



# TI-83 Plus

图象计算器

## 参考手册



© 1999 Texas Instruments Incorporated.

<sup>SM</sup> TI, TI-GRAPH LINK, Calculator-Based Laboratory, CBL, Calculator-Based Ranger 和 CBR 是 Texas Instruments Incorporated 的商标。

RSA Data Security, Inc 专有许可的 U.S.专利号为 4,405,829

IBM 是 International Business Machines Corporation 的注册商标

Macintosh 是 Apple Computer, Inc. 的注册商标

## 重要信息

**Texas Instruments** 仅以“概不保证”方式提供程序和手册，对它们没有任何形式的明确的和默许的担保，同样也不包含（但不限于）用于任何特定目的销售和适用的默许担保。

在任何情况下 **Texas Instruments** 对因购买或使用这些资料所造成的任何特殊的、连带的、意外的、间接的损失承担责任，但是，不管损失如何，**Texas Instruments** 应负的责任不会超过本产品的购买价格。此外，**Texas Instruments** 不对任何形式的其他团体使用这些资料相异的声明承担责任。

本手册描述如何使用 **TI-83 Plus** 图象计算器。入门是 **TI-83 Plus** 功能的概述。第 1 章讲述 **TI-83 Plus** 如何运行。其他章节讲述各种交互功能。第 17 章说明如何综合利用这些功能解决问题。

TI-83 Plus 的特殊功能 .....	ix
入门：第一步！	
TI-83 Plus 键盘 .....	2
TI-83 Plus 菜单 .....	4
起始步骤 .....	5
输入计算式：二次公式 .....	6
转换为分数：二次公式 .....	7
显示复数结果：二次公式 .....	8
定义函数：带盖箱 .....	9
定义数值运算表：带盖箱 .....	10
运算表细化：带盖箱 .....	11
设置观察窗口：带盖箱 .....	12
显示和跟踪图形：带盖箱 .....	13
图形细化：带盖箱 .....	15
求计算的最大值：带盖箱 .....	16
TI-83 Plus 其它功能 .....	17
<b>第 1 章：运行 TI-83 Plus</b>	
开启和关闭 TI-83 Plus .....	1-2
设置显示对比度 .....	1-3
显示 .....	1-4
输入表达式和指令 .....	1-6
TI-83 Plus 编辑键 .....	1-8
设置方式 .....	1-9
使用 TI-83 Plus 变量名 .....	1-13
存储变量值 .....	1-15
调用变量值 .....	1-16
ENTRY（最后输入）存储区 .....	1-17
Ans（最后结果）存储区 .....	1-19
TI-83 Plus 菜单 .....	1-20
VARs 和 VARs Y-VARS 菜单 .....	1-22
Equation Operating System (EOS™) .....	1-23
出错情况 .....	1-25

**第 2 章：数学、角度和关系运算**

入门：抛掷硬币 .....	2-2
键盘数学运算 .....	2-3
MATH 运算 .....	2-5
使用方程求解器 .....	2-8
MATH NUM (数值) 运算 .....	2-13
输入和使用复数 .....	2-16
MATH CPX (复数) 运算 .....	2-18
MATH PRB (概率) 运算 .....	2-20
ANGLE 运算 .....	2-23
TEST (关系) 运算 .....	2-25
TEST LOGIC (布尔) 运算 .....	2-26

**第 3 章：函数图象**

入门：画圆 .....	3-2
定义图象 .....	3-3
设置图象方式 .....	3-4
定义函数 .....	3-5
选择和取消选择函数 .....	3-7
设置函数图象样式 .....	3-9
设置观察窗口变量 .....	3-11
设置图象格式 .....	3-13
显示图象 .....	3-15
用自由移动光标研究图象 .....	3-17
用 TRACE 研究图象 .....	3-18
用 ZOOM 指令研究图象 .....	3-20
使用 ZOOM MEMORY .....	3-23
使用 CALC (计算) 运算 .....	3-25

**第 4 章：参数图象**

入门：小球的轨迹 .....	4-2
定义和显示参数图象 .....	4-4
研究参数图象 .....	4-7

**第 5 章：极坐标图象**

入门：极坐标玫瑰线 .....	5-2
定义和显示极坐标图象 .....	5-3
研究极坐标图象 .....	5-6

---

## 第 6 章：序列图象

入门：森林和树木 .....	6-2
定义和显示序列图象 .....	6-3
选定坐标轴组合 .....	6-8
研究序列图象 .....	6-9
作 Web 图 .....	6-11
用 Web 图举例说明收敛性 .....	6-12
作相位图 .....	6-13
比较 TI-83 Plus 和 TI-82 序列变量 .....	6-15
TI-83 Plus 和 TI-82 的击键差别 .....	6-16

## 第 7 章：运算表

入门：函数的根 .....	7-2
设置运算表 .....	7-3
定义因变量 .....	7-4
显示运算表 .....	7-5

## 第 8 章：DRAW 指令

入门：画切线 .....	8-2
使用 DRAW 菜单 .....	8-3
清除图形 .....	8-4
画线段 .....	8-5
画水平线和垂直线 .....	8-6
画切线 .....	8-8
画函数和反函数 .....	8-9
在图象上遮蔽区域 .....	8-10
画圆 .....	8-11
在图象上放置文本 .....	8-12
在图象上使用画笔画图 .....	8-13
在图象上画点 .....	8-14
画像素 .....	8-16
存储图象图形 (Pic) .....	8-17
重新调用图象图形 (Pic) .....	8-18
存储图象数据库 (GDB) .....	8-19
重新调用图象数据库 (GDB) .....	8-20

## 第 9 章：拆分屏幕

入门：研究单位圆 .....	9-2
使用拆分屏幕 .....	9-3
Horiz（水平）拆分屏幕 .....	9-4
G-T（图象—运算表）拆分屏幕 .....	9-5
Horiz 和 G-T 方式下的 TI-83 Plus 象素 .....	9-6

**第 10 章：矩阵**

入门：线性方程组 .....	10-2
定义矩阵 .....	10-3
观察和编辑矩阵元素 .....	10-4
在表达式中使用矩阵 .....	10-7
显示和拷贝矩阵 .....	10-8
对矩阵进行数学运算 .....	10-9
使用 MATRX MATH 运算 .....	10-12

**第 11 章：数组**

入门：生成序列 .....	11-2
命名数组 .....	11-3
存储和显示数组 .....	11-4
输入数组名 .....	11-6
将公式与数组名相连 .....	11-7
在表达式中使用数组 .....	11-9
LIST OPS 菜单 .....	11-10
LIST MATH 菜单 .....	11-17

**第 12 章：统计**

入门：摆锤长度与周期 .....	12-2
建立统计分析 .....	12-10
使用统计数组编辑器 .....	12-11
将公式与数组名相连 .....	12-14
拆离数组名中的公式 .....	12-16
切换统计数组编辑器环境 .....	12-17
统计数组编辑器环境 .....	12-18
STAT EDIT 菜单 .....	12-20
回归模型功能 .....	12-22
STAT CALC 菜单 .....	12-24
统计变量 .....	12-29
程序中的统计分析 .....	12-30
统计绘图 .....	12-31
程序中的统计绘图 .....	12-37

**第 13 章：推理统计与分布**

入门：总体平均身高 .....	13-2
推理统计编辑器 .....	13-6
STAT TESTS 菜单 .....	13-9
推理统计输入描述 .....	13-26
检验和区间输出变量 .....	13-28
分布函数 .....	13-29
遮蔽分布 .....	13-35

## 第 14 章：应用

Applications 菜单 .....	14-2
入门：投资汽车 .....	14-3
入门：计算复利 .....	14-4
使用 TVM Solver .....	14-5
使用金融函数 .....	14-6
计算货币的时间价值 (TVM) .....	14-7
计算现金流转 .....	14-9
计算分期付款 .....	14-10
换算利息 .....	14-13
计算日期期间的天数/定义支付方法 .....	14-14
使用 TVM 变量 .....	14-15
CBL/CBR 应用程序 .....	14-16
数据采集方式和选项 .....	14-18
开始数据采集 .....	14-25
停止数据采集 .....	14-26

## 第 15 章：CATALOG、字符串和双曲函数

浏览 TI-83 Plus CATALOG .....	15-2
输入和使用字符串 .....	15-3
将字符串保存到字符串变量中 .....	15-4
CATALOG 中的字符串函数和指令 .....	15-6
CATALOG 中的双曲函数 .....	15-10

## 第 16 章：程序设计

入门：圆柱体积 .....	16-2
创建和删除程序 .....	16-4
输入命令行和执行程序 .....	16-6
编辑程序 .....	16-7
拷贝和重命名程序 .....	16-8
PRGM CTL (控制) 指令 .....	16-9
PRGM I/O (输入/输出) 指令 .....	16-17
将其他程序当作子程序调用 .....	16-23
运行汇编语言程序 .....	16-24

## 第 17 章：实例

使用框图比较实验结果 .....	17-2
绘制分段函数 .....	17-4
绘制不等式 .....	17-5
求解非线性方程组 .....	17-6
使用程序创建 Sierpinski 三角形 .....	17-7
绘制蛛网吸引点 .....	17-8
使用程序估计系数 .....	17-9
绘制单位圆和三角曲线 .....	17-10
求曲线间面积 .....	17-11
使用参数方程：转轮问题 .....	17-12
演示微积分基本原理 .....	17-14
计算正 N 边形的面积 .....	17-16
计算和绘图抵押贷款支付 .....	17-18

**第 18 章：内存和变量管理**

检查可用内存 .....	18-2
从内存中删除项目 .....	18-4
清除输入项和数组元素 .....	18-5
重新设置 TI-83 Plus .....	18-6
归档和取消归档变量 .....	18-10
组合和取消组合变量 .....	18-13
碎片收集 .....	18-17
ERR:ARCHIVE FULL 消息 .....	18-20

**第 19 章：通讯链接**

入门：发送变量 .....	19-2
TI-83 Plus 链接 .....	19-4
选择发送项 .....	19-6
接收项 .....	19-7
传送项 .....	19-8
将项目传送到其他 TI-83 Plus .....	19-10
将项目传送到 TI-83 .....	19-11
传送数组到 TI-82 .....	19-12
从 TI-82 传送到 TI-83 Plus .....	19-13
从 TI-83 Plus 向 TI-73 发送数据 .....	19-14
从 TI-73 向 TI-83 Plus 发送数据 .....	19-15
备份内存 .....	19-16

**附录 A：表格和参考信息**

函数和指令表 .....	A-2
TI-83 Plus 菜单图 .....	A-39
变量 .....	A-49
统计公式 .....	A-50
金融公式 .....	A-55

**附录 B：通用信息**

电池信息 .....	B-2
疑难解答 .....	B-4
出错情况 .....	B-6
精度信息 .....	B-11
TI 产品维修和保修信息 .....	B-13

**索引**



### 快速刷新 — 升级能力



TI-83 Plus 使用“快速刷新”技术，该技术允许您不必购买新的计算器就可以升级软件版本。

---

**详细信息，请参阅：**  
第 19 章

---

凭借这一新功能，您可以从 Internet 更新 TI-83 Plus。升级的软件版本包括维护版和新的应用程序以及升级版。其中维护版是免费的，而新的应用程序和升级版则可以从 TI 网站购买，其网址为：[education.ti.com](http://education.ti.com)

### 184K 字节内存

TI-83 Plus 内置 184K 字节内存。其中 RAM（随机存取内存）大约占有 24K，它们用于计算以及存储函数、程序和数

---

**详细信息，请参阅：**  
第 18 章

---

据。大约 160K 的用户数据归档区允许您将数据、程序、应用程序或任何其他变量存储在一个安全位置，以免它们被用户在无意中编辑或删除。您也可以通过将变量归档到用户数据区来释放 RAM。

### 应用程序

可以安装应用程序来定制 TI-83 Plus，以使其满足您的工作需要。充足的 160K 归档区空间允许您一次最多可以存储十个 (10) 应用程序。也可以将应用程序存储在计算机上，以便日后使用或用于设备之间的链接。

---

**详细信息，请参阅：**  
第 18 章

---

### 归档

可以将变量存储在 TI-83 Plus 的用户数据归档区，它是从 RAM 中分离出来的一个保护区域。使用用户数据归档区，您可以：

---

**详细信息，请参阅：**  
第 18 章

---

- 将数据、程序、应用程序或任何其他变量存储在一个安全位置，以免它们被用户在无意中编辑或删除。
- 通过归档变量来释放 RAM，以产生额外的 RAM。

通过归档不需要经常编辑的变量，您可以释放 RAM 以将其用于那些需要额外内存的应用程序。

---

## Calculator-Based Laboratory™ (CBL™) 和 Calculator-Based Ranger™ (CBR™)

TI-83 Plus 已经安装了 CBL/CBR 应用程序。如果连接了（可选）CBL 或 CBR 附件，那么您就可以使用 TI-83 Plus 分析各种实际数据。

---

**详细信息，请参阅：**  
第 14 章

---

CBL 和 CBR 允许您使用从所执行的操作中收集的数据来研究长度、速度、加速度和时间之间精确而又科学的关系。

CBL 和 CBR 的区别是：CBL 允许您使用几种不同的探测器（可以分析温度、光、电压类型或者声速数据）；而 CBR 使用内置的 Sonic 探测器收集数据。可以将 CBL 和 CBR 附件链接在一起以同时收集多种类型的数据。您可以在 CBL 和 CBR 的用户手册中获得详细信息。

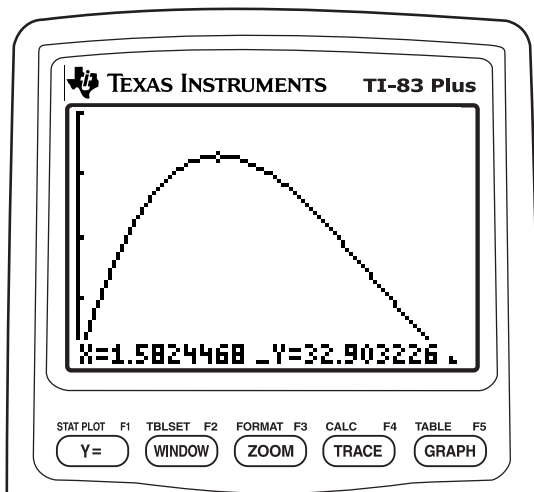
## TI-83 Plus 参考手册 的应用样例

二次公式	页 6
带盖箱	页 9
抛掷硬币	页 2-2
画圆	页 3-2
小球的轨迹	页 4-2
极坐标玫瑰线	页 5-2
森林和树木	页 6-2
用 Web 图举例说明收敛性	页 6-12
食肉动物 — 猎物模型 (相位图)	页 6-13
函数的根	页 7-2
画切线	页 8-2
研究单位圆	页 9-2
求解线性方程组	页 10-2
生成序列	页 11-2
摆锤长度与周期	页 12-2
阿拉斯加白昼的小时数	页 12-28
总体平均身高	页 13-2
投资汽车	页 14-3
计算复利	页 14-4
计算未付的贷款余额	页 14-11
使用 CBL/CBR 应用程序	页 14-16
圆柱体积 (程序)	页 16-2
使用框图比较实验结果	页 17-2
绘制分段函数	页 17-4
绘制不等式	页 17-5
求解非线性方程组	页 17-6
Sierpinski 三角形 (程序)	页 17-7
绘制蛛网吸引点	页 17-8
估计系数 (程序)	页 17-9
绘制单位圆和三角曲线	页 17-10
求曲线间面积	页 17-11
参数方程: 转轮问题	页 17-12
微积分基本原理	页 17-14
计算正 N 边形面积	页 17-16
抵押贷款支付的计算和绘图	页 17-18
发送变量 (使用通讯链接)	页 19-2

# 入门： 第一步！

## 目录

TI-83 Plus 键盘.....	1
TI-83 Plus 菜单.....	4
起始步骤.....	5
输入计算式：二次公式.....	6
转换为分数：二次公式.....	7
显示复数结果：二次公式.....	8
定义函数：带盖箱.....	9
定义数值运算表：带盖箱.....	10
运算表细化：带盖箱.....	11
设置观察窗口：带盖箱.....	12
显示和跟踪图形：带盖箱.....	13
图形细化：带盖箱.....	15
求计算的最大值：带盖箱.....	16
TI-83 Plus 其他功能.....	17



键盘一般分成以下区域：图形键、编辑键、高级功能键和科学计算器键。

## 键盘区

图形键访问交互图形功能。

编辑键用于编辑表达式和数值。

高级功能键显示访问高级功能的菜单。

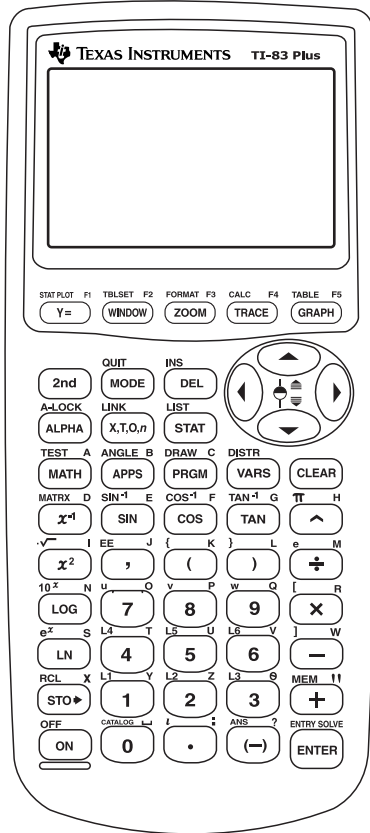
科学计算器键访问标准科学计算器的功能。

图形键

编辑键

高级  
功能键

科学计算  
器键



## 使用着色键盘

为便于找到所需键，对 TI-83 Plus 上的键盘进行了着色。

淡灰色键为数字键，键盘右侧的蓝色键为常用数学函数键，键盘顶部的蓝色键用于设置和显示图形。按蓝色键 **[APPS]** 可访问应用程序（例如，“金融”应用程序）。

每个键的基本功能印在键上。例如，按 **[MATH]**，显示 MATH 菜单。

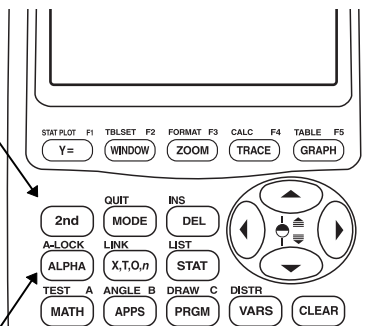
## 使用 **[2nd]** 和 **[ALPHA]** 键

每个键的第二功能以黄色印在键的上方。当按下黄色 **[2nd]** 键后，以黄色印在其他键上方的字符、缩写或单词对于下一次击键激活。例如，按 **[2nd]** 后按 **[MATH]**，显示 TEST 菜单。本指南把这种击键组合称作 **[2nd] [TEST]**。

每个键的字母函数以绿色印在键的上方。按绿色 **[ALPHA]** 键后，以绿色印在其他键上方的字母对于下一次击键激活。例如，按 **[ALPHA]** 后按 **[MATH]**，字母 **A** 输入。本指南把这种击键组合称作 **[ALPHA] [A]**。

**[2nd]** 键访问以黄色印在每个键上方的第二功能。

**[ALPHA]** 键访问以绿色印在每个键上方的字母函数。



## 显示菜单

在使用 TI-83 Plus 时，需经常访问菜单的菜单项。

按下显示菜单的键后，菜单暂时覆盖工作区屏幕。  
例如按 **[MATH]**，MATH 菜单以全屏幕方式显示。

```
5+9
```

```
MATH NUM CPX PRB
1: 1/2 Frac
2: 1/10 Dec
3: 1/3
4: 1/4 J(
5: *J
6: fMin(
7: fMax(
```

选择菜单项后，工作区屏幕通常重新显示出来。

```
5+9^3
```

## 从一个菜单移到另一个菜单

一些键访问多个菜单。按这样的键后，在顶行显示所有能访问的菜单名。亮显某个菜单名后，则显示此菜单的菜单项。按 **[▶]** 和 **[◀]** 来亮显菜单名。

```
MATH NUM CPX PRB
1: abs(
2: round(
3: iPart(
4: fPart(
5: int(
6: min(
7: max(
```

## 选择菜单项

当前菜单项旁边的数字或字母亮显。如果菜单超出屏幕，所显示的最后一个菜单项用向下箭头(↓)替代冒号(:)。如果菜单项滚动显示到最后，所显示的第一个菜单项用向上箭头(↑)替代冒号。可以用以下两种方法之一选择菜单项：

- 按 **[▶]** 或 **[◀]** 移动光标到菜单项的数字或字母，然后按 **[ENTER]**。
- 按表示菜单项旁边数字或字母的键或组合键。

```
MATH NUM CPX PRB
1: abs(
2: round(
3: iPart(
4: fPart(
5: int(
6: min(
7: max(
```

```
MATH NUM CPX PRB
3: iPart(
4: fPart(
5: int(
6: min(
7: max(
8: lcm(
9: cD(
```

## 不作选择退出菜单

可以用以下三种方法之一不作选择退出菜单：

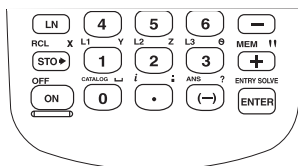
- 按 **[CLEAR]** 返回以前屏幕。
- 按 **[2nd] [QUIT]** 返回主屏幕。
- 按表示另一个菜单或屏幕的键。

```
5+9^3
```

在开始本章的样例问题之前，遵循本页步骤重设 TI-83 Plus 以将 RAM 内存清零。这样可保证本章描述的击键产生图示结果。

执行以下步骤重新设置 TI-83 Plus:

1. 按 **ON** 打开计算器。
2. 按下并放开 **2nd**，然后按 **[MEM]**（在 **+** 上方）。



如果按 **2nd**，则在下次按键时，将获得该键上方以黄色印刷的操作。**[MEM]** 是 **+** 键的 **2nd** 操作。

显示 MEMORY 菜单。



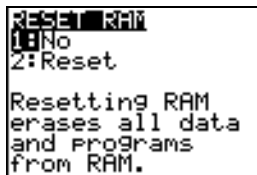
3. 按 **7** 选择 **7:Reset**。

显示 RAM ARCHIVE ALL 菜单。



4. 按 **1** 显示 RESET RAM 菜单。

5. 按 **2** 选择 **2:Reset**。

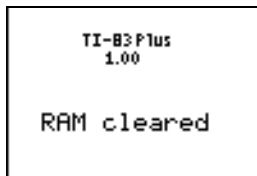


RAM 内存清零。

6. 按回车键显示主屏幕。

当重设 TI-83 Plus 后，显示对比度也被重设。

- 若屏幕非常亮，按下并放开 **2nd**，然后按住 **▲** 加黑屏幕。
- 若屏幕非常暗，按下并放开 **2nd**，然后按住 **▼** 加亮屏幕。

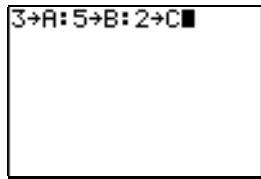




## 输入计算式：二次公式

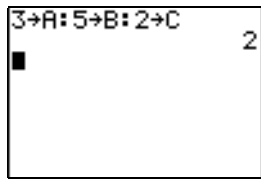
使用二次公式求解二次方程  $3X^2 + 5X + 2 = 0$  和  $2X^2 - X + 3 = 0$ 。先由方程  $3X^2 + 5X + 2 = 0$  开始。

1. 按 **3** **[STO]** **[ALPHA]** **[A]** (在 **[MATH]** 上方) 存储  $X^2$  项的系数。
2. 按 **[ALPHA]** **[:]** (在 **[□]** 上方)。冒号允许在一行输入多条指令。
3. 按 **5** **[STO]** **[ALPHA]** **[B]** (在 **[APPS]** 上方) 存储  $X$  项的系数。按 **[ALPHA]** **[:]** 在同一行上输入新指令。  
按 **2** **[STO]** **[ALPHA]** **[C]** (在 **[PRGM]** 上方) 存储常数项。



3->A:5->B:2->C

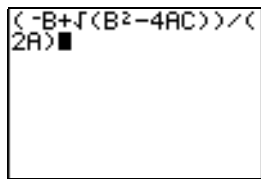
4. 按 **[ENTER]** 将输入数值存储到变量 A、B 和 C 中。  
最后输入的存储值显示在显示区右侧。光标下移一行，准备接受下次输入。



3->A:5->B:2->C 2

5. 按 **[□]** **[(-)]** **[ALPHA]** **[B]** **[+]** **[2nd]** **[√]** **[ALPHA]** **[B]** **[x<sup>2</sup>]** **[□]** **4** **[ALPHA]** **[A]** **[ALPHA]** **[C]** **[□]** **[□]** **[+/-]** **[□]** **2** **[ALPHA]** **[A]** **[□]**  
输入二次公式一个解的表达式，

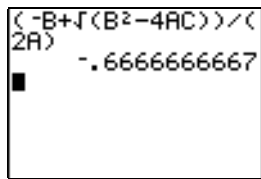
$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



$(-B + \sqrt{B^2 - 4AC}) / (2A)$

6. 按 **[ENTER]** 寻求方程  $3X^2 + 5X + 2 = 0$  的一个解。

解显示在显示区右侧。光标下移一行，准备接受下一个表达式的输入。



$(-B + \sqrt{B^2 - 4AC}) / (2A)$  -.6666666667

## 转换为分数：二次公式

可用分数形式显示解。

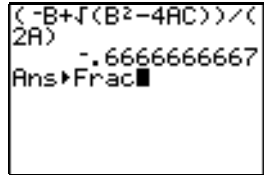
1. 按  $\boxed{\text{MATH}}$  显示 MATH 菜单。



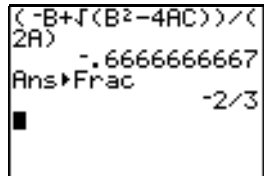
2. 按 **1** 从 MATH 菜单选择 **1**→Frac。

当按下 **1** 时，**Ans**→Frac 显示在主屏幕中。

**Ans** 是一个变量，包含上次计算的解。



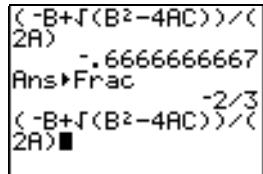
3. 按  $\boxed{\text{ENTER}}$  转换结果为分数形式。



为减少击键，可以再次调用已输入的表达式，进行编辑后作为新的计算式。

4. 按  $\boxed{2\text{nd}}$   $\boxed{\text{ENTRY}}$  (在  $\boxed{\text{ENTER}}$  上方) 调出分数转换输入，然后再按  $\boxed{2\text{nd}}$   $\boxed{\text{ENTRY}}$  调出二次表达式，

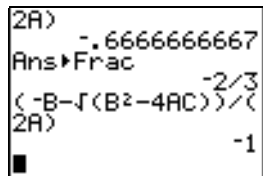
$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$



5. 按  $\boxed{\leftarrow}$  移动光标到公式中 **+** 符号，按  $\boxed{\text{EDIT}}$  编辑二次表达式为：

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

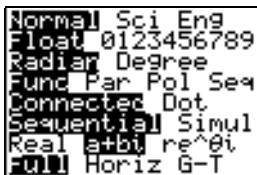
6. 按  $\boxed{\text{ENTER}}$  求二次方程  $3X^2 + 5X + 2 = 0$  的另一个解。



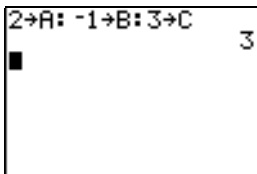
## 显示复数结果：二次公式

现在求解方程  $2X^2 - X + 3 = 0$ 。在设置 **a+bi** 复数方式后，TI-83 Plus 显示复数结果。

- 按 **[MODE]** **[<]** **[<]** **[<]** **[<]** **[<]** (6次)，然后按 **[>]** 定位光标在 **a+bi** 上。按 **[ENTER]** 选择 **a+bi** 复数方式。



- 按 **[2nd]** **[QUIT]** (在 **[MODE]** 上方) 返回主屏幕，然后按 **[CLEAR]** 清除主屏幕。

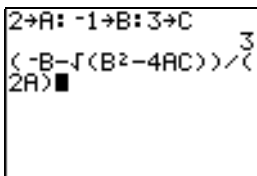


- 按 **2** **[STO>]** **[ALPHA]** **[A]** **[ALPHA]** **[:]** **[<]** **1** **[STO>]** **[ALPHA]** **[B]** **[ALPHA]** **[:]** **3** **[STO>]** **[ALPHA]** **[C]** **[ENTER]**。

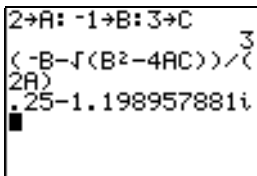
新方程  $X^2$  项的系数、 $X$  项的系数和常数项分别存储于 A、B 和 C。

- 按 **[2nd]** **[ENTRY]** 调出存储指令，然后再按 **[2nd]** **[ENTRY]** 调出二次表达式，

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



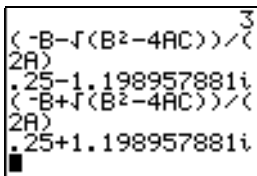
- 按 **[ENTER]** 求方程  $2X^2 - X + 3 = 0$  的一个解。



- 重复按 **[2nd]** **[ENTRY]** 直到显示该二次表达式：

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

- 按 **[ENTER]** 求方程：  $2X^2 - X + 3 = 0$  的另一个解。



**注意：** 求解方程实数解的另一种方法是使用内置 Equation Solver (第2章)。

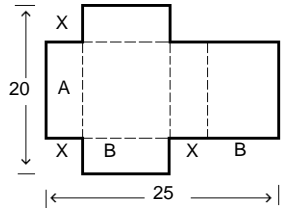
# 定义函数：带盖箱

取一张  $20\text{ cm} \times 25\text{ cm}$  的纸，然后在两角剪掉  $X \times X$  的方形，从另外两角剪掉  $X \times 12.5\text{ cm}$  的矩形，如下图所示。折叠纸成一个带盖子的箱体，则  $X$  为何值时箱体体积  $V$  最大？可使用运算表和绘图解决此问题。

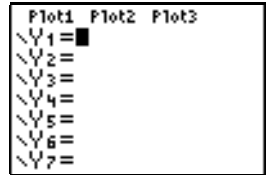
首先定义一个描述箱体体积的函数。

由图：  
 $2X + A = 20$   
 $2X + 2B = 25$   
 $V = A \cdot B \cdot X$

代入：  
 $V = (20 - 2X)(25/2 - X)X$

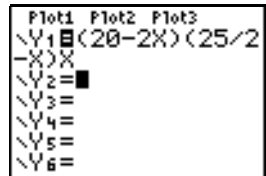


- 按  $\boxed{Y=}$  显示  $Y=$  编辑器，用于定义运算表和绘图的函数。



- 按  $\boxed{\text{20}} \boxed{=} \boxed{2} \boxed{X.T.O.n} \boxed{)} \boxed{(} \boxed{25} \boxed{=} \boxed{2} \boxed{X.T.O.n} \boxed{)} \boxed{X.T.O.n} \boxed{\text{ENTER}}$  用  $X$  来定义体积函数  $Y_1$ 。

$\boxed{X.T.O.n}$  允许快速输入  $X$ ，而不必按  $\boxed{\text{ALPHA}}$ 。  
 $=$  符号亮显表示选择了  $Y_1$ 。



TI-83 Plus 的运算表功能显示函数的数值信息。可以使用第 9 页定义的函数数值运算表来估算问题的解。

- 按 **2nd** [TBLSET] (在 **WINDOW** 上方) 显示 TABLE SETUP 菜单。

TABLE SETUP	
TblStart=0	
ΔTbl=1	
Indnt:	<b>AUTO</b> Ask
Depend:	<b>AUTO</b> Ask

- 按 **ENTER** 接受 **TblStart=0**。

- 按 **1** **ENTER** 定义运算表步长 **ΔTbl=1**。保留 **Indpnt: Auto** 和 **Depend: Auto** 以便运算表自动生成。

- 按 **2nd** [TABLE] (在 **GRAPH** 上方) 显示运算表。

注意到 **Y1** (箱体体积) 的极大值是在当 **X** 大约为 **4**，即在 **3** 到 **5** 之间时取得。

X	Y1	
0	0	
1	207	
2	336	
3	388	
4	408	
5	375	
6	312	

X=0

- 按住 **▼** 滚动运算表直到显示 **Y1** 的负值。

注意到此问题中 **X** 的最大长度是在 **Y1** (箱体体积) 符号从正号变到负号时取得，即在 **10** 和 **11** 之间。

X	Y1	
6	312	
7	231	
8	144	
9	63	
10	0	
11	-33	
12	-84	

X=12

- 按 **2nd** [TBLSET]。

注意到 **TblStart** 已变为 **6** 来映射运算表上次显示的第一行。(在步骤 5，运算表中显示 **X** 的第一个值为 **6**。)

TABLE SETUP	
TblStart=6	
ΔTbl=1	
Indnt:	<b>AUTO</b> Ask
Depend:	<b>AUTO</b> Ask

可调整运算表显示方式获得已定义函数的更多信息。赋予  $\Delta Tbl$  一个较小值可以使运算表显示细化。

- 按  $3$   $\square$   $\text{ENTER}$  设置  $TblStart$ 。按  $\square$   $1$   $\square$   $\text{ENTER}$  设置  $\Delta Tbl$ 。

这样调整运算表来获得对于最大体积  $Y_1$  的  $X$  的更准确估算。

TABLE SETUP	
TblStart=3	
$\Delta Tbl=.1$	
Indent: <b>Auto</b>	Ask
Depend: <b>Auto</b>	Ask

- 按  $2$   $\square$   $\text{nd}$  [TABLE]。
- 按  $\square$  和  $\square$  滚动运算表。

注意到  $Y_1$  最大值为 **410.26**，是取在  $X=3.7$  时。因此，取最大值条件为  $3.6 < X < 3.8$ 。

X	Y <sub>1</sub>	
3.6	410.11	
3.7	410.26	
3.8	409.94	
3.9	409.19	
4	408	
4.1	406.39	
4.2	404.38	

X=4.2

- 按  $2$   $\square$   $\text{nd}$  [TBLSET]。按  $3$   $\square$   $6$   $\square$   $\text{ENTER}$  设置  $TblStart$ 。按  $\square$   $01$   $\square$   $\text{ENTER}$  设置  $\Delta Tbl$ 。

TABLE SETUP	
TblStart=3.6	
$\Delta Tbl=.01$	
Indent: <b>Auto</b>	Ask
Depend: <b>Auto</b>	Ask

- 按  $2$   $\square$   $\text{nd}$  [TABLE]，然后按  $\square$  和  $\square$  滚动运算表。

显示四个相同的最大值 **410.26**，分别取在  $X=3.67$ 、 $3.68$ 、 $3.69$  和  $3.70$  处。

X	Y <sub>1</sub>	
3.66	410.25	
3.67	410.26	
3.68	410.26	
3.69	410.26	
3.7	410.26	
3.71	410.26	
3.72	410.23	

X=3.72

- 按  $\square$  或  $\square$  移动光标到 **3.67**。按  $\square$  移动光标到  $Y_1$  列。

$Y_1$  在  $X=3.67$  处的精确值显示在底行上，为 **410.261226**。

X	Y <sub>1</sub>	
3.66	410.25	
3.67	410.26	
3.68	410.26	
3.69	410.26	
3.7	410.26	
3.71	410.26	
3.72	410.23	

Y<sub>1</sub>=410.261226

- 按  $\square$  显示其他最大值。

$Y_1$  在  $X=3.68$  处的精确值为 **410.264064**、在  $X=3.69$  处为 **410.262318**、在  $X=3.7$  时为 **410.256**。

如果能够以 .01 厘米精确测量并剪切纸的话，箱体体积的最大值将在剪切边长为 **3.68** 时取得。

X	Y <sub>1</sub>	
3.66	410.25	
3.67	410.26	
3.68	410.26	
3.69	410.26	
3.7	410.26	
3.71	410.23	
3.72	410.23	

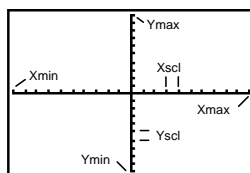
Y<sub>1</sub>=410.264064

也可使用 TI-83 Plus 的绘图功能求上面所定义函数的最大值。当图形激活后，观察窗口会定义坐标面的显示区。窗口变量值决定观察窗口的大小。

1. 按 **WINDOW** 显示窗口编辑器，在此可观察和编辑窗口变量的值。

```
WINDOW
Xmin=-10
Xmax=10
Xscl=1
Ymin=-10
Ymax=10
Yscl=1
Xres=1
```

标准窗口变量定义观察窗口如图所示。**Xmin**、**Xmax**、**Ymin** 和 **Ymax** 定义显示边界。**Xscl** 和 **Yscl** 定义 **X** 和 **Y** 轴标记距离，**Xres** 控制分辨率。



2. 按 **0** **ENTER** 定义 **Xmin**。
3. 按 **20** **÷** **2** 使用表达式定义 **Xmax**。

```
WINDOW
Xmin=0
Xmax=20/2
Xscl=1
Ymin=-10
Ymax=10
Yscl=1
Xres=1
```

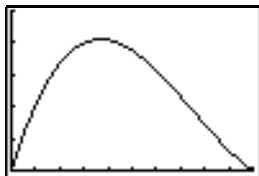
4. 按 **ENTER**。计算表达式，结果 **10** 存于 **Xmax**。按 **ENTER** 确认 **Xscl** 为 **1**。
5. 按 **0** **ENTER** **500** **ENTER** **100** **ENTER** **1** **ENTER** 定义其余窗口变量。

```
WINDOW
Xmin=0
Xmax=10
Xscl=1
Ymin=0
Ymax=500
Yscl=100
Xres=1
```

既然已经定义用于绘图的函数和窗口，就可以显示并仔细研究图形。可以使用 TRACE 功能跟踪函数。

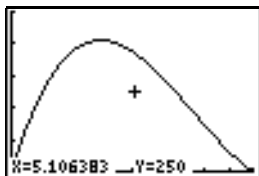
1. 按 **GRAPH** 在观察窗口绘制所选函数的图形。

显示  $Y_1=(20-2X)(25/2-X)X$  的图形。



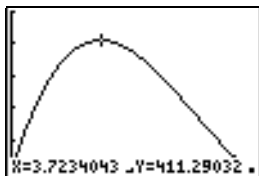
2. 按 **▢** 激活自由移动图形光标。

图形光标位置的 **X** 和 **Y** 坐标值显示在底行。



3. 按 **◀**、**▶**、**▲** 和 **▼** 移动自由光标到函数最大值的大概位置。

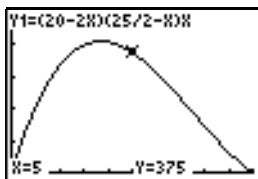
随着光标的移动，**X** 和 **Y** 坐标值连续更新。





4. 按 **[TRACE]**。跟踪光标显示在函数 **Y1** 上。

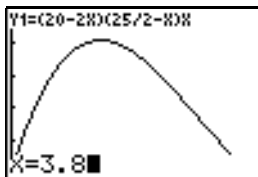
正在跟踪的函数显示在左上角。



5. 按 **[←]** 和 **[→]** 沿 **Y1** 跟踪，一次一个 **X** 点，在每个 **X** 点处计算 **Y1** 的值。

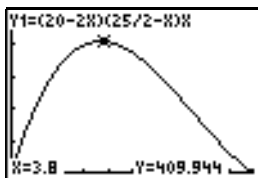
也可输入对应最大值的 **X** 估算值。

6. 按 **3** **[.]** **8**。在 **TRACE** 期间，按下数字键时，**X=** 提示显示在左下角。



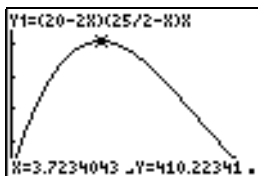
7. 按 **[ENTER]**。

跟踪光标跳到 **Y1** 函数在 **X=3.8** 点处的值。



8. 按 **[←]** 和 **[→]** 直到 **Y** 取最大值。

这就是 **Y1(X)** 对应于 **X** 像素值的最大值。实际精确最大值可能位于像素值之间。



为便于识别函数的最大值、最小值、根和交点，可以在指定位置使用 ZOOM 指令放大观察窗口。

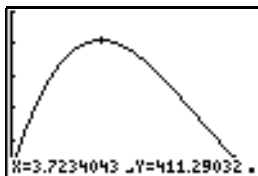
1. 按 **ZOOM** 显示 ZOOM 菜单。

此菜单为典型 TI-83 Plus 菜单。要选择菜单项，可按菜单项旁边的数字或字母；或者按 **▾** 直到菜单项的数字或字母亮显，然后按 **ENTER**。

```
ZOOM MEMORY
1:ZBox
2:Zoom In
3:Zoom Out
4:ZDecimal
5:ZSquare
6:ZStandard
7↓ZTrig
```

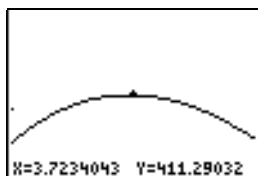
2. 按 **2** 选择 **2:Zoom In**。

图形再次显示。光标已变，表示在使用 ZOOM 指令。



3. 当光标接近函数的最大值处时（如第 14 页的步骤 8），按 **ENTER**。

新观察窗口显示。Xmax-Xmin 和 Ymax-Ymin 都以缩放因子的默认值 4 进行调整。



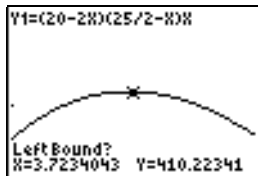
4. 按 **WINDOW** 显示新窗口的设置。

```
WINDOW
Xmin=2.4734042...
Xmax=4.9734042...
Xscl=1
Ymin=348.79032...
Ymax=473.79032...
Yscl=100
Xres=1
```

可以使用 CALCULATE 菜单操作来计算函数的局部最大值。

- 按  $\boxed{2nd} \boxed{[CALC]}$  (在  $\boxed{[TRACE]}$  上方) 显示 CALCULATE 菜单。按 **4** 选择 **4:maximum**。

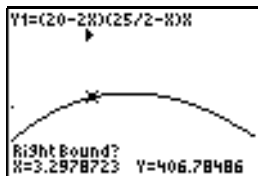
图形再次显示并带有 **Left Bound?** 提示。



- 按  $\boxed{\blacktriangleright}$  跟踪曲线到最大值左面的点，然后按  $\boxed{[ENTER]}$ 。

屏幕顶部的  $\blacktriangleright$  表示选择的边界。

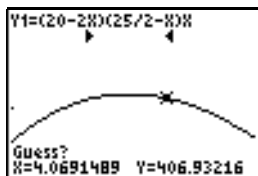
显示 **Right Bound?** 提示。



- 按  $\boxed{\blacktriangleleft}$  跟踪曲线到最大值右面的点，然后按  $\boxed{[ENTER]}$ 。

屏幕顶部的  $\blacktriangleleft$  表示选择的边界。

显示 **Guess?** 提示。

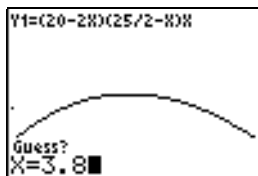


- 按  $\boxed{\blacktriangleleft}$  跟踪到最大值附近一点，然后按  $\boxed{[ENTER]}$ 。

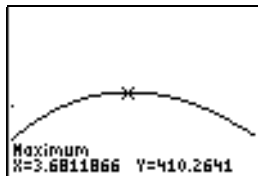
或者按 **3**  $\boxed{.}$  **8**，然后按  $\boxed{[ENTER]}$  输入最大值的估计值。

当在 TRACE 里按下数字键时，**X=** 提示显示在左下角。

注意计算的最大值与用自由移动光标、跟踪光标和运算表求得的最大值有何不同。



**注意：**在上面的步骤 2 和 3，可以用步骤 4 描述的方法直接输入 Left Bound 和 Right Bound 的值。



入门部分已经介绍了基本的 TI-83 Plus 操作。此手册详细地描述入门中使用的功能，它也包含 TI-83 Plus 的其他特征和功能。

### 绘图

可以存储、绘图和分析多至 10 个函数（第 3 章）、6 个参数函数（第 4 章）、6 个极坐标函数（第 5 章）和多至 3 个序列（第 6 章）。可以使用 DRAW 指令注释图形（第 8 章）。

### 序列

可以产生序列并把它们绘成对时间的图形。或者可以绘成 web 图或相位图（第 6 章）。

### 运算表

可以创建函数运算表来同时分析多个函数（第 7 章）。

### 拆分屏幕

可以水平拆分屏幕用以显示图形和相应的编辑器（如 Y= 编辑器）、运算表、统计数组编辑器或主屏幕。也可以垂直拆分屏幕用以同时显示图形和它的运算表（第 9 章）。

### 矩阵

可以输入并存储多至 10 个矩阵，可以对它们进行标准矩阵运算（第 10 章）。

### 数组

在统计分析中可以输入并存储内存允许的尽可能多的数组。可以把公式与数组相连以自动进行计算。可以使用数组同时对多个数值进行表达式计算和绘制曲线族（第 11 章）。

### 统计

可以进行基于数组的一元、二元统计分析，包括逻辑和正弦回归分析。可以绘制数据的直方图、xy 直线图、散射图、改进或标准的箱线图、或正态概率图。可以定义或存储多至三个统计图形（第 12 章）。

### 推理统计

可以执行 16 个假设检验和置信区间、15 个分布函数。可以以图形或数字方式显示假设检验结果 (第 13 章)。

### 应用程序

可以使用诸如“金融”、Calculator-Based Laboratory™ (CBL™) 或 Calculator-Based Ranger™ (CBR™) 等应用程序。通过“金融”应用程序，可以使用货币时间价值 (TVM) 函数来分析金融业务，如：年金、贷款、抵押、租赁和储蓄等；可以使用现金流函数来分析货币在经过相同时间周期后的价值；可以用分期付款函数计算分期付款问题。通过 CBL/CBR 应用程序及 CBL 或 CBR (可选) 附件，可以利用各种探测器采集真实的数据 (第 14 章)。

### CATALOG

CATALOG 为 TI-83 Plus 上所有函数和指令的便捷、以字母顺序排列的列表。可以从 CATALOG 粘贴任意函数或指令到当前光标处 (第 15 章)。

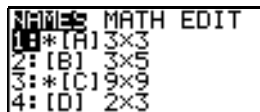
### 程序设计

可以输入并存储包含多方面控制及输入/输出指令的程序 (第 16 章)。

### 归档

归档操作可以在用户数据归档区 (该区域的变量不会被用户在无意中编辑或删除) 存储数据、程序或其他变量，还可以为需要额外内存的变量释放 RAM 内存。

在变量名称的左边标记星号 (\*) 来显示已归档的变量 (第 16 章)。



A screenshot of the TI-83 Plus calculator screen showing a list of variables and their dimensions. The screen is divided into three columns: NAMES, MATH, and EDIT. The variables listed are: 1: \*(A) 3x3, 2: (B) 3x5, 3: \*(C) 9x9, and 4: (D) 2x3. The asterisk (\*) indicates that variables A, C, and D are archived.

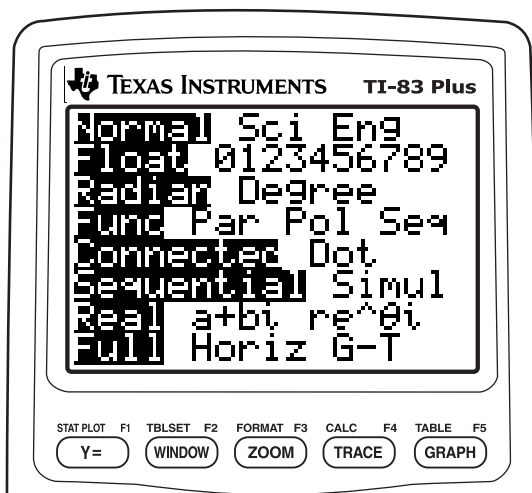
### 通讯链接

TI-83 Plus 配备有端口，用于与其他 TI-83 Plus、TI-83、TI-82、TI-73、Calculator-Based Laboratory™ (CBL™) 系统或 Calculator-Based Ranger™ (CBR™) 系统连接和通讯。为此 TI-83 Plus 附带一根单元—单元链接电缆 (第 19 章)。通过 TI-GRAPH LINK™ (单独销售)，可以将 TI-83 Plus 连接到个人计算机上。TI web 站点提供了未来的升级软件，用户可从该站点将软件下载到个人计算机，并使用 TI-GRAPH LINK™ 来升级 TI-83 Plus。

# 1 运行 TI-83 Plus

## 目录

开启和关闭 TI-83 Plus.....	2
设置显示对比度.....	3
显示.....	4
输入表达式和指令.....	6
TI-83 Plus 编辑键.....	8
设置方式.....	9
使用 TI-83 Plus 变量名.....	13
存储变量值.....	15
调用变量值.....	16
ENTRY（最后输入）存储区.....	17
Ans（最后结果）存储区.....	19
TI-83 Plus 菜单.....	20
VARS 和 VARS Y-VARS 菜单.....	22
Equation Operating System (EOS™).....	23
出错情况.....	25



## 开启计算器

要开启 TI-83 Plus，按 **[ON]**。

- 如果以前已按 **[2nd] [OFF]** 关闭了计算器，TI-83 Plus 显示上次使用时的主屏幕并清除任何错误。
- 如果以前是 Automatic Power Down™ (APD™) 已经关闭了计算器，TI-83 Plus 将以关闭前的方式显示，包括显示内容、光标和任何错误。
- 如果 TI-83 Plus 处于关闭状态并将它连接到其他计算器或个人电脑上，那么当连接完成时，TI-83 Plus 将会被“唤醒”。
- 如果 TI-83 Plus 处于关闭状态并已连接到其他计算器或个人计算机上，那么任何通讯活动都会“唤醒”TI-83 Plus。

为延长电池寿命，如果在五分钟内无任何动作，APD 将自动关闭 TI-83 Plus。

## 关闭计算器

要手工关闭 TI-83 Plus，按 **[2nd] [OFF]**。

- 所有设置和内存内容由 Constant Memory™ 保持。
- 所有错误均被清除。

## 电池

TI-83 Plus 使用四节 AAA 碱性电池和一个用户可更换的备份锂电池（CR1616 或 CR1620）。要更换电池而不丢失存储在内存中的任何信息，请按照附录 B 的步骤进行。

## 调节显示对比度

可以调节显示对比度来适合您的视角和光照条件。在改变对比度设置时，在右上角有一数字表明当前的级别，可从 **0**（最亮）到 **9**（最暗）调节。如果对比度太亮或太暗可能看不到此数字。

**注意：** TI-83 Plus 有 40 种对比度设置，因此每个从 **0** 到 **9** 之间的数字代表四种设置。

TI-83 Plus 在关闭后仍在内存中保留对比度设置。

执行以下步骤调节对比度：

1. 按下并释放 **[2nd]** 键。
2. 按住 **[<]** 或 **[>]**，两键分别位于对比度符号（黄色、半阴影圆形）下方和上方。
  - **[<]** 加亮屏幕。
  - **[>]** 加暗屏幕。

**注意：** 如果调节对比度设置为 **0**，显示可能完全变成空白。为恢复屏幕，按下并释放 **[2nd]**，然后按住 **[>]** 直到显示出现。

## 更换电池时间

当电池电量不足时，在下列情况下将会显示电量不足的信息：

- 打开计算器。
- 下载新的应用程序。
- 试图升级为新软件。

为在更换电池时不丢失内存中的任何信息，请按照附录 B 的步骤进行。

计算器一般在第一次显示电量不足后可继续运行一到两个星期。这一时间过后，TI-83 Plus 自动关闭且不再运行。此时必须更换电池。所有内存内容应保留。

**注意：** 如果不是频繁地使用计算器，第一次出现电量不足信息后，电池的寿命可能超过两个星期。



## 显示类型

TI-83 Plus 既可显示文本又可显示图象。第 3 章描述图象。第 9 章描述 TI-83 Plus 如何使用水平或垂直拆分屏幕同时显示图象和文本。

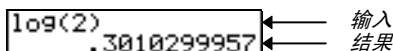
## 主屏幕

主屏幕是 TI-83 Plus 的基本屏幕。在此屏幕上，输入要执行的指令和要计算的表达式。结果也在这里显示。

## 显示输入和结果

TI-83 Plus 屏幕可以最多显示 8 行文本，每行最多 16 个字符。如果所有显示行充满，文本向上滚动以便于显示。如果主屏幕、Y= 编辑器（第 3 章）、程序编辑器（第 16 章）上的表达式长度大于一行，则它自动换行到下一行起始处。在数字编辑器，如窗口屏幕（第 3 章）中，长表达式左右滚动显示。

当输入在主屏幕上被执行后，结果显示在下一行的右侧。



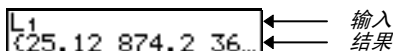
TI-83 Plus 屏幕显示：

```
log(2)
.3010299957
```

← 输入  
← 结果

方式设置控制 TI-83 Plus 解释表达式和显示结果的方式（页 1-9）。

如果结果（例如数组或矩阵）太长，在一行显示不下时，则在结果的右侧或左侧显示省略号 (...)。按  $\rightarrow$  和  $\leftarrow$  显示结果。



TI-83 Plus 屏幕显示：

```
L1
(25.12 874.2 36...
```

← 输入  
← 结果

## 返回主屏幕

按  $2^{nd}$  [QUIT] 从其他任何屏幕返回主屏幕。

## 忙指示器

TI-83 Plus 正在计算或绘图时，一个垂直移动直线作为忙指示器显示在屏幕右上角。当暂停一个图形或程序时，忙指示器变成垂直移动的散点线。

---

## 显示光标

许多情况下，光标形状表明在按下下一个键或者选择将其粘贴为字符的下一菜单项时将要发生的动作。

光标	形状	下一击键的作用
输入	实矩形条 ■	在当前光标处输入一字符；替换已有字符
插入	下划线 —	在光标位置前插入一字符
第二功能	反箭头 ⏮	输入第二功能字符（键盘上黄色的）或执行第二功能运算
大写字母	反 A Ⓐ	输入大写字母（键盘上绿色的）或执行 SOLVE
充满	棋盘状矩形 ■	不能输入，在提示处已输入最大字符数或内存已满

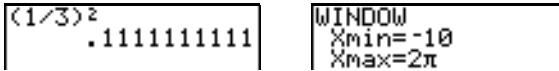
在插入时如果按 **[ALPHA]**，光标变为带下划线的 A (**A**)；在插入时按 **[2nd]**，下划线光标变为带下划线的 ↑ (**↑**)。

在图形和编辑器里有时显示其他形状光标，这将在其他章节里介绍。

## 何为表达式？

表达式是指一组数字、变量、函数及其参数、或这些元素的组合。一个表达式只有一个输出值。在 TI-83 Plus 里，可按书写顺序输入表达式。例如： $\pi R^2$  是一个表达式。

可以使用主屏幕里的表达式计算求值。在大部分需要数值的位置，都可以使用表达式作为输入值。



## 输入表达式

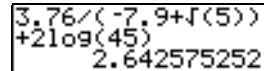
为建立表达式，可从键盘和菜单输入数字、变量和函数。当按下 **ENTER** 后，就认为表达式输入完成而不管光标位置如何。根据 **Equation Operating System (EOS™)** 规则（页 1-23）计算整个表达式，然后显示结果。

大部分 TI-83 Plus 函数和运算都用由几个字符组成的符号表示。必须从键盘或菜单输入这种符号而非拼写输入。例如：为计算 45 的对数值，必须按 **LOG** 45，而不是输入字母 L、O 和 G。如果输入了 LOG，TI-83 Plus 解释为输入变量 L、O 和 G 的蕴含乘法。

---

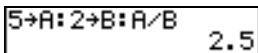
计算  $3.76 \div (-7.9 + \sqrt{5}) + 2 \log 45$ 。

3 **□** **76** **□** **÷** **□** **(** **□** **(** **□** **7** **□** **9** **□** **+** **□** **2nd** **□** **[√]** **□** **)** **□** **)**  
**□** **+** **□** **2** **□** **LOG** **□** **45** **□** **)**  
**□** **ENTER**



## 一行中多项输入

为在一行中输入两个或更多的表达式或指令，用冒号 (**ALPHA** [:]) 分隔它们。所有指令存储在最后输入项里 (ENTRY；页 1-17)。

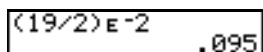


---

## 输入以科学记数法表示的数

执行以下步骤输入以科学记数法表示的数：

1. 输入指数前的部分。可以是表达式。
2. 按  $\boxed{2nd}$  [ $EE$ ]。E 粘贴到光标处。
3. 若指数为负，按  $\boxed{\ominus}$ ，然后输入指数，可以是一位或两位。

The image shows a TI-83 Plus calculator screen. The display is divided into two sections. The top section shows the expression  $(19/2)E-2$  and the bottom section shows the decimal value  $.095$ . The entire display is enclosed in a rectangular border.

输入以科学记数法表示的数后，TI-83 Plus 并不自动以科学记数法或工程记数法显示结果。方式设置（页 1-9）和数字的大小决定显示格式。

## 函数

函数返回一个值。例如： $\div$ 、 $-$ 、 $+$ 、 $\sqrt{\quad}$ （和  $\log(\quad)$  都是页 1-6 示例中的函数。一般地，在 TI-83 Plus 里函数的第一个字母小写。大部分函数带有至少一个参数，表示为函数名后有一左括号  $(\quad)$ 。例如： $\sin(\quad)$  需要一个参数， $\sin(\text{value})$ 。

## 指令

指令启动一个操作。例如：**ClrDraw** 是一条指令用于清除图象上所有绘出的元素。指令不能用在表达式里。一般地，指令的第一个字母大写。一些指令的参数多于一个，表示为指令名后有一个左括号  $(\quad)$ 。例如：**Circle**（需要三个参数，**Circle(X,Y,radius)**）。

## 中断计算

要中断进行中的计算或绘图（由忙指示器表示），按  $\boxed{ON}$ 。

中断计算后，显示菜单。

- 要返回主屏幕，选择 **1:Quit**。
- 要转至中断处，选择 **2:Goto**。

中断绘图后，显示部分图形。

- 要返回主屏幕，按  $\boxed{CLEAR}$  或任意非图形键。
- 要重新绘图，按图形键或选择绘图指令。

击键	结果
▸ 或 ◀	在表达式中移动光标；按住这些键重复此操作。
▲ 或 ▼	在占用多行的表达式中将光标从一行移到另一行；按住这些键重复此操作。 在主屏幕中表达式的顶行，▲ 移动光标到表达式开始处。 在主屏幕中表达式的底行，▼ 移动光标到表达式结束处。
2nd ◀	移动光标到表达式开始处。
2nd ▸	移动光标到表达式结束处。
ENTER	计算表达式或执行指令。
CLEAR	在主屏幕中的文本行上，清除当前行。 在主屏幕中的空白行上，清除主屏幕上所有显示。 在编辑器中，清除光标当前位置的表达式或值；不存储零值。
DEL	删除光标处字符；按住此键重复此操作。
2nd [INS]	改变光标为下划线 ( _ )；在下划光标前插入字符；要结束插入，按 2nd [INS] 或按 ◀、▲、▸ 或 ▼。
2nd	改变光标为 <b>■</b> ；下次击键执行 2nd 运算（键左上方的黄色运算），要撤消 2nd，再按 2nd。
ALPHA	改变光标为 <b>■</b> ；下次击键粘贴大写字母（键右上方的绿色字符）或执行 SOLVE（第 10 和 11 章）；要撤消 ALPHA，按 ALPHA 或按 ◀、▲、▸ 或 ▼。
2nd [A-LOCK]	改变光标为 <b>■</b> ；设置字母锁；随后击键（击大写字母键）粘贴大写字母；要撤消字母锁，按 ALPHA；如果出现输入名称提示（例如，输入群组或程序名称），那么将自动设置字母锁。
X.T.θ.n	击键时在 <b>Func</b> 方式下粘贴 <b>X</b> 、在 <b>Par</b> 方式下粘贴 <b>T</b> 、在 <b>Pol</b> 方式下粘贴 <b>θ</b> 、或在 <b>Seq</b> 方式下粘贴 <b>n</b> 。

# 设置方式

## 检查方式设置

方式设置控制 TI-83 Plus 如何显示和解释数字和图形。方式设置在TI-83 Plus 关闭后由“恒定内存”功能保持。所有数值，包括矩阵和数组的元素，都根据当前方式设置显示。

要显示方式设置，按 **[MODE]**。当前设置亮显。下表中的默认值亮显。随后几页详细描述方式设置。

Normal	Sci Eng	数字记数法
Float	0123456789	小数位数
Radian	Degree	角度单位
Func	Par Pol Seq	作图类型
Connected	Dot	是否连接图象点
Sequential	Simul	是否同时绘图
Real	$a+bi$ $re^{\theta i}$	实数、直角坐标复数、极坐标复数
Full	Horiz G-T	整屏、两种拆分屏幕方式

## 改变方式设置

执行以下步骤改变方式设置：

1. 按 **[↓]** 或 **[↑]** 移动光标到要改变设置的行。
2. 按 **[▶]** 或 **[◀]** 移动光标到要改变的设置
3. 按 **[ENTER]**。

## 由程序设置方式

可通过将方式名作为指令输入在程序中设置方式；例如：**Func** 或 **Float**。在一空白程序命令行，选择方式屏幕中的方式设置，指令粘贴到光标处。

```
PROGRAM:TEST
:Func█
```

### Normal、Sci、Eng

记数法方式仅影响结果在主屏幕的显示。数值结果可显示多至 10 位和一个两位的指数。可以以任何格式输入数字。

**Normal** 记数法方式是表示数字通常所用的方法，小数点左边是整数位，右边是小数位，如 **12345.67**。

**Sci** (科学) 记数法方式用两部分表示数字。有效数位的小数点左侧为一位，E 的右侧是 10 的幂，如 **1.234567E4**。

**Eng** (工程) 记数法方式和科学记数法相似。但不同之处在于，在小数位前可有一位、两位或三位整数，底数是 10，指数为 3 的倍数，如 **12.34567E3**。

**注意：**如果选择 **Normal** 记数法，而结果不能显示为 10 位（或绝对值小于 .001），则 TI-83 Plus 以科学记数法表示结果。

### Float、0123456789

**Float** (浮点) 十进制方式最多显示 10 位，外加符号和小数点。

**0123456789** (定点) 十进制方式指定小数点后显示的位数（0 到 9）。将光标放在所要求的小数位上，然后按 **ENTER**。

十进制表示设置适用于 **Normal**、**Sci** 和 **Eng** 记数法方式。

十进制表示设置适用于下面的数：

- 主屏幕显示的结果
- 图形上的坐标（第 3、4、5 和 6 章）
- **Tangent**( **DRAW** 指令直线方程、**x** 和 **dy/dx** 的值（第 8 章）
- **CALCULATE** 运算结果（第 3、4、5 和 6 章）
- 回归模型执行后存储的回归方程（第 12 章）

---

## **Radian 、 Degree**

角度方式控制 TI-83 Plus 如何解释三角函数和极坐标/平面直角坐标转换中的角度值。

**Radian** 方式以弧度解释角度值。结果以弧度方式显示。

**Degree** 方式以度解释角度值。结果以度方式显示。

## **Func 、 Par 、 Pol 、 Seq**

图象方式定义绘图参数。在第 3、4、5 和 6 章中将详细描述这些方式。

**Func**（函数）绘图方式作函数图，**Y** 是 **X** 的函数（第 3 章）。

**Par**（参数）绘图方式作关系图，**X** 和 **Y** 都是 **T** 的函数（第 4 章）。

**Pol**（极坐标）绘图方式作函数图，**r** 是 **θ** 的函数（第 5 章）。

**Seq**（序列）绘图方式作序列图（第 6 章）。

## **Connected 、 Dot**

**Connected** 绘图方式对每个由所选函数计算出的点用线连接。

**Dot** 绘图方式只绘制由所选函数计算的点。



### Sequential、Simul

**Sequential** 顺序绘图方式在完成一个函数的计算和作图后再对下一个函数进行计算和作图。

**Simul** (同时) 顺序绘图方式对单个 **X** 计算并绘制其所有选定函数的图象, 然后对下一个 **X** 值进行相同的操作。

**注意:** 不论选择哪种作图方式, TI-83 Plus 先顺序绘制所有统计图, 然后再绘制函数的图象。

### Real、 $a+bi$ 、 $re^{\theta i}$

**Real** 方式不显示复数结果, 除非输入的是复数。

显示复数结果有两种方式。

- **$a+bi$**  (直角坐标复数方式) 以  $a+bi$  格式显示复数。
- **$re^{\theta i}$**  (极坐标复数方式) 以  $re^{\theta i}$  格式显示复数。

### Full、Horiz、G-T

**Full** 屏幕方式使用整屏显示图象或编辑屏幕。

每种拆分屏幕方式同时显示两个屏幕。

- **Horiz** (水平) 方式在屏幕的上半屏显示当前图形; 在下半屏显示主屏幕或某个编辑器 (第 9 章)。
- **G-T** (图形 — 表) 方式在左半屏显示当前图形, 右半屏显示表 (第 9 章)。

## 变量和定义项

在 TI-83 Plus 可以输入和使用多种类型的数据，包括实数、复数、矩阵、数组、函数、统计图、图象数据库、图象图形和字符串。

TI-83 Plus 使用指定的变量名和其他存储在内存的项。对于数组，也可以创建五个字符的数组名。

变量类型	名称
实数	<b>A</b> 、 <b>B</b> 、...、 <b>Z</b>
复数	<b>A</b> 、 <b>B</b> 、...、 <b>Z</b>
矩阵	[A]、[B]、[C]、...、[J]
数组	<b>L1</b> 、 <b>L2</b> 、 <b>L3</b> 、 <b>L4</b> 、 <b>L5</b> 、 <b>L6</b> 和用户定义的名称
函数	<b>Y1</b> 、 <b>Y2</b> 、...、 <b>Y9</b> 、 <b>Y0</b>
参数方程	<b>X1T</b> 和 <b>Y1T</b> 、...、 <b>X6T</b> 和 <b>Y6T</b>
极坐标函数	<b>r1</b> 、 <b>r2</b> 、 <b>r3</b> 、 <b>r4</b> 、 <b>r5</b> 、 <b>r6</b>
序列函数	<b>u</b> 、 <b>v</b> 、 <b>w</b>
统计图	<b>Plot1</b> 、 <b>Plot2</b> 、 <b>Plot3</b>
图象数据库	<b>GDB1</b> 、 <b>GDB2</b> 、...、 <b>GDB9</b> 、 <b>GDB0</b>
图象图形	<b>Pic1</b> 、 <b>Pic2</b> 、...、 <b>Pic9</b> 、 <b>Pic0</b>
字符串	<b>Str1</b> 、 <b>Str2</b> 、...、 <b>Str9</b> 、 <b>Str0</b>
应用程序	Applications
应用程序变量	Application variables
群组	Grouped variables
系统变量	<b>Xmin</b> 、 <b>Xmax</b> 和其他

### 关于变量的注意事项

- 只要内存允许，可以创建尽可能多的数组名（第 11 章）。
- 程序有用户定义的名字，并和变量共享内存（第 16 章）。
- 在主屏幕或程序里，变量可以存储到矩阵（第 10 章）、数组（第 11 章）、字符串（第 15 章）、如 **Xmax**（第 1 章）和 **TblStart**（第 7 章）之类的系统变量、以及所有 **Y=** 函数（第 3、4、5、6 章）。
- 在编辑器里，变量可以存储到矩阵、数组和 **Y=** 函数（第 3 章）。
- 在主屏幕、程序、编辑器里，数值可以存储到矩阵或数组元素。
- 可以使用 **DRAW STO** 菜单项存储和调用图象数据库和图形（第 8 章）。
- 虽然大多数变量可以归档，但是系统变量（包括 **r**、**t**、**x**、**y** 和  **$\theta$** ）不能归档（第 18 章）。
- **Apps** 是存储在 **Flash ROM** 中的独立应用程序。**AppVars** 是变量占位符，用于存储由独立的应用程序创建的变量。用户不能编辑或更改 **AppVars** 中的变量，而只能在创建这些变量的应用程序中进行编辑或更改。

## 存储变量值

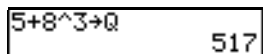
---

### 存储变量值

通过使用变量名，可以存储值，也可以从内存调用值。当计算包含变量名的表达式时，使用变量的当前值。

要在主屏幕或程序里使用 **[STO▶]** 键将值存储到某个变量，以空行开始，然后执行以下步骤：

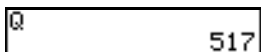
1. 输入需存储的数值。可以是表达式。
2. 按 **[STO▶]**。→ 复制到光标处。
3. 按 **[ALPHA]** 后按要存储数值的变量字母。
4. 按 **[ENTER]**。如果输入表达式，则进行计算，结果存储到变量中。



A calculator screen with a black border. The top line shows the expression  $5+8^3\rightarrow Q$  and the bottom line shows the result 517.

### 显示变量值

为显示某个变量的值，在主屏幕的空行上输入变量名，然后按 **[ENTER]**。



A calculator screen with a black border. The top line shows the variable name Q and the bottom line shows the result 517.

