的数据变量所代表的列）。数组4不一定要有变量名，也可以不取名为c1-c99。

{0,2,3,4,3,4,6} > L1 [ENTER]

OneVar L1 [ENTER]

ShowStat [ENTER]

Done

STAT VARS	
$\Sigma$	=3.142857
$\Sigma x$	=22.
$\Sigma x^2$	=96.
$\Sigma x$	=1.864454
nStat	=7.
min1	=0.
st	=2.
medStat	=3.

<Enter>=BK

## or MATH/Test 菜单

布尔表达式1 or 布尔表达式2 ⇒ 布尔表达式

返回true或false或是原始输入值的化简形式。

若其中一个或两个表达式化简为true，则返回true。仅当两个表达式的计算结果均为false时才返回false。

注：参见 **xor**。

$x \geq 3$  or  $x \geq 4$  [ENTER]

$x \geq 3$

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

使用or运算逐位比较2个实整数。在计算机器的内部运算中，2个整数均会被转换为带符号的32位二进制数。在相应位进行比较时，任一个位值为1则结果为1；仅当两个位均为0时结果才为0。返回值为位的比较结果，且按Base模式显示。

您可以按任何基数输入整数。对于按2进制或16进制输入的整数，您必须分别使用0b或0h作为前缀。若不带前缀，任何整数均会被视为10进制数（以10为基数）。

若您输入的十进制整数超过了带符号的32位二进制数字的表示范围，对称求模运算将使该值落入正确的范围内。

注：参见 **xor**。

在十六进制基数模式下：

0h7AC36 or 0h3D5F [ENTER] 0h7BD7F

重要提示：是零，而不是字母O。

在二进制基数模式下：

0b100101 or 0b100 [ENTER] 0b100101

注：二进制输入数可以达32位（不计前缀0b）。十六进制输入数可以达8位。



## ord() MATH/String 菜单

**ord(字符串)** ⇒ 整数  
**ord(数组*n*)** ⇒ 数组

返回字符串的第一个字符的编码值，或返回一个数组，其元素为由数组*n*中各元素第一个字符的编码值组。

参见附录B 可了解字符编码的完整列表。

```
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 (输入/输出) 屏幕上的文本坐标 (行, 列) 处显示 表达式或字符串 (*exprOrString*)。

表达式可以包含如**►DD** 和 **►Rect**等转换运算。您也可以使用**►**算子对单位制和数基进行转换。

若 **Pretty Print = ON**, 则 *exprOrString* 为 "pretty printed (清楚显示)"。

在程序I/O (输入/输出) 屏幕上, 您可以按下**[F5]**显示主屏幕, 或在程序中使用 **DispHome**。

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
      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

返回 (*r, θ*) 对的等值*x*坐标值。

**注:** 根据当前的角度模式, *θ* 自变量可以是角度或弧度。若自变量为表达式, 您可以使用<sup>°</sup>或<sup>r</sup>来临时更改角度模式设置。

在Radian (弧度) 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

返回 (*r, θ*) 对的等值*y*坐标值。

**注:** 根据当前的角度模式, *θ* 自变量可以是角度或弧度。若自变量为表达式, 您可以使用<sup>°</sup>或<sup>r</sup>来临时更改角度模式设置。

在Radian (弧度) 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*n*, 非负整数)**

其先进的编程功能可让您找到并提取 表达式 *n* 的化简形式中的所有子表达式。

例如，若表达式 $t$ 化简为 $\cos(\pi * x + 3)$ ：

- 则 **cos()** 函数有一个自变量  $(\pi * x + 3)$ 。
- $(\pi * x + 3)$  之和有两个操作数： $\pi * x$  和 3。
- 数字 3 无自变量或操作数。
- 乘积  $\pi * x$  有两个操作数： $\pi$  和  $x$ 。
- 变量  $x$  和符号常数  $\pi$  无自变量或操作数。

若  $x$  已经赋有数值且您按下 **ENTER**，则计算  $\pi * x$  的数值，结果加到 3 上，然后计算余弦值。由于 **cos()** 在**最后**才使用，所以 **cos()** 为**级别最高**的算子。

<b>part(表达式<math>t</math>)</b> $\Rightarrow$ 数字	<b>part(cos(<math>\pi * x + 3</math>))</b> <b>ENTER</b>	1
化简表达式 $t$ 并返回级别最高的自变量或操作数的数量。若表达式 $t$ 为数字、变量、或如 $\pi$ 、 $e$ 、 $I$ 、或 $\infty$ 等符号常数，则该操作返回 0。	注：cos( $\pi * x + 3$ ) 有一个自变量。	
<b>part(表达式<math>t</math>, 0)</b> $\Rightarrow$ 字符串	<b>part(cos(<math>\pi * x + 3</math>), 0)</b> <b>ENTER</b>	"cos"
化简表达式 $t$ 并返回内容为级别最高的函数名称或操作数的字符串。若表达式 $t$ 为数字、变量、或如 $\pi$ 、 $e$ 、 $I$ 、或 $\infty$ 等符号常数，则该操作返回 <b>string(表达式<math>t</math>)</b> 。		
<b>part(表达式<math>t</math>, <math>n</math>)</b> $\Rightarrow$ 表达式	<b>part(cos(<math>\pi * x + 3</math>), 1)</b> <b>ENTER</b>	$3 + \pi * x$
化简表达式 $t$ 并返回第 $n$ 个自变量或操作数，其中 $n$ 大于 0 且小于等于由 <b>part(表达式<math>t</math>)</b> 返回的级别最高的自变量或操作数的数量。否则，将返回出错信息。	注：化简改变了自变量的次序。	

通过对 **part()** 的变化形式进行组合，您可以在表达式 $t$ 的化简结果中提取所有子表达式。如右边的例子所示，您可以存储自变量或操作数，并使用 **part()** 来进一步提取子表达式。

**注：**当使用 **part()** 时，在求和与求积时无须考虑的任何特定的顺序。

如  $(x+y+z)$  和  $(x-y-z)$  这样的表达式在计算器内部表示为  $(x+y)+z$  和  $(x-y)-z$ 。这会影响到返回给第一和第二自变量的值。因为技术上的原因，**part(x+y+z, 1)** 会返回  $y+x$  而不是  $x+y$ 。

类似地， $x*y*z$  在计算器内部表示为  $(x*y)*z$ 。同样，因为技术上的原因，第一自变量返回形式为  $y*x$  而不是  $x*y$ 。

<b>part(cos(<math>\pi * x + 3</math>))</b> <b>ENTER</b>	1
<b>part(cos(<math>\pi * x + 3</math>), 0)</b> <b>ENTER</b>	"cos"
<b>part(cos(<math>\pi * x + 3</math>), 1)</b> $\Rightarrow$ <b>temp</b> <b>ENTER</b>	$3 + \pi * x$
<b>temp</b> <b>ENTER</b>	$\pi * x + 3$
<b>part(temp, 0)</b> <b>ENTER</b>	"+"
<b>part(temp)</b> <b>ENTER</b>	2
<b>part(temp, 2)</b> <b>ENTER</b>	3
<b>part(temp, 1)</b> $\Rightarrow$ <b>temp</b> <b>ENTER</b>	$\pi * x$
<b>part(temp, 0)</b> <b>ENTER</b>	"*"
<b>part(temp)</b> <b>ENTER</b>	2
<b>part(temp, 1)</b> <b>ENTER</b>	$\pi$
<b>part(temp, 2)</b> <b>ENTER</b>	$x$
<b>part(x+y+z)</b> <b>ENTER</b>	2
<b>part(x+y+z, 2)</b> <b>ENTER</b>	$z$
<b>part(x+y+z, 1)</b> <b>ENTER</b>	$y+x$
<b>part(x*y*z)</b> <b>ENTER</b>	2
<b>part(x*y*z, 2)</b> <b>ENTER</b>	$z$
<b>part(x*y*z, 1)</b> <b>ENTER</b>	$y * x$

当您从矩阵中提取子表达式时，应记住矩阵是以数组列表的形式加以存储，如右侧例子所示。

```
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
```

右侧中的程序编辑器设计的函数中采用 **getType()** 和 **part()** 来部分执行符号微分。学习并完成该函数可帮助您学会如何手工进行微分运算。您甚至可以加入不可微的函数，如 **Bessel**（贝塞耳）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,∞,∫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

将出错信息传递至下一级。

若“errornum”为0，**PassErr** 不起作用。

程序中的 **Else** 子句应使用 **ClrErr** 或 **PassErr**。若出错信息被处理或省略，则使用 **ClrErr**。若不知如何处理出错信息，则使用 **PassErr** 将其发送到下一个出错信息处理句柄(参见 **ClrErr**。)

参见 **ClrErr** 程序清单示例：

## Pause CATALOG

### Pause [表达式]

暂停程序执行。若您纳入了表达式，则在程序I/O（输入/输出）屏幕上显示出来。

表达式可以包含如**DD**和**Rect**等转换运算。您也可以使用**▶**算子对单位制和数基进行转换。

若表达式的结果过大，超出了单屏显示范围，您可以使用光标键来滚动屏幕。

当您按下 **[ENTER]**时程序恢复执行。

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]

关闭用于作图的某些特定点。在双图模式下，仅影响激活的图形。

如果不带参数，则关闭所有点。

PlotsOff 1,2,5 **[ENTER]**

Done

PlotsOff **[ENTER]**

Done

## PlotsOn CATALOG

### PlotsOn [1] [, 2] [, 3] ... [, 9]

开启用于作图的某些特定点。在双图模式下，仅影响激活的图形。

如果不带参数，则开启所有点。

PlotsOn 2,4,5 **[ENTER]**

Done

PlotsOn **[ENTER]**

Done

## ▶Polar MATH/Matrix/向量 ops 菜单

### 向量 ▶Polar

以极坐标形式 $[r \angle \theta]$ 显示向量。向量必须为2维且呈行向量或列向量形式。

**注：**▶Polar 为显示格式指令，但不是转换指令。您可以仅在输入行的末尾使用它，且它不会更新 ans。

**注：**参见▶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（角度）模式返回  $(r \angle \theta)$ 。
- Radian（弧度）模式返回  $re^{i\theta}$ 。

复数值可以为任何复数形式。然而， $re^{i\theta}$ 形式的输入会在Degree（角度）模式产生错误。

**注：**您必须对 $(r \angle \theta)$ 形式的极坐标输入使用括号。

在Radian（弧度）模式下：

3+4▶Polar **[ENTER]**

$$e^{i \cdot \frac{\pi}{2} - \tan^{-1}(3/4)} \cdot 5$$

(4∠π/3)▶Polar **[ENTER]**

$$e^{\frac{i \cdot \pi}{3}} \cdot 4$$

在Degree（角度）模式下：

3+4▶Polar **[ENTER]**

$$(5 \angle 90 - \tan^{-1}(3/4))$$

## 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

将第一自变量看作一个降次多项式的系数来处理, 并返回该多项式, 用于计算第二自变量的值。

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 itemList, var

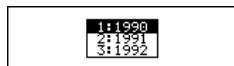
显示弹出式菜单, 其内容为 itemList 中的字符串, 并等待您选定一个项目, 然后将您选定的编号存储在 var 中。

itemList 的元素必须为字符串: { item1String, item2String, item3String, ...}

若 var 已经存在且有一个有效项目编号, 则将该项目作为缺省选择显示。

itemList 必须包含至少1个选项。

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]

进行幂回归并更新所有的系统统计变量。

除数组5外, 所有的数组必须具有相同的维数。

数组1 代表x数组。

数组2 代表y数组。

数组3 代表频率。

数组4 代表分类代码。

数组5 代表分类所包含的数组。

注: 数组1 到 数组4 必须有变量名或取名为 c1-c99 (数据/矩阵编辑器中最末位的数据变量所代表的列)。数组5 不一定要有变量名, 也可以不取名为 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 ...} Done

PowerReg L1,L2 [ENTER]

ShowStat [ENTER]

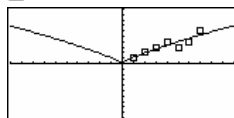


[ENTER]

Regeq(x)→y1(x) [ENTER] Done

NewPlot 1,1,L1,L2 [ENTER] Done

[GRAPH]



## Prgm CATALOG

Prgm

⋮

EndPrgm

用于标明程序开始所需的指令。程序的最后一行必须为 EndPrgm。

程序段:

:prgmmname()

:Prgm

:

:EndPrgm

## Product (PI) 参见第266页上的 Π()。

## product() MATH/List 菜单

**product**(数组[, 起点[, 终点]]) ⇒ 表达式

返回数组中所含元素的乘积。起点和终点为可选项。它们指明了元素的范围。

**product**({1,2,3,4}) [ENTER] 24

**product**({2,x,y}) [ENTER]  $2 \cdot x \cdot y$

**product**{4,5,8,9},2,3 [ENTER] 40

**product**(矩阵I[, 起点[, 终点]]) ⇒ 矩阵

返回由矩阵I中各列所含元素的乘积组成的行向量。起点和终点为可选项。它们指明了元素的范围。

**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** var1[, var2 [, var3] ...

在程序I/O (输入/输出) 屏幕上使用提示符 var1? 为自变量列表中的每个变量显示提示。将输入的表达式储存在相应的变量中。

**Prompt** 必须包含至少一个自变量。

程序段:

```
⋮  
Prompt A,B,C  
⋮  
EndPrgm
```

## propFrac() MATH/Algebra 菜单

**propFrac**(表达式I, var) ⇒ 表达式

**propFrac**(rational\_number) 返回的 rational\_number (有理数) 为一个整数和一个分数之和, 该整数和分数具有相同的符号, 且分母比分子大。

**propFrac**(rational\_expression, var) 返回适当比值之和, 它是关于变量 var 的多项式。在各个适当比值中分母中变量 var 的次数应大于分子中变量 var 的次数。各因式中 var 的同次幂汇集在一起。各项及其因子将按主变量 var 进行分类。

如果省略 var, 则得到一个关于主变量的适当的分数展开形式。应首先给出关于主变量形式的多项式部分的系数, 然后依次类推。

对于有理表达式, **propFrac**() 比 **expand**() 要快, 但不能完全替代之。

**propFrac**(4/3) [ENTER]  $1 + 1/3$

**propFrac**(-4/3) [ENTER]  $-1 - 1/3$

**propFrac**((x^2+x+1)/(x+1)+(y^2+y+1)/(y+1),x) [ENTER]

$$\text{propFrac} \left\{ \frac{x^2+x+1}{x+1} + \frac{y^2+y+1}{y+1} \right\}$$

**propFrac**(ans(1))

$$\text{propFrac} \left\{ \frac{1}{x+1} + x + \frac{y^2+y}{y+1} \right\}$$

## PtChg CATALOG

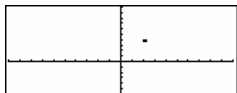
**PtChg** x, y

**PtChg** x数组, y数组

显示Graph (图形) 屏幕并对最靠近窗口坐标(x, y) 的屏幕像素进行反转。

**注: PtChg** 到 **PtText** 给出了一系列类似的例子。

**PtChg** 2,4 [ENTER]



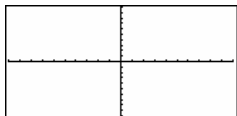
## PtOff CATALOG

**PtOff** x, y

**PtOff** x数组, y数组

显示Graph (图形) 屏幕并关闭最靠近窗口坐标(x, y) 的屏幕像素。

**PtOff** 2,4 [ENTER]

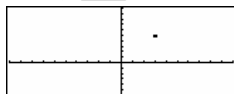


## PtOn CATALOG

PtOn  $x, y$   
PtOn  $x$ 数组,  $y$ 数组

显示Graph (图形) 屏幕并开启最靠近窗口坐标 $(x, y)$ 的屏幕像素。

PtOn 3,5 [ENTER]



## ptTest() CATALOG

ptTest ( $x, y$ )  $\Rightarrow$  布尔常数表达式  
ptTest ( $x$ 数组,  $y$ 数组)  $\Rightarrow$  布尔常数表达式

返回true或false。仅当最接近窗口坐标 $(x, y)$ 的屏幕像素处于开启状态时返回true。

ptTest(3,5) [ENTER]

true

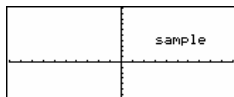
## PtText CATALOG

PtText  $string, x, y$

显示Graph (图形) 屏幕并将字符串 $string$ 放置在屏幕上最接近指定窗口坐标 $(x, y)$ 的像素处。

$string$ 的首字母的左上角处于上述坐标位置。

PtText "sample",3,5 [ENTER]



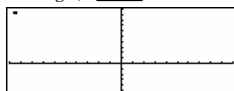
## PxlChg CATALOG

PxlChg 行, 列  
PxlChg 行数组, 列数组

显示Graph (图形) 屏幕并反转像素坐标(行, 列)处的像素。

**注:** 重新作图会清除所有已经画出的项目。

PxlChg 2,4 [ENTER]




## PxlCrcI CATALOG


PxlCrcI 行, 列,  $r$ ,  $drawMode$

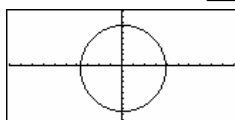
显示Graph (图形) 屏幕并以像素坐标(行, 列)为圆心,  $r$ 为半径作圆。

若  $drawMode = 1$ , 作圆(缺省情况)。  
若  $drawMode = 0$ , 关闭作出的圆。  
若  $drawMode = -1$ , 沿着圆形进行像素反转。

**注:** 重新作图将清除所有已经画出的项目。参见Circle。

 PxlCrcI 40,80,30,1 [ENTER]

 PxlCrcI 50,125,40,1 [ENTER]



## PxlHorz CATALOG

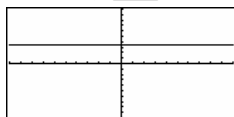
PxlHorz 行[,  $drawMode$ ]

显示Graph (图形) 屏幕并画出位于像素位置行的水平线。

若  $drawMode = 1$ , 画出线段(缺省情况)。  
若  $drawMode = 0$ , 关闭线段。  
若  $drawMode = -1$ , 将画出的线段关闭或将关闭的线段画出(沿线段进行反转像素)。

**注:** 重新作图会清除所有已画出的项目。参见LineHorz。

PxlHorz 25,1 [ENTER]



## PxlLine CATALOG

**PxlLine** 行起点, 列起点, 行终点, 列终点 [, drawMode]


显示Graph (图形) 屏幕并画出位于像素坐标(行起点, 列起点)和(行终点, 列终点)间的线段, 包括2个端点。


若 *drawMode* = 1, 画出线段 (缺省情况)。

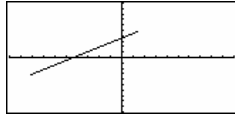
若 *drawMode* = 0, 关闭线段。

若 *drawMode* = -1, 将画出的线段关闭或将关闭的线段画出 (沿线段进行像素反转)。

**注:** 重新作图会清除所有已画出的项目。  
参见Line。

 **PxlLine 50,15,20,90,1** [ENTER]

 **PxlLine 80,20,30,150,1** [ENTER]



## PxlOff CATALOG

**PxlOff** 行, 列

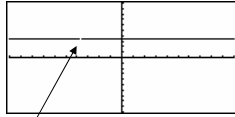
**PxlOff** 行数组, 列数组

显示Graph (图形) 屏幕并关闭像素坐标(行, 列)处的像素。

**注:** 重新作图会清除所有已画出的项目。

**PxlHorz 25,1** [ENTER]

**PxlOff 25,50** [ENTER]



25,50

## PxlOn CATALOG

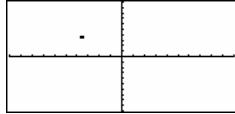
**PxlOn** 行, 列

**PxlOn** 行数组, 列数组

显示Graph (图形) 屏幕并开启像素坐标(行, 列)处的像素。

**注:** 重新作图会清除所有已画出的项目。

**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

若像素坐标(行, 列)处的像素处于开启状态, 则返回 true。若该像素处于关闭状态, 则返回 false。

**注:** 重新作图会清除所有已画出的项目。

**PxlOn 25,50** [ENTER]

 HOME


 [CALC HOME]

**PxlTest(25,50)** [ENTER]

true

**PxlOff 25,50** [ENTER]

 HOME

 [CALC HOME]

**PxlTest(25,50)** [ENTER]

false


## PxlText CATALOG


**PxlText** string, 行, 列

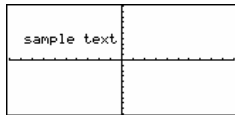
显示Graph (图形) 屏幕并将字符串 *string* 置于屏幕上, 起点为像素坐标(行, 列)。

*string* 的首字母的左上角处于上述坐标位置。

**注:** 重新作图会清除所有已画出的项目。

 **PxlText "sample text",20,10** [ENTER]

 **PxlText "sample text",20,50** [ENTER]





## PxlVert CATALOG

### PxlVert 列, drawMode]

显示Graph (图形) 屏幕并画出位于像素位置列的垂直线。

若  $drawMode = 1$ , 画出线段 (缺省情况)。

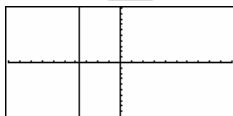
若  $drawMode = 0$ , 关闭线段。

若  $drawMode = -1$ , 将画出的线段关闭或将关闭的线段画出 (沿线段进行像素反转)。

**注:** 重新作图会清除所有已画出的项目。

参见LineVert。

### PxlVert 50,1 [ENTER]



## QR MATH/MatrixMatrix 菜单

### QR matrix, qMatName, rMatName, tol]

采用Householder法对实数或复数 *matrix* 进行QR分解。所求Q矩阵和R矩阵存储在指定的MatNames中。Q矩阵为酉矩阵。R矩阵为上三角阵。

作为可选项, 如果矩阵的任何元素的绝对值小于 *tol*, 它将按零值处理。仅当矩阵具有浮点输入项且不含任何未赋值的符号变量时使用此容许值。否则, *tol* 会被忽略。

- 若您使用  $\square$  [ENTER] 或设置模式为 Exact/Approx=APPROXIMATE, 则计算过程使用浮点算法。
- 若 *tol* 被忽略或未使用, 则缺省的容许值计算如下:

$$5E^{-14} * \max(\dim(matrix)) \\ * \text{rowNorm}(matrix)$$

QR分解采用Householder变换进行数值计算。采用Gram-Schmidt正交化方法进行符号运算。*qMatName* 中的列向量是 *matrix* 所定义的空间上的规范正交基。

m1中的浮点数字(9.) 使得结果以浮点形式进行计算。

### [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]

$$\begin{bmatrix} .123... & .904... & .408... \\ .492... & .301... & -.816... \\ .861... & -.301... & .408... \end{bmatrix}$$

### rm [ENTER]

$$\begin{bmatrix} 8.124... & 9.601... & 11.078... \\ 0. & .904... & 1.809... \\ 0. & 0. & 0. \end{bmatrix}$$

### [m,n;0,p]>m1 [ENTER]

$$\begin{bmatrix} m & n \\ 0 & p \end{bmatrix}$$

### 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]

进行二次多项式回归并更新系统统计变量。

除数组5外, 所有的数组必须具有相同的维数。

数组1代表x数组。

数组2代表y数组。

数组3代表频率。

数组4代表分类代码。

数组5代表分类所包含的数组。

**注:** 数组1到数组4必须有变量名或取名为c1-c99 (数据/矩阵编辑器中最末位的数据变量所代表的列)。数组5不一定要有变量名, 也可以不取名为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**

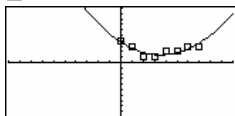
**Regeq(x)→y1(x)** **ENTER**

**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]

进行四次多项式回归并更新系统统计变量。

除数组5外, 所有的数组必须具有相同的维数。

数组1代表x数组。

数组2代表y数组。

数组3代表频率。

数组4代表分类代码。

数组5代表分类所包含的数组。

**注:** 数组1到数组4必须有变量名或取名为c1-c99 (数据/矩阵编辑器中最末位的数据变量所代表的列)。数组5不一定要有变量名, 也可以不取名为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

返回与(*x*,*y*)自变量对等效的θ-坐标值。

**注:** 根据当前的角度模式, 返回的结果可以是角度或弧度。

在Degree (角度) 模式下:

**R►Pθ**(*x*,*y*) **ENTER**