### 归档变量

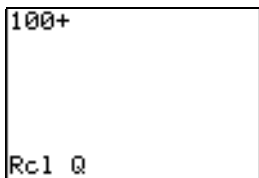
可以在名为用户数据归档区的内存区域（该区域的数据不会被用户在无意中编辑或删除）中归档数据、程序或者其他变量。在变量名的左边会出现一个星号 (\*) 来标识已归档的变量。不能编辑或执行已归档的变量，而只能对其进行查阅或取消归档操作。例如，如果您归档了数组 **L1**，那么您可以在内存中查阅到 **L1**，但是如果您选择它并将其粘贴到主屏幕上，那么您会发现：在解除归档之前将无法查阅 **L1** 变量的内容或编辑该变量。

## 使用调用 (RCL)

执行以下步骤调用和复制变量内容到当前光标位置。要退出 RCL，按 **[CLEAR]**。

1. 按 **[2nd]** **[RCL]**。Rcl 和编辑光标显示在屏幕底行。
2. 以下面五种方式之一输入变量名。
  - 按 **[ALPHA]**，然后按变量字母。
  - 按 **[2nd]** **[LIST]**，然后选择数组名，或按 **[2nd]** **[Ln]**。
  - 按 **[2nd]** **[MATRX]**，然后选择矩阵名。
  - 按 **[VARS]** 显示 VARS 菜单或按 **[VARS]** **[▶]** 显示 VARS Y-VARS 菜单后，选择变量类型，然后选择变量或函数名。
  - 按 **[PRGM]** **[◀]**，然后选择程序名（只在程序编辑器里适用）。

所选的变量名显示在底行且光标消失。



3. 按 **[ENTER]**。变量内容插入到执行这些步骤以前的光标处。



**注意：**可以编辑粘贴到表达式中的字符而不影响内存中的值。

## ENTRY（最后输入）存储区

---

### 使用 ENTRY（最后输入）

在主屏幕上按 **[ENTER]** 计算表达式或执行指令时，表达式或指令存储在一个称为 ENTRY（最后输入）的存储区域。关闭 TI-83 Plus 后，ENTRY 内容保存在内存中。

要调用 ENTRY，按 **[2nd]** **[ENTRY]**。最后输入项粘贴到当前光标位置，可以编辑和执行。在主屏幕或编辑器里，当前行内容被清除，最后输入项粘贴到本行。

由于 TI-83 Plus 只在按 **[ENTER]** 后更新 ENTRY 内容，因此，即使已经开始输入下一个表达式，仍可以调用上一个输入。

---

5 <b>[+]</b> 7 <b>[ENTER]</b> <b>[2nd]</b> <b>[ENTRY]</b>	5+7 5+7■	12
---	-------------	----

---

### 访问先前输入

TI-83 Plus 尽可能多地在 ENTRY 里保存以前的输入，最长达 128 字节。要翻看这些输入，反复按 **[2nd]** **[ENTRY]**。若单一输入多于 128 字节，仍由 ENTRY 保留它，但不能放置于 ENTRY 存储区。

---

1 <b>[STO▶]</b> <b>[ALPHA]</b> A <b>[ENTER]</b> 2 <b>[STO▶]</b> <b>[ALPHA]</b> B <b>[ENTER]</b> <b>[2nd]</b> <b>[ENTRY]</b>	1→A 2→B 2→B■	1 2
---	--------------------	--------

---

如果在显示最早存储的输入后按 **[2nd]** **[ENTRY]**，则最新存储的输入再次显示，然后是次最新的输入，等等。

---

<b>[2nd]</b> <b>[ENTRY]</b>	1→A 2→B 1→A■	1 2
-----------------------------	--------------------	--------

---

## 重复执行先前输入

在粘贴最后输入到主屏幕并编辑后（如果选择编辑），可以执行此输入。  
按 **[ENTER]** 执行输入。

为重复执行显示的输入，再按 **[ENTER]**。每次重复执行在下一行右侧显示结果，而输入本身并不重复显示。

0 <b>[STO]</b> <b>[ALPHA]</b> N	0 → N	0
<b>[ENTER]</b>		
<b>[ALPHA]</b> N <b>[+]</b> 1 <b>[STO]</b> <b>[ALPHA]</b> N <b>[ALPHA]</b> [:] <b>[ALPHA]</b> N	N + 1 → N: N <sup>2</sup>	1
<b>[x<sup>2</sup>]</b> <b>[ENTER]</b>		4
<b>[ENTER]</b>		9
<b>[ENTER]</b>		

## 在同一行多项输入

为存储两个或多个表达式或指令到 **ENTRY**，用冒号分隔每个表达式或指令后，按 **[ENTER]**。由冒号分隔的所有表达式和指令都存储到 **ENTRY**。

按 **[2nd]** **[ENTRY]** 会使所有由冒号分隔的表达式和指令粘贴到当前光标位置。可以编辑任一输入项，然后在按 **[ENTER]** 后执行所有这些输入。

对于方程  $A = \pi r^2$ ，使用试错法寻求一个面积为 200 平方厘米的圆的半径。用 8 作为第一次估计。

8 <b>[STO]</b> <b>[ALPHA]</b> R <b>[ALPHA]</b> [:] <b>[2nd]</b> <b>[π]</b> <b>[ALPHA]</b> R <b>[x<sup>2</sup>]</b>	8 → R: πR <sup>2</sup>	201.0619298
<b>[ENTER]</b>	8 → R: πR <sup>2</sup> █	
<b>[2nd]</b> <b>[ENTRY]</b>		
<b>[2nd]</b> <b>[←]</b> 7 <b>[2nd]</b> <b>[INS]</b> <b>[.]</b> 95	8 → R: πR <sup>2</sup>	201.0619298
<b>[ENTER]</b>	7.95 → R: πR <sup>2</sup>	198.5565097

继续进行直到结果达到满意精度为止。

## 清除 ENTRY

**Clear Entries**（第 18 章）将清除 TI-83 Plus 保存在 **ENTRY** 存储区的所有数据。

## Ans（最后结果）存储区

---

### 在表达式中使用 Ans

在主屏幕或程序里成功地进行表达式计算后，TI-83 Plus 保存结果到一个称为 **Ans**（最后结果）的存储区。**Ans** 可能是一个实数或复数、数组、矩阵或字符串。关闭 TI-83 Plus 后，**Ans** 中的值保留在内存。

可以在大多数位置使用变量 **Ans** 来代表最后结果。按  $\boxed{2\text{nd}} \boxed{[\text{ANS}]}$  将复制变量名 **Ans** 到光标位置。计算表达式时，TI-83 Plus 用 **Ans** 值完成计算。

计算一块 1.7 米乘 4.2 米的菜园面积。然后在番茄总产量为 147 个的条件下，计算每平方米的产量。

1  $\boxed{.}$  7  $\boxed{\times}$  4  $\boxed{.}$  2  
 $\boxed{[\text{ENTER}]}$   
147  $\boxed{\div}$   $\boxed{2\text{nd}} \boxed{[\text{ANS}]}$   
 $\boxed{[\text{ENTER}]}$

1.7*4.2	7.14
147/Ans	20.58823529

### 继续表达式

可以使用 **Ans** 作为下一表达式的第一项输入而不用再次输入值或按  $\boxed{2\text{nd}} \boxed{[\text{ANS}]}$ 。在主屏幕的一空白行上，输入函数。TI-83 Plus 粘贴变量名 **Ans** 到屏幕，然后是函数。

5  $\boxed{\div}$  2  
 $\boxed{[\text{ENTER}]}$   
 $\boxed{\times}$  9  $\boxed{.}$  9  
 $\boxed{[\text{ENTER}]}$

5/2	2.5
Ans*9.9	24.75

### 存储结果

要保存结果，在计算另一个表达式前应保存 **Ans** 的值到某一变量。

计算半径为 5 米的圆面积，接着计算半径为 5 米，高为 3.3 米的圆柱体积，然后保存结果在变量 V 里。

$\boxed{2\text{nd}} \boxed{[\pi]}$  5  $\boxed{x^2}$   
 $\boxed{[\text{ENTER}]}$   
 $\boxed{\times}$  3  $\boxed{.}$  3  
 $\boxed{[\text{ENTER}]}$   
 $\boxed{\text{STO}}$   $\boxed{[\text{ALPHA}]}$  V  
 $\boxed{[\text{ENTER}]}$

$\pi 5^2$	78.53981634
Ans*3.3	259.1813939
Ans→V	259.1813939



## 使用 TI-83 Plus 菜单

可以使用菜单完成大部分 TI-83 Plus 操作。当按某一个键或组合键来显示菜单时，在屏幕顶行出现一个或多个菜单名。

- 顶行左侧的菜单名亮显。在那个菜单里，至多显示由亮显菜单项 **1** 开始的 7 个菜单项。
- 数字或字母表示菜单里的每个菜单项位置，顺序为 **1** 到 **9**，然后是 **0**，接着为 **A**、**B**、**C** 等。LIST NAMES、PRGM EXEC 和 PRGM EDIT 菜单只标记菜单项 **1** 到 **9** 和 **0**。
- 当菜单除显示的之外还有菜单项时，向下箭头 (↓) 代替所显示的最后一项旁边的冒号。
- 当一菜单项以省略号 (...) 结尾，表示选中它时还有子菜单或编辑器。
- 如果一菜单项左边标有 (\*)，那么表示该项目存储在用户数据归档区（第 18 章）。

RAM FREE	19849
ARC FREE	147345
*[A]	83
[B]	146
▶*[C]	20
[D]	65

要显示顶行所列任何其他菜单，按 **▶** 或 **◀** 直到该菜单名亮显。菜单显示时，光标位于菜单第一项，而和最初菜单里光标位置无关。

**注意：**在附录 A 中的 Menu Map 显示了每个菜单、菜单下每个操作、用于显示菜单的键或组合键。

## 滚动菜单

按 **↓** 可以向下滚动菜单，按 **↑** 向上滚动菜单。

按 **[ALPHA] ↓** 向下一次滚动六个菜单项，按 **[ALPHA] ↑** 向上一次滚动六个菜单项。计算器上位于 **↓** 和 **↑** 之间的绿色箭头是向上和向下翻页符号。

按 **▶** 从第一项直接前进到最后一项，按 **◀** 从最后一项直接返回到第一项。

---

## 选择菜单项

可以用以下两种方式之一选择菜单项。

- 按要选择菜单项的数字或字母。光标可在菜单的任何位置，且所选菜单项不必显示在屏幕上。
- 按  $\blacktriangledown$  或  $\blacktriangleleft$  移动光标到所要菜单项，然后按 **ENTER**。

选择菜单项后，TI-83 Plus 一般显示选择前的屏幕。

**注意：**在 LIST NAMES、PRGM EXEC 和 PRGM EDIT 菜单里，只标记菜单项 1 到 9 和 0，这样按适当数字键就可选择菜单项。要移动光标到以大写字母或  $\theta$  开始的第一个菜单项，按那个大写字母或  $\theta$  的组合键。如果没有以这种字符开始的项，光标移动到下一菜单项。

---

计算  $^3\sqrt{27}$ 。



## 不作选择退出菜单

可以用以下四种方式之一退出菜单而不作选择。

- 按 **2nd** [QUIT] 返回主屏幕。
- 按 **CLEAR** 返回以前屏幕。
- 按表示另外菜单的键或组合键，如 **MATH** 或 **2nd** [LIST]。
- 按表示另外屏幕的键或组合键，如 **Y=** 或 **2nd** [TABLE]。

## VARS 菜单

可以直接在表达式中输入函数名和系统变量名或直接存储它们。

要显示 VARS 菜单，按 **[VARS]**。所有 VARS 菜单项显示子菜单，在子菜单里显示系统变量名。1:Window、2:Zoom 和 5:Statistics 每个都有多个子菜单。

---

### VARS Y-VARS

1: Window...	X/Y、T/θ 和 U/V/W 变量
2: Zoom...	ZX/ZY、ZT/Zθ 和 ZU 变量
3: GDB...	Graph database 变量
4: Picture...	Picture 变量
5: Statistics...	XY、Σ、EQ、TEST 和 PTS 变量
6: Table...	TABLE 变量
7: String...	String 变量

---

## 从 VARS 菜单或 VARS Y-VARS 菜单选择变量

要显示 VARS Y-VARS 菜单，按 **[VARS]** **[▶]**。1:Function、2:Parametric 和 3:Polar 显示 Y= 函数变量的子菜单。

---

### VARS Y-VARS

1: Function...	$Y_n$ 函数
2: Parametric...	$X_nT$ 、 $Y_nT$ 函数
3: Polar...	$r_n$ 函数
4: On/Off...	选择/取消选择函数开关

---

**注意：**序列变量 (**u**, **v**, **w**) 位于键盘上，是 **[7]**、**[8]** 和 **[9]** 键的第二功能。

若要在 VARS 或 VARS Y-VARS 菜单里选择变量，执行以下步骤：

1. 显示 VARS 或 VARS Y-VARS 菜单。
  - 按 **[VARS]** 显示 VARS 菜单。
  - 按 **[VARS]** **[▶]** 显示 VARS Y-VARS 菜单。
2. 选择变量类型，如 VARS 菜单的类型 **2:Zoom**；或选择 VARS Y-VARS 菜单的类型 **3:Polar**。子菜单显示。
3. 如果从 VARS 菜单选择了 **1:Window**、**2:Zoom** 或 **5:Statistics** 类型，可以按 **[▶]** 或 **[◀]** 显示其他子菜单。
4. 从菜单选择变量名。此变量粘贴到光标位置。

## 计算次序

Equation Operating System (EOS™) 定义在 TI-83 Plus 上输入和计算表达式中函数的次序。EOS 允许以简单明了的次序输入数字和函数。

EOS 以下面的次序计算表达式中的函数：

次序	函数
1	在参数之前的函数，如 $\sqrt{\quad}$ 、 $\sin(\quad)$ 、或 $\log(\quad)$
2	在参数后输入的函数，如 $2^{\quad}$ 、 $-1^{\quad}$ 、 $!\quad$ 、 $^{\circ}$ 、 $r$ 和换算
3	幂和根，如 $2^{\wedge}5$ 或 $5^{\wedge}\sqrt{32}$
4	排列 ( $nPr$ ) 和组合 ( $nCr$ )
5	乘法、蕴含乘法和除法
6	加法和减法
7	关系函数，如 $>$ 或 $\leq$
8	逻辑算子 <b>and</b>
9	逻辑算子 <b>or</b> 和 <b>xor</b>

在同一优先权级别，EOS 从左到右计算函数。

括号中的计算优先进行。

### 蕴含乘法

TI-83 Plus 认可蕴含乘法，因此不必在所有情形下都按 [ ] 来表达乘法。  
例如：TI-83 Plus 解释  $2\pi$ 、 $4\sin(46)$ 、 $5(1+2)$  和  $(2*5)7$  为蕴含乘法。

**注意：**TI-83 Plus 的蕴含乘法规则，尽管与 TI-83 相象，但不同于 TI-82。例如：  
TI-83 Plus 计算  $1/2X$  是以  $(1/2)*X$  方式进行，而 TI-82 计算  $1/2X$  以  $1/(2*X)$   
方式进行（第 2 章）。

### 括号

在一对括号中的任何计算首先进行。如，在表达式  $4(1+2)$  中，EOS 先计算括号中的部分  $1+2$ ，然后用结果 3 乘以 4。

$4*1+2$	6
$4(1+2)$	12

可以省略表达式最后的右括号 ( )。所有左括号中的元素在表达式结尾自动被右括号括起来。这也适合存储或显示转换指令前的开括号。

**注意：**数组名、矩阵名或 Y= 函数名后的开括号不表示蕴含乘法，而是指数组（第 11 章）、或矩阵（第 10 章）中的元素，或者指用于求解 Y= 函数的值。

### 负值

要输入负数，使用负值键。按 [ - ] 后输入数字。在 TI-83 Plus 里，取负处于 EOS 层次的第三级。第一级中的函数，如平方在取负之前计算。

例如： $-X^2$ ，计算结果为负数（或 0）。使用括号求负数的平方。

$-2^2$	-4	$2 \rightarrow R$	2
$(-2)^2$	4	$-R^2$	-4
		$(-R)^2$	4

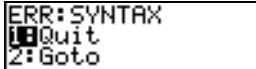
**注意：**使用 [ - ] 键作减法，用 [ - ] 键取负。如果按 [ - ] 输入负数，如在  $9 \times [ - ] 7$  中；或按 [ - ] 表示减法，如  $9 [ - ] 7$ ，在这两种情况下都会发生错误。如果按 [ALPHA] A [ ] [ALPHA] B，将被解释为蕴含乘法  $(A*B)$ 。

## 诊断错误

在执行以下操作时 TI-83 Plus 检测是否有错误发生。

- 计算表达式
- 执行指令
- 绘图
- 存储数值

一旦 TI-83 Plus 检测到错误，将以菜单标题形式返回出错信息，如 ERR:SYNTAX 或 ERR:DOMAIN。附录 B 描述每个错误类型和可能的出错原因。



```
ERR:SYNTAX
1:Quit
2:Goto
```

- 如果选择 **1:Quit**（或按 **2nd** [QUIT] 或 **CLEAR** 键），显示主屏幕。
- 如果选择 **2:Goto**，则显示先前屏幕，光标位于出错位置或附近。

**注意：**如果在程序执行时 Y= 函数的内容出现语法错误，则 **Goto** 选项返回到 Y= 编辑器，而不是程序。

## 修正错误

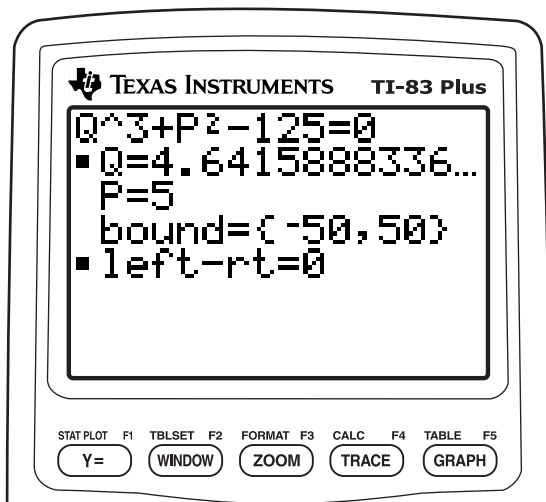
执行以下步骤来修正错误：

1. 记录错误类型 (**ERR:error type**)。
2. 选择 **2:Goto**（如果可用）。先前屏幕显示，光标位于出错位置或附近。
3. 确定错误。如果不能识别错误，请参考附录 B。
4. 修正表达式。

# 2 数学、角度 和关系运算

## 目录

入门：抛掷硬币.....	2
键盘数学运算.....	3
MATH 运算.....	5
使用方程求解器.....	8
MATH NUM（数值）运算.....	13
输入并使用复数.....	16
MATH CPX（复数）运算.....	18
MATH PRB（概率）运算.....	20
ANGLE 运算.....	23
TEST（关系）运算.....	25
TEST LOGIC（布尔）运算.....	26



## 入门：抛掷硬币

入门是对本章内容的快速介绍。详细信息请阅读本章。

假设您想建立抛掷一枚普通硬币 10 次的模型，想跟踪 10 次抛掷结果有几次是正面。并对这种模拟执行 40 次。对于一枚普通硬币，抛掷结果为正面的概率为 0.5，反面的概率也为 0.5。

1. 从主屏幕开始，按  $\boxed{\text{MATH}}$   $\boxed{\text{2}}$  显示 MATH PRB 菜单，按  $\boxed{7}$  选择 **7:randBin**(随机二项式)。**randBin**( 被粘贴到主屏幕。按  $\boxed{10}$  输入抛掷硬币的次数。按  $\boxed{.}$ 。按  $\boxed{5}$  输入正面的概率。按  $\boxed{.}$ 。按  $\boxed{40}$  输入模拟次数。按  $\boxed{\text{ENTER}}$ 。

```
randBin(10,.5,40
)
```

2. 按  $\boxed{\text{ENTER}}$  计算表达式，将产生 40 个元素的数组，并显示前 7 个元素。数组包含每组的正面次数，每组抛掷 10 次硬币。数组有 40 个元素是因为模拟进行了 40 次。在此例中，在第一组中正面为 5 次，第二组中 5 次，等等。

```
randBin(10,.5,40
)
{5 5 7 4 6 6 3 ...
```

3. 按  $\boxed{\text{D}} \text{或} \boxed{\text{F}}$  观看数组中其他次数。省略号 (...) 表示数组长度超出了屏幕。

4. 按  $\boxed{\text{STO}} \text{▶} \boxed{\text{2nd}} \boxed{\text{L1}} \boxed{\text{ENTER}}$  存储数据到数组名 **L1** 中。然后可以在另一个操作，如绘直方图中（第 12 章）使用这些数据。

```
randBin(10,.5,40
)
{5 5 7 4 6 6 3 ...
Ans→L1
{5 5 7 4 6 6 3 ...
```

**注意：**由于 **randBin**( 产生随机数，您的数组元素可能与例子中的数据不同。

```
randBin(10,.5,40
)
{5 5 7 4 6 6 3 ...
Ans→L1
...2 5 3 6 5 7 5 ...
```



## 使用数组进行数学运算

数组的数学运算返回一个数组，此数组的每个元素由参与运算数组的对应元素依次计算获得。如果在表达式中使用两个数组，则数组长度必须相同。

$$\boxed{(1, 2) + (3, 4) + 5} \\ \boxed{(9, 11)}$$

### + (加)、- (减)、\* (乘)、/ (除)

可以对实数及复数、表达式、数组和矩阵进行 + (加,  $\oplus$ )、- (减,  $\ominus$ )、\* (乘,  $\otimes$ ) 和 / (除,  $\oslash$ ) 运算，但不能对矩阵进行 / 运算。

$$\begin{array}{ll} \text{valueA} + \text{valueB} & \text{valueA} - \text{valueB} \\ \text{valueA} * \text{valueB} & \text{valueA} / \text{valueB} \end{array}$$

## 三角函数

可以对实数、表达式、数组使用三角 (trig) 函数 (正弦,  $\sin$ ; 余弦,  $\cos$ ; 和正切,  $\tan$ )。当前角度方式设置影响对角度的解释。例如,  $\sin(30)$  在 Radian 方式下返回  $-0.9880316241$ ; 在 Degree 方式下返回  $.5$ 。

$$\sin(\text{value}) \quad \cos(\text{value}) \quad \tan(\text{value})$$

也可以对实数、表达式、数组使用反三角函数 (反正弦,  $\sin^{-1}$ ; 反余弦,  $\cos^{-1}$ ; 和反正切,  $\tan^{-1}$ )。当前角度方式设置影响角度的解释。

$$\sin^{-1}(\text{value}) \quad \cos^{-1}(\text{value}) \quad \tan^{-1}(\text{value})$$

注意：三角函数不能用于复数。

### ^ (幂)、^2 (平方)、√ (开方)

可以对实数及复数、表达式、数组和矩阵进行 ^ (幂,  $\wedge$ )、^2 (平方,  $\square^2$ ) 和 √ (开方,  $\sqrt{\quad}$ ) 运算，但不能对矩阵进行 √ (运算)。

$$\text{value}^{\text{power}} \quad \text{value}^2 \quad \sqrt{\text{value}}$$

### -1 (求逆)

可以对实数及复数、表达式、数组和矩阵进行  $^{-1}$  (求逆,  $\square^{-1}$ ) 运算。乘法的逆元等价于倒数,  $1/x$ 。

$$\text{value}^{-1} \\ \boxed{5^{-1}} \quad .2$$

## log(, 10^(, ln(

可以对实数或复数、表达式和数组进行 **log(** (对数,  $\text{LOG}$ )、**10^(** (以 10 为底的幂,  $\text{10}^x$ ) 和 **ln(** (自然对数,  $\text{LN}$ ) 运算。

**log(value)**

**10^(power)**

**ln(value)**

## e^( (以 e 为底指数)

**e^(** (以 e 为底指数,  $\text{e}^x$ ) 返回以常数 **e** 为底的乘幂。可以对实数或复数、表达式和数组进行 **e^(** 运算。

**e^(power)**

$\text{e}^{(5)}$   
148.4131591

## e (常数)

**e** (常数,  $\text{e}$ ) 作为常数存储在 TI-83 Plus 里。按  $\text{e}$  复制 **e** 到光标位置。在计算中, TI-83 Plus 使用 2.718281828459 代替 **e**。

$\text{e}$   
2.718281828

## - (取负)

**-** (取负,  $\text{-}$ ) 返回 *value* 的负值。可以对实数或复数、表达式、数组和矩阵进行 **-** 运算。

**-value**

EOS 规则 (第 1 章) 确定何时取负值。例如,  $-\mathbf{A}^2$  返回一个负数, 因为在取负之前先进行平方。使用括号求负数的平方, 如  $(-\mathbf{A})^2$ 。

$2 \rightarrow \mathbf{A} : \{ -\mathbf{A}^2, (-\mathbf{A})^2, -2^2, (-2)^2 \}$   
{ -4 4 -4 4 }

注意: 在 TI-83 Plus 里, 取负符号 (-) 比在按  $\text{-}$  时显示的减符号 (-) 短且位置较高。

## $\pi$ (Pi)

$\pi$  (Pi,  $\pi$ ) 作为常数存储在 TI-83 Plus 里。在计算中, TI-83 Plus 使用 3.1415926535898 代替  $\pi$ 。

$\pi$   
3.141592654

## MATH 菜单

为显示 MATH 菜单，按 **[MATH]**。

---

MATH	NUM	CPX	PRB
1: $\blacktriangleright$ Frac			以分数显示结果。
2: $\blacktriangleright$ Dec			以小数显示结果。
3: $\text{^}3$			求立方。
4: $\text{^}3\sqrt{}$			求立方根。
5: $\text{^}x\sqrt{}$			求 $x^{\text{th}}$ 方根。
6: fMin(			求函数最小值。
7: fMax(			求函数最大值。
8: nDeriv(			求数值导数。
9: fnInt(			求函数积分。
0: Solver...			显示方程求解器。

---

### $\blacktriangleright$ Frac, $\blacktriangleright$ Dec

**$\blacktriangleright$ Frac**（以分数显示）以有理数显示结果。可以对实数或复数、表达式、数组和矩阵使用  **$\blacktriangleright$ Frac**。如果结果不能化简或结果的分母超过三位，则返回结果的小数值。只能在值 *value* 后使用  **$\blacktriangleright$ Frac**。

*value*  **$\blacktriangleright$ Frac**

**$\blacktriangleright$ Dec**（以小数显示）是以小数形式显示结果。可以对实数或复数、表达式、数组和矩阵使用  **$\blacktriangleright$ Dec**。只可在值 *value* 后使用  **$\blacktriangleright$ Dec**。

*value*  **$\blacktriangleright$ Dec**

```

1/2+1/3 $\blacktriangleright$ Frac  5/6
Ans $\blacktriangleright$ Dec
.8333333333
    
```

**3 (立方) 、  $\sqrt[3]{}$  (立方根)**

**3 (立方)** 返回值 *value* 的立方。可以使用 **3** 求实数或复数、表达式、数组和方阵的三次方。

$$value^3$$

**$\sqrt[3]{}$  (立方根)** 返回值 *value* 的立方根。可以使用  **$\sqrt[3]{}$**  (求实数或复数、表达式和数组的立方根)。

$$\sqrt[3]{(value)}$$

```
(2,3,4,5)^3
(8 27 64 125)
 $\sqrt[3]{}$ (Ans)
(2 3 4 5)
```

**$\sqrt[x]{}$  (根)**

**$\sqrt[x]{}$  ( $x^{th}$  方根)** 返回值 *value* 的  $x^{th}$  方根。可以对实数或复数、表达式和数组使用  **$\sqrt[x]{}$** 。

$$x^{th}root\sqrt[x]{value}$$

```
5 $\sqrt[3]{}$ 32
2
```

**fMin( , fMax(**

**fMin(** (函数最小值) 和 **fMax(** (函数最大值) 返回表达式 *expression* 在变量 *variable* 下限 *lower* 和上限 *upper* 区间内的最大值或最小值。**fMin(** 和 **fMax(** 在表达式 *expression* 中不适用。精度由误差 *tolerance* 控制 (如果不指定, 默认值为  $1E-5$ )。

$$fMin(expression, variable, lower, upper[, tolerance])$$

$$fMax(expression, variable, lower, upper[, tolerance])$$

**注意:** 在本指南中, 可选参数及配对的逗号用方括号 ([]) 括起。

```
fMin(sin(A), A, - $\pi$ ,  $\pi$ )
-1.570797171
fMax(sin(A), A, - $\pi$ ,  $\pi$ )
1.570797171
```

## nDeriv(

**nDeriv(** (数值导数) 返回表达式 *expression* 在指定值 *value* 处和精度  $\epsilon$  (如果不指定, 默认值为  $1E-3$ ) 下对变量 *variable* 的近似导数。**nDeriv(** 只对实数有效。

**nDeriv(expression,variable,value[, $\epsilon$ ])**

**nDeriv(** 使用对称差商方法计算, 近似地取通过这些点的割线斜率为数值导数值。

$$f'(x) = \frac{f(x+\epsilon) - f(x-\epsilon)}{2\epsilon}$$

随  $\epsilon$  变小, 近似值会更精确。

```
nDeriv(A^3,A,5,.01)
              75.0001
nDeriv(A^3,A,5,.0001)
              75
```

可以在表达式 *expression* 中使用 **nDeriv(** 一次。由于用来计算 **nDeriv(** 的方法, TI-83 Plus 在不可微分的点会返回错误导数值。

## fnInt(

**fnInt(** (函数积分) 返回表达式 *expression* 对变量 *variable* 在指定下限 *lower* 和上限 *upper* 和误差 *tolerance* 内 (如果不指定, 默认值为  $1E-5$ ) 的数值积分 (Gauss-Kronrod 方法)。**fnInt(** 只对实数有效。

**fnInt(expression,variable,lower,upper[,tolerance])**

```
fnInt(A^2,A,0,1)
.3333333333
```

**提示:** 为加快绘制积分图的速度 (当 **fnInt(** 用在  $Y=$  方程中时), 按 **GRAPH** 之前加大窗口变量 **Xres** 的值。

### Solver

**Solver** 显示方程求解器，可以用于求解一个方程中的任何变量。方程是假定等于零的。**Solver** 只对实数有效。

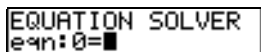
当选择 **Solver** 后，会显示以下两个屏幕之一。

- 方程编辑器（请参阅下面步骤 1 的图）在方程变量 **eqn** 为空时显示。
- 交互式求解器编辑器（请参阅页 2-9 上的步骤 3 的图）在有方程存储在 **eqn** 时显示。

### 在方程求解器中输入表达式

要在方程求解器中输入表达式，假定变量 **eqn** 为空情况下，执行以下步骤：

1. 从 MATH 菜单选择 **0:Solver**，显示方程编辑器。

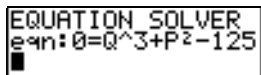


```
EQUATION SOLVER
eqn: 0=
```

2. 用以下三种方法之一输入表达式。

- 在方程求解器直接输入表达式。
- 从 VARS Y-VARS 菜单粘贴 Y= 变量名到方程求解器。
- 按  $\boxed{2nd}$   $\boxed{[RCL]}$ ，从 VARS Y-VARS 菜单粘贴 Y= 变量名，然后按  $\boxed{ENTER}$ 。表达式粘贴到方程求解器。

表达式输入后，存储于变量 **eqn** 中。



```
EQUATION SOLVER
eqn: 0=Q^3+P^2-125
```

## 在方程求解器中输入表达式 (续)

- 按 **ENTER** 或  $\square$ 。显示交互式求解器编辑器。

```
Q^3+P^2-125=0
Q=0
P=0
bound=(-1E99,1...
```

- 存储在 **eqn** 里的方程设为零并显示在顶行。
- 方程中的变量以它们在方程中出现的顺序显示。任何存储在变量中的值也都显示出来。
- 默认的下限和上限显示在编辑器最后一行 (**bound=(-1E99,1E99)**)。
- 如果编辑器内容超出屏幕,  $\downarrow$  显示在底行的第一列。

**提示:** 要用求解器求解如  $K=.5MV^2$  的方程, 在方程编辑器输入 **eqn:0=K-.5MV<sup>2</sup>**。

## 输入并编辑变量值

当在交互式求解器编辑器里输入或编辑变量值时, 新值存储到内存中该变量位置。

可以输入表达式来表示变量值, 在移动到下一个变量前会计算表达式。在迭代的每一步中表达式必须转化为实数。

可以将方程存储到任何 **VAR** **Y-VARS** 变量, 如 **Y1** 或 **r6**, 然后在方程中引用变量。交互式求解器编辑器显示方程中引用的所有 **Y=** 函数变量。

```
\Ys X^2-4AC
\Y0=
```

```
EQUATION SOLVER
eqn:0=Ys+7
```

```
Ys+7=0
X=0
A=0
C=0
bound=(-1E99,1...
```

### 在方程求解器中求解变量

在方程存储到 **eqn** 后执行以下步骤，用方程求解器求解某一变量：

1. 从 **MATH** 菜单选择 **0:Solver**，显示交互式求解器编辑器（如果还没有显示）。

```
Q^3+P^2-125=0
Q=0
P=0
bound={-1E99, 1...
```

2. 输入或编辑每个已知变量值。除未知变量外所有变量必须有值。按 **ENTER** 或  移动光标到下一变量。

```
Q^3+P^2-125=0
Q=0
P=5
bound={-1E99, 1...
```

3. 输入求解变量的初始估计值。这是可选的，但可帮助迅速求解。此外，对于带有多根的方程，**TI-83 Plus** 会试图显示最接近估计值的结果。

```
Q^3+P^2-125=0
Q=4
P=5
bound={-1E99, 1...
```

默认估计值由  $\frac{(upper+lower)}{2}$  计算。



## 在方程求解器中求解变量 (续)

4. 编辑 **bound={lower,upper}**。 *lower* 和 *upper* 是 TI-83 Plus 的求解区间。这是可选的，但可帮助快速求解。默认值为 **bound={-1E99,1E99}**。
5. 移动光标到要求解的变量，然后按 **[ALPHA] [SOLVE]** (在 **[ENTER]** 键上方)。

```
Q^3+P^2-125=0
■ Q=4.6415888336...
P=5
bound={-50,50}
■ left-rt=0
```

- 结果显示在要求解的变量旁边。第一列显示一个实心方形，标记变量已求解，说明方程等式成立。省略号表示数值超出屏幕。  
**注意：**当某个数超出屏幕时，按 **[▢]** 滚动到数值末尾以确定该数值是否以负或正指数结尾。一个很小的数可能看起来象一个非常大的数，除非滚动到右边看到指数。
- 变量值在内存中更新。
- **left-rt=diff** 显示在编辑器的底行。 *diff* 是方程左右两边的差。在第一列一个实心方形显示在 **left-rt=** 旁边，表示方程在已求解变量的新值处已计算。

### 编辑存储于 eqn 的方程

当交互式方程求解器显示时，要编辑或替换存储于 **eqn** 的方程，按  $\square$  直到显示方程编辑器，然后编辑方程。

### 方程有多根

一些方程有多个解。可以输入新估计值（页 2-10）或新区间（页 2-11）来寻求其它解。

### 进一步求解

在求解完一个变量后，可以在交互式求解器编辑器里继续研究结果。编辑一个或多个变量值。当编辑任何变量值时，以前的解和 **left-rt=diff** 旁边的实心方形消失。移动光标到现在要求解的变量，然后按 **[ALPHA] [SOLVE]**。

### 控制由求解器或 solve( 得到的解

TI-83 Plus 用迭代过程求解方程。要控制求解过程，输入接近解的区间和区间内的初始估计值。这会帮助快速求解。此外，这也会决定方程有多解时需要哪个解。

### 在主屏幕或程序中使用 solve(

函数 **solve(** 只在 CATALOG 菜单或程序里有效。它返回表达式 *expression* 对变量 *variable* 在给定初始估计值 *guess*、求解下限 *lower* 和上限 *upper* 区间内的结果（根）。默认下限 *lower* 为 -1E99。默认上限 *upper* 为 1E99。**solve(** 只对实数有效。

**solve(expression,variable,guess[{lower,upper}])**

假定 *expression* 等于零。变量 *variable* 的值不会在内存中更新。估计值 *guess* 可能是数值或含有两个数值的数组。在表达式 *expression* 计算之前，除了变量 *variable* 之外所有表达式 *expression* 中的变量值必须存储，下限 *lower* 和上限 *upper* 必须以数组格式输入。

```
5→P
solve(Q^3+P^2-125
,Q,4,{-50,50})
4.641588834
```

## MATH NUM 菜单

要显示 MATH NUM 菜单，按 **MATH** 。

MATH **NUM** CPX PRB

1: abs(	绝对值
2: round(	四舍五入
3: iPart(	整数部分
4: fPart(	小数部分
5: int(	最大整数
6: min(	最小值
7: max(	最大值
8: lcm(	最小公倍数
9: gcd(	最大公约数

### abs(

**abs(** (绝对值) 返回实数或复数 (模)、表达式、数组和矩阵的绝对值。

**abs(value)**

```
abs(-256)
abs({1.25, -5.67})
)
(1.25 5.67)
```

**注意:** **abs(** 在 MATH CPX 菜单里也有效。

### round(

**round(** 返回已四舍五入到 *#decimals* ( $\leq 9$ ) 的数值、表达式、数组或矩阵。若 *#decimals* 省略, *value* 四舍五入到能显示的位数, 最多 10 位。

**round(value[,#decimals])**

```
round( $\pi$ , 4)
3.1416
```

```
123456789012+C
1.23456789E11
C-round(C)
12
123456789012-123
456789000
12
```

**iPart(、 fPart(**

**iPart(** (整数部分) 返回实数或复数、表达式、数组和矩阵的整数部分。

**iPart(value)**

**fPart(** (小数部分) 返回实数或复数、表达式、数组和矩阵的小数部分。

**fPart(value)**

```
iPart(-23.45) -23
fPart(-23.45) -.45
```

**int(**

**int(** (最大整数) 返回实数或复数、表达式、数组和矩阵的最大整数 ( $\leq$ )。

**int(value)**

```
int(-23.45) -24
```

**注意：**对于给定值 *value*，非负数和负整数的 **int(** 和 **iPart(** 结果相同，但对负小数前者比 **iPart(** 的结果小 1。

## min( , max(

**min(** (最小值) 返回数值 *valueA* 和数值 *valueB* 中较小者或数组 *list* 中的最小元素。如果比较数组 *listA* 和数组 *listB* , **min(** 返回一个数组, 数组中每个元素为相应两个数组元素中较小者。如果比较数组 *list* 和数值 *value* , **min(** 用数组 *list* 中每个元素与数值 *value* 比较。

**max(** (最大值) 返回数值 *valueA* 和数值 *valueB* 中较大者或数组 *list* 中的最大元素。如果比较数组 *listA* 和数组 *listB* , **max(** 返回一个数组, 数组中每个元素为相应两个数组元素中较大者。如果比较数组 *list* 和数值 *value* , **max(** 用数组 *list* 中每个元素与数值 *value* 比较。

**min(valueA,valueB)**

**max(valueA,valueB)**

**min(list)**

**max(list)**

**min(listA,listB)**

**max(listA,listB)**

**min(list,value)**

**max(list,value)**

```
min(3,2+2)          3
min({3,4,5},4)     {3 4 4}
max({4,5,6})       6
```

注意: **min(** 和 **max(** 在 LIST MATH 菜单里也有效。

## lcm( , gcd(

**lcm(** 返回数值 *valueA* 和 *valueB* 的最小公倍数, 这两个数值必须是非负整数。当指定数组 *listA* 和 *listB* 时, **lcm(** 返回每对元素的最小公倍数数组。如果指定数组 *list* 和数值 *value* , **lcm(** 寻求数组 *list* 每个元素与数值 *value* 的最小公倍数。

**gcd(** 返回数值 *valueA* 和 *valueB* 的最大公约数, 这两个数值必须是非负整数。当指定数组 *listA* 和 *listB* 时, **gcd(** 返回每对元素的最大公约数数组。如果指定数组 *list* 和数值 *value* , **gcd(** 寻求数组 *list* 每个元素与数值 *value* 的最大公约数。

**lcm(valueA,valueB)**

**gcd(valueA,valueB)**

**lcm(listA,listB)**

**gcd(listA,listB)**

**lcm(list,value)**

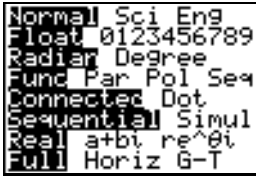
**gcd(list,value)**

```
lcm(2,5)           10
gcd({48,66},{64,122}) {16 2}
```

## 复数方式

TI-83 Plus 以直角坐标和极坐标方式显示复数。要选择复数方式，按 **[MODE]**，然后选择两种方式之一。

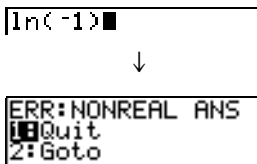
- **a+bi**（直角坐标复数方式）
- **re<sup>∠</sup>θi**（极坐标复数方式）



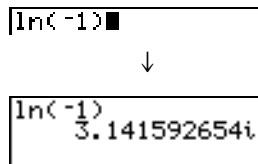
在 TI-83 Plus 里，复数可以存储于变量。此外，复数也是合法的数组元素。

在 **Real** 方式下，复数结果返回错误，除非以复数作为输入。例如，在 **Real** 方式下，**ln(-1)** 返回错误；在 **a+bi** 方式下，**ln(-1)** 返回结果。

**Real** 方式



**a+bi** 方式



## 输入复数

复数以直角坐标方式存储，但可以以直角坐标形式或极坐标形式输入而不管方式设置如何。复数分量可以是实数或计算结果为实数的表达式；表达式在命令执行时进行计算。

### 注意 Radian 和 Degree 方式

弧度方式是复数计算的推荐方式。TI-83 Plus 在内部转换所有输入的三角数值为弧度方式，但不为指数、对数或双曲函数转换数值。

在度方式下，复数恒等式如  $e^{i\theta} = \cos(\theta) + i \sin(\theta)$  一般不成立，因为余弦和正弦值已转换为弧度而  $e^{i\theta}$  却没有。例如， $e^{i45} = \cos(45) + i \sin(45)$  在内部处理为  $e^{i45} = \cos(\pi/4) + i \sin(\pi/4)$ 。在弧度方式下复数恒等式通常成立。

## 解释复数结果

复数结果，包括数组元素，将以直角坐标或极坐标方式显示，以哪种方式显示由方式设置或显示转换指令（页 2-19）而定。在下面示例中，设置为极坐标复数( $re^{\theta i}$ )和 **Radian** 方式。

```
(2+i)-(1e^(π/4i))
)
1.325654296e^(...
```

## 直角坐标复数方式

直角坐标复数方式以  $a+bi$  方式识别和显示复数，其中  $a$  是实数分量， $b$  是虚数分量， $i$  是等于  $\sqrt{-1}$  的常数。

```
ln(-1)
3.141592654i
```

要以直角坐标方式输入复数，输入  $a$  (*real component*) 的值，按  $\oplus$  或  $\ominus$ ，输入  $b$  (*imaginary component*) 的值，然后按  $\boxed{2nd} [i]$  (常数)。

*real component(+ or -)imaginary component i*

```
4+2i
4+2i
```

## 极坐标复数方式

极坐标复数方式以  $re^{\theta i}$  方式识别和显示复数，其中  $r$  是模长， $e$  是自然对数底， $\theta$  是角度， $i$  是等于  $\sqrt{-1}$  的常数。

```
ln(-1)
3.141592654e^(1...
```

要以极坐标方式输入复数，输入  $r$  (*magnitude*) 的值，按  $\boxed{2nd} [e^x]$  (指数函数)，输入  $\theta$  (*angle*) 的值，按  $\boxed{2nd} [i]$  (常数)，然后按  $\boxed{\square}$ 。

*magnitudee^(anglei)*

```
10e^(π/3i)
10e^(1.04719755...
```

## MATH CPX 菜单

要显示 MATH CPX 菜单，按  $\boxed{\text{MATH}} \boxed{\blacktriangleright} \boxed{\blacktriangleright}$ 。

MATH NUM **CPX** PRB

1: conj(	返回共轭复数。
2: real(	返回实数分量。
3: imag(	返回虚数分量。
4: angle(	返回极坐标角度。
5: abs(	返回模。
6: $\blacktriangleright$ Rect	以直角坐标方式显示结果。
7: $\blacktriangleright$ Polar	以极坐标方式显示结果。

### conj(

**conj(** (共轭) 返回复数或复数数组的共轭复数。

**conj( $a+bi$ )** 对  $a+bi$  方式返回  $a-bi$ 。

**conj( $re^{i\theta}$ )** 对  $re^{i\theta}$  方式返回  $re^{-i\theta}$ 。

```
conj(3+4i)
3-4i
```

```
conj(3e^(4i))
3e^(2.283185307...
```

### real(

**real(** (实数分量) 返回复数或复数数组的实数分量。

**real( $a+bi$ )** 返回  $a$ 。

**real( $re^{i\theta}$ )** 返回  $r \cdot \cos(\theta)$ 。

```
real(3+4i)
3
```

```
real(3e^(4i))
-1.960930863
```

### imag(

**imag(** (虚数分量) 返回复数或复数数组的虚数 (非实数) 分量。

**imag( $a+bi$ )** 返回  $b$ 。

**imag( $re^{i\theta}$ )** 返回  $r \cdot \sin(\theta)$ 。

```
imag(3+4i)
4
```

```
imag(3e^(4i))
-2.270407486
```



## angle(

**angle(** 返回复数或复数数组的极坐标角度，可由  $\tan^{-1}(b/a)$  计算，其中 **b** 是虚数分量，**a** 是实数分量。计算结果在第二象限时用  $+\pi$  调整，在第三象限时用  $-\pi$  调整。

**angle(a+bi)** 返回  $\tan^{-1}(b/a)$ 。

**angle(re^( $\theta$ i))** 返回  $\theta$ ，其中  $-\pi < \theta < \pi$ 。

```
angle(3+4i)
.927295218
```

```
angle(3e^(4i))
-2.283185307
```

## abs(

**abs(** (绝对值) 返回复数或复数数组的模， $\sqrt{(real^2+imag^2)}$ 。

**abs(a+bi)** 返回  $\sqrt{(a^2+b^2)}$ 。

**abs(re^( $\theta$ i))** 返回  $r$  (模)。

```
abs(3+4i)
5
```

```
abs(3e^(4i))
3
```

## ►Rect

**►Rect** (以直角坐标显示) 以直角坐标方式显示复数结果。仅当它在表达式结尾时有效，对实数结果无效。

*complex result*►**Rect** 返回  $a+bi$ 。

```
√(-2)►Rect
1.414213562i
```

## ►Polar

**►Polar** (以极坐标显示) 以极坐标方式显示复数结果。仅当它在表达式结尾时有效，对实数结果无效。

*complex result*►**Polar** 返回  $re^{(\theta i)}$ 。

```
√(-2)►Polar
1.414213562e^(1...
```

## MATH PRB 菜单

要显示 MATH PRB 菜单，按  $\boxed{\text{MATH}} \boxed{\text{4}}$ 。

MATH	NUM	CPX	PRB
1:	rand		随机数产生器
2:	nPr		排列数
3:	nCr		组合数
4:	!		阶乘
5:	randInt(		随机整数产生器
6:	randNorm(		正态分布随机 #
7:	randBin(		二项式分布随机 #

## rand

**rand** (随机数) 产生并返回大于 0 且小于 1 的一个或多个随机数。要产生随机数数组，为 *numtrials* (试验数) 指定  $> 1$  的整数。*numtrials* 默认值为 1。

**rand**[(*numtrials*)]

**提示：**要产生 0 到 1 之外的随机数，可以在表达式中包含 **rand**。例如，**rand\*5** 产生大于 0 且小于 5 之间的随机数。

每次执行 **rand**，TI-83 Plus 对于给定起始值均产生相同随机数序列。

TI-83 Plus 为 **rand** 起始值的出厂设置是 **0**。为产生不同的随机数序列，存储任意非零值到 **rand**。要恢复出厂设置的起始值，存储 **0** 到 **rand** 或重置为默认值 (第 18 章)。

**注意：**起始值也影响 **randInt()**、**randNorm()** 和 **randBin()** 指令 (页 2-22)。

```
rand
.1272157551
.2646513087
1→rand          1
rand(3)
(.7455607728 .8...
```

## nPr 、 nCr

**nPr** (排列数) 返回从基体 *items* 每次取 *number* 个数的排列数。基体 *items* 和 *number* 必须是非负整数。基体 *items* 和 *number* 可以是数组。

*items nPr number*

**nCr** (组合数) 返回从基体 *items* 每次取 *number* 个数的组合数。基体 *items* 和 *number* 必须是非负整数。基体 *items* 和 *number* 可以是数组。

*items nCr number*

5 nPr 2	20
5 nCr 2	10
{2,3} nPr {2,2}	{2,6}

## ! (阶乘)

**!** (阶乘) 返回整数或 0.5 的倍数的阶乘。对于数组，返回每个整数或 0.5 的倍数的阶乘。值 *value* 必须  $\geq -1.5$  且  $\leq 69$ 。

*value!*

6!	720
{5,4,6}!	{120 24 720}

**注意：**阶乘用关系  $(n+1)! = (n+1)*n!$  递归计算，直到 *n* 减小到 0 或 -1/2。在这时，定义  $0! = 1$  或  $(-1/2)! = \sqrt{\pi}$  用于完成计算。因此：

$n! = n*(n-1)*(n-2)* \dots *2*1$ ，如果 *n* 是  $\geq 0$  的整数

$n! = n*(n-1)*(n-2)* \dots *1/2*\sqrt{\pi}$ ，如果  $n+1/2$  是  $\geq 0$  的整数

*n!* 为一错误，如果 *n* 和  $n+1/2$  都不是  $\geq 0$  的整数。

(变量 *n* 等于上面描述的 *value*。)

randInt(

**randInt(** (随机整数) 产生并显示在指定整数下限 *lower* 和整数上限 *upper* 区间内的一个随机整数。要产生随机数组, 为 *numtrials* (试验数) 指定一个  $>1$  的整数; 如果不指定, 默认值为 1。

**randInt(lower,upper[,numtrials])**

```
randInt(1,6)+randInt(1,6)
6
randInt(1,6,3)
{2 1 5}
```

randNorm(

**randNorm(** (随机正态) 由指定正态分布产生并显示一个随机实数。每个产生的值可能是任意实数, 但大部分在区间  $[\mu-3(\sigma), \mu+3(\sigma)]$  内。要产生随机数组, 为 *numtrials* (试验数) 指定一个  $>1$  的数; 如果不指定, 默认值为 1。

**randNorm( $\mu,\sigma,numtrials$ )**

```
randNorm(0,1)
.0772076175
randNorm(35,2,10)
0)
{34.02701938 37...
```

randBin(

**randBin(** (随机二项式) 由指定二项式分布产生并显示一个随机整数。*numtrials* (试验数) 必须  $\geq 1$ 。*prob* (成功概率) 必须在  $\geq 0$  且  $\leq 1$  之间。为产生随机数组, 为 *numsimulations* (模拟数) 指定  $>1$  的整数, 如果不指定, 默认值为 1。

**randBin(numtrials,prob[,numsimulations])**

```
randBin(5,.2)
3
randBin(7,.4,10)
{3 3 2 5 1 2 2 ...}
```

**注意:** 存储在 **rand** 的起始值也影响 **randInt(**、**randNorm(** 和 **randBin(** 指令 (页 2-20)。

## ANGLE 菜单

要显示 ANGLE 菜单，按  $\boxed{2nd}$  [ANGLE]。ANGLE 菜单显示角度指示器和指令。**Radian/Degree** 方式设置影响 TI-83 Plus 解释 ANGLE 菜单输入。

ANGLE	
1: °	度记号
2: '	DMS 分记号
3: "	弧度记号
4: ▶DMS	以度/分/秒显示
5: R▶Pr(	给定 X 和 Y，返回 r
6: R▶Pθ(	给定 X 和 Y，返回 θ
7: P▶Rx(	给定 R 和 θ，返回 x
8: P▶Ry(	给定 R 和 θ，返回 y

## DMS 输入记号

DMS（度/分/秒）输入记号包括度符号 (°)、分符号 (')、秒符号 (")。度 *degrees* 必须是实数；分 *minutes* 和秒 *seconds* 必须是  $\geq 0$  的实数。

*degrees°minutes'seconds"*

例如，输入 30 度 1 分 23 秒。如果角度方式不是设置为 **Degree**，必须使用 ° 以便 TI-83 Plus 能够解释参数为度、分、秒。

**Degree** 方式

```
sin(30°1'23")
.5003484441
```

**Radian** 方式

```
sin(30°1'23")
-.9842129995
sin(30°1'23"°)
.5003484441
```

## ° (度)

° (度) 指定角度或角度数组为度表示而不管当前角度设置方式。在 **Radian** 方式，可以使用 ° 转换为弧度。

*value*°

*{value1,value2,value3,value4,...,value n}*°

° 也以 DMS 格式指定度 *degrees* (D)。

' (分) 以 DMS 格式指定分 *minutes* (M)。

" (秒) 以 DMS 格式指定秒 *seconds* (S)。

**注意：**" 不在 ANGLE 菜单中，要输入 "，按  $\boxed{ALPHA}$  ["]。

r (弧度)

r (弧度) 指定角度或角度数组为弧度表示而不管当前角度设置方式。在 **Degree** 方式，可以使用 r 转换弧度为度。

value<sup>r</sup>

**Degree** 方式

```
sin((π/4)r)
.7071067812
sin(0, π/2)r
0 13
(π/4)r
45
```

►DMS

►DMS (度/分/秒) 以 DMS 格式显示结果 *answer* (页 2-23)。要以度、分、秒格式解释结果 *answer*，方式设置必须是 **Degree**。►DMS 只在行结尾有效。

*answer*►DMS

```
54°32'30"*2
109.0833333
Ans►DMS
109°5'0"
```

R►Pr(, R►Pθ(, P►Rx(, P►Ry(

R►Pr( 转换直角坐标为极坐标并返回模 r。R►Pθ( 转换直角坐标为极坐标并返回角度 θ。x 和 y 可以是数组。

R►Pr(x,y), R►Pθ(x,y)

```
R►Pr(-1,0) 1
R►Pθ(-1,0) 3.141592654
```

注意：设置了 Radian 方式。

P►Rx( 转换极坐标为直角坐标并返回 x。P►Ry( 转换极坐标为直角坐标并返回 y。r 和 θ 可以是数组。

P►Rx(r,θ), P►Ry(r,θ)

```
P►Rx(1,π) -1
P►Ry(1,π) 0
```

注意：设置了 Radian 方式。

## TEST 菜单

要显示 TEST 菜单，按  $\boxed{2\text{nd}} \text{ [TEST]}$ 。

运算符...	返回 1 (真) 若...
<b>TEST</b> LOGIC	
1: =	相等
2: $\neq$	不等
3: >	大于
4: $\geq$	大于等于
5: <	小于
6: $\leq$	小于等于

=、 $\neq$ 、>、 $\geq$ 、<、 $\leq$

关系运算符比较值  $valueA$  和值  $valueB$ ，并在判断为真时返回 **1**；判断为假时返回 **0**。值  $valueA$  和值  $valueB$  可以是实数、表达式或数组。只对于 = 和  $\neq$ ，值  $valueA$  和  $valueB$  也可以是矩阵或复数。如果值  $valueA$  和  $valueB$  是矩阵，则必须有相同维数。

关系运算符经常用于程序中控制程序流向或用于绘图中控制指定值的函数图形。

$$valueA=valueB$$

$$valueA\neq valueB$$

$$valueA>valueB$$

$$valueA\geq valueB$$

$$valueA<valueB$$

$$valueA\leq valueB$$

```

25=26
(1,2,3)<3      0
(1,2,3)≠(3,2,1)  0
(1,2,3)≠(1,1,0)  0
(1,2,3)≠(3,2,1)  0
(1,2,3)≠(1,0,1)  0
    
```

## 使用关系运算

根据 EOS 规则（第 1 章），关系运算符在数学函数之后执行。

- 表达式  $2+2=2+3$  返回 **0**。TI-83 Plus 根据 EOS 规则先进行加法，然后比较 4 和 5。
- 表达式  $2+(2=2)+3$  返回 **6**。由于判断在括号中，TI-83 Plus 先进行关系判断，然后进行 2 加 1，加 3 运算。

## TEST LOGIC 菜单

要显示 TEST LOGIC 菜单，按 **[2nd] [TEST] [▶]**。

运算符...	返回 1 (真) 若...
TEST LOGIC	
1: and	两个值都非零 (真)。
2: or	至少一个非零 (真)。
3: xor	只有一个为零 (真)。
4: not(	值为零 (真)。

## 布尔运算符

布尔运算符经常用于程序中控制程序流向或用于绘图中控制指定值的函数图形。值被解释为零 (假) 或非零 (真)。

### and、or、xor

**and**、**or** 和 **xor** (异或) 根据下表原则在表达式为真时，返回 **1**；在表达式为假时返回 **0**。值 *valueA* 和值 *valueB* 可以是实数、表达式或数组。

*valueA* and *valueB*

*valueA* or *valueB*

*valueA* xor *valueB*

valueA	valueB		and	or	xor
≠0	≠0	返回	1	1	0
≠0	0	返回	0	1	1
0	≠0	返回	0	1	1
0	0	返回	0	0	0

### not(

**not(** 在值 *value* (可以为表达式) 是 **0** 时返回 **1**。

**not(value)**

## 使用布尔运算

布尔逻辑运算经常与关系运算一起使用。在下面程序中，指令将 **4** 存储到变量 **C**。

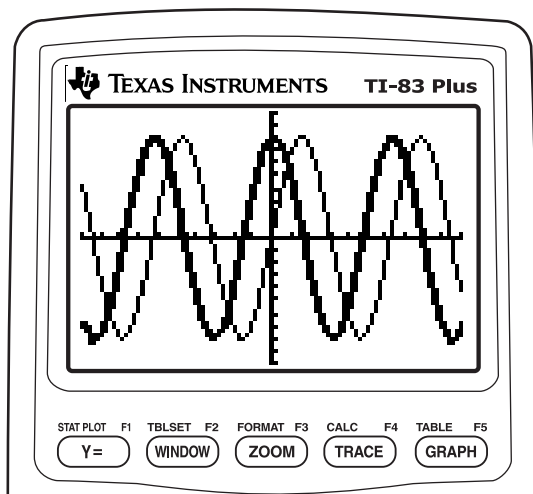
```
PROGRAM: BOOLEAN
:2→A:3→B
:If A=2 and B=3
:Then:4→C
:Else:5→C
:End
```



# 3 函数 图象

## 目录

入门：画圆.....	2
定义图象.....	3
设置图象方式.....	4
定义函数.....	5
选择和取消选择函数.....	7
设置函数图象样式.....	9
设置观察窗口变量.....	11
设置图象格式.....	13
显示图象.....	15
用自由移动光标研究图象.....	17
用 TRACE 研究图象.....	18
用 ZOOM 指令研究图象.....	20
使用 ZOOM MEMORY.....	23
使用 CALC（计算）运算.....	25



入门是对本章内容的快速介绍。详细信息请阅读本章。

画一个半径为 10，圆心为标准观察窗口原点的圆。要画此圆，必须分别输入圆上下两部分的公式，然后使用 ZSquare（方形缩放）调整显示以使函数显示为一个圆。

1. 在 **Func** 方式，按  $\boxed{Y=}$  显示 Y= 编辑器。按  $\boxed{2nd}$   $\boxed{[ \sqrt{ } ]}$   $\boxed{100}$   $\boxed{[ ]}$   $\boxed{X,T,\theta,r}$   $\boxed{[ x^2 ]}$   $\boxed{[ ]}$   $\boxed{ENTER}$  输入表达式  $Y=\sqrt{(100-X^2)}$ ，它定义圆的上半部。

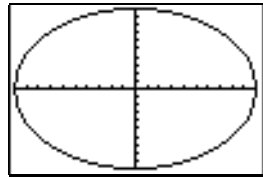
表达式  $Y=-\sqrt{(100-X^2)}$  定义圆的下半部。在 TI-83 Plus 中，可以将一个函数定义为另一个函数的函数。要定义  $Y2=-Y1$ ，按  $\boxed{[ ]}$  输入负号，按  $\boxed{[ VARS ]}$   $\boxed{[ \blacktriangleright ]}$  显示 VARS Y-VARS 菜单。然后按  $\boxed{[ ENTER ]}$  选择 **1:Function**。显示 FUNCTION 子菜单，然后按 **1** 来选择 **1:Y1**。

```
Plot1 Plot2 Plot3
Y1  $\sqrt{(100-X^2)}$ 
Y2  $-Y1$ 
Y3 =
Y4 =
Y5 =
Y6 =
Y7 =
```

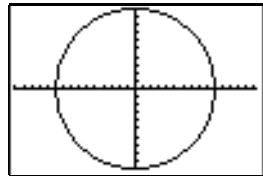
2. 按  $\boxed{[ ZOOM ]}$  **6** 选择 **6:ZStandard**。这是重设窗口变量为标准值的快速方法。它也对函数进行绘图而不必按  $\boxed{[ GRAPH ]}$ 。

注意到函数在标准观察窗口显示为一个椭圆。

```
Plot1 Plot2 Plot3
Y1  $\sqrt{(100-X^2)}$ 
Y2  $-Y1$ 
Y3 =
Y4 =
Y5 =
Y6 =
Y7 =
```



3. 为调整显示以便每个象素有相同的宽度和高度，按  $\boxed{[ ZOOM ]}$  **5** 选择 **5:ZSquare**。函数重绘，现在显示象一个圆。



4. 要看 ZSquare 窗口变量，请按  $\boxed{[ WINDOW ]}$  并注意新的 Xmin、Xmax、Ymin 和 Ymax 值。

```
WINDOW
Xmin=-15.16129...
Xmax=15.161290...
Xscl=1
Ymin=-10
Ymax=10
Vsc1=1
Xres=1
```

## TI-83 Plus — 绘图方式类似

第 3 章只描述函数绘图，但这里所示的步骤对于每种 TI-83 Plus 绘图方式都类似。第 4、5 和 6 章阐述参数图象、极坐标图象和序列图象的独特方面。

## 定义图象

执行以下步骤以任何一种绘图方式定义图象，某些步骤有时不是必需的。

1. 按 **[MODE]** 并设置适当绘图方式（页 3-4）。
2. 按 **[Y=]** 并在 Y= 编辑器中输入、编辑或选择一个或多个函数（页 3-5 和 3-7）。
3. 若需要，请取消选择统计图（页 3-7）。
4. 设置每个函数的图象样式（页 3-9）。
5. 按 **[WINDOW]** 并定义观察窗口变量（页 3-11）。
6. 按 **[2nd] [FORMAT]** 并选择图象格式设置（页 3-13）。

## 显示和研究图象

在定义完图象后，按 **[GRAPH]** 显示它。使用本章描述的 TI-83 Plus 工具研究函数特性。

## 存储图象为以后使用

可存储定义当前图象的元素到 10 个图象数据库变量（**GDB1** 到 **GDB9**，和 **GDB0**；第 8 章）中的任意一个。以后要重建当前图象，只须调用存储该图象的图象数据库即可。

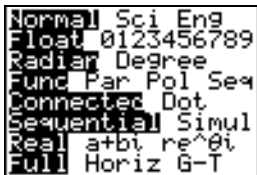
下述类型的信息存储到 **GDB**。

- Y= 函数
- 图象样式设置
- 窗口设置
- 格式设置

可以存储当前图象显示的图形到 10 个图象图形变量（**Pic1** 到 **Pic9**，和 **Pic0**；第 8 章）中的任意一个。然后可以在当前图象上追加一个或多个存储的图形。

### 检查和改变图象方式

要显示方式屏幕，请按 **[MODE]**。默认设置为下面的亮显部分。要绘制函数图象，在输入窗口变量值和输入函数之前必须选择 **Func** 方式。



TI-83 Plus 有以下四种图象方式。

- **Func**（函数图象）
- **Par**（参数图象；第 4 章）
- **Pol**（极坐标图象；第 5 章）
- **Seq**（序列图象；第 6 章）

其他方式设置影响图象结果，第 1 章阐述了每种方式设置。

- **Float** 或 **0123456789**（定点）十进制方式影响所显示的图象坐标。
- **Radian** 或 **Degree** 角度方式影响一些函数的解释。
- **Connected** 或 **Dot** 绘图方式影响所选函数的绘制。
- **Sequential** 或 **Simul** 图象顺序方式在选择多个函数时影响函数绘图。

### 由程序设置方式

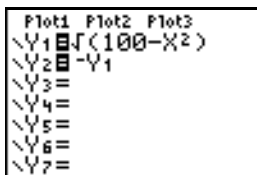
要由程序设置图象方式或其他方式，在程序编辑器一空行开始，并执行以下步骤：

1. 按 **[MODE]** 显示方式设置。
2. 按 **[↓]**、**[→]**、**[←]** 和 **[↑]** 定位光标到要选择的方式。
3. 按 **[ENTER]** 粘贴方式名到光标处。

程序执行后方式改变。

## 在 Y= 编辑器中显示函数

为显示 Y= 编辑器，请按  $\boxed{Y=}$ 。可以存储多至 10 个函数到函数变量 **Y1** 到 **Y9** 和 **Y0** 中。可以立即绘制一个或多个已定义函数的图象。在下面的示例中，定义和选择了函数 **Y1** 和 **Y2**。



## 定义或编辑函数

执行以下步骤来定义或编辑函数：

1. 按  $\boxed{Y=}$  显示 Y= 编辑器。
2. 按  $\boxed{\leftarrow}$  移动光标到要定义或编辑的函数。要删除一个函数，按  $\boxed{\text{CLEAR}}$ 。
3. 输入或编辑表达式以定义函数。
  - 可以在表达式中使用函数和变量（包括矩阵和数组）。当表达式计算结果为非实数时，此值不绘制，也不返回错误。
  - 函数自变量是 **X**。Func 方式定义  $\boxed{X.T.\Theta.n}$  为 **X**。要输入 **X**，按  $\boxed{X.T.\Theta.n}$  或按  $\boxed{\text{ALPHA}} \boxed{X}$ 。
  - 输入第一个字符后，= 亮显，表示选择了这个函数。随着表达式的输入，它在 Y= 编辑器里作为用户定义的函数被存储到变量 **Y<sub>n</sub>**。
4. 按  $\boxed{\text{ENTER}}$  或  $\boxed{\rightarrow}$  移动光标到下一个函数。

## 从主屏幕或程序定义函数

要从主屏幕或程序定义函数，由一空行开始并执行以下步骤：

1. 按  $\boxed{\text{ALPHA}} \boxed{[ ]}$ ，输入表达式，然后再按  $\boxed{\text{ALPHA}} \boxed{[ ]}$ 。
2. 按  $\boxed{\text{STO}} \blacktriangleright$ 。
3. 按  $\boxed{\text{VAR}} \blacktriangleright \mathbf{1}$  从 VARS Y-VARS 菜单选择 **1:Function**。
4. 选择函数名，粘贴函数名到主屏幕或程序编辑器的光标位置处。
5. 按  $\boxed{\text{ENTER}}$  完成本指令。

"expression"  $\rightarrow Y_n$



指令执行后，TI-83 Plus 将表达式存储到指定变量  $Y_n$ ，选择函数，然后显示信息 **Done**。

## 计算表达式中 $Y =$ 函数

可以计算一个  $Y =$  函数  $Y_n$  在指定  $X$  值 *value* 处的值，值 *values* 为数组时返回数组。

$Y_n(\text{value})$

$Y_n(\{\text{value1}, \text{value2}, \text{value3}, \dots, \text{value } n\})$



### 选择和取消选择函数

可以在 Y= 编辑器中选择和取消选择函数（开启和关闭）。选择函数后 = 符号亮显。TI-83 Plus 只对所选函数绘图。可以选择 Y1 到 Y9 和 Y0 函数中的任何一个或全部。

要在 Y= 编辑器中选择或取消选择函数，请按下列步骤进行：

1. 按  $\boxed{\text{Y=}}$  显示 Y= 编辑器。
2. 移动光标到要选择或取消选择的函数。
3. 按  $\boxed{\text{=}}$  定位光标在函数 = 符号上。
4. 按  $\boxed{\text{ENTER}}$  改变选择状态。

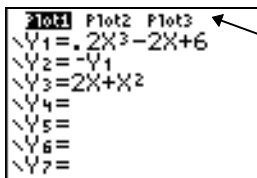
当输入或编辑函数时，它自动被选择。当清除函数时，它被取消选择。

### 在 Y= 编辑器中开启或关闭统计图

要在 Y= 编辑器中观察和改变统计图的开启/关闭状态，使用

**Plot1 Plot2 Plot3**（Y= 编辑器的顶行）。当某个图打开时，其名字在顶行上亮显。

要在 Y= 编辑器中改变统计图的开启/关闭状态，按  $\boxed{\blacktriangle}$  和  $\boxed{\blacktriangleright}$  将光标置于 **Plot1**、**Plot2** 或 **Plot3** 上，然后按  $\boxed{\text{ENTER}}$ 。



**Plot1** 开启。  
**Plot2** 和 **Plot3** 关闭。

### 从主屏幕或程序选择 and 取消选择函数

要从主屏幕或程序选择或取消选择函数，由空白行开始，并执行以下步骤：

1. 按 **[VARS]** **[▶]** 显示 VARS Y-VARS 菜单。
2. 选择 **4:On/Off** 显示 ON/OFF 子菜单。
3. 选择 **1:FnOn** 开启一个或多个函数，或选择 **2:FnOff** 关闭一个或多个函数。所选指令拷贝到光标位置。
4. 输入表示要开启或关闭函数的数字（**1** 到 **9**，或 **0**；不是变量  $Y_n$ ）。
  - 如果输入两个或多个数字，用逗号分隔。
  - 要开启或关闭所有函数，在 **FnOn** 或 **FnOff** 后不输入数字。

**FnOn**[function#,function#,...,function n]

**FnOff**[function#,function#,...,function n]

5. 按 **[ENTER]**。当指令执行后，设置了当前方式下每个函数的状态，显示 **Done**。

例如，在 **Func** 方式，**FnOff :FnOn 1,3** 关闭  $Y=$  编辑器中所有函数，然后开启  $Y_1$  和  $Y_3$ 。








```
FnOff :FnOn 1,3
Done
```

```
Plot1 Plot2 Plot3
\Y1 2X^3-2X+6
\Y2 -Y1
\Y3 X^2
\Y4 =
\Y5 =
\Y6 =
\Y7 =
```



## Y= 编辑器中图象样式图标

下表描述可用于函数绘图的图象样式。使用这些样式后，由视觉就能区分绘制在一起的不同函数。例如，可以设置 **Y1** 为实线，**Y2** 为点线，**Y3** 为粗线。

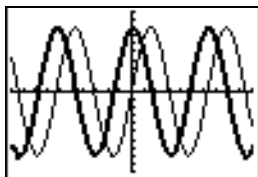
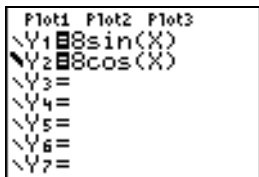
图标	样式	描述
	实线	用实线连接绘图点；在 <b>Connected</b> 方式下这是默认样式
	粗线	用粗实线连接绘图点
	上阴影	阴影覆盖图象上方区域
	下阴影	阴影覆盖图象下方区域
	轨迹	圆形光标跟踪图象的前沿并画出轨迹
	动画	圆形光标跟踪图象的前沿但不画出轨迹
	点	用小点代表绘图点；在 <b>Dot</b> 方式下这是默认样式

**注意：**一些图象样式不是在所有绘图方式下都能用。第 4、5 和 6 章列出用于 **Par**、**Pol** 和 **Seq** 方式的样式。

## 设置图象样式

执行以下步骤设置函数的图象样式：

1. 按 **[Y=]** 显示 Y= 编辑器。
2. 按 **[↓]** 和 **[↑]** 移动光标到那个函数。
3. 按 **[←]** **[←]** 向左移动光标，越过 = 符号到第一列的图象样式图标。插入光标显示。（步骤 2 和步骤 3 可交换。）
4. 反复按 **[ENTER]** 轮转七种图象样式，轮转顺序与上表的排列顺序相同。
5. 选择一个样式后按 **[→]**、**[↑]** 或 **[↓]**。

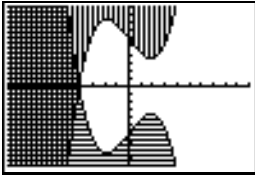


## 上阴影和下阴影

当为两个或多个函数选择  $\overline{\square}$  或  $\underline{\square}$  时, TI-83 Plus 轮转四种阴影图案。

- 垂直线以  $\overline{\square}$  或  $\underline{\square}$  图象样式遮蔽第一个函数。
- 水平线遮蔽第二个函数。
- 左斜线遮蔽第三个函数。
- 右斜线遮蔽第四个函数。
- 对于第五个  $\overline{\square}$  或  $\underline{\square}$  函数又回到垂直线, 重复上述顺序。

当阴影区域相交时, 图案重叠。



**注意:** 有些  $Y=$  函数, 如  $Y1=\{1,2,3\}X$  绘制一族曲线, 为这样的函数选择  $\overline{\square}$  或  $\underline{\square}$  时, 四种阴影图案轮转着用于每个曲线族成员。

## 从程序设置图象样式

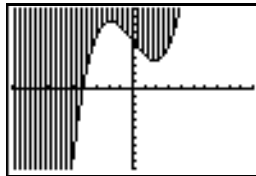
要从程序设置图象样式, 则从 PRGM CTL 菜单选择 **H:GraphStyle(**。要显示此菜单, 在程序编辑器中按  $\overline{\text{PRGM}}$ 。  $function\#$  是当前图象方式下  $Y=$  函数名的数字。  $graphstyle\#$  是一个 1 到 7 的整数, 对应某种图象样式, 如下所示:

- |                                 |                               |                                |
|---------------------------------|-------------------------------|--------------------------------|
| 1 = $\text{—}$ (实线)             | 2 = $\overline{\square}$ (粗线) | 3 = $\overline{\square}$ (上阴影) |
| 4 = $\underline{\square}$ (下阴影) | 5 = $\text{—}$ (轨迹)           | 6 = $\text{—}$ (动画)            |
|                                 |                               | 7 = $\cdot$ (点)                |

**GraphStyle( $function\#,graphstyle\#$ )**

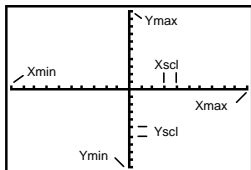
例如, 当此程序在 **Func** 方式下执行, **GraphStyle(1,3)** 设置  $Y1$  为  $\overline{\square}$ (上阴影)。

```
PROGRAM: SHADE
: " .2X^3-2X+6 " → Y1
: GraphStyle(1,3)
: DispGraph
```



## TI-83 Plus 观察窗口

观察窗口是指由  $X_{min}$ 、 $X_{max}$ 、 $Y_{min}$  和  $Y_{max}$  定义的坐标平面的一部分。 $X_{scl}$  (X 尺度) 定义 x 坐标轴上的标记距离,  $Y_{scl}$  (Y 尺度) 定义 y 坐标轴上的标记距离。要关闭标记, 设置  $X_{scl}=0$  和  $Y_{scl}=0$ 。



```
WINDOW
Xmin=-10
Xmax=10
Xscl=1
Ymin=-10
Ymax=10
Yscl=1
Xres=1
```

## 显示窗口变量

按 **WINDOW** 显示当前窗口变量值。上面右边的窗口编辑器显示了在 **Func** 图象方式和 **Radian** 角度方式下的默认值。窗口变量随图象方式不同而不同。

**Xres** 只对函数图象设置象素分辨率 (1 到 8)。默认值为 1。

- 在  $X_{res}=1$  时, 在 x 坐标轴上每个象素处计算和绘制函数。
- 在  $X_{res}=8$  时, 沿 x 坐标轴每隔八个象素计算和绘制函数。

**提示:** 小的  $X_{res}$  值能提高图象分辨率, 但可能使得 TI-83 Plus 绘图速度较慢。

## 改变窗口变量值

执行以下步骤在窗口编辑器里改变窗口变量值:

1. 按 **▼** 或 **▲** 将光标移到要修改的窗口变量。
2. 编辑变量值, 该值可以是表达式。
  - 输入新的值, 将清除原始值。
  - 移动光标到指定位, 然后编辑此位。
3. 按 **ENTER**、**▼** 或 **▲**。如果输入的是表达式, 则 TI-83 Plus 计算它, 并存储新值。

**注意:** 为绘图,  $X_{min}<X_{max}$  和  $Y_{min}<Y_{max}$  必须成立。

### 从主屏幕或程序存储到窗口变量

要存储数值（或表达式）到窗口变量，则在一空行开始，并执行以下步骤：

1. 输入要存储的值。
2. 按  $\boxed{\text{STO}}\blacktriangleright$ 。
3. 按  $\boxed{\text{VARS}}$  显示 VARS 菜单。
4. 选择 **1:Window** 显示 **Func** 窗口变量（X/Y 子菜单）。
  - 按  $\boxed{\blacktriangledown}$  显示 **Par** 和 **PoI** 窗口变量（T/θ 子菜单）。
  - 按  $\boxed{\blacktriangledown}\boxed{\blacktriangledown}$  显示 **Seq** 窗口变量（U/V/W 子菜单）。
5. 选择要存储数值的窗口变量。变量名粘贴到当前光标位置。
6. 按  $\boxed{\text{ENTER}}$  完成此指令。

指令执行后，TI-83 Plus 将值存储到窗口变量并显示这个值。

```

14→Xmax
14
    
```

### $\Delta X$ 和 $\Delta Y$

变量  $\Delta X$  和  $\Delta Y$ （VARS (1:Window) X/Y 子菜单上的项 **8** 和 **9**）定义图象上一个象素中心到任一相邻象素中心的距离（作图精度）。当显示图象时， $\Delta X$  和  $\Delta Y$  是由  $X_{\min}$ 、 $X_{\max}$ 、 $Y_{\min}$  和  $Y_{\max}$  计算得到的。

$$\Delta X = \frac{(X_{\max} - X_{\min})}{94} \quad \Delta Y = \frac{(Y_{\max} - Y_{\min})}{62}$$

可以将值存储到  $\Delta X$  和  $\Delta Y$ 。如果这样做， $X_{\max}$  和  $Y_{\max}$  则由  $\Delta X$ 、 $X_{\min}$ 、 $\Delta Y$  和  $Y_{\min}$  计算得到。

## 显示格式设置

要显示格式设置，请按 **[2nd] [FORMAT]**。默认设置为下面的亮显项。

<b>RectGC</b>	PolarGC	设置光标坐标。
<b>CoordOn</b>	CoordOff	设置坐标显示为开启或关闭。
<b>GridOff</b>	GridOn	设置网格关闭或开启
<b>AxesOn</b>	AxesOff	设置坐标轴为开启或关闭。
<b>LabelOff</b>	LabelOn	设置坐标轴标记为关闭或开启。
<b>ExprOn</b>	ExprOff	设置表达式显示为开启或关闭。

格式设置定义图象的显示外观，适用于所有图象方式。**Seq** 图象方式还有另一种方式设置（第 6 章）。

## 改变格式设置

执行以下步骤改变格式设置：

1. 在需要时按 **◀**、**▶**、**▲** 和 **▼** 移动光标到要选择的设置。
2. 按 **[ENTER]** 选择亮显的设置。

## RectGC、PolarGC

**RectGC**（图象直角坐标）以直角坐标 **X** 和 **Y** 显示光标位置。

**PolarGC**（图象极坐标）以极坐标 **R** 和 **θ** 显示光标位置。

**RectGC/PolarGC** 设置决定在绘图、移动自由光标或跟踪时哪个变量被更新。

- **RectGC** 更新 **X** 和 **Y**；若选择 **CoordOn** 格式，则显示 **X** 和 **Y**。
- **PolarGC** 更新 **X**、**Y**、**R** 和 **θ**；若选择 **CoordOn** 格式，则显示 **R** 和 **θ**。

### CoordOn 、 CoordOff

**CoordOn** (坐标开启) 在图象底部显示光标的坐标。如果选择 **ExprOff** 格式, 则函数号显示在右上角。

**CoordOff** (坐标关闭) 不显示函数号或坐标。

### GridOff 、 GridOn

网格点覆盖观察窗口 (页 3-11), 其间距对应于坐标轴上标记的距离。

**GridOff** 不显示网格点。

**GridOn** 显示网格点。

### AxesOn 、 AxesOff

**AxesOn** 显示坐标轴。

**AxesOff** 不显示坐标轴。

在这种设置下, **LabelOff/LabelOn** 格式设置不起作用。

### LabelOff 、 LabelOn

如果选择 **AxesOn** 格式, 则 **LabelOff** 和 **LabelOn** 确定是否显示坐标轴 (X 和 Y) 的轴标。

### ExprOn 、 ExprOff

**ExprOn** 和 **ExprOff** 确定在跟踪光标活动时是否显示  $Y=$  表达式。这种格式设置也适用于统计图。

当选择 **ExprOn** 时, 表达式显示在图象屏幕左上角。

当 **ExprOff** 和 **CoordOn** 二者都被选择时, 右上角的数字表示被跟踪的函数。

### 显示新图象

要显示所选函数的图象，请按 **GRAPH**。TRACE、ZOOM 指令和 CALC 操作自动显示图象。在 TI-83 Plus 绘图时，忙指示器显示。X 和 Y 随着图象的绘制而更新。

### 暂停或中止绘图

在绘图时，可以暂停或中止绘图。

- 按 **ENTER** 暂停；再按 **ENTER** 继续。
- 按 **ON** 中止；然后按 **GRAPH** 重绘。

### 便捷绘图

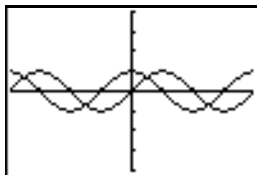
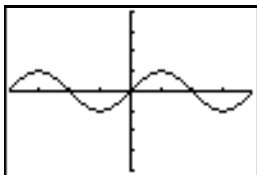
便捷绘图是 TI-83 Plus 的一项功能，即按 **GRAPH** 时立即再次显示上次图象，但必须自图象上次显示后，所有可能引起重新绘图的图象因子仍保持不变。

如果自图象上次显示后执行了任何以下的操作，TI-83 Plus 将在按 **GRAPH** 时以新值为基础重新绘图。

- 改变影响图象的方式设置
- 改变当前图形中的函数
- 选择或取消选择函数或统计图
- 改变所选函数的变量值
- 改变窗口变量或图象格式设置
- 通过选择 **ClrDraw** 清除绘图
- 改变统计图定义

### 覆盖图象上的函数

TI-83 Plus 可以对一个或多个新函数绘图而不用重新绘制已有函数。例如，在  $Y=$  编辑器中存储  $\sin(X)$  到  $Y_1$  后按  $\boxed{\text{GRAPH}}$ ，然后存储  $\cos(X)$  到  $Y_2$  再按  $\boxed{\text{GRAPH}}$ 。函数  $Y_2$  被绘制在初始函数  $Y_1$  之上。



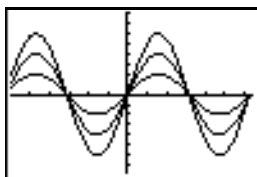
### 绘制一族曲线

如果输入数组 (第 11 章) 作为表达式中的元素，那么 TI-83 Plus 对数组中每个值绘制函数，因此绘制了一族曲线。在 **Simul** 顺序绘图方式下，对每个数组中第一个元素依次绘制所有函数，然后第二个元素，依此类推。

$\{2,4,6\}\sin(X)$  绘制三个函数:  $2 \sin(X)$ 、 $4 \sin(X)$  和  $6 \sin(X)$ 。

```

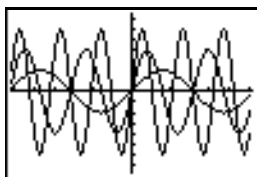
Plot1 Plot2 Plot3
Y1  $\boxed{\{2,4,6\}}\sin(X)$ 
Y2 =
Y3 =
Y4 =
Y5 =
Y6 =
    
```



$\{2,4,6\}\sin(\{1,2,3\}X)$  绘制  $2 \sin(X)$ 、 $4 \sin(2X)$  和  $6 \sin(3X)$ 。

```

Plot1 Plot2 Plot3
Y1  $\boxed{\{2,4,6\}}\sin(\{1,2,3\}X)$ 
Y2 =
Y3 =
Y4 =
Y5 =
Y6 =
    
```



**注意：** 如果使用多个数组，则数组的维数必须相同。



## 自由移动光标

当显示图象后，按  $\leftarrow$ 、 $\rightarrow$ 、 $\uparrow$  或  $\downarrow$  在图象周围移动光标。最初显示图象时，光标不可见。当按  $\leftarrow$ 、 $\rightarrow$ 、 $\uparrow$  或  $\downarrow$  时，光标从观察窗口中心开始移动。

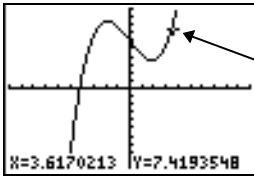
当光标在图象周围移动时，如果选择 **CoordOn** 格式，其位置坐标值显示在屏幕底部。 **Float/Fix** 十进制方式设置决定用于显示坐标值的小数位数。

要显示不带光标和坐标值的图象，请按 **CLEAR** 或 **ENTER**。当按  $\leftarrow$ 、 $\rightarrow$ 、 $\uparrow$  或  $\downarrow$  时，光标从相同位置移动。

## 图象精度

自由移动光标在屏幕上按像素移动。当移动光标到好象在函数上的像素时，光标可能只是靠近，而不是真正在函数上。显示在屏幕底部的坐标值也可能不是函数上的点。要沿函数移动光标，请使用 **TRACE**（页 3-18）。

随光标移动显示的坐标值近似于实际数学坐标值，精度由像素宽度和高度决定。当 **Xmin**、**Xmax**、**Ymin** 和 **Ymax** 值相互靠近（和在 **Zoom In** 中一样）时绘图精度提高，坐标值更近似于数学坐标值。



曲线“上”的自由移动光标

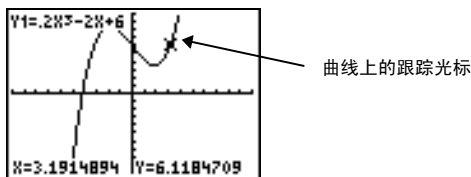
## 开始跟踪

使用 TRACE 沿函数按绘图点移动光标。按 **TRACE** 开始跟踪。如果图象还没显示，按 **TRACE** 显示它。跟踪光标出现在 Y= 编辑器中所选的第一个函数上，位于屏幕上 X 值的中点处。如果选择的是 **CoordOn** 格式，光标坐标值显示在屏幕底部；如果选择的是 **ExprOn** 格式，Y= 表达式显示在屏幕左上角。

## 移动跟踪光标

要移动 TRACE 光标...	这样做:
到前一个或下一个绘图点，	按 <b>◀</b> 或 <b>▶</b> 。
函数上五个绘图点（受 Xres 影响），	按 <b>2nd</b> <b>◀</b> 或 <b>2nd</b> <b>▶</b> 。
到函数任意有效 X 值，	输入该值，然后按 <b>ENTER</b> 。
从一个函数移动到另一个函数，	按 <b>▲</b> 或 <b>▼</b> 。

当跟踪光标沿函数移动时，Y 值由 X 值计算得到，也就是  $Y=Y_n(X)$ 。如果函数在某个 X 值处无定义，Y 值空缺。



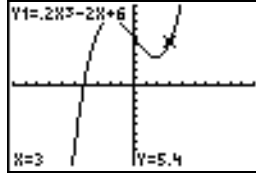
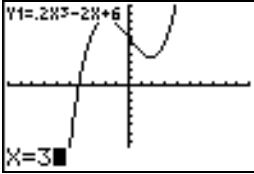
如果跟踪光标移到屏幕顶部或底部之外，屏幕底部的坐标值继续相应改变。

## 移动跟踪光标到另一个函数

按 **▼** 和 **▲** 移动跟踪光标到另一个函数。光标按 Y= 编辑器中所选函数的顺序移动。跟踪光标移动到每个函数的相同 X 值处。如果选择 **ExprOn** 格式，显示的表达式被更新。

### 移动跟踪光标到任何有效 X 值处

要移动跟踪光标到当前函数上任何有效 X 值处，输入该值。当输入第一位数字后，在屏幕左下角显示 **X=** 提示和输入的数字。可以在 **X=** 提示处输入表达式。输入值对于当前观察窗口必须有效。当完成输入后，按 **[ENTER]** 移动光标。



**注意：**此特性不适用于统计图。

### 向左或向右平移

如果跟踪的函数超出屏幕的左边界或右边界，观察窗口自动向左或向右平移。**Xmin** 和 **Xmax** 更新以对应于新的观察窗口。

### 快速缩放

在跟踪时，可以按 **[ENTER]** 调整观察窗口以便光标位置在新观察窗口的中心，即使光标在显示的上方或下方。也允许向上或向下平移。在快速缩放后，光标仍为 **TRACE** 方式。

### 退出和返回 TRACE 方式

当退出和返回 **TRACE** 方式时，跟踪光标显示在离开 **TRACE** 方式时所处的相同位置，除非便捷绘图已经重新绘制了图象（页 3-15）。

### 在程序里使用 TRACE

在程序编辑器中的空行上，按 **[TRACE]**。指令 **Trace** 粘贴到光标位置。当程序执行到此指令时，显示图象且跟踪光标出现在第一个所选函数上。光标坐标值随着跟踪过程而更新。跟踪完函数后，按 **[ENTER]** 恢复程序执行。

## ZOOM 菜单

要显示 ZOOM 菜单, 请按 [ZOOM]。可以用几种方法快速调整观察窗口。所有 ZOOM 指令在程序中都可以用。

### ZOOM MEMORY

1: ZBox	绘制定义观察窗口的矩形。
2: Zoom In	放大光标附近图象。
3: Zoom Out	观察光标附近更多的图象。
4: ZDecimal	设置 $\Delta X$ 和 $\Delta Y$ 为 0.1。
5: ZSquare	在 X 和 Y 轴上设置相同大小像素。
6: ZStandard	设置标准窗口变量。
7: ZTrig	设置内置三角窗口变量。
8: ZInteger	设置 X 和 Y 轴上整数数值。
9: ZoomStat	为当前统计数组设置值。
0: ZoomFit	使 YMin 和 Ymax 落在 XMin 和 Xmax 之间。

## Zoom Cursor

当选择 1:ZBox、2:Zoom In 或 3:Zoom Out 时, 图象上的光标变成缩放光标 (+), 即自由移动光标 (+) 的缩小形状。

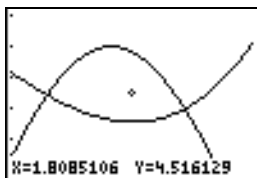
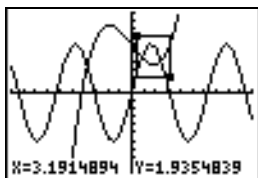
## ZBox

执行以下步骤, 使用 ZBox 定义新观察窗口:

1. 从 ZOOM 菜单选择 1:ZBox。缩放光标显示在屏幕中心。
2. 将缩放光标移动到要定义为矩形框角点的位置, 然后按 [ENTER]。当光标移开第一个定义的角点后, 一个小方点表示此角点。
3. 按 [←]、[↑]、[→] 或 [↓]。随着光标的移动, 矩形框边长在屏幕上按比例加长或变短。

**注意:** 要在按 [ENTER] 之前取消 ZBox, 请按 [CLEAR]。

4. 定义了矩形框后, 按 [ENTER] 重新绘图。



要用 ZBox 在新图象中定义另一个矩形框, 重复步骤 2 到 4。要取消 ZBox, 按 [CLEAR]。

---

## Zoom In、Zoom Out

**Zoom In** 放大光标位置周围的图象部分。**Zoom Out** 显示以光标位置为中心的图象更多部分。**XFact** 和 **YFact** 设置决定缩放程度。

要在图象上放大，执行以下步骤：

1. 检查 **XFact** 和 **YFact**（页 3-24）；必要时改变。
2. 选择 **ZOOM** 菜单中的 **2:Zoom In**。缩放光标显示。
3. 移动缩放光标到新观察窗口的中心点。
4. 按 **[ENTER]**。**TI-83 Plus** 以光标位置为中心按 **XFact** 和 **YFact** 调整观察窗口、更新窗口变量和重新绘制所选函数。
5. 用下面两种方法之一在图象上再次放大。
  - 要在同一点放大，请按 **[ENTER]**。
  - 要在新的点放大，请移动光标到要使之成为新观察窗口中心的点，然后按 **[ENTER]**。

要在图象上缩小，请选择 **3:Zoom Out** 并重复步骤 3 到 5。

要取消 **Zoom In** 或 **Zoom Out**，请按 **[CLEAR]**。

## ZDecimal

**ZDecimal** 立即重绘函数。它更新窗口变量为预先设置值，如下所示。这些值设置  $\Delta X$  和  $\Delta Y$  为 **0.1** 并设置每个像素的 **X** 和 **Y** 值为一位小数位。

**Xmin=-4.7**

**Ymin=-3.1**

**Xmax=4.7**

**Ymax=3.1**

**Xscl=1**

**Yscl=1**

## ZSquare

**ZSquare** 立即重绘函数。基于窗口变量的当前值，重新定义观察窗口。它只在一个方向上调整以便  $\Delta X = \Delta Y$ ，这使得圆的图象看起来象一个圆。**Xscl** 和 **Yscl** 保持不变。当前图象的中心点（不是坐标轴交点）变成新图象的中心点。

### ZStandard

**ZStandard** 立即重绘函数。它更新窗口变量为下面标准值。

<b>Xmin=-10</b>	<b>Ymin=-10</b>	<b>Xres=1</b>
<b>Xmax=10</b>	<b>Ymax=10</b>	
<b>Xscl=1</b>	<b>Yscl=1</b>	

### ZTrig

**ZTrig** 立即重绘函数。更新窗口变量为适于绘制三角函数图象的预设值。**Radian** 方式下的预设值如下所示。

<b>Xmin=<math>-(47/24)\pi</math></b>	<b>Ymin=-4</b>
<b>Xmax=<math>(47/24)\pi</math></b>	<b>Ymax=4</b>
<b>Xscl=<math>\pi/2</math></b>	<b>Yscl=1</b>

### ZInteger

**ZInteger** 重新定义观察窗口为下面所示的尺寸。要使用 **ZInteger**，移动光标到要使之成为新窗口中心的点，然后按 **[ENTER]**；**ZInteger** 重绘函数。

<b><math>\Delta X=1</math></b>	<b>Xscl=10</b>
<b><math>\Delta Y=1</math></b>	<b>Yscl=10</b>

### ZoomStat

**ZoomStat** 重新定义观察窗口以便显示所有统计数据点。对于标准框图和改进式框图，只调整 **Xmin** 和 **Xmax**。

### ZoomFit

**ZoomFit** 立即重绘函数。**ZoomFit** 重新计算 **YMin** 和 **Ymax** 的值以便包含所选函数在当前 **XMin** 和 **XMax** 之间的最小和最大 **Y** 值。**XMin** 和 **XMax** 不变。

## ZOOM MEMORY 菜单

要显示 ZOOM MEMORY 菜单，请按 **[ZOOM]** **[▶]**。

---

### ZOOM MEMORY

1: Zprevious	使用以前的观察窗口。
2: ZoomSto	存储用户定义的窗口。
3: ZoomRcl	调用用户定义的窗口。
4: SetFactors...	改变 <b>Zoom In</b> 和 <b>Zoom Out</b> 因子。

---

## ZPrevious

**ZPrevious** 用上次执行 ZOOM 指令之前所显示图象的窗口变量值重绘图象。

## ZoomSto

**ZoomSto** 立即存储当前观察窗口。图象显示出来，并且当前窗口变量的值存储到用户定义的 ZOOM 变量 **ZXmin**、**ZXmax**、**ZXscl**、**ZYmin**、**ZYmax**、**ZYscl** 和 **ZXres** 中。

这些变量对所有的图象方式有效。例如，在 **Func** 方式下改变 **ZXmin** 的值则在 **Par** 方式下它也改变了。

## ZoomRcl

**ZoomRcl** 在用户定义的观察窗口内绘制所选函数。用户定义的观察窗口由 **ZoomSto** 指令所存储的值确定。更新窗口变量为用户定义的值，然后画出图象。

### ZOOM FACTORS

缩放因子 **XFact** 和 **YFact** 是大于或等于 1 的正数（不必为整数）。它们定义放大或缩小因子，这些因子用于在某一点 **Zoom In** 或 **Zoom Out**。

#### 检查 XFact 和 YFact

要显示用于检查 **XFact** 和 **YFact** 当前值的 ZOOM FACTORS 屏幕，请从 ZOOM MEMORY 菜单选择 **4:SetFactors**。其中所显示的值为默认值。

```
ZOOM FACTORS
XFact=4
YFact=4
```

#### 改变 XFact 和 YFact

可以用以下两种方法之一改变 **XFact** 和 **YFact** 的值。

- 输入新值。原值在输入第一个数字后自动清除。
- 将光标移到要改变的数字位，然后输入一个值或按 **[DEL]** 删除它。

#### 从主屏幕或程序来使用 ZOOM MEMORY 菜单项

从主屏幕或程序中，可直接存储到任何用户定义的 ZOOM 变量。

```
-5→Zxmin:5→Zxmax
5
```

在程序中可以从 ZOOM MEMORY 菜单选择 **ZoomSto** 和 **ZoomRcl** 指令。



## CALCULATE 菜单

要显示 CALCULATE 菜单, 请按  $\boxed{2nd}$  [CALC]。使用此菜单上的菜单项来分析当前图象函数。

### CALCULATE

1: value	对给定 X 值计算函数 Y 的值。
2: zero	求函数零点 (和 x 轴交点)。
3: minimum	求函数最小值。
4: maximum	求函数最大值。
5: intersect	求两个函数交点。
6: dy/dx	求函数数值导数。
7: $\int f(x)dx$	求函数数值积分。

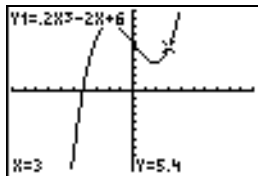
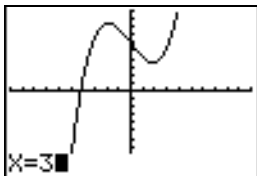
## value

**value** 对给定 X 值计算一个或多个当前所选函数的值。

**注意:** 当显示一个 X 值时, 按  $\boxed{CLEAR}$  清除此值。当无值显示时, 按  $\boxed{CLEAR}$  取消 **value** 运算。

执行以下步骤计算所选函数在 X 处的值:

1. 从 CALCULATE 菜单选择 **1:value**。显示图象且左下角显示 **X=**。
2. 为 X 输入一个在 **Xmin** 和 **Xmax** 之间的实数 (或表达式)。
3. 按  $\boxed{ENTER}$ 。



光标在 Y= 编辑器中所选第一个函数上, 位于刚输入的 X 值处, 而且即使选择了 **CoordOff** 格式, 也显示出坐标。

要在函数间移动光标, 并使光标位于输入的 X 值处, 请按  $\boxed{\leftarrow}$  或  $\boxed{\rightarrow}$ 。要恢复自由移动光标, 请按  $\boxed{\leftarrow}$  或  $\boxed{\rightarrow}$ 。

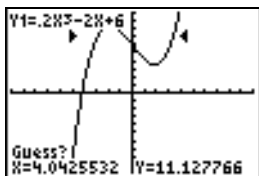
## zero

**zero** 使用 **solve**( 求一个函数的零点 (与  $x$  轴交点或根)。函数与  $x$  轴交点可能不止一个, 但 **zero** 只求与输入的估计值最接近的零点。

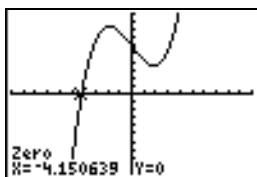
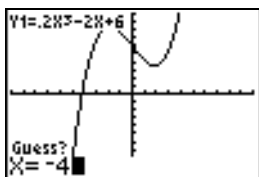
**zero** 求解正确零点所花时间的长短取决于所指定左右边界值的准确性和估计值的准确性。

执行以下步骤求函数零点:

1. 从 CALCULATE 菜单选择 **2:zero**。显示当前图象, 其左下角显示 **Left Bound?** (左边界)。
2. 按  $\blacktriangleleft$  或  $\blacktriangleright$  将光标移到要求其零点的函数上。
3. 按  $\blacktriangleleft$  或  $\blacktriangleright$  (或者输入一个值) 选择区间左边界的  $x$  值, 然后按  $\boxed{\text{ENTER}}$ 。图象屏幕中的指示符  $\blacktriangleright$  表示左边界。 **Right Bound?** 显示在左下角。按  $\blacktriangleleft$  或  $\blacktriangleright$  (或输入值) 来选择右边界的  $x$  值, 然后按  $\boxed{\text{ENTER}}$ 。图象屏幕中的指示符  $\blacktriangleleft$  表示右边界。然后 **Guess?** 显示在左下角。



4. 按  $\blacktriangleleft$  或  $\blacktriangleright$  (或输入值) 选择区间内靠近函数零点的一个点, 然后按  $\boxed{\text{ENTER}}$ 。



即使在选择 **CoordOff** 格式的情况下, 光标也会出现在结果处, 并显示出光标处的坐标值。要将光标移到其他函数的相同  $x$  值处, 请按  $\blacktriangleleft$  或  $\blacktriangleright$ 。要恢复自由移动光标, 请按  $\blacktriangleleft$  或  $\blacktriangleright$ 。

## minimum、maximum

**minimum** 和 **maximum** 求函数在指定区间内的最小值或最大值，误差为  $1E-5$ 。

执行以下步骤求最小值或最大值：

1. 从 CALCULATE 菜单选择 **3:minimum** 或 **4:maximum**。显示当前图象。
2. 选择函数，用 **zero** 中所述方法（步骤 2 到 4；页 3-26）设置左边界、右边界和估计值。

即使选择了 **CoordOff** 格式，光标也会出现在结果处，并显示其坐标值：**Minimum** 或 **Maximum** 显示在左下角。

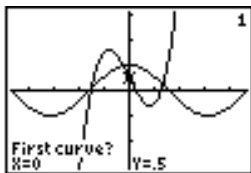
要将光标移到其他函数上的相同  $x$  值处，请按  $\uparrow$  或  $\downarrow$ 。要恢复自由移动光标，请按  $\leftarrow$  或  $\rightarrow$ 。

## intersect

**intersect** 使用 **solve**( 求两个或多个函数交点的坐标值。为使用 **intersect**，交点必须出现在显示屏幕上。

执行以下步骤求交点：

1. 从 CALCULATE 菜单选择 **5:intersect**。显示当前图象且左下角显示 **First curve?**（第一条曲线）。



2. 按  $\downarrow$  或  $\uparrow$ ，如有必要，移动光标到第一个函数，然后按 **ENTER**。**Second curve?** 显示在左下角。
3. 按  $\downarrow$  或  $\uparrow$ ，如有必要，移动光标到第二个函数，然后按 **ENTER**。
4. 按  $\rightarrow$  或  $\leftarrow$  移动光标到估计为交点位置的点，然后按 **ENTER**。

即使选择了 **CoordOff** 格式，光标也会出现在结果处，并显示其坐标值。**Intersection** 显示在左下角。要恢复自由移动光标，请按  $\leftarrow$ 、 $\uparrow$ 、 $\rightarrow$  或  $\downarrow$ 。

## dy/dx

**dy/dx** (数值导数) 求函数在某点的数值导数 (斜率), 误差为  $\epsilon=1E-3$ 。

执行以下步骤求函数在某点的斜率:

1. 从 CALCULATE 菜单选择 **6:dy/dx**, 显示当前图象。
2. 按  $\blacktriangleleft$  或  $\blacktriangleright$  选择要求其数值导数的函数。
3. 按  $\blacktriangleleft$  或  $\blacktriangleright$  (或输入值) 选择 **X** 值, 用于求此点的导数, 然后按 **ENTER**。

光标出现在结果处且显示数值导数值。

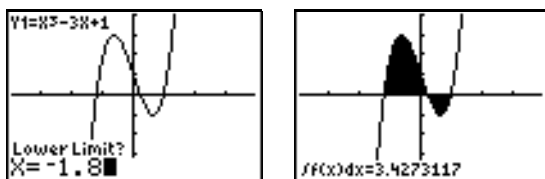
要将光标移到其他函数上的相同 **x** 值处, 请按  $\blacktriangleleft$  或  $\blacktriangleright$ 。要恢复自由移动光标, 请按  $\blacktriangleleft$  或  $\blacktriangleright$ 。

## ∫f(x)dx

**∫f(x)dx** (数值积分) 求函数在指定区间内的数值积分。它使用 **fnInt()** 函数, 误差为  $\epsilon=1E-3$ 。

执行以下步骤求函数的数值积分:

1. 从 CALCULATE 菜单选择 **7:∫f(x)dx**。显示当前图象且左下角处显示 **Lower Limit?**。
2. 按  $\blacktriangleleft$  或  $\blacktriangleright$  移动光标到要计算其积分的函数上。
3. 如同为 **zero** 设置左右边界一样 (步骤 3; 页 3-26) 设置上下限。积分值显示, 积分区域用阴影覆盖。

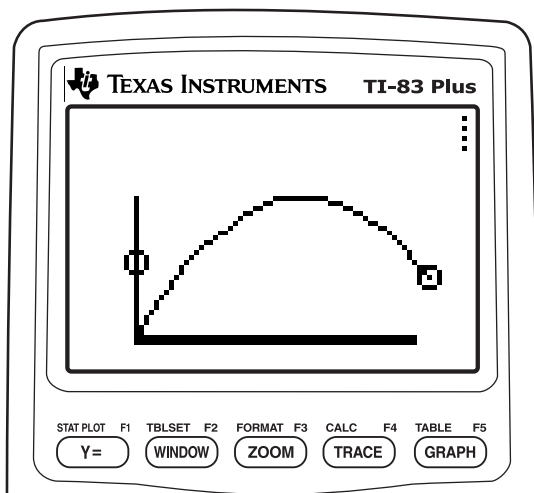


**注意:** 阴影区域是一个图形。使用 **ClrDraw** (第 8 章) 或任何调用便捷绘图的操作清除阴影区域。

# 4 参数 图象

## 目录

入门：小球的轨迹.....	1
定义和显示参数图象.....	4
研究参数图象.....	7



# 入门：小球的轨迹

入门是对本章内容的快速介绍。详细信息请阅读本章。

本节根据小球的参数方程作出它的轨迹图象，小球是以 30 米/秒的初速度及与水平面成 25 度的夹角向上抛出的。忽略除重力之外的所有影响，小球可抛多远？何时落到地面？上升多高？

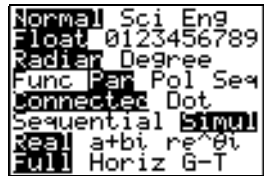
对于初速度  $v_0$  和抛出角度  $\theta$ ，小球的位置是时间的函数，有水平和垂直两个分量。

$$\text{水平: } X1(t)=tv_0\cos(\theta) \quad \text{垂直: } Y1(t)=tv_0\sin(\theta)-\frac{1}{2}gt^2$$

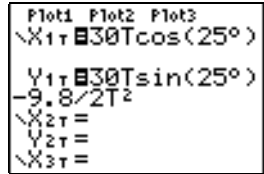
可作出小球运动的水平和垂直向量的图象。

$$\begin{array}{ll} \text{垂直向量:} & X2(t)=0 & Y2(t)=Y1(t) \\ \text{水平向量:} & X3(t)=X1(t) & Y3(t)=0 \\ \text{重力常数:} & & g=9.8 \text{ m/sec}^2 \end{array}$$

1. 按 **MODE**。按  $\downarrow \downarrow \downarrow \rightarrow$  **ENTER** 选择 **Par** 方式。按  $\downarrow \downarrow \rightarrow$  **ENTER** 选择 **Simul** 对本例的三个参数方程同时作图。

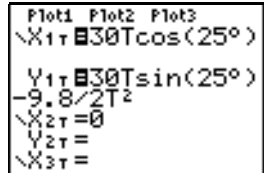


2. 按  $\boxed{Y=}$ 。按 **30**  $\boxed{X,T,\theta,n}$  **COS** **25**  $\boxed{2nd}$  **[ANGLE]** **1** (选择  $^\circ$ )  $\boxed{\rightarrow}$  **ENTER**，根据 **T** 定义 **X1T**。
3. 按 **30**  $\boxed{X,T,\theta,n}$  **SIN** **25**  $\boxed{2nd}$  **[ANGLE]** **1**  $\boxed{\rightarrow}$   $\boxed{\square}$  **9.8**  $\boxed{\div}$  **2**  $\boxed{X,T,\theta,n}$   $\boxed{x^2}$  **ENTER** 定义 **Y1T**。

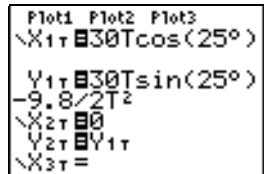


由 **X2T** 和 **Y2T** 定义垂直分量向量。

4. 按 **0** **ENTER** 定义 **X2T**。



5. 按 **VARS**  $\rightarrow$  显示 **VARS Y-VARS** 菜单。按 **2** 显示 **PARAMETRIC** 子菜单。按 **2** **ENTER** 定义 **Y2T**。



由  $X_{3T}$  和  $Y_{3T}$  定义水平分量向量。

6. 按  $\boxed{\text{VARS}} \rightarrow \boxed{2}$ ，然后按  $\boxed{1} \boxed{\text{ENTER}}$  定义  $X_{3T}$ 。按  $\boxed{0} \boxed{\text{ENTER}}$  定义  $Y_{3T}$ 。

```
Plot1 Plot2 Plot3
Y1T=30Tsin(25°)
-9.8/2T²
√X2T=0
Y2T=VY1T
√X3T=XX1T
Y3T=0
√X4T=
```

7. 按  $\boxed{\leftarrow} \boxed{\leftarrow} \boxed{\leftarrow} \boxed{\text{ENTER}}$  将  $X_{3T}$  和  $Y_{3T}$  的图象样式改变为  $\leftarrow$ 。按  $\boxed{\leftarrow} \boxed{\text{ENTER}} \boxed{\text{ENTER}}$  将  $X_{2T}$  和  $Y_{2T}$  的图象样式改变为  $\leftarrow$ 。按  $\boxed{\leftarrow} \boxed{\text{ENTER}} \boxed{\text{ENTER}}$  将  $X_{1T}$  和  $Y_{1T}$  的图象样式改变为  $\leftarrow$ 。（以上击键假定所有初始的图象样式设置为  $\leftarrow$ 。）

```
Plot1 Plot2 Plot3
←X1T=30Tcos(25°)
Y1T=30Tsin(25°)
-9.8/2T²
←X2T=0
Y2T=VY1T
←X3T=XX1T
```

8. 按  $\boxed{\text{WINDOW}}$ 。输入窗口变量的值。

Tmin=0      Xmin=-10      Ymin=-5  
Tmax=5      Xmax=100      Ymax=15  
Tstep=.1    Xscl=50      Yscl=10

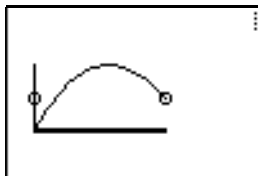
```
WINDOW
↑Tstep=.1
Xmin=-10
Xmax=100
Xscl=50
Ymin=-5
Ymax=15
Yscl=10
```

9. 按  $\boxed{2\text{nd}} \boxed{\text{FORMAT}} \rightarrow \boxed{\downarrow} \rightarrow \boxed{\downarrow} \rightarrow \boxed{\downarrow} \rightarrow \boxed{\downarrow} \rightarrow \boxed{\downarrow} \rightarrow \boxed{\downarrow} \rightarrow \boxed{\downarrow} \rightarrow \boxed{\downarrow} \rightarrow \boxed{\text{ENTER}}$  设置 **AxesOff**，关闭坐标轴。

```
rectGC PolarGC
CoordOn CoordOff
GridOff GridOn
HxesOn HxesOff
LabelOff LabelOn
ExprOn ExprOff
```

10. 按  $\boxed{\text{GRAPH}}$ 。图象同步显示小球的飞行轨迹和运动的垂直及水平分量向量。

**提示：**若要模拟小球在空中的飞行，将  $X_{1T}$  和  $Y_{1T}$  的图象样式设置为  $\leftarrow$ （动画）。



11. 按  $\boxed{\text{TRACE}}$  获得数值结果，这样就回答了本节开始部分的问题。

对第一个参数方程 ( $X_{1T}$  和  $Y_{1T}$ ) 的跟踪起始于 **Tmin**。如果按  $\boxed{\rightarrow}$  来跟踪曲线，则光标沿小球对时间轨迹运动。**X** (距离)、**Y** (高度) 和 **T** (时间) 的值显示在屏幕底部。

```
X1T=30Tcos(25°) Y1T=30Tsin(25°)
T=.7
X=19.032464 Y=6.4739835
```

## TI-83 Plus 图象方式的相似性

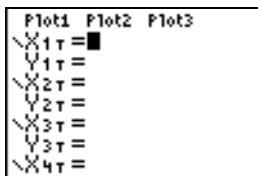
定义参数图象的步骤与定义函数图象的步骤相似。本章假定您已熟悉第3章：函数图象。第4章详细描述了参数图象与函数图象的不同之处。

### 设置参数图象方式

按 **[MODE]** 显示方式屏幕。若要根据参数方程作图，必须在输入窗口变量和输入参数方程的分量之前选择 **Par** 图象方式。

### 显示参数 Y= 编辑器

选择 **Par** 图象方式之后，按 **[Y=]** 显示参数 Y= 编辑器。



在此编辑器中，最多可以显示并输入六个方程的 **X** 和 **Y** 分量，即 **X1T** 和 **Y1T** 到 **X6T** 和 **Y6T**，每个分量由自变量 **T** 定义。参数图象的应用一般是作随时间变化的方程图象。

### 选择图象样式

**X1T** 到 **X6T** 左边的图标表示每个参数方程的图象样式（第3章）。**Par** 方式下，默认的图象样式为 **—**（实线），即绘图点的连接。实线、**—**（粗线）、**⋯**（轨迹）、**⋯**（动画）和 **·**（点）样式为参数图象的可用样式。



---

## 定义和编辑参数方程

若要定义或编辑参数方程，执行步骤与第 3 章的定义或编辑函数的步骤相似。参数方程的自变量为 **T**。在 **Par** 图象方式下，可用两种方法之一输入参数变量 **T**：

- 按  $[X, T, \theta, n]$ 。
- 按  $[ALPHA] [T]$ 。

两个分量，**X** 和 **Y**，定义了一个独立的参数方程。必须对这两个分量都进行定义。

## 选择和取消选择参数方程

TI-83 Plus 仅作出选定参数方程的图象。在 **Y=** 编辑器中，当 **X** 和 **Y** 分量的 **=** 号都被亮显时，此参数方程被选定。可选定任一个或所有的方程，从 **X1T** 和 **Y1T** 到 **X6T** 和 **Y6T**。

若要更改选择状态，将光标移到 **X** 或 **Y** 分量的 **=** 号上并按  $[ENTER]$ ，则 **X** 和 **Y** 分量的状态都被改变。

## 设置窗口变量

若要显示窗口变量值，按  $[WINDOW]$ 。这些变量定义了观察窗口。以下值为 **Par** 图象方式使用 **Radian** 角度方式时的默认变量值。

---

<b>Tmin</b> =0	用于计算的最小 <b>T</b> 值
<b>Tmax</b> =6.2831853...	用于计算的最大 <b>T</b> 值 ( $2\pi$ )
<b>Tstep</b> =.1308996...	<b>T</b> 值的增量 ( $\pi/24$ )
<b>Xmin</b> =-10	可显示的最小 <b>X</b> 值
<b>Xmax</b> =10	可显示的最大 <b>X</b> 值
<b>Xscl</b> =1	<b>X</b> 分段标记之间的间隔
<b>Ymin</b> =-10	可显示的最小 <b>Y</b> 值
<b>Ymax</b> =10	可显示的最大 <b>Y</b> 值
<b>Yscl</b> =1	<b>Y</b> 分段标记之间的间隔

---

**注意：**若要确保有足够多的点用于绘图，可更改 **T** 窗口变量。

## 设置图象格式

若要显示当前图象格式设置，请按  $\boxed{2nd}$  [FORMAT]。第 3 章详细描述了格式设置。其它图象方式也使用这些格式设置，**Seq** 图象方式有一个有关附加坐标轴的格式设置。

## 显示图象

按  $\boxed{GRAPH}$  时，TI-83 Plus 绘出选定参数方程的图象。它计算对应于每个 **T** 值（从 **Tmin** 到 **Tmax**，时间间隔为 **Tstep**）的 **X** 和 **Y** 分量，然后绘出由 **X** 和 **Y** 定义的每个点。窗口变量定义了观察窗口。

绘图过程中，**X**、**Y** 和 **T** 不断更新。

便捷绘图可应用于参数图象（第 3 章）。

## 窗口变量和 Y-VARS 菜单

在主屏幕或程序中可执行下列操作：

- 将方程的 **X** 或 **Y** 分量名作为变量以进入函数。

```

X1T*.5
  94.70916375
    
```

- 存储参数方程。

<pre> "sin(T)"→X1T Done "cos(T)"→Y1T Done         </pre>	<pre> Plot1 Plot2 Plot3 \X1T sin(T)  Y1T cos(T) \X2T =  Y2T =         </pre>
--	--

- 选择或取消选择参数方程。

<pre> FnOff 1 Done         </pre>	<pre> Plot1 Plot2 Plot3 \X1T =cos(T)  Y1T =sin(T) \X2T =  Y2T =         </pre>
-----------------------------------	--

- 直接将值存储到窗口变量。

```

360→Tmax
  360
    
```

### 自由移动光标

**Par** 图象和 **Func** 图象中的自由移动光标操作相同。

在 **RectGC** 格式中，移动光标可更新 **X** 和 **Y** 的值；如果选择 **CoordOn** 格式，则显示 **X** 和 **Y**。

在 **PolarGC** 格式中，**X**、**Y**、**R** 和  **$\theta$**  被更新；如果选择 **CoordOn** 格式，则显示 **R** 和  **$\theta$** 。

### TRACE

若要激活 **TRACE**，按 **TRACE**。当 **TRACE** 被激活，可沿着方程的图象移动跟踪光标，一次移动一个 **Tstep**。开始跟踪时，跟踪光标位于第一个选定函数的 **Tmin** 位置。如果选择 **ExprOn**，则显示此函数。

在 **RectGC** 格式中，如果选择 **CoordOn** 格式，则 **TRACE** 更新并显示 **X**、**Y** 和 **T** 的值。

在 **PolarGC** 格式中，更新 **X**、**Y**、**R**、 **$\theta$**  和 **T**；如果选择 **CoordOn** 格式，则显示 **R**、 **$\theta$**  和 **T**。**X** 和 **Y**（或 **R** 和  **$\theta$** ）值是根据 **T** 计算出来的。

对某个函数若要一次移动 5 个绘图点，按 **2nd** **↓** 或 **2nd** **↑**。如果将光标移出屏幕顶端或底部，则屏幕底部的坐标值继续作相应变化。

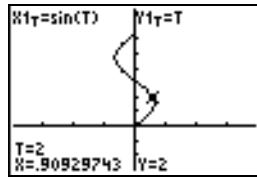
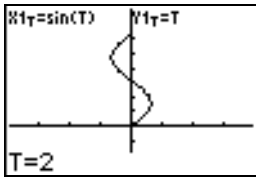
在 **Par** 图象中可使用快速缩放；但不可平移（第 3 章）。

移动跟踪光标到任意有效 T 值

对当前函数若要将跟踪光标移动到任意有效的 **T** 值，可输入此值。当输入第一个数字时，在屏幕左下角显示一个 **T=** 提示和刚输入的数字。可在 **T=** 提示处输入一个表达式。输入值必须是对当前观察窗口有效的。输入完成后，按 **[ENTER]** 移动光标。

```

Plot1 Plot2 Plot3
X1T=sin(T)
Y1T=T
    
```



ZOOM

**Par** 图象和 **Func** 图象中的 ZOOM 操作相同。仅影响 **X** (**Xmin**、**Xmax** 和 **Xscl**) 和 **Y** (**Ymin**、**Ymax** 和 **Yscl**) 窗口变量。

**T** 窗口变量 (**Tmin**、**Tmax** 和 **Tstep**) 仅在选择 **ZStandard** 时受影响。**VARS ZOOM** 子菜单的 **ZT/Z0** 菜单项中 **1:ZTmin**、**2:ZTmax** 和 **3:ZTstep** 是用于 **Par** 图象的 zoom memory 变量。

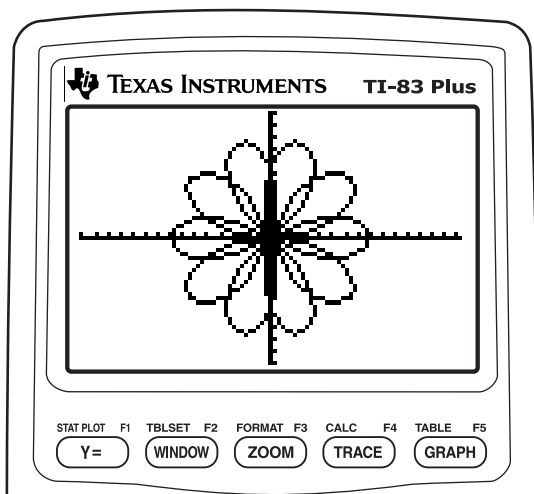
CALC

**Par** 图象和 **Func** 图象中的 CALC 操作相同。用于 **Par** 图象的 CALCULATE 菜单项为 **1:value**、**2:dy/dx**、**3:dy/dt** 和 **4:dx/dt**。

# 5 极坐标 图象

## 目录

入门：极坐标玫瑰线.....	2
定义和显示极坐标图象.....	3
研究极坐标图象.....	6



## 入门：极坐标玫瑰线

入门是对本章内容的快速介绍。详细信息请阅读本章。

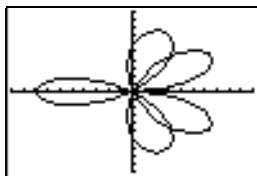
极坐标方程  $R=Asin(B\theta)$  的图象是玫瑰线。本节根据这个方程作出该图象，其中  $A=8$ ， $B=2.5$ ，然后研究使用其他  $A$  和  $B$  值时的玫瑰线形状。

1. 按 **MODE** 显示方式屏幕。按  $\downarrow \downarrow \downarrow \rightarrow \rightarrow$  **ENTER** 选择 **Pol** 图象方式。其他方式设置则选择默认值（左边的选项）。

```
Plot1 Plot2 Plot3
√r1 8sin(2.5θ)
√r2 =
√r3 =
√r4 =
√r5 =
√r6 =
```

2. 按 **Y=** 显示极坐标  $Y=$  编辑器。按 **8 SIN 2.5** **X,T,θ,n** **ENTER** 定义  $r_1$ 。

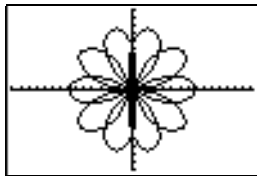
3. 按 **ZOOM** **6** 选择 **6:ZStandard**，这时在标准观察窗口显示出方程的图象。图象仅显示玫瑰线的 5 个花瓣，并且不是对称显示。这是因为标准窗口设置  $\theta_{\max}=2\pi$  并且定义窗口（而不是象素）为正方形。



4. 按 **WINDOW** 显示窗口变量。按  $\downarrow$  **4** **2nd** **[π]** 增加  $\theta_{\max}$  的值到  $4\pi$ 。

```
WINDOW
θmin=0
θmax=4π
θstep=.1308996...
Xmin=-10
Xmax=10
Xscl=1
↓Ymin=-10
```

5. 按 **ZOOM** **5** 选择 **5:ZSquare** 并绘出图象。



6. 重复步骤 2 到 5，在极坐标方程  $r_1=Asin(B\theta)$  中使用新的  $A$  和  $B$  值。观察新值如何影响图象。

## TI-83 Plus 图象方式的相似性

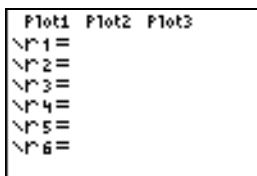
定义极坐标图象的步骤与定义函数图象的步骤相似。本章假定您已熟悉第 3 章：函数图象。第 5 章详细描述了极坐标图象与函数图象的不同之处。

## 设置极坐标图象方式

按 **[MODE]** 显示方式屏幕。若要根据极坐标方程作图，必须在输入窗口变量值和极坐标方程之前选择 **Pol** 图象方式。

## 显示极坐标 Y= 编辑器

选择 **Pol** 图象方式之后，按 **[Y=]** 显示极坐标 Y= 编辑器。



在此编辑器中，最多可以显示并输入六个极坐标方程，**r1** 到 **r6**。根据自变量  $\theta$  来定义每个方程（页 5-4）。

## 选择图象样式

**r1** 到 **r6** 左边的图标表示每个极坐标方程的图象样式（第 3 章）。**Pol** 方式下，默认的图象样式为  $\backslash$ （实线），即绘图点的连线。实线、 $\equiv$ （粗线）、 $\dot{\curvearrowright}$ （轨迹）、 $\dot{\curvearrowleft}$ （动画）和  $\cdot$ （点）样式为极坐标图象的可用样式。

### 定义和编辑极坐标方程

若要定义或编辑极坐标方程，请执行第 3 章中定义或编辑函数的步骤。极坐标方程的自变量为  $\theta$ 。在 **Pol** 图象方式下，可用下述两种方法之一输入极坐标变量  $\theta$ ：

- 按  $[X,T,\theta,n]$ 。
- 按  $[\text{ALPHA}][\theta]$ 。

### 选择和取消选择极坐标方程

TI-83 Plus 仅作出所选极坐标方程的图象。在 **Y=** 编辑器中，当 **=** 符号亮显时，此极坐标方程被选定。可选定任一个或所有的方程。

若要更改选择状态，请将光标移到 **=** 符号上并按  $[\text{ENTER}]$ 。

### 设置窗口变量

若要显示窗口变量值，请按  $[\text{WINDOW}]$ 。这些变量定义了观察窗口。下表中的值为以 **Radian** 角度方式绘制 **Pol** 图象时的默认值。

---

$\theta_{\min}=0$	用于计算的最小 $\theta$ 值
$\theta_{\max}=6.2831853\dots$	用于计算的最大 $\theta$ 值 ( $2\pi$ )
$\theta_{\text{step}}=.1308996\dots$	$\theta$ 值的增量 ( $\pi/24$ )
$X_{\min}=-10$	可显示的最小 <b>X</b> 值
$X_{\max}=10$	可显示的最大 <b>X</b> 值
$X_{\text{scl}}=1$	<b>X</b> 分段标记之间的间隔
$Y_{\min}=-10$	可显示的最小 <b>Y</b> 值
$Y_{\max}=10$	可显示的最大 <b>Y</b> 值
$Y_{\text{scl}}=1$	<b>Y</b> 分段标记之间的间隔

---

**注意：**要确保绘制足够多的点，可更改  $\theta$  窗口变量。



## 设置图象格式

若要显示当前图象格式设置，请按  $\boxed{2nd}$  [FORMAT]。第 3 章详细介绍了格式设置。其他图象方式也可用这些格式设置。

## 显示图象

按  $\boxed{GRAPH}$  时，TI-83 Plus 绘出所选极坐标方程的图象。它计算对应于每个  $\theta$  值（从  $\theta_{min}$  到  $\theta_{max}$ ，间隔为  $\theta_{step}$ ）的  $R$ ，然后绘出每个点。窗口变量定义了观察窗口。

绘图过程中， $X$ 、 $Y$ 、 $R$  和  $\theta$  不断更新。

便捷绘图可应用于极坐标图象（第 3 章）。

## 窗口变量和 Y-VARS 菜单

在主屏幕或程序中可执行下列操作：

- 将方程名作为变量来进入函数。

```
r1+r2      8
```

- 存储极坐标方程。

```
"5θ"→r1      Done      Plot1 Plot2 Plot3
                               \r1=5θ
                               \r2=
```

- 选择或取消选择极坐标方程。

```
FnOfff 1      Done      Plot1 Plot2 Plot3
                               \r1=5θ
                               \r2=
```

- 直接将值存储到窗口变量。

```
θ→θmin      0
```

## 自由移动光标

**Pol** 图象和 **Func** 图象中自由移动光标的操作相同。在 **RectGC** 格式中，移动光标可更新 **X** 和 **Y** 的值；如果选择 **CoordOn** 格式，则显示 **X** 和 **Y**。在 **PolarGC** 格式中，**X**、**Y**、**R** 和  $\theta$  被更新；如果选择 **CoordOn** 格式，则显示 **R** 和  $\theta$ 。

## TRACE

若要激活 TRACE，请按 **TRACE**。当 TRACE 被激活，可沿着方程的图象移动跟踪光标，一次移动一个  $\theta$ step。开始跟踪时，跟踪光标位于第一个选择的函数，对应于  $\theta$ min。如果选择 **ExprOn** 格式，则显示此方程。

在 **RectGC** 格式中，TRACE 更新 **X**、**Y** 和  $\theta$  的值；如果选择 **CoordOn** 格式，则显示 **X**、**Y** 和  $\theta$ 。在 **PolarGC** 格式中，TRACE 更新 **X**、**Y**、**R** 和  $\theta$ ；如果选择 **CoordOn** 格式，则显示 **R**、 $\theta$ 。

对某个函数若要一次移动 5 个绘图点，请按 **2nd** **↓** 或 **2nd** **↑**。如果将跟踪光标移出屏幕顶端或底部，则屏幕底部的坐标值继续作相应变化。

在 **Pol** 图象方式中可使用快速缩放，但不可平移（第 3 章）。

## 移动跟踪光标到任意有效 $\theta$ 值

对当前函数若要将跟踪光标移动到任意有效的  $\theta$  值，可输入此  $\theta$  值。当输入第一个数字时，在屏幕左下角显示一个  $\theta=$  提示和输入的数字。可在  $\theta=$  提示处输入一个表达式。输入值必须是对当前观察窗口有效的。输入完成后，按 **ENTER** 来移动光标。

## ZOOM

**Pol** 图象和 **Func** 图象中的 ZOOM 操作相同。仅影响 **X**（**Xmin**、**Xmax** 和 **Xscl**）和 **Y**（**Ymin**、**Ymax** 和 **Yscl**）窗口变量。

$\theta$  窗口变量（ $\theta$ min、 $\theta$ max 和  $\theta$ step）仅在选择 **ZStandard** 时受影响。VARS ZOOM 子菜单 ZT/Z $\theta$  项 **4:Z $\theta$ min**、**5:Z $\theta$ max** 和 **6:Z $\theta$ step** 是用于 **Pol** 图象的 zoom memory 变量。

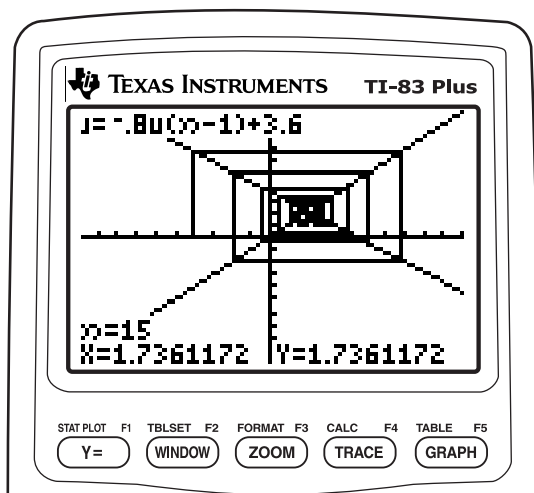
## CALC

**Pol** 图象和 **Func** 图象中的 CALC 操作相同。用于 **Pol** 图象的 CALCULATE 菜单项为 **1:value**、**2:dy/dx** 和 **3:dr/d $\theta$** 。

# 6 序列 图象

## 目录

入门：森林和树木.....	2
定义和显示序列图象.....	3
选定坐标轴组合.....	8
研究序列图象.....	9
作 Web 图.....	11
用 Web 图举例说明收敛性.....	12
作相位图.....	13
比较 TI-83 Plus 和 TI-82 序列变量.....	15
TI-83 Plus 和 TI-82 的击键差别.....	16



# 入门：森林和树木

入门是对本章内容的快速介绍。详细信息请阅读本章。

现要对一个有 4,000 棵树的小森林进行一个新的林业规划。每年将砍伐 20% 的树并且种植 1000 棵新树。这个森林最后会消失吗？森林的规模能稳定下来吗？如果能的话，那么多少年后稳定下来？那时有多少棵树？

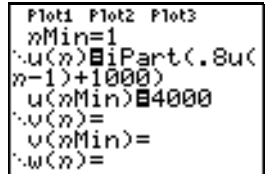
1. 按 **[MODE]**。按  $\downarrow \downarrow \downarrow \rightarrow \rightarrow \rightarrow \rightarrow$  **[ENTER]** 选定 **Seq** 图象方式。



2. 按 **[2nd]** **[FORMAT]**，必要时选择 **Time** 坐标轴格式和 **ExprOn** 格式。



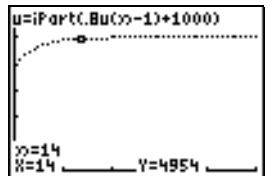
3. 按 **[Y=]**。如果图象样式图标不是  $\cdot$ （点），则按  $\leftarrow$   $\leftarrow$ ，按 **[ENTER]** 直到显示  $\cdot$ ，然后按  $\rightarrow \rightarrow$ 。
4. 由于被砍伐的是整棵树，按 **[MATH]**  $\rightarrow$  **3** 选定 **iPart**（整数部分）。每年度砍伐后，有 80% (.80) 的树保留下来。按  $\leftarrow$  **8** **[2nd]** **[u]**  $\leftarrow$  **[X,T,θ,n]**  $\leftarrow$  **1**  $\leftarrow$  定义每次砍伐后树的棵数。按  $\leftarrow$  **1000**  $\leftarrow$  定义新种树的棵数。按  $\leftarrow$  **4000** 定义计划开始时树的棵数。



5. 按 **[WINDOW]** **0** 设置 **nMin=0**。按  $\leftarrow$  **50** 设置 **nMax=50**。**nMin** 和 **nMax** 计算 50 年内的森林规模。设置其他窗口变量。

**PlotStart=1**    **Xmin=0**    **Ymin=0**  
**PlotStep=1**    **Xmax=50**    **Ymax=6000**  
                  **Xscl=10**    **Yscl=1000**

6. 按 **[TRACE]**。从 **nMin** 处（林业规划的起始处）开始跟踪。按  $\leftarrow$  以年为单位跟踪此序列。序列显示在屏幕的顶端。**n**（年数）、**X**（**X=n**，因为 **n** 对应于 **x** 轴）和 **Y**（树的总数）的值显示在屏幕底部。这个森林何时稳定？稳定在多少棵树？



### TI-83 Plus 图象方式的相似性

定义序列图象的步骤与定义函数图象的步骤相似。本章假定您已熟悉第 3 章：函数图象。第 6 章详细描述了序列图象与函数图象的不同之处。

### 设置序列图象方式

要显示方式屏幕，请按 **MODE**。若要根据序列函数作图，必须在输入窗口变量和输入序列函数之前选择 **Seq** 图象方式。

序列图象自动使用 **Simul** 方式作图，而不管当前的绘图顺序方式设置是什么。

### TI-83 Plus 序列函数 $u$ 、 $v$ 和 $w$

TI-83 Plus 有 3 个可从键盘输入的序列函数： $u$ 、 $v$  和  $w$ 。它们位于 **7**、**8** 和 **9** 键之上。

可根据以下项定义序列函数：

- 自变量  $n$
- 序列函数的前一项，例如  $u(n-1)$
- 序列函数位于前一项之前的项，例如  $u(n-2)$
- 另一序列函数的前一项或位于前一项之前的项，例如  $u(n-1)$  或  $u(n-2)$ ，可在序列  $v(n)$  中被引用。

**注意：**本章中关于  $u(n)$  的描述也适用于  $v(n)$  和  $w(n)$ ；关于  $u(n-1)$  的描述也适用于  $v(n-1)$  和  $w(n-1)$ ；关于  $u(n-2)$  的描述也适用于  $v(n-2)$  和  $w(n-2)$ 。

### 显示序列 Y= 编辑器

选择 **Seq** 图象方式后, 按  $\boxed{Y=}$  显示序列 Y= 编辑器。

```
Plot1 Plot2 Plot3
nMin=1
·u(n)=
u(nMin)=
·v(n)=
v(nMin)=
·w(n)=
w(nMin)=
```

在此编辑器中, 可以显示和输入  $u(n)$ 、 $v(n)$  和  $w(n)$  序列。也可编辑  $nMin$  的值, 即定义用于计算的最小  $n$  值的序列窗口变量。

序列 Y= 编辑器显示  $nMin$  值, 因为它与序列方程  $u(n)$ 、 $v(n)$  和  $w(n)$  的初始值, 即  $u(nMin)$ 、 $v(nMin)$  和  $w(nMin)$  相关。

Y= 编辑器中的  $nMin$  与窗口编辑器中的  $nMin$  相同。如果在一个编辑器中输入一个新的  $nMin$  值, 则这两个编辑器中的  $nMin$  值同时更新。

**注意:**  $u(nMin)$ 、 $v(nMin)$  或  $w(nMin)$  仅用于要求有初始值的递归序列中。

### 选择图象样式

$u(n)$ 、 $v(n)$  和  $w(n)$  左边的图标表示每个序列方程的图象样式 (第 3 章)。在 **Seq** 方式下, 默认的图象样式为  $\cdot$  (点), 即显示离散值。点、 $\text{—}$  (实线) 和  $\text{—}$  (粗线) 样式为序列图象的可用样式。在 **Web** 格式中图象样式被忽略。

### 选择和取消选择序列函数

TI-83 Plus 仅作出所选序列函数的图象。在 Y= 编辑器中, 当  $u(n)=$  和  $u(nMin)=$  的 = 符号都亮显时, 此序列函数被选择。

若要更改序列函数的选择状态, 请将光标移到函数名的 = 符号上并按  $\boxed{\text{ENTER}}$ , 则序列函数  $u(n)$  和它的初始值  $u(nMin)$  的状态都被改变。

## 定义和编辑序列函数

若要定义或编辑序列函数，其操作步骤与第 3 章的定义函数的步骤相似。序列函数的自变量为  $n$ 。

在 **Seq** 图象方式下，有两种方法输入序列变量。

- 按  $\overline{\text{X.T.}\Theta.n}$ 。
- 按  $\overline{2\text{nd}} [\text{CATALOG}] [N]$ 。

可从键盘上输入函数名。

- 若要输入函数名 **u**，则按  $\overline{2\text{nd}} [u]$  ( $\overline{7}$  上面)。
- 若要输入函数名 **v**，则按  $\overline{2\text{nd}} [v]$  ( $\overline{8}$  上面)。
- 若要输入函数名 **w**，则按  $\overline{2\text{nd}} [w]$  ( $\overline{9}$  上面)。

通常，序列是递归或非递归的。序列仅计算连续整数。  $n$  总是一系列连续整数，起始于 0 或任意正整数。

## 非递归序列

在非递归序列中，第  $n$  项是自变量  $n$  的函数。每一项都独立于其他项。

例如，在以下的非递归序列中，可直接计算 **u(5)**，而不需计算 **u(1)** 或任何前面的项。

```
Plot1 Plot2 Plot3
nMin=1
u(n)=2*n
u(nMin)
v(n)=
v(nMin)=
w(n)=
w(nMin)=
```

以上序列方程返回序列 **2, 4, 6, 8, 10, ...**，对应于  $n = 1, 2, 3, 4, 5, \dots$ 。

**注意：**计算非递归序列时可使初始值 **u(nMin)** 保留空白。

## 递归序列

在递归序列中，第  $n$  项的定义与前一项或前一项的前一项相关，即  $u(n-1)$  或  $u(n-2)$ 。递归序列也可能被定义为与  $n$  相关，如  $u(n)=u(n-1)+n$ 。

例如，在以下的序列中，如果不首先计算  $u(1)$ 、 $u(2)$ 、 $u(3)$  和  $u(4)$ ，则无法计算出  $u(5)$ 。

```

Plot1 Plot2 Plot3
nMin=1
u(n)▣2*u(n-1)
u(nMin)▣1
    
```

若初始值  $u(nMin) = 1$ ，以上序列返回 **1, 2, 4, 8, 16, ...**。

**提示：**在 TI-83 Plus 上，必须键入项的每个字符。例如，若要输入  $u(n-1)$ ，则按 **[2nd] [u] [ ] [x.t.o.n] [ ] [ ] [ ]**。

递归序列要求有初始值，因为它们引用未定义的项。

- 如果序列中每一项都被定义为与前一项相关，即  $u(n-1)$ ，则必须指定第 1 项的初始值。

```

Plot1 Plot2 Plot3
nMin=1
u(n)▣.8u(n-1)+5
0
u(nMin)▣100
    
```

- 如果序列中每一项都被定义为与前一项的前一项相关，即  $u(n-2)$ ，则必须指定前 2 项的初始值。初始值在一对大括号 ( { } ) 中输入，用逗号分隔每个值。

```

Plot1 Plot2 Plot3
nMin=1
u(n)▣u(n-1)+u(n-2)
(1,0)
u(nMin)▣(1,0)
    
```

序列  $u(n)$  的第 1 项的值为 0，第 2 项的值为 1。



---

## 设置窗口变量

要显示窗口变量值，请按 **WINDOW**。这些变量定义了观察窗口。下表中的值为以 **Radian** 和 **Degree** 角度方式绘制 **Seq** 图象时的默认值。

---

$nMin=1$	用于计算的最小 $n$ 值
$nMax=10$	用于计算的最大 $n$ 值
$PlotStart=1$	绘出的第一项的号码
$PlotStep=1$	$n$ 值的增量（仅对图象）
$Xmin=-10$	可显示的最小 $X$ 值
$Xmax=10$	可显示的最大 $X$ 值
$Xscl=1$	$X$ 分段标记之间的间隔
$Ymin=-10$	可显示的最小 $Y$ 值
$Ymax=10$	可显示的最大 $Y$ 值
$Yscl=1$	$Y$ 分段标记之间的间隔

---

$nMin$  必须为  $\geq 0$  的整数。  $nMax$ 、**PlotStart** 和 **PlotStep** 必须为  $\geq 1$  的整数。

$nMin$  是用于计算的最小  $n$  值。  $nMin$  也在序列  $Y=$  编辑器中显示。  $nMax$  是用于计算的最大  $n$  值。在  $u(nMin)$ ,  $u(nMin+1)$ ,  $u(nMin+2)$ ,  $\dots$ ,  $u(nMax)$  处计算序列值。

**PlotStart** 是绘出的第一项。 **PlotStart=1** 则从序列的第 1 项开始绘图。若要从序列的第 5 项开始绘图，则设置 **PlotStart=5**。前 4 项被计算但不显示在图象中。

**PlotStep** 是仅用于图象的  $n$  值增量。 **PlotStep** 不影响序列的计算，而是仅指出在图象中绘出哪些点。如果指定 **PlotStep=2**，则序列计算每个连续整数处的值，但在图象中每隔一个整数绘一个点。

# 选定坐标轴组合

## 设置图象格式

若要显示当前图象格式的设置，请按 **[2nd] [FORMAT]**。第 3 章详细描述了格式设置。其他图象方式也可用这些格式设置。屏幕顶行的坐标轴设置仅适用于 **Seq** 方式。

<b>Time</b>	<b>Web</b>	<b>uv</b>	<b>vw</b>	<b>uw</b>	序列绘图类型（坐标轴）
<b>RectGC</b>	<b>PolarGC</b>				直角坐标或极坐标输出
<b>CoordOn</b>	<b>CoordOff</b>				光标坐标显示打开/关闭
<b>GridOff</b>	<b>GridOn</b>				网格显示关闭或打开
<b>AxesOn</b>	<b>AxesOff</b>				坐标轴显示打开或关闭
<b>LabelOff</b>	<b>LabelOn</b>				坐标轴标记显示关闭或打开
<b>ExprOn</b>	<b>ExprOff</b>				表达式显示打开或关闭

## 设置坐标轴格式

对于序列图象，可选择 5 种坐标轴格式。下表显示每个坐标轴设置中可绘在 x 轴和 y 轴上的值。

坐标轴设置	x 轴	y 轴
<b>Time</b>	$n$	$u(n), v(n), w(n)$
<b>Web</b>	$u(n-1), v(n-1), w(n-1)$	$u(n), v(n), w(n)$
<b>uv</b>	$u(n)$	$v(n)$
<b>vw</b>	$v(n)$	$w(n)$
<b>uw</b>	$u(n)$	$w(n)$

关于 **Web** 图的详细信息，请参阅页 6-11 和页 6-12。关于相位图（**uv**、**vw** 和 **uw** 坐标轴设置）的详细信息，请参阅页 6-13。

## 显示序列图象

若要绘出选定的序列函数，请按 **[GRAPH]**。绘图过程中，TI-83 Plus 不断更新 **X**、**Y** 和 **n**。

便捷绘图可应用于序列图象（第 3 章）。

## 自由移动光标

Seq 图象和 Func 图象中的自由移动光标的操作相同。在 RectGC 格式中，移动光标可更新 X 和 Y 的值；如果选择 CoordOn 格式，则显示 X 和 Y。在 PolarGC 格式中，X、Y、R 和  $\theta$  被更新；如果选择 CoordOn 格式，则显示 R 和  $\theta$ 。

## TRACE

坐标轴格式设置会影响 TRACE。

当选定 Time、uv、vw 或 uw 坐标轴格式时，TRACE 沿着序列图象移动跟踪光标，一次移动一个 PlotStep 增量。若要一次移动 5 个绘图点，请按  $\boxed{2nd} \boxed{\downarrow}$  或  $\boxed{2nd} \boxed{\leftarrow}$ 。

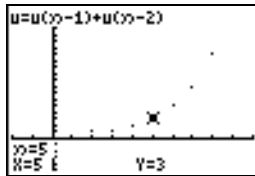
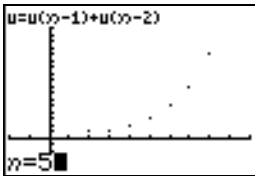
- 开始跟踪时，跟踪光标位于第一个选定序列中由 PlotStart 指定的项，即使它位于观察窗口之外。
- 快速缩放应用于所有方向。若要在移动跟踪光标后使光标位于观察窗口中心，请按  $\boxed{ENTER}$ 。跟踪光标返回到 nMin。

在 Web 格式中，光标的轨迹有助于标识序列中具有吸引和排斥性态的点。开始跟踪时，跟踪光标位于 x 轴上第 1 个选定函数的初始值处。

**提示：**跟踪过程中若要将光标移动到指定的  $n$  处，请输入  $n$  的值，并按  $\boxed{ENTER}$ 。例如，若要将光标快速移动到序列起始处，将 nMin 粘贴到  $n=$  提示处并按  $\boxed{ENTER}$ 。

## 将跟踪光标移动到任意有效的 $n$ 值

对当前函数若要将跟踪光标移动到任意有效的  $n$  值，可输入此  $n$  值。输入第一个数字时，在屏幕左下角显示一个  $n=$  提示和输入的数字。可在  $n=$  提示处输入一个表达式。输入值必须是对当前观察窗口有效的。输入完成后，按  $\boxed{ENTER}$  来移动光标。



## ZOOM

Seq 图象和 Func 图象中的 ZOOM 操作相同。仅影响 X ( Xmin 、 Xmax 和 Xscl ) 和 Y ( Ymin 、 Ymax 和 Yscl ) 窗口变量。

PlotStart 、 PlotStep 、 nMin 和 nMax 仅在选定 ZStandard 时受影响。 VARS Zoom 子菜单 ZU 项 1 到 7 是用于 Seq 图象的 ZOOM MEMORY 变量。

## CALC

仅用于 Seq 图象的 CALC 操作是 value 。

- 当选择 Time 坐标轴格式时， value 显示对应于指定  $n$  值的 Y (  $u(n)$  值)。
- 当选择 Web 坐标轴格式时， value 画出 web 图并显示对应于指定  $n$  值的 Y (  $u(n)$  值)。
- 当选择 uv 、 vw 或 uw 坐标轴格式时， value 按照坐标轴格式设置显示 X 和 Y 。例如，对于 uv 坐标轴格式， X 表示  $u(n)$ ， Y 表示  $v(n)$ 。

## 计算 u 、 v 和 w

若要输入序列名 u 、 v 或 w ，请按  $\boxed{2nd}$  [u]、 [v] 或 [w]。可用 3 种方法计算这些序列。

- 计算序列的第  $n$  个值。
- 计算序列中的一系列值。
- 使用  $u(nstart, nstop[, nstep])$  生成一个序列。  $nstep$  为可选项；默认值为 1 。

```
"n²"→u:u(3)
u({1,3,5,7,9}) 9
{1 9 25 49 81}
u(1,9,2)
{1 9 25 49 81}
```

## 作 Web 图

若要选择 **Web** 坐标轴格式，请按  $\boxed{2nd}$  [FORMAT]  $\boxed{\downarrow}$  [ENTER]。Web 图绘出  $u(n)$  与  $u(n-1)$  的关系曲线，可以用它来研究递归序列的极限情况（收敛、发散或振荡），及改变初始值后序列如何变化。

## Web 图的有效函数

当选择 **Web** 坐标轴格式时，如果序列无法作出适当的图象或产生错误，则：

- 它只有一个递归级（ $u(n-1)$ ，不是  $u(n-2)$ ）。
- 它不能直接引用  $n$ 。
- 它不能引用任何除它之外定义的序列。

## 显示图象屏幕

在 **Web** 格式中，按  $\boxed{GRAPH}$  可显示图象屏幕。TI-83 Plus 可：

- 在 **AxesOn** 格式下画出一条  $y=x$  参考线。
- 将  $u(n-1)$  作为自变量画出选定序列的图象。

**注意：**当序列与  $y=x$  参考线相交时，有潜在的收敛点存在。然而，根据序列的初始值，序列可能收敛于这一点，也可能不收敛于这一点。

## 画出 Web 图

按  $\boxed{TRACE}$  可激活跟踪光标。屏幕显示序列图象和当前的  $n$ 、**X** 和 **Y** 值（**X** 表示  $u(n-1)$ ，**Y** 表示  $u(n)$ ）。按  $\boxed{\downarrow}$  逐步反复从  $nMin$  开始画出 web 图。在 **Web** 格式中，跟踪光标沿下面的路线运动。

1. 起始于 **x** 轴上初始值  $u(nMin)$  处（此处 **PlotStart=1**）。
2. 垂直（向上或向下）移动到序列图象。
3. 水平移动到  $y=x$  参考线。
4. 连续按  $\boxed{\downarrow}$  可反复进行以上垂直和水平移动过程。

## 示例：收敛性

1. 在 **Seq** 方式下按  $\boxed{Y=}$  显示序列  $Y=$  编辑器。确定将图象样式设置为点，然后按照下图定义  $nMin$ 、 $u(n)$  和  $u(nMin)$ 。

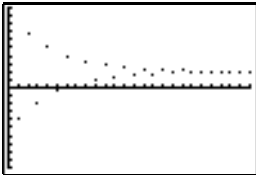
```

Plot1 Plot2 Plot3
nMin=1
:u(n)=-.8u(n-1)+
3.6
u(nMin)=-4)
:v(n)=
v(nMin)=
:w(n)=
    
```

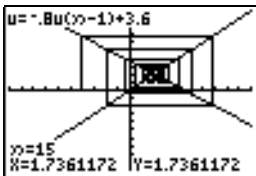
2. 按  $\boxed{2nd}$  [FORMAT]  $\boxed{ENTER}$  设置 **Time** 坐标轴格式。
3. 按  $\boxed{WINDOW}$  并设置变量如下。

<b>nMin=1</b>	<b>Xmin=0</b>	<b>Ymin=-10</b>
<b>nMax=25</b>	<b>Xmax=25</b>	<b>Ymax=10</b>
<b>PlotStart=1</b>	<b>Xscl=1</b>	<b>Yscl=1</b>
<b>PlotStep=1</b>		

4. 按  $\boxed{GRAPH}$  作出序列图象。



5. 按  $\boxed{2nd}$  [FORMAT] 并选择 **Web** 坐标轴设置。
6. 按  $\boxed{WINDOW}$  并更改变量如下。  
**Xmin=-10**                      **Xmax=10**
7. 按  $\boxed{GRAPH}$  作出序列图象。
8. 按  $\boxed{TRACE}$ ，然后按  $\boxed{\blacktriangleright}$  画出 web 图。显示的光标坐标  $n$ 、 $X(u(n-1))$  和  $Y(u(n))$  相应改变。按  $\boxed{\blacktriangleright}$  时，显示一个新的  $n$  值，跟踪光标位于序列上。再次按  $\boxed{\blacktriangleright}$  时， $n$  值保持不变，光标移到  $y=x$  参考线上。跟踪 web 时，将不断重复这个过程。



## 用 uv、vw 和 uw 作图

相位图坐标轴设置 **uv**、**vw** 和 **uw** 表明了两序列之间的关系。若要选定一个相位图坐标轴设置，请按 **[2nd]** **[FORMAT]**，并按 **[▶]** 直到光标位于 **uv**、**vw** 或 **uw** 上，然后按 **[ENTER]**。

坐标轴设置	x 轴	y 轴
<b>uv</b>	<b>u(n)</b>	<b>v(n)</b>
<b>vw</b>	<b>v(n)</b>	<b>w(n)</b>
<b>uw</b>	<b>u(n)</b>	<b>w(n)</b>

## 示例：食肉动物 — 猎物模型

现用食肉动物 — 猎物模型来确定某区域内某种食肉动物及其猎物要保持两物种平衡时的数量。

本例使用此模型来确定狼与野兔的平衡数目，初始数目为野兔 200 只 (**u(nMin)**)，狼 50 只 (**v(nMin)**)。

其他变量如下（括号中为给定值）：

- R = 野兔的数目
- M = 假定狼不存在时野兔的增长率 (.05)
- K = 狼与野兔共存时野兔的死亡率 (.001)
- W = 狼的数目
- G = 狼与野兔共存时狼的增长率 (.0002)
- D = 野兔不存在时狼的死亡率 (.03)
- $n$  = 时间（单位为月）
- $R_n = R_{n-1}(1+M-KW_{n-1})$
- $W_n = W_{n-1}(1+GR_{n-1}-D)$

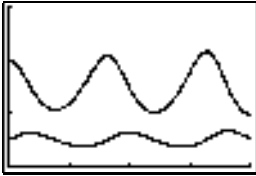
1. 在 **Seq** 方式下按 **[Y=]** 显示序列 **Y=** 编辑器。按照下图定义  $R_n$  和  $W_n$  的序列和初始值。将序列  $R_n$  作为 **u(n)** 输入，将序列  $W_n$  作为 **v(n)** 输入。

```

Plot1 Plot2 Plot3
nMin=1
u(n)≡u(n-1)*(1+
.05-.001*v(n-1))
u(nMin)≡(200)
v(n)≡v(n-1)*(1+
.0002*u(n-1)-.03
)
v(nMin)≡(50)
w(n)=
w(nMin)=
    
```

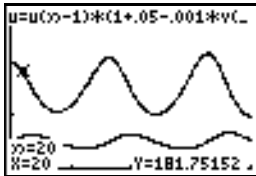
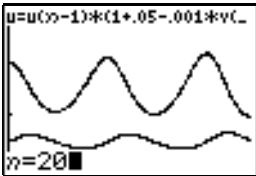
示例：食肉动物 — 猎物模型 (续)

2. 按 **[2nd]** **[FORMAT]** **[ENTER]** 选择 **Time** 坐标轴格式。
3. 按 **[WINDOW]** 并设置变量如下。  
 $nMin=0$        $Xmin=0$        $Ymin=0$   
 $nMax=400$      $Xmax=400$      $Ymax=300$   
 $PlotStart=1$     $Xscl=100$      $Yscl=100$   
 $PlotStep=1$
4. 按 **[GRAPH]** 作出序列图象。

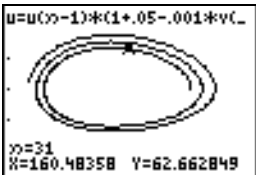


5. 按 **[TRACE]** **[▶]** 按时间 ( $n$ ) 变化单独跟踪野兔的数目 ( $u(n)$ ) 和狼的数目 ( $v(n)$ )。

提示：TRACE 时按一个数字，然后按 **[ENTER]** 可跳到指定的  $n$  值 (月) 处。



6. 按 **[2nd]** **[FORMAT]** **[▶]** **[ENTER]** 选择 **uv** 坐标轴格式。
7. 按 **[WINDOW]** 并改变某些变量的值如下。  
 $Xmin=84$        $Ymin=25$   
 $Xmax=237$      $Ymax=75$   
 $Xscl=50$        $Yscl=10$
8. 按 **[TRACE]**。跟踪 400 代中野兔的数目 ( $X$ ) 和狼的数目 ( $Y$ )。



注意：按 **[TRACE]** 时， $u$  的方程显示在屏幕左上角。按 **[▶]** 或 **[◀]** 可查看  $v$  的方程。



## 序列和窗口变量

如果熟悉 TI-82 可参考下表。下表显示 TI-83 Plus 序列和序列窗口变量，以及相关的 TI-82 序列和序列窗口变量。

TI-83 Plus	TI-82
在 Y= 编辑器中:	
<b><math>u(n)</math></b>	<b>Un</b>
<b><math>u(nMin)</math></b>	<b>UnStart</b> (窗口变量)
<b><math>v(n)</math></b>	<b>Vn</b>
<b><math>v(nMin)</math></b>	<b>VnStart</b> (窗口变量)
<b><math>w(n)</math></b>	不可用
<b><math>w(nMin)</math></b>	不可用
在窗口编辑器中:	
<b>nMin</b>	<b>nStart</b>
<b>nMax</b>	<b>nMax</b>
<b>PlotStart</b>	<b>nMin</b>
<b>PlotStep</b>	不可用

## 序列击键变化

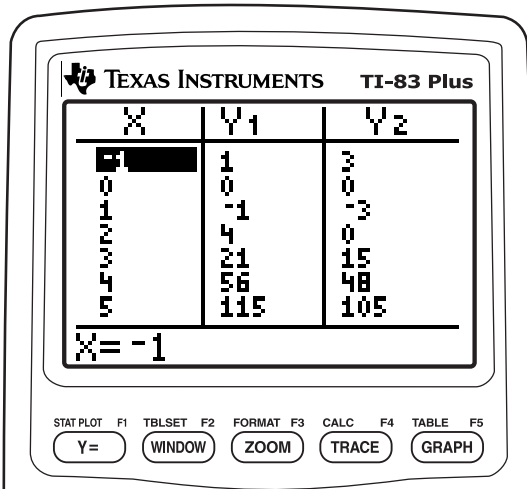
如果熟悉 TI-82 可参考下表。下表对 TI-83 Plus 与 TI-82 之间的序列名语法和变量语法进行了比较。

TI-83 Plus / TI-82	在 TI-83 Plus 上, 按:	在 TI-82 上, 按:
$n / n$	$\boxed{X,T,\theta,n}$	$\boxed{2nd} [n]$
$u(n) / Un$	$\boxed{2nd} [u]$ $\boxed{\square} \boxed{X,T,\theta,n} \boxed{\square}$	$\boxed{2nd} [Y-VARS] \boxed{4} \boxed{1}$
$v(n) / Vn$	$\boxed{2nd} [v]$ $\boxed{\square} \boxed{X,T,\theta,n} \boxed{\square}$	$\boxed{2nd} [Y-VARS] \boxed{4} \boxed{2}$
$w(n)$	$\boxed{2nd} [w]$ $\boxed{\square} \boxed{X,T,\theta,n} \boxed{\square}$	不可用
$u(n-1) / Un-1$	$\boxed{2nd} [u]$ $\boxed{\square} \boxed{X,T,\theta,n} \boxed{-} \boxed{1} \boxed{\square}$	$\boxed{2nd} [U_{n-1}]$
$v(n-1) / Vn-1$	$\boxed{2nd} [v]$ $\boxed{\square} \boxed{X,T,\theta,n} \boxed{-} \boxed{1} \boxed{\square}$	$\boxed{2nd} [V_{n-1}]$
$w(n-1)$	$\boxed{2nd} [w]$ $\boxed{\square} \boxed{X,T,\theta,n} \boxed{-} \boxed{1} \boxed{\square}$	不可用

# 7 运算表

## 目录

入门：函数的根.....	2
设置运算表.....	3
定义因变量.....	4
显示运算表.....	5



入门是对本章内容的快速介绍。详细信息请阅读本章。

对于 -10 和 10 之间的每个整数来计算函数  $Y = X^3 - 2X$  的值。函数值改变多少次符号？在哪些 X 值处改变？

- 按 **MODE**  $\downarrow$   $\downarrow$   $\downarrow$  **ENTER** 设置 **Func** 图象方式。
- 按 **Y=**。按 **X,T,Θ,n** **MATH** 3 选定 3。然后按  $\square$  2 **X,T,Θ,n** 输入函数  $Y1=X^3-2X$ 。

Calculator screen showing function definition:

```

P1ot1 P1ot2 P1ot3
Y1 X^3-2X
Y2 =
Y3 =
Y4 =
Y5 =
Y6 =
Y7 =
    
```

- 按 **2nd** **[TBLSET]** 显示 TABLE SETUP 屏幕。按  $\square$  10 **ENTER** 设置 **TblStart=-10**。按 1 **ENTER** 设置 **ΔTbl=1**。

按 **ENTER** 选择 **Indpnt: Auto**（自动生成自变量值）。按  $\square$  **ENTER** 选择 **Depend: Auto**（自动生成因变量值）。

Calculator screen showing TABLE SETUP settings:

```

TABLE SETUP
TblStart=-10
ΔTbl=1
Indpnt: Auto Ask
Depend: Auto Ask
    
```

- 按 **2nd** **[TABLE]** 显示运算表屏幕。

X	Y1
-10	-980
-9	-711
-8	-496
-7	-329
-6	-204
-5	-115
-4	-56

X=-10

- 按  $\square$  直到发现 **Y1** 值符号改变。改变多少次符号？在哪些 X 值处改变？

X	Y1
-3	-21
-2	-4
-1	1
0	0
1	-1
2	4
3	21

X=3

## TABLE SETUP 屏幕

按 **[2nd]** [TBLSET] 可显示 TABLE SETUP 屏幕。



### TblStart、ΔTbl

**TblStart**（运算表起始点）定义了自变量的初始值。**TblStart** 仅应用于自变量自动生成的情况（即选择 **Indpnt: Auto** 时）。

**ΔTbl**（运算表步长）定义了自变量的增量。

**注意：**在 **Seq** 方式下，**TblStart** 和 **ΔTbl** 必须是整数。

### Indpnt: Auto, Indpnt: Ask, Depend: Auto, Depend: Ask

选项	运算表特性
<b>Indpnt: Auto</b> <b>Depend: Auto</b>	在自变量列和所有的因变量列中自动显示值。
<b>Indpnt: Ask</b> <b>Depend: Auto</b>	运算表为空；当输入一个自变量值时，则自动计算并显示所有对应的因变量值。
<b>Indpnt: Auto</b> <b>Depend: Ask</b>	自动显示自变量值；若要产生一个因变量值，可将光标移动到相应单元格并按 <b>[ENTER]</b> 。
<b>Indpnt: Ask</b> <b>Depend: Ask</b>	运算表为空；输入自变量值；若要产生一个因变量值，可将光标移动到相应单元格并按 <b>[ENTER]</b> 。

### 在主屏幕或程序中建立运算表

要在主屏幕或程序中将值存储到 **TblStart**、**ΔTbl** 或 **TblInput**，可从 VARS TABLE 子菜单中选定变量名。**TblInput** 是当前运算表自变量值的列表。

在程序编辑器中按 **[2nd]** [TBLSET] 时，可选择 **IndpntAuto**、**IndpntAsk**、**DependAuto** 和 **DependAsk**。

## 在 Y= 编辑器中定义因变量

在 Y= 编辑器中，输入定义因变量的函数。只有在 Y= 编辑器中选定的函数才显示在运算表中。使用当前的图象方式。在 **Par** 方式中，必须定义每个参数方程分量（第 4 章）。

## 在运算表编辑器中编辑因变量

以下是在运算表编辑器中编辑选定 Y= 函数的步骤：

1. 按 **[2nd]** **[TABLE]** 显示运算表，然后按 **[▶]** 或 **[◀]** 移动光标到一个因变量列。
2. 按 **[▲]** 直到光标在列顶部的函数名上。此函数显示在底部行。

X	Y1	
0	0	
1	-1	
2	4	
3	21	
4	56	
5	115	
6	204	
Y1 = X <sup>3</sup> - 2X		

3. 按 **[ENTER]**。光标移动到底行。编辑此函数。

X	Y1	
0	0	
1	-1	
2	4	
3	21	
4	56	
5	115	
6	204	
Y1 = X <sup>3</sup> - 2X		

X	Y1	
0	0	
1	-1	
2	4	
3	21	
4	56	
5	115	
6	204	
Y1 = X <sup>3</sup> - 4X		

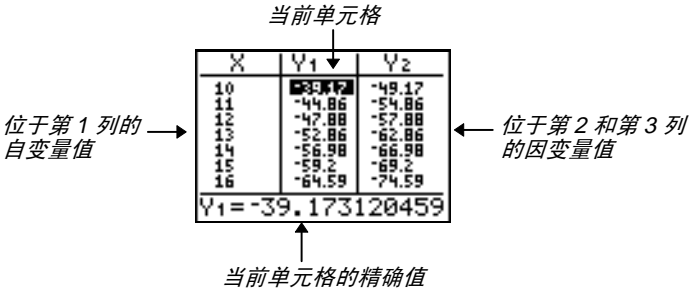
4. 按 **[ENTER]** 或 **[▼]**。计算出新值。运算表和 Y= 函数自动更新。

X	Y1	
0	0	
1	-3	
2	0	
3	15	
4	48	
5	105	
6	192	
Y1 = 0		

**注意：** 也可使用此功能来查看定义因变量的函数，而无须离开运算表。

## 运算表

按  $\boxed{2nd}$  [TABLE] 显示运算表。



注意：必要时运算表可将精确值截断。

## 自变量和因变量

当前的图象方式决定了在运算表中显示哪些自变量和因变量（第 1 章）。例如，在上面的运算表中，由于设置为 **Func** 图象方式，则显示自变量 **X** 和因变量 **Y1** 和 **Y2**。

图象方式	自变量	因变量
<b>Func</b> （函数）	<b>X</b>	<b>Y1 到 Y9，和 Y0</b>
<b>Par</b> （参数）	<b>T</b>	<b>X1T/Y1T 到 X6T/Y6T</b>
<b>Pol</b> （极坐标）	<b><math>\theta</math></b>	<b>r1 到 r6</b>
<b>Seq</b> （序列）	<b>n</b>	<b>u(n)、v(n) 和 w(n)</b>

## 从主屏幕或程序中清除运算表

在主屏幕中，从 CATALOG 内选择 **ClrTable** 指令。若要清除此运算表，请按  $\boxed{ENTER}$ 。

在程序中，从 PRGM I/O 菜单或 CATALOG 中选择 **9:ClrTable**。则运算表被清除。如果选择 **IndpntAsk**，则运算表所有的自变量和因变量值被清除。如果选择 **DependAsk**，则运算表所有的因变量值被清除。

滚动自变量值

如果选择 **Indpnt: Auto**，则在自变量列中可按  $\uparrow$  和  $\downarrow$  以显示更多的值。滚动列时，相应的因变量值也被显示出来。如果选择 **Depend: Ask**，将不显示任何因变量值。

X	Y <sub>1</sub>	Y <sub>2</sub>
0	0	0
1	4	3
2	4	0
3	24	15
4	56	48
5	115	105
6	204	192

X=0

X	Y <sub>1</sub>	Y <sub>2</sub>
0	1	3
1	0	0
2	-1	-3
3	4	0
4	24	15
5	56	48
6	115	105

X= -1

**注意：**可从输入的 **TblStart** 值处回滚。滚动时，**TblStart** 自动更新为运算表顶行中显示的值。上例中，**TblStart=0** 并且  $\Delta\text{Tbl}=1$ ，因此生成并显示 **X=0, ..., 6**；但按  $\uparrow$  回滚后，则运算表显示 **X=-1, ..., 5**。

显示其他因变量

如果定义了多于 2 个的因变量，则最初只显示前 2 个选定的 Y= 函数。按  $\downarrow$  或  $\uparrow$  可显示由其他选定的 Y= 函数定义的因变量。自变量始终显示在左边的列中，除了在 **Par** 图象方式和 **G-T** 拆分屏幕方式设置下进行跟踪时。

X	Y <sub>2</sub>	Y <sub>3</sub>
-4	-4	-28
-3	-6	-18
-2	-6	-10
-1	-4	-4
0	0	0
1	6	2
2	14	2

Y<sub>3</sub> = -28

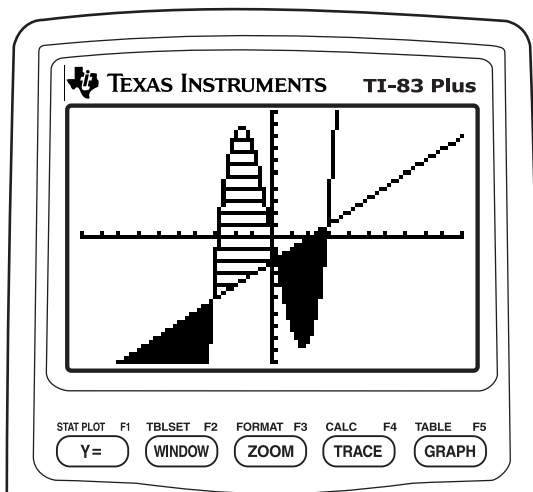
**提示：**若要在运算表中同时显示 2 个不是顺次定义的 Y= 函数的因变量，则要转到 Y= 编辑器并取消选择这 2 个 Y= 函数之间的函数。例如，若要同时在运算表中显示 **Y<sub>4</sub>** 和 **Y<sub>7</sub>**，则要转到 Y= 编辑器并取消选择 **Y<sub>5</sub>** 和 **Y<sub>6</sub>**。



# 8 DRAW 指令

## 目录

入门：画切线.....	2
使用 DRAW 菜单.....	3
清除图形.....	4
画线段.....	5
画水平线和垂直线.....	6
画切线.....	8
画函数和反函数.....	9
在图象上遮蔽区域.....	10
画圆.....	11
在图象上放置文本.....	12
在图象上使用画笔画图.....	13
在图象上画点.....	14
画像素.....	16
存储图象图形 (Pic).....	17
重新调用图象图形 (Pic).....	18
存储图象数据库 (GDB).....	19
重新调用图象数据库 (GDB).....	20



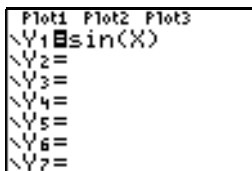
## 入门：画切线

入门是对本章内容的快速介绍。详细信息请阅读本章。

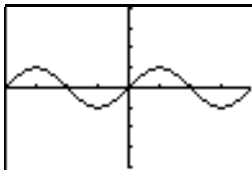
假设您需要找到函数  $Y = \sin(X)$  在  $X = \sqrt{2}/2$  处的切线方程。

开始前，必要时，在方式屏幕中选定 **Radian** 和 **Func** 方式。

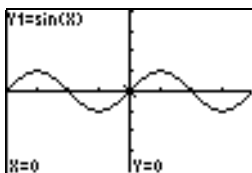
1. 按  $\boxed{Y=}$  显示  $Y=$  编辑器。按  $\boxed{\text{SIN}} \boxed{\text{X,T,}\theta,n} \boxed{\square}$  将  $\sin(X)$  保存在  $Y_1$  中。



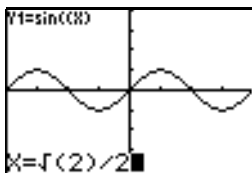
2. 按  $\boxed{\text{ZOOM}} \boxed{7}$  选择 **7:ZTrig**，即在 Zoom Trig 窗口中作图。