```
■ R►Pθ(x, y)
  90·sin(y) - tan⁴(x/y)
```

在Radian (弧度) 模式下:

**R►Pθ**(3,2) **ENTER**

**R►Pθ**([3,-4,2],[0,π/4,1.5]) **ENTER**

```
■ R►Pθ(3, 2)          tan⁴(2/3)
■ R►Pθ([3 -4 2],[0 π/4 1.5])
  [0 tan⁴(16/π) + π/2 .643501]
```

## **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

返回与(*x*,*y*)自变量对等效的r-坐标值。

在Radian (弧度) 模式下:

**R►Pr**(3,2) **ENTER**

**R►Pr**(*x*,*y*) **ENTER**

**R►Pr**([3,-4,2],[0,π/4,1.5]) **ENTER**

```
■ R►Pr(3, 2)          √13
■ R►Pr(x, y)          √x² + y²
■ R►Pr([3 -4 2],[0 π/4 1.5])
  [3 √(π² + 256)/4 2.5]
```

## **rand()** MATH/Probability 菜单

**rand**(*n*) ⇒ 表达式

*n* 为大于0的整数。

在无参数的情况下, 按次序返回0到1之间的下一个随机数。当自变量为正数时, 返回一个区间[1, *n*]内的随机整数。当自变量为负数时, 返回一个处于区间[-*n*, -1]内的随机整数。

**RandSeed** 1147 **ENTER**

Done

↑ (设置随机数种)

**rand**() **ENTER**

.158...

**rand**(6) **ENTER**

5

**rand**(-100) **ENTER**

-49

## **randMat()** MATH/Probability 菜单

**randMat**(*numRows*, *numColumns*) ⇒ 矩阵

返回一个指定维数, 其元素值在-9到9之间的整数矩阵。

该函数的2个自变量必须化简为整数。

**RandSeed** 1147 **ENTER**

Done

**randMat**(3,3) **ENTER**

```
[ 8 -3 6
 -2 3 -6
 0 4 -6]
```

**注:** 在您每次按下**ENTER**时, 该矩阵中的数值都会改变。

## **randNorm()** MATH/Probability 菜单

**randNorm**(*mean*, *sd*) ⇒ 表达式

从指定的正态分布中返回一个十进制小数。该数可以为任何实数, 但必须尽可能地落在区间[*mean*-3\**sd*, *mean*+3\**sd*]内。

**RandSeed** 1147 **ENTER**

Done

**randNorm**(0,1) **ENTER**

.492...

**randNorm**(3,4.5) **ENTER**

-3.543...

## randPoly() MATH/Probability 菜单

**randPoly**(*var*, *order*) ⇒ 表达式

返回关于变量 *var* 的指定阶数的多项式。其系数为处于 -9 到 9 范围内的随机整数。首项的系数不得为 0。

*order* 必须取 0-99。

**RandSeed** 1147

Done

**randPoly**(x,5)

$-2 \cdot x^5 + 3 \cdot x^4 - 6 \cdot x^3 + 4 \cdot x - 6$

## RandSeed MATH/Probability 菜单

**RandSeed** *number*

若 *number* = 0，将随机数种子设置为随机数发生器的出厂缺省值。若 *number* ≠ 0，则该函数被用来产生 2 个随机数种子，分别存储在系统变量 *seed1* 和 *seed2* 中。

**RandSeed** 1147

Done

**rand()**

.158...

## RclGDB CATALOG

**RclGDB** *GDBvar*

将所有设置存储在 Graph（图形）数据库的变量 *GDBvar* 中。

要查看设置列表，可参见 **StoGDB**。

**注：**在您恢复 *GDBvar* 之前，里面必须有储存信息。

**RclGDB** *GDBvar*

Done

## RclPic CATALOG

**RclPic** *picVar* [, *row*, *column*]

显示 Graph（图形）屏幕并使用 OR 逻辑将存放在 *picVar* 中的图形添加到左上角的像素坐标为 (*row*, *column*) 的位置。

*picVar* 必须为图形数据格式。

缺省情况的坐标为 (0, 0)。

## real() MATH/Complex 菜单

**real**(表达式 *n*) ⇒ 表达式

返回自变量的实部。

**注：**所有未定义的变量均作为实变量处理。参见 **imag()**。

**real**(2+3*j*)

2

**real**(z)

z

**real**(x+*j*)

x

**real**(数组 *n*) ⇒ 数组

返回所有元素的实部。

**real**({a+*j* b,3,*j*)

{a 3 0}

**real**(矩阵 *n*) ⇒ 矩阵

返回所有元素的实部。

**real**({a+*j* b,3;c,*j*)

$\begin{bmatrix} a & 3 \\ c & 0 \end{bmatrix}$

## ►Rect MATH/Matrix/向量 ops 菜单

向量 ►**Rect**

以直角坐标 [x, y, z] 的形式显示 **向量**。该向量必须为 2 维或 3 维，且可以为行向量或列向量。

**注：**►**Rect** 为显示格式指令，而不是转换指令。您只能在输入行的末尾使用该指令，但它并不会更新 *ans*。

**注：**可参见 ►**Polar**。

[3,∠π/4,∠π/6]►**Rect**

$\begin{bmatrix} 3 \cdot \frac{\sqrt{2}}{4} & 3 \cdot \frac{\sqrt{2}}{4} & \frac{3 \cdot \sqrt{3}}{2} \end{bmatrix}$

[a,∠b,∠c]

$\begin{bmatrix} a \cdot \cos(b) \cdot \sin(c) \\ a \cdot \sin(b) \cdot \sin(c) \\ a \cdot \cos(c) \end{bmatrix}$

## 复数值 $\blacktriangleright$ Rect

以直角坐标形式  $a+bi$  显示的复数值 (complex Value)。该复数值可以为任何复数形式。然而, 形式为  $re^{i\theta}$  的条目会在 Degree (角度) 模式下产生出错信息。

**注:** 您必须对形式为  $(r\angle\theta)$  的极坐标条目使用括号。

在 Radian (弧度) 模式下:

$$4e^{i(\pi/3)} \blacktriangleright \text{Rect} \text{ [ENTER]} \quad 4 \cdot e_3^{\pi}$$

$$(4\angle\pi/3) \blacktriangleright \text{Rect} \text{ [ENTER]} \quad 2+2 \cdot \sqrt{3} \cdot i$$

在 Degree (角度) 模式下:

$$(4\angle 60) \blacktriangleright \text{Rect} \text{ [ENTER]} \quad 2+2 \cdot \sqrt{3} \cdot i$$

**注:** 欲从键盘键入  $\blacktriangleright$  Rect, 可按下  $[\text{2nd}] \blacktriangleright$  获得  $\blacktriangleright$  算子。欲键入  $\angle$ , 可按下  $[\text{2nd}] [\angle]$ 。

## ref() MATH/Matrix 菜单

ref(矩阵  $n$ , tol)  $\Rightarrow$  矩阵

返回矩阵  $n$  的行梯矩阵。

作为可选项, 如果矩阵的任何元素的绝对值小于  $tol$ , 它将按零值处理。仅当矩阵含有浮点输入项且不含任何未赋值的符号变量时使用此容许值。否则,  $tol$  会被忽略。

- 若您使用  $\square$  [ENTER] 或设置模式为 Exact/Approx=APPROXIMATE, 则计算过程使用浮点算法。

- 若  $tol$  被省略或未使用, 则缺省的容许值计算如下:

$$5E^{-14} * \max(\dim(\text{矩阵 } n)) * \text{rowNorm}(\text{矩阵 } n)$$

**注:** 参见 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]  $\blacktriangleright$  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)  $\Rightarrow$  表达式

remain(数组1, 数组2)  $\Rightarrow$  数组

remain(矩阵1, 矩阵2)  $\Rightarrow$  矩阵

根据如下恒等式所定义, 返回第一自变量关于第二自变量的余数:

$$\text{remain}(x,0) = x$$

$$\text{remain}(x,y) = x - y \cdot \text{iPart}(x/y)$$

作为结果, 注意  $\text{remain}(-x,y) = -\text{remain}(x,y)$ 。结果要不为0, 要不与第1自变量具有相同的正负号。

**注:** 参见 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]  $\begin{bmatrix} 3 & 0 & 1 \end{bmatrix}$

remain([9, -7; 6, 4], [4, 3; 4, -3]) [ENTER]

$$\begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}$$

## Rename CATALOG

Rename oldVarName, newVarName

将变量 oldVarName 重命名为 newVarName。

{1,2,3,4}  $\blacktriangleright$  L1 [ENTER] {1,2,3,4}

Rename L1, List1 [ENTER] Done

List1 [ENTER] {1,2,3,4}

## Request CATALOG

Request *promptString*, *var*

如果**Request** 位于 **Dialog...EndDialog** 结构内, 则创建一个输入框让用户键入数据。若它是一条单独存在的指令, 则创建一个对话框用于输入数据。在任何情况下, 如果变量*var*中包含了字符串, 它将作为缺省选项在输入框中以高亮方式显示。 *promptString* 必须小于等于20个字符。

此指令可以单独使用或作为对话结构的一部分使用。

Request "Enter Your Name",str1 [ENTER]



## Return CATALOG

Return [表达式]

返回作为函数结果的表达式。在 **Func...EndFunc** 块或 **Prgm...EndPrgm** 块内使用。

**注:** 使用不带自变量的**Return** 指令退出程序。

**注:** 在主屏幕上以一长行的形式输入文本(无断行)。

```
Define factorial(nn)=Func
:local answer,count:1→ answer
:For count,1,nn
:answer* count→ answer:EndFor
:Returnanswer:EndFunc [ENTER] Done
factorial(3) [ENTER] 6
```

## right() MATH/List 菜单

right(数组*i*, *num*) ⇒ 数组

返回数组*i*中最右边*num*个元素。

若您省略 *num*, 则返回整个数组*i*。

right({1,3,-2,4},3) [ENTER] {3 -2 4}

right(sourceString*i*, *num*) ⇒ string

返回字符串 *sourceString* 最右边的*num*个字符。

若您省略 *num*, 则返回整个 *sourceString*。

right("Hello",2) [ENTER] "lo"

right(comparison) ⇒ 表达式

返回方程式或不等式的右侧。

right(x<3) [ENTER] 3

## rotate() MATH/Base 菜单

rotate(整数*i*, #ofRotations) ⇒ 整数

对一个二进制整数进行循环移位。您可以输入任何进制的整数*i*: 该整数将自动转换为一个带符号的32位二进制形式。若整数*i*的取值超出了二进制整数的表示范围, 对称求模运算将使该值落入正确的范围内。

若#of Rotations 为正数, 则向左进行循环移位; 若#of Rotations 为负数, 则向右进行循环移位。缺省情况下取值为 -1 (右移1位)。

例如, 在向右循环移位的情况下:

在二进制模式下:

```
rotate(0b1111010110000110101) [ENTER]
0b100000000000000111101011000011010
rotate(256,1) [ENTER] 0b1000000000
```

在十六进制模式下:

```
rotate(0h78E) [ENTER] 0h3C7
rotate(0h78E, -2) [ENTER] 0h800001E3
rotate(0h78E,2) [ENTER] 0h1E38
```

▶ 各数位向右旋转。  
 0b00000000000001111010110000110101  
 ↑  
 最右的数位旋转到最左。

结果为：

0b1000000000000111101011000011010

根据Base（基数）模式显示结果。

**重要提示：** 在输入二进制或十六进制的数字时，一定要使用前缀0b或0h（零，而不是字母0）。

<b>rotate(数组<i>n</i>,#ofRotations) ⇒ 数组</b> 返回向右或向左循环移位#of Rotations个元素后的数组 <i>n</i> 的副本。但不改变数组 <i>n</i> 。 若#of Rotations为正数，则向左进行循环移位；若#of Rotations为负数，则向右进行循环移位。缺省情况下取值为-1(右移1位)。	在十进制模式下： <b>rotate({1,2,3,4})</b> <input type="text" value="ENTER"/> {4 1 2 3} <b>rotate({1,2,3,4}, - 2)</b> <input type="text" value="ENTER"/> {3 4 1 2} <b>rotate({1,2,3,4},1)</b> <input type="text" value="ENTER"/> {2 3 4 1}
<b>rotate(字符串<i>n</i>,#ofRotations) ⇒ 字符串</b> 返回向右或向左循环移位#of Rotations个字符的字符串 <i>n</i> 的副本。但不改变字符串 <i>n</i> 。 若#of Rotations为正数，则向左进行循环移位；若#of Rotations为负数，则向右进行循环移位。缺省情况下取值为-1(右移1位)。	<b>rotate("abcd")</b> <input type="text" value="ENTER"/> "dabc" <b>rotate("abcd", - 2)</b> <input type="text" value="ENTER"/> "cdab" <b>rotate("abcd",1)</b> <input type="text" value="ENTER"/> "bcda"

## round() MATH/Number 菜单

<b>round(表达式<i>n</i>, digits) ⇒ 表达式</b> 按四舍五入返回小数点后保留指定位数的自变量的值。 <i>digits</i> 必须为范围在 0–12 的整数。若不含 <i>digits</i> ，返回圆整为 12 位有效数字的自变量。 <b>注：</b> 数字的显示模式可能对显示结果有影响。	<b>round(1.234567,3)</b> <input type="text" value="ENTER"/> 1.235
<b>round(数组<i>n</i>, digits) ⇒ 数组</b> 返回一个数组，其元素按四舍五入保留小数点后指定位数。	<b>round({<math>\pi</math>,<math>\sqrt{(2)}</math>,ln(2)},4)</b> <input type="text" value="ENTER"/> {3.1416 1.4142 .6931}
<b>round(矩阵<i>n</i>, digits) ⇒ 矩阵</b> 返回一个矩阵，其元素按四舍五入保留小数点后指定位数。	<b>round( ln(5),ln(3);<math>\pi</math>,<math>e^e(1)</math> ,1)</b> <input type="text" value="ENTER"/> $\begin{bmatrix} 1.6 & 1.1 \\ 3.1 & 2.7 \end{bmatrix}$

## rowAdd() MATH/Matrix/Row ops 菜单

<b>rowAdd(矩阵1, rIndex1, rIndex2) ⇒ 矩阵</b> 返回矩阵1的一个副本，其中第rIndex2行被第rIndex1行和第rIndex2行的和所替代。	<b>rowAdd({3,4; - 3, - 2},1,2)</b> <input type="text" value="ENTER"/> $\begin{bmatrix} 3 & 4 \\ 0 & 2 \end{bmatrix}$ <b>rowAdd({a,b;c,d},1,2)</b> <input type="text" value="ENTER"/> $\begin{bmatrix} a & b \\ a+c & b+d \end{bmatrix}$
---	--

## rowDim() MATH/Matrix/Dimensions 菜单

<b>rowDim(矩阵) ⇒ 表达式</b> 返回矩阵的行数。 <b>注：</b> 参见colDim()。	<b> 1,2;3,4;5,6 </b> <input type="text" value="M1"/> <input type="text" value="ENTER"/> $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ <b>rowdim(M1)</b> <input type="text" value="ENTER"/> 3
--	---

## rowNorm() MATH/Matrix/Norms 菜单

rowNorm(矩阵) ⇒ 表达式

返回矩阵中各行元素的绝对值之和的最大值。

注：所有矩阵元素必须简化为数字。参见 colNorm()。

rowNorm([-5,6,-7;3,4,9;9,-9,-7])

25

## rowSwap() MATH/Matrix/Row ops 菜单

rowSwap(矩阵1, rIndex1, rIndex2) ⇒ 矩阵

返回矩阵1，且将其第rIndex1行和第rIndex2行相互交换。

[1,2;3,4;5,6] → Mat

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

rowSwap(Mat,1,3)

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

## RpicPic CATALOG

RpicPic picVar[, row][, column]

清空Graph（图形）屏幕并在像素坐标(row, column)处放置图片picVar。若您不想清空屏幕，可使用RclPic。

picVar必须为图形数据类型的变量。若包含row和column，则指定了图形左上角的像素坐标。缺省情况下该坐标为(0, 0)。

注：对于不满全屏的图片，该函数只清空受新图片影响的区域。

## rref() MATH/Matrix 菜单

rref(矩阵r, tol) ⇒ 矩阵

返回矩阵r的简化行梯形式。

rref([-2, -2, 0, -6; 1, -1, 9, -9;

-5, 2, 4, -4])

$$\begin{bmatrix} 1 & 0 & 0 & 66/71 \\ 0 & 1 & 0 & 147/71 \\ 0 & 0 & 1 & -62/71 \end{bmatrix}$$

作为可选项，如果矩阵的任何元素的绝对值小于tol，它将按零值处理。仅当矩阵具有浮点输入项且不含任何未赋值的符号变量时使用此容许值。否则，tol会被忽略。

- 若您使用   或设置模式为 Exact/Approx=APPROXIMATE，则计算过程使用浮点算法。

- 若tol被忽略或未使用，则缺省的容许值计算如下：

$5e^{-14} * \max(\dim(\text{矩阵})) * \text{rowNorm}(\text{矩阵})$

注：参见 rref()。

rref([a,b,x;c,d,y])

$$\begin{bmatrix} 1 & 0 & \frac{d \cdot x - b \cdot y}{a \cdot d - b \cdot c} \\ 0 & 1 & \frac{-(c \cdot x - a \cdot y)}{a \cdot d - b \cdot c} \end{bmatrix}$$

## sec() MATH/Trig 菜单

sec(表达式) ⇒ 表达式

sec(数组) ⇒ 数组

返回表达式1的正切值，或返回一个数组，其元素为数组1中所对应元素的正切值。

注：根据当前的角度模式，自变量可以是角度或弧度模式。

在Degree（角度）模式下：

sec(45)

$\sqrt{2}$

sec({1,2,3,4})

$$\frac{1}{\cos(1)} \quad 1.000\dots \quad \frac{1}{\cos(4)}$$



**sec<sup>-1</sup>()** MATH/Trig 菜单

**sec<sup>-1</sup>(表达式)** ⇒ 表达式  
**sec<sup>-1</sup>(数组)** ⇒ 数组

在Degree (角度) 模式下:  
 $\sec^{-1}(1)$  [ENTER] 0

在Radian (弧度) 模式下:  
 $\sec^{-1}(\{1,2,5\})$  [ENTER]  
 $0 \frac{\pi}{3} \cos^{-1}(1/5)$

返回正切值为表达式1的角度值, 或返回一个数组, 其元素为数组1中所对应元素的反正切值。  
**注:** 根据当前的角度模式, 自变量可以是角度或弧度模式。

**sech()** MATH/Hyperbolic 菜单

**sech(表达式)** ⇒ 表达式  
**sech(数组)** ⇒ 数组

$\operatorname{sech}(3)$  [ENTER]  $\frac{1}{\cosh(3)}$

$\operatorname{sech}(\{1,2,3,4\})$  [ENTER]  
 $\frac{1}{\cosh(1)} .198\dots \frac{1}{\cosh(4)}$

返回表达式1的双曲正切值, 或返回一个数组, 其元素为数组1中所对应元素的双曲正切值。

**sech<sup>-1</sup>()** MATH/Hyperbolic 菜单

**sech<sup>-1</sup>(表达式)** ⇒ 表达式  
**sech<sup>-1</sup>(数组)** ⇒ 数组

在弧-角和直角坐标形式的复数模式下:  
 $\operatorname{sech}^{-1}(1)$  [ENTER] 0  
 $\operatorname{sech}^{-1}(\{1,-2,2,1\})$  [ENTER]  
 $0 \left(\frac{2\sqrt{3}}{3}\right) i \ 1.074\dots i$

返回表达式1的反双曲正切值, 或返回一个数组, 其元素为数组1中所对应元素的反双曲正切值。

**Send** CATALOG

**Send** 数组

程序段:  
 CBL 2™/CBL™ (Calculator-Based Laboratory™)(基于计算器的实验室™) 或  
 CBR™ (Calculator-Based Ranger™)(基于计算器的测距仪™)指令。发送数组到连接端口。  
 ⋮  
 :Send {1,0}  
 :Send {1,2,1}  
 ⋮


**SendCalc** CATALOG

**SendCalc** var

程序段:  
 ⋮  
 :a+b>x  
 :SendCalc x  
 ⋮

发送变量 var到连接端口, 在此连接到发送端口上的另一个端口可接受到变量值。接受设备必须显示在主屏幕上或必须从程序中执行**GetCalc**。

在您从 TI-89、TI-92 Plus、或 Voyage™ 200 向 TI-92发送数据时, 如果TI-92 从程序中执行 **GetCalc**会产生错误。此时, 发送设备必须改用**SendCalc**。

 **SendCalc** var[,port]

将 var的内容从TI-89 Titanium 发送到另一台TI-89 Titanium。

若没有指定端口, 或指定 port = 0, 如果USB端口处于连接状态, 则TI-89 Titanium将使用其传送数据, 否则将使用I/O端口传送。

若port = 1, 则TI-89 Titanium 仅使用USB 端口传送数据。

若port = 2, 则TI-89 Titanium 仅使用I/O端口传送数据。

## SendChat CATALOG

**SendChat** *var*

改指令通常是 **SendCalc** 的一种替代形式，在接受设备是 TI-92 是十分有用(或应用于允许使用 TI-92、Voyage™ 200、或 TI-92 Plus 的通用“会话”程序)。参见 **SendCalc** 可了解更多信息。

**SendChat** 仅仅发送与 TI-92 兼容的变量，在“会话”程序中往往如此。然而，**SendChat** 不会发送存档变量、或是 TI-89 的图形数据库等。

程序段：