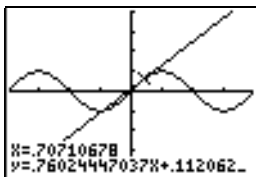
3. 按  $\boxed{2nd} \boxed{\text{DRAW}} \boxed{5}$  选择 **5:Tangent(**。开始执行切线指令。



4. 按  $\boxed{2nd} \boxed{\sqrt{\square}} \boxed{2} \boxed{\square} \boxed{\div} \boxed{2}$ 。



5. 按  $\boxed{\text{ENTER}}$ 。画出切线， $X$  值和切线方程显示在图象上。



## DRAW 菜单

要显示 DRAW 菜单，请按 **[2nd] [DRAW]**。TI-83 Plus 对这些指令的解释取决于是否从主屏幕或程序编辑器或直接从图象上访问此菜单。

---

### DRAW POINTS STO

1: ClrDraw	清除所有已画出的元素。
2: Line(	在两点间画一条线段。
3: Horizontal	画一条水平线。
4: Vertical	画一条垂直线。
5: Tangent(	画一条与函数图象相切的线段。
6: DrawF	画函数的图象。
7: Shade(	遮蔽两个函数间的区域。
8: DrawInv	画函数的反函数图象。
9: Circle(	画圆。
0: Text(	在图象屏幕中画文本。
A: Pen	激活自由形式画图工具。

---

## 在图象上画图之前

DRAW 指令在图象上画图。因而，在使用 DRAW 指令之前，应考虑是否要执行以下一个或多个操作：

- 在方式屏幕中更改方式设置。
- 在格式屏幕中更改格式设置。
- 在 Y= 编辑器中输入或编辑函数。
- 在 Y= 编辑器中选择或取消选择函数。
- 更改窗口变量值。
- 开启或关闭统计图。
- 使用 **ClrDraw**（页 8-4）清除已有的图形。

**注意：**如果在图象上画图后再执行上面列出的任一个操作，当再次显示图象时，则图象被重新画出，但不显示画图。

## 在图象上画图

在 **Func**、**Par**、**PoI** 和 **Seq** 图象上，可使用除 **DrawInv** 之外的任何 DRAW 菜单指令来画图。**DrawInv** 仅对 **Func** 图象有效。所有的 DRAW 指令的坐标是显示的 x 坐标和 y 坐标值。

可使用大多数 DRAW 菜单和 DRAW POINTS 菜单指令在图象上直接画图，使用光标来指示坐标。也可以在主屏幕或程序中执行这些指令。当选定一个 DRAW 菜单指令时，如果不显示图象，则显示主屏幕。

## 显示图象时清除图形

所有使用 DRAW 指令在图象上画出的点、线和阴影都是临时性的。

若要清除当前显示图象上的图形，则从 DRAW 菜单选定 **1:ClrDraw**。当前图象被重画和显示，图象上无任何图形元素。

## 从主屏幕或程序清除图形

若要从主屏幕或程序中清除图象上的图形，从主屏幕上或程序编辑器中的一个空行开始。从 DRAW 菜单选定 **1:ClrDraw**。则此指令被拷贝到光标所在的位置。按 **ENTER**。

当 **ClrDraw** 执行时，它清除所有当前图象上的图形并显示信息 **Done**。当再次显示这个图象时，所有画出的点、线、圆和阴影区域将消失。

```
ClrDraw          Done
```

**注意：**清除图形前，可用 **StorePic** 存储它们（页 8-17）。

## 直接在图象上画线段

要在显示图象时画线段，可按下列步骤进行：

1. 从 DRAW 菜单选择 **2:Line(**。
2. 将光标置于线段的起点，然后按 **[ENTER]**。
3. 将光标移动到线段的终点。移动光标时线段也就显示出来。按 **[ENTER]**。

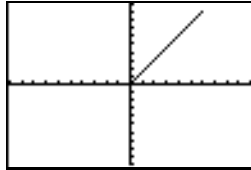
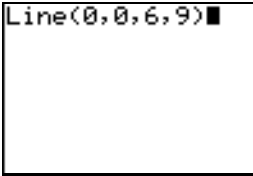


若要继续画线段，则重复步骤 2 和 3。按 **[CLEAR]** 可取消 **Line(**。

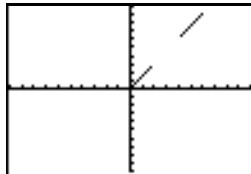
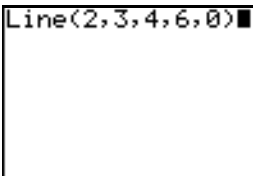
## 从主屏幕或程序画线段

**Line(** 也可在坐标  $(X1,Y1)$  和  $(X2,Y2)$  之间画线段。值也可以表达式形式输入。

**Line(X1,Y1,X2,Y2)**



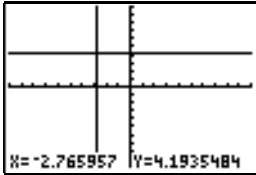
若要消除线段，则输入 **Line(X1,Y1,X2,Y2,0)**。



### 直接在图象上画线

要在图象显示时画水平线或垂直线，可按下列步骤进行：

1. 从 DRAW 菜单选择 **3:Horizontal** 或 **4:Vertical**。移动光标时直线也就显示出来。
2. 将光标放在要画的线所通过的 y 坐标处（画水平线）或 x 坐标处（画垂直线）。
3. 按 **ENTER** 在图象上画直线。



若要继续画直线，则重复步骤 2 和 3。

按 **CLEAR** 可取消 **Horizontal** 或 **Vertical**。

---

## 从主屏幕或程序画线

**Horizontal** (水平线) 在  $Y=y$  处画出一条水平线。  $y$  可以是一个表达式但不能是一个数组。

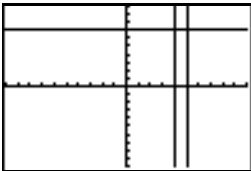
**Horizontal  $y$**

**Vertical** (垂直线) 在  $X=x$  处画出一条垂直线。  $x$  可以是一个表达式但不能是一个数组。

**Vertical  $x$**

若要使 TI-83 Plus 画出一条以上的水平线或垂直线，可用冒号(:)分隔每个指令。

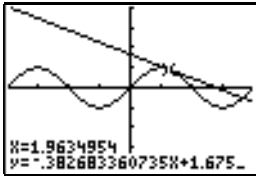
```
Horizontal 7:Ver  
tical 4:Vertical  
5■
```



## 直接在图象上画切线

要在图象显示时画切线，可按下列步骤进行：

1. 从 DRAW 菜单选择 **5:Tangent(**。
2. 按  $\downarrow$  和  $\uparrow$  将光标移动到要画切线的函数上。如果已选择 **ExprOn**，则当前图象的 **Y=** 函数显示在屏幕左上角。
3. 按  $\rightarrow$  和  $\leftarrow$  或输入一个数字在函数上选择要画切线的点。
4. 按 **ENTER**。在 **Func** 方式下，在函数上选择画切线的点的 **X** 值和切线的方程显示在屏幕底部。在其他方式下，显示 **dy/dx** 值。

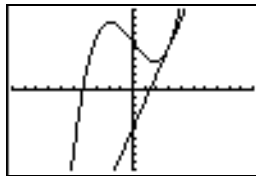
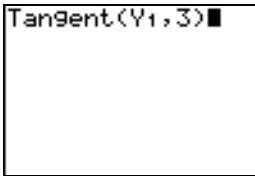


**提示：**如果要查看较少小数位数的 **X** 和 **Y** 的方程，可在方式屏幕中更改定点十进制小数位设置值。

## 从主屏幕或程序画切线

**Tangent(** (切线) 在点 **X=value** 处画出一条与 **X** 的 *expression* (例如 **Y1** 或 **X<sup>2</sup>**) 相切的线。 **X** 可以是一个表达式。 *expression* 解释为 **Func** 方式下的表达式。

**Tangent(expression,value)**

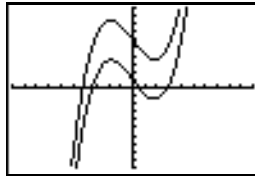
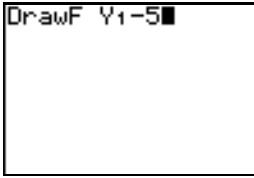




## 画函数

**DrawF**（画函数）在当前图象上将 *expression* 作为一个 **X** 的函数画出。当从 **DRAW** 菜单选定 **6:DrawF** 时，TI-83 Plus 返回到主屏幕或程序编辑器。**DrawF** 不是交互式的。

**DrawF** *expression*

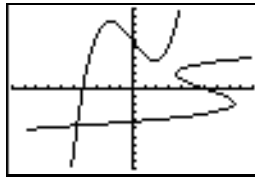


**注意：**不能在 *expression* 中使用数组来画一组曲线。

## 画反函数

**DrawInv**（画反函数）画出 *expression* 的反函数，即在 **y** 轴上画 **X** 值，在 **x** 轴上画 **Y** 值。从 **DRAW** 菜单选择 **8:DrawInv** 时，TI-83 Plus 返回到主屏幕或程序编辑器。**DrawInv** 不是交互式的。**DrawInv** 仅用于 **Func** 方式。

**DrawInv** *expression*



**注意：**不能在 *expression* 中使用数组来画一族曲线。

## 遮蔽图象

若要在图象上遮蔽一个区域，请从 DRAW 菜单选择 **7:Shade()**。则此指令被粘贴到主屏幕或程序编辑器。

**Shade()** 在当前图象上根据 **X** 画出 *lowerfunc* 和 *upperfunc*，并且将 *lowerfunc* 之上、*upperfunc* 之下的区域遮蔽。仅有满足  $lowerfunc < upperfunc$  的区域被遮蔽。

如果包含 *Xleft* 和 *Xright*，则它们指定遮蔽区域的左右边界。*Xleft* 和 *Xright* 必须是 **Xmin** 和 **Xmax** 之间的数，**Xmin** 和 **Xmax** 为默认值。

*pattern* 指定一种遮蔽图案（共 4 种）。

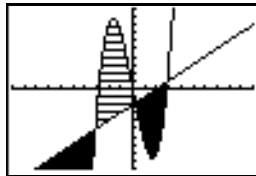
<i>pattern</i> =1	垂直（默认）
<i>pattern</i> =2	水平
<i>pattern</i> =3	负向倾斜 45°
<i>pattern</i> =4	正向倾斜 45°

*patres* 指定一种遮蔽分辨率（共 8 种）。

<i>patres</i> =1	遮蔽每个象素（默认）
<i>patres</i> =2	遮蔽每 2 个象素中的第 1 个象素
<i>patres</i> =3	遮蔽每 3 个象素中的第 1 个象素
<i>patres</i> =4	遮蔽每 4 个象素中的第 1 个象素
<i>patres</i> =5	遮蔽每 5 个象素中的第 1 个象素
<i>patres</i> =6	遮蔽每 6 个象素中的第 1 个象素
<i>patres</i> =7	遮蔽每 7 个象素中的第 1 个象素
<i>patres</i> =8	遮蔽每 8 个象素中的第 1 个象素

**Shade(lowerfunc,upperfunc[,Xleft,Xright,pattern,patres])**

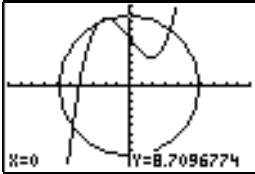
```
Shade(X^3-8X,X-2)
:Shade(X-2,X^3-8X
,-3,2,2,3)
```



## 直接在图象上画圆

要直接在图象显示时使用光标画圆，可按下列步骤进行：

1. 从 DRAW 菜单选择 **9:Circle(**。
2. 将光标放置在要画的圆的圆心，按 **[ENTER]**。
3. 将光标移动到圆周上一点，按 **[ENTER]** 在图象上画出圆。

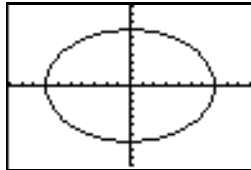
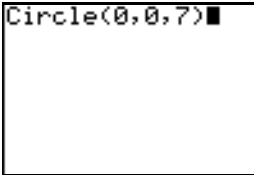


**注意：**因为是直接在显示屏幕上画图，所以画出的圆是圆形的，而与窗口变量值无关。当从主屏幕或程序使用 **Circle(** 指令画圆时，当前的窗口变量会影响圆的形状。

若要继续画圆，可重复步骤 2 和 3。按 **[CLEAR]** 可取消 **Circle(**。

**Circle(** 使用圆心  $(X,Y)$  和半径 *radius* 画圆。这些值可以是表达式。

**Circle(X,Y,radius)**



**提示：**当从主屏幕或程序使用 **Circle(** 指令画圆时，当前的窗口变量值会影响圆的形状。在画圆之前使用 **ZSquare**（第 3 章）调整窗口变量可以使画出的圆保持圆形。

## 直接在图象上放置文本

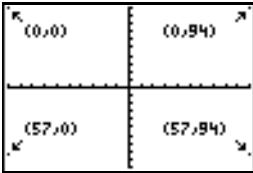
要在图象显示时放置文本，可按下列步骤进行：

1. 从 DRAW 菜单选择 **0:Text(**。
2. 将光标放在文本开始处。
3. 输入字符。按 **[ALPHA]** 或 **[2nd] [A-LOCK]** 输入字母和  $\theta$ 。可输入 TI-83 Plus 函数、变量和指令。字体是比例字体，因此可在图象上放置的字符数是不确定的。输入时，字符被放置在图象上。

按 **[CLEAR]** 可取消 **Text(**。

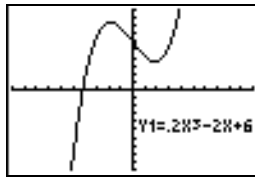
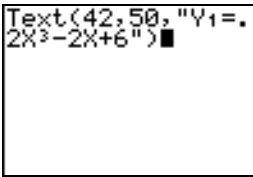
## 从主屏幕或程序放置文本到图象上

**Text(** 在当前图象上放置包含 *value* 的字符，*value* 可包含 TI-83 Plus 函数和指令。第 1 个字符左上角的象素为 (*row*,*column*)，*row* 是 0 和 57 之间的整数，*column* 是 0 和 94 之间的整数。*row* 和 *column* 可以是表达式。



**Text(row,column,value,value ...)**

*value* 可以是包含在双引号 (") 之间的文本，它也可以是表达式。TI-83 Plus 将计算表达式并显示结果（最多 10 个字符）。



## 拆分屏幕

在 **Horiz** 拆分屏幕时，*row* 的最大值是 25。在 **G-T** 拆分屏幕时，*row* 的最大值是 45，*column* 的最大值是 46。

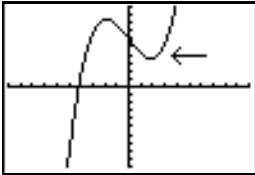
### 在图象上使用画笔画图

**Pen** 仅用于在图象上直接画图。不能从主屏幕或程序执行 **Pen**。

要在显示图象时画图，可按下列步骤进行：

1. 从 DRAW 菜单选择 **A:Pen**。
2. 将光标放在开始画图的点。按 **[ENTER]** 开启画笔。
3. 移动光标。画出的图出现在图象上，随着光标移动，一次遮蔽 1 个像素。
4. 按 **[ENTER]** 关闭画笔。

例如，**Pen** 可以创建指向选定函数的极小值的箭头。



若要继续在图象上画图，请将光标移动到一个新的位置，在此处再开始画图，然后重复步骤 2、3 和 4。按 **[CLEAR]** 可取消 **Pen**。

## DRAW POINTS 菜单

若要显示 DRAW POINTS 菜单，请按 **[2nd] [DRAW] [▢]**。TI-83 Plus 对这些指令的解释取决于是否从主屏幕或程序编辑器或直接从图象上访问此菜单。

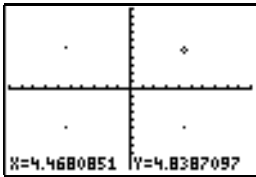
DRAW POINTS STO

1: Pt-On(	开启一个点。
2: Pt-Off(	关闭一个点。
3: Pt-Change(	更改点的开启或关闭状态。
4: Px1-On(	开启一个像素。
5: Px1-Off(	关闭一个像素。
6: Px1-Change(	更改像素的开启或关闭状态。
7: px1-Test(	如果像素开启，返回 1；如果关闭，返回 0。

## 使用 Pt-On( 在图象上直接画点

要在图象上画点，可按下列步骤进行：

1. 从 DRAW POINTS 菜单选择 **1:Pt-On(**。
2. 将光标移动到画点的位置。
3. 按 **[ENTER]** 画出这个点。



若要继续画点，可重复步骤 2 和 3。按 **[CLEAR]** 可取消 **Pt-On(**。

## 使用 Pt-Off( 消除点

要在图象上消除(关闭)一个已画出的点,可按下列步骤进行:

1. 从 DRAW POINTS 菜单选择 **2:Pt-Off(**。
2. 将光标移动到要消除的点处。
3. 按 **[ENTER]** 消除这个点。

若要继续消除点,可重复步骤 2 和 3。按 **[CLEAR]** 可取消 **Pt-Off(**。

## 使用 Pt-Change( 更改点

要在图象上更改(开启或关闭)一个点,可按下列步骤进行:

1. 从 DRAW POINTS 菜单选择 **3:Pt-Change(**(更改点)。
2. 将光标移动到要更改的点处。
3. 按 **[ENTER]** 更改这个点的开启/关闭状态。

若要继续更改点,可重复步骤 2 和 3。按 **[CLEAR]** 可取消 **Pt-Change(**。

## 从主屏幕或程序画点

**Pt-On(**(开启点)在  $(X=x,Y=y)$  处开启点。**Pt-Off(**关闭点。**Pt-Change(**更改点的开启或关闭。*mark* 是可选项;它决定点的外观;指定 **1**、**2** 或 **3**, 它们表示:

**1** = • (点; 默认值)

**2** = □ (框)

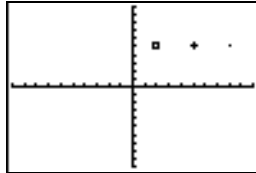
**3** = + (十字)

**Pt-On(*x,y[,mark]*)**

**Pt-Off(*x,y[,mark]*)**

**Pt-Change(*x,y*)**

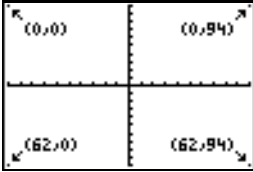
```
Pt-On(2,5,2):Pt-  
On(5,5,3):Pt-On(  
8,5,1)
```



**注意:** 如果使用 **Pt-On(** 来指定 *mark* 开启一个点,则在使用 **Pt-Off(** 关闭这个点时必须指定 *mark*。**Pt-Change(** 无 *mark* 选项。

## TI-83 Plus 像素

TI-83 Plus 显示器上的一个像素是一个正方形的点。 **Pxl-** (像素) 指令允许您使用光标在图象上开启、关闭或更改一个像素 (点)。当从 **DRAW POINTS** 菜单选定一个像素指令时, TI-83 Plus 返回到主屏幕或程序编辑器。像素指令不是交互式的。



### 使用 Pxl-On( 和 Pxl-Off( 开启和关闭像素

**Pxl-On(** (开启像素) 在  $(row,column)$  处开启像素,  $row$  是 0 和 62 之间的整数,  $column$  是 0 和 94 之间的整数。

**Pxl-Off(** 关闭像素。 **Pxl-Change(** 更改像素的开启和关闭状态。

**Pxl-On( $row,column$ )**

**Pxl-Off( $row,column$ )**

**Pxl-Change( $row,column$ )**

### 使用 pxl-Test(

当图象上  $(row,column)$  处的像素开启时, **pxl-Test(** (测试像素) 返回 1, 关闭时返回 0。  $row$  必须是 0 和 62 间的整数。  $column$  必须是 0 和 94 间的整数。

**pxl-Test( $row,column$ )**

### 拆分屏幕

在 **Horiz** 拆分屏幕时, **Pxl-On(**、 **Pxl-Off(**、 **Pxl-Change(** 和 **pxl-Test(** 中的  $row$  的最大值是 30。

在 **G-T** 拆分屏幕时, **Pxl-On(**、 **Pxl-Off(**、 **Pxl-Change(** 和 **pxl-Test(** 中的  $row$  的最大值是 50,  $column$  的最大值是 46。



## DRAW STO 菜单

若要显示 DRAW STO 菜单，请按  $\boxed{2nd}$   $\boxed{[DRAW]}$   $\boxed{[ ]}$ 。当在 DRAW STO 菜单中选定一个指令时，TI-83 Plus 返回到主屏幕或程序编辑器。图形和图象数据库指令不是交互式的。

---

### DRAW POINTS STO

1:StorePic	存储当前图形。
2:RecallPic	重新调用保存的图形。
3:StoreGDB	存储当前图象数据库。
4:RecallGDB	重新调用保存的图象数据库。

---

## 存储图象图形

最多可存储 10 个图象图形，它们都是当前图象显示的图形，存储到图形变量 **Pic1** 至 **Pic9**，或 **Pic0**。稍后，可从主屏幕或程序中将存储的图形叠加到显示的图象之上。

图形包含画图元素、绘出的函数、坐标轴和分段标记。图形不包含坐标轴标记、上下边界指示符、提示符或光标坐标。被这些项隐藏显示的所有部分与图形一起存储。

要存储一个图象图形，可按下列步骤进行：

1. 从 DRAW STO 菜单选择 **1:StorePic**。则 **StorePic** 被粘贴到当前光标所在处。
2. 输入要存储图形的图形变量的数字（**1** 到 **9**，或 **0**）。例如，输入 **3**，则 TI-83 Plus 将图形存储到 **Pic3**。

```
StorePic 3
```

**注意：**也可从 PICTURE 的子菜单 ( $\boxed{VARS}$  **4**) 中选定一个变量。则此变量被粘贴到 **StorePic**。

3. 按  $\boxed{ENTER}$  显示当前图象并存储此图形。

### 重新调用图象图形

要重新调用图象图形，可按下列步骤进行：

1. 从 DRAW STO 菜单选择 **2:RecallPic**。则 **RecallPic** 被粘贴到当前光标所在处。
2. 输入要重新调用图形的图形变量的数字（**1** 到 **9**，或 **0**）。例如，输入 **3**，则 TI-83 Plus 将重新调用存储到 **Pic3** 的图形。

```
RecallPic 3
```

**注意：**也可从 PICTURE 的子菜单 (**VAR**S 4) 中选定一个变量。则此变量被粘贴到 **RecallPic**。

3. 按 **ENTER** 显示当前图象和叠加于它之上的图形。

**注意：**图形作为图的组合，因此无法跟踪一条作为图形组成部分的曲线。

### 删除图象图形

若要从内存中删除图象图形，则使用 MEMORY MANAGEMENT /DELETE 子菜单（第 18 章）。

### 什么是图象数据库?

图象数据库 (GDB) 包含用于定义某个图象的元素的集合。可从这些元素重新创建此图象。最多可存储 10 个 GDB，存储到变量 **GDB1** 到 **GDB9**，或 **GDB0**，重新调用它们可重新创建图象。

一个 GDB 存储图象的 5 个元素。

- 图象方式
- 窗口变量
- 格式设置
- Y= 编辑器的所有函数，每个函数的选择状态
- 每个 Y= 函数的图象样式

GDB 不包含画图项或统计图定义。

### 存储图象数据库

要存储图象数据库，可按下列步骤进行：

1. 从 DRAW STO 菜单选择 **3:StoreGDB**。则 **StoreGDB** 被粘贴到当前光标所在处。
2. 输入要存储图象数据库的 GDB 变量的数字（**1** 到 **9**，或 **0**）。例如，输入 **7**，则 TI-83 Plus 将 GDB 存储到 **GDB7**。

```
StoreGDB 7
```

**注意：**也可从 GDB 的子菜单 (**VAR**S) **3** 中选定一个变量。则此变量被粘贴到 **StoreGDB**。

3. 按 **ENTER** 将当前数据库存储到指定的 GDB 变量。

### 重新调用图象数据库

**警告：**重新调用 GDB 时，它取代所有已有的 Y= 函数。因此在重新调用一个已存储的 GDB 之前，应考虑将当前 Y= 函数存储到另一个数据库。

要重新调用图象数据库，可按下列步骤进行：

1. 从 DRAW STO 菜单选择 **4:RecallGDB**。则 **RecallGDB** 被粘贴到当前光标所在处。
2. 输入要重新调用 GDB 的 GDB 变量的数字（**1** 到 **9**，或 **0**）。例如，输入 **7**，则 TI-83 Plus 将重新调用存储到 **GDB7** 的 GDB。

```
RecallGDB ?
```

**注意：**也可从 GDB 的子菜单 (**VAR**S) **3** 中选定一个变量。则此变量被粘贴到 **RecallGDB**。

3. 按 **ENTER** 将当前的 GDB 替换为重新调用的 GDB。新的图象不绘出。必要时 TI-83 Plus 自动更改图象方式。

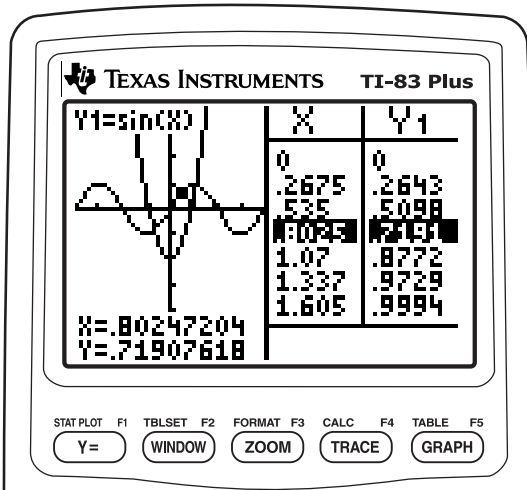
### 删除图象数据库

要从内存中删除一个 GDB，可使用 MEMORY MANAGEMENT/DELETE 子菜单（第 18 章）。

# 9 拆分 屏幕

## 目录

入门：研究单位圆.....	2
使用拆分屏幕.....	3
Horiz（水平）拆分屏幕.....	4
G-T（图象—运算表）拆分屏幕.....	5
Horiz 和 G-T 方式下的 TI-83 Plus 象素.....	6



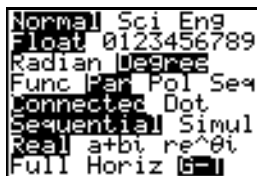
# 入门：研究单位圆

入门是对本章内容的快速介绍。详细信息请阅读本章。

使用 G-T（图象 — 运算表）拆分屏幕方式来研究单位圆以及它和常用角度  $0^\circ$ 、 $30^\circ$ 、 $45^\circ$ 、 $60^\circ$ 、 $90^\circ$  等三角函数值之间的关系。

1. 按 **MODE** 显示方式屏幕。按  $\square \square \square \square$  **ENTER** 选择 **Degree** 方式。按  $\square \square \square$  **ENTER** 选择 **Par**（参数）图象方式。

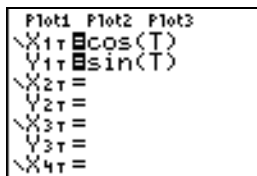
按  $\square \square \square \square \square \square \square$  **ENTER** 选择 **G-T**（图象 — 运算表）拆分屏幕方式。



2. 按 **2nd** **[FORMAT]** 显示格式屏幕。按  $\square \square \square \square$  **ENTER** 选择 **ExprOff**。



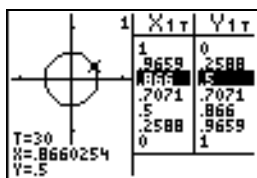
3. 按 **[Y=]** 显示 **Par** 图象方式下的 **Y=** 编辑器。  
按 **[COS]** **[X,T,θ,n]** **[ ]** **ENTER** 将 **cos(T)** 存储为 **X1T**。  
按 **[SIN]** **[X,T,θ,n]** **[ ]** **ENTER** 将 **sin(T)** 存储为 **Y1T**。



4. 按 **[WINDOW]** 显示窗口编辑器。输入窗口变量的值如下：

**Tmin=0**      **Xmin=-2.3**      **Ymin=-2.5**  
**Tmax=360**   **Xmax=2.3**      **Ymax=2.5**  
**Tstep=15**   **Xscl=1**      **Yscl=1**

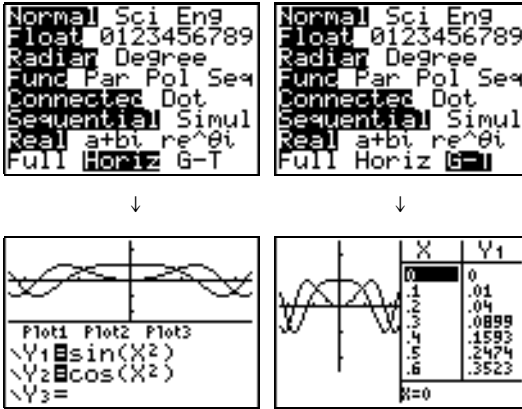
5. 按 **[TRACE]**。屏幕左部显示在 **Degree** 方式下作出的单位圆，跟踪光标被激活。**T=0**（图象上跟踪光标的坐标）时，可从屏幕右部的运算表看出 **X1T** (**cos(T)**) 的值是 **1**，**Y1T** (**sin(T)**) 的值是 **0**。按  $\square$  移动光标，间隔是  $15^\circ$  角。跟踪过程中，运算表中每个角度的近似标准函数值亮显。



## 设置拆分屏幕方式

若要设置拆分屏幕方式，按 **[MODE]**，然后将光标移动到方式屏幕的底行。

- 选择 **Horiz**（水平）则显示图象屏幕和另一个水平拆分的屏幕。
- 选择 **G-T**（图象 — 运算表）则显示图象屏幕和垂直拆分的运算表屏幕。



当按任一作用于任一一半拆分屏幕的键时，则此拆分屏幕被激活。

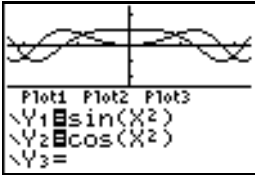
一些屏幕无法显示为拆分屏幕。例如，如果在 **Horiz** 或 **G-T** 方式下按 **[MODE]**，则方式屏幕显示为全屏。如果然后按显示任一一半拆分屏幕的键，例如 **[TRACE]**，则返回到拆分屏幕。

当在 **Horiz** 或 **G-T** 方式下按一个键或组合键，则光标被放置在此键作用的那一半屏幕。例如，如果按 **[TRACE]**，则光标被放置在显示图象的那一半屏幕。如果按 **[2nd]** **[TABLE]**，则光标被放置在显示运算表的那一半屏幕。

TI-83 Plus 将保留拆分屏幕方式直到更改回 **Full** 屏幕方式。

## Horiz 方式

在 Horiz（水平）拆分屏幕方式下，一条水平线将屏幕拆分为上下两部分。



上面部分显示图象。

下面部分显示以下编辑器之一：

- 主屏幕（4行）
- Y= 编辑器（4行）
- 统计数组编辑器（2行）
- 窗口编辑器（3个设置）
- 运算表编辑器（2行）

## Horiz 方式下两部分之间切换

若要使用拆分屏幕的上部：

- 按 **GRAPH** 或 **TRACE**。
- 选择一个 **ZOOM** 或 **CALC** 操作。

若要使用拆分屏幕的下部：

- 按可显示主屏幕的任意键或组合键。
- 按 **Y=**（Y= 编辑器）。
- 按 **STAT** **ENTER**（统计数组编辑器）。
- 按 **WINDOW**（窗口编辑器）。
- 按 **2nd** **TABLE**（运算表编辑器）。

## Horiz 方式下的全屏显示

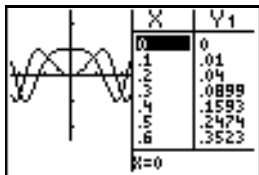
在 Horiz 拆分屏幕方式下，所有其他的屏幕都被显示为全屏。

在 Horiz 方式下若要从全屏返回到 Horiz 拆分屏幕，则按可显示图象、主屏幕、Y= 编辑器、统计数组编辑器、窗口编辑器或运算表编辑器的任意键或组合键。



## G-T 方式

在 G-T (图象 — 运算表) 拆分屏幕方式下，一条垂直线将屏幕拆分为左右两部分。



左半部显示图象。

右半部显示运算表。

## G-T 方式下两部分之间切换

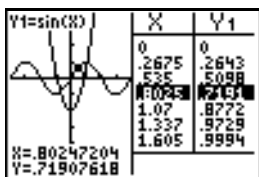
若要使用拆分屏幕的左半部：

- 按 **GRAPH** 或 **TRACE** 。
- 选择一个 ZOOM 或 CALC 操作。

若要使用拆分屏幕的右半部，按 **2nd** [**TABLE**]。

## G-T 方式下使用 **TRACE**

G-T 方式下，在沿着拆分屏幕左半部的图象移动跟踪光标的过程中，右半部的运算表自动滚动以匹配当前光标值。



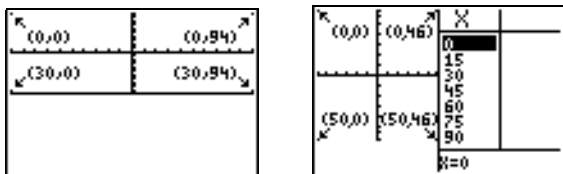
**注意：**在 Par 图象方式下跟踪时，方程的两个分量 ( $XnT$  和  $YnT$ ) 分别显示在表格的两列中。跟踪时，自变量 T 的当前值显示在图象上。

## G-T 方式下的全屏显示

在 G-T 拆分屏幕方式下，除图象和运算表之外的所有屏幕都以全屏显示。

在 G-T 方式下若要从全屏返回到 G-T 拆分屏幕，则按可显示图象或运算表的任意键或组合键。

## Horiz 和 G-T 方式下的 TI-83 Plus 象素



注意：上面括弧中的每对数字分别表示已开启的角点象素的行和列。

### DRAW POINTS 菜单象素指令

对于 `Pxl-On(`、`Pxl-Off(`、`Pxl-Change(` 和 `pxl-Test(` :

- Horiz 方式下, *row* 必须  $\leq 30$  ; *column* 必须  $\leq 94$  。
- G-T 方式下, *row* 必须  $\leq 50$  ; *column* 必须  $\leq 46$  。

`Pxl-On(row,column)`

### DRAW 菜单 Text( 指令

对于 `Text(` 指令:

- Horiz 方式下, *row* 必须  $\leq 25$  ; *column* 必须  $\leq 94$  。
- G-T 方式下, *row* 必须  $\leq 45$  ; *column* 必须  $\leq 46$  。

`Text(row,column,"text")`

### PRGM I/O 菜单 Output( 指令

对于 `Output(` 指令:

- Horiz 方式下, *row* 必须  $\leq 4$  ; *column* 必须  $\leq 16$  。
- G-T 方式下, *row* 必须  $\leq 8$  ; *column* 必须  $\leq 16$  。

`Output(row,column,"text")`

### 从主屏幕或程序设置拆分屏幕方式

若要从程序中设置 Horiz 或 G-T , 可按以下步骤进行:

1. 在程序编辑器中当光标移动到一个空行时按 `MODE` 。
2. 选择 Horiz 或 G-T 。

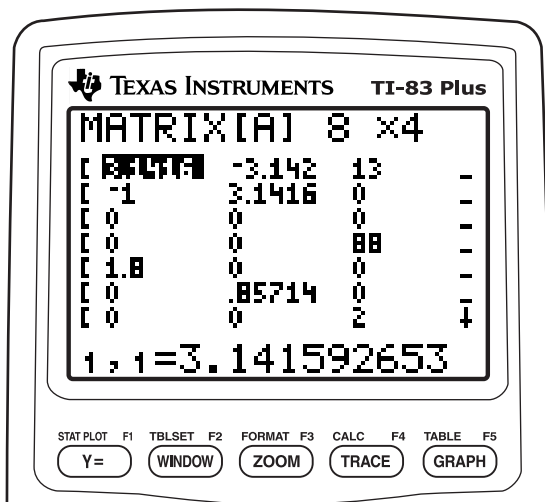
此指令被粘贴到光标所在处。程序执行过程中遇到此指令则设置拆分屏幕方式。在程序执行完后设置仍保持。

注意：也可从 CATALOG (第 15 章) 将 Horiz 或 G-T 粘贴到主屏幕或程序编辑器。

# 10 矩阵

## 目录

入门：线性方程组.....	2
定义矩阵.....	3
观察和编辑矩阵元素.....	4
在表达式中使用矩阵.....	7
显示和拷贝矩阵.....	8
对矩阵进行数学运算.....	9
使用 MATRX MATH 运算.....	12



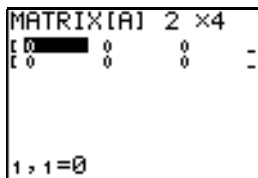
## 入门：线性方程组

入门是对本章内容的快速介绍。详细信息请阅读本章。

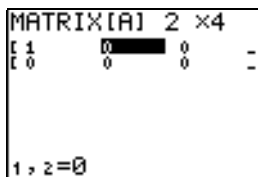
求  $X + 2Y + 3Z = 3$  和  $2X + 3Y + 4Z = 3$  的解。在 TI-83 Plus 中，可通过将方程系数作为矩阵元素输入来解线性方程组，然后使用 `rref`( 获得简化的行梯形矩阵。

1. 按 `2nd` [MATRX]。按 `▸ ▸` 显示 MATRIX EDIT 菜单。按 `1` 选择 **1: [A]**。
2. 按 `2` [ENTER] `4` [ENTER] 定义一个 2x4 矩阵。矩形光标指示当前元素。省略号 (...) 表示屏幕之外还有其他列。
3. 按 `1` [ENTER] 输入第 1 个元素。矩形光标移动到第 1 行的第 2 列。
4. 按 `2` [ENTER] `3` [ENTER] `3` [ENTER] 完成第 1 行的输入，第 1 行表示  $X + 2Y + 3Z = 3$ 。
5. 按 `2` [ENTER] `3` [ENTER] `4` [ENTER] `3` [ENTER] 完成第 2 行的输入，第 2 行表示  $2X + 3Y + 4Z = 3$ 。
6. 按 `2nd` [QUIT] 返回到主屏幕。必要时按 `CLEAR` 清除主屏幕。按 `2nd` [MATRX] `▸` 显示 MATRIX MATH 菜单。按 `▸` 跳到菜单末尾。选择 **B:rref**( 将 `rref`( 拷贝到主屏幕。
7. 按 `2nd` [MATRX] `1` 从 MATRIX NAMES 菜单选择 **1: [A]**。按 `▸` [ENTER]，此矩阵的简化行梯形矩阵显示出来，并保存在 **Ans** 中。

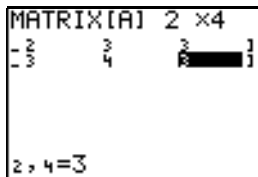
$$\begin{array}{ll} 1X - 1Z = -3 & \text{则} \quad X = -3 + Z \\ 1Y + 2Z = 3 & \text{则} \quad Y = 3 - 2Z \end{array}$$



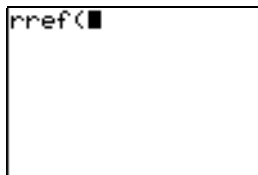
```
MATRIX[A] 2 x4
[ 0 0 0 -
[ 0 0 0 -
1, 1=0
```



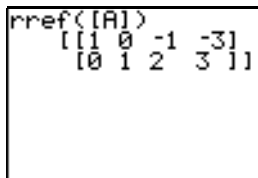
```
MATRIX[A] 2 x4
[ 1 0 0 -
[ 0 0 0 -
1, 2=0
```



```
MATRIX[A] 2 x4
- 2 3 3 -
- 3 4 3 -
2, 4=3
```



```
rref(
```



```
rref([A])
[[1 0 -1 -3]
 [0 1 2 3]]
```

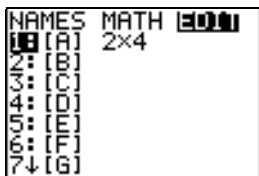
## 什么是矩阵？

矩阵是一个 2 维数组。在矩阵编辑器中可显示、定义或编辑矩阵。TI-83 Plus 有 10 个矩阵变量，即 [A] 到 [J]。可直接在表达式中定义一个矩阵。根据可用的内存大小，矩阵最多可有 99 行或列。在 TI-83 Plus 矩阵中只能存储实数。

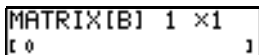
## 选择矩阵

在编辑器中定义或显示矩阵之前，首先必须选择矩阵名。若要选择矩阵名，可按以下步骤进行：

1. 按  $\boxed{2nd}$  [MATRX]  $\boxed{\downarrow}$  显示 MATRX EDIT 菜单。则显示所有以前定义过的矩阵的维数。



2. 选择要定义的矩阵。显示 MATRX EDIT 屏幕。



## 接受或更改矩阵维数

矩阵的维数 (*row*  $\times$  *column*) 显示在顶行。新矩阵的维数是  $1 \times 1$ 。每次编辑矩阵时必须接受或更改矩阵的维数。当选择一个矩阵来定义时，光标突出显示行的维数。

- 若要接受行的维数，请按  $\boxed{ENTER}$ 。
- 若要更改行的维数，请输入行数（最大为 99），然后按  $\boxed{ENTER}$ 。

光标移动到列的维数，必须用同样的方法接受或更改列的维数。按  $\boxed{ENTER}$  后，矩形光标移动到第 1 个矩阵元素。

## 显示矩阵元素

设置矩阵维数之后，可观察此矩阵并输入矩阵元素的值。在新矩阵中，所有值都是 0。

从 **MATRIX EDIT** 菜单选择矩阵并输入或接受矩阵的维数。矩阵编辑器中心部分最多可显示矩阵的 7 行和 3 列，必要时使用简略形式显示元素的值。矩形光标指示的当前元素的精确值显示在底行。

```
MATRIX[A] 8 x4
[ 0.0000  -3.142  13  -
[ -1  3.1416  0  -
[ 0  0  0  -
[ 0  0  88  -
[ 1.8  0  0  -
[ 0  .85714  0  -
[ 0  0  2  ↓
1, 1=3.141592653
```

上面为一个  $8 \times 4$  矩阵。左边或右边列的省略号表示还有其他的列。右边列的  $\uparrow$  或  $\downarrow$  表示还有其他的行。

## 删除矩阵

若要从内存中删除矩阵，请使用 **MEMORY MANAGEMENT/DELETE** 子菜单（第 18 章）。

## 观察矩阵

矩阵编辑器有两种环境，即观察和编辑。在观察环境中，可使用光标键在矩阵元素间快速移动。亮显的元素精确值显示在底行。

从 **MATRIX EDIT** 菜单选择矩阵，然后可输入或接受矩阵的维数。

```
MATRIX[A] 8 ×4
[ 3.14159  -3.142  13  --
[ -1      3.1416  0   --
[ 0       0       0   --
[ 0       0       88  --
[ 1.8     0       0   --
[ 0       .85714  0   --
[ 0       0       2   ↓
1, 1=3.141592653
```

## 观察环境键

键	功能
← 或 →	在当前行移动矩形光标
↓ 或 ↑	在当前列移动矩形光标；在顶行，↑ 将光标移动到列的维数；在列的维数，↓ 将光标移动到行的维数
<b>ENTER</b>	切换到编辑环境；激活底行的编辑光标
<b>CLEAR</b>	切换到编辑环境；清除底行的值
任意输入字符	切换到编辑环境；清除底行的值；拷贝字符到底行
<b>2nd</b> [INS]	无
<b>DEL</b>	无

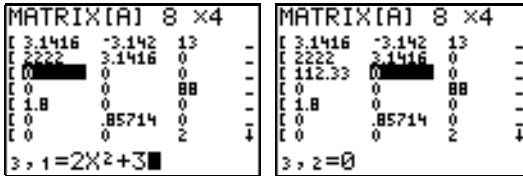
## 编辑矩阵元素

在编辑环境中，编辑光标出现在底行上。若要编辑矩阵元素值，可按以下步骤进行：

1. 从 **MATRIX EDIT** 菜单选择矩阵，然后输入或接受它的维数。
2. 按 **←**、**↑**、**→** 和 **↓** 将光标移动到要更改的矩阵元素。
3. 按 **ENTER**、**CLEAR** 或一个输入键切换到编辑环境。
4. 使用下文描述的编辑环境键更改矩阵元素的值。可输入一个表达式，在离开编辑环境时计算它的值。

**注意：**出错时，可按 **CLEAR** **ENTER** 恢复矩形光标处的值。

5. 按 **ENTER**、**↑** 或 **↓** 移到另一个元素。



## 编辑环境键

键	功能
<b>←</b> 或 <b>→</b>	在值中移动编辑光标
<b>↓</b> 或 <b>↑</b>	将显示在底行的值存储到矩阵元素；切换到观察环境并在此列中移动矩形光标
<b>ENTER</b>	将显示在底行的值存储到矩阵元素；切换到观察环境并将矩形光标移动到下一个行元素。
<b>CLEAR</b>	清除底行的值
任意输入字符	将字符拷贝到底行中的编辑光标处
<b>2nd</b> <b>[INS]</b>	激活插入光标
<b>DEL</b>	删除底行中编辑光标处的字符



## 在表达式中使用矩阵

若要在表达式中使用矩阵，可执行以下步骤之一：

- 从 **MATRX NAMES** 菜单拷贝矩阵名。
- 使用 **[2nd] [RCL]** 将矩阵内容重新调用到表达式中（第 1 章）。
- 直接输入矩阵（见下一部分）。

## 在表达式中输入矩阵

在矩阵编辑器中可输入、编辑和存储矩阵。也可直接在表达式中输入矩阵。

若要在表达式中输入矩阵，可按以下步骤进行：

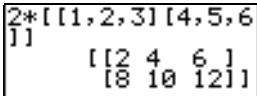
1. 按 **[2nd] [[]]** 指示矩阵的开始。
2. 按 **[2nd] [[]]** 指示行的开始。
3. 对于行的每个元素，输入一个值，它可以是一个表达式。值之间用逗号分隔。
4. 按 **[2nd] [I]** 指示行的结束。
5. 重复步骤 2 到 4 输入所有的行。
6. 按 **[2nd] [I]** 指示矩阵的结束。

**注意：**结尾的 **]]** 在表达式最后或 **→** 前是不需要的。

最后的矩阵使用以下形式显示：

**[[*element*<sub>1,1</sub>,...,*element*<sub>1,n</sub>],..., [*element*<sub>*m*,1</sub>,...,*element*<sub>*m*,n</sub>]]**

所有的表达式都是在执行输入时计算的。

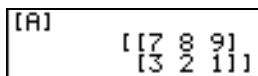


A calculator display showing the input of a matrix expression. The screen displays "2\*[[1,2,3][4,5,6]]" with a cursor at the end. The matrix part is shown as two rows: the first row contains "1 2 3" and the second row contains "4 5 6".

**注意：**输出时不显示用于分隔元素的逗号（它是必须输入的）。

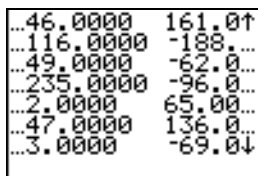
## 显示矩阵

若要在主屏幕中显示矩阵的内容，请从 **MATRX NAMES** 菜单选择此矩阵，然后按 **ENTER**。



[A] [ [7 8 9] [3 2 1] ]

左边或右边列的省略号表示还有其他的列。右边列的  $\uparrow$  或  $\downarrow$  表示还有其他的行。按  $\rightarrow$ 、 $\leftarrow$ 、 $\downarrow$  和  $\uparrow$  可滚动此矩阵。

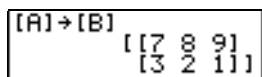


...46.0000 161.0 $\uparrow$   
...116.0000 -188.0...  
...49.0000 -62.0...  
...235.0000 -96.0...  
...2.0000 65.00...  
...47.0000 136.0...  
...3.0000 -69.0 $\downarrow$

## 拷贝一个矩阵到另一个矩阵

若要拷贝一个矩阵，可按以下步骤进行：

1. 按 **2nd** [**MATRX**] 显示 **MATRX NAMES** 菜单。
2. 选择要拷贝的矩阵名。
3. 按 **STO $\rightarrow$** 。
4. 再次按 **2nd** [**MATRX**] 并选择新矩阵名，它是拷贝的目标矩阵。
5. 按 **ENTER** 拷贝已有矩阵到新矩阵名。

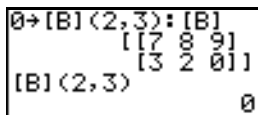


[A]  $\rightarrow$  [B] [ [7 8 9] [3 2 1] ]

## 访问矩阵元素

在主屏幕或程序中，可将一个值存储为矩阵元素，或调用矩阵元素的值。此元素必须在当前定义的矩阵之中。从 **MATRX NAMES** 菜单中选择 *matrix*。

**[matrix](row,column)**



0  $\rightarrow$  [B] (2,3) : [B]  
[ [7 8 9] [3 2 0] ]  
[B] (2,3) 0

## 对矩阵进行数学运算

TI-83 Plus 键盘、MATH 菜单、MATH NUM 菜单和 MATH TEST 菜单上的许多数学运算可用于矩阵。不过，矩阵维数必须匹配。下面每个运算都创建一个新矩阵；而原始矩阵不变。

### + (加)、- (减)、\* (乘)

若要加 (+) 或减 (-) 矩阵，则维数必须相同。结果矩阵的元素是运算矩阵相应元素的和或差。

$$\text{matrixA} + \text{matrixB}$$

$$\text{matrixA} - \text{matrixB}$$

若将两个矩阵相乘 ( $\otimes$ )，则 *matrixA* 的列维数必须匹配 *matrixB* 的行维数。

$$\text{matrixA} \otimes \text{matrixB}$$

[A]	$\begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix}$	[A]+[B]	$\begin{bmatrix} 2 & 7 \\ 7 & 7 \end{bmatrix}$
[B]	$\begin{bmatrix} 0 & 5 \\ 4 & 3 \end{bmatrix}$	[A]*[B]	$\begin{bmatrix} 8 & 16 \\ 16 & 27 \end{bmatrix}$

将 *matrix* 乘 *value* 或将 *value* 乘 *matrix*，则返回一个矩阵，它的元素是相应的 *matrix* 的元素乘以 *value* 后所得的值。

$$\text{matrix} * \text{value}$$

$$\text{value} * \text{matrix}$$

[A]*3	$\begin{bmatrix} 6 & 6 \\ 9 & 12 \end{bmatrix}$
-------	---

### - (求负)

对一个矩阵求负 ( $\ominus$ ) 则返回一个矩阵，其元素的符号与矩阵 *matrix* 相应元素的相反。

$$-\text{matrix}$$

[A]	$\begin{bmatrix} 2 & -2 \\ 3 & 4 \end{bmatrix}$
-[A]	$\begin{bmatrix} -2 & 2 \\ -3 & -4 \end{bmatrix}$

abs(

**abs**(绝对值, MATH NUM 菜单) 返回一个矩阵, 其元素是 *matrix* 中对应元素的绝对值。

**abs(matrix)**

```
[C]
  [[-23 -69]
   [-25 -14]]
abs(C)
  [[23 69]
   [25 14]]
```

round(

**round**( ( MATH NUM 菜单) 返回一个矩阵。它将 *matrix* 中的每个元素舍入为 *#decimals* ( $\leq 9$ )。如果省略 *#decimals*, 则元素被舍入为 10 位小数。

**round(matrix[,#decimals])**

```
MATRIX[A] 2 x2
[[ 1.259  2.333 ]
 [ 3.662  4.121 ] ]
round(A,2)
[[1.26 2.33]
 [3.66 4.12]]
```

<sup>-1</sup> (求逆)

使用 <sup>-1</sup> 运算 ( $x^{-1}$ ) 对一个矩阵求逆 ( $\wedge^{-1}$  无效)。*matrix* 必须是方阵, 其行列式值不能为 0。

*matrix*<sup>-1</sup>

```
MATRIX[A] 2 x2
[[ 1  2 ]
 [ 3  4 ] ]
[A]-1
[[ -2  1 ]
 [ 1.5 -0.5 ] ]
```

幂

若要矩阵求幂, 则 *matrix* 必须是方阵。可使用  $2(x^2)$ 、 $3$  ( MATH 菜单) 或  $\wedge$ power ( $\wedge$ ) (求 *power* 次方, *power* 是 0 到 255 之间的整数)。

*matrix*<sup>2</sup>

*matrix*<sup>3</sup>

*matrix*<sup>power</sup>

```
MATRIX[A] 2 x2
[[ 1  2 ]
 [ 3  4 ] ]
[A]3
[[ 37  54 ]
 [ 81 118 ] ]
[A]5
[[1069 1558]
 [2337 3406 ] ]
```

## 关系运算

若要使用关系运算  $=$  和  $\neq$  来比较两个矩阵（TEST 菜单），则它们的维数必须相等。 $=$  和  $\neq$  根据元素间的相互关系比较  $matrixA$  和  $matrixB$ 。其他的关系运算对矩阵无效。

$matrixA=matrixB$  在每个比较为真时返回 **1**；如果有一个比较为假则返回 **0**。

$matrixA\neq matrixB$  在至少有一个比较为假时返回 **1**；在比较都为真时返回 **0**。

[A]	$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix}$	[A]=[B]	0
[B]	$\begin{bmatrix} 3 & 2 & 1 \\ 1 & 2 & 3 \end{bmatrix}$	[A] $\neq$ [B]	1

## iPart(、fPart(、int(

**iPart(**（整数部分）、**fPart(**（分数部分）和 **int(**（最大整数）是 MATH NUM 菜单中的运算。

**iPart(** 返回其元素是  $matrix$  中对应元素的整数部分的矩阵。

**fPart(** 返回其元素是  $matrix$  中对应元素的分数部分的矩阵。

**int(** 返回其元素是  $matrix$  中对应元素的最大整数的矩阵。

**iPart(matrix)**

**fPart(matrix)**

**int(matrix)**

[D]	$\begin{bmatrix} 1.25 & 3.333 \\ 100.5 & 47.15 \end{bmatrix}$	iPart([D])	$\begin{bmatrix} 1 & 3 \\ 100 & 47 \end{bmatrix}$
		fPart([D])	$\begin{bmatrix} .25 & .333 \\ .5 & .15 \end{bmatrix}$

## MATRX MATH 菜单

若要显示 MATRX MATH 菜单，请按  $\boxed{2nd} \boxed{[MATRX]} \boxed{\blacktriangleright}$ 。

NAMES **MATH** EDIT

1: det(	计算行列式的值。
2: T	转置矩阵。
3: dim(	返回矩阵维数。
4: Fill(	使用一个常数填充所有元素。
5: identity(	返回单位矩阵。
6: randM(	返回随机矩阵。
7: augment(	对两个矩阵进行追加运算。
8: Matrx $\blacktriangleright$ list(	将矩阵存储为数组。
9: List $\blacktriangleright$ matrx(	将数组存储为矩阵。
0: cumSum(	返回矩阵的累积和。
A: ref(	返回矩阵的行梯形矩阵。
B: rref(	返回简化的行梯形矩阵。
C: rowSwap(	交换矩阵中的两行。
D: row+(	两行相加；结果存储在第 2 行。
E: *row(	将某行乘以一个数字。
F: *row+(	将某行乘以一个数字，并添加到第 2 行。

### det(

**det(** (行列式) 返回方阵 *matrix* 的行列式值 (一个实数)。

**det(matrix)**

### T (转置)

**T** (转置) 返回一个矩阵，这个矩阵的每个元素 (行，列) 与 *matrix* 中相应的元素 (列，行) 互换位置。

*matrix*<sup>T</sup>

[A]	$\begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix}$	[A] <sup>T</sup>	$\begin{bmatrix} 1 & 3 \\ 2 & 2 \\ 3 & 1 \end{bmatrix}$
-----	--	------------------	---

### 使用 dim( 访问矩阵维数

**dim(** (维数) 返回一个数组，其中包含 *matrix* 的维数 (*{rows columns}*)。

**dim(matrix)**

**注意：****dim(matrix)** $\blacktriangleright$ **L<sub>n</sub>:L<sub>n</sub>(1)** 返回矩阵的行数。**dim(matrix)** $\blacktriangleright$ **L<sub>n</sub>:L<sub>n</sub>(2)** 返回矩阵的列数。

<b>dim</b> ( $\begin{bmatrix} 2 & 7 & 1 \\ -8 & 3 & 1 \end{bmatrix}$ )	$\{2\ 3\}$	<b>dim</b> ( $\begin{bmatrix} 2 & 7 & 1 \\ -8 & 3 & 1 \end{bmatrix}$ ) $\blacktriangleright$ <b>L<sub>1</sub>:L<sub>1</sub>(1)</b>	2
--	------------	--	---

## 使用 dim( 创建矩阵

使用 **dim(** (和 **STO►**) 可创建一个新的矩阵 *matrixname*。新矩阵的维数为  $rows \times columns$ ，它的元素都为 **0**。

**{rows,columns}→dim(matrixname)**

```
(2,2)→dim([E])
           (2 2)
[E]
           [[0 0]
            [0 0]]
```

## 使用 dim( 重新定义矩阵维数

**dim(** (和 **STO►**) 一起使用时，将一个已有矩阵 *matrixname* 的维数重新定义为  $rows \times columns$ 。原矩阵 *matrixname* 中的元素，在新矩阵范围之列的保持不变，而不在新矩阵范围之列的则删除掉。另外创建的元素为 **0**。

**{rows,columns}→dim(matrixname)**

## Fill(

**Fill(** 将矩阵 *matrixname* 的每个元素都赋值为 *value*。

**Fill(value,matrixname)**

```
Fill(5,[E])
           Done
[E]
           [[5 5]
            [5 5]]
```

## identity(

**identity(** 返回 *dimension* 行  $\times$  *dimension* 列的单位矩阵。

**identity(dimension)**

## randM(

**randM(** (创建随机矩阵) 返回一个  $rows \times columns$  的随机矩阵，其元素是  $\geq -9$  且  $\leq 9$  的整数。存储在 **rand** 函数中的起始值控制元素值的大小(第 2 章)。

**randM(rows,columns)**

```
0→rand:randM(2,2)
)
           [[0 -7]
            [8 8]]
```

**augment(**

**augment(** 将 *matrixA* 作为新的列追加到 *matrixB*。*matrixA* 和 *matrixB* 必须有相同的行数。

**augment(matrixA,matrixB)**

```

[[1,2][3,4]]→[A]
[[5,6][7,8]]→[B]
]:augment([A],[B]
])
      [[1 2 5 6]
       [3 4 7 8]]
    
```

**Matrlist(**

**Matrlist(** (将矩阵存储为数组) 用 *matrix* 的每个列的元素填充相应的 *listname*。**Matrlist(** 忽略多余的 *listname* 参数。同样, **Matrlist(** 忽略多余的 *matrix* 列。

**Matrlist(matrix,listnameA,...,listname n)**

```

[A]      [[1 2 3]
           [4 5 6]]
Matrlist([A],L1
,L2,L3)  →  L1      (1 4)
           L2      (2 5)
           L3      (3 6)
           Done
    
```

**Matrlist(** 也可使用 *matrix* 中指定的 *column#* 的元素填充 *listname*。若要使用 *matrix* 中指定的列填充一个数组, 必须在 *matrix* 后输入 *column#*。

**Matrlist(matrix,column#,listname)**

```

[A]      [[1 2 3]
           [4 5 6]]
Matrlist([A],3,
L1)      →  L1      (3 6)
           Done
    
```

**Listmatr(**

**Listmatr(** (将数组存储为矩阵) 用每个 *list* 的元素填充 *matrixname* 中相应的列。如果数组的维数不相同, 则 **Listmatr(** 用 0 填充每个外加的 *matrixname* 行。复数数组是无效的。

**Listmatr(listA,...,list n,matrixname)**

```

(1,2,3)→LX  (1 2 3)
(4,5,6)→LY  (4 5 6)
(7,8,9)→LB  (7 8 9)
           →  Listmatr(LX,LY,
           LB,[C])
           Done
           [C]      [[1 4 7]
                    [2 5 8]
                    [3 6 9]]
    
```



## cumSum(

**cumSum(** 返回 *matrix* 中元素的累积和，从第 1 个元素开始累计。每个元素都是所在的列从第 1 个元素到它自己的累积和。

**cumSum(matrix)**

[D]	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$	cumSum([D])	$\begin{bmatrix} 1 & 2 \\ 4 & 6 \\ 9 & 12 \end{bmatrix}$
-----	---	-------------	--

## 行运算

MATRIX MATH 菜单项 **A** 到 **F** 是行运算。可在表达式中使用行运算。行运算不改变内存中的 *matrix*。所有行数和值都可作为表达式输入。可从 MATRIX NAMES 菜单选择运算的矩阵。

## ref( , rref(

**ref(** (行梯形矩阵) 返回实数矩阵 *matrix* 的行梯形矩阵。矩阵的列数必须大于或等于行数。

**ref(matrix)**

**rref(** (简化的行梯形矩阵) 返回实数矩阵 *matrix* 的简化行梯形矩阵。矩阵的列数必须大于或等于行数。

**rref(matrix)**

[B]	$\begin{bmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$	ref([B])	$\begin{bmatrix} 1 & 1.142857143... \\ 0 & 1 & ... \end{bmatrix}$
		rref([B])	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \end{bmatrix}$

**rowSwap(**

**rowSwap(** 返回一个矩阵。它交换 *matrix* 中的行 *rowA* 和行 *rowB*。

**rowSwap(matrix,rowA,rowB)**

```
[F]
  [[2 3 6 9]
   [5 8 4 7]
   [2 5 1 0]
   [6 3 8 5]]
```

```
rowSwap([F],2,4)
  [[2 3 6 9]
   [6 3 8 5]
   [2 5 1 0]
   [5 8 4 7]]
```

**row+(**

**row+(** (行加法) 返回一个矩阵。它将 *matrix* 中的 *rowA* 和 *rowB* 相加，并将结果存储在 *rowB* 中。

**row+(matrix,rowA,rowB)**

```
[[2,5,7][8,9,4]]
→[D]
  [[2 5 7]
   [8 9 4]]
```

```
row+([D],1,2)
  [[2 5 7]
   [10 14 11]]
```

**\*row(**

**\*row(** (行乘法) 返回一个矩阵。它将 *matrix* 中的 *row* 乘以 *value*，并将结果存储在 *row* 中。

**\*row(value,matrix,row)**

**\*row+(**

**\*row+(** (行乘法和加法) 返回一个矩阵。它将 *matrix* 中的 *rowA* 乘以 *value*，再与 *rowB* 相加，并将结果存储在 *rowB* 中。

**\*row+(value,matrix,rowA,rowB)**

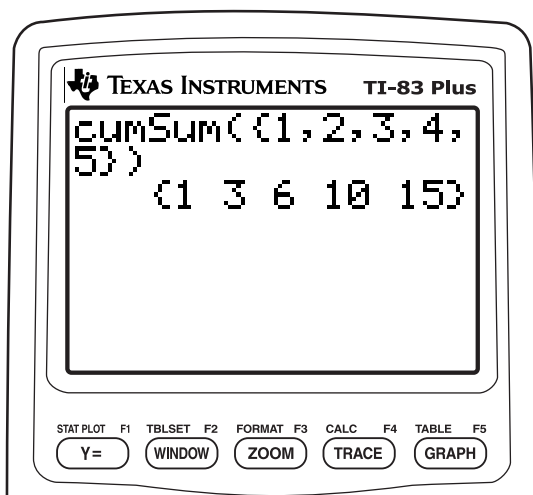
```
[[1,2,3][4,5,6]]
→[E]
  [[1 2 3]
   [4 5 6]]
```

```
*row+(3,[E],1,2)
  [[1 2 3]
   [7 11 15]]
```

# 11 数组

## 目录

入门：生成序列.....	2
命名数组.....	3
存储和显示数组.....	4
输入数组名.....	6
将公式与数组名相连.....	7
在表达式中使用数组.....	9
LIST OPS 菜单.....	10
LIST MATH 菜单.....	17



## 入门：生成序列

入门是对本章内容的快速介绍。详细信息请阅读本章。

计算序列  $1/A^2$  的前 8 项。将结果保存在用户所创建的数组中。然后用分数形式显示结果。本例从主屏幕的一个空行开始。

1. 按  $\boxed{2\text{nd}} \boxed{[\text{LIST}]} \boxed{\blacktriangleright}$  显示 LIST OPS 菜单。

```
NAMES OPS MATH
1:SortA(
2:SortD(
3:dim(
4:Fill(
5:seq(
6:cumSum(
7:↓List(
```

2. 按 **5** 选定 **5:seq(**，将 **seq(** 粘贴到当前光标处。

3. 按 **1**  $\boxed{[ ]}$   $\boxed{[\text{ALPHA}]} \boxed{[A]} \boxed{[x^2]} \boxed{[ ]}$   $\boxed{[\text{ALPHA}]} \boxed{[A]} \boxed{[ ]}$  **1**  $\boxed{[ ]}$  **8**  $\boxed{[ ]}$  **1**  $\boxed{[ ]}$  输入此序列。

```
seq(1/A^2,A,1,8,1
)→SEQ1
```

4. 按  $\boxed{[\text{STO} \blacktriangleright]}$ ，然后按  $\boxed{2\text{nd}} \boxed{[\text{ALPHA}]}$  开启字母锁。按  $\boxed{[S]}$   $\boxed{[E]}$   $\boxed{[Q]}$ ，然后按  $\boxed{[\text{ALPHA}]}$  关闭字母锁。按 **1** 完成数组名的输入。

5. 按  $\boxed{[\text{ENTER}]}$  生成数组并存储在 **SEQ1**。此数组显示在主屏幕。省略号 (...) 表示数组显示超出了观察窗口。反复按  $\boxed{\blacktriangleright}$  (或按住  $\boxed{\blacktriangleright}$ ) 滚动数组并观察所有数组元素。

```
seq(1/A^2,A,1,8,1
)→SEQ1
{1 .25 .1111111...
█
```

6. 按  $\boxed{2\text{nd}} \boxed{[\text{LIST}]}$  显示 LIST NAMES 菜单。按 **7** 选择 **7:seq(** 将 **LSEQ1** 粘贴到当前光标处。(如果 **SEQ1** 不是 LIST NAMES 菜单的第 **7** 项，则将光标移动到 **SEQ1**，然后按  $\boxed{[\text{ENTER}]}$ 。)

```
NAMES OPS MATH
1:L1
2:L2
3:L3
4:L4
5:L5
6:L6
7:SEQ1
```

7. 按  $\boxed{[\text{MATH}]}$  显示 MATH 菜单。按 **1** 选择 **1:Frac**，将 **Frac** 粘贴到当前光标处。

8. 按  $\boxed{[\text{ENTER}]}$  用分数形式显示序列。反复按  $\boxed{\blacktriangleright}$  (或按住  $\boxed{\blacktriangleright}$ ) 滚动数组并观察所有数组元素。

```
seq(1/A^2,A,1,8,1
)→SEQ1
{1 .25 .1111111...
LSEQ1Frac
{1 1/4 1/9 1/16...
█
```

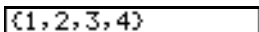
## 使用 TI-83 Plus 数组名 L1 到 L6

TI-83 Plus 在内存中有 6 个数组名：**L1**、**L2**、**L3**、**L4**、**L5** 和 **L6**。数组名 **L1** 到 **L6** 位于键盘的数字键 **[1]** 到 **[6]** 上方。若要将这些数组名之一粘贴到一个有效屏幕，按 **[2nd]**，然后按相应的键。当重设内存时，**L1** 到 **L6** 被存储在统计数组编辑器列 **1** 到 **6**。

## 在主屏幕创建数组名

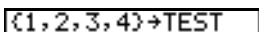
要在主屏幕中创建一个数组名，请按下列步骤进行：

1. 按 **[2nd]** **[{]**，输入一个或多个数组元素，然后按 **[2nd]** **[}]**。用逗号分隔数组元素。数组元素可以是实数、复数或表达式。



```
{1, 2, 3, 4}
```

2. 按 **[STO▶]**。
3. 按 **[ALPHA]** **[A 到 Z 之间的字母或 θ]** 输入数组名的第 1 个字母。
4. 输入 0 到 4 个字母、 $\theta$  或数字来完成数组名的输入。



```
{1, 2, 3, 4}→TEST
```

5. 按 **[ENTER]**。数组显示在下一行。数组名和数组元素被存储在内存中。数组名成为 LIST NAMES 菜单的一项。



```
{1, 2, 3, 4}→TEST
(1 2 3 4)
NAME OPS MATH
1: SEQ1
2: T123
3: TEST
```

**注意：**如果要在统计数组编辑器中观察一个用户创建的数组，则必须将它保存在统计数组编辑器（第 12 章）。

也可在以下 4 处创建数组名：

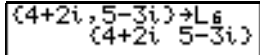
- 在统计数组编辑器的 **Name=** 提示处
- 在统计图编辑器中的 **Xlist:**、**Ylist:** 或 **Data List:** 提示处
- 在推理统计编辑器中的 **List:**、**List1:**、**List2:**、**Freq:**、**Freq1:**、**Freq2:**、**XList:** 或 **YList:** 提示处
- 在主屏幕上使用 **SetUpEditor**

可以创建 TI-83 Plus 内存空间允许的足够多数组名。

### 将元素存储到数组中

可用两种方法之一存储数组元素。

- 在主屏幕中使用大括号和 **STO►**。



```
(4+2i, 5-3i)→L6
(4+2i 5-3i)
```

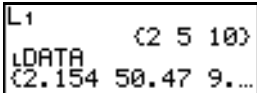
- 使用统计数组编辑器（第 12 章）。

数组最多有 999 个元素。

**提示：**当把一个复数存储到数组中时，整个数组转换为复数数组。若要将此数组转换为实数数组，显示主屏幕，然后输入 **real(listname)►listname**。

### 在主屏幕中显示数组

若要在主屏幕中显示数组元素，输入数组名（必要时前面输入 **L**；请参阅页 11-16），然后按 **ENTER**。省略号指示数组元素超出了观察窗口。反复按 **▢**（或按住 **▢**）滚动数组并观察所有数组元素。



```
L1
(2 5 10)
L1DATA
(2.154 50.47 9....)
```

---

## 拷贝数组

拷贝一个数组，将它存储到另一个数组。

```
LTEST  
  {1 2 3 4}  
LTEST→TEST2  
  {1 2 3 4}
```

## 访问数组元素

可将一个值存储到一个指定的数组元素 *element*，或从这个元素重新调用此值。可存储值到当前数组长度内的任意元素或当前长度外的某个元素。

*listname(element)*

```
{1,2,3}→L3  
  {1 2 3}  
4→L3(4):L3  
  {1 2 3 4}  
L3(2)  
  2
```

## 从内存中删除数组

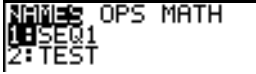
要从内存中删除包括 L1 到 L6 的数组，请使用 MEMORY MANAGEMENT/DELETE 子菜单（第 18 章）。重设内存可恢复 L1 到 L6。在统计数组编辑器中删除数组不能将它从内存中删除。

## 在图象中使用数组

可用数组画出一组曲线（第 3 章）。

## 使用 LIST NAMES 菜单

要显示 LIST NAMES 菜单，请按 [2nd] [LIST]。每一项都是用户创建的数组名。LIST NAMES 菜单项自动按字母顺序排序。只有前 10 项用 1 到 9，然后是 0 作为标记。要跳到以 A 到 Z 之间的特定字母或  $\theta$  开头的第 1 个数组名，请按 [ALPHA] 和该字母或  $\theta$ 。

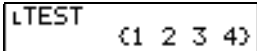


**提示：**从菜单顶部，按 [ ] 可移动到底部。从底部按 [ ] 可返回到顶部。

**注意：**LIST NAMES 菜单省略了数组名 L1 到 L6。可从键盘直接输入 L1 到 L6（页 11-3）。

从 LIST NAMES 菜单选定一个数组名时，此数组名被粘贴到当前光标处。

- 当数组名被粘贴到非数组名数据也有效的地方（如主屏幕）时，数组名符号 L 显示在数组名之前。

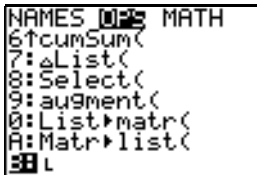


- 当数组名被粘贴到只有数组名是有效输入时，例如统计数组编辑器的 **Name=** 提示或统计图编辑器的 **XList:** 和 **YList:** 提示，则 L 符号不显示在数组名之前。

## 直接输入用户创建的数组名

要直接输入一个用户创建的数组名，请按下列步骤进行：

- 按 [2nd] [LIST] [ ] 显示 LIST OPS 菜单。
- 选择 **B:L**，粘贴 L 到当前光标处。L 不总是必需的（页 11-16）。



**注意：**也可从 CATALOG（第 15 章）粘贴 L 到当前光标处。

- 输入构成数组名的字符。





## 将公式与数组名相连

如果将一个公式与一个数组名相连，则此数组中的每个元素都是公式的计算结果。执行时，相连的公式必须涉及一个数组。

当相连的公式有任何改变时，相连的数组自动更新。

- 当编辑一个被公式引用的数组元素时，则相连此公式的数组中相应元素更新。
- 当编辑公式本身时，相连此公式的数组所有元素更新。

例如，下面第 1 个屏幕表示元素被存储到 **L3**，公式 **L3+10** 连接数组名 **LADD10**。双引号表示此公式被连接到 **LADD10**。**LADD10** 中的每个元素是 **L3** 的元素与 10 的和。

```
(1, 2, 3)→L3      {1 2 3}
"L3+10"→LADD10
L3+10
LADD10             {11 12 13}
```

下面屏幕显示另一个数组 **L4**。**L4** 的元素是连接到 **L3** 的同一公式的和。然而，没有输入引号，因此公式没有连接到 **L4**。

在下一行，**-6→L3(1):L3** 将 **L3** 的第 1 个元素更改为 **-6**，然后重新显示 **L3**。

```
L3+10→L4          {11 12 13}
-6→L3(1):L3      {-6 2 3}
```

最后一个屏幕显示编辑 **L3** 可更新 **LADD10**，但是不能更改 **L4**。这是因为公式 **L3+10** 被连接到 **LADD10**，但是没有连接到 **L4**。

```
LADD10             {4 12 13}
L4                  {11 12 13}
```

注意：若要观察连接到一个数组名的公式，使用统计数组编辑器（第 12 章）。

### 在主屏幕或程序中将公式连接到数组

要从主屏幕的一个空行或从程序中将公式连接到数组名，请按下列步骤进行：

1. 按  $\boxed{\text{ALPHA}}$  ["], 输入公式（必须涉及一个数组），再按  $\boxed{\text{ALPHA}}$  ["]。

**注意：**当公式中包含一个以上的数组名时，每个数组的长度必须相同。

2. 按  $\boxed{\text{STO}}$ 。

3. 输入要连接公式的数组名。

- 按  $\boxed{2\text{nd}}$ ，然后输入 TI-83 Plus 数组名 L1 到 L6。
- 按  $\boxed{2\text{nd}}$  [LIST] 并从 LIST NAMES 菜单选择用户创建的数组名。
- 使用  $\boxed{\downarrow}$  直接输入用户创建的数组名（页 11-16）。

4. 按  $\boxed{\text{ENTER}}$ 。

The image shows a TI-83 Plus calculator screen with the following text:   
{4,8,9}→L1  
{4 8 9}  
"5\*L1"→L1LIST  
5\*L1  
L1LIST  
(20 40 45)

**注意：**统计数组编辑器在每个和公式相连的数组名旁显示一个公式锁符号。第 12 章讲述了如何用统计数组编辑器将公式连接到数组，编辑连接的公式，以及从数组中拆离公式。

### 从数组中拆离公式

要从数组中拆离（清除）连接的公式，可使用以下几种方法：

例如：

- 在主屏幕中输入  $\boxed{\text{""}}$ →*listname*。
- 编辑连接公式的数组的任意元素。
- 使用统计数组编辑器（第 12 章）。
- 使用 **ClrList** 或 **ClrAllList** 从数组中拆离公式（第 18 章）。

## 在表达式中使用数组

可以使用以下 3 种方法之一在表达式中使用数组。按 **[ENTER]** 时，对每个数组元素计算表达式，然后显示一个数组。

- 在表达式中使用 **L1-L6** 或者用户创建的任何数组名。

Calculator screen showing two lines of input and output:  
Line 1:  $(2, 5, 10) * L1$  →  $(2 \ 5 \ 10)$   
Line 2:  $20 / L1$  →  $(10 \ 4 \ 2)$

- 直接输入数组元素（页 11-3 的步骤 1）。

Calculator screen showing input and output:  
Input:  $20 / (2, 5, 10)$   
Output:  $(10 \ 4 \ 2)$

- 使用 **[2nd] [RCL]** 重新调用数组内容到光标处的表达式（第 1 章）。

Calculator screen showing two steps:  
Step 1: **Rcl L1** (cursor at the end)  
Step 2:  $(2, 5, 10)^2$  →  $(4 \ 25 \ 100)$

**注意：**必须将用户创建的数组名通过从 LIST NAMES 菜单中选择它们来粘贴到 **Rcl** 提示处。不能用 **L** 直接输入数组名。

## 在数学运算中使用数组

可用数组作为一些数学运算的输入值。其他章节和附录 A 指出了数组是否有效。对每个数组元素进行计算，然后显示数组。

- 当在运算中使用数组时，运算必须是对数组的每个元素是有效的。在图象中，无效的元素被忽略，例如  $\sqrt{(1, 0, -1)}$  中的  $-1$ 。

Calculator screen showing input and output:  
Input:  $\sqrt{(1, 0, -1)}$   
Output: 返回一个错误。

Calculator screen showing input and output:  
Input: **Plot1 Plot2 Plot3**  
 $\sqrt{Y1} \sqrt{(1, 0, -1)}$   
Output: 画出  $X * \sqrt{1}$  和  $X * \sqrt{0}$ ，但忽略  $X * \sqrt{-1}$ 。

- 当在一个二元运算中使用两个数组时，每个数组的长度必须相等。对相应的元素进行计算。

Calculator screen showing input and output:  
Input:  $(1, 2, 3) + (4, 5, 6)$   
Output:  $(5 \ 7 \ 9)$

- 当在一个二元运算中使用一个数组和一个值时，这个值被用于数组的每个元素。

Calculator screen showing input and output:  
Input:  $(1, 2, 3) + 4$   
Output:  $(5 \ 6 \ 7)$

## LIST OPS 菜单

要显示 LIST OPS 菜单，请按 **[2nd] [LIST] [▢]**。

NAMES **OPS** MATH

1: SortA(	按升序排序数组。
2: SortD(	按降序排序数组。
3: dim(	设置数组长度。
4: Fill(	用常数填充所有元素。
5: seq(	创建序列。
6: cumSum(	返回累计求和的数组。
7: ΔList(	返回相邻元素的差。
8: Select(	选择指定的数据点。
9: augment(	合并两个数组。
0: List▶matr(	将数组存储到矩阵。
A: Matr▶list(	将矩阵存储到数组。
B: L	指定数组名数据类型。

### SortA(、SortD(

**SortA(**（按升序排序）将数组元素按照从小到大的顺序排序。**SortD(**（按降序排序）将数组元素按照从大到小的顺序排序。复数数组按模来排序。

对于一个数组，**SortA(**和**SortD(**将 *listname* 的元素排序并更新内存中的数组。

**SortA(listname)**

```
(5,6,4)→L3
SortA(L3)
L3
      Done
      (4 5 6)
```

**SortD(listname)**

```
SortD(L3)
L3
      Done
      (6 5 4)
```

对于两个或更多的数组，**SortA(**和**SortD(**按 *keylistname* 排序，然后将每个 *dependlist* 的元素排序，顺序与 *keylistname* 中相应元素的顺序相同。所有数组必须有相同的长度。

**SortA(keylistname,dependlist1[,dependlist2,...,dependlist n])**

**SortD(keylistname,dependlist1[,dependlist2,...,dependlist n])**

```
(5,6,4)→L4
(1,2,3)→L5
SortA(L4,L5)
L4
      Done
      (4 5 6)
L5
      (3 1 2)
```

```
SortA(L4,L5)
L4
      Done
      (4 5 6)
L5
      (3 1 2)
```

**注意：**在本例中，**5** 是 **L4** 的第 1 个元素，**1** 是 **L5** 的第 1 个元素。用 **SortA(L4,L5)** 排序后，**5** 成为 **L4** 的第 2 个元素，同样，**1** 成为 **L5** 的第 2 个元素。

**注意：****SortA(**和**SortD(**与 STAT EDIT 菜单中的 **SortA(**和**SortD(**相同（第 12 章）。

## 用 dim( 查找数组长度

**dim(** (长度) 返回 *list* 的长度 (元素的数目)。

**dim(list)**

```
dim({1,3,5,7}) 4
```

## 用 dim( 创建数组

可用 **dim(** 和 **STO▶** 创建新的数组 *listname* , 长度 *length* 在 1 到 999 之间, 新数组的元素都为 0。

*length*→**dim(listname)**

```
3→dim(L2)      3
L2              {0 0 0}
```

## 用 dim( 重定义数组长度

可用 **dim** 和 **STO▶** 重定义已有的数组 *listname* , 长度 *length* 在 1 到 999 之间。

- 新数组中原有的元素保持不变。
- 新数组中多出的元素由 0 填充。
- 原有数组中多出的元素被删除。

*length*→**dim(listname)**

```
{4,8,6}→L1     {4 8 6}
4→dim(L1)      4
L1              {4 8 6 0}
```

```
3→dim(L1)      3
L1              {4 8 6}
```

## Fill(

**Fill(** 将 *listname* 中的每个元素替换为 *value* 。

**Fill(value,listname)**

```
{3,4,5}→L3     {3 4 5}
Fill(8,L3)     Done
L3              {8 8 8}
```

```
Fill(4+3i,L3)  Done
L3              {4+3i 4+3i 4+3i}
```

注意: **dim(** 和 **Fill(** 与 **MATRIX MATH** 菜单中的 **dim(** 和 **Fill(** 相同 (第 10 章)。

**seq(**

**seq(**(序列) 返回一个数组，它的每个元素是对 *variable* 的每个值（即从 *begin* 到 *end*，增量为 *increment*）计算 *expression* 的结果。*variable* 不需在内存中定义。*increment* 可以是负数；默认的 *increment* 是 1。**seq(** 不能用在 *expression* 之中。复数数组无效。

**seq(expression,variable,begin,end[,increment])**

```
seq(A^2,A,1,11,3)
  {1 16 49 100}
```

**cumSum(**

**cumSum(**(累计求和) 返回 *list* 中由第 1 个元素开始的累计求和。*list* 的元素可以是实数或复数。

**cumSum(list)**

```
cumSum({1,2,3,4,
5})
  {1 3 6 10 15}
```

**ΔList(**

**ΔList(** 返回一个由 *list* 中相邻元素的差组成的数组。**ΔList** 使 *list* 的第 2 个元素减第 1 个元素，第 3 个元素减第 2 个元素，依次类推。差的数组总是比初始数组 *list* 少一个元素。*list* 的元素可以是实数或复数。

**ΔList(list)**

```
{20,30,45,70}→LD
IST
  {20 30 45 70}
ΔList(LDIST)
  {10 15 25}
```

**Select(**



**Select(** 从一个散点图或 xy 直线图（仅应用于它们）中选择一个或多个指定的数据点，然后将选定的数据点存储到两个新数组，即 *xlistname* 和 *ylistname*。例如，可用 **Select(** 选择并分析部分绘出的 CBL 或 CBR 数据。

**Select(xlistname,ylistname)**

**注意：**使用 **Select(** 之前，必须选定（开启）一个散点图或 xy 直线图。此外，这个图必须显示在当前观察窗口内（页 11-13）。

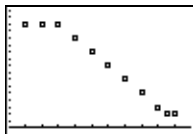
## 使用 Select( 之前

在使用 **Select(** 之前, 请执行以下步骤:

1. 创建两个数组名并输入数据。
2. 开启一个统计图, 选择  (散射图) 或  (xy 直线), 对 **Xlist:** 和 **Ylist:** 输入两个数组名 (第 12 章)。
3. 使用 **ZoomStat** 画出数据 (第 3 章)。

```
{1,2,3,4,5,6,7,8  
,9,9.5,10}→DIST  
{1,2,3,4,5,6,7...  
{15,15,15,13,11,  
9,7,5,3,2,2}→TIM  
E  
{15,15,15,13,11...
```

```
Plot1 Plot2 Plot3  
Off Off Off  
Type: [ ] [ ] [ ]  
Xlist: DIST  
Ylist: TIME  
Mark: [ ] [ ] [ ]
```



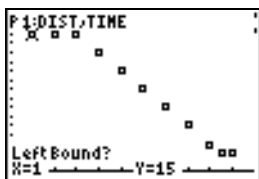
## 使用 Select( 从图中选定数据点

要从一个散射图或 xy 直线图中选定数据点, 请按下列步骤进行:

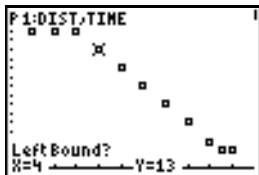
1. 按 **[2nd]** **[LIST]** **[ ]** **8** 从 LIST OPS 菜单选择 **8:Select(**。 **Select(** 被粘贴到主屏幕。
2. 输入 *xlistname*, 按 **[ ]**, 输入 *ylistname*, 然后按 **[ ]** 指明数组名, 选定的数据将被存储到这两个数组中。

```
Select(L1,L2) [ ]
```

3. 按 **[ENTER]**。屏幕显示图象, 左下角显示 **Left Bound?**。

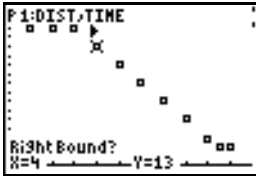


4. 按 **[ ]** 或 **[ ]** (如果选择 1 个以上的统计图) 将光标移动到要选择数据点的统计图上。
5. 按 **[ ]** 和 **[ ]** 将光标移动到要作为左边界的统计图数据点。

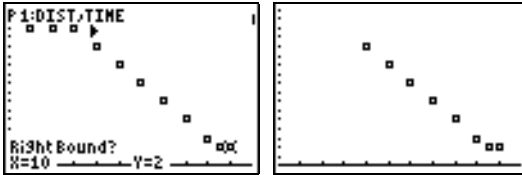


使用 Select( 从图中选定数据点 (续)

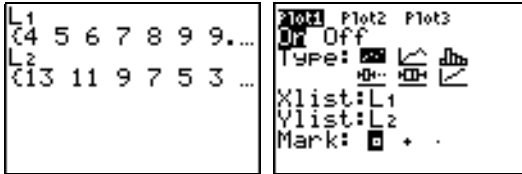
- 按 **[ENTER]**。图象屏幕上有个 **▶** 指示符显示左边界。屏幕左下角显示 **Right Bound?**。



- 按 **[◀]** 或 **[▶]** 将光标移动到要作为右边界的统计图点，然后按 **[ENTER]**。



选定点的 x 值和 y 值被存储在 *xlistname* 和 *ylistname*。一个新的 *xlistname* 和 *ylistname* 的统计图替代了原来的统计图。统计图编辑器中的数组被更新。



**注意：**两个新数组 (*xlistname* 和 *ylistname*) 将包括作为左边界和右边界的点。同样， $left-bound\ x-value \leq right-bound\ x-value$  必须成立。



## augment(

**augment(** 将 *listA* 和 *listB* 的元素合并。数组元素可以是实数或复数。

**augment(listA,listB)**

```
(1,17,21)→L3
      {1 17 21}
augment(L3,(25,3
0,41})
(1 17 21 25 30 ...
```

## List→matr(

**List→matr(** (将数组存储到矩阵中) 将所有数组中每个元素填充到 *matrix-name* 的对应列。如果数组的长度不相等, 则 **List→matr(** 将多余的 *matrix-name* 行用 **0** 填充。复数数组无效。

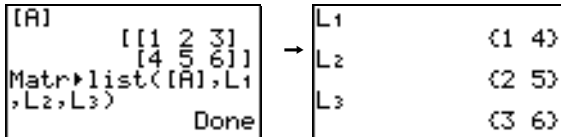
**List→matr(list1,list2,...,list n,matrixname)**

```
(1,2,3)→LX
      {1 2 3}
(4,5,6)→LY
      {4 5 6}
(7,8,9)→LB
      {7 8 9}
→
List→matr(LX,LY,
LB,[C])
      Done
[C]
      [[1 4 7]
       [2 5 8]
       [3 6 9]]
```

**Matr▶list(**

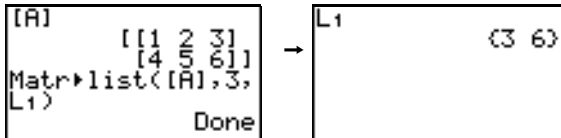
**Matr▶list(** (将矩阵存储到数组中) 将 *matrix* 中的每个列作为元素填充相应的 *listname*。如果 *listname* 参数的数目超过 *matrix* 中列的数目, 则 **Matr▶list(** 忽略多余的 *listname* 参数。同样, 如果 *matrix* 中列的数目超过 *listname* 参数的数目, 则 **Matr▶list(** 忽略多余的 *matrix* 列。

**Matr▶list(matrix,listname1,listname2,...,listname n)**



**Matr▶list(** 也可用 *matrix* 中指定的列 *column#* 作为元素填充相应的 *listname*。若要用 *matrix* 中指定的列填充数组, 则必须在 *matrix* 后输入 *column#*。

**Matr▶list(matrix,column#,listname)**



位于 1 到 5 个字符之前的 **L** 表示这些字符是用户创建的数组 *listname*。*listname* 可包含字母、 $\theta$  和数字, 但第一个字符必须是 A 到 Z 之间的字母或  $\theta$ 。

**Llistname**

一般在输入用户创建的数组名于其他输入也有效的地方 (例如在主屏幕中) 时, **L** 必须位于用户创建的数组名之前。没有 **L**, 则 TI-83 Plus 可能将用户创建的数组名误解为两个或多个字符的蕴含乘法。

在输入用户创建的数组名于只有数组名是有效输入 (例如在统计数组编辑器的 **Name=** 提示处或统计图编辑器的 **Xlist:** 和 **Ylist:** 提示处) 时, 则 **L** 不必出现在用户创建的数组名之前。如果在不必要时输入 **L**, 则 TI-83 Plus 将忽略此输入。

## LIST MATH 菜单

要显示 LIST MATH 菜单，请按  $\boxed{2nd}$  [LIST]  $\boxed{\blacktriangleleft}$ 。

NAMES OPS **MATH**

1: min(	返回数组的最小元素。
2: max(	返回数组的最大元素。
3: mean(	返回数组的平均值。
4: median(	返回数组的中位数。
5: sum(	返回数组元素的和。
6: prod(	返回数组元素的积。
7: stdDev(	返回数组的标准差。
8: variance(	返回数组的方差。

### min( , max(

**min(** (最小值) 和 **max(** (最大值) 返回数组 *listA* 的最小或最大元素。如果比较两个数组，则它返回一个数组，它包含数组 *listA* 和 *listB* 中每一对元素的最小值或最大值。对于复数数组，则返回模的最小值或最大值。

**min(listA,listB)**

**max(listA,listB)**

```
min({1,2,3},{3,2,1})
      {1 2 1}
max({1,2,3},{3,2,1})
      {3 2 3}
```

注意: **min(** 和 **max(** 与 MATH NUM 菜单的 **min(** 和 **max(** 相同。

### mean( , median(

**mean(** 返回数组 *list* 的平均值。**median(** 返回数组 *list* 的中位数。*freqlist* 的默认值是 1。每个 *freqlist* 元素是 *list* 中相应元素的连续出现的频率。对复数数组无效。

**mean(list,freqlist)**

**median(list,freqlist)**

```
mean({1,2,3},{3,2,1})
      1.666666667
median({1,2,3})
      2
```

**sum( , prod(**

**sum(** (和) 返回数组 *list* 中元素的总和。 *start* 和 *end* 是可选项；它们指定元素的范围。 *list* 元素可以是实数或复数。

**prod(** 返回数组 *list* 中所有元素的积。 *start* 和 *end* 是可选项；它们指定元素的范围。 *list* 元素可以是实数或复数。

**sum(list[,start,end])      prod(list[,start,end])**

```
L1 (1 2 5 8 10)
sum(L1)                26
sum(L1,3,5)           23
```

```
L1 (1 2 5 8 10)
Prod(L1)               800
Prod(L1,3,5)          400
```

**数值序列的和与积**

将 **sum(** 或 **prod(** 与 **seq(** 组合可获得：

$$\sum_{x=lower}^{upper} expression(x) \quad \prod_{x=lower}^{upper} expression(x)$$

若要计算  $\sum 2^{(N-1)}$ , N=1 到 4 :

```
sum(seq(2^(N-1),
N,1,4,1))
15
```

**stdDev( , variance(**

**stdDev(** 返回数组 *list* 中元素的标准差。 *freqlist* 的默认值是 1。每个 *freqlist* 元素是 *list* 中相应元素的连续出现的频率。对复数数组无效。

**variance(** 返回数组 *list* 中元素的方差。 *freqlist* 的默认值是 1。每个 *freqlist* 元素是 *list* 中相应元素的连续出现的频率。对复数数组无效。

**stdDev(list[,freqlist])      variance(list[,freqlist])**

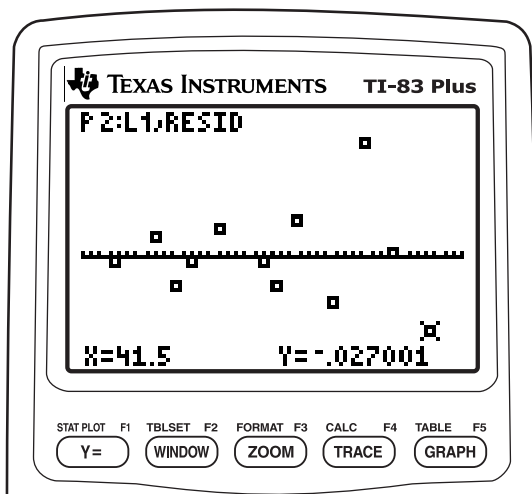
```
stdDev({1,2,5,-6
,3,-2})
3.937003937
```

```
variance({1,2,5,
-6,3,-2})
15.5
```

# 12 统计

## 目录

入门：摆锤长度与周期.....	2
建立统计分析.....	10
使用统计数组编辑器.....	11
将公式与数组名相连.....	14
拆离数组名中的公式.....	16
切换统计数组编辑器环境.....	17
统计数组编辑器环境.....	18
STAT EDIT 菜单.....	20
回归模型功能.....	22
STAT CALC 菜单.....	24
统计变量.....	29
程序中的统计分析.....	30
统计绘图.....	31
程序中的统计绘图.....	37



## 入门：摆锤长度与周期

入门是对本章内容的快速介绍。详细信息请阅读本章。

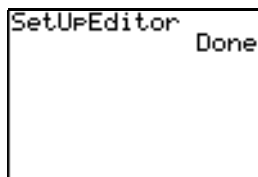
一组学生试图确定摆锤长度与其周期（摆锤往返摆动一次）之间的数学关系。他们用绳和垫圈制作了一简单的摆锤，然后将之从天花板挂下。他们记录了 12 种绳长时摆锤的周期。\*

长度 (cm)	时间 (sec)
6.5	0.51
11.0	0.68
13.2	0.73
15.0	0.79
18.0	0.88
23.1	0.99
24.4	1.01
26.6	1.08
30.5	1.13
34.3	1.26
37.6	1.28
41.5	1.32

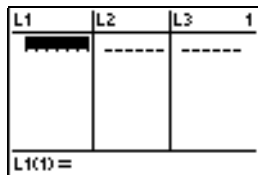
- 按 **MODE**  $\downarrow$   $\downarrow$   $\downarrow$  **ENTER** 设置 **Func** 图象方式。
- 按 **STAT** **5** 选择 **5:SetUpEditor**。**SetUpEditor** 粘贴到主屏幕。

按 **ENTER**。将删除统计数组编辑器 **1-20** 列中的数组，并将数组 **L1-L6** 保存到 **1-6** 列中。

**注意：**删除统计数组编辑器中的数组，不会将之从内存中删除。



- 按 **STAT** **1** 选择 **STAT EDIT** 菜单中的 **1:Edit**，显示统计数组编辑器。如果元素保存在 **L1** 和 **L2** 中，按  $\uparrow$  将光标移至 **L1**，然后按 **CLEAR** **ENTER**  $\downarrow$   $\uparrow$  **CLEAR** **ENTER** 清除这两个数组。按  $\leftarrow$  将矩形条光标返回至 **L1** 的第一行。



\* 该例选自由北卡罗来纳科学与数学专科学校编写的 *Contemporary Precalculus Through Applications*，并加以改编。已得到 Janson 出版公司 (Dedham, MA. 1-800-322-MATH) 的认可。© 1992. 版权所有。

4. 按 **6** **□** **5** **[ENTER]**，将第一个摆锤绳长 (6.5 cm) 保存到 **L1**。矩形条光标移至下一行。重复该步骤，将页 12-2 中表内的 12 个绳长值输入到 **L1**。

L1	L2	L3	1
24.4			
26.6			
30.9			
34.3			
37.6			
41.5			
---			
L1(13) =			

5. 按 **□** 将矩形条光标移至 **L2** 的第一行。

按 **□** **51** **[ENTER]**，将第一个时间值 (.51 sec) 保存到 **L2** 中。矩形条光标移至下一行。重复该步骤，将页 12-2 中表内的 12 个时间值输入到 **L2** 中。

L1	L2	L3	2
24.4	1.01		
26.6	1.08		
30.9	1.13		
34.3	1.28		
37.6	1.28		
41.5	1.32		
---			
L2(13) =			

6. 按 **[Y=]** 显示 **Y=** 编辑器。

如果必要，按 **[CLEAR]** 清除函数 **Y1**。必要时，按 **[□]**、**[ENTER]** 和 **[▶]** 关闭 **Y=** 编辑器顶行中的 **Plot1**、**Plot2** 和 **Plot3**（第 3 章）。必要时，按 **[▼]**、**[◀]** 和 **[ENTER]** 取消选择全部函数。

Plot1	Plot2	Plot3
Y1 =		
Y2 =		
Y3 =		
Y4 =		
Y5 =		
Y6 =		
Y7 =		

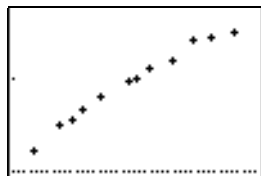
7. 按 **[2nd]** **[STAT PLOT]** **1** 选择 **STAT PLOTS** 菜单中的 **1:Plot1**。显示出 plot 1 的统计图编辑器。

Plot1	Plot2	Plot3
On Off		
Type: [ ] [ ] [ ]		
Xlist: L1		
Ylist: L2		
Mark: [ ] + .		

8. 按 **[ENTER]** 选择 **On**，开启 plot 1。按 **[▼]** **[ENTER]** 选择 **[ ]**（散点图）。按 **[▼]** **[2nd]** **[L1]** 为 plot 1 指定 **Xlist:L1**。按 **[▼]** **[2nd]** **[L2]** 为 plot 1 指定 **Ylist:L2**。按 **[▼]** **[▶]** **[ENTER]** 选择 **+** 作为散点图中每个数据点的 **Mark**。

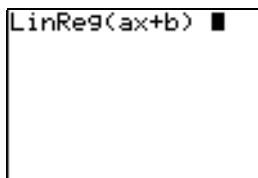
Plot1	Plot2	Plot3
On Off		
Type: [ ] [ ] [ ]		
Xlist: L1		
Ylist: L2		
Mark: [ ] + [ ] .		

9. 按 **[ZOOM]** **9** 选择 **ZOOM** 菜单中的 **9:ZoomStat**。窗口变量自动调整，并显示 plot 1。这是一个时间—长度数据的散射图。



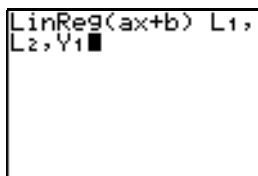
因为时间—长度数据的散点图看上去几乎是线性的，所以可以用直线拟合这些数据。

10. 按 **[STAT]** **[>]** **4** 选择 STAT CALC 菜单中的 **4:LinReg(ax+b)**（线性回归方式）。**LinReg(ax+b)** 粘贴到主屏幕。



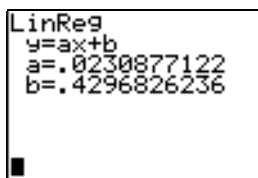
```
LinReg(ax+b) ■
```

11. 按 **[2nd]** **[L1]** **[,]** **[2nd]** **[L2]** **[,]**。按 **[VARS]** **[>]** **1** 显示子菜单 VARS Y-VARS FUNCTION，然后按 **1** 选择 **1:Y1**。**L1**、**L2** 和 **Y1** 作为 **LinReg(ax+b)** 的参数粘贴到主屏幕。



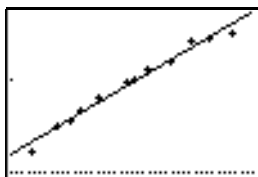
```
LinReg(ax+b) L1,  
L2,Y1 ■
```

12. 按 **[ENTER]** 执行 **LinReg(ax+b)**。对 **L1** 和 **L2** 中的数据作线性回归计算。**a** 和 **b** 的值显示在主屏幕中。线性回归方程保存在 **Y1** 中。计算所得的离差自动保存在数组名 **RESID** 中，该数组名成为 LIST NAMES 菜单中的一项。



```
LinReg  
y=ax+b  
a=.0230877122  
b=.4296826236  
■
```

13. 按 **[GRAPH]**，显示出回归线和散点图。





回归线正好处于散射图的中间，似乎拟合得很好。不过，离差图可提供有关拟合情况的更多信息。

14. 按 **[STAT]** 1 选择 **1:Edit**。将显示统计数组编辑器。

按 **[▶]** 和 **[▲]** 移动光标至 **L3**。

按 **[2nd]** **[INS]**。一个未命名列显示在 **3** 中；**L3**、**L4**、**L5** 和 **L6** 右移一列。在输入行中显示 **Name=** 提示，且字母锁开启。

L1	L2	<del>LIST</del> 3
6.5	.51	
11	.68	
13.2	.73	
15	.79	
18	.88	
23.1	.99	
24.4	1.01	

Name=**█**

15. 按 **[2nd]** **[LIST]** 显示 **LIST NAMES** 菜单。

如果必要，按 **[▼]** 将光标移至数组名 **RESID** 上。

NAME	OPS	MATH
<b>RESID</b>		

16. 按 **[ENTER]** 选择 **RESID**，并将之粘贴到统计数组编辑器中的 **Name=** 提示处。

L1	L2	<del>LIST</del> 3
6.5	.51	
11	.68	
13.2	.73	
15	.79	
18	.88	
23.1	.99	
24.4	1.01	

Name=RESID~~█~~

17. 按 **[ENTER]**。**RESID** 保存到统计数组编辑器的列 **3** 中。

反复按 **[▼]** 以检查离差。

L1	L2	<del>LIST</del> 3
6.5	.51	-.0698
11	.68	-.0026
13.2	.73	-.0044
15	.79	.014
18	.88	.02474
23.1	.99	.02698
24.4	1.01	.01698

RESID = (-.0697527...

注意到前三个离差是负的，它们对应 **L1** 中最短的摆锤绳长。接下去五个离差是正的，最后四个中的三个是负的。后者对应于 **L1** 中较长的摆锤绳长。绘制离差图将更清楚地表示出这种方式。

18. 按 **[2nd]** **[STAT PLOT]** **2**，选择菜单 STAT PLOTS 中的 **2:Plot2**。将显示 plot 2 的统计图编辑器。



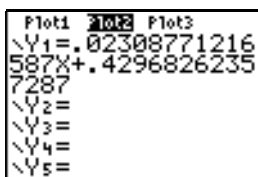
19. 按 **[ENTER]** 选择 **On**，开启 plot 2。

按 **[ ]** **[ENTER]** 选择 **[ ]**（散点图）。按 **[ ]** **[2nd]** **[L1]** 为 plot 2 指定 **Xlist:L1**。按 **[ ]** **[R]** **[E]** **[S]** **[I]** **[D]**（字母锁开启）为 plot 2 指定 **Ylist:RESID**。按 **[ ]** **[ENTER]** 选择 **[ ]** 作为散点图中每个数据点的标记。

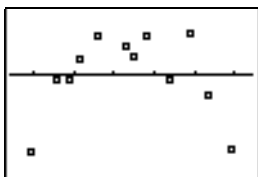


20. 按 **[Y=]** 显示 **Y=** 编辑器。

按 **[ ]** 将光标移至等号“=”，然后按 **[ENTER]** 取消选定 **Y1**。按 **[ ]** **[ENTER]** 关闭 plot 1。



21. 按 **[ZOOM]** **9** 选择 **ZOOM** 菜单中的 **9:ZoomStat**。窗口变量自动调整，并显示 plot 2。这是离差的散点图。

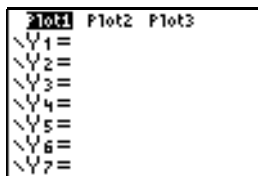


注意离差的方式：一组负离差，接着一组正离差，最后是另一组负离差。

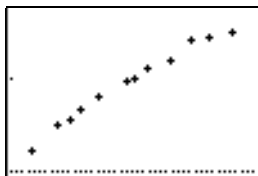
这种离差方式表明，存在一种曲线，它与这组线性模型不适合的数据相关。离差图突出一趋势向下的曲线，因而用趋势向下的曲线模型拟合这些数据可能更精确些。也许平方根函数合适。试着采用幂回归拟合为  $y = a * x^b$  形式的函数。

22. 按 **[Y=]** 显示 Y= 编辑器。

按 **[CLEAR]** 清除 Y1 中的线性回归方程。按 **[↑] [ENTER]** 打开 plot 1。按 **[↓] [ENTER]** 关闭 plot 2。

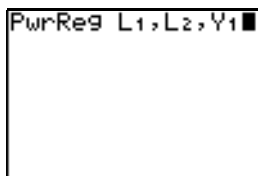


23. 按 **[ZOOM] 9** 选择 ZOOM 菜单中的 **9:ZoomStat**。窗口变量将自动调整，显示出原来时间—长度数据的散射图 (plot 1)。

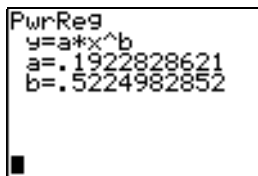


24. 按 **[STAT] [↓] [ALPHA] [A]** 选择 STAT CALC 菜单中的 **A:PwrReg**。PwrReg 粘贴到主屏幕。

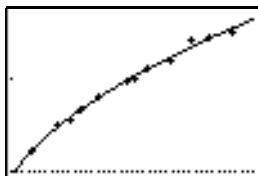
按 **[2nd] [L1] [↓] [2nd] [L2] [↓]**。按 **[VARS] [↓] 1** 显示 VARS Y-VARS FUNCTION 子菜单，然后按 **1** 选择 **1:Y1**。L1、L2 和 Y1 作为 PwrReg 的参数粘贴到主屏幕。



25. 按 **[ENTER]** 计算幂回归。a 和 b 的值显示在主屏幕中。幂回归方程保存到 Y1 中。计算得到的离差并自动保存到数组名 RESID 中。



26. 按 **[GRAPH]**，显示出回归线和散射图。



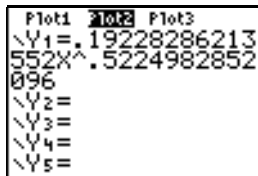
新函数  $y = .192x^{-.522}$  似乎很好地拟合这些数据。要得到更详细的信息，请检查离差图。

27. 按  $\boxed{Y=}$  显示 Y= 编辑器。

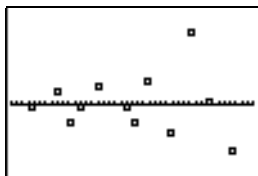
按  $\boxed{\leftarrow}$   $\boxed{ENTER}$  取消选定  $Y_1$ 。

按  $\boxed{\rightarrow}$   $\boxed{ENTER}$  关闭 plot 1。按  $\boxed{\rightarrow}$   $\boxed{ENTER}$  打开 plot 2。

**注意：**步骤 19 定义了 plot 2 以绘制对摆绳长度 (L1) 的离差 (RESID)。



28. 按  $\boxed{ZOOM}$   $\boxed{9}$  选择 ZOOM 菜单中的 **9:ZoomStat**。窗口变量自动调整，并显示 plot 2。这是离差的散点图。



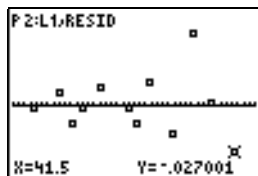
新离差图表明，离差的符号是随机的，其幅值随摆绳长度的增加而增加。

要观察离差的幅值，请继续下列步骤：

29. 按  $\boxed{TRACE}$ 。

按  $\boxed{\rightarrow}$  和  $\boxed{\leftarrow}$  跟踪这些数据，观察每点上 Y 的值。

对于该模型，最大的正离差约为 0.041，最小的负离差约为 -0.027，其他离差的幅值均小于 0.02。



既然得到了一个较好的反映出摆长与其周期相互关系的模型，就可以利用该模型，预测不同摆绳长度时摆锤的周期。要预测摆长为 20cm 和 50cm 的摆锤的周期，请继续下列步骤：

30. 按 **[VARS]** **[▶]** **1** 显示 VARS Y-VARS FUNCTION 子菜单，接着按 **1** 选择 **1:Y1**。**Y1** 粘贴到主屏幕中。

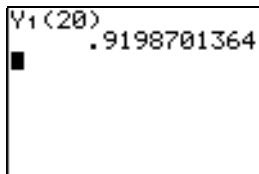


Y1

31. 按 **[□]** **20** **[□]** 输入摆绳长度 20cm。

按 **[ENTER]** 计算预测时间，约为 0.92 秒。

基于离差分析，可以认为 0.92 秒的预测值与实际值的差值在 0.02 秒范围内。



Y1(20)  
.9198701364

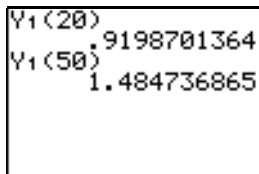
32. 按 **[2nd]** **[ENTRY]** 重新调用“最后输入”。

按 **[◀]** **[◀]** **[◀]** **5** 改变摆绳长度为 50 cm。

33. 按 **[ENTER]** 计算预测时间，约为 1.48 秒。

因为 50cm 的摆绳长度大于这组数据中的长度，并且离差随摆长增加而增加，所以该估计的误差可能较大。

**注意：**也可以用运算表来进行预测，运算表的 TABLE SETUP 设置为 **Indpnt:Ask** 和 **Depend:Auto**（第 7 章）。



Y1(20)  
.9198701364  
Y1(50)  
1.484736865

## 使用数组保存数据

用于统计分析的数据保存在数组中，使用统计数组编辑器可以创建和编辑数组。TI-83 Plus 在内存中有 6 个数组变量 **L1-L6**，可将数据保存到其中以进行统计计算。也可以将数据保存到创建的数组名中（第 11 章）。

## 建立统计分析

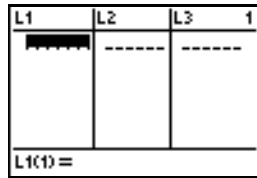
要建立统计分析，请执行以下步骤。详细信息请阅读本章。

1. 将统计数据输入到一个或多个数组中。
2. 绘制这些数据。
3. 计算统计变量或用一种模型拟合这些数据。
4. 绘制数据的回归方程。
5. 绘制给定回归模型的离差数组。

## 显示统计数组编辑器

统计数组编辑器是一个运算表，可以保存、编辑和观察多达 20 个内存中的数组。通过统计数组编辑器还可以创建数组名。

要显示统计数组编辑器，先按 **[STAT]**，然后选择 **STAT EDIT** 菜单中的 **1:Edit**。



顶行显示数组名。L1-L6 在内存重新设置后保存在 1-6 列中。右上角显示当前列号。

底行是输入行，所有输入的数据在此显示。本行的特征随当前环境（页 12-17）而变。

中心区域可显示多达 3 个数组，每个数组多达 7 个元素，必要时将显示缩写数据。输入行显示当前元素的精确值。

## 在统计数组编辑器中输入数组名

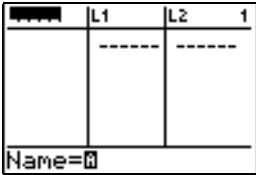
按下列步骤在统计数组编辑器中输入数组名：

1. 通过下述两种方法之一在输入行中显示 **Name=** 提示。

- 将光标移至将插入数组的列数组名上，然后按 **[2nd] [INS]**。显示一个未命名列，其余数组右移一列。
- 按 **[<]** 直到光标位于顶行，然后按 **[>]** 直到到达未命名列。

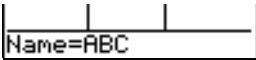
**注意：**如果所有 20 列中都保存有数组名，则必须删除一个数组名，为未命名列保留空间。

在输入行中显示 **Name=** 提示，且字母锁开启。

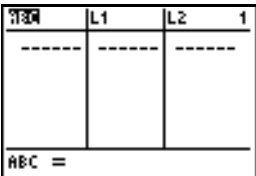


2. 通过下述四种方法之一输入合法的数组名。

- 从 **LIST NAMES** 菜单（第 11 章）中选择一个数组名。
- 从键盘输入 **L1**、**L2**、**L3**、**L4**、**L5** 或 **L6**。
- 直接从键盘输入现有的用户创建数组名。
- 输入用户所创建的新数组名（页 12-12）。



3. 按 **[ENTER]** 或 **[>]**，将数组名和其元素（如果有）保存到统计数组编辑器的当前列中。




要开始输入、滚动或编辑数组元素，按 **[>]**，则显示出矩形条光标。

**注意：**如果在步骤 2 中输入的数组名已保存在统计数组编辑器的另一列中，那么数组和其元素（如果有）从以前列移至当前列，其余列相应地移动。

### 在统计数组编辑器中创建数组名

要在统计数组编辑器中创建数组名，按下列步骤进行：

1. 按照页 12-11 上的步骤 1，显示 **Name=** 提示。
2. 按 [字母 A-Z 或  $\theta$ ] 输入名字的第一个字母。第一个字符不能是数字。
3. 输入 0-4 个字母、 $\theta$  或数字组成新的用户创建数组名。数组名长度为 1-5 个字符。
4. 按 **ENTER** 或 ，将数组名保存在统计数组编辑器的当前列中。数组名成为 LIST NAMES 菜单（第 11 章）中的一项。

### 从统计数组编辑器中删除数组

要从统计数组编辑器中删除数组，先将光标移到数组名上，然后按 **DEL**。该数组不会从内存中删除，只是从统计数组编辑器中删除。

**注意 1：** 要从内存中删除数组名，使用 MEMORY MANAGEMENT/ DELETE 子菜单（第 18 章）。

**注意 2：** 如果归档某个数组，将会从统计数组编辑器中删除该数组。


### 删除所有数组并恢复 L1-L6

可以从统计数组编辑器中删除所有用户创建的数组，并通过下述两种方法之一将数组名 L1-L6 恢复到列 1-6 中。

- 使用不带任何参数的 **SetUpEditor**（页 12-21）。
- 重设全部内存（第 18 章）。

### 清除数组中所有元素

通过下述五种方法之一可以清除数组中所有元素。

- 使用 **ClrList** 清除指定的数组（页 12-20）。
- 在统计数组编辑器中，按  将光标移至数组名上，然后按 **CLEAR** **ENTER**。
- 在统计数组编辑器中，将光标移到每个元素上，然后逐个按 **DEL**。
- 在主屏幕或程序编辑器中，输入 **0→dim(listname)**，将数组名 *listname* 的维数置成 0（第 11 章）。
- 使用 **ClrAllLists** 清除内存中所有的数组（第 18 章）。



## 编辑数组元素

按下列步骤编辑数组元素：

1. 将矩形条光标移到要编辑的元素上。
2. 按 **[ENTER]** 将光标移到输入行中。

**注意：**如果要替换当前值，则可以不按 **[ENTER]** 而输入新值。当输入第一个字符时，当前值自动清除。

3. 在输入行中编辑元素。

- 按一个或多个键输入新的值。当输入第一个字符时，当前值自动清除。
- 按 **[←]** 将光标移到要在之前插入字符的位置，按 **[2nd] [INS]**，然后输入一个或多个字符。
- 按 **[→]** 将光标移到要删除的字符上，按 **[DEL]** 删除该字符。

要取消所做编辑并恢复矩形条光标处原来的元素，按 **[CLEAR] [ENTER]**。

ABC	L1	L2	1
5	-----	-----	
10			
15			
20			
25			
-----			
ABC(3)=25*1000			

**注意：**可以输入表达式和变量作为元素。

4. 按 **[ENTER]**、**[↑]** 或 **[↓]** 更新数组。如果输入表达式，则计算表达式的值；如果只输入变量，则变量的存储值显示为数组元素。

ABC	L1	L2	1
5	-----	-----	
10			
25,000			
20			
25			
-----			
ABC(4)=20			

当在统计数组编辑器中编辑数组元素时，数组在内存中立刻更新。

## 将公式与数组名相连

### 将公式与统计数组编辑器中的数组名相连

可以将公式与统计数组编辑器中的数组名相连，然后显示和编辑计算得到的数组元素。运行时相连的公式必须涉及某一数组。第 11 章详细地描述了将公式与数组名相连这一概念。

要将公式与保存在统计数组编辑器中的数组名相连，请按下列步骤进行：

1. 按 **[STAT]** **[ENTER]** 显示统计数组编辑器。
2. 按 **[↑]** 将光标移到顶行。
3. 必要时，按 **[←]** 或 **[→]** 将光标移到要连接公式的数组名上。

**注意：**如果在输入行中显示带引号的公式，那么已有公式与该数组名相连。要编辑该公式，先按 **[ENTER]**，然后进行编辑。

4. 按 **[ALPHA]** **["]**，输入公式后再按 **[ALPHA]** **["]**。

**注意：**如果没有用引号，TI-83 Plus 计算并显示计算后的同一初始数组，但不为将来的计算连接公式。

ABC	L1	L2	Z
5	-----	-----	
10			
25000			
20			
25			
-----			
L1 = " LABC+10" ■			

**注意：**公式中引用的任何用户创建的数组名之前必须加上符号 **L**（第 11 章）。

5. 按 **[ENTER]**。TI-83 Plus 计算每个数组元素，结果保存到与公式相连的数组名中。一个锁定符号显示在统计数组编辑器中，紧挨着与公式相连的数组名。

锁定符号

ABC	L1	#	L2	Z
5	15		-----	
10	20			
25000	25010			
20	30			
25	35			
-----				
L1(1) = 15				

## 显示公式生成的数组时使用统计数组编辑器

当编辑相连公式中引用的数组的元素时，TI-83 Plus 更新公式连接的数组中相应元素（第 11 章）。

ABC	L1	L2	1
5	15	---	---
10	20	---	---
25000	25010	---	---
20	30	---	---
25	35	---	---
---	---	---	---
ABC(1) = 5			

ABC	L1	L2	1
6	16	---	---
10	20	---	---
25000	25010	---	---
20	30	---	---
25	35	---	---
---	---	---	---
ABC(2) = 10			

当与公式相连的数组显示在统计数组编辑器中时，如果编辑或输入另一显示数组的元素，TI-83 Plus 接收每次编辑或输入所花费的时间将比编辑器中没有显示这种与公式相连的数组时稍长。

**提示：**为缩短编辑时间，可水平滚动数组，直到编辑器中没有显示与公式相连的数组；或者重新布置统计数组编辑器，使得编辑器中没有显示与公式相连的数组。

## 处理相连公式引起的错误

在主屏幕中，可以将公式连接到某个数组，而该公式又引用一个维数为 0 的数组（第 11 章）。然而，由这个公式生成的数组直到此公式引用的数组中至少输入一个元素后才能在统计数组编辑器或主屏幕中显示。

相连公式所引用数组的全部元素对相连的公式必须是有效的。例如，如果设置的是 **Real** 方式，且相连的公式为  $\log(L1)$ ，那么 **L1** 的每个元素必须大于 0，因为负数的对数返回的结果为复数。

**提示：**如果试图在统计数组编辑器中显示公式生成的数组时返回一个错误菜单，则可选择 **2:Goto**，记下与该数组相连的公式，然后按 **[CLEAR]** **[ENTER]** 清除连接。之后，可以使用统计数组编辑器查找错误源。作适当修改后，重新将公式与数组相连。

如果不想清除公式，则可以选择 **1:Quit**，在主屏幕中显示引用的数组，查找并编辑错误源。要在主屏幕中编辑数组的元素，将新值保存在 *listname(element#)* 中（第 11 章）。

### 拆离数组名中的公式

通过下述几种方法，可以拆离数组名中的公式。

例如：

- 在统计数组编辑器中，将光标移到公式连接的数组的名字上，按 **[ENTER]** **[CLEAR]** **[ENTER]**，所有数组元素不变，但公式被拆离，锁定符号消失。
- 在统计数组编辑器中，将光标移到公式连接的数组的元素上，按 **[ENTER]** 编辑该元素，然后再按 **[ENTER]**。该元素发生变化，公式被拆离，同时锁定符号消失。其他所有数组元素不变。
- 使用 **ClrList**（页 12-20）。清除一个或多个所指定数组的所有元素，以及所有和它们连接的公式，这些数组上的锁定符号同时消失。所有数组名不变。
- 使用 **ClrAllLists**（第 18 章）。清除内存中全部数组的所有元素，以及拆离所有数组名中的所有公式，同时所有锁定符号消失。所有数组名不变。

### 编辑由公式生成的数组元素

如上文所述，一种拆离数组名中公式的方法是编辑公式连接的数组元素。

TI-83 Plus 可通过编辑公式生成的数组元素，避免偶然拆离数组名中的公式。

由于这种保护特性，编辑公式生成的数组元素时必须先按 **[ENTER]**。

这种保护特性不允许删除公式连接的数组元素。要删除这种数组元素，必须先用上述方法拆离公式。

## 统计数组编辑器环境

统计数组编辑器有四种环境。

- 观察元素环境
- 编辑元素环境
- 观察数组名环境
- 输入数组名环境

统计数组编辑器首先显示观察元素环境。要在这四种环境之间切换，先选择 STAT EDIT 菜单中的 **1:Edit**，然后按下述步骤进行：

ABC	L1	•	L2	1
5	15		---	
10	20		---	
2.5E7	2.5E7		---	
20	30		---	
25	35		---	
-----	-----		-----	
ABC = {5, 10, 25000...}				

1. 按 **▶** 将光标移到数组名上。这时处于观察数组名的环境。按 **▶** 和 **◀** 观察保存在统计数组编辑器其他列中的数组名。

ABC	L1	•	L2	1
5	15		---	
10	20		---	
2.5E7	2.5E7		---	
20	30		---	
25	35		---	
-----	-----		-----	
ABC = {5, 10, 25000...}				

2. 按 **ENTER**。这时处于编辑元素环境，可以编辑数组中的任何元素。当前数组的所有元素显示在输入行中的大括号 ({} ) 内。按 **▶** 和 **◀** 观察更多的数组元素。

ABC	L1	•	L2	2
5	15		---	
10	20		---	
2.5E7	2.5E7		---	
20	30		---	
25	35		---	
-----	-----		-----	
L1(2)=25000010				

3. 再按 **ENTER**。这时处于观察元素环境。按 **▶**、**◀**、**▼** 和 **▲** 观察其他数组元素。当前元素的精确值显示在输入行中。

ABC	L1	•	L2	2
5	15		---	
10	20		---	
2.5E7	2.5E7		---	
20	30		---	
25	35		---	
-----	-----		-----	
L1(2)=5000010				

4. 再按 **ENTER**。这时处于编辑元素环境。可以在输入行中编辑当前元素。

ABC	L1	•	L2	2
5	15		---	
10	20		---	
2.5E7	2.5E7		---	
20	30		---	
25	35		---	
-----	-----		-----	
Name=0				

5. 按 **▶** 直到光标处于数组名上，然后按 **2nd** [INS]。这时处于输入数组名环境。

ABC	L1	•	L2	2
5	15		---	
10	20		---	
2.5E7	2.5E7		---	
20	30		---	
25	35		---	
-----	-----		-----	
L1 = "LABC+10"				

6. 按 **CLEAR**。这时处于观察数组名环境。

ABC	L1	•	L2	2
5	15		---	
10	20		---	
2.5E7	2.5E7		---	
20	30		---	
25	35		---	
-----	-----		-----	
L1(1)=15				

7. 按 **▼**。这时返回到观察元素环境。

## 观察元素环境

在观察元素环境中，输入行中显示数组名、当前元素在数组中的位置和当前元素的精确值（最多 12 个字符）。省略号 (...) 表示元素值的长度大于 12 个字符。

ABC	L1	#	L2	2
5	15		-----	
10	20			
25E7	25010			
20	30			
25	35			
-----				
ABC(3)=2500010				

按 **[ALPHA]** **▼** 向下翻页 6 个数组元素；按 **[ALPHA]** **▲** 向上翻页 6 个元素；要删除数组元素，按 **[DEL]**，下面的元素上移一行。要插入新的元素，按 **[2nd]** **[INS]**，其默认值为 **0**。

## 编辑元素环境

在编辑元素环境中，输入行中显示的数据取决于上一次环境。

- 从观察元素环境切换到编辑元素环境时，显示的是当前元素的精确值。可以编辑该元素的值。按 **▼** 和 **▲** 编辑其他的数组元素。

ABC	L1	#	L2	1
5	15		-----	
10	20			
25000	25010			
20	30			
25	35			
-----				
ABC(3)=25000				

→

ABC	L1	#	L2	1
5	15		-----	
10	20			
25000	25010			
20	30			
25	35			
-----				
ABC(3)=5000				

- 从观察数组名环境切换到编辑元素环境时，显示的是数组中所有元素的精确值。省略号表示数组元素超出了屏幕。可以按 **▼** 和 **▲** 编辑数组中的任何元素。

ABC	L1	#	L2	1
5	15		-----	
10	20			
25000	25010			
20	30			
25	35			
-----				
ABC = (5, 10, 25000...				

→

ABC	L1	#	L2	1
5	15		-----	
10	20			
25000	25010			
20	30			
25	35			
-----				
ABC = (5, 10, 25000...				

**注意：**在编辑元素环境中，仅当它是从观察数组名环境切换过来的，才能将公式连接到数组名。

## 观察数组名环境

在观察数组名环境中，输入行中显示数组名和数组元素。

统计器	L1	#	L2	1
5	15		-----	
10	20			
25000	25010			
20	30			
25	35			
-----	-----			
ABC = {5, 10, 25000...				

要从统计数组编辑器中删除数组，按 **[DEL]**。其余数组左移一列。该数组没有从内存中删除。

要在当前列中插入数组名，按 **[2nd] [INS]**。其余列右移一列。

## 输入数组名环境

在输入数组名环境中，输入行中显示 **Name=** 提示，且字母锁开启。

在 **Name=** 提示处，可以创建新的数组名，或者从键盘粘贴 **L1-L6** 中的数组名，或者粘贴 **LIST NAMES** 菜单中现有的数组名（第 11 章）。在 **Name=** 提示处不需要符号 **L**。

统计器	ABC	L1	#	1
	5	15		
	10	20		
	25000	25010		
	20	30		
	25	35		
	-----	-----		
Name=				

要退出输入数组名环境而不输数组名，按 **[CLEAR]**。统计数组编辑器切换到观察数组名环境。

## STAT EDIT 菜单

要显示 STAT EDIT 菜单，按 **[STAT]**。

### EDIT CALC TESTS

- |                |                |
|----------------|----------------|
| 1: Edit...     | 显示统计数组编辑器。     |
| 2: SortA(      | 按升序排序数组。       |
| 3: SortD(      | 按降序排序数组。       |
| 4: ClrList     | 删除数组的所有元素。     |
| 5: SetUpEditor | 设置统计数组编辑器中的数组。 |

**注意：**第 13 章：推理统计描述了 STAT TESTS 菜单项。

## SortA(、SortD(

**SortA(**（按升序排序）将数组元素按照从小到大的顺序排序。**SortD(**（按降序排序）将数组元素按照从大到小的顺序排序。复数数组按模来排序。

**SortA(** 和 **SortD(** 都可以通过下述两种方法之一进行排序。

- 对于数组名 *listname*，**SortA(** 和 **SortD(** 排序数组名 *listname* 中的元素，并更新内存中的数组。
- 对于两个或更多的数组，**SortA(** 和 **SortD(** 按 *keylistname* 排序，然后将每个 *dependlist* 的元素排序，顺序与 *keylistname* 中相应元素的顺序相同。这样，可以排序 **X** 中的二元数据，而数据对的对应关系不变。所有数组必须有相同的长度。

排序后的数组在内存中更新。

**SortA(listname)**

**SortD(listname)**

**SortA(keylistname,dependlist1[,dependlist2,...,dependlist n])**

**SortD(keylistname,dependlist1[,dependlist2,...,dependlist n])**

<pre>(5, 4, 3)→L<sub>3</sub> (5 4 3) (1, 2, 3)→L<sub>4</sub> (1 2 3) SortA(L<sub>3</sub>,L<sub>4</sub>) Done</pre>	<pre>L<sub>3</sub> L<sub>4</sub> (3 4 5) (3 2 1)</pre>
--	--

**注意：****SortA(** 和 **SortD(** 与 LIST OPS 菜单上的 **SortA(** 和 **SortD(** 相同。

## ClrList

**ClrList** 从内存中清除一个或多个数组名 *listnames* 的元素。**ClrList** 也清除与数组名 *listnames* 连接的公式。

**ClrList listname1,listname2,...,listname n**

**注意：**要从内存中清除所有数组名中的所有元素，使用 **ClrAllLists**（第 18 章）。



## SetUpEditor

通过 **SetUpEditor**，可以设置统计数组编辑器，按指定的次序显示一个或多个数组名 *listname*。可以指定 0-20 个数组名 *listname*。

另外，如果想使用恰巧已归档的 *listname*，那么 **SetUp Editor** 会自动取消归档该 *listname*，同时将其放在统计数组编辑器中。

### SetUpEditor [*listname1,listname2,...,listname n*]

**SetUpEditor** 对于 1-20 个数组名，先从统计数组编辑器中删除所有数组名，然后按指定的次序，从列 1 开始，将数组名 *listname* 保存到统计数组编辑器的列中。

```

SetUpEditor RESI
D,L3,L6,TIME,LO
G,A123
Done
    
```

RESID	L3	L6	# 1
.00112	1	11	
.00692	2	12	
-.0104	3	13	
-.0015	4	14	
.0094	5	15	
-.0018	6	16	
-.0106	-----		
RESID(0) = -.0013125...			

TIME	LONG	A123	# 4
60	56	5	
120	82	10	
30	74	15	
180	52	20	
-----	36	25	
	98	30	
	74	-----	
TIME(0) = 60			

如果输入的数组名 *listname* 还没有在内存中保存，则创建该数组名 *listname*，并保存到内存中。它成为 LIST NAMES 菜单中的一项。

## 恢复 L1-L6 到统计数组编辑器中

**SetUpEditor** 没有数组名时，先从统计数组编辑器中删除所有数组名，然后将数组名 L1-L6 恢复到统计数组编辑器的 1-6 列中。

```

SetUpEditor
Done
    
```

L1	L2	L3	# 1
6.5	.51	1	
11	.68	2	
12.2	.72	3	
15	.88	4	
18	.88	5	
23.1	.98	6	
24.4	1.01	-----	
L1(0) = 6.5			

L4	L5	L6	# 4
	-----	11	
		12	
		13	
		14	
		15	
		16	
		-----	
L4(0) =			

## 回归模型功能

STAT CALC 菜单的项 **3** 至 **C** 是回归模型（页 12-24）。自动离差数组和自动回归方程功能适用于所有的回归模型。分析显示方式适用于某些回归模型。

### 自动离差数组

执行回归模型时，自动离差数组功能计算离差，并将之保存到数组名 **RESID** 中。**RESID** 成为 LIST NAMES 菜单中的一项（第 11 章）。

```
LIST NAMES OPS MATH
1:ABC
2:RESID
```

TI-83 Plus 使用下面的公式计算 **RESID** 数组元素。下一节介绍变量 **RegEQ**。

$$\mathbf{RESID} = Ylistname - \mathbf{RegEQ}(Xlistname)$$

### 自动回归方程

每个回归模型都有一个可选的参数 *regequ*，可以为该参数指定一个 Y= 变量，如 **Y1**。执行时，回归方程自动保存到指定的 Y= 变量，并选择了 Y= 函数。

```
{1,2,3}→L1: {-1, -
2, -5}→L2
(-1 -2 -5)
LinReg(ax+b) L1,
L2, Y3
```

```
LinReg
y=ax+b
a=-2
b=1.333333333
```

```
Plot1 Plot2 Plot3
\Y1=
\Y2=
\Y3=-2X+1.333333
3333333
```

不管是否为 *regequ* 指定一个 Y= 变量，回归方程总是保存到 TI-83 Plus 变量 **RegEQ** 中，该变量是 VARS Statistics EQ 子菜单中的项 **1**。

```
XV Σ EQ TEST PTS
1:RegEQ
2:a
3:b
```

**注意：**对于回归方程，可以使用定点十进制小数位设置值以控制小数点后保存的位数（第 1 章）。不过，将位数限得较小可能影响拟合的精度。

## 分析显示方式

当执行某些回归模型时，TI-83 Plus 计算并保存分析值  $r$ （相关系数）和  $r^2$ （可靠系数）或  $R^2$ （可靠系数）。

对于下述回归模型计算并保存  $r$  和  $r^2$ 。

<b>LinReg(ax+b)</b>	<b>LnReg</b>	<b>PwrReg</b>
<b>LinReg(a+bx)</b>	<b>ExpReg</b>	

对于下述回归模型计算并保存  $R^2$ 。

<b>QuadReg</b>	<b>CubicReg</b>	<b>QuartReg</b>
----------------	-----------------	-----------------

对于 **LnReg**、**ExpReg** 和 **PwrReg**，它们的  $r$  和  $r^2$  是基于线性变换的数据进行计算得到的。例如，对于 **ExpReg** ( $y=ab^x$ )， $r$  和  $r^2$  是基于  $\ln y = \ln a + x(\ln b)$  进行计算得到的。

默认情况下，执行回归模型时，这些值不与模型的结果一起显示出。不过，可以通过执行指令 **DiagnosticOn** 或 **DiagnosticOff** 来设置分析显示方式。这两个指令在 CATALOG 中（第 15 章）。

```
CATALOG
det(
DiagnosticOff
DiagnosticOn
dim(
```

**注意：**要从主屏幕中设置 **DiagnosticOn** 或 **DiagnosticOff**，先按 **2nd** [CATALOG]，然后选择所要方式的指令。选择的指令粘贴到主屏幕中。按 **ENTER** 设置方式。

设置 **DiagnosticOn** 后，如果执行回归模型，分析值将与回归模型的结果一起显示出来。

<pre>DiagnosticOn Done LinReg(ax+b) L1, L2</pre>	<pre>LinReg y=ax+b a=-2 b=1.333333333 r^2=.9230769231 r=-.9607689228</pre>
--	--

设置 **DiagnosticOff** 后，如果执行回归模型，分析值将不会与回归模型的结果一起显示出来。

<pre>DiagnosticOff Done LinReg(ax+b) L1, L2</pre>	<pre>LinReg y=ax+b a=-2 b=1.333333333</pre>
---	---

## STAT CALC 菜单

要显示 STAT CALC 菜单，按 **[STAT]** **[▶]**。

EDIT	CALC	TESTS
1:	1-Var Stats	计算一元统计。
2:	2-Var Stats	计算二元统计。
3:	Med-Med	计算中位数—中位数线。
4:	LinReg(ax+b)	用线性模型拟合数据。
5:	QuadReg	用二次模型拟合数据。
6:	CubicReg	用三次模型拟合数据。
7:	QuartReg	用四次模型拟合数据。
8:	LinReg(a+bx)	用线性模型拟合数据。
9:	LnReg	用对数模型拟合数据。
0:	ExpReg	用指数模型拟合数据。
A:	PwrReg	用幂模型拟合数据。
B:	Logistic	用逻辑模型拟合数据。
C:	SinReg	用正弦模型拟合数据。

对于 STAT CALC 菜单的每一项，如果 *Xlistname* 和 *Ylistname* 都没有指定，那么默认的数组名为 **L1** 和 **L2**。如果没有指定 *freqlist*，那么默认值是每个数组元素出现 **1** 次。

### 数据点出现的频率

对于 STAT CALC 菜单的大部分菜单项，可以指定一个表示数据出现次数或频数的数组 (*freqlist*)。

*freqlist* 中的每个元素表示对应的数据点或数据对在所分析的数据集中出现的次数。

例如，如果 **L1={15,12,9,14}** 和 **LFREQ={1,4,1,3}**，那么 TI-83 Plus 将指令 **1-Var Stats L1** 和 **LFREQ** 解释为：**15** 出现 1 次；**12** 出现 4 次；**9** 出现 1 次；**14** 出现 3 次。

*freqlist* 中的每个元素必须  $\geq 0$ ，并且至少有一个元素  $> 0$ 。

非整数的 *freqlist* 元素是有效的。这在输入频数表示为百分数或相加为 **1** 的部分时是有用的。然而，如果 *freqlist* 包含非整数的频数，那么 **Sx** 和 **Sy** 不会定义，在统计结果中不会显示 **Sx** 和 **Sy** 的值。

## 1-Var Stats

**1-Var Stats**（一元统计）对所测单变量进行数据分析。*freqlist* 中的每个元素表示 *Xlistname* 中对应数据点的出现频数。*freqlist* 元素必须是大于 0 的实数。

**1-Var Stats** [*Xlistname*,*freqlist*]

```
1-Var Stats L1,L
z
```

## 2-Var Stats

**2-Var Stats**（二元统计）分析成对的数据。*Xlistname* 是自变量；*Ylistname* 是因变量。*freqlist* 中的每个元素表示每个对应数据对 (*Xlistname*, *Ylistname*) 的出现频数。

**2-Var Stats** [*Xlistname*,*Ylistname*,*freqlist*]

## Med-Med (ax+b)

**Med-Med**（中位数—中位数）通过采用中位数—中位数线（阻力线）技术用模型  $y=ax+b$  拟合数据，计算概要点  $x_1$ ， $y_1$ ， $x_2$ ， $y_2$  和  $x_3$ ， $y_3$ 。

**Med-Med** 显示 **a**（斜率）和 **b**（*y*-交点）的值。

**Med-Med** [*Xlistname*,*Ylistname*,*freqlist*,*regequ*]

```
Med-Med L3,L4,Y2

```

```
Med-Med
y=ax+b
a=.875
b=1.541666667
```

## LinReg (ax+b)

**LinReg(ax+b)**（线性回归）采用最小二乘拟合法用模型  $y=ax+b$  拟合数据。它显示 **a**（斜率）和 **b**（*y*-交点）的值；如果 **DiagnosticOn** 已设置，它也显示  $r^2$  和  $r$  的值。

**LinReg(ax+b)** [*Xlistname*,*Ylistname*,*freqlist*,*regequ*]

## QuadReg (ax<sup>2</sup>+bx+c)

**QuadReg**（二次回归）用二次多项式  $y=ax^2+bx+c$  拟合数据。它显示 **a**、**b** 和 **c** 的值；如果 **DiagnosticOn** 已设置，它也显示  $R^2$  的值。对于三个数据点，该方程是多项式拟合；对于四个以上的数据点，则是多项式回归。要求至少三个数据点。

**QuadReg** [*Xlistname*,*Ylistname*,*freqlist*,*regequ*]

**CubicReg— $(ax^3+bx^2+cx+d)$** 

**CubicReg** (三次回归) 用三次多项式  $y=ax^3+bx^2+cx+d$  拟合数据。它显示 **a**、**b**、**c** 和 **d** 的值；如果 **DiagnosticOn** 已设置，它也显示 **R<sup>2</sup>** 的值。对于四个数据点，该方程是多项式拟合；对于五个以上的数据点，则是多项式回归。要求至少四个数据点。

**CubicReg** [*Xlistname*,*Ylistname*,*freqlist*,*regequ*]

**QuartReg— $(ax^4+bx^3+cx^2+dx+e)$** 

**QuartReg** (四次回归) 用四次多项式  $y=ax^4+bx^3+cx^2+dx+e$  拟合数据。它显示 **a**、**b**、**c**、**d** 和 **e** 的值；如果 **DiagnosticOn** 已设置，它也显示 **R<sup>2</sup>** 的值。对于五个数据点，该方程是多项式拟合；对于六个以上的数据点，则是多项式回归。要求至少五个数据点。

**QuartReg** [*Xlistname*,*Ylistname*,*freqlist*,*regequ*]

**LinReg— $(a+bx)$** 

**LinReg(a+bx)** (线性回归) 采用最小二乘拟合法用模型  $y=a+bx$  拟合数据。它显示 **a** (**y**-交点) 和 **b** (斜率) 的值；如果 **DiagnosticOn** 已设置，它也显示 **r<sup>2</sup>** 和 **r** 的值。

**LinReg(a+bx)** [*Xlistname*,*Ylistname*,*freqlist*,*regequ*]

**LnReg— $(a+b \ln(x))$** 

**LnReg** (对数回归) 采用最小二乘法拟合和变换值  $\ln(x)$  与  $y$ ，用模型  $y=a+b \ln(x)$  拟合数据。它显示 **a** 和 **b** 的值；如果 **DiagnosticOn** 已设置，它也显示 **r<sup>2</sup>** 和 **r** 的值。

**LnReg** [*Xlistname*,*Ylistname*,*freqlist*,*regequ*]

**ExpReg— $(ab^x)$** 

**ExpReg** (指数回归) 采用最小二乘法拟合和变换值  $x$  与  $\ln(y)$ ，用模型  $y=ab^x$  拟合数据。它显示 **a** 和 **b** 的值；如果 **DiagnosticOn** 已设置，它也显示 **r<sup>2</sup>** 和 **r** 的值。

**ExpReg** [*Xlistname*,*Ylistname*,*freqlist*,*regequ*]

---

## PwrReg— $(ax^b)$

**PwrReg**（幂回归）采用最小二乘法拟合和变换值  $\ln(x)$  与  $\ln(y)$ ，用模型  $y=ax^b$  拟合数据。它显示 **a** 和 **b** 的值；如果 **DiagnosticOn** 已设置，它也显示 **r<sup>2</sup>** 和 **r** 的值。

**PwrReg** [*Xlistname*,*Ylistname*,*freqlist*,*regequ*]

## Logistic— $c/(1+a*e^{-bx})$

**Logistic** 采用迭代最小二乘法拟合，用模型  $y=c/(1+a*e^{-bx})$  拟合数据。它显示 **a**、**b** 和 **c** 的值。

**Logistic** [*Xlistname*,*Ylistname*,*freqlist*,*regequ*]

## SinReg— $a \sin(bx+c)+d$

**SinReg**（正弦回归）采用迭代最小二乘法拟合，用模型  $y=a \sin(bx+c)+d$  拟合数据。它显示 **a**、**b**、**c** 和 **d** 的值，要求至少四个数据点。要求每个周期中至少取两个数据点，以避免出现同样的频率估计值。

**SinReg** [*iterations*,*Xlistname*,*Ylistname*,*period*,*regequ*]

*iterations* 是算法求解时迭代的最大次数。*iterations* 的值可以是  $\geq 1$  且  $\leq 16$  之间的整数；若没有指定，其默认值是 3。算法可能在还没有到达最大的迭代次数 *iterations* 时就求得解。一般地，*iterations* 的值较大时，执行的次数将会增加，**SinReg** 的精度也会变高；反之亦然。

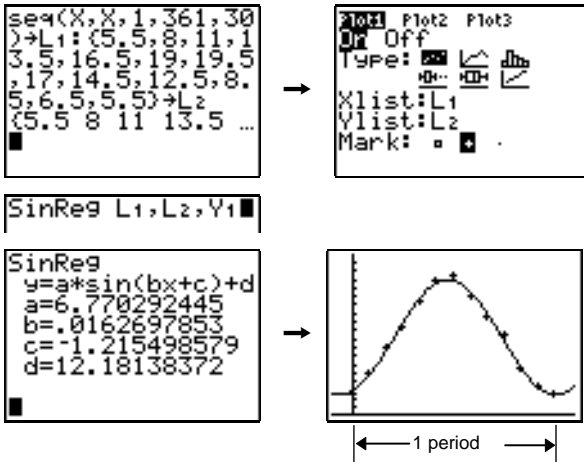
*period*（周期）的估计是可选项。如果没有指定 *period*，那么 *Xlistname* 中时间值的差必须相等，并且时间值必须按升序排列；如果指定 *period*，算法可能很快求得解，或者如果忽略了 *period* 的值，算法可能不該求得解时求得解。如果指定 *period*，那么 *Xlistname* 中时间值的差可以不等。

**注意：****SinReg** 的输出与 **Radian/Degree** 方式设置无关，总是以弧度表示。

下一页中列举了一个 **SinReg** 示例。

SinReg 示例：阿拉斯加一年中白昼的小时数

计算阿拉斯加一年中白昼的小时数的回归模型。



对于杂乱的数据，当指定 *period* 的精确估计时将得到较好的收敛结果。可以通过下述两种方法之一得到 *period* 的估计。

- 对数据绘图，然后跟踪确定一个完整周期的起点和终点之间的 *x* 距离。右上方的图例画出了一个完整的周期。
- 对数据绘图，然后跟踪确定 *N* 个完整周期的起点和终点之间的 *x* 距离。然后用总距离除以 *N*。

在首次尝试使用 **SinReg** 和 *iterations* 的默认值拟合数据后，可能发现拟合结果几乎是正确的，但不是最优的。要得到最优的拟合结果，执行 **SinReg 16,Xlistname,Ylistname,2π / b**，其中 *b* 是上一次执行 **SinReg** 后得到的值。



## 统计变量

统计变量按下文所述的方式进行计算和保存。要选取这些变量用于表达式中，按 **[VARS]**，并选择 **5:Statistics**，然后选择 **VARS** 菜单（表示在下表中标题为 **VARS** 菜单的列中）。如果编辑数组或者改变分析类型，则清除所有统计变量。

变量	1-Var Stats	2-Var Stats	其他	VARS 菜单
<b>x</b> 的平均值	$\bar{x}$	$\bar{x}$		XY
<b>x</b> 的总和	$\Sigma x$	$\Sigma x$		$\Sigma$
<b>x</b> <sup>2</sup> 的总和	$\Sigma x^2$	$\Sigma x^2$		$\Sigma$
<b>x</b> 的样本标准差	<b>Sx</b>	<b>Sx</b>		XY
<b>x</b> 的总体标准差	$\sigma x$	$\sigma x$		XY
数据点数	<b>n</b>	<b>n</b>		XY
<b>y</b> 的平均值		$\bar{y}$		XY
<b>y</b> 的总和		$\Sigma y$		$\Sigma$
<b>y</b> <sup>2</sup> 的总和		$\Sigma y^2$		$\Sigma$
<b>y</b> 的样本标准差		<b>Sy</b>		XY
<b>y</b> 的总体标准差		$\sigma y$		XY
<b>x * y</b> 的总和		$\Sigma xy$		$\Sigma$
<b>x</b> 的最小值	<b>minX</b>	<b>minX</b>		XY
<b>x</b> 的最大值	<b>maxX</b>	<b>maxX</b>		XY
<b>y</b> 的最小值		<b>minY</b>		XY
<b>y</b> 的最大值		<b>maxY</b>		XY
第一个四分位数	<b>Q1</b>			PTS
中位值	<b>Med</b>			PTS
第三个四分位数	<b>Q3</b>			PTS
回归/拟合系数			<b>a, b</b>	EQ
多项式、 <b>Logistic</b> 和 <b>SinReg</b> 系数			<b>a, b, c, d, e</b>	EQ
相关系数			<b>r</b>	EQ
可靠系数			<b>r<sup>2</sup>, R<sup>2</sup></b>	EQ
回归方程			<b>RegEQ</b>	EQ
概要点（仅对 <b>Med-Med</b> ）			<b>x1, y1, x2, y2, x3, y3</b>	PTS

### Q1 和 Q3

第一个四分位数 (**Q1**) 是在 **minX** 和 **Med**（中位值）之间的点的的中位值。第三个四分位数 (**Q3**) 是在 **Med** 和 **maxX** 之间的点的的中位值。

### 输入统计数据

可以从程序中输入统计数据、计算统计结果，并用模型拟合数据。可以在程序中直接将统计数据输入到数组中（第 11 章）。

```
PROGRAM:STATS
:(1,2,3)→L1
:(-1,-2,-5)→L2
```

### 统计计算

要从程序中完成统计计算，请按下列步骤进行：

1. 在程序编辑器的空行上，从 **STAT CALC** 菜单中选择计算的类型。
2. 输入计算中要用到的数组名。数组名用逗号分开。
3. 如果要将回归方程保存到  $Y=$  变量中，则输入一个逗号，然后输入  $Y=$  变量的名称。

```
PROGRAM:STATS
:(1,2,3)→L1
:(-1,-2,-5)→L2
:LinReg(ax+b) L1
,L2,Y2
:■
```

## 绘制数组中统计数据的图形的步骤

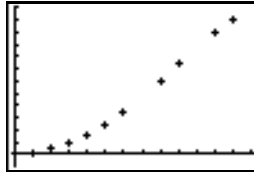
可以绘制保存在数组中的统计数据的图形。可用的 6 种图形类型是：散射图、xy 直线图、直方图、改进式框图、标准框图和正态概率曲线。最多可定义 3 种图形。

要绘制数组中的统计数据的图形，请按下列步骤进行：

1. 将统计数据保存在一个或多个数组中。
2. 适当地选择或取消选择 Y= 函数。
3. 定义统计图。
4. 打开要显示的图形。
5. 定义观察窗口。
6. 显示并研究曲线图。

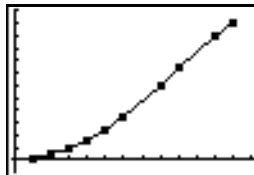
### Scatter (Scatter)

**Scatter** 图成对地绘制 **Xlist** 和 **Ylist** 中的数据点，每个点用框 (□)、十字线 (+) 或点 (•) 表示。**Xlist** 和 **Ylist** 的长度必须相同。**Xlist** 和 **Ylist** 可以用同一个数组。



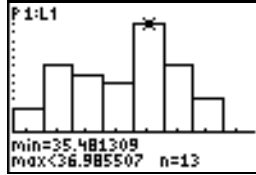
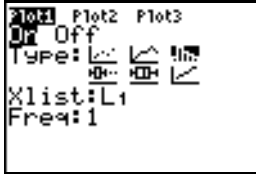
### xyLine (xyLine)

**xyLine** 是一种散射图，图中绘出每个数据点，并按数据点在 **Xlist** 和 **Ylist** 中出现的顺序将它们连接起来。可能要在绘制前使用 **SortA** (或 **SortD**) 将数组排序 (页 12-20)。



 (Histogram)

**Histogram** 绘制一元数据。窗口变量 **Xscl** 的值决定了每根条的宽度，条从 **Xmin** 处开始。**ZoomStat** 调整 **Xmin**、**Xmax**、**Ymin** 和 **Ymax** 以包括所有的数值，同时也调整 **Xscl**。不等式  $(Xmax - Xmin) / Xscl \leq 47$  必须成立。出现在条边上的值算入右侧的条中。

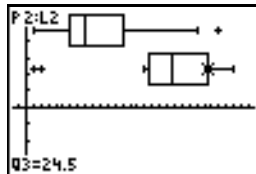


 (ModBoxplot)

**ModBoxplot** (改进式框图) 象标准框图一样绘制一元数据，但不包括超出四分位数  $1.5 * \text{四分位数间距}$  的点 (四分位数间距定义为第三个四分位数 **Q3** 与第一个四分位数 **Q1** 的差)。这些点使用所选的 **Mark** (**□**、**+** 或 **•**) 在点触线后个别地绘制。可以跟踪这些称作局外点的点。

除了当局外点是最大的点 (**maxX**) 或最小的点 (**minX**) 时，局外点的提示为 **x=**。如果局外点存在，那么每个点触线的末端显示 **x=**；否则显示 **minX** 和 **maxX**。框由 **Q1**、**Med** (中位值) 和 **Q3** 定义 (页 12-29)。

框图是根据 **Xmin** 和 **Xmax** 绘制的，但忽略了 **Ymin** 和 **Ymax**。当绘制了两个框图时，第一个处于屏幕的顶部；而第二个处于屏幕的中央。当绘制了三个框图时，第一个处于屏幕的顶部，第二个处于屏幕的中央，第三个处于屏幕的底部。

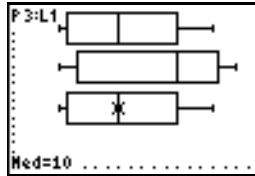


## (Boxplot)

**Boxplot**（标准框图）绘制一元数据。图中的点触线从这组数据中的最小数据点（ $\min X$ ）延伸到第一个四分位数（ $Q_1$ ），从第三个四分位数（ $Q_3$ ）延伸到最大点（ $\max X$ ）。框由  $Q_1$ 、**Med**（中位值）和  $Q_3$  定义（页 12-29）。

框图是根据  $X_{\min}$  和  $X_{\max}$  绘制的，但忽略了  $Y_{\min}$  和  $Y_{\max}$ 。当绘制了两个框图时，第一个处于屏幕的顶部；而第二个处于屏幕的中央。当绘制了三个框图时，第一个处于屏幕的顶部，第二个处于屏幕的中央，第三个处于屏幕的底部。

```
STAT PLOTS
1:Plot1...On
  [ ] L1 1
2:Plot2...On
  [ ] L2 1
3:Plot3...Off
  [ ] L3 1
4↓PlotsOff
```



## (NormProbPlot)

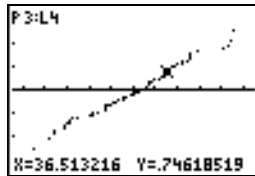
**NormProbPlot**（正态概率曲线）绘制 **Data List** 中每次的观测值  $X$  对相应的标准正态分布的四分位数  $z$  的曲线。如果绘出的点靠近一条直线，那么该曲线表示数据是正态的。

在 **Data List** 域中输入有效的数组名。选择 **X** 或 **Y** 作为 **Data Axis** 设置。

- 如果选择 **X**，TI-83 Plus 将在  $x$  轴上绘制数据，在  $y$  轴上绘制  $z$  值。
- 如果选择 **Y**，TI-83 Plus 将在  $y$  轴上绘制数据，在  $x$  轴上绘制  $z$  值。

```
randNorm(35,2.90
)→L4
{35.11436075 36...
```

```
Plot1 Plot2 Plot3
[ ] Off
Type: [ ] [ ] [ ]
      [ ] [ ] [ ]
Data List:L4
Data Axis:Y
Mark: [ ] [ ]
```



定义图形

要定义图形，请按下列步骤进行：

1. 按 **2nd** [STAT PLOT]。显示出 STAT PLOTS 菜单，连同当前绘图定义。



2. 选择要使用的图形。统计图编辑器显示出来，用于所选择的图形。



3. 如果要立即绘制统计数据，则按 **ENTER** 选择 **On**。不管选择 **On** 还是 **Off**，都将保存该定义。
4. 选择图形类型。每种类型都有下表中列出的可选项提示。

图形类型	XList	YList	Mark	Freq	数据数组	数据轴
Scatter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
xyLine	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Histogram	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ModBoxplot	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Boxplot	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NormProbPlot	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

5. 输入数组名或者选择图形类型的可选项。
  - **Xlist** (包含自变量的数组名)
  - **Ylist** (包含因变量的数组名)
  - **Mark** (□、+ 或 •)
  - **Freq** (Xlist 元素的频数数组，默认值为 1)
  - **Data List** (NormProbPlot 的数组名)
  - **Data Axis** (绘制 Data List 的数轴)

## 显示其他统计图编辑器

每个统计图都有唯一的统计图编辑器。当前统计图（**Plot1**、**Plot2** 或 **Plot3**）的名称亮显在统计图编辑器的顶行中。要显示不同图形的编辑器，按  $\leftarrow$ 、 $\rightarrow$  和  $\uparrow$  将光标移到顶行中的图形名上，然后按 **[ENTER]**。这时显示出所选图形的编辑器，所选择的图形名仍呈亮显状态。

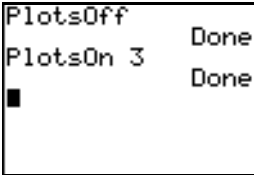


## 打开和关闭统计图

**PlotsOn** 和 **PlotsOff** 允许从主屏幕或程序中打开或关闭统计图。没有给定图形号时，**PlotsOn** 打开所有图形，**PlotsOff** 关闭所有图形；给定一个或多个图形号（**1**、**2** 和 **3**）时，**PlotsOn** 打开指定的图形，而 **PlotsOff** 关闭指定的图形。

**PlotsOff [1,2,3]**

**PlotsOn [1,2,3]**



**注意：**也可以打开和关闭 Y= 编辑器顶行中的统计图（第 3 章）。

### 定义观察窗口

统计图显示在当前图象中。要定义观察窗口，按 **WINDOW** ，然后输入窗口变量的值。**ZoomStat** 重新定义观察窗口以显示所有的统计数据点。

### 跟踪统计图

跟踪散射图或  $xy$  直线图时，跟踪从数组中的第一个元素开始。

跟踪直方图时，光标从第一列的顶部中心开始移到下一列的顶部中心。

跟踪框图时，跟踪从 **Med**（中位值）处开始。按 **▢** 跟踪到 **Q1** 和 **minX**；按 **▣** 跟踪到 **Q3** 和 **maxX**。

当按 **▢** 或 **▣** 移到另一幅图形或另一个  $Y=$  函数时，跟踪移到图中的当前点或起始点（不是最近的象素）。

**ExprOn/ExprOff** 格式设置适用于统计图（第 3 章）。当选择 **ExprOn** 时，图号和绘制的数据数组显示在左上角中。



## 在程序中定义统计图

要从程序中显示统计图，先定义图形，然后再显示图象。

要从程序中定义统计图，先在程序编辑器中的空行上将数据输入到一个或多个数组中，然后执行下列步骤：

1. 按 **2nd** [STAT PLOT] 显示 STAT PLOTS 菜单。

```
PLOTS TYPE MARK
1:Plot1(
2:Plot2(
3:Plot3(
4:PlotsOff
5:PlotsOn
```

2. 选择要定义的图形，将把 **Plot1(**、**Plot2(** 或 **Plot3(** 粘贴到光标处。

```
PROGRAM:PLOT
:(1,2,3,4)→L1
:(5,6,7,8)→L2
:Plot2(█
```

3. 按 **2nd** [STAT PLOT] **▸** 显示 STAT TYPE 菜单。

```
PLOTS TYPE MARK
1:Scatter
2:xyLine
3:Histogram
4:ModBoxPlot
5:BoxPlot
6:NormProbPlot
```

4. 选择图形的类型，将把图形类型的名称粘贴到光标处。

```
PROGRAM:PLOT
:(1,2,3,4)→L1
:(5,6,7,8)→L2
:Plot2(Scatter█
```

在程序中定义统计图 (续)

5. 按  $\square$ 。输入数组名，名字之间用逗号隔开。
6. 按  $\square$   $\square$   $\square$  [STAT PLOT]  $\square$  显示 STAT PLOT MARK 菜单。（如果步骤 4 中选择的是 **3:Histogram** 或 **5:Boxplot**，就没有必要这么做。）

```

PLOTS TYPE MARK
1: □
2: +
3: .
    
```

选择数据点的标记类型 (□、+ 或·)。所选的标记符号粘贴到光标处。

7. 按  $\square$  [ENTER] 结束命令行。

```

PROGRAM:PLOT
:(1,2,3,4)→L1
:(5,6,7,8)→L2
:Plot2(Scatter,L
1,L2,□)
:■
    
```

从程序中显示统计图

要从程序中显示图形，使用 **DispGraph** 指令（第 16 章）或者任何 **ZOOM** 指令（第 3 章）。

```

PROGRAM:PLOT
:(1,2,3,4)→L1
:(5,6,7,8)→L2
:Plot2(Scatter,L
1,L2,□)
:DispGraph
:■
    
```

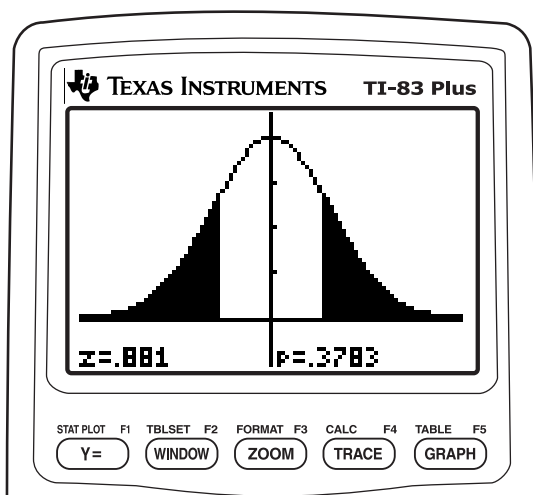
```

PROGRAM:PLOT
:(1,2,3,4)→L1
:(5,6,7,8)→L2
:Plot2(Scatter,L
1,L2,□)
:ZoomStat
:■
    
```

# 13 推理统计 与分布

## 目录

入门：总体平均身高.....	2
推理统计编辑器.....	6
STAT TESTS 菜单.....	9
推理统计输入描述.....	26
检验和区间输出变量.....	28
分布函数.....	29
遮蔽分布.....	35



# 入门：总体平均身高

入门是对本章内容的快速介绍。详细信息请阅读本章。

假设给定下列女性总体的随机样本，要估计其平均身高。因为一个生物总体的身高趋向于正态分布，因此在估计平均值时可使用一个  $t$  分布置信区间。下面的 10 个身高值是 90 个值中的前 10 个值，这些值是从总体估计平均值为 165.1 厘米，标准差为 6.35 厘米（`randNorm(165.1,6.35,90)`，起始值是 789）的一个正态分布总体中随机产生的。

## 10 个女性的身高（厘米）

169.43 168.33 159.55 169.97 159.79 181.42 171.17 162.04 167.15 159.53

1. 按 `[STAT]` `[ENTER]` 显示统计数组编辑器。

按 `[ $\square$ ]` 将光标移到 **L1** 上，然后按 `[2nd]` `[INS]`。底行显示 **Name=** 提示。`[ $\square$ ]` 光标指示字母锁已开启。已有数组名的列右移。

**注意：**统计编辑器可能不象右图显示那样，它依赖于已存储的数组。

	L1	L2	1
	-----	-----	
Name=L1			

2. 在 **Name=** 提示下输入 `[H]` `[G]` `[H]` `[T]`，然后按 `[ENTER]`。这样创建了一个存储女性身高数据的数组。

按 `[ $\square$ ]` 将光标移到数组的第 1 行。底行显示 **HGHT(1)=**。

HGHT	L1	L2	1
	-----	-----	
HGHT(1)=			

3. 按 `169` `[ $\square$ ]` `43` 输入第 1 个身高值。输入时，它显示在底行。

按 `[ENTER]`。值显示在第 1 行，矩形条光标移到下一行。

用同样方法输入其他 9 个身高值。

HGHT	L1	L2	3
159.79			
181.42			
171.17			
162.04			
167.15			
159.53			
HGHT(1)=			

4. 按 **[STAT]** **[↓]** 显示 STAT TESTS 菜单，然后按 **[↓]** 直到 **8:TInterval** 亮显。

```

EDIT CALC TESTS
2:T-Test...
3:2-SampZTest...
4:2-SampTTest...
5:1-PropZTest...
6:2-PropZTest...
7:ZInterval...
8:TInterval...
  