```
:  
:  
:a+b>x  
:SendChat x  
:
```

## seq() MATH/List 菜单

**seq**(*表达式, var, low, high, step*) ⇒ *数组*

以 *step* 为步长，从 *low* 到 *high* 增加 *var* 的值，计算 *表达式* 的值并以数组形式返回结果。*var* 的初始内容在 **seq()** 执行完毕后并不改变。

*var* 不能为系统变量。

缺省情况下，*step* = 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]** 得到： 1.549...

## setDate() CATALOG

**setDate**(*年,月,日*) ⇒ *原数组*

将时钟设置为自变量中给出的数据，并返回一个数组。(注：其中 *年* 的范围必须为 1997 - 2132。)返回的数组格式为 {*原年,原月,原日*} 的格式。返回的数据为原来的时钟数值。

输入形式为4位整数的年份值。月份和日期可以为1位或2位整数。

**setDate**(2001,10,31) **[ENTER]** {2001 11 1}

## setDtFmt() CATALOG

**setDtFmt**(*整数*) ⇒ *原整数*

根据自变量设置桌面数据格式，并返回原来的日期格式数值。

整数值：

1 = MM/DD/YY 5 = YY.MM.DD  
2 = DD/MM/YY 6 = MM-DD-YY  
3 = MM.DD.YY 7 = DD-MM-YY  
4 = DD.MM.YY 8 = YY-MM-DD

## setFold() CATALOG

**setFold**(*newFolderName*) ⇒ *oldFolderString*

以字符串形式返回当前文件夹的名称，并设置 *newFolderName*(*新文件夹名*) 为当前文件夹。

文件夹 *新文件夹名* 必须存在。

**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**(*modeNameString, settingString*) ⇒ *string*

**setGraph**("Graph Order", "Seq") **[ENTER]** "SEQ"

将Graph模式 *modeNameString* (模式名字符串) 设置为 *settingString*, 并返回原来的模式设置。保存原来的设置以便今后恢复使用。

*modeNameString* 是一个字符串, 它表示您希望对进行设置的模式的名称。它必须是下表中模式名称中的一个。

*settingString* 也是一个字符串, 它表示您想重新设定的新模式名称。它必须为下列特定模式设置中的一个。

`setGraph("Coordinates","Off")`  
`[ENTER]` "RECT"

**注:** 在输入模式名称时可以选择用大写字母和空格。

模式名称	设置
"Coordinates"	"Rect", "Polar", "Off"
"Graph Order"	"Seq", "Simul" <sup>1</sup>
"Grid"	"Off", "On" <sup>2</sup>
"Axes"	"Off", "On" (非三维作图模式) "Off", "Axes", "Box" (三维作图模式)
"Leading Cursor"	"Off", "On" <sup>2</sup>
"Labels"	"Off", "On"
"Style"	"Wire Frame", "Hidden Surface", "Contour Levels", "Wire and Contour", "Implicit Plot" <sup>3</sup>
"Seq Axes"	"Time", "Web", "U1-vs-U2" <sup>4</sup>
"DE Axes"	"Time", "t-vs-y'", "y-vs-y'", "y1-vs-y2", "y1-vs-y2'", "y1'-vs-y2'" <sup>5</sup> <b>提示:</b> 欲输入素数符号 ('), 可按下 <code>[2nd][']</code> 。
"Solution Method"	"RK", "Euler" <sup>5</sup>
"Fields"	"SlpFld", "DirFld", "FldOff" <sup>5</sup>

<sup>1</sup>在序列、三维、或微分方程作图模式中不可用。

<sup>2</sup>在三维作图模式中不可用。

<sup>3</sup>仅在三维作图模式中可用。

<sup>4</sup>仅在序列作图模式中可用。

<sup>5</sup>仅在微分方程作图模式中可用。

## setMode() CATALOG

**setMode(modeNameString, settingString)** ⇒ string  
**setMode(list)** ⇒ stringList

将模式 *modeNameString* 设置为新值 *settingString*，并返回该模式的当前设置。

*modeNameString* 是一个字符串，它表示您希望对进行设置的模式的名称。它必须是下表中模式名称中的一个。

*settingString* 也是一个字符串，它表示您想重新设定的新模式名称。它必须为下列特定模式设置中的一个。

数组包含关键字的字符串对，并对其进行立即设置。推荐在多模式设置时使用该功能。若每个字符串对在输入时均按显示的次序单独使用 **setMode()**，则给出的示例可能无法运行。

使用 **setMode(var)** 可以将设置恢复到存放在 **getMode("ALL")** 返回的 *var* 中的值。

**注：**欲设置或返回与 Unit System 模式有关的信息，可使用 **setUnits()** 或 **getUnits()** 而不是 **setMode()** 或 **getMode()**。

**setMode("Angle","Degree")**  
 $\boxed{\text{ENTER}}$  "RADIAN"

**sin(45)**  $\boxed{\text{ENTER}}$   $\frac{\sqrt{2}}{2}$

**setMode("Angle","Radian")**  
 $\boxed{\text{ENTER}}$  "DEGREE"

**sin( $\pi/4$ )**  $\boxed{\text{ENTER}}$   $\frac{\sqrt{2}}{2}$

**setMode("Display Digits",  
 "Fix 2")**  $\boxed{\text{ENTER}}$  "FLOAT"

$\pi$   $\boxed{\text{ENTER}}$  3.14

**setMode("Display Digits",  
 "Float")**  $\boxed{\text{ENTER}}$  "FIX 2"

$\pi$   $\boxed{\text{ENTER}}$  3.141...

**setMode({"Split Screen",  
 "Left-Right","Split 1 App",  
 "Graph","Split 2 App","Table"})**  
 $\boxed{\text{ENTER}}$   
 {"Split 2 App" "Graph"  
 "Split 1 App" "Home"  
 "Split Screen" "FULL"}

**注：**在输入模式名称时可以选择用大写字母和空格。同时，这些示例的结果在您的计算器上可能有不同的显示结果。

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/矩阵 Editor", "Program Editor", "Text Editor", "Numeric Solver", "Flash App"
"Split 2 App"	"Home", "Y= Editor", "Window Editor", "Graph", "Table", "Data/矩阵 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" (Voyage™ 200 only)
"Exact/Approx"	"Auto", "Exact", "Approximate"
"Base"	"Dec", "Hex", "Bin"
"Language"	"English", "Alternate Language"
"Apps Desktop"	"Off", "On"

## setTable() CATALOG

**setTable(modeNameString, settingString) ⇒ string**

将表格参数 *modeNameString* 设置为 *settingString*, 并返回原来的参数设置。保存原来的设置以便今后恢复使用。

*modeNameString* 是一个字符串, 它表示您希望对进行设置的参数名称。它必须是下表中参数名称中的一个。

*settingString* 为指定新参数设置的字符串。它必须是下表中参数名称中的一个。

setTable("Graph <-> Table","ON")  
[ENTER] "OFF"

setTable("Independent","AUTO")  
[ENTER] "ASK"

• [TblSet]



**注:** 在输入参数名称时可以选择大写字母和空格。

参数名称	设置
"Graph <-> Table"	"Off", "On"
"Independent"	"Auto", "Ask"

## setTime() CATALOG

**setTime(时,分,秒) ⇒ 原数组**

setTime(11,32,50) {10 44 49}

将时钟设置为自变量中给出的数据, 并返回一个数组。返回的数组格式为 {原时,原分,原秒}。返回的数据是原来的时钟值。

以24小时制输入时间, 则 13点 = 下午1点。

## setTmFmt() CATALOG

**setTmFmt(整数) ⇒ 原整数**

按自变量设置桌面时间格式并返回原来的时间格式式。

整数值:

12 = 12小时制

24 = 24小时制

## setTmZn() CATALOG

**setTmZn(整数) ⇒ 原整数**

根据自变量设置桌面时区并返回原来的时区值。

返回的整数代表当前所设时区相对以格林尼治标准时间 (GMT) 的偏移分钟数。例如, 若该时区相对与GMT的偏移为2小时, 则计算器返回值为120 (分钟)。

代表GMT以西的时区的整数为负值。

代表GMT以东的时区的整数为正值。

若格林尼治标准时间为14:07:07, 则:

在科罗拉多州丹佛为8:07:07 a.m. (山地白昼时间)  
(相对GMT为-360分钟)

在比利时布鲁塞尔为16:07:07 p.m. (中欧标准时间)  
(相对GMT为+120分钟)

## setUnits() CATALOG

**setUnits(数组1) ⇒ 数组**

将缺省单位设置为数组1中指定的数值, 并返回组成为原来的缺省单位设置的数组。

- 要指定内置SI (公制) 或英/美制, 数组1可使用以下形式:

{"SI" 或 {"ENG/US"}

- 要指定自定义缺省单位, 数组1可使用以下形式:

{"CUSTOM", "cat1", "unit1" [, "cat2", "unit2", ...]}

其中各 *cat* 和 *unit* 对指定了类别和其缺省单位。(您仅可以指定内建单位, 而不能指定用户自定义单位。)任何未指定类别均采用原来的自定义单位。

- 要返回原来类别的自定义缺省单位, 数组1可采用以下形式:

{} "CUSTOM"




若您希望不同情况下采用不同缺省值, 可创建单独的数组并将这些值存入不同名称的数组中。要使用某个缺省值, 可在 **setUnits()** 中指定数组名称。

您可以使用 **setUnits()** 来恢复先前用 **setUnits()** → *var* 或 **getUnits()** → *var* 保存的设置。

所有的单位名称必须以下划线\_开始。

  [\_]  
  [2nd] [\_]

您也可以从菜单中选择单位, 只需按下:

 [2nd] [UNITS]  
  [UNITS]