```

5. 按 **[ENTER]** 选择 **8:TInterval**。显示 **TInterval** 的推理统计编辑器。如果没有为 **Inpt:** 选择 **Data**，按 **[↓]** **[ENTER]** 选择 **Data**。

```

TInterval
Inpt:DATA Stats
List:HGHT
Freq:1
C-Level:.99
Calculate
  
```

在 **List:** 提示处按 **[↓]** 和 **[H]** **[G]** **[H]** **[T]**（字母锁开启）。

按 **[↓]** **[↓]** **[□]** **99** 在 **C-Level:** 提示处输入 99% 的置信度。

6. 按 **[↓]** 将光标移到 **Calculate** 上，然后按 **[ENTER]**。计算置信区间，**TInterval** 结果显示在主屏幕中。

```

TInterval
(159.74,173.94)
x̄=166.838
Sx=6.907879237
n=10
  
```

解释所得的结果。

第 1 行，即 **(159.74,173.94)** 表示总体平均值的 99% 置信区间大约在 159.74 厘米和 173.94 厘米之间，范围约为 14.2 厘米。

.99 置信度表示在大量的样本中，我们期望所计算区间 99% 包含总体平均值。示例中实际的总体平均值是 165.1 厘米（介绍：页 13-2），它在计算区间之中。

第 2 行给出用来计算此区间的样本平均身高  $\bar{x}$ 。第 3 行给出样本标准差 **Sx**。底行给出样本大小 **n**。

要获得更精确的女性身高的总体平均值  $\mu$ ，将样本大小增加到 90。使用样本平均值  $\bar{x}$  (163.8) 和样本标准差  $Sx$  (7.1)，这些值是由较大的随机样本（介绍；页 13-2）计算得到的。这次使用 **Stats**（概要统计）输入选项。

7. 按 **[STAT]** **[↓]** **8** 显示 **TInterval** 的推理统计编辑器。

按 **[▶]** **[ENTER]** 选择 **Inpt:Stats**。编辑器发生变化以便可输入概要统计。

```
TInterval
Inpt:Data Stats
x̄:166.838
Sx:6.907879237...
n:10
C-Level: .99
Calculate
```

8. 按 **[▼]** **163** **[.]** **8** **[ENTER]** 存储 163.8 到  $\bar{x}$ 。

按 **[7]** **[.]** **1** **[ENTER]** 存储 7.1 到  $Sx$ 。

按 **90** **[ENTER]** 存储 90 到  $n$ 。

```
TInterval
Inpt:Data Stats
x̄:163.8
Sx:7.1
n:90
C-Level: .99
Calculate
```

9. 按 **[▼]** 将光标移到 **Calculate** 上，然后按 **[ENTER]** 计算新的 99% 置信区间。结果显示在主屏幕中。

```
TInterval
(161.83, 165.77)
x̄=163.8
Sx=7.1
n=90
```

如果女性总体的身高分布是正态分布，分布的平均值  $\mu$  为 165.1 厘米，标准差  $\sigma$  为 6.35 厘米，那么哪个身高仅被 5% 的女性（第 95 个百分位）超过？

10. 按 **[CLEAR]** 清除主屏幕。

按 **[2nd]** **[DISTR]** 显示 **DISTR**（分布）菜单。

```
DISTR DRAW
1:normalpdf(
2:normalcdf(
3:invNorm(
4:tpdf(
5:tcdf(
6:X2pdf(
7:X2cdf(
```

11. 按 **3** 将 **invNorm(** 粘贴到主屏幕。

按  $\square$  **.95**  $\square$  **165**  $\square$  **.1**  $\square$  **6**  $\square$  **.35**  $\square$  **ENTER**。

**.95** 是区域， **165.1** 是  $\mu$ ， **6.35** 是  $\sigma$ 。

```
invNorm(.95,165.
1,6.35)
175.5448205
```

结果显示在主屏幕中，它表明有 5% 的女性身高超过 175.5 厘米。

现在绘出图象并遮蔽总体最高的 5%。

12. 按 **WINDOW** 并设置窗口变量为以下值。

**Xmin=145**    **Ymin=-.02**    **Xres=1**  
**Xmax=185**    **Ymax=.08**  
**Xscl=5**      **Yscl=0**

```
WINDOW
Xmin=145
Xmax=185
Xscl=5
Ymin=-.02
Ymax=.08
Yscl=0
Xres=1
```

13. 按 **2nd** **[DISTR]**  $\square$  显示 **DISTR DRAW** 菜单。

```
DISTR 0310
1:ShadeNorm(
2:Shade_t(
3:ShadeX^2(
4:ShadeF(
```

14. 按 **ENTER** 将 **ShadeNorm(** 粘贴到主屏幕。

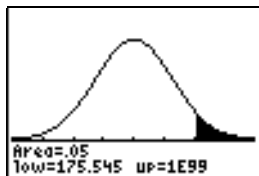
按 **2nd** **[ANS]**  $\square$  **1** **2nd** **[EE]** **99**  $\square$  **165**  $\square$  **.1**  $\square$  **6**  $\square$  **.35**  $\square$ 。

**Ans**（步骤 11 的结果是 175.5448205）是下限。  
**1E99** 是上限。正态曲线由平均值  $\mu$  (165.1) 和标准差  $\sigma$  (6.35) 定义。

```
invNorm(.95,165.
1,6.35)
175.5448205
ShadeNorm(Ans,1E
99,165.1,6.35)■
```

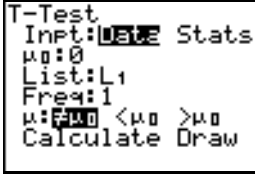
15. 按 **ENTER** 画图并遮蔽正态曲线。

**Area** 是第 95 个百分位以上的区域。 **low** 是下限。  
**up** 是上限。



## 显示推理统计编辑器

当从主屏幕中选择假设检验或置信区间指令时，显示相应的推理统计编辑器。编辑器根据每个检验或区间的输入要求不同而变化。下面是 **T-Test** 的推理统计编辑器。



**注意：**如果选择 **ANOVA** 指令，它将粘贴到主屏幕中。 **ANOVA** 没有编辑器屏幕。

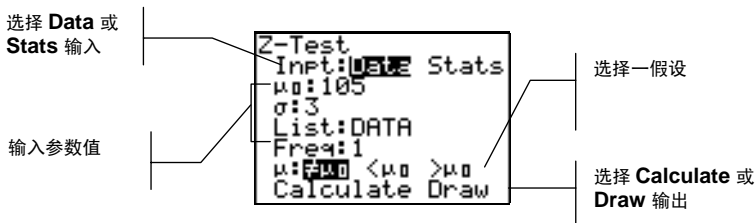
## 使用推理统计编辑器

要使用推理统计编辑器，请按以下步骤进行：

1. 从 **STAT TESTS** 菜单选择一个假设检验或置信区间。显示相应编辑器。
2. 选择 **Data** 或 **Stats** 输入（如果此选项有效）。显示相应编辑器。
3. 在编辑器中对每个参数输入实数、数组名或表达式。
4. 选择所要检验的择一假设（**≠**、**<** 或 **>**）（如果此选项有效）。
5. 对 **Pooled** 选项，选择 **No** 或 **Yes**（如果此选项有效）。
6. 选择 **Calculate** 或 **Draw**（当 **Draw** 有效时）执行指令。
  - 选择 **Calculate** 时，结果显示在主屏幕中。
  - 选择 **Draw** 时，结果显示在图象中。

本章介绍上述步骤中提到的每个假设检验和置信区间指令的选项。





### 选择 Data 或 Stats

大多数推理统计编辑器提示您选择两种输入类型之一。（**1-PropZTest** 和 **2-PropZTest**、**1-PropZInt** 和 **2-PropZInt**、 $\chi^2$ -Test，和 **LinRegTTest** 例外。）

- 选择 **Data** 输入数据数组。
- 选择 **Stats** 输入概要统计，例如  $\bar{x}$ 、**Sx** 和 **n**。

要选择 **Data** 或 **Stats**，将光标移到 **Data** 或 **Stats** 上，然后按 **[ENTER]**。

### 输入参数值

推理统计编辑器要求每个参数都要有一个值。如果了解某个参数符号的含义，请参阅页 13-26 和 13-27 中的表。

在所有推理统计编辑器中输入值时，TI-83 Plus 将它们存储在内存中，从而无需重新输入每个值就可运行许多检验或区间。

### 选择择一假设 ( $\neq < >$ )

对于假设检验，大多数推理统计编辑器提示选择 3 种择一假设之一。

- 第 1 个是  $\neq$  择一假设，例如 **Z-Test** 的  $\mu \neq \mu_0$ 。
- 第 2 个是  $<$  择一假设，例如 **2-SampTTest** 的  $\mu_1 < \mu_2$ 。
- 第 3 个是  $>$  择一假设，例如 **2-PropZTest** 的  $p_1 > p_2$ 。

要选择择一假设，将光标移到相应的择一假设上，然后按 **[ENTER]**。

### 选择合并选项

**Pooled** (只有 **2-SampTTest** 和 **2-SampTInt**) 指定对于计算是否合并方差。

- 如果不需合并方差, 选择 **No**。总体方差可以不等。
- 如果需要合并方差, 选择 **Yes**。假定总体方差相等。

要选择 **Pooled** 选项, 将光标移到 **Yes**, 然后按 **[ENTER]**。

### 选择计算或绘制假设检验

在假设检验的推理统计编辑器中输入所有参数之后, 必须选择是在主屏幕中 (**Calculate**) 还是在图象屏幕中 (**Draw**) 查看计算结果。

- **Calculate** 计算检验结果并在主屏幕中显示输出。
- **Draw** 绘出检验结果的图象, 并在图象中显示检验统计结果和 **p** 值。窗口变量自动调整以适合此图象。

要选择 **Calculate** 或 **Draw**, 将光标移到 **Calculate** 或 **Draw** 上, 然后按 **[ENTER]**。将立即执行所选的指令。

### 选择计算置信区间

在置信区间的推理统计编辑器中输入所有参数之后, 选择 **Calculate** 显示结果。 **Draw** 选项无效。

按 **[ENTER]** 时, **Calculate** 计算置信区间结果, 并在主屏幕中显示输出。

### 忽略推理统计编辑器

要将假设检验或置信区间指令粘贴到主屏幕中而不显示相应的推理统计编辑器, 则从 **CATALOG** 菜单选择指令。附录 A 讲述了每个假设检验和置信区间指令的输入语法。

```
2-SampZTest(
```

**注意:** 可将一个假设检验或置信区间指令粘贴到程序的命令行上。在程序编辑器中, 从 **CATALOG** (第 15 章) 或 **STAT TESTS** 菜单选择指令。

## STAT TESTS 菜单

要显示 STAT TESTS 菜单，请按 **[STAT]** **[ $\blacktriangleleft$ ]**。在选择推理统计指令时，将显示相应的推理统计编辑器。

大多数 STAT TESTS 指令将一些输出变量存储到内存中。多数输出变量位于 TEST 子菜单（ VARS 菜单； **5:Statistics** ）。若要查看这些变量的列表，请参阅页 13-28 。

---

### EDIT CALC TESTS

1: Z-Test...	检验 1 个 $\mu$ ， $\sigma$ 已知
2: T-Test...	检验 1 个 $\mu$ ， $\sigma$ 未知
3: 2-SampZTest...	检验比较 2 个 $\mu$ ， $\sigma$ 已知
4: 2-SampTTest...	检验比较 2 个 $\mu$ ， $\sigma$ 未知
5: 1-PropZTest...	检验 1 个比例
6: 2-PropZTest...	检验比较 2 个比例
7: ZInterval...	1 个 $\mu$ 的置信区间， $\sigma$ 已知
8: TInterval...	1 个 $\mu$ 的置信区间， $\sigma$ 未知
9: 2-SampZInt...	2 个 $\mu$ 差的置信区间， $\sigma$ 已知
0: 2-SampTInt...	2 个 $\mu$ 差的置信区间， $\sigma$ 未知
A: 1-PropZInt...	1 个比例的置信区间
B: 2-PropZInt...	2 个比例差的置信区间
C: $\chi^2$ -Test...	双向表格的 $\chi^2$ 检验
D: 2-SampFTest...	检验比较 2 个 $\sigma$
E: LinRegTTest...	回归斜率和 $\rho$ 的 $t$ 检验
F: ANOVA(	单向方差分析

---

**注意：**当计算一个新的检验或区间时，所有以前的输出变量是无效的。

## STAT TESTS 指令的推理统计编辑器

在本章中说明每条 STAT TESTS 指令时，显示其独有的推理统计编辑器，其中包含示例参数。

- 提供 **Data/Stats** 输入选择的指令显示两种输入屏幕。
- 不提供 **Data/Stats** 输入选择的指令只显示一种输入屏幕。

然后显示每条 STAT TESTS 指令独有的输出屏幕，其中包含示例结果。

- 提供 **Calculate/Draw** 输出选择的指令显示两种屏幕：计算和图象结果。
- 只提供 **Calculate** 输出选择的指令在主屏幕中显示计算结果。

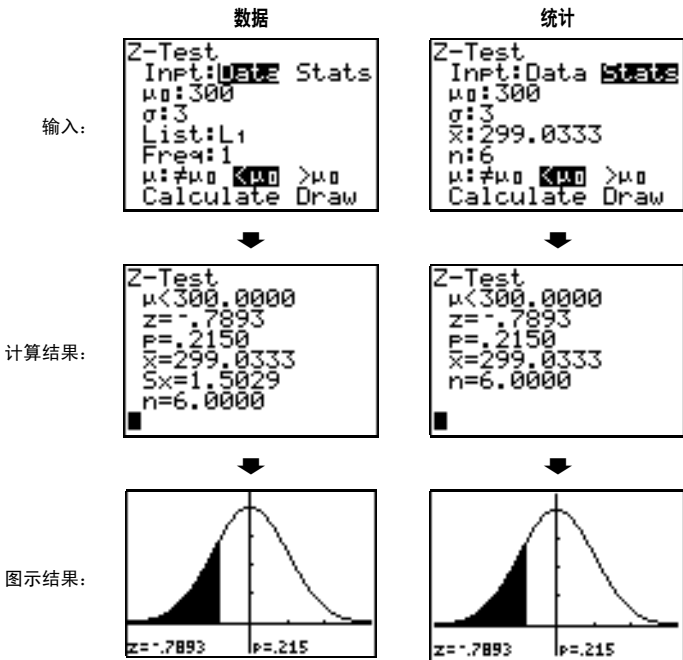
Z-Test

**Z-Test** (单样本  $z$  检验; 项 1) 执行单个未知总体平均值  $\mu$  的假设检验 (总体标准差  $\sigma$  已知)。它对以下择一条件之一检验零假设  $H_0: \mu = \mu_0$ 。

- $H_a: \mu \neq \mu_0$  ( $\mu: \neq \mu_0$ )
- $H_a: \mu < \mu_0$  ( $\mu: < \mu_0$ )
- $H_a: \mu > \mu_0$  ( $\mu: > \mu_0$ )

本例中:

$L_1 = \{299.4 \ 297.7 \ 301 \ 298.9 \ 300.2 \ 297\}$



**注意:** 页 13-10 到 13-25 中的所有示例都假设定点十进制方式设置为 4 (第 1 章)。如果设置十进制表示方式为 **Float** 或不同的定点十进制表示设置, 则其输出可能不同于本例中的输出。

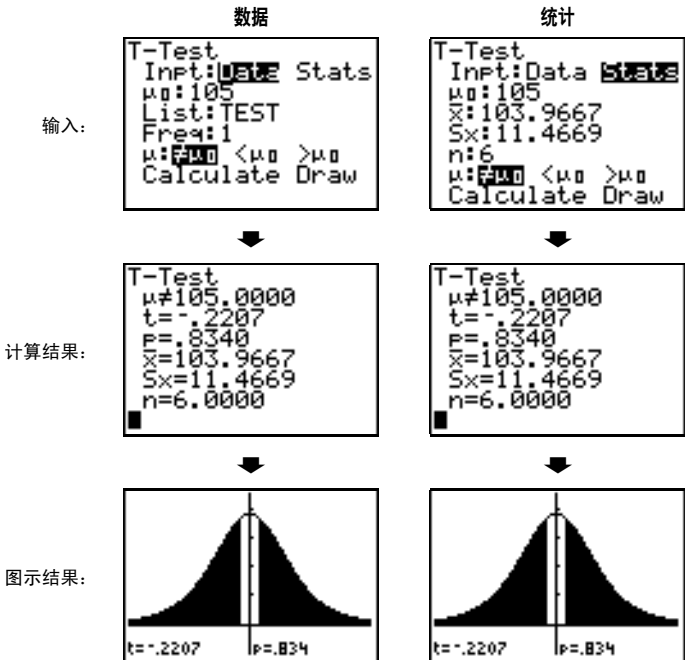
## T-Test

**T-Test** (单样本  $t$  检验; 项 2) 执行单个未知总体平均值  $\mu$  的假设检验 (总体标准差  $\sigma$  未知)。它对以下择一条件之一检验零假设  $H_0: \mu = \mu_0$ 。

- $H_a: \mu \neq \mu_0$  ( $\mu: \neq \mu_0$ )
- $H_a: \mu < \mu_0$  ( $\mu: < \mu_0$ )
- $H_a: \mu > \mu_0$  ( $\mu: > \mu_0$ )

本例中:

TEST={91.9 97.8 111.4 122.3 105.4 95}



2-SampZTest

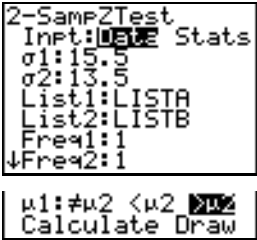
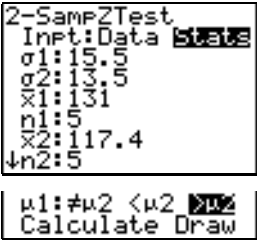
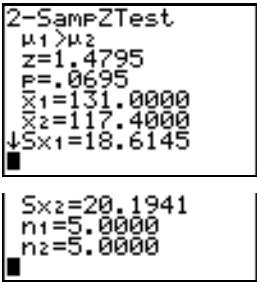
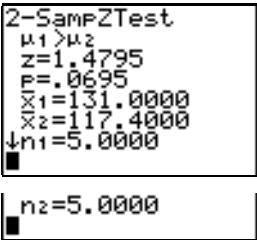
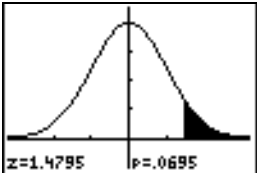
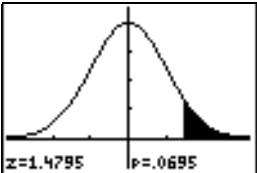
**2-SampZTest** (双样本  $z$  检验; 项 3) 检验两个独立样本的总体平均值  $\mu_1$  和  $\mu_2$  的等式 (总体标准差  $\sigma_1$  和  $\sigma_2$  已知)。它对以下择一条件之一检验零假设  $H_0: \mu_1 = \mu_2$ 。

- $H_a: \mu_1 \neq \mu_2$  ( $\mu_1 \neq \mu_2$ )
- $H_a: \mu_1 < \mu_2$  ( $\mu_1 < \mu_2$ )
- $H_a: \mu_1 > \mu_2$  ( $\mu_1 > \mu_2$ )

本例中:

LISTA={154 109 137 115 140}

LISTB={108 115 126 92 146}

	数据	统计
输入:		
计算结果:		
图示结果:		

## 2-SampTTest

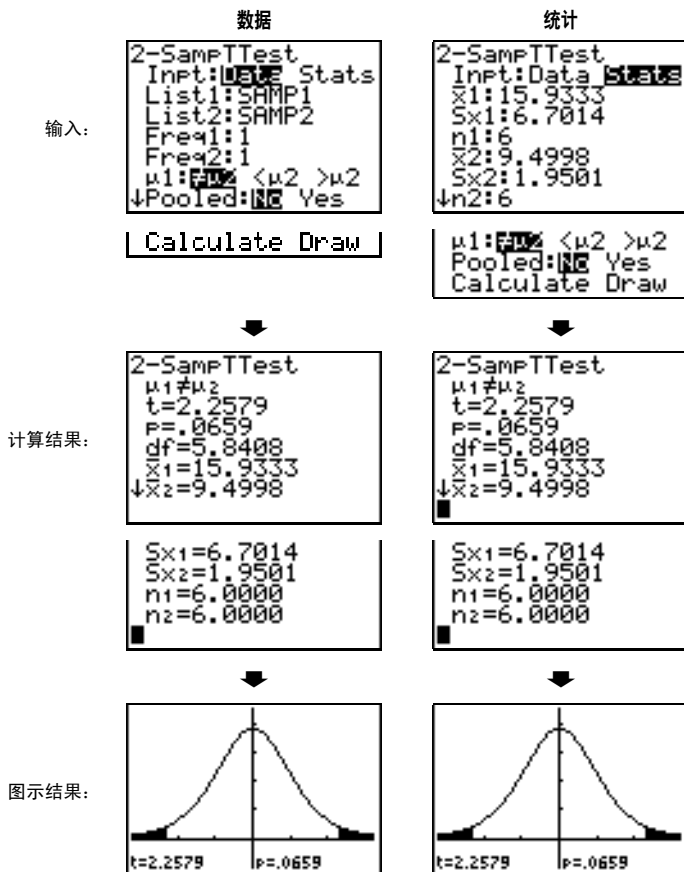
**2-SampTTest** (双样本  $t$  检验; 项 4) 检验两个独立样本的总体平均值  $\mu_1$  和  $\mu_2$  的等式 (总体标准差  $\sigma_1$  和  $\sigma_2$  未知)。它对以下择一条件之一检验零假设  $H_0: \mu_1 = \mu_2$ 。

- $H_a: \mu_1 \neq \mu_2$  ( $\mu_1 \neq \mu_2$ )
- $H_a: \mu_1 < \mu_2$  ( $\mu_1 < \mu_2$ )
- $H_a: \mu_1 > \mu_2$  ( $\mu_1 > \mu_2$ )

本例中:

**SAMP1**={12.207 16.869 25.05 22.429 8.456 10.589}

**SAMP2**={11.074 9.686 12.064 9.351 8.182 6.642}



1-PropZTest

**1-PropZTest** (单比例  $z$  检验; 项 5) 检验未知的成功比例 (prop), 其输入为样本的成功次数  $x$  和样本的观测次数  $n$ 。**1-PropZTest** 对以下择一条件之一检验零假设  $H_0: \text{prop}=p_0$ 。

- $H_a: \text{prop} \neq p_0$  (**prop:≠p0**)
- $H_a: \text{prop} < p_0$  (**prop:<p0**)
- $H_a: \text{prop} > p_0$  (**prop:>p0**)

输入:

```

1-PropZTest
P0:.5
x:2048
n:4040
PROP:≠P0 <P0 >P0
Calculate Draw
    
```



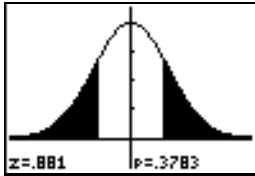
计算结果:

```

1-PropZTest
PROP#:5000
z=.8810
P=.3783
p=.5069
n=4040.0000
    
```



图示结果:





## 2-PropZTest

**2-PropZTest** (双比例  $z$  检验; 项 6) 进行检验以比较两个总体的成功比例 ( $p_1$  和  $p_2$ )。其输入为每个样本的成功次数 ( $x_1$  和  $x_2$ ) 和每个样本的观测次数 ( $n_1$  和  $n_2$ )。**2-PropZTest** 对以下择一条件之一检验零假设  $H_0: p_1=p_2$  (使用合并样本比例  $\hat{p}$ )。

- $H_a: p_1 \neq p_2$  (**p1: $\neq$ p2**)
- $H_a: p_1 < p_2$  (**p1:<p2**)
- $H_a: p_1 > p_2$  (**p1:>p2**)

输入:

```
2-PropZTest
x1:45
n1:61
x2:38
n2:62
P1:0.7377 <P2 >P2
Calculate Draw
```



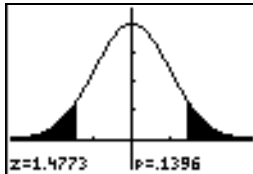
计算结果:

```
2-PropZTest
P1 $\neq$ P2
z=1.4773
P=.1396
 $\hat{p}_1$ =.7377
 $\hat{p}_2$ =.6129
 $\downarrow$  $\hat{p}$ =.6748
█
```

```
n1=61.0000
n2=62.0000
```



图示结果:



ZInterval

ZInterval (单样本  $z$  置信区间; 项 7) 计算未知总体平均值  $\mu$  的置信区间 (总体标准差  $\sigma$  已知)。计算得到的置信区间取决于用户指定的置信度。

本例中:

L1={299.4 297.7 301 298.9 300.2 297}

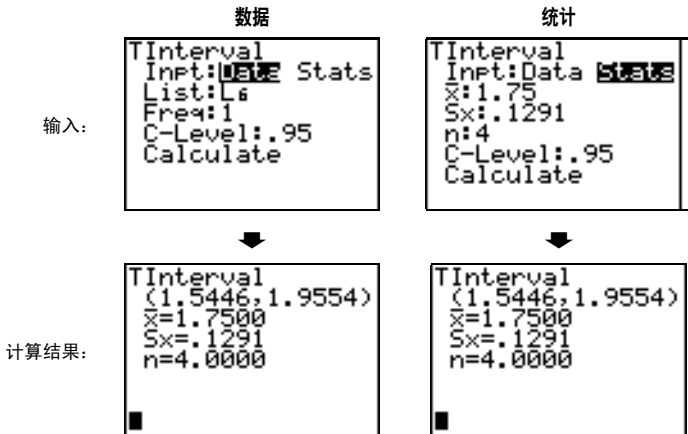


## Interval

**Interval** (单样本  $t$  置信区间; 项 8) 计算未知总体平均值  $\mu$  的置信区间 (总体标准差  $\sigma$  未知)。计算得到的置信区间取决于用户指定的置信度。

本例中:

**L6={1.6 1.7 1.8 1.9}**



2-SampZInt

**2-SampZInt** (双样本  $z$  置信区间; 项 9) 计算两个总体平均值差 ( $\mu_1 - \mu_2$ ) 的置信区间 (总体标准差  $\sigma_1$  和  $\sigma_2$  已知)。计算得到的置信区间取决于用户指定的置信度。

本例中:

LISTC={154 109 137 115 140}

LISTD={108 115 126 92 146}

	数据	统计
输入:	<pre>2-SampZInt Inpt: Data Stats σ1:15.5 σ2:13.5 List1:LISTC List2:LISTD Freq1:1 ↓Freq2:1</pre>	<pre>2-SampZInt Inpt:Data Stats σ1:15.5 σ2:13.5 x̄1:131 n1:5 x̄2:117.4 ↓n2:5</pre>
	<pre>C-Level:.99 Calculate</pre>	<pre>C-Level:.99 Calculate</pre>
	↓	↓
计算结果:	<pre>2-SampZInt (-10.08,37.278) x̄1=131.0000 x̄2=117.4000 Sx1=18.6145 Sx2=20.1941 ↓n1=5.0000</pre>	<pre>2-SampZInt (-10.08,37.278) x̄1=131.0000 x̄2=117.4000 n1=5.0000 n2=5.0000</pre>
	<pre>n2=5.0000</pre>	

## 2-SampTInt

**2-SampTInt** (双样本  $t$  置信区间; 项 0) 计算两个总体平均值差 ( $\mu_1 - \mu_2$ ) 的置信区间 (总体标准差  $\sigma_1$  和  $\sigma_2$  未知)。计算得到的置信区间取决于用户指定的置信度。

本例中:

SAMP1={12.207 16.869 25.05 22.429 8.456 10.589}

SAMP2={11.074 9.686 12.064 9.351 8.182 6.642}

	数据	统计
输入:	<pre>2-SampTInt Inpt: Data Stats List1: SAMP1 List2: SAMP2 Freq1: 1 Freq2: 1 C-Level: .95 ↓Pooled: <input type="checkbox"/> Yes</pre>	<pre>2-SampTInt Inpt: Data Stats x1: 15.9333 Sx1: 6.7014 n1: 6 x2: 9.4998 Sx2: 1.9501 ↓n2: 6</pre>
	<pre>Calculate</pre>	<pre>C-Level: .95 Pooled: <input type="checkbox"/> Yes Calculate</pre>
	↓	↓
计算结果:	<pre>2-SampTInt (-.5848, 13.452) df=5.8408 x1=15.9333 x2=9.4998 Sx1=6.7014 ↓Sx2=1.9501</pre>	<pre>2-SampTInt (-.5849, 13.452) df=5.8408 x1=15.9333 x2=9.4998 Sx1=6.7014 ↓Sx2=1.9501</pre>
	<pre>n1=6.0000 n2=6.0000</pre>	<pre>n1=6.0000 n2=6.0000</pre>

## 1-PropZInt

**1-PropZInt** (单比例  $z$  置信区间; 项 **A**) 计算未知成功比例的置信区间, 其输入为样本的成功次数  $x$  和样本的观测次数  $n$ 。计算得到的置信区间取决于用户指定的置信度。

输入:

```
1-PropZInt
x:2048
n:4040
C-Level:.99
Calculate
```



计算结果:

```
1-PropZInt
(.4867,.5272)
p=.5069
n=4040.0000
```

## 2-PropZInt

**2-PropZInt** (双比例  $z$  置信区间; 项 **B**) 计算两个总体的成功比例差 ( $p_1 - p_2$ ) 的置信区间, 其输入为每个样本的成功次数 ( $x_1$  和  $x_2$ ) 和每个样本的观测次数 ( $n_1$  和  $n_2$ )。计算得到的置信区间取决于用户指定的置信度。

输入:

```
2-PropZInt
x1:49
n1:61
x2:38
n2:62
C-Level:.95
Calculate
```



计算结果:

```
2-PropZInt
(.0334,.3474)
p1=.8033
p2=.6129
n1=61.0000
n2=62.0000
■
```

$\chi^2$ -Test

$\chi^2$ -Test ( $\chi^2$  检验; 项 C) 计算关联的  $\chi^2$  检验, 这种关联是指所指定 *Observed* (观测) 矩阵中次数的双向表格中的关联。双向表格的零假设  $H_0$  是: 行变量与列变量之间不存在关联。择一假设是: 变量是相关的。

计算  $\chi^2$ -Test 前, 在矩阵中输入观测次数。在  $\chi^2$ -Test 编辑器中的 **Observed:** 提示处输入矩阵变量名, 默认值是 [A]。在 **Expected:** 提示处, 输入矩阵变量名, 用于存储计算得到的期望数, 变量名的默认值是 [B]。

矩阵编辑器:

```

MATRIX[A] 3 × 2
[ 5.0000  19.0000  ]
[ 8.0000  16.0000  ]
[ 11.0000 13.0000  ]
    
```

注意: 按 [2nd] [MATRX]  $\downarrow$   $\square$  1, 从 MATRIX EDIT 菜单选择 1:[A]。

输入:

```

 $\chi^2$ -Test
Observed: [A]
Expected: [B]
Calculate Draw
    
```



计算结果:

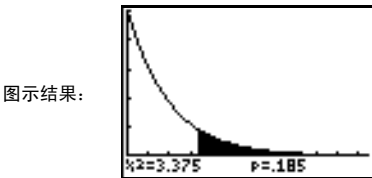
```

 $\chi^2$ -Test
 $\chi^2=3.3750$ 
 $p=.1850$ 
 $df=2.0000$ 
    
```

```

[B]
[[ 8.0000  16.000...
 [ 8.0000  16.000...
 [ 8.0000  16.000...
    
```

注意: 按 [2nd] [MATRX] [B] [ENTER], 显示矩阵 [B]。





## 2-SampFTest

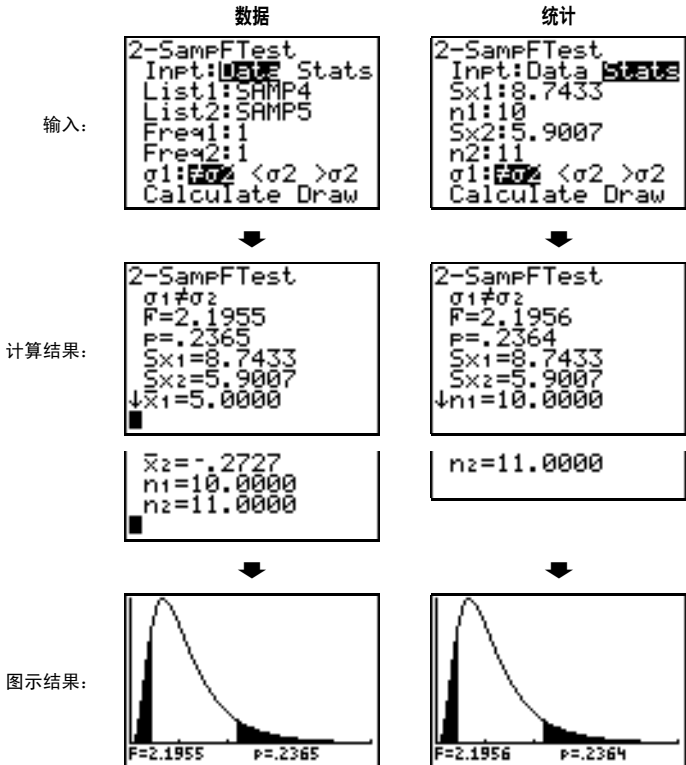
**2-SampFTest** (双样本 F 检验; 项 D) 计算 F 检验以比较两个正态总体标准差 ( $\sigma_1$  和  $\sigma_2$ )。总体平均值和标准差都是未知的。**2-SampFTest** 使用样本方差的比  $Sx1^2/Sx2^2$  对以下择一条件之一检验零假设  $H_0: \sigma_1 = \sigma_2$ 。

- $H_a: \sigma_1 \neq \sigma_2$  ( $\sigma_1: \neq \sigma_2$ )
- $H_a: \sigma_1 < \sigma_2$  ( $\sigma_1: < \sigma_2$ )
- $H_a: \sigma_1 > \sigma_2$  ( $\sigma_1: > \sigma_2$ )

本例中:

SAMP4={ 7 -4 18 17 -3 -5 1 10 11 -2}

SAMP5={-1 12 -1 -3 3 -5 5 2 -11 -1 -3}



LinRegTTest

**LinRegTTest** (线性回归  $t$  检验; 项 **E**) 对给定的数据计算线性回归, 并对方程  $y=\alpha+\beta x$  的斜率  $\beta$  和相关系数  $\rho$  的值计算  $t$  检验。它对以下择一条件之一检验零假设  $H_0: \beta=0$  (即  $\rho=0$ )。

- $H_a: \beta \neq 0$  and  $\rho \neq 0$  ( $\beta$  &  $\rho: \neq 0$ )
- $H_a: \beta < 0$  and  $\rho < 0$  ( $\beta$  &  $\rho: < 0$ )
- $H_a: \beta > 0$  and  $\rho > 0$  ( $\beta$  &  $\rho: > 0$ )

回归方程自动存储到 **RegEQ** ( VARS Statistics EQ 子菜单) 中。如果在 **RegEQ**:提示处输入一个  $Y=$  变量名, 则计算得到的回归方程将自动存储到指定的  $Y=$  方程中。在以下示例中, 回归方程存储到后来选择 (开启) 的 **Y1** 中。

本例中:

L3={ 38 56 59 64 74}  
L4={ 41 63 70 72 84}

输入:

```
LinRegTTest
Xlist:L3
Ylist:L4
Freq:1
 $\beta$  &  $\rho: \neq 0 < 0 > 0$ 
RegEQ:Y1
Calculate
```



计算结果:

```
LinRegTTest
y=a+bx
 $\beta \neq 0$  and  $\rho \neq 0$ 
t=15.9405
p=5.3684e-4
df=3.0000
 $\downarrow$ a=-3.6596
```

```
Plot1 Plot2 Plot3
Y1=-3.6596+1.1969X
Y2=
Y3=
Y4=
Y5=
Y6=
```

```
 $\uparrow$ b=1.1969
s=1.9820
r2=.9883
r=.9941
```

当 **LinRegTTest** 执行时, 将创建离差数组, 并自动将其存储到数组 **RESID** 中。 **RESID** 被放在 LIST NAMES 菜单中。

**注意:** 对于回归方程, 可使用定点十进制表示方式设置来控制小数点后的位数 (第 1 章)。然而, 限制小数位数较小时可能影响拟合的精度。

## ANOVA(

**ANOVA**(单向方差分析; 项 **F**) 为比较 2 到 20 个总体的平均值而计算单向方差分析。比较这些平均值的 ANOVA 过程涉及样本数据的方差分析。它对择一假设  $H_a$  ( $\mu_1 \dots \mu_k$  不全相等) 检验零假设  $H_0: \mu_1 = \mu_2 = \dots = \mu_k$ 。

**ANOVA**(list1,list2[,...,list20])

本例中:

L1={7 4 6 6 5}

L2={6 5 5 8 7}

L3={4 7 6 7 6}

输入:

```
ANOVA(L1,L2,L3)
```



计算结果:

```
One-way ANOVA
F=.3111
p=.7384
Factor
df=2.0000
SS=.9333
↓ MS=.4667
```

```
Error
df=12.0000
SS=18.0000
MS=1.5000
SXP=1.2247
```

**注意:** **SS** 是平方和, **MS** 是均方。

## 推理统计输入描述

本节的表格描述了在本章中所讨论的推理统计输入。在推理统计编辑器中为这些输入键入值。这些输入在表中出现的顺序与在本章中出现的顺序相同。

输入	描述
$\mu_0$	正在检验的总体平均值的假设值。
$\sigma$	已知的总体标准差；必须是 $> 0$ 的实数。
List	包含检验数据的数组名。
Freq	包含 List 数据中频数值的数组名。频数默认值为 1。所有元素必须是 $\geq 0$ 的整数。
Calculate/Draw	确定产生检验和置信区间输出的类型。 <b>Calculate</b> 将输出显示在主屏幕中。检验时， <b>Draw</b> 绘出结果的图象。
$\bar{x}$ , Sx, n	单样本检验和区间的概要统计（平均值、标准差和样本大小）。
$\sigma_1$	双样本检验和区间中第 1 个总体中已知的总体标准差。必须是 $> 0$ 的实数。
$\sigma_2$	双样本检验和区间中第 2 个总体中已知的总体标准差。必须是 $> 0$ 的实数。
List1, List2	双样本检验和区间中包含检验数据的数组名。默认值分别为 L1 和 L2。
Freq1, Freq2	双样本检验和区间中包含 List1 和 List2 中数据频数的数组名。频数默认值为 1。所有元素必须是 $\geq 0$ 的整数。
$\bar{x}_1$ , Sx1, n1, $\bar{x}_2$ , Sx2, n2	双样本检验和区间中样本 1 和样本 2 的概要统计（平均值、标准差和样本大小）。
Pooled	指定对于 <b>2-SampTTest</b> 和 <b>2-SampTInt</b> 是否合并方差。 <b>No</b> 要求 TI-83 不合并方差。 <b>Yes</b> 要求 TI-83 合并方差。

输入	描述
$p_0$	<b>1-PropZTest</b> 中期望的样本比例。必须是一个 $0 < p_0 < 1$ 的实数。
$x$	<b>1-PropZTest</b> 和 <b>1-PropZInt</b> 中样本的成功次数。必须是 $\geq 0$ 的整数。
$n$	<b>1-PropZTest</b> 和 <b>1-PropZInt</b> 中样本的观测次数。必须是 $> 0$ 的整数。
$x_1$	<b>2-PropZTest</b> 和 <b>2-PropZInt</b> 中样本 1 的成功次数。必须是 $\geq 0$ 的整数。
$x_2$	<b>2-PropZTest</b> 和 <b>2-PropZInt</b> 中样本 2 的成功次数。必须是 $\geq 0$ 的整数。
$n_1$	<b>2-PropZTest</b> 和 <b>2-PropZInt</b> 中样本 1 的观测次数。必须是 $> 0$ 的整数。
$n_2$	<b>2-PropZTest</b> 和 <b>2-PropZInt</b> 中样本 2 的观测次数。必须是 $> 0$ 的整数。
<b>C-Level</b>	区间指令的置信度。必须 $\geq 0$ 且 $< 100$ 。如果它 $\geq 1$ ，则假定它是一个百分数并被 100 除。默认值是 0.95。
<b>Observed (Matrix)</b>	矩阵名，表示 $\chi^2$ -Test 中次数的双向表格的观测值的行和列。 <b>Observed</b> 必须包含所有 $\geq 0$ 的整数。矩阵维数必须至少为 $2 \times 2$ 。
<b>Expected (Matrix)</b>	存储期望值的矩阵名。 <b>Expected</b> 在 $\chi^2$ -Test 成功完成后创建。
<b>Xlist, Ylist</b>	包含 <b>LinRegTTest</b> 中数据的数组名。默认值分别为 <b>L1</b> 和 <b>L2</b> 。 <b>Xlist</b> 和 <b>Ylist</b> 的长度必须相等。
<b>RegEQ</b>	$Y=$ 变量名的提示，变量中存储计算得到的回归方程。如果指定了 $Y=$ 变量，则自动选择（开启）方程。默认时仅将回归方程存储到 <b>RegEQ</b> 变量。

## 检验和区间输出变量

推理统计变量如下表所示进行计算。要访问这些变量用于表达式中，则按 **[VARS]**、**5 (5:Statistics)**，然后选择下表最后一列中列出的 **VARS** 菜单。

变量	检验	区间	LinRegTTest, ANOVA	VARS 菜单
p 值	<b>p</b>		<b>p</b>	TEST
检验统计	<b>z, t, <math>\chi^2</math>, F</b>		<b>t, F</b>	TEST
自由度	<b>df</b>	<b>df</b>	<b>df</b>	TEST
样本 1 和样本 2 的 x 值样本平均值	$\bar{x}1, \bar{x}2$	$\bar{x}1, \bar{x}2$		TEST
样本 1 和样本 2 的 x 值样本标准差	<b>Sx1, Sx2</b>	<b>Sx1, Sx2</b>		TEST
样本 1 和样本 2 的数据点数	<b>n1, n2</b>	<b>n1, n2</b>		TEST
合并的标准差	<b>SxP</b>	<b>SxP</b>	<b>SxP</b>	TEST
估计的样本比例	$\hat{p}$	$\hat{p}$		TEST
总体 1 的估计样本比例	$\hat{p}1$	$\hat{p}1$		TEST
总体 2 的估计样本比例	$\hat{p}2$	$\hat{p}2$		TEST
置信区间对		<b>lower, upper</b>		TEST
x 值的平均值	$\bar{x}$	$\bar{x}$		XY
x 的样本标准差	<b>Sx</b>	<b>Sx</b>		XY
数据点的数量	<b>n</b>	<b>n</b>		XY
直线的标准差			<b>s</b>	TEST
回归/拟合系数			<b>a, b</b>	EQ
相关系数			<b>r</b>	EQ
可靠系数			<b>r<sup>2</sup></b>	EQ
回归方程			<b>RegEQ</b>	EQ

**注意：**不能归档以上列出的变量。

## DISTR 菜单

要显示 DISTR 菜单，请按 **[2nd]** [DISTR]。

DISTR	DRAW	
1:	normalpdf(	正态概率密度
2:	normalcdf(	正态分布概率
3:	invNorm(	反正态累积分布
4:	tpdf(	t 概率密度
5:	tcdf(	t 分布概率
6:	$\chi^2$ pdf(	$\chi^2$ 概率密度
7:	$\chi^2$ cdf(	$\chi^2$ 分布概率
8:	Fpdf(	F 概率密度
9:	Fcdf(	F 分布概率
0:	binompdf(	二项概率
A:	binomcdf(	二项累积密度
B:	poissonpdf(	泊松概率
C:	poissoncdf(	泊松累积密度
D:	geometpdf(	几何概率
E:	geometcdf(	几何累积密度

**注意：**-1E99 和 1E99 指定了无穷。例如，如果要观察 *upperbound* 左边的区域，可指定 *lowerbound* = -1E99。

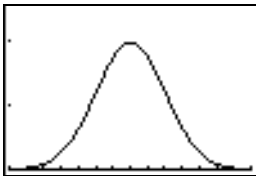
## normalpdf(

**normalpdf(** 计算正态分布的概率密度函数 (pdf) 在指定  $x$  处的值。默认值为：平均值  $\mu=0$ ，标准差  $\sigma=1$ 。若要绘出这个正态分布，将 **normalpdf(** 粘贴到  $Y=$  编辑器中。概率密度函数 (pdf) 是：

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \sigma > 0$$

**normalpdf(x[, $\mu$ , $\sigma$ ])**

```
Plot1 Plot2 Plot3
\Y1=normalpdf(X,
35,2)
```



**注意：**对于本例，

**Xmin = 28**

**Xmax = 42**

**Ymin = 0**

**Ymax = .25**

**提示：**为绘出正态分布，可设置窗口变量 **Xmin** 和 **Xmax** 以便平均值  $\mu$  位于它们之间，然后从 **ZOOM** 菜单选择 **0:ZoomFit**。

**normalcdf(**

**normalcdf(** 对于指定平均值  $\mu$  和标准差  $\sigma$  计算下限 *lowerbound* 和上限 *upperbound* 之间的正态分布概率。默认值为:  $\mu=0$ ,  $\sigma=1$ 。

**normalcdf(lowerbound,upperbound[, $\mu,\sigma$ ])**

```
normalcdf(-1E99,
36, 35, 2)
.6914624678
```

**invNorm(**

**invNorm(** 对于指定平均值  $\mu$  和标准差  $\sigma$  的正态分布曲线下给定的区域 *area* 计算反正态累积分布函数。它计算与  $x$  值左侧区域 *area* 相关联的  $x$  值。 $0 \leq area \leq 1$  必须成立。默认值为  $\mu=0$ ,  $\sigma=1$ 。

**invNorm(area[, $\mu,\sigma$ ])**

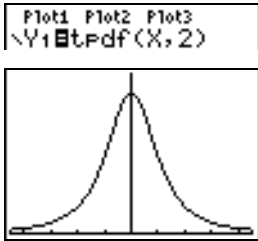
```
invNorm(.6914624
678, 35, 2)
36.00000004
```

**tpdf(**

**tpdf(** 计算  $t$  累积分布的概率密度函数 (pdf) 在指定  $x$  值处的值。 $df$  (自由度) 必须  $>0$ 。若要绘出  $t$  累积分布, 将 **tpdf(** 粘贴到 Y= 编辑器中。概率密度函数 (pdf) 是:

$$f(x) = \frac{\Gamma[(df+1)/2]}{\Gamma(df/2)} \frac{(1+x^2/df)^{-(df+1)/2}}{\sqrt{\pi df}}$$

**tpdf(x,df)**



注意: 对于本例,  
**Xmin = -4.5**  
**Xmax = 4.5**  
**Ymin = 0**  
**Ymax = .4**



**tcdf**(

**tcdf**( 对于指定的  $df$  (自由度) 计算下限 *lowerbound* 和上限 *upperbound* 之间的  $t$  分布概率,  $df$  必须  $> 0$  )。

**tcdf**(*lowerbound,upperbound,df*)

```
tcdf(-2,3,18)
.9657465644
```

**$\chi^2$ pdf**(

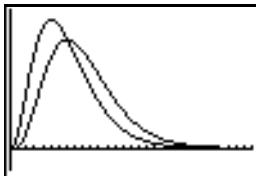
**$\chi^2$ pdf**( 计算  $\chi^2$  分布的概率密度函数 (pdf) 在指定  $x$  值处的值。  $df$  (自由度) 必须是  $> 0$  的整数。若要绘出  $\chi^2$  分布, 将  **$\chi^2$ pdf**( 粘贴到  $Y=$  编辑器中。概率密度函数 (pdf) 是:

$$f(x) = \frac{1}{\Gamma(df/2)} (1/2)^{df/2} x^{df/2 - 1} e^{-x/2}, x \geq 0$$

**$\chi^2$ pdf**( $x,df$ )

```
F1ot1 F1ot2 F1ot3
\Y1 X^2 Pdf(X, 9)
\Y2 X^2 Pdf(X, 7)
\Y3 =
\Y4 =
\Y5 =
\Y6 =
\Y7 =
```

注意: 对于本例,  
**Xmin = 0**  
**Xmax = 30**  
**Ymin = .02**  
**Ymax = .132**



**$\chi^2$ cdf**(

**$\chi^2$ cdf**( 对于指定的  $df$  (自由度) 计算下限 *lowerbound* 和上限 *upperbound* 之间的  $\chi^2$  分布概率,  $df$  必须是  $> 0$  的整数。

**$\chi^2$ cdf**(*lowerbound,upperbound,df*)

```
\chi^2cdf(0,19.023,9)
.9750019601
```

**Fpdf**(

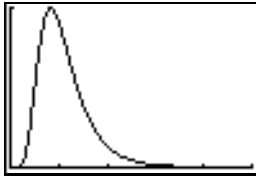
**Fpdf**( 计算 F 分布的概率密度函数 (pdf) 在指定  $x$  值处的值。自由度分子 *numerator df* (自由度) 和自由度分母 *denominator df* 必须是  $> 0$  的整数。要绘出 F 分布, 请将 **Fpdf**( 粘贴到 Y= 编辑器中。概率密度函数 (pdf) 是:

$$f(x) = \frac{\Gamma[(n+d)/2]}{\Gamma(n/2)\Gamma(d/2)} \left(\frac{n}{d}\right)^{n/2} x^{n/2-1} (1+nx/d)^{-(n+d)/2}, x \geq 0$$

其中  $n$  = 自由度的分子  
 $d$  = 自由度的分母

**Fpdf(x,numerator df,denominator df)**

```
Plot1 Plot2 Plot3
\Y1 FPDF(X, 24, 19)
```



注意: 对于本例,  
**Xmin = 0**  
**Xmax = 5**  
**Ymin = 0**  
**Ymax = 1**

**Fcdf**(

**Fcdf**( 对于指定的自由度分子 *numerator df* (自由度) 和自由度分母 *denominator df* 计算下限 *lowerbound* 和上限 *upperbound* 之间的 F 分布概率。 *numerator df* 和 *denominator df* 必须是  $> 0$  的整数。

**Fcdf(lowerbound,upperbound,numerator df,denominator df)**

```
Fcdf(0, 2.4523, 24, 19)
.9749989576
```

## binompdf(

**binompdf(** 在指定离散二项分布的试验数 *numtrials* 和每次试验的成功概率 (*p*) 时计算它在 *x* 处的概率。*x* 可以是整数或是整数数组。 $0 \leq p \leq 1$  必须成立。*numtrials* 必须是  $> 0$  的整数。如果不指定 *x*，则返回从 0 到 *numtrials* 的概率数组。概率密度函数 (pdf) 是：

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}, x = 0, 1, \dots, n$$

其中  $n = \text{numtrials}$

**binompdf(numtrials,p[,x])**

```
binompdf(5,.6,{3
,4,5})
(.3456 .2592 .0...
```

## binomcdf(

**binomcdf(** 在指定离散二项分布的试验数 *numtrials* 和每次试验的成功概率 (*p*) 时计算它在 *x* 处的累积概率。*x* 可以是实数或是实数数组。 $0 \leq p \leq 1$  必须成立。*numtrials* 必须是  $> 0$  的整数。如果不指定 *x*，则返回累积概率数组。

**binomcdf(numtrials,p[,x])**

```
binomcdf(5,.6,{3
,4,5})
(.66304 .92224 ...
```

## poissonpdf(

**poissonpdf(** 在指定离散泊松分布的平均值  $\mu$  时计算它在 *x* 处的概率。 $\mu$  必须是  $> 0$  的实数。*x* 可以是整数或是整数数组。概率密度函数 (pdf) 是：

$$f(x) = e^{-\mu} \mu^x / x!, x = 0, 1, 2, \dots$$

**poissonpdf( $\mu,x$ )**

```
Poissonpdf(6,10)
.0413030934
```

### poissoncdf(

**poissoncdf(** 在指定离散泊松分布的平均值  $\mu$  时计算它在  $x$  处的累积概率。  
 $\mu$  必须是  $> 0$  的实数。  $x$  可以是实数或是实数数组。

**poissoncdf( $\mu, x$ )**

```
Poissoncdf(.126,  
{0,1,2,3})  
{.8816148468 .9...
```

### geometpdf(

**geometpdf(** 在指定离散几何分布的成功概率  $p$  时计算它在  $x$ ，即首次成功的试验次数处的概率。  $0 \leq p \leq 1$  必须成立。  $x$  可以是整数或是整数数组。概率密度函数 (pdf) 是：

$$f(x) = p(1-p)^{x-1}, x = 1, 2, \dots$$

**geometpdf( $p, x$ )**

```
GeometPdf(.4,6)  
.031104
```

### geometcdf(

**geometcdf(** 在指定离散几何分布的成功概率  $p$  时计算它在  $x$ ，即首次成功的试验次数处的累积概率。  $0 \leq p \leq 1$  必须成立。  $x$  可以是实数或是实数数组。

**geometcdf( $p, x$ )**

```
Geometcdf(.5, {1,  
2,3})  
{.5 .75 .875}
```

## DISTR DRAW 菜单

要显示 DISTR DRAW 菜单, 请按  $\boxed{2nd}$  [DISTR]  $\boxed{\downarrow}$ 。DISTR DRAW 指令绘出不同类型的密度函数, 遮蔽由下限 *lowerbound* 和上限 *upperbound* 指定的区域, 并显示计算得到的面积值。

若要清除绘图, 则从 DRAW 菜单中选择 **1:ClrDraw** (第 8 章)。

**注意:** 执行 DISTR DRAW 指令前, 必须设置窗口变量使屏幕适应所要求的分布。

---

### DISTR DRAW

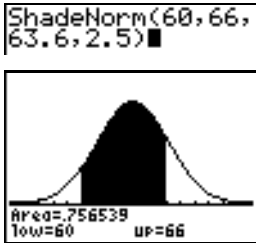
- 1: ShadeNorm(           遮蔽正态分布。
  - 2: Shade\_t(            遮蔽 *t* 分布。
  - 3: Shade $\chi^2$ (         遮蔽  $\chi^2$  分布。
  - 4: ShadeF(             遮蔽 **F** 分布。
- 

**注意:**  $-1E99$  和  $1E99$  指定了无穷。例如, 如果要观察上限 *upperbound* 左边的区域, 可指定 *lowerbound* =  $-1E99$ 。

## ShadeNorm(

**ShadeNorm(** 绘出由平均值  $\mu$  和标准差  $\sigma$  指定的正态密度函数, 并遮蔽下限 *lowerbound* 和上限 *upperbound* 之间的区域。默认值为:  $\mu=0$ ,  $\sigma=1$ 。

**ShadeNorm(lowerbound,upperbound[, $\mu,\sigma$ ])**

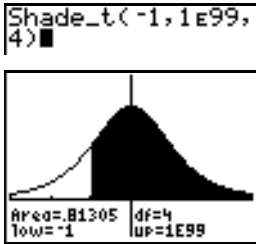


**注意:** 对于本例,  
**Xmin = 55**  
**Xmax = 72**  
**Ymin = -.05**  
**Ymax = .2**

Shade\_t(

Shade\_t( 绘出由  $df$  (自由度) 指定的  $t$  分布的密度函数, 并遮蔽下限  $lowerbound$  和上限  $upperbound$  之间的区域。

Shade\_t(lowerbound,upperbound,df)

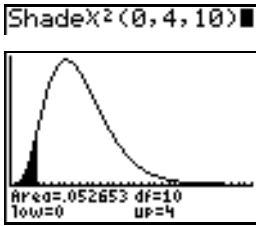


注意: 对于本例,  
Xmin = -3  
Xmax = 3  
Ymin = -.15  
Ymax = .5

Shade $\chi^2$ (

Shade $\chi^2$ ( 绘出由  $df$  (自由度) 指定的  $\chi^2$  分布的密度函数, 并遮蔽下限  $lowerbound$  和上限  $upperbound$  之间的区域。

Shade $\chi^2$ (lowerbound,upperbound,df)

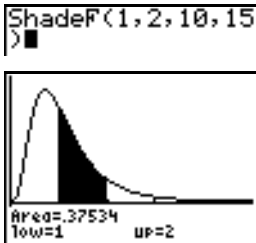


注意: 对于本例,  
Xmin = 0  
Xmax = 35  
Ymin = -.025  
Ymax = .1

ShadeF(

ShadeF( 绘出由自由度分子  $numerator\ df$  (自由度) 和自由度分母  $denominator\ df$  指定的  $F$  分布的密度函数, 并遮蔽下限  $lowerbound$  和上限  $upperbound$  之间的区域。

ShadeF(lowerbound,upperbound,numerator df,denominator df)

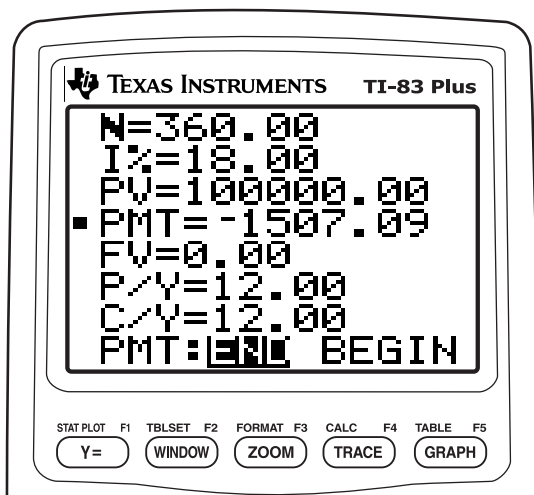


注意: 对于本例,  
Xmin = 0  
Xmax = 5  
Ymin = -.25  
Ymax = .9

# 14 应用

## 目录

Applications 菜单.....	2
入门：投资汽车.....	3
入门：计算复利.....	4
使用 TVM Solver.....	5
使用金融函数.....	6
计算货币的时间价值 (TVM).....	7
计算现金流.....	9
计算分期付款.....	10
换算利息.....	13
计算日期期间的天数/定义支付方式.....	14
使用 TVM 变量.....	15
CBL/CBR 应用程序.....	16
数据采集方式和选项.....	18
开始数据采集.....	25
停止数据采集.....	26

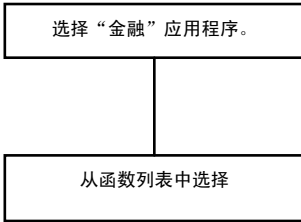


TI-83 Plus 提供的“金融”和 CBL/CBR 应用程序已列于 APPLICATIONS 菜单中。除“金融”应用程序外，用户可根据存储空间大小来添加和删除应用程序。“金融”应用程序内置于 TI-83 Plus 代码中，不能删除。

用户可以购买 TI-83 Plus 软件的其他应用程序，以进一步定制计算器的功能。计算器在 ROM 内存中特别为应用程序保留了 160 KB 的空间。

## 运行“金融”应用程序的步骤

使用“金融”应用程序时，请按下列基本步骤执行。



按 **[APPS]** **[ENTER]**。

```
APPLICATIONS
1: Finance...
2: CBL/CBR
```

```
MODE VARS
1: TVM Solver...
2: tvm_Pmt
3: tvm_I%
4: tvm_PV
5: tvm_N
6: tvm_FV
7: ↓nPV(
```



入门是对本章内容的快速介绍。详细信息请阅读本章。

您已找到要购买的汽车，汽车的价格为 9,000。您在 4 年中每月能支付 250。年利率 (APR) 为多少时您才有可能买得起这种汽车呢？

1. 按 **MODE** **▾** **▾** **▾** **▾** **ENTER** 将定点十进制表示方式设置设为 **2**。TI-83 Plus 以 2 位小数显示所有的数。

```
Normal Sci Eng
Float 0123456789
Radian Degree
Func Par Pol Seq
Connected Dot
Sequential Simul
Real a+bi re^ti
Full Horiz G-T
```

2. 按 **APPS** **ENTER**，在 APPLICATIONS 菜单中选择 **1:Finance**。

```
▶▶▶ VARS
1▶TVM Solver...
2:tvm_Pmt
3:tvm_I%
4:tvm_PV
5:tvm_N
6:tvm_FV
7↓npv(
```

3. 按 **ENTER**，在 CALC VARS 菜单中选择 **1:TVM Solver**。屏幕显示 TVM Solver。

```
N=0.00
I%=0.00
PV=0.00
PMT=0.00
FV=0.00
P/Y=1.00
C/Y=1.00
PMT:END BEGIN
```

按 **48** **ENTER** 将 48 (月) 保存到 **N** 中。按 **▾** **9000** **ENTER** 将 9,000 保存到 **PV** 中。按 **▾** **250** **ENTER** 将 -250 保存到 **PMT** 中。(负号表示现金流出。)按 **0** **ENTER** 将 0 保存到 **FV** 中。按 **12** **ENTER**，将 12 (每年支付 12 次) 保存到 **P/Y** 中，同时将 12 (每年 12 个复合周期) 保存到 **C/Y** 中。将 **P/Y** 置成 12，计算年利率 (月复利) **1%**。按 **▾** **ENTER** 选择 **PMT:END**，它表示在每期期末支付。

```
N=48.00
I%=0.00
PV=9000.00
PMT=-250.00
FV=0.00
P/Y=12.00
C/Y=12.00
PMT:END BEGIN
```

4. 按 **▾** **▾** **▾** **▾** **▾** **▾** 将光标移到提示符 **I%** 上。按 **ALPHA** **[SOLVE]** 求解 **1%**。您所期待的年利率是多少呢？

```
N=48.00
I%=14.90
PV=9000.00
PMT=-250.00
FV=0.00
P/Y=12.00
C/Y=12.00
PMT:END BEGIN
```

当年利率（月复利）是多少时，1,250 在 7 年后将累积到 2,000？

注意：因为当求解复利问题时不存在支付，所以 PMT 必须置成 0，P/Y 必须置成 1。

1. 按 **[APPS]** **[ENTER]**，在 APPLICATIONS 菜单中选择 **1:Finance**。

```
0:APPS VARS
1:TVM Solver...
2:tvm_Pmt
3:tvm_I%
4:tvm_PV
5:tvm_N
6:tvm_FV
7:↓nPV()
```

2. 按 **[ENTER]**，在 CALC VARS 菜单中选择 **1:TVM Solver**。屏幕显示 TVM Solver。按 **7** 输入以年为单位的期数。按 **□ □ □ 1250** 输入现值作为现金流出（投资）。按 **□ 0** 指定不存在支付。按 **□ 2000** 输入未来值作为现金流入（收益）。按 **□ 1** 输入每年的支付期数。按 **□ 12** 将每年的复利计算期设成 12。

```
N=7
I%=0
PV=-1250
PMT=0
FV=2000
P/Y=1
C/Y=12
PMT:END BEGIN
```

3. 按 **□ □ □ □ □** 将光标置于 **I%** 提示处。

```
N=7
I%=■
PV=-1250
PMT=0
FV=2000
P/Y=1
C/Y=12
PMT:END BEGIN
```

4. 按 **[ALPHA]** **[SOLVE]** 求解年利率 **I%**。

```
N=7.00
I%=6.73
PV=-1250.00
PMT=0.00
FV=2000.00
P/Y=1.00
C/Y=12.00
PMT:END BEGIN
```

## 使用 TVM Solver

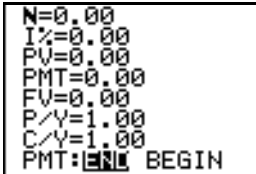
TVM Solver 显示货币的时间价值 (TVM) 变量。TVM Solver 根据给定的 4 个变量的值求解第 5 个变量。

FINANCE VARS 菜单一节中 (页 14-15) 介绍这 5 个 TVM 变量 (**N**、**I%**、**PV**、**PMT** 和 **FV**)、**P/Y** 和 **C/Y**。

TVM Solver 中的 **PMT: END BEGIN** 对应于 FINANCE CALC 菜单中的项 **Pmt\_End** (期末支付) 和 **Pmt\_Bgn** (期初支付)。

要求解未知 TVM 变量, 请按下列步骤进行:

1. 按 **[APPS]** **[ENTER]** **[ENTER]** 显示 TVM Solver。下面的屏幕显示了这些变量的默认值, 此时定点十进制表示方式设置为 2 个小数位。

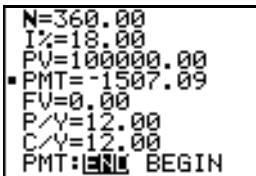


```
N=0.00
I%=0.00
PV=0.00
PMT=0.00
FV=0.00
P/Y=1.00
C/Y=1.00
PMT: [ ] BEGIN
```

2. 输入 4 个 TVM 变量的已知值。

**注意:** 以正数输入现金流入; 以负数输入现金流出。

3. 输入 **P/Y** 的值, 同时自动把相同的值输入到 **C/Y** 中; 如果 **P/Y**  $\neq$  **C/Y**, 则可将不同的值输入到 **C/Y** 中。
4. 选择 **END** 或 **BEGIN** 指定支付方法。
5. 将光标放到要求解的 TVM 变量上。
6. 按 **[ALPHA]** **[SOLVE]** 开始计算, 结果显示在 TVM Solver 中, 并保存到相应的 TVM 变量中。左列中的方块指示符表示所指变量的值是求解得到的。



```
N=360.00
I%=18.00
PV=100000.00
PMT=-1507.09
FV=0.00
P/Y=12.00
C/Y=12.00
PMT: [ ] BEGIN
```

### 输入现金流入和现金流出

使用 TI-83 Plus 的金融函数时，必须以正数输入现金流入（收到的现金），以负数输入现金流出（支付的现金）。TI-83 Plus 计算和显示结果时遵循这种约定。

### FINANCE CALC 菜单

要显示 FINANCE CALC 菜单，请按 **[APPS]** **[ENTER]**。

---

<b>CALC</b>	<b>VARs</b>	
<b>1:</b>	<b>TVM Solver...</b>	显示 TVM Solver。
<b>2:</b>	<b>tvm_Pmt</b>	计算每次支付的金额。
<b>3:</b>	<b>tvm_1%</b>	计算年利率。
<b>4:</b>	<b>tvm_PV</b>	计算现值。
<b>5:</b>	<b>tvm_N</b>	计算支付期数。
<b>6:</b>	<b>tvm_FV</b>	计算未来值。
<b>7:</b>	<b>npv(</b>	计算净现值。
<b>8:</b>	<b>irr(</b>	计算内部回报率。
<b>9:</b>	<b>bal(</b>	计算分期付款计划表的余额。
<b>0:</b>	<b>ΣPrn(</b>	计算分期付款计划表的本金总额。
<b>A:</b>	<b>ΣInt(</b>	计算分期付款计划表的利息总额。
<b>B:</b>	<b>▶Nom(</b>	计算额定利率。
<b>C:</b>	<b>▶Eff(</b>	计算实际利率。
<b>D:</b>	<b>dbd(</b>	计算两个日期之间的天数。
<b>E:</b>	<b>Pmt_End</b>	选择普通年金（期末）。
<b>F:</b>	<b>Pmt_Bgn</b>	选择期初年金（期初）。

---

在主屏幕中使用这些函数设置和执行金融计算。

### TVM Solver

**TVM Solver** 显示 TVM Solver（页 14-5）。

## 计算货币的时间价值

使用货币的时间价值 (TVM) 函数 (菜单项 **2-6**) 可分析金融业务, 如年金、贷款、抵押、租赁和储蓄。

每个 TVM 函数有 0-6 个参数, 它们必须是实数。所指定作为这些函数参数的值不会保存到 TVM 变量中 (页 14-15)。

**注意:** 要将值保存到 TVM 变量中, 可使用 TVM Solver (页 14-5), 或者使用 **STO** 和 FINANCE VARS 菜单 (页 14-15) 中的任一 TVM 变量。

如果输入的参数少于 6 个, 那么 TI-83 Plus 将用预先保存的 TVM 变量值代替未指定的参数。

如果输入 TVM 函数的参数, 则必须将参数放到括号中。

### tvm\_Pmt

tvm\_Pmt 计算每次支付的金额。

tvm\_Pmt[(N,I%,PV,FV,P/Y,C/Y)]

```
N=360
I%=8.5
PV=100000
PMT=0
FV=0
P/Y=12
C/Y=12
PMT: END BEGIN
```

```
tvm_Pmt          -768.91
tvm_Pmt(360,9.5)
                  -840.85
```

**注意:** 在上述示例中, 这些值先保存到 TVM Solver 的 TVM 变量中, 然后在主屏幕中用 TVM Solver 中的值来计算支付金额 (tvm\_Pmt)。接着, 将利率变为 9.5, 以说明利率对支付金额的影响。

## tvm\_I%

tvm\_I% 计算年利率。

tvm\_I%[(N,PV,PMT,FV,P/Y,C/Y)]

tvm_I%(48,10000, -250,0,12)	
Ans→I%	9.24
	9.24

## tvm\_PV

tvm\_PV 计算现值。

tvm\_PV[(N,I%,PMT,FV,P/Y,C/Y)]

360→N:11→I%:-100	
0→PMT:0→FV:12→P/	
Y	12.00
tvm_PV	105006.35

## tvm\_N

tvm\_N 计算支付期数。

tvm\_N[(I%,PV,PMT,FV,P/Y,C/Y)]

6→I%:9000→PV:-35	
0→PMT:0→FV:3→P/Y	
	3.00
tvm_N	36.47

## tvm\_FV

tvm\_FV 计算未来值。

tvm\_FV[(N,I%,PV,PMT,P/Y,C/Y)]

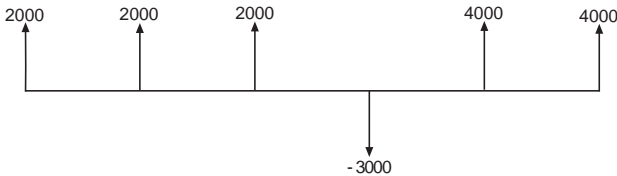
6→N:8→I%:-5500→P	
V:0→PMT:1→P/Y	
	1.00
tvm_FV	8727.81

## 计算现金流

使用现金流函数（菜单项 **7** 和 **8**）可分析多个相同时间周期后货币的价值。可以输入不同的现金流，现金流可以是现金流入或现金流出。介绍 **npv**（和 **irr**）的语法时使用了下列参数。

- *interest rate* 是用来体现一个周期现金流（货币成本）的利率。
- *CFO* 是时刻 0 处的初始现金流，必须是实数。
- *CFList* 是一个数组，它由初始现金流 *CFO* 之后的所有现金流值组成。
- *CFFreq* 是一个数组。每个数组元素分别表示一种分组的（连续的）现金流值的出现频数，现金流值是 *CFList* 中的相应元素。频数的默认值为 1；如果输入它的值，则必须是小于 10,000 的正整数。

例如，下图所示数组中的现金流是不同的。



$CFO = 2000$

$CFList = \{2000, -3000, 4000\}$

$CFFreq = \{2, 1, 2\}$

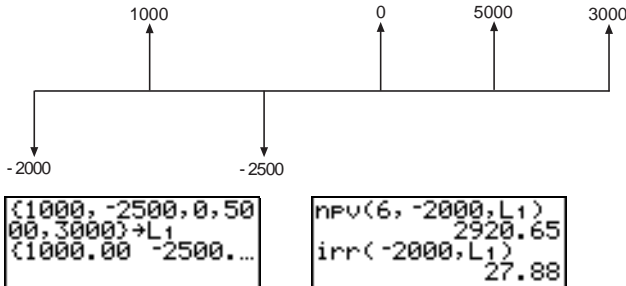
## npv(, irr(

**npv**（净现值）是现金流入和现金流出的现值之和。如果 **npv** 的结果是正数，则表示投资有盈利。

**npv**(*interest rate*, *CFO*, *CFList*, [*CFFreq*])

**irr**（内部回报率）是现金流的净现值等于 0 时的利率。

**irr**(*CFO*, *CFList*, [*CFFreq*])



## 计算分期付款计划表

使用分期付款函数（菜单项 **9**、**0** 和 **A**）可计算分期付款计划表的余额、本金总额及利息总额。

**bal**(

**bal**(用 **I%**、**PV** 和 **PMT** 的保存值计算分期付款计划表的余额。*npmt* 是用来计算余额的支付次数，它必须是小于 10,000 的正整数。*roundvalue* 规定计算器计算余额时所采用的内部精度；如果没有指定 *roundvalue*，TI-83 Plus 则使用 **Float/Fix** 十进制表示方式的当前设置。

**bal**(*npmt* [, *roundvalue*])

```
100000→PV:8.5→I%  
:-768.91→PMT:12→  
P/Y  
12.00
```

```
bal(12) 99244.07
```

**ΣPrn**(、**ΣInt**(

**ΣPrn**(用 **I%**、**PV** 和 **PMT** 的保存值计算分期付款计划表所指定的时间间隔内的本金总额。*pmt1*、*pmt2* 分别是这段时间中的期初支付额和期末支付额，它们必须是小于 10,000 的正整数。*roundvalue* 规定计算器计算本金时所采用的内部精度；如果没有指定 *roundvalue*，TI-83 Plus 则使用 **Float/Fix** 十进制表示方式的当前设置。

**注意：**计算本金前必须输入 **I%**、**PV** 和 **PMT** 的值。

**ΣPrn**(*pmt1*, *pmt2* [, *roundvalue*])

**ΣInt**(用 **I%**、**PV** 和 **PMT** 的保存值计算分期付款计划表所指定的时间间隔内的利息总额。*pmt1*、*pmt2* 分别是这段时间中的期初支付额和期末支付额，它们必须是小于 10,000 的正整数。*roundvalue* 规定计算器计算利息时所采用的内部精度；如果没有指定 *roundvalue*，TI-83 Plus 则使用 **Float/Fix** 十进制表示方式的当前设置。

**ΣInt**(*pmt1*, *pmt2* [, *roundvalue*])

```
360→N:100000→PV:  
8.5→I%:-768.91→P  
MT:12→P/Y  
12.00
```

```
ΣPrn(1,12) -755.93  
ΣInt(1,12) -8470.99
```



## 分期付款计划表实例：计算未付的贷款余额

您想用 30 年抵押贷款购置一套住宅，贷款的年利率 (APR) 为 8%，每月支付的金额为 800。现在，计算每次支付后未付的贷款余额，并以图象和表格的方式显示计算结果。

1. 按 **[MODE]**。按 **[<] [D] [D] [D] [ENTER]** 将定点十进制表示方式设置成 **2**。按 **[<] [D] [ENTER]** 选择 **Par** 图象方式。

```
Normal Sci Eng
Float 0123456789
Radian Degree
Func Par Pol Seq
Connected Dot
Sequential Simul
Real a+bi re^θi
Full Horiz G-T
```

2. 按 **[APPS] [ENTER] [ENTER]** 显示 TVM Solver。
3. 按 **[ENTER] 360** 输入支付次数。按 **[<] 8** 输入利率。按 **[<] [D] [C] 800** 输入支付金额。按 **[<] 0** 输入抵押贷款的未来值。按 **[<] 12** 输入每年的支付次数，也将每年的复利计算期设为 **12**。按 **[<] [D] [ENTER]** 选择 **PMT:END**。

```
N=360.00
I%=8.00
PV=0.00
PMT=-800.00
FV=0.00
P/Y=12.00
C/Y=12.00
PMT:END BEGIN
```

4. 按 **[<] [<] [<] [<] [<]** 将光标置于 **PV** 提示之上。按 **[ALPHA] [SOLVE]** 求解现值。

```
N=360.00
I%=8.00
PV=109026.80
PMT=-800.00
FV=0.00
P/Y=12.00
C/Y=12.00
PMT:END BEGIN
```

5. 按 **[Y=]** 显示参数 **Y=** 编辑器，关闭所有统计图象。按 **[X,T,θ,n]** 将 **X1T** 定义为 **T**。按 **[<] [APPS] [ENTER] 9 [X,T,θ,n] [ ]** 将 **Y1T** 定义为 **bal(T)**。

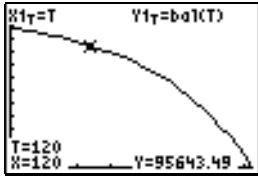
```
Plot1 Plot2 Plot3
X1T T
Y1T bal(T)
```

分期付款计划表实例：计算未付的贷款余额 (续)

6. 按 **[WINDOW]** 显示窗口变量。输入下列值：

<b>Tmin=0</b>	<b>Xmin=0</b>	<b>Ymin=0</b>
<b>Tmax=360</b>	<b>Xmax=360</b>	<b>Ymax=125000</b>
<b>Tstep=12</b>	<b>Xscl=50</b>	<b>Yscl=10000</b>

7. 按 **[TRACE]** 绘制图象，并激活跟踪光标。按 **[▶]** 和 **[◀]** 研究这个表示这段时间未付余额的图象。然后按 **[ENTER]** 查看指定时间 **T** 处的余额。



8. 按 **[2nd]** **[TBLSET]**，并输入下列值：

**TblStart=0**  
**ΔTbl=12**

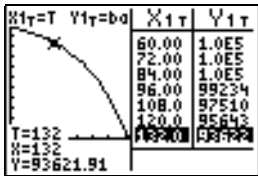
9. 按 **[2nd]** **[TABLE]** 显示未付余额 ( $Y_{1T}$ ) 表。

T	$X_{1T}$	$Y_{1T}$
0.00	0.00	109027
12.00	12.00	108116
24.00	24.00	107130
36.00	36.00	106061
48.00	48.00	104905
60.00	60.00	103652
72.00	72.00	102295

T=0

10. 按 **[MODE]** **[◀]** **[◀]** **[◀]** **[◀]** **[◀]** **[▶]** **[▶]** **[ENTER]** 选择 **G-T** 拆分屏幕方式，在这种方式下同时显示图象和表格。

按 **[TRACE]**，在表中显示出  $X_{1T}$  (时间) 和  $Y_{1T}$  (余额)。



## 换算利息

使用利息换算函数（菜单项 **B** 和 **C**），可将利率由年实际利率换算成额定利率（▶**Nom**()），或者由额定利率换算成年实际利率（▶**Eff**()）。