```
setUnits({"SI"}) [ENTER]
{"SI" "Area" "NONE"
"Capacitance" " _F" ...}
```

```
setUnits({"CUSTOM","Length",
" _cm","Mass"," _gm"}) [ENTER]
{"SI" "Length" " _m"
"Mass" " _kg" ...}
```

**注:** 您的屏幕可能显示不同的单位。

## Shade CATALOG

**Shade** *expr1*, *expr2*, [*xlow*], [*xhigh*], [*pattern*], [*patRes*]

显示Graph屏幕，作出表达式*expr1*和*expr2*的图形并用阴影表示的*expr1*小于*expr2*的部分。( *expr1*和*expr2*必须为以*x*为自变量的表达式。)

若指令中含*xlow*和*xhigh*，则指定了阴影的左右边界。有效输入位于*xmin*和*xmax*间。缺省值即为*xmin*和*xmax*。

*pattern* 指定了下列4种阴影类型之一：

- 1 = 垂直 (缺省情况)
- 2 = 水平
- 3 = 负45度斜线
- 4 = 正45度斜线

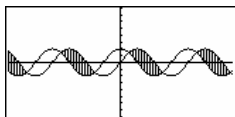
*patRes* 指定了阴影类型的解析度：

- 1 = 实心阴影
- 2 = 1像素间隔 (缺省情况)
- 3 = 2像素间隔
- ⋮
- 10 = 9像素间隔

**注：**在Graph屏幕上可以通过**Shade**指令得到交互式阴影。可以通过**Style**指令为具体函数自动添加阴影。**Shade**指令在三维作图模式下无效。

在ZoomTrig 视窗中：

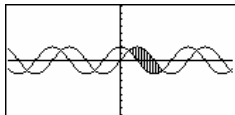
**Shade cos(x),sin(x)** [ENTER]



[HOME] [CALC HOME]

**ClrDraw** [ENTER] **Shade cos(x),sin(x),0,5** [ENTER]

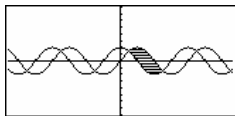
Done



[HOME] [CALC HOME]

**ClrDraw** [ENTER] **Shade cos(x),sin(x),0,5,2** [ENTER]

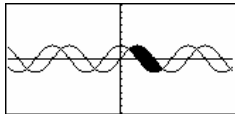
Done



[HOME] [CALC HOME]

**ClrDraw** [ENTER] **Shade cos(x),sin(x),0,5,2,1** [ENTER]

Done



## shift() CATALOG

**shift**(*整数*n**,*#ofShifts*) ⇒ *整数*

对一个二进制整数进行移位。您可以输入任何进制的*整数*n**；该整数将自动转换为一个带符号的32位二进制形式。若*整数*n**的取值超出了二进制整数的表示范围，对称求模运算将使该值落入正确的范围内。

若*#ofShifts*为正数，向左移位。若*#ofShifts*为负数，则向右移位。缺省值为-1(右移1位)。

在向右移位中，最右边的数位出局，同时在最左边的数位上插入0或1。在向左移位中，最左边的数位出局，同时在最右边的数位上插入0。

例如，在向右移位中：

在二进制模式下：

**shift(0b1111010110000110101)** [ENTER] **0b111101011000011010**

**shift(256,1)** [ENTER] **0b100000000**

在十六制模式下：

**shift(0h78E)** [ENTER] **0h3C7**

**shift(0h78E,-2)** [ENTER] **0h1E3**

**shift(0h78E,2)** [ENTER] **0h1E38**

**重要提示：**在输入二进制或十六进制的数字时，一定要使用前缀0b或0h(零，而不是字母O)。

↑ 各数位向右移。  
 0b00000000000001111010110000110101  
 ↑ 若最左数位为0则插入0, 若  
 最左数位为1则插入1。 ↓ 出局

结果为:

0b000000000000000111101011000011010

根据Base (数基) 模式显示结果。首尾的0不显示。

**shift(数组1[, #ofShifts]) ⇒ 数组**

返回将数组1向右或向左移位#ofShifts个元素后的一个副本。但并不改变数组1。

若#ofShifts为正数, 向左移位。若#ofShifts为负数, 则向右移位。缺省值为-1(右移1个元素)。

通过移位引入到数组首位或末位的元素被设为符号"undef"。

在十进制模式下:

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[, #ofShifts]) ⇒ string**

返回将字符串1向右或向左移位#ofShifts个元素后的一个副本。但并不改变字符串1。

若#ofShifts为正数, 向左移位。若#ofShifts为负数, 则向右移位。缺省值为-1(右移1个字符)。

通过移位引入到数组首位或末位的字符被设为空格。

shift("abcd") [ENTER] " abc"

shift("abcd", -2) [ENTER] " ab"

shift("abcd",1) [ENTER] "bcd "

**ShowStat CATALOG**

**ShowStat**

显示一个对话框, 对话框中包括前面已经得到的目前仍然有效的统计结果。若用以统计的数据已经改变, 则统计结果也会自动改变。

在统计运算(如LinReg)结束后使用该指令。

{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, 在表达式1≠0时返回表达式1/abs(表达式1)。

若表达式1为正则返回1。

若表达式1为负则返回-1。

若复数格式模式为REAL, 则sign(0) 返回

±1; 否则, 返回自身的值。

sign(0) 代表复数域中的单位圆。

对于数组或矩阵, 返回所有元素的符号。

sign(-3.2) [ENTER] -1.

sign({2,3,4, -5}) [ENTER] {1 1 1 -1}

sign(1+abs(x)) [ENTER] 1

若复数格式模式为REAL:

sign(|-3,0,3|) [ENTER] [-1 ±1 1]



## simult() MATH/Matrix 菜单

**simult(系数矩阵, 常数向量, tol) ⇒ 矩阵**

返回包含线性方程组的解的列向量。

系数矩阵必须为包含方程系数的方阵。

常数向量必须与系数矩阵具有相同的行数(相同的维数), 且包含常数项。

作为可选项, 如果矩阵的任何元素的绝对值小于 *tol*, 它将按零值处理。仅当矩阵具有浮点输入项且不含任何未赋值的符号变量时使用此容许值。否则, *tol* 会被忽略。

- 若您使用  $\square$  [ENTER] 或设置模式为 Exact/Approx=APPROXIMATE, 则计算过程使用浮点算法。
- 若 *tol* 被忽略或未使用, 则缺省的容许值计算如下:

$$5E^{-14} * \max(\text{dim}(\text{系数矩阵})) \\ * \text{rowNorm}(\text{系数矩阵})$$

**simult(系数矩阵, 常数矩阵, tol) ⇒ 矩阵**

求解多个具有相同系数和不同常数项的线性方程组。

常数矩阵中的各列必须包含方程组的常数项。所得矩阵的各列包含相应方程组的解。

求解  $x$  和  $y$ :

$$\begin{cases} x + 2y = 1 \\ 3x + 4y = -1 \end{cases}$$

**simult([1,2;3,4],[1;-1]) [ENTER]**

$$\begin{bmatrix} -3 \\ 2 \end{bmatrix}$$

方程组的解为  $x = -3$  且  $y = 2$ 。

求解:

$$\begin{cases} ax + by = 1 \\ cx + dy = 2 \end{cases}$$

**[a,b;c,d]→matx1 [ENTER]**

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

**simult(matx1,[1;2]) [ENTER]**

$$\begin{bmatrix} -(2 \cdot b - d) \\ a \cdot d - b \cdot c \\ \hline 2 \cdot a - c \\ a \cdot d - b \cdot c \end{bmatrix}$$

求解:

$$\begin{cases} x + 2y = 1 & x + 2y = 2 \\ 3x + 4y = -1 & 3x + 4y = -3 \end{cases}$$

**simult([1,2;3,4],[1,2;-1,-3])**

**[ENTER]**

$$\begin{bmatrix} -3 & -7 \\ 2 & 9/2 \end{bmatrix}$$

对于第一个方程组,  $x = -3$  且  $y = 2$ 。对于第二个方程组,  $x = -7$  且  $y = 9/2$ 。

## sin() $\square$ [2nd] [SIN] 键

$\square$  [SIN] 键

**sin(表达式  $\eta$ ) ⇒ 表达式**

**sin(数组  $\eta$ ) ⇒ 数组**

**sin(表达式  $\eta$ )** 以表达式形式返回自变量的正弦值。

**sin(数组  $\eta$ )** 返回一个数组, 其元素为数组  $\eta$  中对应元素的正弦值。

**注:** 根据当前的角度模式, 自变量可以是角度或弧度模式。您可以使用  $^\circ$  或  $^r$  临时改变角度模式设置。

在 Degree (角度) 模式下:

**sin( $(\pi/4)^r$ ) [ENTER]**

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

**sin(45) [ENTER]**

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

**sin({0.60,90}) [ENTER]**

$$\left\{ 0 \frac{\sqrt{3}}{2} \right\}$$

在 Radian (弧度) 模式下:

**sin( $\pi/4$ ) [ENTER]**

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

**sin(45 $^\circ$ ) [ENTER]**

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

**sin(方阵  $\eta$ ) ⇒ 方阵**

返回方阵  $\eta$  的矩阵正弦。该运算不同于对各元素求正弦的计算。欲了解与计算方法相关的信息, 可参见 **cos()**。

方阵  $\eta$  必须可对角线化。结果总包含有浮点数。

在 Radian (弧度) 角模式下:

**sin([1,5,3;4,2,1;6,-2,1]) [ENTER]**

$$\begin{bmatrix} .942... & -.045... & -.031... \\ -.045... & .949... & -.020... \\ -.048... & -.005... & .961... \end{bmatrix}$$

## $\sin^{-1}()$ [SIN<sup>-1</sup>] 键 [2nd] [SIN<sup>-1</sup>] 键

$\sin^{-1}$ (表达式 $\eta$ )  $\Rightarrow$  表达式

$\sin^{-1}$ (数组 $\eta$ )  $\Rightarrow$  数组

$\sin^{-1}$ (表达式 $\eta$ ) 以表达式形式返回一个角度值, 其正弦值为表达式 $\eta$ 。

$\sin^{-1}$ (数组 $\eta$ ) 返回一个数组, 其元素为数组 $\eta$ 中所对应元素的反正弦值。

**注:** 根据当前的角度模式设置, 返回的结果可以是角度或是弧度形式。

在Degree (角度) 模式下:

$\sin^{-1}(1)$  [ENTER] 90

在Radian (弧度) 模式下:

$\sin^{-1}(\{0, .2, .5\})$  [ENTER] {0 .201... .523...}

$\sin^{-1}$ (方阵 $\eta$ )  $\Rightarrow$  方阵

返回方阵 $\eta$ 的矩阵反正弦。该运算不同于对各元素求反正弦的计算。欲了解与计算方法相关的信息, 可参见 $\cos()$ 。

方阵 $\eta$  必须可对角线化。结果中总包含有浮点数。

在Radian (弧度) 模式和Rectangular (直角) 复数格式模式下:

$\sin^{-1}(\{1, 5, 3; 4, 2, 1; 6, -2, 1\})$  [ENTER]

$\begin{bmatrix} -.164...-.064... \cdot i & 1.490...-.2105... \cdot i & ... \\ .725...-1.515... \cdot i & .947...-.778... \cdot i & ... \\ 2.083...-.2632... \cdot i & -1.790...+1.271... \cdot i & ... \end{bmatrix}$

## $\sinh()$ MATH/Hyperbolic 菜单

$\sinh$ (表达式 $\eta$ )  $\Rightarrow$  表达式

$\sinh$ (数组 $\eta$ )  $\Rightarrow$  数组

$\sinh$ (表达式 $\eta$ ) 以表达式形式返回自变量的双曲正弦值。

$\sinh$ (数组 $\eta$ ) 返回一个数组, 其元素为数组 $\eta$ 中所对应元素的双曲正弦值。

$\sinh(1.2)$  [ENTER] 1.509...

$\sinh(\{0, 1, 2, 3, \})$  [ENTER] {0 1.509... 10.017...}

$\sinh$ (方阵 $\eta$ )  $\Rightarrow$  方阵

返回方阵 $\eta$ 的矩阵双曲正弦。该运算不同于对各元素求双曲正弦的计算。欲了解与计算方法相关的信息, 可参见 $\cos()$ 。

方阵 $\eta$  必须可对角线化。结果中总包含有浮点数。

在Radian (弧度) 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 $\eta$ )  $\Rightarrow$  表达式

$\sinh^{-1}$ (数组 $\eta$ )  $\Rightarrow$  数组

$\sinh^{-1}$ (表达式 $\eta$ ) 以表达式形式返回自变量的反双曲正弦值。

$\sinh^{-1}$ (数组 $\eta$ ) 返回一个数组, 其元素为数组 $\eta$ 中所对应元素的反双曲正弦值。

$\sinh^{-1}(0)$  [ENTER] 0

$\sinh^{-1}(\{0, 2, 1, 3\})$  [ENTER] {0 1.487...  $\sinh^{-1}(3)$ }

$\sinh^{-1}$ (方阵 $\eta$ )  $\Rightarrow$  方阵

返回方阵 $\eta$ 的矩阵反双曲正弦。该运算不同于对各元素求反双曲正弦的计算。欲了解与计算方法相关的信息, 可参见 $\cos()$ 。

方阵 $\eta$  必须可对角线化。结果中总包含有浮点数。

在Radian (弧度) 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 [, [iterations], [period], 数组3, 数组4]

计算正弦回归并更新所有的统计数据。

除数组4外, 所有数组必须具有相同维数。

数组1代表x数组。

数组2代表y数组。

数组3代表分类代码。

数组4代表分类所包含的数组。

*iterations* 指定了求解的最大尝试次数(1 到 16)。若省略, 则通常尝试8次。一般而言, 该数值越大, 则结果的精度越高, 但也需要运行更长时间, 反之亦然。

*period* 指定了预计的周期。若省略, 则数组1中各元素之间的差值应该相等且按顺序排列。若指定了 *period*, x数组元素之间的差值可以不等。

**注:** 数组1 到 数组3 必须有变量名或取名为 c1-c99 (数据/矩阵编辑器中最末位的数据变量所代表的列)。数组4 不一定要有变量名, 也可以不取名为 c1-c99。

无论角度模式的设置如何, **SinReg** 的输出通常为弧度形式。

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]

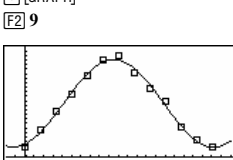


[ENTER] regeq(x) → y1(x) [ENTER] Done

NewPlot 1,1,L1,L2 [ENTER] Done

[G] [GRAPH]

[F2] 9



## solve() MATH/Algebra 菜单

solve(方程式, var) ⇒ 布尔表达式

solve(不等式, var) ⇒ 布尔表达式

返回方程或不等式中变量 *var* 的候选实数解。该函数的功能是产生所有候选解。然而, 有些方程或不等式可能会具有无穷多个解。

对于未定义变量的某些赋值组合, 候选解可能不是有限实数解。

对于 Exact/Approx 模式的 AUTO (自动) 设置, 其目的是当存在简明解时求得其精确解, 以及当精确解不存在时以近似迭代算法搜索其增补解。

由于最大公约数会从分子和分母中自动消去, 则解可能仅仅出现在单侧或两侧的界限处。

对于类型为  $\geq$ ,  $\leq$ ,  $<$ , 或  $>$  的不等式, 只有在不等式为线性且仅含变量 *var* 时才会有显解。

对于 Exact/Approx 模式的 EXACT 设置, 无法求解的部分将以隐式方程或不等式的形式返回。

使用 “|” 算子来约束解的区间和/或其它出现在方程或不等式中的变量。当您在单个区间内求解时, 您可以使用不等式算子将该区间从后续的搜索中排除。

solve(a\*x^2+b\*x+c=0,x) [ENTER]

$$x = \frac{\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a}$$

$$\text{或 } x = \frac{-\sqrt{b^2 - 4 \cdot a \cdot c} - b}{2 \cdot a}$$

ans(1) a=1 and b=1 and c=1

[ENTER]

Error: Non-real result

solve((x-a)e^x(x)=^-x\*(x-a),x) [ENTER]

x = a 或 x = -.567...

(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

solve(5x-2 ≥ 2x,x) [ENTER]

x ≥ 2/3

exact(solve((x-a)e^x(x)=^-x\*(x-a),x)) [ENTER]

e^x + x = 0 或 x = a

在 Radian (弧度) 角度模式下:

solve(tan(x)=1/x,x)|x>0 and x<1 [ENTER] x=.86

在找不到实数解时返回false。而**solve()**可确定当变量var为某个有限实数时则可满足方程式或不等式时，返回true。

由于**solve()**通常是返回一个布尔运算结果，您可以使用“and”、“or”、“not”将由**solve()**得到的解相互组合或与其它布尔表达式组合。

解中可能包含一个唯一的形为@n<sub>j</sub>的未定义的新变量，其中j为处于区间1-255内的整数。上述变量可赋任意整数值。

在实数模式下，奇分母分数幂仅表示实数支。否则，如分数幂、对数、以及反三角函数等多分支表达式仅表示其主支。因此，**solve()**仅产生与实数支或是主支相对应的解。

**注：**参见**cSolve()**、**cZeros()**、**nSolve()**、和**zeros()**。

**solve(x=x+1,x)** [ENTER] false

**solve(x=x,x)** [ENTER] true

**2x - 1 ≤ 1 and solve(x^2 ≠ 9,x)** [ENTER]  
x ≤ 1 and x ≠ - 3

在Radian（弧度）角度模式下：

**solve(sin(x)=0,x)** [ENTER] x = @n1 · π

**solve(x^(1/3)=- 1,x)** [ENTER] x = - 1

**solve(√(x)=- 2,x)** [ENTER] false

**solve(- √(x)=- 2,x)** [ENTER] x = 4

**solve(方程1 and 方程2 [and ... ], {varOrGuess1, varOrGuess2[, ... ]})** ⇒ 布尔表达式

返回联立代数方程的候选复数解，其中各varOrGuess指定了您希望求解的变量。

作为可选项，您可以为变量指定一个初始估计值。每个varOrGuess的形式必须为：

变量

- 或 -

变量 = 实数 或 非实数

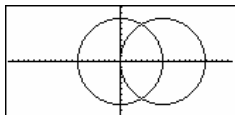
例如，x有效，则x=3。

**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

若所有方程均为多项式，且您未指定任何初始估计值，则**Solve()**使用Gröbner/Buchberger词法消元法来求得**全部**实数解。

例如，假设有一圆，其圆心在原点，半径为r；另一圆的半径也为r，圆心在第一个圆与正向x轴的交点处。使用**solve()**求两个圆的交点。



如右例中r所示，联立多项式方程可以包含无具体数值的其他变量，但可以在解中用给定数值进行替换。

$$\begin{aligned} &\text{solve}(x^2+y^2=r^2 \text{ and } \\ &(x-r)^2+y^2=r^2, \{x,y\}) \text{ [ENTER]} \\ &\quad x = \frac{r}{2} \text{ and } y = \frac{\sqrt{3} \cdot r}{2} \\ &\quad \text{or } x = \frac{r}{2} \text{ and } y = \frac{-\sqrt{3} \cdot r}{2} \end{aligned}$$

解中也可以包含未出现在方程式中的求解变量。例如，您可以将z作为求解变量将原例扩展变化为2个半径为r的平行相交圆柱。

这些圆柱解说明解系可能包含形式为@k的任意常数，此处k为1到255之间的整数下标。当您使用**ClrHome**或 $\boxed{\text{F1}}$  8:Clear Home时下标复位为1。

$$\begin{aligned} &\text{solve}(x^2+y^2=r^2 \text{ and } \\ &(x-r)^2+y^2=r^2, \{x,y,z\}) \text{ [ENTER]} \\ &\quad x = \frac{r}{2} \text{ and } y = \frac{\sqrt{3} \cdot r}{2} \text{ and } z = @1 \\ &\quad \text{or } x = \frac{r}{2} \text{ and } y = \frac{-\sqrt{3} \cdot r}{2} \text{ and } z = @1 \end{aligned}$$

对于多项式方程组，计算时间或内存占用很大程度上取决于求解变量的排列次序。若您的初始选择会占用过多内存或时间，您可以尝试重新排列方程中和或**varOrGuess**列表中变量的次序。

若您不用任何估计值，且所有的方程式不是任何变量的多项式，而所有方程都是求解变量的线性表达式，则**Solve()**使用高斯消元法来确定所有实数解。

$$\begin{aligned} &\text{solve}(x+e^z) \cdot y=1 \text{ and } x-y=\sin(z), \{x,y\}) \\ &\text{ [ENTER]} \\ &\quad x = \frac{e^z \cdot \sin(z)+1}{e^z+1} \text{ and } y = \frac{-(\sin(z)-1)}{e^z+1} \end{aligned}$$

若一个方程组既不是其任何变量的多项式，也不是求解变量的线性表达式，则通过近似叠代法**Solve()**最多只能求得一个解。为此，求解变量的数量必须等于方程的数量，且方程中的所有其它变量必须化简为数字形式。

$$\begin{aligned} &\text{solve}(e^z) \cdot y=1 \text{ and } y=\sin(z), \{y,z\}) \\ &\text{ [ENTER]} \\ &\quad y=.041... \text{ and } z=3.183... \end{aligned}$$

在带有估计值的情况下，各个求解变量从估计值开始搜索；否则，应从0.0开始。

使用估计值依次搜索其它解值。为了能够收敛，估计值应该尽可能的靠近解值。

$\text{solve}(e^z(z)*y=1 \text{ and } y=\sin(z), \{y,z=2\pi\})$   
 $\boxed{\text{ENTER}}$

$y=.001\dots$  and  $z=6.281\dots$

## 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] ...

将第一自变量的元素按升序排列。

如果您加入了其它自变量，那么这些自变量的元素也将跟随第一自变量重新排列，以保持与第一自变量元素的相对位置不变。

所有自变量必须是数组和向量。所有自变量必须有相同的维数。

{2,1,4,3} > List1  $\boxed{\text{ENTER}}$

{2,1,4,3}

SortA List 1  $\boxed{\text{ENTER}}$

Done

List 1  $\boxed{\text{ENTER}}$

{1 2 3 4}

{4,3,2,1} > List 2  $\boxed{\text{ENTER}}$

{4 3 2 1}

SortA List 2, List 1  $\boxed{\text{ENTER}}$

Done

List 2  $\boxed{\text{ENTER}}$

{1 2 3 4}

List 1  $\boxed{\text{ENTER}}$

{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] ...

与SortA类似，只是SortD以降序排列元素。

{2,1,4,3} > List 1  $\boxed{\text{ENTER}}$

{2 1 4 3}

{1,2,3,4} > List 2  $\boxed{\text{ENTER}}$

{1 2 3 4}

SortD List 1, List 2  $\boxed{\text{ENTER}}$

Done

List 1  $\boxed{\text{ENTER}}$

{4 3 2 1}

List 2  $\boxed{\text{ENTER}}$

{3 4 1 2}

## Sphere MATH/Matrix/Vector ops 菜单

vector **Sphere**

以球坐标形式 $[\rho \ \theta \ \phi]$ 显示行向量或列向量。

向量必须是一个三维向量，也可以为行向量或列向量。

**注：** **Sphere** 为显示格式指令，而不是转换指令。您只能在输入行的末尾使用它。

[1,2,3] **Sphere**

$\boxed{\text{ENTER}}$

[3.741...  $\angle$  1.107...  $\angle$  .640...]

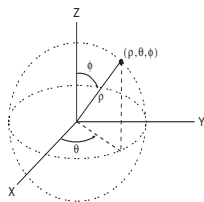
[2,  $\angle$   $\pi/4$ , 3] **Sphere**

$\boxed{\text{ENTER}}$

[3.605...  $\angle$  .785...  $\angle$  .588...]

$\boxed{\text{ENTER}}$

$[\sqrt{13} \ \angle \frac{\pi}{4} \ \angle \cos^{-1}(\frac{3 \cdot \sqrt{13}}{13})]$



## startTmr() CATALOG

**startTmr()**  $\Rightarrow$  整数

为计时器设置起始时间，以其整数表示返回当前的时钟值。您可以在**checkTmr()**中将起始时间作为自变量输入以确定已历经的秒数。

您可以同时运行多个计时器。

**注：** 参见 **checkTmr()** 和 **timeCnv()**。

**startTmr()**  $\boxed{\text{ENTER}}$

148083315

**checkTmr**(148083315)

34

**startTmr()**  $\Rightarrow$  Timer1

$\vdots$

**startTmr()**  $\Rightarrow$  Timer2

$\vdots$

**checkTmr**(Timer1)  $\Rightarrow$  Timer1 Value

$\vdots$

**checkTmr**(Timer2)  $\Rightarrow$  Timer2 Value

## stdDev() MATH/Statistics 菜单

**stdDev(数组, 频率数组)** ⇒ 表达式

返回数组中元素的标准差。

频率数组中的各元素为数组中对应元素出现的次数。

**注:** 数组必须包含至少2个元素。

**stdDev({a,b,c})** [ENTER]

**stdDev({1,2,5,-6,3,-2})** [ENTER]

$$\sqrt{\frac{a^2 - a(b+c) + b^2 - b \cdot c}{3}}$$
  
stdDev({1 2 5 -6 3 -2})  
2.7005544695

**stdDev({1,3,2,5,-6,4},{3,2,5})** [ENTER] 4.33345

**stdDev(矩阵I, 频率矩阵)** ⇒ 矩阵

返回由矩阵I中列的标准差所组成的行向量。

频率矩阵中的各元素为矩阵I中对应元素出现的次数。

**注:** 矩阵I必须包含至少2个元素。

**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 GDBvar**

创建图形数据库(GDB)变量，其内容为当前的：

- \* 作图模式
- \* Y= 函数
- \* 窗口变量
- \* 图形格式设定
  - 1图或2图的设置(在2图模式下为拆分屏幕和比例设定)
  - 角度模式
  - 实数/复数模式
- \* 序列模式或微分方程模式下的初始条件
- \* 表格标记
- \* tblStart, Δtbl, tblInput

您可以使用**RclGDB GDBvar**来存储图形环境。

**\*注:** 双图模式下两个图形均可使用这些项目。

## Stop CATALOG

**Stop**

程序执行停止指令。

程序段：