### ▶**Nom**(

▶**Nom**( 计算额定利率。*effective rate*（实际利率）和 *compounding periods*（复利计算期）必须是实数。*compounding periods*（复利计算期）必须大于 0。

▶**Nom**(*effective rate, compounding periods*)

```
▶Nom(15.87,4)
      15.00
```

### ▶**Eff**(

▶**Eff**( 计算实际利率。*nominal rate*（额定利率）和 *compounding periods*（复利计算期）必须是实数。*compounding periods*（复利计算期）必须大于 0。

▶**Eff**(*nominal rate, compounding periods*)

```
▶Eff(8,12)
      8.30
```

### dbd(

使用日期函数 **dbd**( (菜单项 **D**) )，采用“实际天数计算”法计算两个日期之间的天数。*date1* 和 *date2* 是标准日历中的日期或日期数组。

**注意：**日期必须在 1950 年 — 2049 年之间。

**dbd**(*date1*,*date2*)

可按下述两种格式之一输入 *date1* 和 *date2* 。

- MM.DDYY (美国)
- DDMM.YY (欧洲)

小数位数区分这两种日期格式。

```
dbd(12.3190, 12.3
192)
731.00
```

### 定义支付方法

**Pmt\_End** 和 **Pmt\_Bgn** (菜单项 **E** 和 **F**) 分别规定普通年金交易和期初年金交易。不管执行哪个命令，TVM Solver 都将更新。

#### Pmt\_End

**Pmt\_End** (期末支付) 规定普通年金。普通年金是指在支付期末进行支付，大多数贷款属于此类。**Pmt\_End** 为默认值。

**Pmt\_End**

在 TVM Solver 的 **PMT:END BEGIN** 行中，选择 **END**，将 **PMT** 设置为普通年金。

#### Pmt\_Bgn

**Pmt\_Bgn** (期初支付) 规定期初年金。期初年金是指在支付期初进行支付，大多数租赁属于此类。

**Pmt\_Bgn**

在 TVM Solver 的 **PMT:END BEGIN** 行中，选择 **BEGIN**，将 **PMT** 设置为期初年金。

## FINANCE VARS 菜单

要显示 FINANCE VARS 菜单，按 [APPS] [ENTER] [↓]。可以在主屏幕中使用 TVM 函数中的 TVM 变量，并将数值保存到这些变量中。

---

CALC	<b>VARS</b>	
1: N		总支付期数
2: I%		年利率
3: PV		现值
4: PMT		支付金额
5: FV		未来值
6: P/Y		每年的支付期数
7: C/Y		复利计算期数/年

---

## N、I%、PV、PMT、FV

**N**、**I%**、**V**、**PMT** 和 **FV** 这 5 个 TVM 变量表示上表所述的普通金融业务的要素。**I%** 表示年利率，基于 **P/Y** 和 **C/Y** 的值可将其换算成每期利率。

## P/Y 和 C/Y

**P/Y** 表示金融业务中每年的支付期数。

**C/Y** 表示金融业务中每年的复利计算期数。

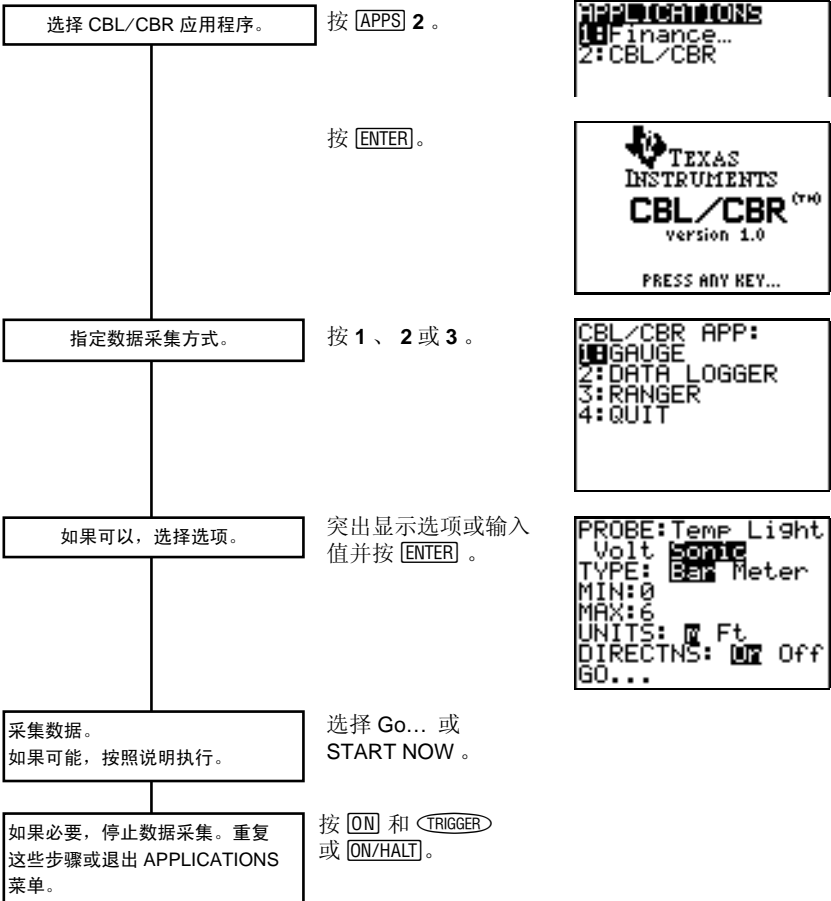
当把值保存到 **P/Y** 中时，**C/Y** 自动变成同样的值。如要把不同的值保存到 **C/Y** 中，则必须先将值保存到 **P/Y** 中，然后再给 **C/Y** 赋值。

# CBL/CBR 应用程序

CBL/CBR 应用程序允许用户采集各种实际数据。TI-83 Plus 提供的 CBL/CBR 应用程序已列于 APPLICATIONS 菜单中 (APPS 2)。

## 运行 CBL/CBR 应用程序的步骤

在使用 CBL/CBR 应用程序时，请按照以下基本步骤执行。用户不必每次执行所有步骤。



## 选择 CBL/CBR 应用程序

使用 CBL/CBR 应用程序，用户需要 CBL 或 CBR（如果可能）、TI-83 Plus 和设备间的连接线缆。

1. 按 **[APPS]**。



2. 选择 **2:CBL/CBR**，来设置 TI-83 Plus 使用哪一个应用程序。首先出现信息屏幕。



3. 按任意键继续显示下一个菜单。



## 在 CBL/CBR APP 菜单指定“数据采集方式”

通过 CBL 或 CBR，用户可以使用以下三种方法之一来采集数据：GAUGE（条形图或表盘图）、DATA LOGGER（Temp-Time、Light-Time、Volt-Time 或 Sonic-Time 图形）或 RANGER（它运行 RANGER 程序，该程序是内置的 CBR 数据采集程序）。

CBL/CBR APP 菜单包含数据采集方式。

CBL/CBR APP:

- |                |  |
|----------------|--|
| 1: GAUGE       | 以条形图或表盘图显示结果。  |
| 2: DATA LOGGER | 以 Temp-Time、Light-Time、Volt-Time 或 Sonic-Time 图显示结果。 |
| 3: RANGER      | 设置和运行 RANGER 程序。                                     |
| 4: QUIT        | 退出 CBL/CBR 应用程序。                                     |

**注意：**CBL 和 CBR 的区别在于：CBL 允许用户使用几种不同的探测器中的一种来采集数据，包括：Temp（温度）、Light、Volt（电压）或 Sonic。CBR 只能使用内置的 Sonic 探测器来采集数据。详细信息请参阅 CBL 和 CBR 的用户手册。

## 为每种数据采集方式指定选项

在 CBL/CBR APP 菜单选定数据采集方式后，出现的屏幕上显示该方式的对应选项。用户所选择的方式以及为该方式所选择的数据采集选项，确定了使用 CBR 还是 CBL。参考下面章节的图表以查找所使用的应用程序的选项。

### GAUGE

GAUGE 数据采集方式允许用户选择四种不同的探测器中的一种：Temp、Light、Volt 或 Sonic。

1. 按 [APPS] 2 [ENTER]。
2. 选择 1:GAUGE。
3. 选择选项。

```
CBL/CBR APP:
1: GAUGE
2: DATA LOGGER
3: RANGER
4: QUIT
```

```
PROBE: Temp Light
      Volt Sonic
TYPE: Bar Meter
MIN: 0
MAX: 100
UNITS: °F
DIRECTNS: Off
GO...
```



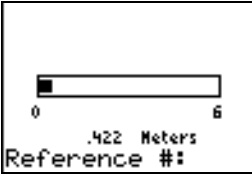
当选择探测器选项时，所有其他选项也随之做相应的改变。使用 **▶** 和 **◀** 在探测器选项间移动。要选择探测器，使用光标键突出显示所需的探测器，然后按 **[ENTER]**。

GAUGE 选项 (默认值)				
PROBE:	Temp	Light	Volt	Sonic
Type:	Bar 或 Meter			
Min:	0	0	-10	0
Max:	100	1	10	6
Units:	°C 或 °F	mW/cm <sup>2</sup>	Volt	m 或 Ft
Directions:	On 或 Off			

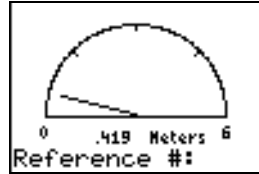
## TYPE

GAUGE 数据采集结果将根据 TYPE：Bar 或 Meter 来显示。使用光标键突出显示所需类型，然后按 **[ENTER]**。

Bar



Meter



## MIN 和 MAX

MIN 和 MAX 是指对于特定探测器的最大和最小的 UNIT 值。默认值在 19 页的表中列出。关于特定的 MIN/MAX 范围，请参阅 CBL 和 CBR 手册。使用数字键输入值。

## UNITS

结果将根据指定的 UNITS 显示。要指定测量单位（只适于 Temp 或 Sonic 探测器），请使用光标键突出所需的选项，然后按 **[ENTER]**。

### DIRECTNS (提示)

如果 DIRECTNS=On，那么计算器将在屏幕上显示操作步骤提示，它将帮助用户设置和运行数据采集。要选择 On 或 Off，使用光标键突出显示所需的选项，然后按 **[ENTER]**。

使用 Sonic 数据采集探测器时，如果 DIRECTNS=On，则在启动应用程序前，计算器显示菜单让用户选择 **1:CBL** 或 **2:CBR**。这将确保用户得到正确的提示。按 **1** 指定 CBL 或按 **2** 指定 CBR。

### 数据采集注释和结果

要标注指定的数据点，按 **[ENTER]** 来暂停数据采集。用户将会看到 Reference#: 提示符。使用数字键输入数字。计算器将自动使用下面的数组名称（用户不能重命名这些数组）将参考值和相应的结果转换为数组元素：

探测器	注释标签 (X) 存储于:	数据结果 (Y) 存储于:
Temp	LTREF	LTEMP
Light	LLREF	LLIGHT
Volt	LVREF	LVOLT
Sonic	LDREF	LDIST

要查看其中某个数组的所有元素，用户可将这些数组添加到“数组”编辑器中，这和添加其他数组是一样的。可从 **[2nd] [LIST] NAMES** 菜单获取数组名称。

**注意：**这些数组只是注释标签的临时占位符和特定探测器的数据结果。因此用户每次采集数据并为任何一个探测器输入注释时，这两个与该探测器相关的数组将由注释标签和最新采集的数据结果覆盖。

如果用户需要保存多个数据采集的注释标签和数据结果，则可以在每次数据采集之后，将所有需要保存的数组元素拷贝到一个名称不同的数组中。

而且，DATA LOGGER 数据采集方式将数据结果保存为相同的数组名称，并覆盖先前的数据采集结果（即使这些数据是由 GAUGE 数据采集方式采集的）。

## DATA LOGGER

1. 按 **[APPS]** **2** **[ENTER]**。

```
CBL/CBR APP:
1: GAUGE
2: DATA LOGGER
3: RANGER
4: QUIT
```

2. 选择 **2:DATA LOGGER** 。

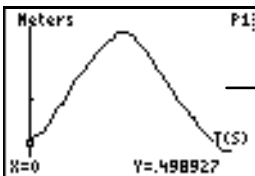
```
PROBE: Temp Light
Volt Sonic
#SAMPLES: 99
INTRVL(SEC): 1
UNITS: °C °F
PLOT: RealTime End
DIRECTNS: On Off
GO...
```

DATA LOGGER 数据采集方式允许从四个不同的探测器中选择一个：Temp、Light、Volt 或 Sonic。使用 CBL 可选择所有的探测器，而使用 CBR 则只能使用 Sonic 探测器。

当选择探测器选项时，所有其他选项将做相应的改变。使用 **[↑]** 和 **[↓]** 在探测器选项间移动。要选择探测器，使用光标键突出显示所需的探测器选项，然后按 **[ENTER]**。

DATA LOGGER 选项 (默认值)				
	Temp	Light	Volt	Sonic
#SAMPLES:	99	99	99	99
INTRVL (SEC):	1	1	1	1
UNITS:	°C 或 °F	mW/cm <sup>2</sup>	Volt	Cm 或 Ft
PLOT:	RealTime 或 End			
DIRECTNS:	On 或 Off			
Ymin (WINDOW):	0	0	-10	0
Ymax (WINDOW):	100	1	10	6

DATA LOGGER 数据采集结果将显示为 Temp-Time、Light-Time、Volt-Time 或 Distance-Time 图。



以米为单位的 Distance-Time 图  
(Sonic 探测器)。

### #SAMPLES

#SAMPLES 指采集多少个数据样本后开始绘图。例如，如果 #SAMPLES=99，则数据采集将在采集第 99 个样本后停止。使用数字键输入值。

### INTRVL (SEC)

INTRVL (SEC) 以秒为单位指定采集的数据样本的时间间隔。例如，如果需要采集 99 个样本并且 INTRVL=1，则数据采集将需要 99 秒。使用数字键输入值。有关间隔限制的详细信息，请参阅 CBL 或 CBR 手册。

### UNITS

结果将根据指定的 UNITS 显示。要指定测量单位（只适于 Temp 或 Sonic），请使用光标键突出显示测量单位，然后按 **[ENTER]**。

### PLOT

用户可以指定是否需要计算器采集实时 (RealTme) 样本，即数据采集后计算器立即绘出数据点，或等到所有的数据点都采集完毕 (End) 后才显示图形。使用光标键突出显示所需选项，然后按 **[ENTER]**。

### Ymin 和 Ymax

要指定最终图形的 Ymin 和 Ymax 值，按 **[WINDOW]** 来查看 PLOT WINDOW 屏幕。使用 **[↑]** 和 **[↓]** 在选项间移动。使用数字键输入 Ymin 和 Ymax。按 **[2nd]** **[QUIT]** 返回 DATA LOGGER 选项屏幕。

### DIRECTNS (提示)

如果 DIRECTNS=On，则计算器在屏幕显示操作步骤提示，它将帮助用户设置和运行数据采集。要选择 On 或 Off，使用光标键突出显示所需选项，然后按 **[ENTER]**。

使用 Sonic 数据采集探测器时，如果 DIRECTNS=On，则计算器在开始应用程序前显示菜单，让用户选择 **1:CBL** 或 **2:CBR**。这将确保用户获得正确的提示。按 **1** 指定 CBL，或按 **2** 指定 CBR。

---

## 数据采集结果

计算器将自动使用下面的数组名称（用户不能重命名这些数组）将所有收集的数据点转换为数组元素：

探测器	时间值 (X) 存储于:	数据结果 (Y) 存储于:
Temp	LTEMP	LTEMP
Light	LTLGHT	LLIGHT
Volt	LVOLT	LVOLT
Sonic	LTDIST	LDIST

要查看其中某个数组的所有元素，用户可将数组添加到“数组”编辑器中，这和添加其他数组是一样的。可从 **2nd** [LIST] NAMES 菜单获取数组名称。

**注意：**这些数组只是特定探测器所收集的数据结果的临时占位符。因此用户每次使用任何一个探测器采集数据时，与该探测器相关的数组将被最新采集的数据结果覆盖。

如果用户需要保存多个数据采集的数据结果，则可以在每次数据采集后，将所有需要保存的数组元素拷贝到一个不同名称的数组中。

而且，GAUGE 数据采集方式将数据结果保存为相同的数组名称，覆盖先前的数据采集结果（即使这些数据是由 DATA LOGGER 数据采集方式采集的）。

### RANGER

选择 RANGER 的数据采集方式将运行 CBR RANGER 程序，它是一个专为 TI-83 Plus 定制的程序，可以使其与 CBR 兼容。当采集进程被暂停时，CBR RANGER 将从 RAM 中删除。要再次运行 CBR RANGER 程序，请按 [APPS] 并选择 CBL/CBR 应用程序。

**注意：**Ranger 数据采集方式只使用 Sonic 探测器。

1. 按 [APPS] 2 [ENTER]。
2. 选择 3:RANGER 。

```
CBL/CBR APP:
1: GAUGE
2: DATA LOGGER
3: RANGER
4: QUIT
```

3. 按 [ENTER]。

```
TEXAS INSTRUMENTS
:
RANGER (V1.00)
:
PRESS [ENTER]
```

4. 选择选项。

```
MAIN MENU
1: SETUP/SAMPLE
2: SET DEFAULTS
3: APPLICATIONS
4: PLOT MENU
5: TOOLS
6: QUIT
```

关于 RANGER 程序及选项说明的详细信息，请参阅 CBR™ 手册中的“入门”。

### 采集数据

指定数据采集方式的所有选项后，在 GAUGE 或 DATA LOGGER 选项屏幕中选择 Go 选项。如果用户使用 RANGER 数据采集方式，则在 MAIN 菜单中选择 1:SETUP/SAMPLE，然后选择 START NOW。

- 如果 DIRECTNS=Off，则 GAUGE 和 DATA LOGGER 数据采集立即开始。
- 如果 DIRECTNS=On，则计算器显示操作步骤提示。

如果 PROBE=Sonic，则计算器先显示菜单让用户选择 1:CBL 或 2:CBR。这将确保用户获得正确的提示。按 1 指定 CBL，或按 2 指定 CBR。

如果在 RANGER 数据采集方式的 MAIN 菜单中选择 START NOW，则计算器显示一个提示屏幕。按 **ENTER** 开始数据采集。

要停止 GAUGE 数据采集方式，按 TI-83 Plus 上的：[CLEAR]。

在采集指定数量的样本后，DATA LOGGER 和 RANGER 数据采集方式将会终止。要在采样结束前终止数据采集，应：

1. 按 TI-83 Plus 上的 [ON]。
2. 按 CBR 上的 [TRIGGER] 或 CBL 上的 [ON/HALT]。

要在采集未开始时，从 GAUGE 或 DATA LOGGER 选项菜单中退出，请按 [2nd] [QUIT]。

要在采集未开始时，从 RANGER 选项菜单中退出，请选择 MAIN 菜单。选择 6:QUIT 返回到 CBL/CBR APP 菜单。

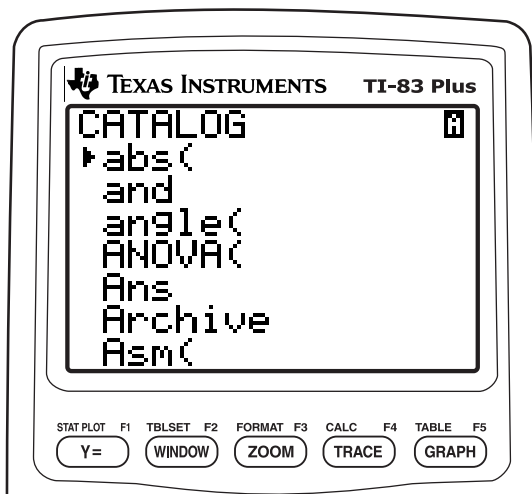
按 CBL/CBR APP 菜单中的 4:QUIT 返回到 TI-83 Plus 主屏幕。



# 15 CATALOG、字符串 和双曲函数

## 目录

浏览 TI-83 Plus CATALOG .....	2
输入和使用字符串 .....	3
将字符串保存到字符串变量中 .....	4
CATALOG 中的字符串函数和指令 .....	6
CATALOG 中的双曲函数 .....	10



## CATALOG 是什么？

CATALOG 是 TI-83 Plus 中按字母顺序排列的所有函数和指令的列表。可以通过菜单或键盘访问 CATALOG 的每一项，但下列函数和指令除外：

- 6 个字符串函数（页 15-6）
- 6 个双曲函数（页 15-10）
- **solve**( 指令（在没有方程求解器编辑器时）（第 2 章）
- 推理统计函数（在没有推理统计编辑器时）（第 13 章）

**注意：**从主屏幕可以执行的 CATALOG 编程命令只有 **GetCalc**(、**Get**( 和 **Send**(。

## 选择 CATALOG 中的某一项

要选择一个 CATALOG 项，请按下列步骤进行：

1. 按 **[2nd]** **[CATALOG]** 显示 CATALOG。



第一列中的 **▶** 是选择光标。

2. 按 **▼** 或 **▲** 滚动 CATALOG，直到选择光标指向要选择的项。
  - 要跳转到以某个字母开始的第一项，按该字母，字母锁开启。
  - 以数字开始的项，根据数字后的第一个字母按字母顺序进行排列。例如，**2-PropZTest**( 在以字母 **P** 开始的项中。
  - 以符号形式出现的函数，如 **+**、**^-1**、**<** 和 **√**(，跟着以字母 **Z** 开始的最后一项。要跳转到第一个符号！，按 **[0]**。
3. 按 **[ENTER]** 将所选的项粘贴到当前屏幕中。



**提示：**按 **▲** 从 CATALOG 菜单的顶端移到底端；按 **▼** 则从底端移到顶端。

## 字符串是什么？

字符串是用引号围起来的一串字符。在 TI-83 Plus 中，字符串有两个主要应用。

- 定义在程序中显示的文本。
- 在程序中接收从键盘的输入。

字符是用来构成字符串的基本单元。

- 数字、字母和空格当作一个字符。
- 指令或函数名，如 **sin(** 或 **cos(**，当作一个字符。TI-83 Plus 将指令或函数名当作一个字符解释。

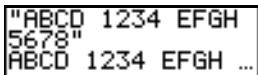
## 输入字符串

按下列步骤，在主屏幕或程序中的空行上输入字符串：

1. 按 **[ALPHA]** **["]** 表示字符串开始。
2. 输入构成字符串的字符。
  - 用数字、字母、函数名或指令名的任何组合创建字符串。
  - 要输入空格，按 **[ALPHA]** **[\_]**。
  - 要在一行中输入几个字母字符，按 **[2nd]** **[A-LOCK]** 激活字母锁。
3. 按 **[ALPHA]** **["]** 表示字符串结束。

*"string"*

4. 按 **[ENTER]**。在主屏幕中，字符串显示在下一行中，并不带引号。省略号 (...) 表示字符串超出了屏幕。要滚动字符串，按 **[▶]** 和 **[◀]**。



```
"ABCD 1234 EFGH
5678"
ABCD 1234 EFGH ...
```

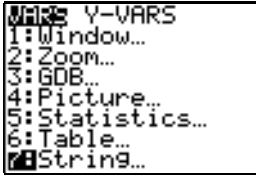
**注意：** 引号不算作字符串字符。

## 字符串变量

TI-83 Plus 有 10 个可保存字符串的变量。字符串变量可以与字符串函数和指令一起使用。

要显示 VARS STRING 菜单，请按下列步骤进行：

1. 按 **[VARS]** 显示 VARS 菜单。将光标移到 **7:String** 上。



2. 按 **[ENTER]** 显示 STRING 子菜单。



## 将字符串保存到字符串变量中

要将字符串保存到字符串变量中，请按下列步骤进行：

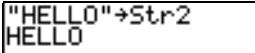
1. 按 **[ALPHA]** **[*i*]**，输入字符串，再按 **[ALPHA]** **[*i*]**。
2. 按 **[STO▶]**。
3. 按 **[VARS]** **7** 显示 VARS STRING 菜单。
4. 选择用来保存字符串的字符串变量（**Str1-Str9** 或 **Str0**）。



```
STRING
1:Str1
2:Str2
3:Str3
4:Str4
5:Str5
6:Str6
7↓Str7
```

所选字符串变量粘贴到当前光标处，靠近保存符 (➔)。

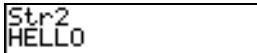
5. 按 **[ENTER]** 将字符串保存到字符串变量中。在主屏幕中，保存的字符串显示在下一行中，并不带引号。



```
"HELLO"➔Str2
HELLO
```

## 显示字符串变量的内容

要在主屏幕中显示字符串变量的内容，先从 VARS STRING 菜单中选择字符串变量，然后按 **[ENTER]**，字符串显示出来。



```
Str2
HELLO
```

## 显示 CATALOG 中的字符串函数和指令

字符串函数和指令只能从 CATALOG 得到。下表列出的字符串函数和指令按它们在 CATALOG 菜单项中出现的顺序排列。表中的省略号表示还有其他 CATALOG 项。

---

### CATALOG

...	
Equ►String(	将方程转换为字符串。
expr(	将字符串转换成表达式。
...	
inString(	返回字符所在串中的位置。
...	
length(	返回字符串的字符长度。
...	
String►Equ(	将字符串转换成方程。
sub(	返回一个字符串的子集作为一个新的字符串。
...	

---

### + (合并)

要合并两个以上的字符串，请按下列步骤进行：

1. 输入 *string1*，它可以是字符串或字符串名。
2. 按  $\oplus$ 。
3. 输入 *string2*，它可以是字符串或字符串名。必要时，按  $\oplus$ ，然后输入 *string3*；依次类推。

*string1+string2+string3...*

4. 按  $\text{ENTER}$  将这些字符串作为单个字符串显示。

```
"HIJK "→Str1:Str  
1+"LMNOP"  
HIJK LMNOP
```

### 从 CATALOG 中选择字符串函数

要选择字符串函数或指令，并将之粘贴到当前屏幕中，请按页 15-2 中的步骤执行。

## EquString(

**EquString(** 将保存在某个 VARS Y-VARS 变量中的方程转换成字符串。**Yn** 包含方程。**Strn** (从 **Str1** 到 **Str9**，或 **Str0**) 是一个字符串变量，方程将当作一个字符串保存在这个变量中。

### EquString(Yn,Strn)

```
"3X"→Y1
Done
EquString(Y1,Str1)
Done
Str1
3X
```

## expr(

**expr(** 将 *string* 中包含的字符串转换成一个表达式，并执行这个表达式。*string* 可以是字符串，也可以是字符串变量。

### expr(string)

```
2→X:"5X"→Str1
5X
expr(Str1)→A
A
10
10
```

```
expr("1+2+X²")
7
```

## inString(

**inString(** 返回子字符串 *substring* 第一个字符在 *string* 中的字符位置。*string* 可以是字符串或字符串变量。*start* 是可选项，表示开始搜索的字符位置，默认值为 1。

### inString(string,substring[,start])

```
inString("PQRSTU", "STU")
4
inString("ABCABC", "ABC", 4)
4
```

**注意：**如果 *string* 中不包含 *substring*，或者 *start* 大于 *string* 的长度，**inString(** 将返回 **0**。

**length(**

**length(** 返回 *string* 中的字符数。 *string* 可以是字符串或字符串变量。

**注意:** 指令或函数名, 如 **sin(** 或 **cos(**, 被当作一个字符。

**length(string)**

```
"WXYZ"→Str1
WXYZ
length(Str1)      4
```

**String→Equ(**

**String→Equ(** 将 *string* 转换成方程, 并将方程保存到 **Yn** 中。 *string* 可以是字符串或字符串变量。 **String→Equ(** 是 **Equ→String(** 的逆过程。

**String→Equ(string, Yn)**

```
"2X"→Str2
2X
String→Equ(Str2,
Y2)
Done
```

```
Plot1 Plot2 Plot3
\Y1=
\Y2=2X
```



## sub(

**sub(** 返回一个字符串，它是现有 *string* 的子串。*string* 可以是字符串或字符串变量。*begin* 表示子串第一个字符的位置序号，*length* 表示子串中的字符数。

**sub(string,begin,length)**

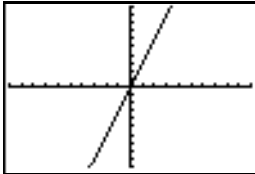
```
"ABCDEFGH"→Str5
ABCDEFGH
sub(Str5,4,2)
DE
```

## 程序执行期间将函数输入到图象中

在程序中，通过使用下述命令，可以在程序执行期间将函数输入到图象中。

```
PROGRAM: INPUT
:Input "ENTRY=",
Str3
:String→Eqw(Str3
,Y3)
:DispGraph
```

```
PrgrmINPUT
ENTRY=3X■
```



**注意：**执行这个程序时，在 **ENTRY=** 提示处输入一个函数，该函数将保存到 **Y3** 中。

## 双曲函数

双曲函数只能从 CATALOG 中得到。下表中列出的双曲函数按它们在 CATALOG 菜单项中出现的顺序排列。表中的省略号表示还有其他 CATALOG 项。

---

CATALOG	
...	
cosh(	双曲余弦
cosh <sup>-1</sup> (	反双曲余弦
...	
sinh(	双曲正弦
sinh <sup>-1</sup> (	反双曲正弦
...	
tanh(	双曲正切
tanh <sup>-1</sup> (	反双曲正切
...	

---

### sinh(、cosh( 和 tanh(

sinh(、cosh( 和 tanh( 是双曲函数。它们对实数、表达式和数组都是有效的。

sinh(*value*)

cosh(*value*)

tanh(*value*)

```
sinh(.5)
      .5210953055
cosh(.25,.5,1)
(1.0314131 1.12...
```

### sinh<sup>-1</sup>(、cosh<sup>-1</sup>( 和 tanh<sup>-1</sup>(

sinh<sup>-1</sup>( 是双曲反正弦函数。cosh<sup>-1</sup>( 是双曲反余弦函数。tanh<sup>-1</sup>( 是双曲反正切函数。它们对实数、表达式和数组都是有效的。

sinh<sup>-1</sup>(*value*)

cosh<sup>-1</sup>(*value*)

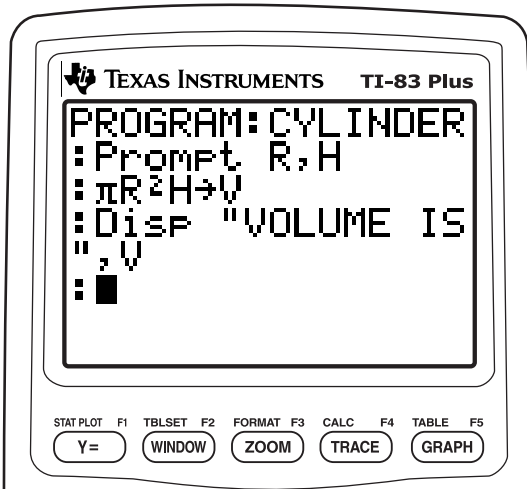
sinh<sup>-1</sup>(*value*)

```
sinh-1(.0,1)
(0 .881373587)
tanh-1(-.5)
      -.5493061443
```

# 16 程序设计

## 目录

入门：圆柱体积.....	1
创建和删除程序.....	4
输入命令行和执行程序.....	6
编辑程序.....	7
拷贝和重命名程序.....	8
PRGM CTL（控制）指令.....	9
PRGM I/O（输入/输出）指令.....	17
将其他程序当作子程序调用.....	23
运行汇编语言程序.....	24



入门是对本章内容的快速介绍。详细信息请阅读本章。

程序是由 TI-83 Plus 按顺序执行的命令组成的集合，这些命令就好像是从键盘输入的一样。试创建一个程序，要求提示输入圆柱的半径  $R$  和高度  $H$ ，然后计算它的体积。

1. 按  $\boxed{\text{PRGM}} \rightarrow \rightarrow$  显示 PRGM NEW 菜单。

```
EXEC EDIT NEW
1:Create New
```

2. 按  $\boxed{\text{ENTER}}$  选择 **1:Create New**。显示出 **Name=** 提示，且字母锁开启。按  $\boxed{\text{C}} \boxed{\text{Y}} \boxed{\text{L}} \boxed{\text{I}} \boxed{\text{N}} \boxed{\text{D}} \boxed{\text{E}} \boxed{\text{R}}$ ，然后按  $\boxed{\text{ENTER}}$  将程序命名为 **CYLINDER**。

现在已进入程序编辑器。第 2 行第 1 列中的冒号 (:) 表示命令行的开始。

```
PROGRAM:CYLINDER
:█
```

3. 按  $\boxed{\text{PRGM}} \rightarrow \mathbf{2}$  从 PRGM I/O 菜单中选择 **2:Prompt**。Prompt 拷贝到命令行上。按  $\boxed{\text{ALPHA}} \boxed{\text{R}} \boxed{\text{,}} \boxed{\text{ALPHA}} \boxed{\text{H}}$  输入半径和高度的变量名。按  $\boxed{\text{ENTER}}$ 。

```
PROGRAM:CYLINDER
:Prompt R,H
:█
```

4. 按  $\boxed{\text{2nd}} \boxed{\pi} \boxed{\text{ALPHA}} \boxed{\text{R}} \boxed{\text{x}^2} \boxed{\text{ALPHA}} \boxed{\text{H}} \boxed{\text{STO}} \rightarrow \boxed{\text{ALPHA}} \boxed{\text{V}} \boxed{\text{ENTER}}$  输入表达式  $\pi R^2 H$ ，并将之保存到变量 **V** 中。

```
PROGRAM:CYLINDER
:Prompt R,H
:πR²H→V
:█
```

- 按 **[PRGM]** **[3]** 从 PRGM I/O 菜单中选择 **3:Disp**。**Disp** 粘贴到命令行上。按 **[2nd]** **[A-LOCK]** **[\*]** **[V]** **[O]** **[L]** **[U]** **[M]** **[E]** **[ $\bar{\_}$ ]** **[I]** **[S]** **[\*]** **[ALPHA]** **[ $\bar{\_}$ ]** **[ALPHA]** **[V]** **[ENTER]** 设置程序使得在某一行上显示文本 **VOLUME IS**，它的下一行上显示 **V** 的计算值。

```
PROGRAM:CYLINDER
:Prompt R,H
: $\pi$ R2H $\rightarrow$ V
:Disp "VOLUME IS
",V
:█
```

- 按 **[2nd]** **[QUIT]** 显示主屏幕。
- 按 **[PRGM]** 显示 PRGM EXEC 菜单。菜单项是已保存的程序名。

```
PRGM EXEC EDIT NEW
CYLINDER
```

- 按 **[ENTER]** 将 **prgmCYLINDER** 粘贴到当前光标处。（如果 **CYLINDER** 不是 PRGM EXEC 菜单中的项 **1**，那么在按 **[ENTER]** 之前必须将光标移到 **CYLINDER** 上。）

```
prgmCYLINDER█
```

- 按 **[ENTER]** 执行这个程序。输入半径为 **1.5**，然后按 **[ENTER]**。输入高度为 **3**，再按 **[ENTER]**。这时显示出文本 **VOLUME IS**、**V** 的值和 **Done**。重复步骤 7 至步骤 9，对 **R** 和 **H** 输入其他值。

```
prgmCYLINDER
R=?1.5
H=?3
VOLUME IS
      21.20575041
      Done
```

## 程序是什么？

程序是由一个或多个命令行组成的集合。每个命令行中包含一条或多条指令。执行程序时，TI-83 Plus 按输入指令的顺序执行命令行中的指令。TI-83 Plus 可存储的程序数量和大小只受可用内存的限制。

## 创建新程序

要创建新程序，请按下列步骤进行：

1. 按 **[PRGM]** **[↓]** 显示 PRGM NEW 菜单。

```
EXEC EDIT NEW
1:Create New
```

2. 按 **[ENTER]** 选择 **1:Create New**。这时，显示出 **Name=** 提示，字母锁开启。
3. 按字母 A-Z 或  $\theta$  输入新程序名的第 1 个字符。

**注意：**程序名的长度为 1-8 个字符。第 1 个字符必须是字母 A-Z 或  $\theta$ ，第 2 到第 8 个字符可以是字母、数字或  $\theta$ 。

4. 输入 0-7 个字母、数字或  $\theta$  组成新程序名。
5. 按 **[ENTER]**。显示出程序编辑器。
6. 输入一个或多个程序命令（页 16-6）。
7. 按 **[2nd]** **[QUIT]** 退出程序编辑器，并返回到主屏幕中。

## 管理内存和删除程序

要检查是否有足够的内存可用于要输入的程序，请：

1. 按 **[2nd]** **[MEM]** 显示 MEMORY 菜单。
2. 选择 **2:Mem Mgmt/Del** 以显示 MEMORY MANAGEMENT/DELETE 菜单（第 18 章）。
3. 选择 **7:Prgm** 以显示 PRGM 编辑器。

```
RAM FREE    24790
ARC FREE    130832
*PROGRAM1   20
▶ PROGRAM2   22
```

TI-83 Plus 用字节来表示内存数量。

用户可以使用两种方法之一来增加可用内存。可以删除一个或多个程序，也可以将一些程序归档。

要通过删除特定程序来增加可用内存，请：

1. 按 **2nd** [MEM]，然后从 MEMORY 菜单中选择 **2:Mem Mgmt/Del**。

```
MEMORY
1>About
2:Mem Mgmt/Del...
3:Clear Entries
4:ClrAllLists
5:Archive
6:UnArchive
7↓Reset...
```

2. 选择 **7:Prgm** 显示 PRGM 编辑器（第 18 章）。

```
RAM FREE 24790
ARC FREE 130832
*PROGRAM1 20
▶ PROGRAM2 22
```

3. 按 **▴** 和 **▾** 可以将选择光标 (**▶**) 移至要删除的程序的旁边，然后按 **DEL** 键将程序从内存中删除。

**注意：**用户将收到一则消息，请用户确认删除操作。选择 **2:yes** 可以继续。

要退出 PRGM 编辑器屏幕而不删除任何项目，请按 **2nd** [QUIT] 键显示主屏幕。

要通过归档程序来增加可用内存，请：

1. 按 **2nd** [MEM]，然后从 MEMORY 菜单中选择 **2:Mem Mgmt/Del**。
2. 选择 **2:Mem Mgmt/Del** 显示 MEM MGMT/DEL 菜单。
3. 选择 **7:Prgm...** 显示 PRGM 菜单。

```
RAM FREE 24790
ARC FREE 130832
▶*PROGRAM1 20
PROGRAM2 22
```

4. 按 **ENTER** 归档程序。在该程序的左边会出现一个星号，它表示这是一个已归档的程序。

要在此屏幕中取消归档程序，请将光标移至已归档程序的旁边，然后按 **ENTER** 键。这样星号就会消失。

**注意：**不能编辑或执行已归档程序。要编辑或执行一个已归档程序，用户首先必须对其取消归档。

## 输入程序命令行

可以在命令行中输入任何能从主屏幕中执行的指令或表达式。在程序编辑器中，新的命令行都以冒号开始。在单个命令行中输入多个指令或表达式时，用冒号将它们隔开。

**注意：**命令行的长度可以大于编辑器屏幕的宽度；长的命令行会自动转换到下一行。

即使在程序编辑器中，也可以显示和选择菜单。通过下述两种方法之一可以从菜单返回到程序编辑器中。

- 选择一个菜单项，将把此项粘贴到当前命令行中。
- 按 **CLEAR**。

当输入完命令行时，按 **ENTER**。光标移到下一命令行中。

程序可以访问保存在内存中的变量、数组、矩阵和字符串。如果程序将新的值保存到某个变量、数组、矩阵或字符串中，那么它在执行期间将改变内存中的该值。

可以作为子程序调用另一个程序（页 16-16 和页 16-23）。

## 执行程序

要执行程序，由主屏幕中某个空行上开始，然后执行下列步骤：

1. 按 **PRGM** 显示 PRGM EXEC 菜单。
2. 从 PRGM EXEC 菜单（页 16-8）中选择某个程序名。 **prgmname** 粘贴到主屏幕中（例如：**prgmCYLINDER**）。
3. 按 **ENTER** 执行该程序。程序执行期间，忙指示器开启。

程序执行期间，更新最后结果 (**Ans**)，但在执行每个命令时，不更新最后输入项（第 1 章）。

TI-83 Plus 在执行程序时检查错误，而输入程序时并不检查错误。

## 中断程序

如要中断程序运行，则按 **ON**。显示出 ERR: BREAK 菜单。

- 选择 **1:Quit**，返回到主屏幕中。
- 选择 **2:Goto**，返回到中断处。



## 编辑程序

要编辑已保存的程序，请按下列步骤进行：

1. 按 **[PRGM]** **[▶]** 显示 PRGM EDIT 菜单。
2. 从 PRGM EDIT 菜单（页 16-8）中选择程序名。至多显示程序的前 7 行。

**注意：**程序编辑器不显示 ↓ 以指示程序已超出了屏幕。

3. 编辑程序的命令行。

- 将光标移到适当的位置，然后删除、改写或插入。
- 按 **[CLEAR]** 清除命令行中所有程序命令（前导冒号保留），然后输入新的程序命令。

**提示：**按 **[2nd]** **[◀]**，光标移到命令行的起始处；按 **[2nd]** **[▶]**，则移到末尾。按 **[ALPHA]** **[◀]**，光标向下滚动 7 个命令行；按 **[ALPHA]** **[▶]**，则向上滚动 7 个命令行。

## 插入和删除命令行

如要在程序中某处插入新的命令行，则将光标移至要插入新行处，按 **[2nd]** **[INS]**，然后按 **[ENTER]**。冒号表示新的一行。

如要删除某个命令行，则将光标移到这行上，按 **[CLEAR]** 清除行中所有指令和表达式，然后按 **[DEL]** 删除包括冒号在内的命令行。

## 拷贝和重命名程序

要将一个程序的所有命令行拷贝到另一个程序中，先执行“创建新程序”（页 16-4）中的步骤 1 至步骤 5，然后执行下列步骤：

1. 按 **[2nd] [RCL]**。在新程序中程序编辑器的底行上显示 **Rcl**（第 1 章）。
2. 按 **[PRGM] [↓]** 显示 PRGM EXEC 菜单。
3. 从菜单中选择一个程序名。 **prgmname** 粘贴到该程序编辑器的底行。
4. 按 **[ENTER]**。所选程序的全部命令行拷贝到这个新的程序中。

拷贝程序至少有两种方便的用途。

- 可以为常用的指令组创建模板。
- 可以通过将某个程序的内容拷贝到一个新的程序中，重新给它命名。

**注意：**也可以通过使用 RCL，将一个现有程序的全部命令行拷贝到另一个现有程序中。

## 滚动 PRGM EXEC（执行程序）和 PRGM EDIT（编辑程序）菜单

TI-83 Plus 自动地按字母顺序排列 PRGM EXEC 和 PRGM EDIT 菜单中的菜单项。每个菜单只用 **1-9** 和 **0** 标记它的前 10 项。

要跳转到以某个特定字母字符或 **θ** 开始的第一个程序名，按 **[ALPHA] [字母 A-Z 或 θ]**。

**提示：**对于 PRGM EXEC 和 PRGM EDIT 菜单，从菜单顶部按 **[↓]** 移到底部；从菜单底部按 **[↑]** 则移到顶部。按 **[ALPHA] [↓]** 向下滚动 7 个菜单项；按 **[ALPHA] [↑]** 向上滚动 7 个菜单项。

## PRGM CTL 菜单

要显示 PRGM CTL (程序控制) 菜单, 只能从程序编辑器中按 **[PRGM]**。

---

**CTL** I/O EXEC

1: If	创建条件测试。
2: Then	当 <b>If</b> 为真时执行命令。
3: Else	当 <b>If</b> 为假时执行命令。
4: For(	创建增量循环。
5: While	创建条件循环。
6: Repeat	创建条件循环。
7: End	表示块结束。
8: Pause	暂停程序执行。
9: Lbl	定义标记。
0: Goto	跳转到标记。
A: IS>(	增 1, 如果大于则跳过。
B: DS<(	减 1, 如果小于则跳过。
C: Menu(	定义菜单项和分支。
D: prgm	作为子程序来执行程序。
E: Return	从子程序返回。
F: Stop	停止执行。
G: DelVar	在程序内删除变量。
H: GraphStyle(	指定绘制图象的样式。

---

这些菜单项控制执行程序的流向, 使程序执行时重复或跳过一组命令变得十分容易。从菜单中选择某一项时, 它的名字就粘贴到程序命令行中的光标处。

要在不选择菜单项时返回到程序编辑器中, 按 **[CLEAR]**。

## 控制程序流向

程序控制指令告诉 TI-83 Plus 下一步执行程序中的哪条命令。 **If**、**While** 和 **Repeat** 检查已定义的条件以确定下一步执行哪条命令。这些条件经常使用关系或布尔测试 (第 2 章), 就象下面的例子一样:

**If A<7:A+1→A**

或

**If N=1 and M=1:Goto Z**

If

使用 **If** 进行测试，并产生分支。如果 *condition* (条件) 为假 (0)，则跳过紧跟 **If** 的 *command* (命令)；如果 *condition* (条件) 为真 (非零)，则执行下一条命令。**If** 指令可以嵌套。

```
:if condition
:command (如果为真)
:command
```

程序

```
PROGRAM: COUNT
:0→A
:Lbl Z
:A+1→A
:Disp "A IS",A
:If A≥2
:Stop
:Goto Z
```

输出

```
Pr-gmCOUNT
A IS 1
A IS 2
Done
```

If-Then

**If** 后面的 **Then** 在 *condition* (条件) 为真 (非零) 时执行一组 *commands* (命令)。**End** 标志这组 *commands* (命令) 的结尾。

```
:if condition
:Then
:command (如果为真)
:command (如果为真)
:End
:command
```

程序

```
PROGRAM: TEST
:1→X:10→Y
:If X<10
:Then
:2X+3→X
:2Y-3→Y
:End
:Disp X,Y
```

输出

```
Pr-gmTEST
5
17
Done
```

## If-Then-Else

跟在 **If-Then** 后面的 **Else** 在 *condition* (条件) 为假 (0) 时执行一组 *commands* (命令)。**End** 标志这组 *commands* (命令) 的结尾。

```
:if condition
:Then
:command (如果为真)
:command (如果为真)
:Else
:command (如果为假)
:command (如果为假)
:End
:command
```

### 程序

```
PROGRAM:TESTELSE
:Input "X=",X
:If X<0
:Then
: X2→Y
:Else
: X→Y
:End
```

```
:Disp C(X,Y)
```

### 输出

```
Pr9mTESTELSE
X=5          (5 5)
             Done
X=-5        (-5 25)
             Done
```

## For(

**For(** 循环并递增。它使 *variable* (变量) 从 *begin* (初值) 开始, 循环一次递增 *increment* (增量), 直到 *end* (终值)。*increment* (增量) 是可选项 (默认值为 1), 并且可以是负数 (*end<begin*)。*end* 是不能超过的最大值或最小值。**End** 表示循环的结束。**For(** 循环可以嵌套。

```
:For(variable,begin,end[,increment])
:command (当不超过 end 时)
:command (当不超过 end 时)
:End
:command
```

### 程序

```
PROGRAM:SQUARE
:For(A,0,8,2)
:Disp A2
:End
```

### 输出

```
Pr9mSQUARE
0
4
16
36
64
Done
```

While

**While** 当 *condition* (条件) 为真时执行一组 *commands* (命令)。  
*condition* (条件) 通常为关系测试 (第 2 章), 当遇到 **While** 时才进行测试。  
 如果 *condition* (条件) 为真 (非零), 程序将执行这组 *commands* (命令)。  
**End** 表示命令组的结束。当 *condition* (条件) 为假 (0) 时, 程序执行 **End**  
 下面的 *commands* (命令)。**While** 指令可以嵌套。

```
:While condition
:command (当 condition 为真时)
:command (当 condition 为真时)
:End
:command
```

程序

```
PROGRAM: LOOP
:0→I
:0→J
:While I<6
:J+1→J
:I+1→I
:End
:Disp "J=", J
```

输出

```
Pr-gmLOOP
J=
6
Done
```

Repeat

**Repeat** 重复执行一组 *commands* (命令) 直到 *condition* (条件) 为真 (非零)。它类似于 **While**, 但在遇到 **End** 时才测试 *condition* (条件)。因此, 这组 *commands* (命令) 至少执行一次。**Repeat** 指令可以嵌套。

```
:Repeat condition
:command (直到 condition 为真)
:command (直到 condition 为真)
:End
:command
```

程序

```
PROGRAM: RLOOP
:0→I
:0→J
:Repeat I≥6
:J+1→J
:I+1→I
:End
:Disp "J=", J
```

输出

```
Pr-gmRLOOP
J=
6
Done
```

## End

**End** 标志一组 *commands* (命令) 结束。 **For**、**While** 或 **Repeat** 循环指令必须在结尾处包含 **End** 指令； **If-Then** 指令和 **If-Then-Else** 指令也必须与 **End** 指令配对使用。

## Pause

**Pause** 暂停执行程序以便可以查看结果或图形。暂停期间，右上角显示暂停指示符。按 **[ENTER]** 继续执行程序。

- **Pause** 指令不带参数 *value* 时暂停执行程序。如果已执行 **DispGraph** 或 **Disp** 指令，则显示相应屏幕。
- **Pause** 指令带参数 *value* 时，当前主屏幕中显示 *value*。 *value* 可以滚动。

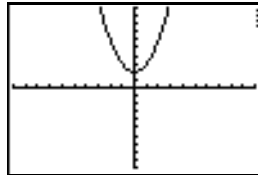
**Pause** [*value*]

### 程序

```
PROGRAM: PAUSE
:10→X
:"X2+2"→Y1
:Disp "X=",X
:Pause
:DispGraph
:Pause
:Disp
```

### 输出

```
PrgmPAUSE
X= 10
```



```
PrgmPAUSE
X= 10
Done
```

Lbl、Goto

Lbl (标记) 和 Goto (跳转) 指令一起来产生程序分支。

Lbl 为命令指定标记 *label*。*label* 可以是 1-2 个字符 (A-Z、0-99 或 0)。

Lbl *label*

Goto 执行后程序跳转到 *label*。

Goto *label*

程序	输出
PROGRAM: CUBE :Lbl 99 :Input A :If A≥100 :Stop :Disp A³ :Pause :Goto 99	Pr9mCUBE ?2 8 ?3 27 ?105 Done

IS>(

IS>( (增 1 并跳过) 将变量 *variable* 加 1。如果结果大于 *value* (可以是表达式), 则跳过下一条命令 *command*; 如果结果 ≤ *value*, 则执行下一条命令 *command*。*variable* 不能是系统变量。

:IS>( *variable,value* )

:*command* (如果结果 ≤ *value* )

:*command* (如果结果 > *value* )

程序	输出
PROGRAM: ISKIP :7→A :IS>(A,6) :Disp "NOT > 6" :Disp "> 6"	Pr9mISKIP > 6 Done

注意: IS>( 不是一条循环指令。



## DS<(

**DS<(** (减 1 并跳过) 将变量 *variable* 减 1。如果结果  $< value$  (可以是表达式), 则跳过下一条命令 *command*; 如果结果  $\geq value$ , 则执行下一条命令 *command*。 *variable* 不能是系统变量。

```
:DS<(variable,value)  
:command (如果结果  $\geq value$ )  
:command (如果结果  $< value$ )
```

### 程序

```
PROGRAM: DSKIP  
:1→A  
:DS<(A,6)  
:DISP "> 6"  
:DISP "NOT > 6"
```

### 输出

```
Pr:gmDSKIP  
NOT > 6  
Done
```

**注意:** **DS<(** 不是一条循环指令。

## Menu(

**Menu(** 在程序中创建分支。如果在程序执行期间遇到 **Menu(**, 则显示带有指定菜单项的菜单屏幕, 暂停指示器开启。选择某个菜单项后程序继续执行。

菜单 *title* (标题) 用引号 (") 括起来, 之后至多有 7 对菜单项。每对菜单项由 *text* (文本) 项 (也用引号括起来) 和 *label* (标记) 项组成。文本项作为菜单选项显示; 标记项表示选择对应的菜单选项后程序将转移的去处。

```
Menu("title","text1",label1,"text2",label2,...)
```

### 程序

```
PROGRAM: TOSSDICE  
:Menu("TOSS DICE"  
,"FAIR DICE",A,  
"WEIGHTED DICE",  
B)
```

### 输出

```
TOSS DICE  
1:FAIR DICE  
2:WEIGHTED DICE
```

上面的程序暂停执行, 直到选择 1 或 2。例如, 如果选择 2, 菜单消失, 程序从 **Lbl B** 处继续执行。

## prgm

使用 **prgm** 可将其他程序作为子程序 (页 16-23) 执行。选择 **prgm** 后, 它将粘贴到光标处。输入字符以表示程序 *name* (名)。使用 **prgm** 等同于从 PRGM EXEC 菜单中选择现有程序, 不过, 它允许输入还未创建的程序的名称。

### prgmname

**注意:** 使用 RCL 时不能直接输入子程序名, 必须从 PRGM EXEC 菜单 (页 16-8) 中粘贴它。

## Return

**Return** 退出子程序, 并返回到调用程序 (页 16-23)。即使在嵌套循环中遇到该指令, 也将执行上述操作, 循环指令终止。被当作子程序调用的程序结尾处隐含着一条 **Return** 指令。在主程序中, **Return** 停止程序执行, 并返回到主屏幕中。

## Stop

**Stop** 终止程序执行, 并返回到主屏幕中。在程序结尾处 **Stop** 是可选的。

## DelVar

**DelVar** 从内存中删除 *variable* (变量) 的内容。

### DelVar variable


```
PROGRAM:DELMATR
:DelVar [A]■
```

## GraphStyle(

**GraphStyle**( 指定绘制图象的样式。 *function#* 是当前绘图方式中 Y= 函数名的序号。 *graphstyle* 是一个 1-7 的数, 这些数对应着图象样式。如下所示:

- |   |  |
|---|--|
| 1 =  (实线)  | 5 =  (轨迹) |
| 2 =  (粗线)  | 6 =  (动画) |
| 3 =  (上阴影) | 7 =  (点)  |
| 4 =  (下阴影) |  |

### GraphStyle(function#,graphstyle)

例如, 在 **Func** 方式下, **GraphStyle(1,5)** 将 Y1 的图象样式设为  (轨迹; 5)。

在所有作图方式下, 不是所有的图象样式都可使用。关于每种图象样式的详细说明, 请查阅第 3 章中的图象样式表。

## PRGM I/O 菜单

要显示 PRGM I/O (程序输入/输出) 菜单, 只能在程序编辑器中按 **[PRGM]** **[▶]**。

CTL **I/O** EXEC

1: Input	输入数值或使用光标。
2: Prompt	提示输入变量值。
3: Disp	显示文本、数值或主屏幕。
4: DispGraph	显示当前图象。
5: DispTable	显示当前运算表。
6: Output(	在指定位置显示文本。
7: getKey	检查键盘以获取击键。
8: ClrHome	清除显示。
9: ClrTable	清除当前运算表。
0: GetCalc(	从另一台 TI-83 Plus 获取变量。
A: Get(	从 CBL 或 CBR 获取变量。
B: Send(	将变量发送到 CBL 或 CBR。

这些指令在程序执行期间控制程序的输入和输出。它们允许在程序执行期间输入数值和显示结果。

要在不选择菜单项时返回到程序编辑器中, 按 **[CLEAR]**。

## 用 Input 显示图象

不带变量的 **Input** 显示当前图象。可以移动自由移动光标, **X** 和 **Y** ( **PolarGC** 格式的 **R** 和  $\theta$  ) 的值随光标的移动而变化。暂停指示器开启。按 **[ENTER]** 恢复程序执行。

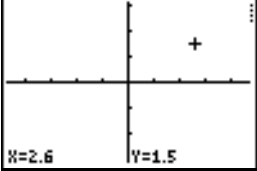
### Input

#### 程序

```
PROGRAM:GINPUT
:FnOff
:ZDecimal
:Input
:Disp X,Y
```

#### 输出

```
PrgmGINPUT
```



```
X=2.6 Y=1.5
```

```
PrgmGINPUT
2.6
1.5
Done
```

用 Input 保存变量值

带 *variable* (变量) 的 **Input** 在执行时显示提示 ? (问号)。 *variable* (变量) 可以是实数、复数、数组、矩阵、字符串或  $Y=$  函数。在程序执行时, 输入一个值, 也可以是表达式, 然后按 **[ENTER]**。该值经计算后保存到变量 *variable* 中, 程序继续执行。

**Input** [*variable*]

可以将文本 *text* 或 **Strn** (字符串变量) 的内容作为提示显示, 文本或字符串内容至多为 16 个字符。程序执行时, 在提示后输入数值, 然后按 **[ENTER]**。输入的值保存到变量 *variable* 中, 程序继续执行。

**Input** ["*text*",*variable*]

**Input** [**Strn**,*variable*]

程序

```
PROGRAM:HINPUT
:Input A
:Input L1
:Input "Y1=",Y1
:Input "DATA=",L
DATA
:Disp Y1(A)
:Disp Y1(L1)

:Disp Y1(LDATA)
```

输出

```
PrgrmHINPUT
?2
?{1,2,3}
Y1="2X+2"
DATA={4,5,6}
      6
      (4 6 8)
      (10 12 14)
      Done
```

**注意:** 当程序执行期间提示输入数组和  $Yn$  函数时, 必须将数组元素放在大括号 ({} ) 中, 将表达式放在引号 (" ) 内。

## Prompt

在程序执行期间，**Prompt** 显示每个 *variable*（变量），一次显示一个，*variable* 之后显示 **=?**。在提示处，输入 *variable* 的值或表达式，然后按 **[ENTER]**。保存值后，程序继续执行。

**Prompt** *variableA* [*variableB*, ..., *variable n*]

### 程序

```
PROGRAM: WINDOW
: Prompt Xmin
: Prompt Xmax
: Prompt Ymin
: Prompt Ymax
```

### 输出

```
Pr-9mWINDOW
Xmin=?-10
Xmax=?10
Ymin=?-3
Ymax=?3
Done
```

**注意：** **Y=** 函数对 **Prompt** 是无效的。

## 显示主屏幕

**Disp**（显示）在不带值时显示主屏幕。如要在程序执行期间查看主屏幕，则在 **Disp** 指令后加上一条 **Pause** 指令。

### Disp

## 显示数值和信息

**Disp** 带一个或多个 *values* 时显示每个 *value* 的值。

**Disp** [*valueA*, *valueB*, *valueC*, ..., *value n*]

- 如果 *value* 是一个变量，则显示它的当前值。
- 如果 *value* 是一个表达式，则先计算表达式，然后在下一行的右侧显示计算结果。
- 如果 *value* 是在引号中的文本，则在当前显示行的左侧显示它。→ 在文本中是无效的。

### 程序

```
PROGRAM:A
:Disp "THE ANSWER
R IS ",π/2
```

### 输出

```
Pr-9mA
THE ANSWER IS
1.570796327
Done
```

如果在 **Disp** 之后遇到 **Pause**，程序将暂停执行，这样就可以检查屏幕。要恢复程序执行，按 **[ENTER]**。

**注意：** 如果矩阵或数组太大以至于不能全部显示，在最后一列中将显示省略号 (...)，但是矩阵或数组不能滚动。要滚动，可使用 **Pause value**（页 16-13）。

## DispGraph

**DispGraph** (显示图象) 显示当前图象。如果在 **DispGraph** 之后遇到 **Pause**，程序将暂停执行，这样就可以检查屏幕。按 **ENTER** 恢复程序执行。

## DispTable

**DispTable** (显示运算表) 显示当前运算表。程序暂停执行，这样就可以检查屏幕。按 **ENTER** 恢复程序执行。

## Output(

**Output(** 在当前主屏幕内从行 *row* (1-8) 和列 *column* (1-16) 处开始显示 *text* 或 *value*，覆盖原有的字符。

**提示：**可能要把指令 **ClrHome** (页 16-21) 放在指令 **Output(** 之前。

表达式经计算后，其结果按当前方式设置显示。矩阵按输入格式显示，并自动转换到下一行。➔ 在文本中是无效的。

**Output(row,column,"text")**

**Output(row,column,value)**

程序

```
PROGRAM: OUTPUT
:3+5→B
:ClrHome
:Output(5,4,"ANS
WER:"
:Output(5,12,B)
```

输出

```
ANSWER: 8
```

在 **Horiz** 拆分屏幕中使用 **Output(** 时，*row* 的最大值是 4。

## getKey

**getKey** 根据下面的键码图，返回一个与最后按键相对应的数。如果没有按键，**getKey** 返回 0。在循环体中使用 **getKey** 可转移控制，例如，创建视频游戏时。

### 程序

```
PROGRAM:GETKEY
:While 1
:getKey→K
:While K=0
:getKey→K
:End
:Disp K
:If K=105
:Stop
:End
```

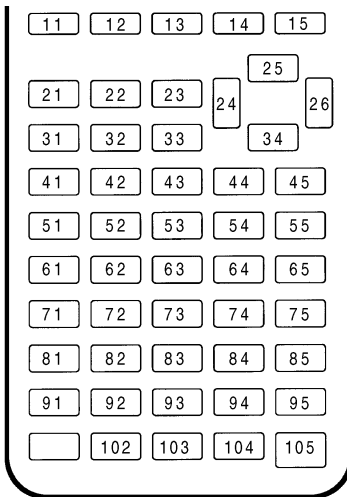
### 输出

```
PrgmGETKEY
41
42
43
105
Done
```

注意：在程序执行期间已按过 **MATH**、**MATRIX**、**PRGM** 和 **ENTER**。

注意：可以在程序执行期间随时按 **ON** 来中断程序执行（页 16-6）。

## TI-83 Plus 键码图



## ClrHome、ClrTable

**ClrHome**（清除主屏幕）在程序执行期间清除主屏幕。

**ClrTable**（清除运算表）在程序执行期间清除运算表中的值。

**GetCalc(**

**GetCalc(** 获取另一台 TI-83 Plus 中变量 *variable* 的内容, 并将之保存到接收 TI-83 Plus 的变量 *variable* 中。 *variable* 可以是实数或复数、数组元素、数组名、矩阵元素、矩阵名、字符串、Y= 变量、图象数据库或图形。

**GetCalc(variable)**

**注意:** **GetCalc(** 在 TI-82 和 TI-83 Plus 计算器连接时不能工作。

**Get( , Send(**

**Get(** 从 Calculator-Based Laboratory™ (CBL™) 系统或 Calculator-Based Ranger™ (CBR™) 中获取数据, 并将之保存到接收 TI-83 Plus 的变量 *variable* 中。 *variable* 可以是实数、数组元素、数组名、矩阵元素、矩阵名、字符串、Y= 变量、图象数据库或图形。

**Get(variable)**

**注意:** 如果将 TI-82 中一个引用 **Get(** 命令的程序传送到 TI-83 Plus 中, TI-83 Plus 则把它解释成上述 **Get(**。使用 **GetCalc(** 从另一台 TI-83 Plus 获取数据。

**Send(** 将变量 *variable* 的内容发送到 CBL 或 CBR。不能用它来把 *variable* 的内容发送到另一台 TI-83 Plus 中。 *variable* 可以是实数、数组元素、数组名、矩阵元素、矩阵名、字符串、Y= 变量、图象数据库或图形。 *variable* 可以是一个包含多个元素的数组。

**Send(variable)**

```
PROGRAM:GETSOUND
:Send(3,00025,
99,1,0,0,0,0,1)

:Get(L1)
:Get(L2)
```

**注意:** 该程序从 CBL 获取声音数据和以秒表示的时间。

**注意:** 可以从 CATALOG 中选取 **Get(**、**Send(** 和 **GetCalc(**，以在主屏幕中执行它们 (第 15 章)。



## 从程序中调用另一个程序

在 TI-83 Plus 中，任何存储的程序都可以当作子程序被另一个程序调用。在某行中单独输入作为子程序使用的程序名。

可以通过下述两种方法之一在命令行上输入程序名。

- 按 **[PRGM]** **[◀]** 显示 PRGM EXEC 菜单，选择程序名（页 16-8）。*prgmname* 粘贴到命令行中当前光标处。
- 从 PRGM CTL 菜单选择 **prgm**，然后输入程序名（页 16-9）。

### *prgmname*

在程序执行期间遇到 *prgmname* 时，程序执行的下一条命令是第 2 个程序的第 1 条命令。当遇到第 2 个程序结尾处的 **Return** 或隐含的 **Return** 时，都返回到第 1 个程序的下一条命令。

#### 程序

```
PROGRAM:VOLCYL
:Input "D=",D
:Input "H=",H
:PrmAREACIR
:A*H→V
:Disp V
```

#### 输出

```
→ PrmVOLCYL
D=4
H=5
62.83185307
Done
```

#### 子程序 ↓ ↑

```
PROGRAM:AREACIR
:D/2→R
:π*R²→A
:Return
```

## 关于程序调用的说明

所有变量都是全局的。

与 **Goto** 和 **Lbl** 一起使用的 *label*（标记）对它所在的程序来说是局部的。一个程序不识别另一个程序中的 *label*（标记）。不能使用 **Goto** 转移到另一个程序中的 *label*（标记）。

**Return** 退出子程序，并返回到调用程序。即使在嵌套的循环中遇到 **Return**，也将如此。

可以运行为 TI-83 Plus 所写的汇编语言程序。一般来说，汇编语言程序比那些使用内置程序编辑器所写的击键程序运行得更快，并且能提供更强的控制功能。

**注意：**由于汇编语言程序对计算器具有强大的控制功能，所以如果用户的汇编语言程序发生错误，那么就可能导致重设计算器并丢失存储在内存中的所有数据、程序和应用程序。

当下载汇编语言程序时，它会作为一个 PRGM 菜单项目和其他程序存储在一起。用户可以：

- 使用 TI-83 Plus 通信链接传送该程序（第 19 章）。
- 使用 MEM MGMT DEL 屏幕删除该程序（第 18 章）。

运行汇编语言程序的语法为：**Asm**(*assemblyprgmname*)

如果用户编写一个汇编语言程序，请从 CATALOG 中使用下列两个指令来标识和编译该程序。

指令	注释
<b>AsmComp</b> ( <i>prgmASM1</i> , <i>prgmASM2</i> )	编译以 ASCII 码编写的汇编语言程序并以十六进制格式进行存储
<b>AsmPrgm</b>	标识汇编语言程序；它必须作为汇编语言程序的第一行进行输入。

要编译一个用户编写的汇编语言程序，请：

1. 按编写程序的步骤（页 16-4）来编写程序，但必须确保在程序的第一行中包含 **AsmPrgm**。
2. 在主屏幕中，按下 **[2nd]** [CATALOG] 键并选择 **AsmComp**( 以便将其粘贴至屏幕。
3. 按 **[PRGM]** 键以显示 PRGM EXEC 菜单。
4. 选择想要编译的程序。它将被粘贴至主屏幕。
5. 按 **[□]** 键并在 CATALOG 中选择 **prgm**。
6. 输入用户为输出程序所选择的名称。

**注意：**该名称必须唯一，不能和已有程序重名。

7. 按 **[□]** 键以完成排序。

参数顺序应如下所示：

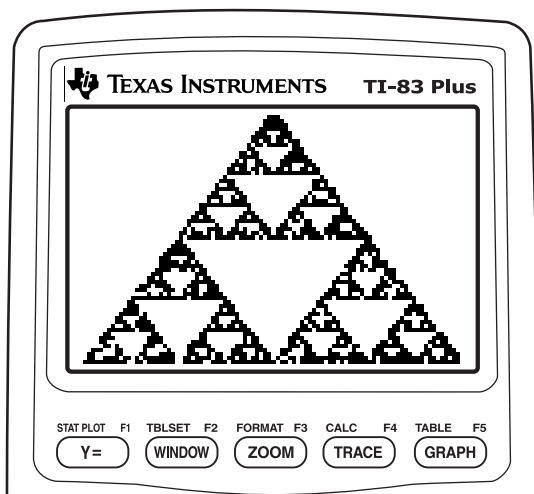
**AsmComp**(*prgmASM1*, *prgmASM2*)

8. 按 **[ENTER]** 键来编译程序并生成输出程序。

# 17 实例

## 目录

使用框图比较实验结果 .....	2
绘制分段函数 .....	4
绘制不等式 .....	5
求解非线性方程组 .....	6
使用程序创建 Sierpinski 三角形 .....	7
绘制蛛网吸引点 .....	8
使用程序估计系数 .....	9
绘制单位圆和三角曲线 .....	10
求曲线间面积 .....	11
使用参数方程：转轮问题 .....	12
演示微积分基本原理 .....	14
计算正 N 边形的面积 .....	16
计算和绘图抵押贷款支付 .....	18



## 问题

实验表明，在识别左手（由右脑控制）和右手（由左脑控制）上所持物体的能力方面，男孩与女孩之间存在显著差异。TI 图象小组对成年男子和女子作了类似的实验。

实验使用 30 个小物体，事先不让参加者看见。首先，让他们用左手逐个拿起其中的 15 个，猜测它们是什么；然后让他们用右手逐个拿起另外 15 个，猜测它们是什么。下表是这次实验猜对次数的数据，现在，使用框图来直观地比较这些数据。

猜对次数

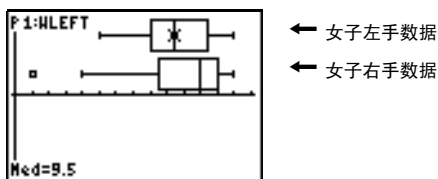
女子左手	女子右手	男子左手	男子右手
8	4	7	12
9	1	8	6
12	8	7	12
11	12	5	12
10	11	7	7
8	11	8	11
12	13	11	12
7	12	4	8
9	11	10	12
11	12	14	11
		13	9
		5	9

## 步骤

- 按 **[STAT]** 5 选择 **5:SetUpEditor**。输入数组名 **WLEFT**、**WRGHT**、**MLEFT** 和 **MRGHT**，它们之间用逗号分开。按 **[ENTER]**。这时统计数组编辑器中仅包括这四个数组。
- 按 **[STAT]** 1 选择 **1:Edit**。
- 在数组 **WLEFT** 中输入每个女子左手猜对的次数（女子左手）。按 **[▶]** 移到 **WRGHT**，输入每个女子右手猜对的次数（女子右手）。
- 与步骤 3 相似，在 **MLEFT**（男子左手）和 **MRGHT**（男子右手）中分别输入每个男子左、右手的猜对次数。
- 按 **[2nd]** **[STAT PLOT]**。选择 **1:Plot1**。打开 plot 1，将它定义成改进框图 **[F5]**，使用 **WLEFT** 中的数据。将光标移到顶行，选择 **Plot2**。打开 plot 2，将它定义成改进框图，使用 **WRGHT** 中的数据。

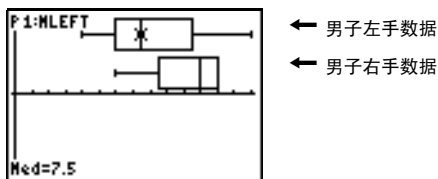
## 步骤 (续)

- 按 **[Y=]**。关闭所有函数。
- 按 **[WINDOW]**。设置 **Xscl=1** 和 **Yscl=0**。按 **[ZOOM] 9** 选择 **9:ZoomStat**，来调整观察窗口，并显示表示女子结果的框图。
- 按 **[TRACE]**。



使用 **[◀]** 和 **[▶]** 对每一个图检查其 **minX**、**Q1**、**Med**、**Q3** 和 **maxX**。请注意女子右手数据的外部点。左手的中位数是多少？右手的呢？根据这两个框图，女子的哪只手猜得更准确些呢？

- 检查男子的结果。用 **MLEFT** 重新定义 plot 1；用 **MRGHT** 重新定义 plot 2。按 **[TRACE]**。



按 **[◀]** 和 **[▶]** 对每个图检查其 **minX**、**Q1**、**Med**、**Q3** 和 **maxX**。您看出两个框图之间有什么差别吗？

- 比较左手的结果。用 **WLEFT** 重新定义 plot 1；用 **MLEFT** 重新定义 plot 2；然后按 **[TRACE]** 对每个图检查其 **minX**、**Q1**、**Med**、**Q3** 和 **maxX**。在左手的猜测能力上，是男子还是女子强呢？
- 比较右手的结果。用 **WRGHT** 定义 plot 1；用 **MRGHT** 定义 plot 2，然后按 **[TRACE]** 检查每个图中的 **minX**、**Q1**、**Med**、**Q3** 和 **maxX**。谁的右手猜得更准确呢？

最初的实验发现，男孩右手的猜测能力较差；而女孩左右手的猜测能力相同。但上述框图所示成年人的结果不是这样。这是因为成年人已学会调整，还是因为样本数还不够大呢？

## 问题

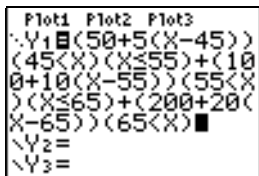
某条道路的汽车时速限制是 45 公里/小时，超速罚款起价 50 元。时速为 46 至 55 公里/小时，每公里加收 5 元；56 至 65 公里/小时，每公里加收 10 元；超过 66 公里/小时，每公里加收 20 元。现在绘制这个描述罚款金额的分段函数的图象。

罚款 (Y) 是每小时公里数 (X) 的函数：

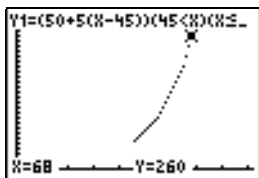
$$\begin{aligned}
 Y &= 0 & 0 < X \leq 45 \\
 Y &= 50 + 5(X - 45) & 45 < X \leq 55 \\
 Y &= 50 + 5 * 10 + 10(X - 55) & 55 < X \leq 65 \\
 Y &= 50 + 5 * 10 + 10 * 10 + 20(X - 65) & 65 < X
 \end{aligned}$$

## 步骤

1. 按 **[MODE]**。选择 **Func** 和默认设置。
2. 按 **[Y=]**。关闭所有函数和统计图。输入 Y= 函数来描述超速罚款。使用 **TEST** 菜单操作来定义该分段函数。将 **Y1** 的图象样式设为 **'·'** (点)。



3. 按 **[WINDOW]**，并设置 **Xmin=-2**、**Xscl=10**、**Ymin=-5** 和 **Yscl=10**。忽略 **Xmax** 和 **Ymax**，它们在步骤 4 中由 **ΔX** 和 **ΔY** 设置。
4. 按 **[2nd]** **[QUIT]** 返回到主屏幕。将 **1** 和 **5** 分别保存到 **ΔX** 和 **ΔY**。**ΔX** 和 **ΔY** 位于 **VARS** Window **X/Y** 子菜单，分别表示相邻像素点中心间的水平和垂直距离。**ΔX** 和 **ΔY** 的值为整数时，跟踪图象所看到的值更精确些。
5. 按 **[TRACE]** 绘制该函数的图象。罚款超过 250 元时速度为多少呢？



## 问题

绘制不等式  $0.4X^3 - 3X + 5 < 0.2X + 4$  的图象。使用 TEST 菜单操作来确定不等式成立和不成立时 X 的值分别是多少？

## 步骤

1. 按 **[MODE]**。选择 **Dot**、**Simul** 及其默认设置。设置 **Dot** 方式，将 Y= 编辑器中所有图象样式图标变成 '·' (点)。
2. 按 **[Y=]**。关闭所有函数和统计图。将不等式的左侧和右侧分别输入到 **Y4** 和 **Y5** 中。

```

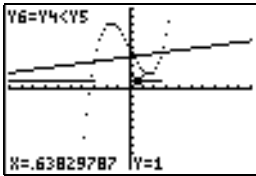
·Y4=.4X^3-3X+5
·Y5=.2X+4
·Y6=
·Y7=
    
```

3. 将不等式语句输入到 **Y6**。如果不等式成立，函数值为 **1**；否则为 **0**。

```

·Y4=.4X^3-3X+5
·Y5=.2X+4
·Y6=Y4<Y5
·Y7=
    
```

4. 按 **[ZOOM]** **6** 在标准窗口中绘制该不等式的图象。
5. 按 **[TRACE]** **[↓]** **[↓]** 移到 **Y6**，然后按 **[↓]** 和 **[→]** 跟踪不等式，观察 **Y** 的值。

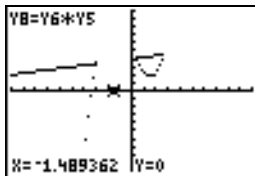
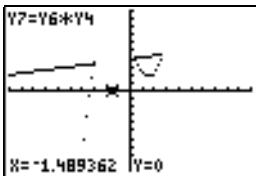


6. 按 **[Y=]**。关闭 **Y4**、**Y5** 和 **Y6**。输入这些方程，只绘制不等式的图象。

```

·Y4=.4X^3-3X+5
·Y5=.2X+4
·Y6=Y4<Y5
·Y7=Y6*Y4
·Y8=Y6*Y5
    
```

7. 按 **[TRACE]**。注意到不等式不成立时 **Y7** 和 **Y8** 的值为 **0**。

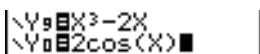


## 问题

利用图象求解方程  $X^3 - 2X = 2\cos(X)$ ，也就是说，求解包含两个未知数由两个方程组成的方程组： $Y = X^3 - 2X$  和  $Y = 2\cos(X)$ 。使用 ZOOM 因子控制图象上显示的小数位数。

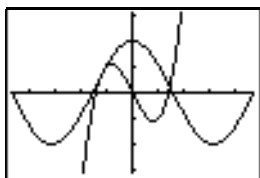
## 步骤

1. 按 [MODE]。选择默认方式设置。按 [Y=]。关闭所有函数和统计图。输入这两个函数。



```
\Y1=X^3-2X
\Y2=2cos(X)
```

2. 按 [ZOOM] 4 选择 4:Zdecimal。显示表明，可能存在两个根（两个函数相交点）。



3. 按 [ZOOM] 4 从 ZOOM MEMORY 菜单中选择 4:SetFactors。设置 XFact=10 和 YFact=10。
4. 按 [ZOOM] 2 选择 2:Zoom In。使用 [←]、[→]、[↑] 和 [↓] 将自由移动光标移到显示右侧中函数的交点上。移动光标时，请注意 X 和 Y 的值有 1 位小数。
5. 按 [ENTER] 放大。移动光标至交点上。移动光标时，请注意此时 X 和 Y 的值有 2 位小数位。
6. 按 [ENTER] 再放大。将自由移动光标准确地移到交点上。请注意小数位数。
7. 按 [2nd] [CALC] 5 选择 5:intersect。按 [ENTER] 选择第 1 条曲线，再按 [ENTER] 选择第 2 条曲线。将跟踪光标移至交点附近，估计交点的坐标值。按 [ENTER]，这个交点的坐标是多少呢？
8. 按 [ZOOM] 4 选择 4:ZDecimal 重新显示原先的图象。
9. 按 [ZOOM]。选择 2:Zoom In，重复步骤 4 至步骤 8，找到显示左侧中函数交点的坐标。



## 设置程序

本程序创建著名分形图，即 Sierpinski 三角形图，并将之保存到一幅图形中。首先，按 **PRGM** **▶** **▶** **1**，命名程序为 **SIERPINS**，然后按 **ENTER**，显示程序编辑器。

## 程序

```
PROGRAM:SIERPINS
:FnOff :ClrDraw
:PlotsOff
:AxesOff
:0→Xmin:1→Xmax
:0→Ymin:1→Ymax
:rand→X:rand→Y
:For(K,1,3000)
:rand→N
:If N≤1/3
:Then
:.5X→X
:.5Y→Y
:End
:If 1/3<N and N≤2/3
:Then
:.5(.5+X)→X
:.5(1+Y)→Y
:End
:If 2/3<N
:Then
:.5(1+X)→X
:.5Y→Y
:End
:Pt-On(X,Y)
:End
:StorePic 6
```

— 设置观察窗口。

— **For** 组的开始。

— **If/Then** 组。

— **If/Then** 组。

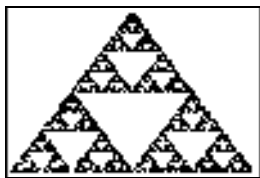
— **If/Then** 组。

画点。

**For** 组的结束。

保存图形。

执行上面的程序后，可以用指令 **RecallPic 6** 再次调出并显示这幅图。



## 问题

使用 **Web** 格式可以识别序列图象中那些具有吸引和排斥特性的点。

## 步骤

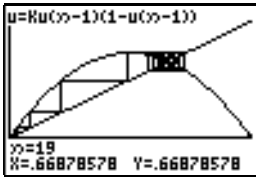
1. 按 **[MODE]**。选择 **Seq** 及其默认方式设置。按 **[2nd]** **[FORMAT]**，选择 **Web** 格式及其默认方式设置。
2. 按 **[Y=]**，清除所有函数，关闭所有统计图。输入对应于表达式  $Y = K X(1-X)$  的序列。

$$u(n)=Ku(n-1)(1-u(n-1))$$

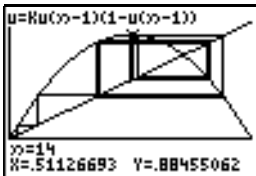
$$u(nMin)=.01$$

3. 按 **[2nd]** **[QUIT]** 返回到主屏幕，然后将 **2.9** 保存到 **K** 中。
4. 按 **[WINDOW]**。设置窗口变量：