```
⋮  
For i,1,10,1  
  If i=5  
    Stop  
  EndFor  
⋮
```

## StoPic CATALOG

**StoPic** *picVar* [, *pxlRow*, *pxlCol*] [, *width*, *height*]

显示作图屏幕并将一个矩形显示区域复制给变量 *picVar*。

如果指令中带 *pxlRow* 和 *pxlCol*，则指定了复制区域的左上角 (缺省值为 0, 0)。

如果指令中带 *width* 和 *height*，则指定了该区域以像素为单位的大小。缺省情况下该值为当前图形屏幕以像素为单位的宽度和高度。

**Store** 参见第271页上的  $\rightarrow$  (store)。

## string() MATH/String 菜单

**string**(表达式)  $\Rightarrow$  *string*

简化表达式并以字符串形式返回结果。

**string(1.2345)** [ENTER] "1.2345"

**string(1+2)** [ENTER] "3"

**string(cos(x)+ $\sqrt{3}$ )** [ENTER] "cos(x) +  $\sqrt{3}$ "

## Style CATALOG

**Style** *equanum*, *stylePropertyString*

在当前图形模式下将系统作图函数 *equanum* 设置为使用作图属性 *stylePropertyString*。

*equanum* 必须为范围 1-99 内的整数，且该函数必须已经存在。

*stylePropertyString* 必须为以下之一: "Line"、"Dot"、"Square"、"Thick"、"Animate"、"Path"、"Above"、或 "Below"。

注意在参变量作图中，仅在参数对的 *x* 部分中包含有样式信息。

有效的样式信息及相应作图模式:

函数:	所有样式
参数/极坐标:	线、点、方块、加粗、动画、路径
序列:	线、点、方块、加粗
三维:	无
微分方程:	线、点、方块、加粗动画、路径

**注:** 在输入 *stylePropertyString* 的名称时可选用大写字母和空格。

**Style 1,"thick"** [ENTER] Done

**Style 10,"path"** [ENTER] Done

**注:** 在函数作图模式下，这些例子是将 *y1(x)* 的样式设置为 "Thick"，将 *y10(x)* 的样式为 "Path"。

## subMat() CATALOG

**subMat**(矩阵 *n*, *startRow*] [, *startCol*] [, *endRow*] [, *endCol*])  $\Rightarrow$  矩阵

返回矩阵 *n* 的指定子矩阵。

缺省值: *startRow*=1、*startCol*=1、*endRow*=末尾行、*endCol*=末尾列

**[1,2,3;4,5,6;7,8,9]**  $\rightarrow$  **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 (Sigma)** 参见第267页上的  $\Sigma()$ 。



## sum() MATH/List 菜单

sum(数组[, start[, end]]) ⇒ 表达式

返回数组中各元素的和。

Start 和 end 为可选项。它们指定了元素的范围。

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I[, start[, end]]) ⇒ 矩阵

返回由矩阵I各列的元素和组成的行向量。

Start 和 end 为可选项。它们指定了行的范围。

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I]) ⇒ 整数

返回活动窗口的数量。同时可设定活动窗口。

注：Window 1 为左窗口或上窗口；Window 2 为右窗口或下窗口。

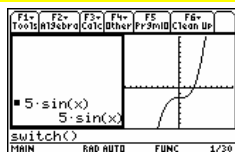
若整数I = 0，则返回活动窗口的数量。

若整数I = 1，激活window 1并返回此前的活动窗口数量。

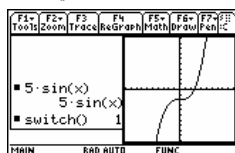
若整数I = 2，激活window 2并返回此前的活动窗口数量。

若整数I 被省略，则切换窗口并返回此前的活动窗口数量。

如果TI-89 Titanium/Voyage™ 200当前并没有显示子屏幕时，整数I 会被忽略。



switch() [ENTER]



## T (转置) MATH/Matrix 菜单

矩阵T ⇒ 矩阵

返回矩阵T的复共轭转置矩阵。

[1,2,3;4,5,6;7,8,9] > mat1 [ENTER]

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

mat1<sup>T</sup> [ENTER]

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

[a,b;c,d] > mat2 [ENTER]

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

mat2<sup>T</sup> [ENTER]

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

[1+i,2+i;3+i,4+i] > mat3 [ENTER]

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

mat3<sup>T</sup> [ENTER]

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

## Table CATALOG

**Table** 表达式 $f$ , 表达式 $l$  [,  $var$ ]

为指定表达式或函数建立表格。

表格中的表达式也可以作图。使用 **Table** 或 **Graph** 命令输入的表达式从1开始按升序进行函数编号。在按下  $\square$  Header 显示表格时, 可以使用所提供的编辑函数对表达式分别进行修改或删除。当前在 Y= Editor 中被选中的函数将暂时不予考虑。

为清除用 **Table** 或 **Graph** 创建的函数, 可执行 **ClrGraph** 命令或显示 Y= Editor。

如果省略了可选自变量  $var$ , 则假定使用当前图形模式的自变量。该指令有效的变化形式如下:

函数作图: **Table**  $expr, x$   
 参变量作图: **Table**  $xExpr, yExpr, t$   
 极坐标作图: **Table**  $expr, \theta$

**注:** **Table** 命令在三维、序列、或微分方程作图模式中无效。此时, 您可以使用 **BldData** 来代替它。

在函数作图模式下:

**Table**  $1.25x * \cos(x)$   $\square$

x	1		
0.	0.		
1.	.67538		
2.	-1.04		
3.	-3.712		
4.	-3.268		

**Table**  $\cos(\text{time}), \text{time}$   $\square$

x	1	2	3
0.	0.	1.	
1.	.67538	.5403	
2.	-1.04	-.4161	
3.	-3.712	-.99	
4.	-3.268	-.6536	

## tan()

**tan**(表达式 $f$ )  $\Rightarrow$  表达式

**tan**(数组 $l$ )  $\Rightarrow$  数组

**tan**(表达式 $f$ ) 以表达式形式返回自变量的正切值。

**tan**(数组 $l$ ) 返回一个数组, 其元素为数组 $l$ 中所对应元素的正切值。

**注:** 根据当前的角度模式, 自变量可以是角度或弧度模式。您可以使用  $^\circ$  或  $r$  临时改变角度模式设置。

在Degree (角度) 角度模式下:

**tan**(( $\pi/4$ ) $^\circ$ )  $\square$  1

**tan**(45)  $\square$  1

**tan**({0,60,90})  $\square$  {0  $\sqrt{3}$  undef}

在Radian (弧度) 角度模式下:

**tan**( $\pi/4$ )  $\square$  1

**tan**(45 $^\circ$ )  $\square$  1

**tan**({ $\pi, \pi/3, -\pi, \pi/4$ })  $\square$  {0  $\sqrt{3}$  0 1}

**tan**(方阵 $l$ )  $\Rightarrow$  方阵

返回方阵 $l$ 的矩阵正切。该运算不同于对各元素求正切的计算。欲了解与计算方法相关的信息, 可参见 **cos()**。

方阵 $l$  必须可对角线化。结果中总包含有浮点数。

在Radian (弧度) 角度模式下:

**tan**({1,5,3;4,2,1;6, -2,1})  $\square$

-28.291...	26.088...	11.114...
12.117...	-7.835...	-5.481...
36.818...	-32.806...	-10.459...

## tan<sup>-1</sup>()

**tan<sup>-1</sup>**(表达式 $f$ )  $\Rightarrow$  表达式

**tan<sup>-1</sup>**(数组 $l$ )  $\Rightarrow$  数组

**tan<sup>-1</sup>**(表达式 $f$ ) 以表达式形式返回一个角度值, 其正切值为表达式 $f$ 。

**tan<sup>-1</sup>**(数组 $l$ ) 返回一个数组, 其元素为数组 $l$ 中所对应元素的反正切值。

**注:** 根据当前的角度模式设置, 返回的结果可以是角度或是弧度形式。

在Degree (角度) 角度模式下:

**tan<sup>-1</sup>**(1)  $\square$  45

在Radian (弧度) 角度模式下:

**tan<sup>-1</sup>**({0,2,..5})  $\square$  {0 .197... .463...}

$\tan^{-1}$ (方阵*n*)  $\Rightarrow$  方阵

返回方阵*n*的矩阵反正切。该运算不同于对各元素求反正切值的计算。欲了解与计算方法相关的信息，可参见`cos()`。

方阵*n*必须可对角线化。结果中总包含有浮点数。

在Radian（弧度）角度模式下：

$\tan^{-1}(\{1,5,3;4,2,1;6,-2,1\})$  [ENTER]

```
[ -.083... 1.266... .622...  
  .748... .630... -.070...  
 1.686... -1.182... .455... ]
```

## **tanh()** MATH/Hyperbolic 菜单

$\tanh$ (表达式*n*)  $\Rightarrow$  表达式

$\tanh(1.2)$  [ENTER]

.833...

$\tanh$ (数组*n*)  $\Rightarrow$  数组

$\tanh(\{0,1\})$  [ENTER]

{0 tanh(1)}

$\tanh$ (表达式*n*) 以表达式形式返回自变量的双曲正切值。

$\tanh$ (数组*n*) 返回一个数组，其元素为数组*n*中所对应元素的双曲正切值。

$\tanh$ (方阵*n*)  $\Rightarrow$  方阵

在Radian（弧度）角度模式下：

返回方阵*n*的矩阵双曲正切。该运算不同于对各元素求双曲正切值的计算。欲了解与计算方法相关的信息，可参见`cos()`。

$\tanh(\{1,5,3;4,2,1;6,-2,1\})$  [ENTER]

```
[ -.097... .933... .425...  
  .488... .538... -.129...  
 1.282... -1.034... .428... ]
```

方阵*n*必须可对角线化。结果中总包含有浮点数。

## **tanh<sup>-1</sup>()** MATH/Hyperbolic 菜单

$\tanh^{-1}$ (表达式*n*)  $\Rightarrow$  表达式

在直角坐标形式的复数格式下：

$\tanh^{-1}$ (数组*n*)  $\Rightarrow$  数组

$\tanh^{-1}(0)$  [ENTER]

0

$\tanh^{-1}$ (表达式*n*) 以表达式形式返回自变量的反双曲正切值。

$\tanh^{-1}$ (数组*n*) 返回一个数组，其元素为数组*n*中所对应元素的反双曲正切值。

$\tanh^{-1}(\{1,2,1,3\})$  [ENTER]

```
{∞ .518... -1.570... · i  $\frac{\ln(2)}{2}$  -  $\frac{\pi}{2}$  · i }
```

$\tanh^{-1}$ (方阵*n*)  $\Rightarrow$  方阵

在Radian（弧度）角度模式和Rectangular（直角）负数格式模式下：

返回方阵*n*的矩阵反双曲正切。该运算不同于对各元素求反双曲正切值的计算。欲了解与计算方法相关的信息，可参见`cos()`。

$\tanh^{-1}(\{1,5,3;4,2,1;6,-2,1\})$  [ENTER]

```
[ -.099...+.164...·i 2.67...-1.490...·i ...  
 -.087...-.725...·i .479...-.947...·i ...  
 .511...-2.083...·i -.878...+1.790...·i ... ]
```

方阵*n*必须可对角线化。结果中总包含有浮点数。

## taylor() MATH/Calculus 菜单

taylor(表达式1, var, order, point) ⇒ 表达式

返回所需要的泰勒多项式。多项式包括了关于 (var-point) 从零到order的非零整数次幂项。如果在此阶上不存在截幂级数或需要负指数或分数指数, 则taylor()会返回它自己。使用代换法或临时乘上一个以 (var-point) 的幂项来确定更一般形式的幂级数。

point是展开点, 缺省值为零。

taylor( $e^{\sqrt{x}}$ , x, 2) [ENTER]

taylor( $e^t$ , t, 4) | t =  $\sqrt{x}$  [ENTER]

$$\begin{aligned} & \text{taylor}(e^{\sqrt{x}}, x, 2) \\ & \text{taylor}(e^t, t, 4) | t = \sqrt{x} \\ & \frac{x^2}{24} + \frac{x^{3/2}}{6} + \frac{x}{2} + \sqrt{x} + 1 \end{aligned}$$

taylor(1/(x\*(x-1)), x, 3) [ENTER]

$$\begin{aligned} & \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3\right) \\ & \text{taylor}\left(\frac{1}{x \cdot (x-1)}, x, 3, 0\right) \end{aligned}$$

expand(taylor(x/(x\*(x-1)),

x,4)/x,x) [ENTER]

$$\begin{aligned} & \text{expand}\left(\frac{\text{taylor}\left(\frac{x}{x \cdot (x-1)}, x, 4\right)}{x}, x\right) \\ & -x^3 - x^2 - x - \frac{1}{x} - 1 \end{aligned}$$

## tCollect() MATHAlgebraTrig 菜单

tCollect(表达式1) ⇒ 表达式

返回一个表达式, 其中正弦和余弦的乘积项和整数幂项均被转化为倍角与和差角的正弦或余弦的线形组合。该变换将三角函数多项式变换为其谐函数的线形组合。

有时在缺省的三角函数化简方法无能为力时, tCollect() 会完成任务。tCollect() 可能会对tExpand()的变换结果进行逆变换。有时对tCollect()的变换结果应用tExpand()变换, 这样通过2个步骤来化简表达式; 反之亦然。

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() MATHAlgebraTrig 菜单

tExpand(表达式1) ⇒ 表达式

返回一个表达式, 其中整数倍角与和差角的正弦和余弦的被展开。由于恒等式  $(\sin(x))^2 + (\cos(x))^2 = 1$ , 因此可能有多种形式的等价解。所以, 在不同出版物上给出的结果可能不同。

有时在缺省的三角函数化简方法无能为力时, tExpand() 会完成您的任务。tExpand() 可能会对tCollect()的变换结果进行逆变换。有时对tExpand()的变换结果应用tCollect()变换, 这样通过2个步骤来化简表达式; 反之亦然。

注: 分度为 $\pi/180$ 的角度模式不能很好地发挥tExpand()展开识别能力。为获得最佳结果, tExpand()应在Radian(弧度)模式下使用。

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

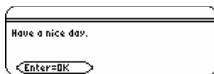
**Text** *promptString*

显示字符串 *promptString* 的对话框。

作为 **Dialog...EndDialog block** 的一部分使用时, *promptString* 显示在对话框中。若作为单条指令使用时, **Text** 将创建对话框来显示字符串。

**Text** "Have a nice day." [ENTER]

Done



**Then** 参见第193页上的**If**。

## timeCnv() CATALOG

**timeCnv**(秒数) ⇒ 数组

将秒数转化为可以更容易理解和计算的时间单位形式。数组形式应该为{日、时、分、秒}。

**注:** 可参见 **checkTmr()** 和 **startTmr()**。

**timeCnv**(152442117)

{1764 9 1 57}

## Title CATALOG

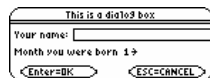
**Title** *titleString*, [*Lbl*]

当它用在 **Toolbar** 或 **Custom** 结构中或是 **Dialog...EndDialog** 程序段中时, 可创建下拉菜单或对话框的标题。

**注:** *Lbl* 仅在 **Toolbar** 结构中有效。在它出现时, 将使菜单选项分支到程序中的特定的标签处。

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
:EndDialog
:
```



## tmpCnv() CATALOG

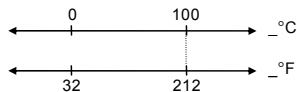
**tmpCnv**(表达式1\_°tempUnit1, \_°tempUnit2)  
⇒ 表达式\_°tempUnit2

将表达式1中指定的温度值从一种单位转化为另一种单位。有效的温度单位为:

°C 摄氏  
°F 华氏  
°K 开氏  
°R 兰金

└─对于 °, 可按下 [2nd] [°].  
└─对于 °, 可按下 [°] [-].  
└─对于 °, 可按下 [2nd] [-].

例如, 100\_°C转化为212\_°F:



若要转化一段温度范围, 可使用 **ΔtmpCnv()**。

**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

**注:** 欲从菜单中选择温度, 可按下:

☰ [2nd] [UNITS]  
☰ [°] [UNITS]

## $\Delta$ tmpCnv() CATALOG

$\Delta$ tmpCnv(表达式1\_°tempUnit1, \_°tempUnit2)  
⇒ 表达式\_°tempUnit2

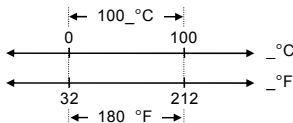
将表达式1指定的一段温度范围(2个温度值之间的差)从一种单位转化为另一种单位。有效的温度单位为:

—°C      摄氏  
—°F      华氏  
—°K      开氏  
—°R      兰金

对于°, 可按下  $\text{[2nd] [°]}$ 。  
对于\_, 可按下  $\text{[ ] [ ]}$ 。  
对于\_, 可按下  $\text{[2nd] [-]}$ 。

1 °C 和 1 °K 具有相同的取值范围, 而 1 °F 和 1 °R 也如此。然而, 1 °C 的范围是 1 °F 的 9/5 倍。

例如, 100 °C 表示的范围 (从 0 °C 到 100 °C) 等效于 180 °F 表示的范围:



若仅仅转化某一定点而不是一段范围的温度, 可使用 tmpCnv()。

欲获取  $\Delta$ , 您可以按下  $\text{[ ] [ ] [D]}$  (or  $\text{[2nd] [CHAR] 1 5}$ )。

$\Delta$ tmpCnv(100\_°c,\_°f)  $\text{[ENTER]}$       180.\_°F

$\Delta$ tmpCnv(180\_°f,\_°c)  $\text{[ENTER]}$       100.\_°C

$\Delta$ tmpCnv(100\_°c,\_°k)  $\text{[ENTER]}$       100.\_°K

$\Delta$ tmpCnv(100\_°f,\_°r)  $\text{[ENTER]}$       100.\_°R

$\Delta$ tmpCnv(1\_°c,\_°f)  $\text{[ENTER]}$       1.8\_°F

**注:** 欲从菜单中选择温度, 可按下:

$\text{[ ] [2nd] [UNITS]}$   
 $\text{[ ] [ ] [UNITS]}$

## Toolbar CATALOG

Toolbar  
block  
EndTBar

创建工具条菜单。

block 可以为单条语句或是用字符 ":" 分隔的一系列语句。这些语句可以是 Title (标题) 或 Item (项)。

Item 语句必须具有标签。而 Title 在不包含 item 的情况下也必须有标签。

程序段:

```
:  
:  
:Toolbar  
: Title "Examples"  
: Item "Trig", t  
: Item "Calc", c  
: Item "Stop", Pexit  
:EndTbar  
:
```

**注:** 当在程序中运行时, 该程序段将创建带有 3 个选项的菜单, 分别分支到程序中的 3 处位置。

## Trace CATALOG

Trace

画出 Smart Graph 并将显迹光标放在最先定义的  $Y = \text{function}$  处, 该函数位于此前定义的光标位置, 或是在需要重画时位于复位位置。

在编辑坐标数值时允许对光标和大多数键进行操作。某些键, 如功能键  $\text{[APPS]}$  和  $\text{[MODE]}$ , 在跟踪轨迹的过程中不起作用。

**注:** 按下  $\text{[ENTER]}$  以返回操作状态。

## Try CATALOG

```
Try
  block1
Else
  block2
EndTry
```

若无错误产生，则执行 *block1*。若在 *block1* 中产生错误，则程序转而执行 *block2*。变量的 *errornum*（出错编号）包含了允许程序进行错误恢复的出错编号。

*block1* 和 *block2* 可以为单条语句或是用字符“.”分隔的一系列语句。

程序段：

```
⋮
:Try
: NewFold(temp)
: Else
: ⓄAlready exists
: ClrErr
: EndTry
⋮
```

注：See **ClrErr** and **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** 的相关统计数据，并更新所有的系统统计变量。

除 *数组5* 外，所有的数组必须具有相同的维数。

*数组1* 代表 *x* 数组。

*数组2* 代表 *y* 数组。

*数组3* 代表频率。

*数组4* 代表分类代码。

*数组5* 代表分类所包含的数组。

注：*数组1* 到 *数组4* 必须有变量名或取名为 *c1-c99*（数据/矩阵编辑器中最末位的数据变量所代表的列）。*数组5* 不一定要有变量名，也可以不取名为 *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 var1[, var2[, var3] ...

将指定变量从用户数据文档内移动到 RAM 中。

您可以象访问 RAM 中的变量一样访问存档变量。然而，由于存档变量自动加锁，您将不能对它进行删除、重命名、或存储。

要将变量存档，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*l*) ⇒ 向量

根据 *向量l* 的格式返回单位向量或单位列向量。

*向量l* 必须为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]

$$\begin{bmatrix} \frac{\sqrt{14}}{14} \\ \frac{14}{7} \\ \frac{\sqrt{14}}{7} \\ \frac{3 \cdot \sqrt{14}}{14} \end{bmatrix}$$

## Unlock CATALOG

Unlock  $var1$ ,  $var2$  [,  $var3$ ]...

给指定变量解锁。

注：要给变量加锁可使用 **Lock** 命令。

## variance() MATH/Statistics 菜单

variance(数组 [, 频率数组])  $\Rightarrow$  表达式

返回数组的方差。

频率数组中的各元素为数组中对应元素出现的次数。

注：数组必须包含至少2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 [, 频率矩阵])  $\Rightarrow$  矩阵

返回由矩阵 $T$ 中各列的方差组成的行向量。

频率矩阵的各元素记录了矩阵 $T$ 中对应元素连续出现的次数。

注：矩阵 $T$ 必须包含至少2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(condition, trueResult [, falseResult] [, unknownResult])  $\Rightarrow$  表达式

根据 condition 取值是否为 true、false、或 unknown，返回 trueResult、falseResult、或 unknownResult。若自变量数量不足以得出合理的结果则返回输入值。

省略 falseResult 和 unknownResult 可获得仅定义在 condition 为 true 的区间内的表达式。

使用 undef falseResult 来定义仅在某个区间内作图的表达式。

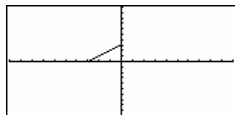
when(x<0,x+3)|x=5 [ENTER]

when(x<0,3+x)

ClrGraph [ENTER]

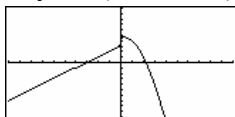
Graph when(x $\geq$  -  $\pi$  and x<0,x+3,undef)

[ENTER]



仅省略 unknownResult 可定义一个两段分段表达式。

Graph when(x<0,x+3,5- x^2) [ENTER]



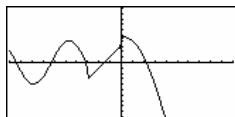


嵌套式地使用**when()**可定义多段分段表达式。



ClrGraph [ENTER] Done

Graph when(x<0,when(x<-π,  
4\* sin(x),2x+3),5- x^2) [ENTER]



**when()**在定义回归函数时非常有用。

when(n>0,n\*factorial(n-1),1)  
→ factorial(n) [ENTER]  
factorial(3) [ENTER]  
3! [ENTER]

Done  
6  
6

## While CATALOG

**While** *condition*  
*block*  
**EndWhile**

只要 *condition* 为真，执行 *block* 中的语句。

*block* 可以为单条语句或是用字符 ":" 分隔的一系列语句。

程序段：

```

:
:
:1→i
:0→temp
:While i<=20
: temp+i→temp
: i+1→i
:EndWhile
:Disp "sum of reciprocals up to 20",temp
:

```

## "With" 参见第270页上的。

## XOR MATH/Test 菜单

布尔表达式1 xor 布尔表达式2 ⇒ 布尔表达式

若布尔表达式1为true，布尔表达式2为false，则返回true，反之亦然。  
若布尔表达式1和布尔表达式2均为true或均为false，则返回false。如果两个原布尔表达式中的任何一个无法确定为true或false，则返回简化的布尔表达式。

注：参见 **or**。

true xor true [ENTER] false  
(5>3) xor (3>5) [ENTER] true

整数1 xor 整数2 ⇒ 整数

使用 **xor** 运算可逐位比较2个实整数。在计算器的内部运算中，2个整数均被转化为带符号的32位二进制数。当比较相应的数位时，若其中1个数位（不是同时2个）为1，结果为1；若其中2个数位同时为0或同时为1，则结果为0。返回的数值代表数位的结果，并根据Base（数基）模式显示出来。

您可以按任何数基输入整数。对于按2进制或16进制输入的整数，您必须分别使用0b或0h作为前缀。若不带前缀，任何整数均会被视为10进制数（以10为基数）。

若您输入的十进制整数超过了带符号的32位二进制数字的表示范围，对称求模运算将使该值落入正确的范围内。

注：参见 **or**。

在十六进制模式下：

0h7AC36 xor 0h3D5F [ENTER] 0h79169  
└ Important: 零，而不是字母O。

在二进制模式下：

0b100101 xor 0b100 [ENTER] 0b100001

注：输入的二进制数可以多达32位（不计前缀0b）。输入的十六进制数可以达8位。

**XorPic** *picVar* [, *row*] [, *column*]

在当前Graph（图形）屏幕显示存储在*picVar*中的图片。

对各像素使用 **xor** 逻辑。仅有那些不在屏幕或图片范围内的像素点会被打开。该指令将关闭那些在上述两个范围内都打开的像素点。

*picVar* 必须包含图形数据类型。

若包含 *row* and *column*，则指定了图片左上角的像素坐标。缺省值为(0, 0)。

**zeros()** MATH/Algebra 菜单**zeros**(表达式, var) ⇒ 数组

返回一个数组，其元素为使表达式=0的变量 *var* 的候选实数值。**zeros()** 通过计算 **exp▶List(solve(表达式=0, var), var)** 完成其运算。

在某些情况下，**zeros()**的结果形式比**solve()**的结果形式使用起来更为方便。然而，**zeros()**的结果形式不能表示隐式解、带不等式的解、或不涉及变量 *var* 的解。

**注：**可参见 **cSolve()**、**cZeros()**和**solve()**。

**zeros**({表达式1, 表达式2}, {varOrGuess1, varOrGuess2, ... }) ⇒ 矩阵

返回联立代数方程的候选实数零点，其中各 *varOrGuess* 都指定了一个未知数。

作为可选项，您可以为变量指定一个初始估计值。每个 *varOrGuess* 的形式必须为：

变量  
- 或 -  
变量 = 实数 或 虚数

例如，x 有效，且 x=3。

**zeros(a \* x^2 + b \* x + c, x)** [ENTER]

$$\left\{ \frac{-\sqrt{b^2 - 4 \cdot a \cdot c} + b}{2 \cdot a}, \frac{\sqrt{b^2 - 4 \cdot a \cdot c} + b}{2 \cdot a} \right\}$$

**a \* x^2 + b \* x + c | x = ans(1) | 2** [ENTER]

0

**exact(zeros(a \* (e^(x) + x) (sign(x) - 1), x))** [ENTER]

{}

**exact(solve(a \* (e^(x) + x) (sign(x) - 1) = 0, x))** [ENTER]

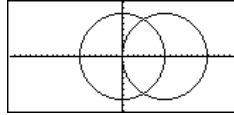
$e^x + x = 0$  or  $x > 0$  or  $a = 0$

若所有方程均为多项式，且您未指定任何初始估计值，则**zeros()** 使用 Gröbner/Buchberger 词法消元法来求得**全部**实数解。

例如，假设有一圆，其圆心在原点，半径为  $r$ ；另一圆的半径也为  $r$ ，圆心在第一个圆与正向  $x$  轴的交点处。使用 **solve()** 求两个圆的交点。

如右例中  $r$  所示，联立多项式方程可以包含无具体数值的其他变量，但在解中用给定数值进行替换。

所得矩阵的各行代表了不同的零值，其元素的排列次序与 `varOrGuess` 数组一致。要提取某一行，可按 `[row]` 对矩阵进行索引。



```
zeros({x^2+y^2- r^2,
(x- r)^2+y^2- r^2},{x,y}) [ENTER]
```

$$\begin{bmatrix} \frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} \\ \frac{r}{2} & -\frac{\sqrt{3} \cdot r}{2} \end{bmatrix}$$

Extract row 2:

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

$$\begin{bmatrix} \frac{r}{2} & -\frac{\sqrt{3} \cdot r}{2} \end{bmatrix}$$

解中也可以包含未出现在方程式中的未知数。例如，您可以将z作为一个未知数将原例扩展变化为2个半径为r的平行相交圆柱。这些圆柱解说明解系可能包含形式为@k的任意常数，此处k为1到255之间的整数下标。当您使用**ClrHome**或 $\boxed{\text{F1}}$  8:Clear Home时下标复位为1。

对于多项式系，计算时间或内存占用很大程度上取决于未知数的排列次序。若您的初始选择会占用过多内存或时间，您可以尝试重新排列方程中和/*varOrGuess*列表中变量的次序。

若您不使用任何估计值，且所有的方程式不是任何变量的多项式，而仅仅是未知数的一次式，则**zeros()**使用高斯排除法来确定所有实数解。

若一个方程组既不是其任何变量的多项式，也不是其未知数的一次式，则通过近似叠代法**zeros()**最多只能确定一个解。为此，求解变量的数量必须等于方程的数量，且方程中的所有其它变量必须简化为数字形式。

在带有估计值的情况下，各个求解变量从估计值开始搜索；否则，应从0.0开始。

使用估计值一个一个地搜索其它的解值。为了能够收敛，估计值应该尽可能的靠近解值。

$\text{zeros}(\{x^2+y^2-r^2, (x-r)^2+y^2-r^2\},\{x,y,z\})$   $\boxed{\text{ENTER}}$

$$\begin{bmatrix} \frac{r}{2} \frac{\sqrt{3} \cdot r}{2} @1 \\ \frac{r}{2} \frac{-\sqrt{3} \cdot r}{2} @1 \end{bmatrix}$$

$\text{zeros}(\{x+e^z \cdot y-1, x-y-\sin(z)\},\{x,y\})$   $\boxed{\text{ENTER}}$

$$\begin{bmatrix} \frac{e^z \cdot \sin(z)+1}{e^z+1} \\ \frac{-(\sin(z)-1)}{e^z+1} \end{bmatrix}$$

$\text{zeros}(\{e^z \cdot y-1, y-\sin(z)\},\{y,z\})$   $\boxed{\text{ENTER}}$

[.041... 3.183...]

$\text{zeros}(\{e^z \cdot y-1, y-\sin(z)\},\{y,z=2\pi\})$   $\boxed{\text{ENTER}}$

[.001... 6.281...]

## ZoomBox CATALOG

### ZoomBox

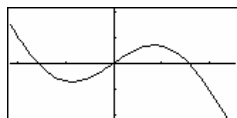
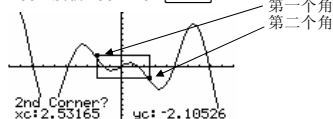
显示Graph (图形) 屏幕，让您画出一个方框来定义新的视窗，并更新窗口内容。

在函数作图模式下：

$1.25x * \cos(x) \Rightarrow y1(x)$   $\boxed{\text{ENTER}}$

**ZoomStd:ZoomBox**  $\boxed{\text{ENTER}}$

**Done**



在第二次按下 $\boxed{\text{ENTER}}$ 定义ZoomBox后的显示结果。

## ZoomData CATALOG

### ZoomData

基于当前定义的图形（和数据）来调整窗口设置，以便对所有统计数据点进行采样并在Graph（图形）屏幕上显示出来。

注：不要调整柱状图的ymin和ymax。

在函数作图模式下：

{1,2,3,4} → L1 [ENTER]

{2,3,4,5} → L2 [ENTER]

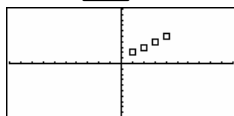
newPlot 1,1,L1,L2 [ENTER]

ZoomStd [ENTER]

{1 2 3 4}

{2 3 4 5}

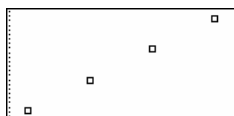
Done



[HOME]

[CALC HOME]

ZoomData [ENTER]



## ZoomDec CATALOG

### ZoomDec

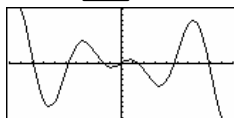
调整视窗，使 $\Delta x$ 和 $\Delta y = 0.1$ ，并在Graph（图形）屏幕上显示出来，此时原点应该在屏幕中央。

在函数作图模式下：

$1.25x * \cos(x) \rightarrow y1(x)$  [ENTER]

ZoomStd [ENTER]

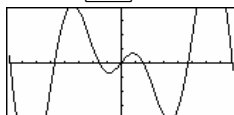
Done



[HOME]

[CALC HOME]

ZoomDec [ENTER]



## ZoomFit CATALOG

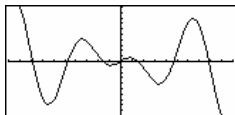
### ZoomFit

显示Graph（图形）屏幕，并计算因变量所需窗口的大小，以便在当前自变量设置下能看到全部图形。

在函数作图模式下：

$1.25x * \cos(x) \rightarrow y1(x)$  [ENTER]

ZoomStd [ENTER]

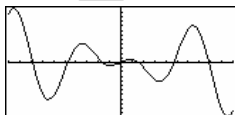


[HOME]



[CALC HOME]

ZoomFit [ENTER]



Done

## ZoomIn CATALOG

### ZoomIn

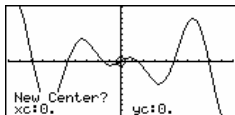
显示Graph（图形）屏幕，您可设置图形放大的中心点，并更新所有视窗。

缩放尺寸取决于Zoom因子xFact和yFact。在3D作图模式下，缩放尺寸取决于xFact、yFact、和zFact。

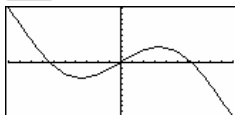
在函数作图模式下：

$1.25x * \cos(x) \rightarrow y1(x)$  [ENTER]

ZoomStd:ZoomIn [ENTER]



[ENTER]



Done

## ZoomInt CATALOG

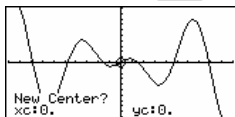
### ZoomInt

显示Graph（图形）屏幕，您可设置图形放大的中心点，并调整窗口设置以便使各个方向上的像素点位置均为整数。

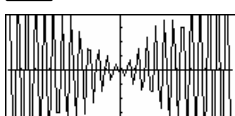
在函数作图模式下：

$1.25x * \cos(x) \rightarrow y1(x)$  [ENTER]

ZoomStd:ZoomInt [ENTER]



[ENTER]



Done

## ZoomOut CATALOG

### ZoomOut

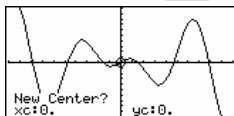
显示Graph (图形) 屏幕, 您可设置图形缩小的中心点, 并更新所有视窗。

缩放尺寸取决于Zoom因子xFact 和 yFact。在3D作图模式下, 缩放尺寸取决于xFact、yFact、和zFact。

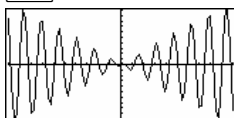
在函数作图模式下:

$1.25x * \cos(x) \rightarrow y1(x)$  **ENTER**  
**ZoomStd:ZoomOut** **ENTER**

Done



**ENTER**



## ZoomPrev CATALOG

### ZoomPrev

显示Graph (图形) 屏幕, 并应用在上次缩放前使用的设置来更新视窗。

## ZoomRcl CATALOG

### ZoomRcl

显示Graph (图形) 屏幕, 并应用存储在中ZoomSto指令中的设置来更新视窗。

## ZoomSqr CATALOG

### ZoomSqr

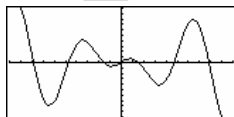
显示Graph (图形) 屏幕, 调整x 或 y 窗口设置, 以便使各像素在坐标系中均代表相等的宽度和高度, 并更新视窗。

在3D作图模式下, **ZoomSqr**将最短的两条轴延伸至与最长轴同样的长度。

在函数作图模式下:

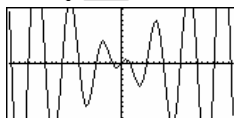
$1.25x * \cos(x) \rightarrow y1(x)$  **ENTER**  
**ZoomStd** **ENTER**

Done



**HOME**

**ZoomSqr** **ENTER**





## ZoomStd CATALOG

### ZoomStd

将窗口变量设置为以下标准值，然后更新视窗。

函数作图：

x: [- 10, 10, 1], y: [- 10, 10, 1] and xres=2

参变量作图：

t: [0, 2 $\pi$ ,  $\pi/24$ ], x: [- 10, 10, 1], y: [- 10, 10, 1]

极坐标作图：

$\theta$ : [0, 2 $\pi$ ,  $\pi/24$ ], x: [- 10, 10, 1], y: [- 10, 10, 1]

序列作图：

nmin=1, nmax=10, plotStrt=1, plotStep=1,  
x: [- 10, 10, 1], y: [- 10, 10, 1]

3D作图：

eye $\theta$ =20, eye $\phi$ =70, eye $\psi$ =0  
x: [- 10, 10, 14], y: [- 10, 10, 14],  
z: [- 10, 10], ncontour=5

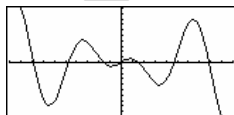
微分方程作图：

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)  $\rightarrow$  y1(x) **ENTER**  
ZoomStd **ENTER**

Done



## ZoomSto CATALOG

### ZoomSto

将当前的窗口设置存储到Zoom内存中。您可以使用ZoomRcl恢复设置。

## ZoomTrig CATALOG

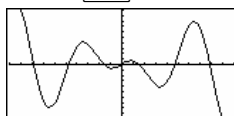
### ZoomTrig

显示Graph（图形）屏幕，设 $\Delta x$ 为 $\pi/24$ ，且 $x_{sc1}$ 为 $\pi/2$ ，以原点为中心，设置y为[- 4, 4, .5]，并更新视窗。

在函数作图模式下：

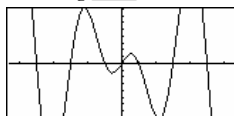
1.25x\* cos(x)  $\rightarrow$  y1(x) **ENTER**  
ZoomStd **ENTER**

Done



**HOME**  
**CALC HOME**

ZoomTrig **ENTER**



+ (加)

**+** 键

表达式1 + 表达式2  $\Rightarrow$  表达式

返回表达式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	{22,π,π/2} ⇒ L1 [ENTER]	{22 π π/2}
矩阵1 + 矩阵2 ⇒ 矩阵	{10,5,π/2} ⇒ L2 [ENTER]	{10 5 π/2}
返回一个数组(或矩阵), 其元素为数组1和数组2(或矩阵1和矩阵2)中所对应元素之和。	L1+L2 [ENTER]	{32 π +5 π}
自变量的维数必须相同。	ans(1)+{π, -5, -π} [ENTER]	{π+32 π 0}
	[a,b;c,d]+[1,0;0,1] [ENTER]	$\begin{bmatrix} a+1 & b \\ c & d+1 \end{bmatrix}$
表达式 + 数组1 ⇒ 数组	15+{10,15,20} [ENTER]	{25 30 35}
数组1 + 表达式 ⇒ 数组	{10,15,20}+15 [ENTER]	{25 30 35}
返回一个数组, 其元素为表达式与数组1中所对应元素相加之和。		
表达式 + 矩阵1 ⇒ 矩阵	20+[1,2;3,4] [ENTER]	$\begin{bmatrix} 21 & 2 \\ 3 & 24 \end{bmatrix}$
矩阵1 + 表达式 ⇒ 矩阵		
返回一个矩阵, 其元素为矩阵1中对角线上所对应元素与表达式相加之和。矩阵1必须为方阵。		
<b>注:</b> 使用.+ (点和) 将表达式分别与各元素相加。		

## - (减) 键

表达式1 - 表达式2 ⇒ 表达式	6 - 2 [ENTER]	4
返回表达式1减去表达式2之差。	π - π/6 [ENTER]	$\frac{5 \cdot \pi}{6}$
数组1 - 数组2 ⇒ 数组	{22,π,π/2} - {10,5,π/2} [ENTER]	{12 π - 5 0}
矩阵1 - 矩阵2 ⇒ 矩阵	{3,4} - [1,2] [ENTER]	[2 2]
返回一个数组(或矩阵), 其元素为数组1(或矩阵1)中各元素中减去数组2(或矩阵2)中所对应元素之差。		
自变量的维数必须相同。		
表达式 - 数组1 ⇒ 数组	15 - {10,15,20} [ENTER]	{5 0 -5}
数组1 - 表达式 ⇒ 数组	{10,15,20} - 15 [ENTER]	{-5 0 5}
返回一个数组, 其元素为表达式减去数组1中各对应元素之差或数组1中各对应元素中减去表达式之差。		
表达式 - 矩阵1 ⇒ 矩阵	20 - [1,2;3,4] [ENTER]	$\begin{bmatrix} 19 & 2 \\ -3 & 16 \end{bmatrix}$
矩阵1 - 表达式 ⇒ 矩阵		
表达式 - 矩阵1 返回表达式乘上单位矩阵再减去矩阵1后得到的矩阵。矩阵1必须为方阵。		
矩阵1 - 表达式 返回矩阵1减去表达式与单位矩阵的乘积后得到的矩阵。矩阵1必须为方阵。		
<b>注:</b> 使用.- (点差) 从各元素中分别减去表达式。		

## \* (乘) 键

表达式1 * 表达式2 ⇒ 表达式	2 * 3.45 [ENTER]	6.9
返回表达式1和表达式2的乘积。	x * y * x [ENTER]	x <sup>2</sup> · y

数组1*数组2 ⇒ 数组	$\{1,0,2,3\} * \{4,5,6\}$ [ENTER]	$\{4, 10, 18\}$
返回由数组1和数组2中的对应元素的乘积组成的数组。	$\{2/a, 3/2\} * \{a^2, b/3\}$ [ENTER]	$\{2 \cdot a \frac{b}{2}\}$
数组的维数必须相同。		

矩阵1*矩阵2 ⇒ 矩阵	$[1,2,3;4,5,6] * [a,d;b,e;c,f]$ [ENTER]	
返回矩阵1和矩阵2的矩阵乘积。	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}$ $\begin{bmatrix} 1 \cdot a + 2 \cdot b + 3 \cdot c & 1 \cdot d + 2 \cdot e + 3 \cdot f \\ 4 \cdot a + 5 \cdot b + 6 \cdot c & 4 \cdot d + 5 \cdot e + 6 \cdot f \end{bmatrix}$	
矩阵1的行数必须与矩阵2的列数相同。		

表达式*数组1 ⇒ 数组	$\pi * \{4,5,6\}$ [ENTER]	$\{4 \cdot \pi, 5 \cdot \pi, 6 \cdot \pi\}$
数组1*表达式 ⇒ 数组		
返回由表达式分别与数组1中各元素的乘积组成的数组。		

表达式*矩阵1 ⇒ 矩阵	$[1,2;3,4] * .01$ [ENTER]	$\begin{bmatrix} .01 & .02 \\ .03 & .04 \end{bmatrix}$
矩阵1*表达式 ⇒ 矩阵		
返回表达式与矩阵1中各元素的乘积所组成的矩阵。	$\lambda * \text{identity}(3)$ [ENTER]	$\begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix}$
注：使用.* (点积) 将表达式分别与各元素相乘。		

**/ (除)      键**

表达式1/表达式2 ⇒ 表达式	$2/3.45$ [ENTER]	$.57971$
返回表达式1除以表达式2的商。	$x^3/x$ [ENTER]	$x^2$

数组1/数组2 ⇒ 数组	$\{1,0,2,3\}/\{4,5,6\}$ [ENTER]	$\{.25 \ 2/5 \ 1/2\}$
返回一个由数组1除以数组2的商组成的数组。		
两个数组的维数必须相同。		

表达式/数组1 ⇒ 数组	$a/\{3,a,\sqrt{a}\}$ [ENTER]	$\{\frac{a}{3} \ 1 \ \sqrt{a}\}$
数组1/表达式 ⇒ 数组		
返回由表达式除以数组1或数组1除以表达式的商组成的数组。	$\{a,b,c\}/(a*b*c)$ [ENTER]	$\{\frac{1}{b \cdot c} \ \frac{1}{a \cdot c} \ \frac{1}{a \cdot b}\}$

矩阵1/表达式 ⇒ 矩阵	$[a,b,c]/(a*b*c)$ [ENTER]	$\begin{bmatrix} \frac{1}{b \cdot c} & \frac{1}{a \cdot c} & \frac{1}{a \cdot b} \end{bmatrix}$
返回由矩阵1除以表达式的商组成的矩阵。		
注：使用./ (点除) 使各元素除以表达式。		

**^ (幂)      键**

表达式1^表达式2 ⇒ 表达式	$4^2$ [ENTER]	$16$
数组1^数组2 ⇒ 数组	$\{a,2,c\}^{\{1,b,3\}}$ [ENTER]	$\{a \ 2b \ c^3\}$

返回以第一自变量为底、第二自变量为幂的计算结果。

对于数组，返回以数组1中各元素为底、数组2中对应元素为幂的计算结果。

在实数域内，化简的奇分母分式幂将使用实数支，而在复数模式下使用主支。

表达式  $\wedge$  数组1  $\Rightarrow$  数组

返回以表达式为底，数组1中元素为幂的计算结果。

$p^{\{a,2,-3\}}$  [ENTER]

$$\{p^a \ p^2 \ \frac{1}{p^3}\}$$

数组1  $\wedge$  表达式  $\Rightarrow$  数组

返回以数组1中元素为底，表达式为幂的计算结果。

$\{1,2,3,4\}^{\wedge} 2$  [ENTER]

$$\{1 \ 1/4 \ 1/9 \ 1/16\}$$

方阵1  $\wedge$  整数  $\Rightarrow$  矩阵

返回方阵1的整数次幂。

方阵1必须为方阵。

若整数 = -1，计算转置矩阵。  
若整数 < -1，以合适的正数幂计算逆矩阵。

$[1,2;3,4]^{\wedge} 2$  [ENTER]

$[1,2;3,4]^{\wedge} -1$  [ENTER]

$[1,2;3,4]^{\wedge} -2$  [ENTER]

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^2$	$\begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$
$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1}$	$\begin{bmatrix} -2 & 1 \\ 3/2 & -1/2 \end{bmatrix}$
$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-2}$	$\begin{bmatrix} 11/2 & -5/2 \\ -15/4 & 7/4 \end{bmatrix}$

### +. (点和) + 键

矩阵1.+ 矩阵2  $\Rightarrow$  矩阵

表达式.+ 矩阵1  $\Rightarrow$  矩阵

矩阵1.+ 矩阵2 返回由矩阵1和矩阵2对应元素之和组成的矩阵。

表达式.+ 矩阵1 返回由表达式与矩阵1中各元素之和组成的矩阵。

$[a,2;b,3].+[c,4;d,5]$  [ENTER]

$x.+[c,4;d,5]$  [ENTER]

$\begin{bmatrix} b & 3 \end{bmatrix} .+ \begin{bmatrix} 5 & d \end{bmatrix}$	$\begin{bmatrix} a+c & 6 \\ b+5 & d+3 \end{bmatrix}$
$x .+ \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} x+c & x+4 \\ x+5 & x+d \end{bmatrix}$

### -. (点差) - 键

矩阵1.- 矩阵2  $\Rightarrow$  矩阵

表达式.- 矩阵1  $\Rightarrow$  矩阵

矩阵1.- 矩阵2 返回由矩阵1和矩阵2对应元素之差组成的矩阵。

表达式.- 矩阵1 返回由表达式与矩阵1中各元素之差组成的矩阵。

$[a,2;b,3].-[c,4;d,5]$  [ENTER]

$x.-[c,4;d,5]$  [ENTER]

$\begin{bmatrix} b & 3 \end{bmatrix} .- \begin{bmatrix} d & 5 \end{bmatrix}$	$\begin{bmatrix} a-c & -2 \\ b-d & -2 \end{bmatrix}$
$x .- \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} x-c & x-4 \\ x-d & x-5 \end{bmatrix}$

### .\* (点积) \* 键

矩阵1.\* 矩阵2  $\Rightarrow$  矩阵

表达式.\* 矩阵1  $\Rightarrow$  矩阵

矩阵1.\* 矩阵2 返回由表达式与矩阵1中各元素的乘积组成的矩阵。

表达式.\* 矩阵1 返回由表达式与矩阵1中各元素的乘积组成的矩阵。

$[a,2;b,3].*[c,4;d,5]$  [ENTER]

$x.*[a,b;c,d]$  [ENTER]

$\begin{bmatrix} b & 3 \end{bmatrix} .* \begin{bmatrix} 5 & d \end{bmatrix}$	$\begin{bmatrix} a \cdot c & 8 \\ 5 \cdot b & 3 \cdot d \end{bmatrix}$
$x .* \begin{bmatrix} a & b \\ c & d \end{bmatrix}$	$\begin{bmatrix} a \cdot x & b \cdot x \\ c \cdot x & d \cdot x \end{bmatrix}$

### ./ (点商) / 键

矩阵1./ 矩阵2  $\Rightarrow$  矩阵

表达式./ 矩阵1  $\Rightarrow$  矩阵

矩阵1./ 矩阵2 返回由表达式与矩阵1中各元素的商组成的矩阵。

表达式./ 矩阵1 返回由表达式与矩阵1中各元素的商组成的矩阵。

$[a,2;b,3]./[c,4;d,5]$  [ENTER]

$x./[c,4;d,5]$  [ENTER]

$\begin{bmatrix} b & 3 \end{bmatrix} ./ \begin{bmatrix} 5 & d \end{bmatrix}$	$\begin{bmatrix} \frac{b}{5} & \frac{3}{d} \end{bmatrix}$
$x ./ \begin{bmatrix} c & 4 \\ 5 & d \end{bmatrix}$	$\begin{bmatrix} \frac{x}{c} & \frac{x}{4} \\ \frac{x}{5} & \frac{x}{d} \end{bmatrix}$



## **= (等于)**    **[=]** 键

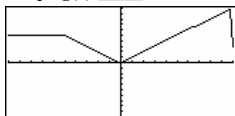
表达式1 = 表达式2 ⇒ 布尔表达式  
数组1 = 数组2 ⇒ 布尔数组  
矩阵1 = 矩阵2 ⇒ 布尔矩阵

若确认表达式1等于表达式2则返回true。  
若确认表达式1不等于表达式2则返回false。  
其它情况则返回等式的化简形式。  
对于数组和矩阵，返回各对应元素间的比较结果。

示例函数给出了使用数学测试符号的结果：  
=, ≠, <, ≤, >, ≥

```
: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

若确认表达式1不等于表达式2则返回true。  
若确认表达式1等于表达式2则返回false。  
其它情况则返回等式的化简形式。  
对于数组和矩阵，返回各对应元素间的比较结果。

参见 "=" (等于) 中的示例。

## **<**    **[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

若确认表达式1小于表达式2则返回true。  
若确认表达式1大于或等于表达式2则返回false。  
其它情况则返回等式的化简形式。  
对于数组和矩阵，返回各对应元素间的比较结果。

参见 "=" (等于) 中的示例。

<

**[<][=] 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

参见 "=" (等于) 中的示例。

若确认表达式1小于或等于表达式2则返回true。

若确认表达式1大于表达式2则返回false。

其它情况则返回等式的化简形式。

对于数组和矩阵，返回各对应元素间的比较结果。

>

**[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

参见 "=" (等于) 中的示例。

若确认表达式1大于表达式2则返回true。

若确认表达式1小于或等于表达式2则返回false。

其它情况则返回等式的化简形式。

对于数组和矩阵，返回各对应元素间的比较结果。

>

**[>][=] 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

参见 "=" (等于) 中的示例。

若确认表达式1大于或等于表达式2则返回true。

若确认表达式1小于表达式2则返回false。

其它情况则返回等式的化简形式。

对于数组和矩阵，返回各对应元素间的比较结果。

**!(阶乘)**

**[<][=] 键**

**[2nd] W 键**

表达式!n ⇒ 表达式  
数组!n ⇒ 数组  
矩阵!n ⇒ 矩阵

5! [ENTER] 120

{5,4,3}! [ENTER] {120 24 6}

返回自变量的阶乘。

[1,2;3,4]! [ENTER]  $\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

对于数组或矩阵，返回由各元素的阶乘组成的数组或矩阵。

在TI-89计算器中，仅能计算非负整数的阶乘。

**& (添加)**

**[<][=] 键**

**[2nd] H 键**

string1 & string2 ⇒ string

"Hello " & "Nick" [ENTER]

"Hello Nick"

返回将string2添加到string1后的文本字符串。

**[( ) (积分)**

**[2nd][J] 键**

[(表达式1, var[, lower] [, upper]) ⇒ 表达式  
[(数组1, var[, order]) ⇒ 数组  
[(矩阵1, var[, order]) ⇒ 矩阵

返回表达式  $f$  关于变量  $var$  从  $lower$  到  $upper$  的积分。

若省略  $lower$  和  $upper$  则返回不定积分。该结果省略了象  $C$  这样的符号常数。

然而，仅在省略  $upper$  时， $lower$  将作为积分常数出现在结果中。

等效的不定积分之间相差一个数值常数。该常数可能被隐藏起来——特别是在不定积分包含对数或反三角函数时。然而，有时会加入分段常数表达式使不定积分在一个大区上有效，此时不使用普通公式。

$$\int(x^2, x, a, b) \text{ [ENTER]} \quad \frac{b^3}{3} - \frac{a^3}{3}$$

$$\int(x^2, x) \text{ [ENTER]} \quad \frac{x^3}{3}$$

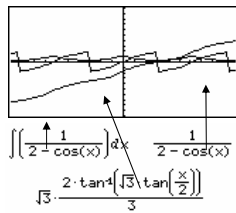
$$\int(a * x^2, x, c) \text{ [ENTER]} \quad \frac{a * x^3}{3} + c$$

$$\int(1/(2 - \cos(x)), x) \rightarrow \text{tmp}(x) \text{ [ENTER]}$$

ClrGraph:Graph tmp(x):Graph

$$1/(2 - \cos(x)):Graph \sqrt{(3)}$$

$$(2 \tan^{-1}(\sqrt{(3)(\tan(x/2))}/3)) \text{ [ENTER]}$$



如果在该段中无法确定其内建函数和算子的显式的有限组合时， $\int()$  将分段返回表达式  $f$  自身。

当同时使用  $lower$  和  $upper$  时，将在区间  $lower < var < upper$  内寻找所有间断点或非连续可导点，并在这些位置将区间分割成不同的子区间。

对于Exact/Approx模式的AUTO（自动）设置，数值积分适用于无法确定不定积分或极限的情形。

对于APPROX设置，若可行，则首先采用数值积分。只有在上述数值积分不适用或不起作用时才采用不定积分。

可嵌套使用 $\int()$ 来计算多重积分。积分极限可能取决于积分函数外面的积分变量。

注：参见nInt()。

$$\int(b * e^{(-x^2)} + a/(x^2 + a^2), x) \text{ [ENTER]}$$

$$\int \left[ \left[ b \cdot e^{-x^2} + \frac{a}{x^2 + a^2} \right] dx \right]$$

$$b \cdot \int \left[ e^{-x^2} dx + \tan^{-1} \left( \frac{x}{a} \right) \right]$$

$$\int(e^{(-x^2)}, x, -1, 1) \text{ [ENTER]} \quad 1.493\dots$$

$$\int(\int(\ln(x+y), y, 0, x), x, 0, a) \text{ [ENTER]}$$

$$\int \int_0^a \int_0^x \ln(x+y) dy dx$$

$$\frac{a^2 \cdot \ln(a)}{2} + a^2 \cdot (\ln(2) - 3/4)$$

## $\sqrt{()}$ (平方根) [2nd][ $\sqrt{\quad}$ ] 键

$$\sqrt{\text{表达式}} \Rightarrow \text{表达式}$$

$$\sqrt{\text{数组}} \Rightarrow \text{数组}$$

返回自变量的平方根。

对于数组，返回数组  $T$  中所有元素的平方根。

$$\sqrt{4} \text{ [ENTER]} \quad 2$$

$$\sqrt{\{9, a, 4\}} \text{ [ENTER]} \quad \{3 \sqrt{a} 2\}$$

## $\Pi()$ (乘积) MATH/Calculus 菜单

$$\Pi(\text{表达式}, var, low, high) \Rightarrow \text{表达式}$$

计算表达式  $f$  在变量  $var$  从  $low$  到  $high$  取值时对应的结果，并返回上述结果的乘积。

$$\Pi(1/n, n, 1, 5) \text{ [ENTER]} \quad \frac{1}{120}$$

$$\Pi(k^2, k, 1, n) \text{ [ENTER]} \quad (n!)^2$$

$$\Pi(\{1/n, n, 2\}, n, 1, 5) \text{ [ENTER]}$$

$$\left\{ \frac{1}{120} 120 32 \right\}$$



$\Pi(\text{表达式}1, \text{var}, \text{low}, \text{low}-1) \Rightarrow 1$	$\Pi(k,k,4,3)$ [ENTER]	1
$\Pi(\text{表达式}1, \text{var}, \text{low}, \text{high}) \Rightarrow 1/\Pi(\text{表达式}1, \text{var}, \text{high}+1, \text{low}-1)$ , 若 $\text{high} < \text{low}-1$	$\Pi(1/k,k,4,1)$ [ENTER]	6
	$\Pi(1/k,k,4,1) \cdot \Pi(1/k,k,2,4)$ [ENTER]	1/4

### $\Sigma()$ (求和) MATH/Calculus 菜单

$\Sigma(\text{表达式}1, \text{var}, \text{low}, \text{high}) \Rightarrow$ 表达式	$\Sigma(1/n,n,1,5)$ [ENTER]	$\frac{137}{60}$
计算表达式1在变量var从low到high取值时所对应的结果，并返回上述结果之和。	$\Sigma(k^2,k,1,n)$ [ENTER]	$\frac{n \cdot (n+1) \cdot (2 \cdot n+1)}{6}$
	$\Sigma(1/n^2,n,1,\infty)$ [ENTER]	$\frac{\pi^2}{6}$
$\Sigma(\text{表达式}1, \text{var}, \text{low}, \text{low}-1) \Rightarrow 0$	$\Sigma(k,k,4,3)$ [ENTER]	0
$\Sigma(\text{表达式}1, \text{var}, \text{low}, \text{high}) \Rightarrow -\Sigma(\text{表达式}1, \text{var}, \text{high}+1, \text{low}-1)$ 若 $\text{high} < \text{low}-1$	$\Sigma(k,k,4,1)$ [ENTER]	-5
	$\Sigma(k,k,4,1)+\Sigma(k,k,2,4)$ [ENTER]	4

### # (间接引用) CATALOG

# varNameString	程序段:
调用名称为varNameString的变量。该功能使您可以在程序中使用字符串来创建和修改变量。	: :Request "Enter Your Name",str1 :NewFold #str1 : : :For i,1,5,1 :ClrGraph :Graph i*x :StoPic #("pic" & string(i)) :EndFor :

### $\ulcorner$ (弧度) MATH/Angle 菜单

表达式 $\ulcorner \Rightarrow$ 表达式	在Degree (角度) 或Radian (弧度) 模式下:	
数组 $\ulcorner \Rightarrow$ 数组		
矩阵 $\ulcorner \Rightarrow$ 矩阵		
在Degree (角度) 模式下, 将表达式1乘以 $180/\pi$ 。在Radian (弧度) 模式下, 原样返回表达式1。	$\cos(\pi/4)\ulcorner$ [ENTER]	$\frac{\sqrt{2}}{2}$
该函数让您能在Degree (角度) 模式中使用弧度角。(在Degree (角度) 模式下, $\sin()$ 、 $\cos()$ 、 $\tan()$ 、和极坐标到直角坐标的转换中的自变量需要采用角度形式。)	$\cos(\{0\ulcorner, (\pi/12)\ulcorner, -\pi\ulcorner\})$ [ENTER]	$\{1 \frac{(\sqrt{3}+1) \cdot \sqrt{2}}{4} - 1\}$
<b>提示:</b> 如果您希望在使用函数或程序时无论采用何种模式, 均强制在函数或程序定义中使用弧度角, 则可使用 $\ulcorner$ 。		



**\_(下划线) [ ] 键 [2nd] [ ] 键**

表达式\_单位 3\_m▶\_ft [ENTER] 9.842...·\_ft

为表达式指定单位。所有单位名称必须以下划线开始。

您可以使用预定义单位或创建您个人使用的单位。对于预定义单位的数组，可参见有关常数和度量单位的内容。您可以按下：

[UNITS]

[UNITS]

来从菜单中选择单位，或您可以直接键入单位名称。

**注：** 欲输入▶，可按下 [▶]。

**变量**

假设z未定义：

当变量未取数值时，将作为代表复数来处理。在缺省情况下，若没有“\_”，则变量视为实数处理。

**real(z)** [ENTER] z  
**real(z\_)** [ENTER] real(z\_)

若变量已赋值，则“\_”可省略且变量保持其初始数据类型。

**imag(z)** [ENTER] 0  
**imag(z\_)** [ENTER] imag(z\_)

**注：** 您也可不用“\_”将一个复数保存到一个变量。然而，为了在**cSolve()**和**cZeros()**这样的函数中获得最佳的计算结果，推荐使用“\_”。

**▶(转换) [▶] 键**

表达式\_单位1▶\_单位2 ⇒ 表达式\_单位2 3\_m▶\_ft [ENTER] 9.842...·\_ft

将表达式从一种单位转换到另一种。上述单位必须属于同一分类。

下划线( )字符指出为单位。对于有效预定义单位的数组，可参见有关常数和度量单位的内容。您可以按下：

[UNITS]

[UNITS]

从菜单中选择单位，或您可以直接键入单位名称。

要在直接键入单位时获得下划线( )，可按下：

[ ]

[ ]

**注：** ▶转换算子不适用于温度单位，此时应改用**tmpCnv()**和**ΔtmpCnv()**。

**10^( ) CATALOG**

**10^(表达式*n*) ⇒ 表达式** **10^(1.5)** [ENTER] 31.622...

**10^(数组*n*) ⇒ 数组**

返回以10为底、自变量为幂的计算结果。

**10^{0,-2,2,a}** [ENTER]  $\{1 \frac{1}{100} 100 10^a\}$

对于数组，返回以10为底、数组*n*中各元素为幂的计算结果。

**10^(方阵*n*) ⇒ 方阵** **10^{(1,5,3;4,2,1;6,-2,1)}** [ENTER]

返回以10为底、方阵*n*为幂的计算结果。该运算不同于求以10为底、方阵*n*的各元素为幂的值。欲了解与计算方法有关的内容，可参见**cos()**。

$\begin{bmatrix} 1.143...E7 & 8.171...E6 & 6.675...E6 \\ 9.956...E6 & 7.115...E6 & 5.813...E6 \\ 7.652...E6 & 5.469...E6 & 4.468...E6 \end{bmatrix}$

方阵*n*必须可对角化。结果中总包含有浮点数。

表达式  $1x^{-1} \Rightarrow$  表达式  
数组  $1x^{-1} \Rightarrow$  数组

返回自变量的倒数。

对于数组，返回数组  $r$  中所有元素的倒数。

$3.1^{-1}$  [ENTER]

.322581

$\{a,4,-1,x-2\}^{-1}$  [ENTER]

$\left\{\frac{1}{a}, \frac{1}{4}, -10, \frac{1}{x-2}\right\}$

方阵  $1x^{-1} \Rightarrow$  方阵

返回方阵  $r$  的逆矩阵。

方阵  $r$  必须为非退化方阵。

$[1,2;3,4]^{-1}$  [ENTER]

$[1,2;a,4]^{-1}$  [ENTER]

$$\begin{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} & \begin{bmatrix} -2 & 1 \\ 3/2 & -1/2 \end{bmatrix} \\ \begin{bmatrix} 1 & 2 \\ a & 4 \end{bmatrix}^{-1} & \begin{bmatrix} -2 & 1 \\ \frac{a}{2(a-2)} & \frac{-1}{2(a-2)} \end{bmatrix} \end{bmatrix}$$

表达式 | 布尔表达式1 [and 布尔表达式2]...  
[and 布尔表达式M]

“with” ( ) 符号为二进制算子。| 左边的操作数为表达式。| 右边的操作数指定了与化简表达式有关的一到两种关系。| 后的多种关系必须用逻辑“and”连接。

“with”算子提供3种基本类型的功能：代换、区间约束、和排除。

代换为等式形式，例如  $x=3$  或  $y=\sin(x)$ 。为了运算最有效，左侧应为简单变量。表达式 | 变量 = 数值 会将数值代入表达式中将变量替换掉。

区间约束采用一个或多个通过“and”算子连接的逻辑不等式。区间约束也允许采用在其他情形下可能无效或无法计算的简化形式。

使用关系算子“不等于” ( $\neq$  或  $\neq$ ) 将特定值排除在考虑范围外。在使用 **cSolve()**、**cZeros()**、**fMax()**、**fMin()**、**solve()**、**zeros()** 等指令时，它主要用来排除某个准确解。

$x+1|x=3$  [ENTER]

4

$x+y|x=\sin(y)$  [ENTER]

$\sin(y)+y$

$x+y|\sin(y)=x$  [ENTER]

$x+y$

$x^3-2x+7 \triangleright f(x)$  [ENTER]

Done

$f(x)|x=\sqrt{3}$  [ENTER]

$\sqrt{3}+7$

$(\sin(x))^2+2\sin(x)-6|\sin(x)=d$  [ENTER]

$d^2+2d-6$

$\text{solve}(x^2-1=0,x)|x>0 \text{ and } x<2$  [ENTER]

$x=1$

$\sqrt{(x)*\sqrt{(1/x)}|x>0$  [ENTER]

1

$\sqrt{(x)*\sqrt{(1/x)}$  [ENTER]

$\sqrt{\frac{1}{x}} \cdot \sqrt{x}$

$\text{solve}(x^2-1=0,x)|x \neq 1$  [ENTER]

$x = -1$

→ (存储) **[STO] 键**

表达式 → var	$\pi/4 \rightarrow \text{myvar}$ [ENTER]	$\frac{\pi}{4}$
数组 → var		
矩阵 → var		
表达式 → fun_name(参数1,...)	$2\cos(x) \rightarrow Y1(x)$ [ENTER]	Done
数组 → fun_name(参数1,...)	$\{1,2,3,4\} \rightarrow \text{Lst5}$ [ENTER]	{1 2 3 4}
矩阵 → fun_name(参数1,...)	$[1,2,3;4,5,6] \rightarrow \text{MatG}$ [ENTER]	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$
若变量 var 不存在, 则创建 var 并以表达式、数组、或矩阵对其赋初值。	"Hello" → str1 [ENTER]	"Hello"
若 var 已经存在且未被加锁或保护, 则用表达式、数组、或矩阵替换其值。		

**提示:** 若您打算使用未定义变量进行符号计算, 应避免对常用单字母变量赋值赋值, 如 a、b、c、x、y、z 等。

⊙ (注释) **Program Editor/Control 菜单** 或按

**键**  
**X 键**

⊙ [文本]

- ⊙ 将文本作为注释行处理, 可以用来对程序指令进行注释。
- ⊙ 可以在行首或行间的任何位置。从 ⊙ 向右直到行尾的任何内容均为注释内容。

程序段:

```

:
:
:⊙ Get 10 points from the Graph
screen
:For i,1,10 ⊙ This loops 10 times
:

```

**0b, 0h** [B] **键** **B 键**  
 [H] **键** **H 键**

**0b** binaryNumber

**0h** hexadecimalNumber

在十进制模式下:

$0b10+0hF+10$  [ENTER] 27

在二进制模式下:

$0b10+0hF+10$  [ENTER] 0b11011

在十六进制模式下:

$0b10+0hF+10$  [ENTER] 0h1B

分别标志二进制数和十六进制数。欲输入二进制数或十六进制数, 在任何数基模式下, 您都必须输入 0b 或 0h 前缀。若不带前缀, 则数字将作为十进制 (数基为 10) 处理。

结果将根据数基模式显示出来。



## 附录 B: TI 产品和服务信息

### **TI 产品和服务信息**

更多关于 TI 产品和服务的信息，请通过电子邮件与 TI 联系或访问万维网上的 TI 计算器主页。

Email: [ti-cares@ti.com](mailto:ti-cares@ti.com)

网址: [education.ti.com](http://education.ti.com)

### **维修和保修信息**

关于保修期限和条款的信息或关于产品维修的信息，请参阅随产品一起封装的保修协议书，或请与当地 Texas Instruments 分销商或代理商联系。





# 索引

## 符号

→, 存储

!, 阶乘

", "秒" 符号

≠, /≠, 不等于

#, 间接引用

$\sqrt{\quad}$ , 平方根

%, 百分比

&, 添加

$\Sigma(\quad)$ , 求和

$\int(\quad)$ , 积分

\*, 乘

' , (方程) 阶数

**[F1]–[F8]** (功能键)

°, "度" 符号

' , "分" 符号

–, 减

–, 求非

∠, 角

.\*, 点积

., 点差

./, 点商

.^, 点幂

., 点和

/, 除号

**[–]** (负号键)

$\Delta\text{list}(\quad)$ , 数组差

$\Delta\text{tmpCnv}(\quad)$ , 温度范围转换

^, 幂

–, 下划线

≤, <=, 小于等于

**[–]** (减号键)

≥, >=, 大于等于

|, 与

Ⓞ, 注释

+, 加

<, 小于

=, 等于

>, 大于

**[←] / [→] [DEL]** (删除符号)

**[☞]** ("手形" 功能调节键)

**[2nd]** ("第二" 功能调节键)

**[2nd] [MEM]** (内存)

**[2nd] [EE]** (指数键)

**[2nd] [▶]** (单位转换)

**[2nd] [RCL]** (调用)

**[2nd] [CATALOG]** (目录)

**[2nd] [CUSTOM]** (定制)

**[2nd] [CHAR]** (字符)

**[2nd] [QUIT]**

▶, 转换

▶Bin, 以 2 进制显示

▶Cylind, 以圆柱坐标向量显示

▶DD, 以 10 进制角度显示

▶Dec, 以 10 进制整数显示

▶DMS, 以度/分/秒显示

▶Hex, 以 16 进制显示

▶Polar, 以极坐标向量显示

▶Rect, 以直角坐标向量显示

▶Sphere, 以球坐标向量显示

**[↑]** ("上档" 功能调节键)

**[◆]** ("钻石" 功能调节键)

**[◆] F** (格式/图形格式)

**[◆] N** (新文档)

**[◆] O** (打开文档)

**[◆] S** (将副本另存为)

**[STO▶]** (存储) 键

**[☉] [☽] [☿]** (光标键)

E, 指数字符号

$\Pi(\quad)$ , 乘积

r, 弧度

T, 转置矩阵

$x^{-1}$ , 倒数

## 数字

0b, 二进制标记

0h, 16 进制标记

$10^{\wedge}()$ , 10 的幂

3D (3 维) 模式

## A

abs(), 绝对值

All category, All(全部)类别

AndPic, 布尔“与”图

and, (布尔运算)和

angle(), 角

animation, 动画

ans(), 最近答案项

APD, 自动断电

APPLICATIONS, [APPS]菜单

APPROX, 近似模式

approx(), 近似值

Apps, 应用软件

Apps desktop, Apps桌面

Archive, 变量存档

arcLen(), 弧长

augment(), 增广/连接

AUTO, 自动模式

avgRC(), 平均变化率

## B

Base mode, 数基模式

BldData, 创建数据

Boolean, 布尔运算

Busy/Pause, 忙/暂停状态

## C

Catalog ([\[2nd\]](#) [CATALOG]), 目录

categories, 类别

CBL/CBL2, 基于计算器的实验室

CBR, 基于计算器的测距仪

ceiling(), 返回大于等于且最接近自变量的整数

cFactor(), 复数因式分解

char(), 字符串

checkTmr(), 检查计时器

Circle, 画圆

ClockOff, 关闭时钟

ClockOn, 开启时钟

ClrDraw, 清除绘图

ClrErr, 清除错误

ClrGraph, 清除图形

ClrHome, 清空主屏幕

ClrIO, 清空

ClrIO, 清空 输入/输出

colDim(), 矩阵列维数

colNorm(), 矩阵列范数

combinations, 组合

comDenom(), 公分母

complex, 复数

conj(), 共轭复数

Constant Memory, 常驻内存

CopyVar, 复制变量

cos(), 余弦函数

$\cos^{-1}()$ , 反余弦函数

cosh(), 双曲余弦函数

$\cosh^{-1}()$ , 双曲反余弦函数

cot(), 余切函数

$\cot^{-1}()$ , 反余切函数

coth(), 双曲反余切函数

$\coth^{-1}()$ , 反双曲余切函数

crossP(), 叉积

csc(), 余割函数

$\csc^{-1}()$ , 反余割函数

csch(), 双曲余割函数

$\operatorname{csch}^{-1}()$ , 反双曲余割函数

cSolve(), 求复数解

CubicReg, 三次回归

cumSum(), 累积和

CustmOff, 关闭定制工具条  
CustmOn, 开启定制工具条  
Cycle, 循环  
CyclePic, 循环图  
cZeros(), 求使表达式为零的自变量复  
数值

## D



d(), 一阶导数  
Data/Matrix Editor, 数据/矩阵编辑器  
dayOfWk(), 星期几  
DE, 微分方程  
Define, 定义  
DEG 度数  
DelFold, 删除文件夹  
DelVar, 删除变量  
denominator, 分母  
derivatives, 导数  
deSolve(), 求解  
det(), 矩阵行列式  
diag(), 矩阵对角线  
Dialog, 定义对话框  
dim(), 维数  
Disp, 显示输入/输出屏幕  
DispG, 显示图形  
DispHome, 显示主屏幕  
DispTbl, 显示表格  
dotP(), 点积  
DrawFunc, 绘图函数  
DrawInv, 绘制反函数  
DrawParm, 绘制参数函数  
DrawPol, 绘制极坐标函数  
DrawSlp, 绘制斜线  
DropDown, 下拉菜单  
DrwCtour, 绘制等高线

## E

e^(),  $e$  的幂

eigVc(), 特征向量  
eigVl(), 特征值  
Else, else(否则)语句  
Elseif, 否则若 (语句)  
EndCustm, 结束定制 (语句)  
EndDlog, 结束对话 (语句)  
EndFor, 结束 For (语句)  
EndFunc, 结束函数 (语句)  
EndIf, 结束 if (语句)  
EndLoop, 结束循环 (语句)  
EndPrgm, 结束程序 (语句)  
EndTBar, 结束工具条 (语句)  
EndTry, 结束 try (语句)  
EndWhile, 结束 while (语句)  
entry(), 输入  
EXACT, 精确模式  
exact(), 求精确值  
Exec, 执行汇编语言  
Exit, 退出  
expIlist(), 以数组形式返回方程解  
expand(), 展开  
expr(), 按字符串生成表达式并执行  
ExpReg, 指数回归

## F

factor(), 因式分解  
File, new (  N), 新建文件  
File, open (  O), 打开文件  
Fill, matrix fill, 矩阵文件  
Flash Apps, 闪存应用程序  
floor(), 返回小于等于且最接近自变量  
的整数  
fMax(), 函数最大值  
fMin(), 函数最小值  
FnOff, 函数关闭  
FnOn, 函数开启  
For, for 语句  
FORMATS (  F), 格式

**format()**, 格式字符串  
**fpart()**, 函数部分  
**Fractions**, 分数  
**Func()**, 定义函数  
**FUNC**, 函数模式

## G

**gcd()**, 最大公因子  
**Get**, 获取/返回 CBL/CBR 值  
**GetCalc**, 获取/返回另一计算器的值  
**getConfig()**, 获取/返回配置  
**getDate()**, 获取时间  
**getDenom()**, 获取/返回分母  
**getDtFmt()**, 获取日期格式  
**getDtStr()**, 获取日期字符串  
**getKey()**, 获取/返回键值  
**getMode()**, 获取/返回模式  
**getNum()**, 获取/返回数值  
**getTime()**, 获取时间  
**getTmFmt()**, 获取时间格式  
**getTmStr()**, 获取时间字符串  
**getTmZn()**, 获取时区值  
**getType()**, 获取/返回类型  
**getUnits()**, 获取/返回单位  
**Goto**, 转到  
**Graph**, 作图, 图形  
**GRAPH FORMATS** (▣ F), 图形格式

## H

**hexadecimal**, 16 进制  
**hidden surface**, 不可见表面  
**highlighting**, 高亮  
**History area**, 历史区域  
**Home screen**, 主屏幕

## I

**I/O**, 输入/输出  
**identity()**, 单位矩阵  
**If**, 条件语句

**imag()**, 虚部  
**Input**, 输入  
**InputSt**, 输入字符串  
**inString()**, 返回子字符串起始位置  
**int()**, 取整  
**intDiv()**, 整除  
**iPart()**, 整数部分  
**isClkOn()**, 时钟处于开启状态  
**isPrime()**, 素数测试  
**Item**, 菜单项

## L

**Lbl**, 标签  
**lcm**, 最小公倍数  
**left()**, 返回数组中最左边起指定数量的元素构成的子数组  
**limit()**, 求极限  
**Line**, 绘制直线  
**LineHorz**, 绘制水平线  
**LineTan**, 绘制切线  
**LineVert**, 绘制垂直线  
**LinReg**, 线性回归  
**list▶mat()**, 用数组按行填充矩阵  
**ln()**, 自然对数  
**LnReg**, 对数回归  
**Local**, 局部变量  
**Lock**, 锁定变量  
**log()**, 对数  
**Logistic**, 逻辑回归  
**Loop**, 循环  
**LU**, 矩阵的下-上分解

## M

**mat▶list()**, 用矩阵按行生成数组  
**max()**, 最大值  
**MATH menu** (2nd [MATH]) 数学菜单  
**mean()**, 均值  
**median()**, 中值

MedMed, 中位数-中位数线性回归  
MEMORY (2nd [MEM]), 内存  
mid(), 返回源字符串中指定位置和长度的子字符串  
min(), 最小值  
mod(), 取模  
MoveVar, 移动变量  
mRow(), 矩阵行运算  
mRowAdd(), 矩阵中某一行乘上某常数再添加到另外一行

## N

nCr(), 组合  
nDeriv(), 导数值  
NewData, 新建数据  
NewFold, 新建文件夹  
newList(), 新建数组  
newMat(), 新建矩阵  
NewPic, 新建图形  
NewPlot, 新建绘图  
NewProb, 新建问题  
nInt(), 数值积分  
norm(), Frobenius 范数  
not, 布尔“非”  
nPr(), 排列  
nSolve(), 求数值解

## O

OneVar, 单变量统计  
or, 布尔“或”  
ord(), 字符的数字编码  
Organizr, 组织器  
OS 操作系统  
Output, 输出  
overwrite mode (2nd [INS])覆盖模式

## P

P►Rx(), 直角 x 坐标轴

P►Ry(), 直角 y 坐标轴  
PAR, 参变量  
part(), 提取子表达式  
PassErr, 传递错误  
Pause, 暂停  
Permutations, 组合  
pixel, 像素  
PlotsOff, 绘图关闭  
PlotsOn, 绘图开启  
POL, 极坐标  
polyEval(), 多项式求值  
PopUp, 弹出式菜单  
PowerReg, 幂回归  
Pretty Print, 清楚显示, 优美打印  
Prgm, 执行程序  
product(), 积  
Prompt(), 提示  
propFrac, 真分数  
PtChg, 点反转  
PtOff, 点关闭  
PtOn, 点开启  
ptTest(), 点测试  
PtText, 将文本置于指定点  
PxlChg, 像素反转  
PxlCrcl, 以像素点作圆  
PxlHorz, 像素水平线  
PxlLine, 像素直线  
PxlOff, 像素关闭  
PxlOn, 像素开启  
pxlTest(), 像素测试  
PxlText, 将文本置于指定像素点  
PxlVert, 像素垂直线

## Q

QR, QR 因数分解  
QuadReg, 二次回归  
QuartReg, 二次回归  
Quit (2nd [QUIT]), 退出  
QWERTY 标准键盘

## R

r, radian, 弧度

R►Pθ(), 极坐标

R►Pr(), 极坐标

RAD, 弧度

rand(), 随机数

randMat(), 随机矩阵

randNorm(), 返回一个符合指定正态分布的随机十进制小数

randPoly(), 随机多项式

RandSeed, 随机数种子

RclGDB, 调用图形数据库

RclPic, 调用图形

real(), 实数

reciprocal, 倒数

ref(), 行梯形矩阵

remain(), 余数

Rename, 重命名

Request, 要求

Return, 返回

right(), 返回数组中最右边起指定数量的元素构成的子数组

rotate(), 循环移位

round(), 四舍五入

rowAdd(), 矩阵中某行加到另一行

rowDim(), 矩阵行维数

rowNorm(), 矩阵行范数

rowSwap(), 矩阵行交换

RplcPic, 替换图形

rref(), 最简行梯形矩阵

## S

sec(), 正割

sec<sup>-1</sup>(), 反正割

sech(), 双曲正割

sech<sup>-1</sup>(), 反双曲正割

Send, 发送数组变量

SendCalc, 发送至计算器

SendChat, 发送会话

seq(), 序列

setDate(), 设定日期

setDtFmt(), 设定日期格式

setFold(), 设定文件夹

setGraph(), 设定图形

setMode(), 设定模式

setTable(), 设定表格

setTime(), 设定时间

setTmFmt(), 设定时间格式

setTmZn(), 设定时区

setUnits(), 设定单位

Shade, 阴影

shift(), 转换

ShowStat, 显示统计结果

sign(), 正负数标记

simult(), 联立方程

sin(), 正弦

sin<sup>-1</sup>(), 反正弦

sinh(), 双曲正弦

sinh<sup>-1</sup>(), 双曲反正弦

SinReg, 正弦曲线回归

SocialSt, 社会学类别

solve(), 求解

SortA, 升序排列

SortD, 降序排列

startTmr(), 启动计时器

stdDev(), 标准偏差

StoGDB, 存储图形数据库

Stop, 停止

StoPic, 存储图形

string(), 简化表达式并以字符串形式返回

subMat(), 子矩阵

sum(), 合计

switch(), 窗口切换

## T

T, transpose, 转置  
Table, 创建表格  
tan(), 正切  
tan<sup>-1</sup>(), 反正切  
tanh(), 双曲正切  
tanh<sup>-1</sup>(), 双曲反正切  
taylor(), 泰勒多项式  
tCollect(), 以倍角、和差角表示的三角变换  
tExpand(), 三角公式展开  
Text, 文本  
Then, then 语句  
timeCnv(), 转换时间  
TI-Presenter 视频演示器  
Title, 标题  
tmpCnv(), 温度转换  
toolbar menus, 工具条菜单  
tracing, 显迹  
Try, try 语句

## U

Unarchiv, 解除(变量)存档  
unitV(), 单位向量  
Unlock, 解锁

## V

variance(), 方差

## W

When(), when 语句  
While(), while 语句  
Wire and Contour, 网和等高线  
WIRE FRAME, 网状框架

## X

xor, 布尔“异或”  
XorPic, 布尔“异或图”

## Y

Y= Editor, 绘图编辑器

## Z

zeros(), 使表达式为零的自变量值  
ZoomBox, 按指定框缩放  
ZoomData, 按数据区域缩放  
ZoomDec, 按指定十进制刻度缩放  
ZoomFit, 适当缩放  
ZoomIn, 缩小  
ZoomInt, 整数坐标缩放  
ZoomOut, 放大  
ZoomPrev, 回到上一状态的大小  
ZoomRcl, 缩放调用  
ZoomSqr, 同刻度缩放  
ZoomStd, 标准缩放  
ZoomSto, 缩放存储  
ZoomTrig, 对称缩放