<b>nMin=0</b>	<b>Xmin=0</b>	<b>Ymin=-.26</b>
<b>nMax=10</b>	<b>Xmax=1</b>	<b>Ymax=1.1</b>
<b>PlotStart=1</b>	<b>Xscl=1</b>	<b>Yscl=1</b>
<b>PlotStep=1</b>		
5. 按 **[TRACE]** 显示图象，然后按 **[▶]** 跟踪蛛网，该蛛网有一个吸引点。



6. 把 **K** 变成 **3.44**，然后跟踪图象以显示有两个吸引点的蛛网。
7. 把 **K** 变成 **3.54**，然后跟踪图象以显示有四个吸引点的蛛网。



## 设置程序

本程序绘制函数  $A \sin(BX)$  的图象，其系数是 1 至 10 之间的随机整数。试着估计该函数的系数，并以  $C \sin(DX)$  形式绘制估计结果的图象。程序直到估计结果正确时才退出。

## 程序

```
PROGRAM:GUESS
:PlotsOff :Func
:Fnoff :Radian
:ClrHome
:"Asin(BX)">Y1
:"Csin(DX)">Y2
:GraphStyle(1,1)
:GraphStyle(2,5)
:Fnoff 2

:randInt(1,10)>A
:randInt(1,10)>B
:0>C:0>D
:-2π>Xmin
:2π>Xmax
:π/2>Xsc1
:-10>Ymin
:10>Ymax
:1>Ysc1

:DispGraph
:Pause
:Fnoff 2
:Lbl Z
:Prompt C,D

:DispGraph
:Pause
:If C=A
:Text(1,1,"C IS OK")
:If C≠A
:Text(1,1,"C IS WRONG")
:If D=B
:Text(1,50,"D IS OK")
:If D≠B
:Text(1,50,"D IS WRONG")

:DispGraph
:Pause
:If C=A and D=B
:Stop
:Goto Z
```

定义方程。

设置 line 和 path 图象样式。

初始化系数。

设置观察窗口。

显示图象。

提示估计。

显示图象。

显示结果。

显示图象。

如果估计正确，则程序退出。

# 绘制单位圆和三角曲线

## 问题

使用参数图象方式绘制单位圆和正弦曲线，以显示它们之间的关系。

任何可以在 **Func** 方式下绘制的函数，可以通过将分量 **X** 定义为 **T**，分量 **Y** 定义为 **F(T)** 来在 **Par** 方式下绘制。

## 步骤

1. 按 **[MODE]**。选择 **Par**、**Simul** 和默认设置。

2. 按 **[WINDOW]**。设置观察窗口：

<b>Tmin=0</b>	<b>Xmin=-2</b>	<b>Ymin=-3</b>
<b>Tmax=2<math>\pi</math></b>	<b>Xmax=7.4</b>	<b>Ymax=3</b>
<b>Tstep=.1</b>	<b>Xscl=<math>\pi/2</math></b>	<b>Yscl=1</b>

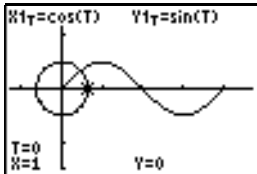
3. 按 **[Y=]**，关闭所有函数和统计图。输入定义圆心为 (0,0) 的单位圆的表达式。

```
Plot1 Plot2 Plot3
X1T █cos(T)
Y1T █sin(T)
X2T █T
Y2T █sin(T)
```

4. 输入定义正弦曲线的表达式。

```
Plot1 Plot2 Plot3
X1T █cos(T)
Y1T █sin(T)
X2T █T
Y2T █sin(T)
```

5. 按 **[TRACE]**。在图象绘制过程中，当发现正弦函数由单位圆展开时，可按 **[ENTER]** 暂停，再按 **[ENTER]** 后绘图继续。



**注意：**可以推广这种展开。请用其他三角函数替换 **Y2T** 中的 **sin(T)** 来展开函数。



## 问题

使用两对参数方程以确定两个运动物体何时在同一平面上靠得最近。

转轮的直径 ( $d$ ) 为 20 米，它正以每 12 秒一圈的速度 ( $s$ ) 逆时针转动。下面的参数方程表示时刻  $T$  时转轮上某个乘客的位置，其中  $\alpha$  表示旋转角度， $(0,0)$  是转轮底部的中心， $(10,10)$  是  $T=0$  时乘客最右侧点的位置。

$$X(T) = r \cos \alpha \quad \text{其中 } \alpha = 2\pi Ts \text{ 和 } r = d/2$$

$$Y(T) = r + r \sin \alpha$$

一个人站在地上，朝转轮上的乘客抛掷一个球。他的手臂与转轮的底部处在同一高度，但在距转轮最低点右侧 25 米 ( $b$ ) 远的  $(25,0)$  处。球抛出时的速度 ( $v_0$ ) 为 22 米/秒，与水平方向呈  $66^\circ$  的角度 ( $\theta$ )。下面的参数方程表示时刻  $T$  时球的位置。

$$X(T) = b - Tv_0 \cos\theta$$

$$Y(T) = Tv_0 \sin\theta - (g/2) T^2 \quad \text{其中 } g = 9.8 \text{ m/sec}^2$$

## 步骤

- 按 **[MODE]**。选择 **Par**、**Simul** 和默认设置。**Simul**（同时）方式模拟两个物体的运动过程。
- 按 **[WINDOW]**。设置观察窗口：

<b>Tmin=0</b>	<b>Xmin=-13</b>	<b>Ymin=0</b>
<b>Tmax=12</b>	<b>Xmax=34</b>	<b>Ymax=31</b>
<b>Tstep=.1</b>	<b>Xscl=10</b>	<b>Yscl=10</b>
- 按 **[Y=]**。关闭所有函数和统计图。输入描述转轮和球轨迹的表达式。将 **X2T** 的图象样式设置为  $\phi$ （轨迹）。

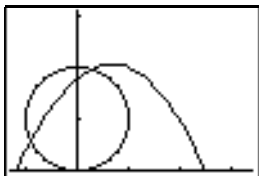
```

Plot1 Plot2 Plot3
\X1T B10cos(πT/6)
Y1T B10+10sin(πT
/6)
φX2T B25-22Tcos(6
6°)
Y2T B22Tsin(66°)
|(9.8/2)T²
    
```

提示：试一试将图象样式设置为  $\phi$  **X1T** 和  $\phi$  **X2T**。在这种设置下，按 **[GRAPH]** 时将模拟转轮上椅子的运动和球从空中落下的过程。

## 步骤 (续)

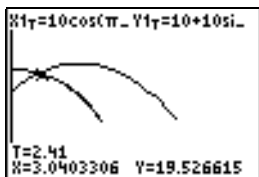
4. 按 **[GRAPH]** 绘制方程的图象。绘制它们时请仔细观察。请注意，球和转轮上乘客的轨迹在转轮的右上象限交叉，它们似乎在这交叉点处离得最近。



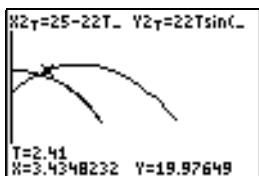
5. 按 **[WINDOW]**。改变观察窗口，以便注意力集中在图象的这一部分。

<b>Tmin=1</b>	<b>Xmin=0</b>	<b>Ymin=10</b>
<b>Tmax=3</b>	<b>Xmax=23.5</b>	<b>Ymax=25.5</b>
<b>Tstep=.03</b>	<b>Xscl=10</b>	<b>Yscl=10</b>

6. 按 **[TRACE]**。图象绘制完毕后，按 **[>]** 靠近转轮上的某个点，球和乘客的轨迹在此点交叉。请注意 **X**、**Y** 和 **T** 的值。



7. 按 **[<]** 移到球的轨迹。请注意 **X** 和 **Y** 的值 (**T** 值不变)。请注意光标所在位置，这是转轮上乘客经过交点时球的位置。球和乘客哪个先到达交叉点？



实际上，可以使用 **[TRACE]** 及时拍下快照，并仔细研究两个运动物体的相关特性。

## 问题 1

使用 MATH 菜单中的函数 **fnInt**( 和 **nDeriv**( 演示由导数和积分定义的函数图象：

$$F(x) = \int_1^x 1/t \, dt = \ln(x), x > 0 \text{ 和}$$

$$D_x \left[ \int_1^x 1/t \, dt \right] = 1/x$$

## 步骤 1

1. 按 **[MODE]**。选择默认设置。

2. 按 **[WINDOW]**。设置观察窗口：

<b>Xmin=.01</b>	<b>Ymin=-1.5</b>	<b>Xres=3</b>
<b>Xmax=10</b>	<b>Ymax=2.5</b>	
<b>Xscl=1</b>	<b>Yscl=1</b>	

3. 按 **[Y=]**。关闭所有函数和统计图。输入 1/T 的从 1 到 X 的数值积分和函数 ln(X)。将 **Y1** 的图象样式设为  $\setminus$  (实线)，**Y2** 的设为  $\curvearrowright$  (轨迹)。

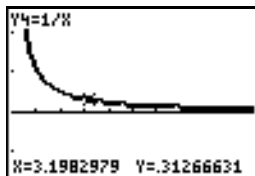
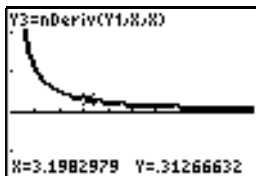
```
Plot1 Plot2 Plot3
\Y1=fnInt(1/T,T,
1,X)
\Y2=ln(X)
```

4. 按 **[TRACE]**。按 **[←]**、**[→]** 和 **[↓]** 比较 **Y1** 和 **Y2** 的值。

5. 按 **[Y=]**。关闭 **Y1** 和 **Y2**，然后输入 1/X 积分的数值微分和函数 1/X。将 **Y3** 和 **Y4** 的图象样式分别设为  $\setminus$  (实线) 和  $\setminus$  (粗线)。

```
Plot1 Plot2 Plot3
\Y1=fnInt(1/T,T,
1,X)
\Y2=ln(X)
\Y3=nDeriv(Y1,X,
X)
\Y4=1/X
```

6. 按 **[TRACE]**。再次使用光标键比较这两个图象化函数，即 **Y3** 和 **Y4** 的值。





## 问题 2

仔细研究由下式定义的函数。

$$y = \int_{-2}^x t^2 dt, \int_0^x t^2 dt \text{ 和 } \int_2^x t^2 dt$$

## 步骤 2

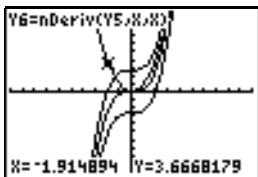
1. 按  $\boxed{Y=}$ 。关闭所有函数和统计图。使用数组同时定义这三个函数。将函数保存到 **Y5** 中。

```
Plot1 Plot2 Plot3
1, X)
Y2=ln(X)
Y3=nDeriv(Y1,X,
X)
Y4=1/X
Y5=fnInt(T^2,T,(-
-2,0,2),X)
```

2. 按  $\boxed{ZOOM}$  6 选择 **6:ZStandard**。
3. 按  $\boxed{TRACE}$ 。请注意，这些函数似乎是相同的，只是在垂直方向移动了一个常量的距离。
4. 按  $\boxed{Y=}$ 。在 **Y6** 中输入 **Y5** 的数值微分。

```
Plot1 Plot2 Plot3
Y3=nDeriv(Y1,X,
X)
Y4=1/X
Y5=fnInt(T^2,T,(-
-2,0,2),X)
Y6=nDeriv(Y5,X,
X)
```

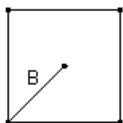
5. 按  $\boxed{TRACE}$ 。请注意，由 **Y5** 定义的两个函数是不同的，但它们的导数相同。



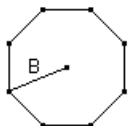
## 问题

使用方程求解器保存正 N 边形面积公式，然后求解每个变量在给定其他变量时的值，最后研究这样的事实，即正 N 边形面积的极限情形是圆的面积  $\pi r^2$ 。

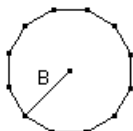
考虑公式  $A = NB^2 \sin(\pi/N) \cos(\pi/N)$ ，它表示边长相等、中心到顶角距离为 B 的正 N 边形的面积。



N = 4 边



N = 8 边



N = 12 边

## 步骤

- 按 **MATH** 0 从 MATH 菜单中选择 **0:Solver**。方程编辑器或是交互式求解器编辑器显示出来。如果显示交互式求解器编辑器，则按 **□** 显示方程编辑器。
- 输入公式  $0=A-NB^2\sin(\pi/N)\cos(\pi/N)$ ，然后按 **ENTER**，则显示交互式求解器编辑器。

```
A-NB^2sin(pi/N)...=0
A=0
N=0
B=0
bound=(-1E99,1...
```

- 输入 **N=4** 和 **B=6**，求中心到顶角距离 (B) 为 6 厘米的正方形面积 (A)。
- 按 **↑** **↑** 将光标移到 **A** 上，然后按 **ALPHA** [SOLVE]。在交互式求解器编辑器中显示 **A** 的结果。

```
A-NB^2sin(pi/N)...=0
A=72.000000000...
N=4
B=6
bound=(-1E99,1...
left-rt=0
```

- 现在，给定面积，求解 **B** 在不同边数时的值。输入 **A=200** 和 **N=6**，求距离 **B** 时，将光标移到 **B** 上，然后按 **ALPHA** [SOLVE]。
- 输入 **N=8**。求距离 **B** 时，将光标移到 **B** 上，然后按 **ALPHA** [SOLVE]。请先后求 **N=9** 和 **N=10** 时 **B** 的值。

## 步骤 (续)

给定  $B=6$ ，求解  $N=10$ 、 $100$ 、 $150$ 、 $1000$  和  $10000$  时多边形的面积，并把计算结果同  $\pi 6^2$ （半径为 6 的圆面积，约等于 113.097）比较。

7. 输入  $B=6$ 。求面积  $A$  时，将光标移到  $A$  上，然后按  $\boxed{\text{ALPHA}} \boxed{[SOLVE]}$ 。依次求解  $N=10$ 、 $N=100$ 、 $N=150$ 、 $N=1000$  和  $N=10000$  时面积  $A$  的值。请注意，随着  $N$  增大，面积  $A$  越来越接近  $\pi B^2$ 。

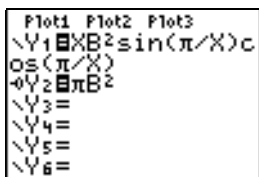
现在绘制方程的图象，以直观地反映多边形的面积是如何随其边数的增大而变化的。

8. 按  $\boxed{\text{MODE}}$ 。选择默认方式设置。

9. 按  $\boxed{\text{WINDOW}}$ 。设置观察窗口：

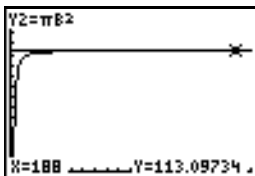
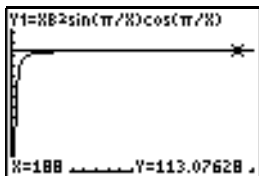
$X_{\min}=0$	$Y_{\min}=0$	$X_{\text{res}}=1$
$X_{\max}=200$	$Y_{\max}=150$	
$X_{\text{scl}}=10$	$Y_{\text{scl}}=10$	

10. 按  $\boxed{Y=}$ 。关闭所有函数和统计图。输入面积的方程，用  $X$  替换  $N$ 。设置所示图象样式。



```
Plot1 Plot2 Plot3
\Y1 XB^2 sin(pi/X) c
os(pi/X)
+Y2 pi B^2
\Y3 =
\Y4 =
\Y5 =
\Y6 =
```

11. 按  $\boxed{\text{TRACE}}$ 。图象绘制完毕后，按  $100 \boxed{\text{ENTER}}$  跟踪  $X=100$ 。按  $150 \boxed{\text{ENTER}}$ ；按  $188 \boxed{\text{ENTER}}$ 。请注意，随着  $X$  增加， $Y$  的值收敛到  $\pi 6^2$ （近似值为 113.097）。 $Y_2 = \pi B^2$ （圆的面积）是  $Y_1$  的水平渐近线。中心到顶角的距离为  $r$  的正  $N$  边形的面积随着  $N$  的增大而逼近半径为  $r$  的圆面积 ( $\pi r^2$ )。



## 问题

您是某家抵押公司的贷款官员，最近增设了 30 年住宅抵押贷款业务，利息为 8%，每月支付 800 元。新住宅业主想知道，如果他们从现在起 20 年内支付 240 次，那么他们将支付多少利息，支付多少本金。

## 步骤

1. 按 **[MODE]**，并将定点小数方式设成 2 位小数位，其他方式设置成默认值。
2. 按 **[APPS]** **[ENTER]** **[ENTER]** 显示 TVM Solver。输入这些值。

```
N=360.00
I%=8.00
PV=0.00
PMT=800.00
FV=0.00
P/Y=12.00
C/Y=12.00
PMT: [END] BEGIN
```

**注意：**输入正数 (800)，将 **PMT** 表示为现金流入。支付金额在图象上将显示为正数。**FV** 输入 0，这是因为一旦贷款全部付清，其未来值为 0。输入 **PMT: END**，因为是在支付期末支付。

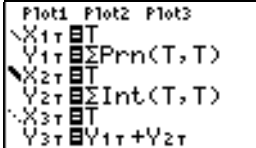
3. 将光标移到 **PV=** 提示上，然后按 **[ALPHA]** **[SOLVE]**，则在 **PV=** 提示上显示出房屋的现值或抵押金额。

```
N=360.00
I%=8.00
PV=-109026.80
PMT=800.00
FV=0.00
P/Y=12.00
C/Y=12.00
PMT: [END] BEGIN
```

步骤 (续)

现在比较每次支付的利息和本金的图象。

- 按 **[MODE]**。设置 **Par** 和 **Simul**。
- 按 **[Y=]**。关闭所有函数和统计图。输入这些方程，并设置图象样式如下：



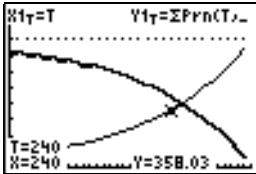
注意： $\Sigma Prn$ (和  $\Sigma Int$ ) 在 **FINANCE** 菜单 (**APPS 1:FINANCE**) 上可以找到。

- 按 **[WINDOW]**。设置这些窗口变量：

<b>Tmin=1</b>	<b>Xmin=0</b>	<b>Ymin=0</b>
<b>Tmax=360</b>	<b>Xmax=360</b>	<b>Ymax=1000</b>
<b>Tstep=12</b>	<b>Xscl=10</b>	<b>Yscl=100</b>

提示：要加快图象显示速度，请将 **Tstep** 改为 **24**。

- 按 **[TRACE]**。图象绘制完毕后，按 **240 [ENTER]** 将跟踪光标移到 **T=240** 上，这相当于支付 20 年。

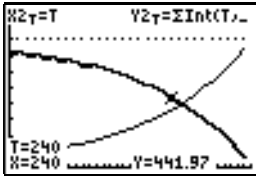


图象表明，对于第 240 次支付 (**X=240**)，支付的 800 元中 358.03 元是本金 (**Y=358.03**)。

注意：支付的总额 (**Y3T=Y1T+Y2T**) 总是 800 元。

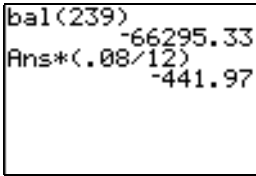
步骤 (续)

8. 按  $\square$  将光标移到由 **X2T** 和 **Y2T** 定义的利息函数上。输入 **240**。



图象表明，对于第 240 次支付 ( $X=240$ )，支付的 800 元中 441.97 元是利息 ( $Y=441.97$ )。

9. 按  $\square$  [QUIT]  $\square$  [APPS]  $\square$  [ENTER] **9** 将 **9:bal**( 粘贴主屏幕中。在图象中检查下列数字。

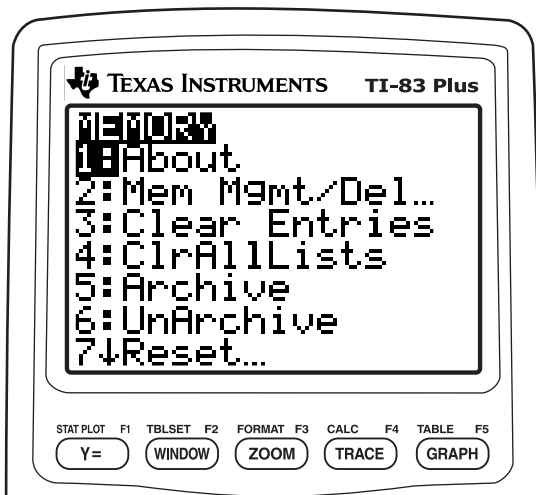


在哪次的月支付时本金份额将超过利息份额?

# 18 内存和变量管理

## 目录

检查可用内存.....	1
从内存中删除项目.....	4
清除输入项和数组元素.....	5
重新设置 TI-83 Plus.....	6
归档和取消归档变量.....	10
组合和取消组合变量.....	13
碎片收集.....	17
ERR:ARCHIVE FULL 消息.....	20



### MEMORY 菜单

可以在任何时候，通过选择 MEMORY 菜单下的项目来检查可用内存或管理现有内存。要访问此菜单，请按 **[2nd] [MEM]**。

---

#### MEMORY

1: About...	显示计算器的信息。
2: Mem Mgmt/Del...	报告可用内存和变量的使用情况。
3: Clear Entries	清除 ENTRY（最后输入项存储）。
4: ClrAllLists	清除内存中的所有数组。
5: Archive...	归档所选变量。
6: UnArchive...	取消归档所选变量。
7: Reset...	显示 RAM、ARCHIVE 和 ALL 菜单。
8: Group...	显示 GROUP 和 UNGROUP 菜单。

---

要检查内存的使用情况，请先按 **[2nd] [MEM]**，然后按 **2:Mem Mgmt/Del**。

### 显示 MEMORY MANAGEMENT/DELETE 菜单

**Mem Mgmt/Del** 显示 MEMORY MANAGEMENT/DELETE 菜单。顶部两行报告可用 RAM 和 ARCHIVE 内存的总数。通过选择屏幕上的菜单项，可以查看每种变量类型正在使用的内存大小。该信息可以帮助用户确定是否需要从内存中删除某些变量来为新数据（例如程序或应用程序）腾出空间。



要检查内存的使用情况，请按下列步骤进行：

1. 按 **[2nd] [MEM]** 显示 MEMORY 菜单。

```
1: About
2: Mem Mgmt/Del...
3: Clear Entries
4: ClrAllLists
5: Archive
6: UnArchive
7: Reset...
```

**注意：**左列的顶部或底部的 ↑ 和 ↓ 表示可以上下滚动屏幕来查看更多的变量类型。

2. 选择 **2: Mem Mgmt/Del** 将显示 MEMORY MANAGEMENT/DELETE 菜单。TI-83 Plus 以字节表示内存数。

```
RAM FREE 24347
ARC FREE 163840
1: All...
2: Real...
3: Complex...
4: List...
5: Matrix...
6: Y-Vars...

7: Prgm...
8: Pic...
9: GDB...
0: String...
A: Apps...
B: AppVars...

C: Group...
```

3. 从列表中选择变量类型以显示内存的使用情况。

**注意：**Real、List、Y-Vars 和 Prgm 变量类型决不能重新置成零，即使在内存清除以后。

**Apps** 是存储在 Flash ROM 中的独立的应用程序。**AppVars** 是变量占位符，用于存储由独立应用程序创建的变量。只能通过创建 **AppVars** 中的变量的应用程序才能对其进行编辑或更改。

要退出 MEMORY MANAGEMENT/DELETE 菜单，按 **[2nd] [QUIT]** 或 **[CLEAR]** 都可。这两种选择都显示主屏幕。

### 删除某一项目

要通过删除一些变量的内容（实数或复数、数组、矩阵、Y= 变量、程序、Apps、AppVars、图形、图象数据库或字符串）来增加可用的内存，请按下列步骤进行。

1. 按 **[2nd]** [MEM] 显示 MEMORY 菜单。
2. 选择 **2:Mem Mgmt/Del** 显示 MEMORY MANAGEMENT/DELETE 菜单。
3. 选择要删除的数据类型，或者选择 **1:All** 以得到各种类型的全部变量的列表。之后，则显示出一个屏幕，其中列出了所选类型的所有变量及每个变量所用的字节数。

例如，如果选择 **4:List**，则显示数组编辑器屏幕。

RAM	FREE	24347
ARC	FREE	163840
	L <sub>1</sub>	12
▶	L <sub>2</sub>	12
	L <sub>3</sub>	12

4. 按 **[↑]** 和 **[↓]** 将选择光标 (▶) 移到要删除的项目旁边，然后按 **[DEL]**，则变量从内存中删除掉。从该屏幕可以逐个删除变量。

**注意：**如果要删除程序或 Apps，将出现请求确认该删除操作的提示。选择 **2:Yes** 将继续该操作。

要退出任何变量屏幕而不再删除时，则按 **[2nd]** [QUIT]，显示主屏幕。

**注意：**不能删除某些系统变量，如上次结果变量 **Ans** 和统计变量 **RegEQ**。

## 清除输入项

**Clear Entries** 清除 ENTRY（最后输入项）存储区中的内容（第 1 章）。要清除 ENTRY 存储区，则按下列步骤进行：

1. 按 **[2nd]** **[MEM]** 显示 MEMORY 菜单。
2. 选择 **3:Clear Entries** 将指令粘贴到主屏幕。
3. 按 **[ENTER]** 清除 ENTRY 存储区。

```
Clear Entries
Done
```

要取消 **Clear Entries**，则按 **[CLEAR]**。

**注意：**如果从程序内选择了 **3:Clear Entries**，**Clear Entries** 指令将被粘贴到程序编辑器中。当程序执行时，**Entry**（最后输入项）被清除掉。

## ClrAllLists

**ClrAllLists** 只将 RAM 中每个数组的维数设为 **0**。

要从所有的数组中清除全部元素，则按下列步骤进行：

1. 按 **[2nd]** **[MEM]** 显示 MEMORY 菜单。
2. 选择 **4:ClrAllLists** 将指令粘贴到主屏幕。
3. 按 **[ENTER]** 将内存中每个数组的维数设为 **0**。

```
ClrAllLists
Done
```

要取消 **ClrAllLists**，请按 **[CLEAR]**。

**ClrAllLists** 不从内存、LIST NAMES 菜单或统计数组编辑器中删除数组名。

**注意：**如果从程序内选择 **4:ClrAllLists**，**ClrAllLists** 指令将被粘贴到程序编辑器。当程序执行时数组被清除掉。

## RAM ARCHIVE ALL 菜单

RAM ARCHIVE ALL 菜单可用来重新设置全部内存（包括默认值）或在保留内存中存储的其它数据，如程序和 Y= 函数的情况下重新设置所选的部分内存。例如，可以选择重新设置所有 RAM 或只恢复默认设置。请注意，如果选择重新设置 RAM，那么 RAM 中的所有数据和程序将被消除。对于归档内存，可以重新设置变量 (Vars)、应用程序 (Apps) 或两者都设置。请注意，如果选择重新设置 Vars，则归档内存中的所有数据和程序将被消除。如果选择重新设置 Apps，则归档内存中的所有应用程序将被消除。

重新设置 TI-83 Plus 上的默认值时，RAM 中的所有默认值将恢复为出厂设置。但不会改变存储的数据和程序。

下面是一些通过重新设置默认值来恢复 TI-83 Plus 默认值的例子。

- 方式设置，如 **Normal**（记数法）、**Func**（图象）、**Real**（数字）和 **Full**（屏幕）
- Y= 函数关闭
- 窗口变量值，如 **Xmin=-10**、**Xmax=10**、**Xscl=1**、**Yscl=1** 和 **Xres=1**
- 统计图形关闭
- 格式设置，如 **CoordOn**（图象坐标开启）、**AxesOn** 和 **ExprOn**（表达式开启）
- **rand** 起始值为 0

## 显示 RAM ARCHIVE ALL 菜单

要显示 TI-83 Plus 上的 RAM ARCHIVE ALL 菜单，请按下列步骤进行：

1. 按 **2nd** [MEM] 显示 MEMORY 菜单。
2. 选择 **7:Reset** 显示 RAM ARCHIVE ALL 菜单。



## 重新设置 RAM 内存

重新设置 RAM 将 RAM 系统变量恢复为出厂设置，并删除所有非系统变量和所有程序。重新设置默认值将所有系统变量恢复为默认设置而不删除 RAM 中的变量和程序。重新设置 RAM 或重新设置默认值不影响用户数据归档区中的变量和应用程序。

**提示：**重新设置所有 RAM 内存之前，可以考虑通过只删除所选的数据恢复足够的可用内存（页 18-4）。

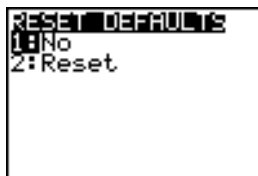
要在 TI-83 Plus 上重新设置所有 RAM 内存或 RAM 默认值，请按以下步骤执行：

1. 从 RAM ARCHIVE ALL 菜单中，选择 **1:ALL RAM** 以显示 RESET RAM 菜单或 **2:Defaults** 以显示 RESET DEFAULTS 菜单。



```
RESET RAM
1:No
2:Reset

Resetting RAM
erases all data
and Programs
from RAM.
```



```
RESET DEFAULTS
1:No
2:Reset
```

2. 如果正在重新设置 RAM，请阅读 RESET RAM 菜单下的消息。
  - 要取消重新设置并返回主屏幕，按 **ENTER**。
  - 要消除 RAM 内存或重新设置默认值，选择 **2:Reset**。根据选择，主屏幕上将显示 **RAM cleared** 或 **Defaults set** 消息。

## 重新设置归档区内存

在 TI-83 Plus 上重新设置归档区内存时，可以选择删除用户数据归档区的所有变量、所有应用程序或变量和应用程序。

要重新设置所有或部分用户数据的归档区内存，请按以下步骤进行：

1. 从 RAM ARCHIVE ALL 菜单，按 **▶** 显示 ARCHIVE 菜单。



```
RAM ARCHIVE ALL
1:Vars...
2:Apps...
3:Both...
```

2. 选择下列选项之一：
  - **1:Vars** 显示 RESET ARC VAR 菜单



```
RESET ARC VAR
1:No
2:Reset

Resetting Vars
erases all data
and Programs
from Archive.
```

- **2:Apps** 显示 RESET ARC APPS 菜单。

```
RESET ARC APPS
1:No
2:Reset

Resetting Apps
erases all APPS
from Archive.
```

- **3:Both** 显示 RESET ARC BOTH 菜单。

```
RESET ARC BOTH
1:No
2:Reset

Resetting Both
erases all data,
programs & APPS
from Archive.
```

### 3. 阅读菜单下的消息。

- 要取消重新设置并返回主屏幕，按 **ENTER**。
- 要继续重新设置，选择 **2:Reset**。将在主屏幕上显示一条消息，它会说明已清除的归档区内存的类型。

## 重新设置所有内存

在 TI-83 Plus 上重新设置所有内存时，将 RAM 和用户数据归档区内存恢复为出厂设置，并删除所有非系统变量、应用程序和程序。将所有系统变量重新设置为默认设置。

**提示：**重新设置所有内存之前，可以考虑通过只删除所选的数据来恢复足够的可用内存（页 18-4）。

要在 TI-83 Plus 上重新设置所有内存，请按以下步骤进行：

1. 在 RAM ARCHIVE ALL 菜单中，按  $\blacktriangleright$   $\blacktriangleright$  显示 ALL 菜单。



```
RAM ARCHIVE ALL
1:All Memory...
```

2. 选择 **1:All Memory** 显示 RESET MEMORY 菜单。



```
RESET MEMORY
1:No
2:Reset
Resetting ALL
will delete all
data, programs &
Apps from RAM &
Archive.
```

3. 阅读 RESET MEMORY 菜单下的消息。

- 要取消重新设置并返回主屏幕，按  $\text{ENTER}$ 。
- 要继续重新设置，选择 **2:Reset**，而在主屏幕上则显示消息 **MEM cleared**。

**注意：**当清除内存时，对比度有时会改变。如果屏幕变色或变白，通过按  $2\text{nd}$   $\blacktriangle$  或  $\blacktriangledown$  调整对比度。

## 归档和取消归档变量

归档操作可以将数据、程序或其他变量储存在用户数据归档区（该区域中的变量不会被用户在无意中编辑或删除）。归档操作还可以为可能需要额外内存的变量释放 RAM。

不能编辑或执行已归档变量，而只能对其进行查看和取消归档操作。例如，如果归档数组 **L1**，那么可以在内存中看到 **L1**；但是如果选择它并将名称 **L1** 粘贴到主屏幕，那么将无法查看其内容或对其进行编辑。

**注意：**并不是对所有变量都可进行归档操作，也不是对所有已归档的变量都可进行取消归档操作。例如，不能归档系统变量（包括 **r**、**t**、**x**、**y** 和 **θ**）。Apps 和群组 (Groups) 总是存在于 Flash ROM 中，因此不需要将它们归档。无法对 Groups 进行取消归档操作，但可以取消组合或将它们删除。

变量类型	名称	归档? (是/否)	取消归档? (是/否)
实数	<b>A</b> 、 <b>B</b> ... <b>Z</b>	是	是
复数	<b>A</b> 、 <b>B</b> ... <b>Z</b>	是	是
矩阵	[A]、[B]、[C]... [J]	是	是
数组	<b>L1</b> 、 <b>L2</b> 、 <b>L3</b> 、 <b>L4</b> 、 <b>L5</b> 、 <b>L6</b> 和用户定义的名称	是	是
程序		是	是
函数	<b>Y1</b> 、 <b>Y2</b> ... <b>Y9</b> 、 <b>Y0</b>	否	不适用
参数方程	<b>X1T</b> 和 <b>Y1T</b> ... <b>X6T</b> 和 <b>Y6T</b>	否	不适用
极坐标函数	<b>r1</b> 、 <b>r2</b> 、 <b>r3</b> 、 <b>r4</b> 、 <b>r5</b> 、 <b>r6</b>	否	不适用
序列函数	<b>u</b> 、 <b>v</b> 、 <b>w</b>	否	不适用
统计图	<b>Plot1</b> 、 <b>Plot2</b> 、 <b>Plot3</b>	否	不适用
Graph 数据库	<b>GDB1</b> 、 <b>GDB2</b> ...	是	是
Graph 图片	<b>Pic1</b> 、 <b>Pic2</b> ...、 <b>Pic9</b> 、 <b>Pic0</b>	是	是
字符串	<b>Str1</b> 、 <b>Str2</b> ... <b>Str9</b> 、 <b>Str0</b>	是	是
表	<b>TblStart</b> 、 <b>Tb1</b> 、 <b>TblInput</b>	否	不适用
应用程序	应用程序	参见上述“注意”	否
应用程序变量	应用程序变量	是	是
群组		参见上述“注意”	否
保留名称的变量	<b>minX</b> 、 <b>maxX</b> 、 <b>RegEQ</b> 及其他	否	不适用
系统变量	<b>Xmin</b> 、 <b>Xmax</b> 及其他	否	不适用



可按下列两种方法进行归档和取消归档：

- 使用 MEMORY 菜单或 CATALOG 中的 **5:Archive** 或 **6:UnArchive** 命令。
- 使用 Memory Management 编辑器屏幕。

在归档或取消归档变量（尤其是字节数较多的变量，例如大程序）之前，可使用 MEMORY 菜单：

- 确定变量的大小。
- 查看是否有足够的可用空间。

对于：	空间要求：
归档	归档区可用空间 > 变量大小
取消归档	RAM 可用空间 > 变量大小

**注意：**如果没有足够的空间，那么有必要取消归档或删除变量。请注意，当取消归档变量时，并不释放用户数据归档区中与该变量相关联所有内存，因为系统要跟踪该变量在 RAM 中曾经及目前的位置。

即使有足够的可用空间，当试图归档变量时也可能看到 Garbage Collection 消息（页 18-16）。依据用户数据归档区中空块的使用情况，可能需要取消归档现有的变量以产生更多的可用空间。

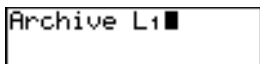
要使用 MEMORY 菜单下的 Archive/UnArchive 选项来归档或取消归档数组变量 (L1)，请执行下列各步：

1. 按 **[2nd] [MEM]** 显示 MEMORY 菜单。

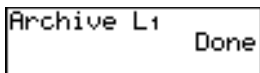


2. 选择 **5:Archive** 或 **6:UnArchive** 将命令置于编辑屏幕上。

3. 按 **[2nd] [L1]** 将 L1 变量置于编辑屏幕上。



4. 按 **[ENTER]** 完成归档过程。



**注意：**将在已归档变量名左边显示一个星号来表示其已被归档。

使用 Memory Management 编辑器归档或取消归档数组变量 (L1):

1. 按 [2nd] [MEM] 显示 MEMORY 菜单。

```
MEMORY
1:About
2:Mem Mgmt/Del...
3:Clear Entries
4:ClrAllLists
5:Archive
6:UnArchive
7↓Reset...
```

2. 选择 2:Mem Mgmt/Del... 显示 MEMORY MANAGEMENT/DELETE 菜单。

```
RAM FREE 25706
ARC FREE 130927
1:All...
2:Real...
3:Complex...
4>List...
5:Matrix...
6↓V-Vars...
```

3. 选择 4>List... 显示 LIST 菜单。

```
RAM FREE 25706
ARC FREE 130927
▶ L1 12
L2 12
L3 12
L4 12
L5 12
L6 12
```

4. 按 [ENTER] 归档 L1。在 L1 的左边将出现星号，表示它是一个已归档的变量。要取消归档该屏幕上的变量，将光标放到已归档变量的旁边并按 [ENTER]，则星号将消失。

```
RAM FREE 25708
ARC FREE 130912
▶*L1 12
L2 12
L3 12
L4 12
L5 12
L6 12
```

5. 按 [2nd] [QUIT] 退出 LIST 菜单。

**注意：**可以对已归档的变量进行链接、删除或取消归档，但不能对其进行编辑。

## 组合变量

组合操作允许对 RAM 中的两个或多个变量进行拷贝，然后将它们作为一个群组存储在用户数据归档区。RAM 中的变量未被消除。对变量进行组合操作之前，这些变量必须已存在于 RAM 中。也就是说，已归档的数据不能包括在群组中。

要创建变量群组，请：

1. 按 **[2nd]** **[MEM]** 显示 MEMORY 菜单。

```
1:MEMO
2:Mem Mgmt/Del...
3:Clear Entries
4:ClrAllLists
5:Archive
6:UnArchive
7:Reset...
8:Group...
```

2. 选择 **8:Group...** 显示 GROUP UNGROUP 菜单。

```
8:GROUP UNGROUP
1:Create New
```

3. 按 **[ENTER]** 显示 GROUP 菜单。

```
GROUP
Name=
```

4. 输入新群组的名称并按 **[ENTER]**。

**注意：**群组名允许包含一至八个字符。第一个字符必须是从 A 到 Z 的字母或 0。第二到第八个字符可以是字母、数字或 0。

```
GROUP
Name=GROUPA
```

5. 选择要组合的数据的类型。可以选择 **1:All+**，这会显示所有可用和已选择类型的所有变量。也可以选择 **2:All-**，这会显示所有可用类型的所有变量，但并不显示已选择类型的变量。所选类型的每个变量将会显示在屏幕上。

```
GROUP
1:All+...
2:All-...
3:Prgm...
4>List...
5:GDB...
6:Pic...
7:Matrix...
```

## 组合和取消组合变量 (续)

例如，假设已在 RAM 中创建一些变量并选择 **2:All-**，则显示下列屏幕。

```
SELECT Done
└─ PROGRAM1 PRGM
  PROGRAM2 PRGM
  GDB1      GDB
  L1        LIST
  L2        LIST
  L3        LIST
  L4        LIST
```

- 按 **▶** 和 **◀** 将选择光标 (**▶**) 移动到要将其拷贝到群组的第一个变量旁，然后按 **[ENTER]**。在所选的要对其进行组合的所有变量左边将出现一个小正方形。

```
SELECT Done
└─ PROGRAM1 PRGM
  ■ PROGRAM2 PRGM
  ■ GDB1      GDB
  ■ L1        LIST
  L2        LIST
  L3        LIST
  L4        LIST
```

重复该选择过程直到选定组成新群组的所有变量，然后按 **▶** 显示 **DONE** 菜单。

```
SELECT Done
└─ Done
```

- 按 **[ENTER]** 完成组合过程。

```
Copying
Variables to
Group:
GROUPA           Done
```

**注意：**只能组合 RAM 中的变量，而不能组合某些系统变量（例如最后答案变量 **Ans** 和统计变量 **RegEQ**）。

---

## 取消组合变量

取消组合可以对存储在用户数据归档区的群组中的变量进行拷贝、将它们置于 RAM 中并取消组合。

要取消组合，请按页 18-16 的步骤执行。

## DuplicateName 菜单

进行取消组合时，如果在 RAM 中发现重复的变量名，则显示 DUPLICATE NAME 菜单。

---

### DuplicateName

1: Rename	提示重命名接受变量。
2: Overwrite	覆盖所接受的重复变量中的数据。
3: Overwrite All	覆盖所有接受的重复变量中的数据。
4: Omit	跳过发送变量的传送。
5: Quit	在重复变量处停止传送。

---

### 关于菜单项目的注意事项

- 选择 **1:Rename** 时，将显示 **Name=** 提示符且字母锁开启。输入新变量名，然后按 **[ENTER]**。重新开始取消组合。
- 选择 **2:Overwrite** 时，会覆盖在 RAM 中发现的重复变量名的数据。重新开始取消组合。
- 选择 **3: Overwrite All** 时，会覆盖 RAM 中发现的所有重复变量名的数据。重新开始取消组合。
- 选择 **4:Omit** 时，不对与 RAM 中发现的重复变量名冲突的变量进行取消组合操作。取消组合从下一个项目重新开始。
- 选择 **5:Quit** 时，停止取消组合且不进行其他改变。

取消组合变量群组:

1. 按 **[2nd] [MEM]** 显示 MEMORY 菜单。



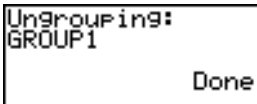
```
1:Mem Mgmt/Del...
2:Clear Entries
3:ClrAllLists
4:Archive
5:UnArchive
6:Reset...
7:Group...
```

2. 选择 **8:Group...** 显示 GROUP UNGROUP 菜单。
3. 按 **[>]** 显示 UNGROUP 菜单。



```
GROUP UNGROUP
1:*GROUP1
2:*GROUPA
3:*GROUPC
```

4. 按 **[↑]** 和 **[↓]** 将选择光标 (**▶**) 移动到要取消组合的群组变量的旁边, 然后按 **[ENTER]**。



```
Ungrouping:
GROUP1

Done
```

取消组合操作完成。

**注意:** 取消组合不会删除用户数据归档区中的群组。要删除该数组, 必须在用户数据归档区中对其进行删除操作。

### 碎片收集消息

如果大量使用用户数据归档区，则可以看到 **Garbage Collect?** 消息。如果试图在没有足够的可用归档内存时归档变量，将出现该消息。TI-83 Plus 将试图重新安排已归档变量以腾出更多的空间。

### 碎片收集消息的响应

- 要取消，选择 **1:No**。
- 如果选择 **1:No**，则将显示消息 **ERR:ARCHIVE FULL**。



- 要继续归档，选择 **2:Yes**。

如果选择 **2:Yes**，则将显示处理消息 **Garbage Collecting...** 或 **Defragmenting...**。

**注意：**无论何时遇到标记要删除的应用程序，都将显示处理消息 **Defragmenting...**。

碎片收集可能会花 20 分钟，这取决于已用于存储变量的归档内存的大小。

碎片收集之后，根据释放的额外空间的大小，变量可能被归档，也可能不被归档。如果没有归档，则可以在取消归档某些变量后重新归档变量。

### 没有消息就不自动执行碎片收集的原因

消息的作用：

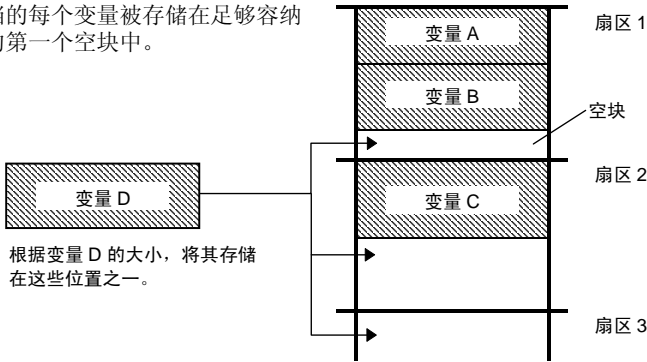
- 告诉用户为什么归档会比平时占用更长的时间。还提醒用户，如果没有足够的内存那么归档会失败。
- 当发现程序进入一个重复填充用户数据归档区的循环时会提醒用户。取消归档并确定原因。

## 有必要进行碎片收集的原因

用户数据归档区被划分为扇区。当第一次开始归档时，变量连续地存储在扇区 1 中。这样一直继续到扇区的结束位置。

已归档变量被存储在单个扇区的一个连续的块中。不同于存储在用户数据归档区中的应用程序，已归档变量不能跨过扇区的边界。如果在扇区中没有足够的剩余空间，则下一个变量将被存储到下一个扇区的起始位置。一般来说，这会在前一个扇区的结束位置留下一个空块。

归档的每个变量被存储在足够容纳它的第一个空块中。



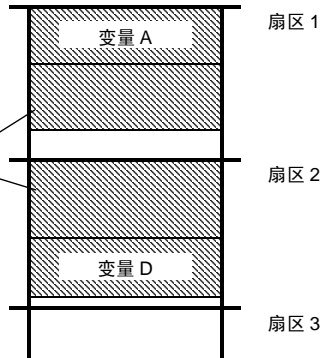
继续执行该过程直到最后一个扇区。根据单个变量的大小，空块可能占大量空间。当正在归档的变量大于任何空块时，将会进行碎片收集。

## 取消归档变量操作如何影响该过程

取消归档变量时，变量被拷贝到 RAM，但实际上并没有从用户数据归档区内存中将其删除。

取消归档变量 B 和 C 之后，它们仍然占用空间。

取消归档变量被“标记要删除”，这表示将在下次碎片收集时将其删除。





## 如果 MEMORY 屏幕显示足够的可用空间

即使 MEMORY 屏幕显示有足够的可用空间归档变量或存储应用程序，也可能出现 **Garbage Collect?** 消息或 **ERR: ARCHIVE FULL** 消息（页 18-20）。

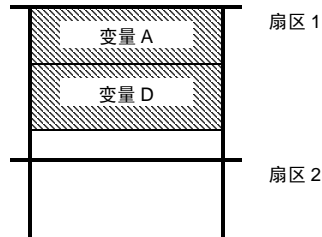
当取消归档变量时，归档区可用空间数量立即增加，但实际上只有下次碎片收集后，这些空间才可用。

如果归档区可用空间数量表明有足够的可用空间用于归档变量，那么在碎片收集之后可能会有足够的空间归档变量（取决于所有空块的可用性）。

## 碎片收集过程

碎片收集过程：

- 从用户数据归档区中删除已取消归档的变量。
- 将剩余变量重新安排到连续的块。



**注意：**在碎片收集期间电力不足可能会导致所有内存（RAM 和归档区）被删除。

## 使用 GarbageCollect 命令

可通过定期优化内存来减少自动进行碎片收集的次数。这可以通过使用 **GarbageCollect** 命令来实现。

要使用 **GarbageCollect** 命令，请执行下列步骤：

1. 按 **[2nd]** **[CATALOG]** 显示 CATALOG。



2. 按 **[↓]** 或 **[↑]** 滚动 CATALOG 直到选择光标指向 **GarbageCollect** 命令。
3. 按 **[ENTER]** 将命令粘贴到当前屏幕。
4. 按 **[ENTER]** 显示 **Garbage Collect?** 消息。
5. 选择 **2:Yes** 开始碎片收集。

即使 MEMORY 屏幕显示有足够的可用空间归档变量或存储应用程序，也可能出现 **ERR:ARCHIVE FULL** 消息。

```
ERR:ARCHIVE FULL
Quit
Largest single...
Variable= 9662
APP      =      0
```

可能显示 **ERR:ARCHIVE FULL** 消息的情况：

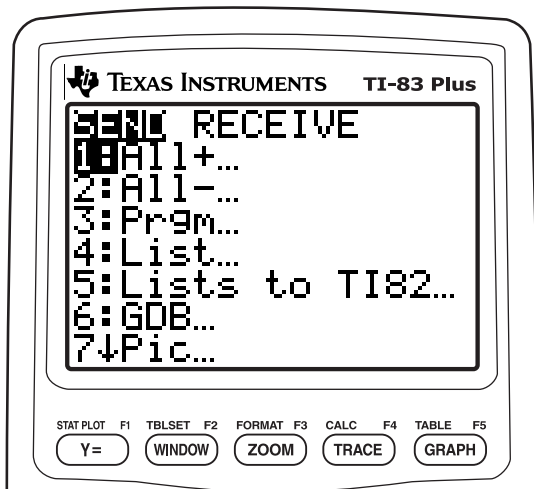
- 当没有足够的空间将变量归档在连续块或单个扇区内时。
  - 当没有足够的空间将应用程序存储在内存的连续块内时。
- 当显示该消息时，它将指出内存中可用于存储变量和应用程序的最大的单个空间。

要解决该问题，请使用 **GarbageCollect** 命令优化内存。如果内存仍然不够，则必须删除变量或应用程序以增加空间。

# 19 通讯链接

## 目录

入门：发送变量.....	2
TI-83 Plus 链接.....	4
选择发送项.....	6
接收项.....	7
传送项.....	8
将项目传送到其他 TI-83 Plus.....	10
将项目传送到 TI-83.....	11
传送数组到 TI-82.....	12
从 TI-82 传送到 TI-83 Plus.....	13
从 TI-83 Plus 传送到 TI-73.....	14
从 TI-73 传送到 TI-83 Plus.....	15
备份内存.....	16



入门是对本章内容的快速介绍，详细信息请阅读本章。

创建并存储变量和矩阵，然后将它们传送到另一台 TI-83 Plus 中。

1. 在发送单元的主屏幕上，按  $5 \square 5 \text{ [STO]} \text{ [ALPHA]} \text{ Q}$ 。按  $\text{[ENTER]}$  将 5.5 存储到 Q。
2. 按  $\text{[2nd]} \text{ [[]] [2nd]} \text{ [[]] 1 \square 2 \text{ [2nd]} \text{ [[]] [2nd]} \text{ [[]] 3 \square 4 \text{ [2nd]} \text{ [[]] [2nd]} \text{ [[]] [STO]} \text{ [2nd]} \text{ [MATRIX] 1}$ 。按  $\text{[ENTER]}$  将矩阵存储到 [A]。
3. 在发送单元，按  $\text{[2nd]} \text{ [MEM]}$  显示 MEMORY 菜单。

```
5.5→Q
5.5
[[1,2][3,4]]→[A]
[[1 2]
 [3 4]]
```

4. 在发送单元，按 **2** 选择 **2:Mem Mgmt/Del**。显示 MEMORY MANAGEMENT/DELETE 菜单。

```
MEMORY
1:About
2:Mem Mgmt/Del...
3:Clear Entries
4:ClrAllLists
5:Archive
6:UnArchive
7↓Reset...
```

5. 在发送单元，按 **5** 以选择 **5:Matrix**。显示 MATRIX 编辑器屏幕。
6. 在发送单元，按  $\text{[ENTER]}$  将 [A] 归档。屏幕显示 \* 以表明 [A] 已归档。

```
RAM FREE 25706
ARC FREE 130927
1:All...
2:Real...
3:Complex...
4>List...
5:Matrix...
6↓Y-Vars...
```

```
RAM FREE 24748
ARC FREE 130882
▶ [A] 47
```

```
RAM FREE 24786
ARC FREE 130832
▶*[A] 47
```

7. 将计算器与链接线缆连接，并将两端插牢。
8. 在接收单元，按 **[2nd]** **[LINK]** **[▶]** 显示 RECEIVE 菜单。按 **1** 选择 **1:Receive**。显示信息 **Waiting...** 并且忙指示器开启。
9. 在发送单元，按 **[2nd]** **[LINK]** 显示 SEND 菜单。
10. 按 **2** 选择 **2:All-**。显示 All- SELECT 屏幕。

```
SEND RECEIVE
1:Receive
```

11. 按 **[◀]** 直到选择光标 (**▶**) 位于 **[A] MATRIX** 旁。按 **[ENTER]**。
12. 按 **[◀]** 直到选择光标位于 **Q REAL** 旁。按 **[ENTER]**。在 **[A]** 和 **Q** 旁边的正方形点表示这些项被选中要发送。

```
SEND RECEIVE
1:All+...
2:All-...
3:Prgm...
4:List...
5:Lists to TI82...
6:GDB...
7↓Pic...
```

```
SELECT TRANSMIT
*[A] MATRIX
Y1 EQU
Y2 EQU
Window WINDOW
RclWindow ZSTO
TblSet TABLE
♦ Q REAL
```

13. 在发送单元，按 **[▶]** 以显示 TRANSMIT 菜单。

```
SELECT TRANSMIT
1:Transmit
```

14. 在发送单元，按 **1** 选择 **1:Transmit** 并开始传送。接收单元显示 **Receiving...** 信息。当项目传送时，接收、发送单元都会显示出所传送变量的名字和类型。

```
Receiving...
*[A] MATRIX
▶ Q REAL
Done
```

### TI-83 Plus 链接性能

TI-83 Plus 具有与另一系统（TI-83 Plus、TI-83、TI-82、TI-73、Calculator-Based Laboratory™ (CBL™) 系统或 Calculator-Based Ranger™ (CBR™) 系统链接或通信的端口。为此，TI-83 Plus 附带单元间的链接线缆。使用 TI-GRAPH LINK™（可选附件），用户还可以将 TI-83 Plus 链接到个人计算机。本章介绍如何与另一计算器链接。

### 链接两个 TI-83 Plus 计算器

用户可以将所有的变量、程序和应用程序传送给另一个 TI-83 Plus，或备份 TI-83 Plus 的整个内存。

- 存储在发送端的 TI-83 Plus RAM 中的变量将被发送到接收端的 TI-83 Plus RAM 中。
- 存储在发送端的 TI-83 Plus 用户数据归档区中的变量和应用程序将被发送到接收端的 TI-83 Plus 用户数据归档区中。

建立通信的软件内置于 TI-83 Plus 中。要将数据从一个 TI-83 Plus 传递到另一个 TI-83 Plus，请按照页 19-6 和页 19-7 中的步骤执行。

### 链接 TI-83 和 TI-83 Plus

如果变量和程序的大小适合 TI-83 Plus 的 RAM，用户可以将其从 TI-83 传送到 TI-83 Plus。TI-83 Plus 的 RAM 略小于 TI-83 的 RAM。

另外，用户可以从 TI-83 Plus 向 TI-83 传送除应用程序、应用程序变量、组合变量、新变量类型或具有新特性的程序（如，**Archive**、**UnArchive**、**Asm**、**AsmComp** 和 **AsmPrgm**）之外的所有变量。

如果 TI-83 Plus 上的归档变量是 TI-83 可以识别和使用的变量类型，那么用户可以将这些变量传送到 TI-83。在传送过程中，它们会被自动发送到 TI-83 的 RAM 中。

建立通信的软件内置于 TI-83 Plus。要从 TI-83 Plus 向 TI-83 传送数据，请按照页 19-11 的步骤执行。

**注意：**用户不能执行从 TI-83 到 TI-83 Plus 的内存备份操作，反之亦然。

---

## 链接 TI-82 和 TI-83 Plus

用户可以从 TI-82 向 TI-83 Plus 传送所有变量和程序。并且，用户还可以从 TI-83 Plus 向 TI-82 传送 L1 到 L6 的实数组（最多可达 99 个元素）。

建立通信的软件内置于 TI-83 Plus。要从 TI-83 Plus 向 TI-82 传送数据，请按照页 19-12 的步骤执行。要从 TI-82 向 TI-83 Plus 传送数据，请按照页 19-13 的步骤执行。

**注意：**用户不能执行从 TI-82 到 TI-83 Plus 的内存备份操作，反之亦然。

## 链接 TI-73 和 TI-83 Plus

用户可以从 TI-73 向 TI-83 Plus，或从 TI-83 Plus 向 TI-73 传送实数、图形、L1 到 L6 的实数组和命名数组。命名数组还能进行交换。然而，TI-73 不能识别  $\theta$ ，所以用户不能将此符号包含在任何发送到 TI-73 的数组名称中。

建立通信的软件内置于 TI-83 Plus。

从 TI-83 Plus 向 TI-73 传送数据，请按照页 19-14 的步骤执行。要从 TI-73 向 TI-83 Plus 传送数据，请按照页 19-15 的步骤执行。

**注意：**用户不能执行从 TI-73 到 TI-83 Plus 的内存备份操作，反之亦然。

## 用线缆连接两台计算器

TI-83 Plus 的链接端口位于计算器底部边缘的中间。

1. 将线缆的一端**非常牢固地**插入到计算器的端口。
2. 将线缆的另一端插入到另一台计算器的端口。

## 链接 CBL/CBR 系统

Calculator-Based Laboratory™ (CBL™) 系统和 Calculator-Based Ranger™ (CBR™) 系统是可选附件，它能够通过单元间线缆与 TI-83 Plus 相连。使用 CBL 或 CBR 和 TI-83 Plus，用户能够采集和分析现实数据。建立通信的软件内置于 TI-83 Plus 中（第 14 章）。

## 链接到个人电脑或 Macintosh®

TI-GRAPH LINK™（需另外购买）是一种链接 TI-83 Plus 以实现与个人电脑通信的附件。

## LINK SEND 菜单

要显示 LINK SEND 菜单，请按 **[2nd]** **[LINK]**。

---

<b>SEND</b>	RECEIVE
1: All+...	显示所有选择的项。
2: All-...	显示所有取消选择的项。
3: Prgm...	显示所有程序名。
4: List...	显示所有数组名。
5: Lists to TI82...	显示数组名 <b>L1-L6</b> 。
6: GDB...	显示所有图象数据库。
7: Pic...	显示所有图形数据类型。
8: Matrix...	显示所有矩阵数据类型。
9: Real...	显示所有实数变量。
0: Complex...	显示所有复数变量。
A: Y-Vars...	显示所有 <b>Y=</b> 变量。
B: String...	显示所有字符串变量。
C: Apps...	显示所有软件应用程序。
D: AppVars...	显示所有软件应用程序变量。
E: Group...	显示所有组合变量。
F: SendId	立即发送计算器 ID 号（用户不必选择 SEND）。
G: SendSW	将升级软件发送给另一 TI-83 Plus。
H: Back Up...	选择 TI-83 Plus 的所有备份。

---

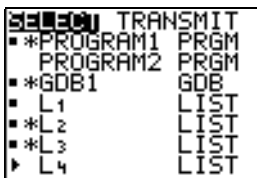
当在 LINK SEND 菜单上选择某项时，就会显示出相应的 SELECT 屏幕。

**注意：**除 All+ SELECT 外，每个 SELECT 屏幕刚显示时没有数据被选中。

## 选择发送项

要在发送单元中选择要发送的项，执行以下步骤：

1. 按 **[2nd]** **[LINK]** 显示 LINK SEND 菜单。
2. 选择描述发送数据类型的菜单项。显示出相应的 SELECT 屏幕。
3. 按 **[◀]** 和 **[▶]** 移动选择光标 (**▶**) 至要选择或取消选择的项上。
4. 按 **[ENTER]** 选择或取消选择项。所选项名用 **■** 标记。



**注意：**项目左侧的星号表示项目已归档（第 18 章）。

5. 重复步骤 3 和 4 选择或取消选择其他项。



## 链接接收菜单

要显示 LINK RECEIVE 菜单，按 **[2nd] [LINK] [↓]**。

---

SEND	RECEIVE
1: Receive	设置单元接收传送数据。

---

## 接收单元

在接收单元，当从 LINK RECEIVE 菜单选择 **1:Receive** 时，将显示出 **Waiting...** 信息和忙指示器。接收单元准备接收传送过来的项。要退出接收模式而不接收项，按 **[ON]** 并从 **Error in Xmit** 菜单选择 **1:Quit**。

要传送项，按照页 19-8 上的步骤进行。

当传送完成后，该单元退出接收模式。可以再选择 **1:Receive** 来接收更多的项。然后接收单元显示出所接收的项列表。按 **[2nd] [QUIT]** 退出接收模式。

## 重名菜单

在传送项期间，如果变量名重复，则在接收单元显示出 **DuplicateName** 菜单。

---

DuplicateName	
1: Rename	提示重命名正接收的变量。
2: Overwrite	覆盖正接收变量中的数据。
3: Omit	跳过正发送变量的传送。
4: Quit	在重复的变量处停止传送。

---

当选择 **1:Rename** 时，将显示出 **Name=** 提示并且字母锁开启。输入新的变量名，然后按 **[ENTER]**，传送重新开始。

当选择 **2:Overwrite** 时，发送单元的数据将覆盖存储在接收单元内的现有数据。传送重新开始。

当选择 **3:Omit** 时，发送单元不再发送重名变量中的数据。继续传送下一项。

当选择 **4:Quit** 时，停止传送并且接收单元退出接收模式。


## 接收单元内存不够

在传送期间，如果接收单元没有足够的内存来接收某项，则在接收单元将显示 **Memory Full** 菜单。

- 要跳过当前传送的项，请选择 **1:Omit**。继续传送下一项。
- 要取消传送并退出接收模式，请选择 **2:Quit**。

## 传送项

在发送单元（页 19-5）选定要发送的项并设置接收单元后（页 19-6），请按下列步骤进行传送：



1. 在发送单元按  显示 TRANSMIT 菜单。

```
SELECT TRANSMIT
1:Transmit
```

2. 确认接收单元中显示出 **Waiting...**，这表示已准备接收（页 19-6）。
3. 按 **[ENTER]** 选择 **1:Transmit**。在发送单元，按传送的顺序逐行显示出各项的名称和类型；然后，在接收单元，按接收的顺序逐行显示出各项的名称和类型。

<pre>*PROGRAM1 PRGM *GDB1 GDB   L1 LIST   *L2 LIST ▶*L3 LIST       Done</pre>	<pre>Receiving... *PROGRAM1 PRGM *GDB1 GDB   L1 LIST   *L2 LIST ▶*L3 LIST       Done</pre>
---	--

**注意：**从发送单元 RAM 中发送的项目被传送到接收单元的 RAM 中。从发送单元的用户数据归档区发送的项目被传送到接收单元的用户数据归档区中。

所选择的项全部传送完毕后，两台计算器上都显示出 **Done** 信息。按  和  从头到尾滚动名字。

## 停止传送

要停止链接传送，请按 **[ON]**。在传送和接收单元中显示 **Error in Xmit** 菜单。要退出错误菜单，请选择 **1:Quit**。

---

## 出错原因

传送错误在下述情况下很快出现：

- 线缆没有和发送单元相连。
- 线缆没有和接收单元相连。  
**注意：**如果连接了线缆，将线缆插牢，然后再试一次。
- 没设置接收单元接收传送。
- 试图在 **TI-73**，**TI-82** 或 **TI-83** 和 **TI-83 Plus** 之间做备份。
- 用户试图从 **TI-83 Plus** 向 **TI-83** 传送的数据中包含 **TI-83** 不能识别的变量和特性。
- **TI-83** 不能识别的新变量类型和特性包括：应用程序、应用程序变量、组合变量、新变量类型或具有新特性的程序（如 **Archive**，**UnArchive**，**SendID**，**SendSW**，**Asm(**，**AsmComp(** 和 **AsmPrgm** )。
- 用户试图从 **TI-83 Plus** 向 **TI-82** 传送实数组 **L1** 到 **L6** 之外的数据，或者没有使用菜单项 **5:Lists to TI82**。
- 用户试图从 **TI-83 Plus** 向 **TI-73** 传送实数、图形、实数数组 **L1** 到 **L6** 之外的数据，或是名称中包括 **θ** 的数组。

虽然在下述两种情形下不会发生传送错误，但可能会妨碍成功传送。

- 试图将 **Get(** 与计算器而不是 **CBL** 或 **CBR** 一起使用。
- 试图将 **GetCalc(** 与 **TI-83** 而不是 **TI-83 Plus** 一起使用。

### 将项目传送到其他 TI-83 Plus

发送或接收数据后，可以将相同的数据从发送单元或者是接收单元反复传送到其他 TI-83 Plus 单元而不需重新选择要发送的数据。当前项仍然被选中。

**注意：** 如果选择 All+ 或 All- 就不能反复传送。

要传送到其他 TI-83 Plus，请按下列步骤进行：

1. 将 TI-83 Plus 设置成接收模式（页 19-7）。
2. 不要选择或取消选择任何新的要传送的项。如果选择或取消选择某一项，那么将清除前一次传送中所有选择项或取消选择项。
3. 将线缆从一台 TI-83 Plus 上拆掉，并将其连接到其他 TI-83 Plus 上。
4. 设置这台 TI-83 Plus 处于接收模式（页 19-7）。
5. 在发送的 TI-83 Plus 上按 **[2nd]** [LINK] 显示 LINK SEND 菜单。
6. 选择用于上一次传送的菜单项。上一次传送的数据仍被选中。
7. 按 **[▶]** 显示 LINK TRANSMIT 菜单。
8. 确认接收单元已准备接收（页 19-7）。
9. 按 **[ENTER]** 选择 **1:Transmit** 并开始传送。

### 将项目传送到 TI-83

用户可以从 TI-83 Plus 向 TI-83 传送除应用程序、应用程序变量、组合变量、新变量类型或具有新特性的程序（如，归档和取消归档）之外的所有变量。

如果 TI-83 Plus 的归档变量是 TI-83 可以识别和使用的变量类型，那么用户可以将这些变量传送到 TI-83。在传送过程中，它们会被自动发送到接收的 TI-83 的 RAM 中。

要将存储在 TI-83 Plus 中数据发送到 TI-83，请按照以下步骤执行。

1. 将 TI-83 设置成接收模式（页 19-7）。
2. 在发送的 TI-83 Plus 上按 **2nd** [LINK] 显示 LINK SEND 菜单。
3. 选择需要传送的菜单项。
4. 按 **▾** 显示 LINK TRANSMIT 菜单。
5. 确认接收单元已设置为接收模式（页 19-7）。
6. 按 **ENTER** 以选择 **1:Transmit** 开始传送。

### 将数组传送到 TI-82

可以从 TI-83 Plus 传送到 TI-82 的数据类型只能是存储在 L1 到 L6 中的实数组数据（每个数组至多可有 99 个元素）。

要将存储在 TI-83 Plus 的数组 L1、L2、L3、L4、L5 或 L6 中的数组数据传送到 TI-82 中，请按下列步骤进行：

1. 将 TI-82 设置成接收模式（页 19-7）。
2. 在发送的 TI-83 Plus 上按 **2nd** [LINK] **5** 选择 **5:Lists to TI82**，显示出 **SELECT** 屏幕。
3. 选择要传送的每一个数组。
4. 按 **▾** 显示 **LINK TRANSMIT** 菜单。
5. 确认接收单元已准备接收（页 19-7）。
6. 按 **ENTER** 选择 **1:Transmit** 并开始传送。

**注意：**如果所选择传送的 TI-83 Plus 数组长度 > 99，则在传送期间接收的 TI-82 将第 99 个元素后的元素舍掉。

### 已解决的 TI-82 和 TI-83 Plus 之间的差异

通常，可以将项目从 TI-82 传送到 TI-83 Plus，但两个产品之间的差异可能影响传送的数据。下表列出了这些差异，当 TI-83 Plus 接收 TI-82 的数据时其内置的软件将自动调整它们。

TI-82	TI-83 Plus
$nMin$	<b>PlotStart</b>
$nStart$	$nMin$
$Un$	<b>u</b>
$Vn$	<b>v</b>
$UnStart$	$u(nMin)$
$VnStart$	$v(nMin)$
$TblMin$	<b>TblStart</b>

例如，如果将一个在命令行中包含  $nStart$  的程序从 TI-82 传送到 TI-83 Plus，然后在接收的 TI-83 Plus 上显示该程序，将会看到  $nMin$  已自动替代了命令行中的  $nStart$ 。

### 未解决的 TI-82 和 TI-83 Plus 之间的差异

TI-83 Plus 中内置的软件不能解决 TI-82 和 TI-83 Plus 之间的某些差异，下面介绍这些差异。在传送完以后必须在 TI-83 Plus 上编辑这些数据来解决这些差异，否则 TI-83 Plus 将曲解这些数据。

TI-83 Plus 重新解释 TI-82 的前缀函数包括开括号，这可能会在传送的表达式中添加括号。

例如，如果将  $\sin X+5$  从 TI-82 传送到 TI-83 Plus，TI-83 Plus 重新将它解释为  $\sin(X+5)$ 。虽然在  $X$  后没有闭合括号，TI-83 Plus 仍将其解释为  $\sin(X+5)$ ，而不是  $5$  和  $\sin(X)$  相加。

如果 TI-83 Plus 不能翻译的 TI-82 指令被传送，当 TI-83 Plus 试图执行这些指令时将会显示出 ERR:INVALID 菜单。例如，在 TI-82 上，按  $\boxed{2nd}$  [Un-1] 后字符组  $Un-1$  粘贴到光标处。但 TI-83 Plus 不能直接将  $Un-1$  翻译为 TI-83 Plus 格式  $u(n-1)$ ，于是就会显示出 ERR:INVALID 菜单。

**注意：**TI-83 Plus 和 TI-82 采用的蕴含乘法规则是不同的。例如， $1/2X$  在 TI-83 Plus 中是作为  $(1/2)*X$  来求值的，而在 TI-82 中是作为  $1/(2*X)$  来求值的（第 2 章）。

### 向 TI-73 传送数组

用户可以从 TI-83 Plus 向 TI-73 传送数组、实数、图形、存储在数组 L1 到 L6 和命名数组中的实数组数据。

**注意：**由于 TI-73 不能识别  $\theta$ ，所以命名数组的数组名称中不能包含  $\theta$ 。

要将存储在 TI-83 Plus 数组 L1，L2，L3，L4，L5 或 L6 中的数组数据传送到 TI-73，请执行以下操作。

1. 设置 TI-73 为接收模式（页 19-7）。
2. 在发送的 TI-83 Plus 上按 **2nd** [LINK] **4** 选择 **4:Lists...**。显示出 SELECT 屏幕。
3. 选择要发送的数组。
4. 在发送的 TI-83 Plus 上按 **▸** 以显示 LINK TRANSMIT 菜单。
5. 确认将接收单元设置为接收模式（页 19-7）。
6. 按 **ENTER** 以选择 **1:Transmit** 并开始传送。



### 将数组传送到 TI-83 Plus

用户可以从 TI-73 向 TI-83 Plus 传送数组、实数、图形、存储在数组 L1 到 L6 和任何命名数组中的实数组数据。

要将存储在 TI-73 数组 L1, L2, L3, L4, L5 或 L6 中的数组数据传送到 TI-83 Plus, 请执行下列操作。

1. 设置 TI-83 Plus 为接收模式 (页 19-7)。
2. 在发送的 TI-73 上按 **[APPS]** 显示 APPLICATIONS 菜单。
3. 在发送的 TI-73 上按 **[ENTER]** 选择 **1:Link** 并显示 LINK SEND 菜单。
4. 选择 **0:Vars to TI83**, 然后选定要发送的数组。
5. 在发送的 TI-73 上按 **[▾]** 以显示 LINK TRANSMIT 菜单。
6. 确认接收单元设置为接收模式 (页 19-7)。
7. 按 **[ENTER]** 以选择 **1:Transmit** 并开始传送。

### 内存备份

要将发送的 TI-83 Plus 内存中的内容准确无误地复制到接收的 TI-83 Plus 的内存中，请将另一单元置成接收模式，然后在接收单元的 LINK SEND 菜单中选择 **H:Back Up**。

- **警告：H:Back Up** 将覆盖接收单元的内存，接收单元内存中所有信息丢失。  
**注意：**如果不想做备份，请选择 **2:Quit** 返回到 LINK SEND 菜单。
- 选择 **1:Transmit** 开始传送。



```
MEMORY BACKUP
1:Transmit
2:Quit
```

### 接收单元

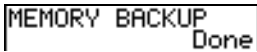
作为安全检查以防止内存信息意外丢失，当接收单元收到备份通知时，将显示信息 **WARNING - Backup**。

- 要继续备份，请选择 **1:Continue**。备份传送开始。
- 要防止备份，请选择 **2:Quit**。

**注意：**在备份的过程中如果返回传送错误，接收单元将重新设置。

### 内存备份结束

当备份结束后，发送和接收计算器都会显示一个确认屏幕。



```
MEMORY BACKUP
Done
```

---

# A 表格和 参考信息

---

## 目录

函数和指令表.....	2
TI-83 Plus 菜单图.....	39
变量.....	50
统计公式.....	51
金融公式.....	55

## 函数和指令表

函数返回一个数值、数组或矩阵。可以在表达式中使用函数。指令启动一个操作。一些函数和指令带有参数，在方括号 ( [ ] ) 内列出了可选参数及相随的逗号。关于某一项的详细信息，如参数描述和限制，请到列在右边的页码处查阅。

从 CATALOG 内，可以将一些函数或指令粘贴到主屏幕或程序编辑器的命令行上。然而，在主屏幕上有些函数和指令是无效的。项在表中的排列顺序和 CATALOG 内的相同。

† 标识仅在程序编辑器中有效的击键，或者在程序编辑器中用于粘贴特定指令的击键。一些击键显示只在程序编辑器中可用的菜单，其他击键只有在程序编辑器中时才粘贴方式、格式或运算表设置指令。

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>abs</b> ( <i>value</i> )	返回一个实数、表达式、数组或矩阵的绝对值。	<b>[MATH]</b> <b>NUM</b> <b>1:abs(</b>	2-13 10-10
<b>abs</b> ( <i>complex value</i> )	返回复数或复数数组的模。	<b>[MATH]</b> <b>CPX</b> <b>5:abs(</b>	2-19
<i>valueA and valueB</i>	如果 <i>valueA</i> 和 <i>valueB</i> 的值都 $\neq 0$ 则返回 1。 <i>valueA</i> 和 <i>valueB</i> 可以是实数、表达式或数组。	<b>[2nd]</b> [TEST] <b>LOGIC</b> <b>1:and</b>	2-26
<b>angle</b> ( <i>value</i> )	返回一个复数或复数数组的极角。	<b>[MATH]</b> <b>CPX</b> <b>4:angle(</b>	2-19
<b>ANOVA</b> ( <i>list1,list2</i> [, <i>list3</i> ,..., <i>list20</i> ])	为比较 2 到 20 个总体的平均值，执行一个单向方差分析。	<b>[STAT]</b> <b>TESTS</b> <b>F:ANOVA(</b>	13-25
<b>Ans</b>	返回上一次结果。	<b>[2nd]</b> [ANS]	1-18
<b>Archive</b>	将指定变量从 RAM 移动到用户数据归档区内存。	<b>[2nd]</b> [MEM] <b>5:Archive</b>	18-10
<b>Asm</b> ( <i>assemblyprgname</i> )	执行一个汇编语言程序。	<b>[2nd]</b> [CATALOG] <b>Asm(</b>	16-24

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项
<b>AsmComp</b> ( <i>prgmASM1</i> , <i>prgmASM2</i> )	编译以 ASCII 码编写的汇编语言程序并将其存为十六进制格式。	$\boxed{2\text{nd}}$ [CATALOG] <b>AsmComp</b> ( 16-24
<b>AsmPrgm</b>	必须将其用作汇编语言程序的第一行。	$\boxed{2\text{nd}}$ [CATALOG] <b>AsmPrgm</b> 16-24
<b>augment</b> ( <i>matrixA</i> , <i>matrixB</i> )	返回将矩阵 <i>matrixB</i> 作为新列追加到矩阵 <i>matrixA</i> 上而成的矩阵。	$\boxed{2\text{nd}}$ [MATRIX] MATH <b>7:augment</b> ( 10-14
<b>augment</b> ( <i>listA</i> , <i>listB</i> )	返回将数组 <i>listB</i> 合并到数组 <i>listA</i> 后的数组。	$\boxed{2\text{nd}}$ [LIST] OPS <b>9:augment</b> ( 11-15
<b>AxesOff</b>	关闭图象的坐标轴。	† $\boxed{2\text{nd}}$ [FORMAT] <b>AxesOff</b> 3-14
<b>AxesOn</b>	开启图象的坐标轴。	† $\boxed{2\text{nd}}$ [FORMAT] <b>AxesOn</b> 3-14
<b>a+bi</b>	设置方式为直角坐标复数方式 (a+bi)。	† $\boxed{\text{MODE}}$ <b>a+bi</b> 1-12
<b>bal</b> ( <i>npmt</i> [, <i>roundvalue</i> ])	用 <b>I%</b> 、 <b>PV</b> 和 <b>PMT</b> 的存储值计算分期付款计划在 <i>npmt</i> 处的余额。 <i>roundvalue</i> 为计算时采用的内部精度。	$\boxed{\text{APPS}}$ <b>1:Finance</b> CALC <b>9:bal</b> ( 14-10
<b>binomcdf</b> ( <i>numtrials</i> , <i>p</i> [, <i>x</i> ])	给定离散二项分布的 <i>numtrials</i> 和每次试验的成功概率 ( <i>p</i> )，计算其在 <i>x</i> 处的累积概率。	$\boxed{2\text{nd}}$ [DISTR] DISTR <b>A:binomcdf</b> ( 13-33

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>binompdf</b> ( <i>numtrials</i> , <i>p</i> [, <i>x</i> ])	给定离散二项分布的 <i>numtrials</i> 和每次试验的成功概率 ( <i>p</i> ), 计算其在 <i>x</i> 处的概率。	$\overline{[2nd]}$ [DISTR] DISTR <b>0:binompdf</b> (	13-33
$\chi^2$ <b>cdf</b> ( <i>lowerbound</i> , <i>upperbound</i> , <i>df</i> )	对指定的自由度 <i>df</i> 计算在 <i>lowerbound</i> 和 <i>upperbound</i> 之间的 $\chi^2$ 分布概率。	$\overline{[2nd]}$ [DISTR] DISTR <b>7:<math>\chi^2</math>cdf</b> (	13-31
$\chi^2$ <b>pdf</b> ( <i>x</i> , <i>df</i> )	对于指定的自由度 <i>df</i> , 计算 $\chi^2$ 分布在指定 <i>x</i> 值处的概率密度函数 (pdf)。	$\overline{[2nd]}$ [DISTR] DISTR <b>6:<math>\chi^2</math>pdf</b> (	13-31
$\chi^2$ - <b>Test</b> ( <i>observedmatrix</i> , <i>expectedmatrix</i> [, <i>drawflag</i> ])	执行 $\chi^2$ 检验。 <i>drawflag</i> =1 画出结果曲线; <i>drawflag</i> =0 计算结果。	† [STAT] TESTS <b>C:<math>\chi^2</math>-Test</b> (	13-22
<b>Circle</b> ( <i>X</i> , <i>Y</i> , <i>radius</i> )	画圆心为 ( <i>X</i> , <i>Y</i> )、半径为 <i>radius</i> 的圆。	$\overline{[2nd]}$ [DRAW] DRAW <b>9:Circle</b> (	8-11
<b>Clear Entries</b>	清除最后输入项存储区域内的内容。	$\overline{[2nd]}$ [MEM] MEMORY <b>3:Clear Entries</b>	18-5
<b>ClrAllLists</b>	设置内存中所有数组的维数为 <b>0</b> 。	$\overline{[2nd]}$ [MEM] MEMORY <b>4:ClrAllLists</b>	18-5
<b>ClrDraw</b>	清除图象或图中所有已画的元素。	$\overline{[2nd]}$ [DRAW] DRAW <b>1:ClrDraw</b>	8-4
<b>ClrHome</b>	清除主屏幕。	† [PRGM] I/O <b>8:ClrHome</b>	16-21
<b>ClrList</b> <i>listname1</i> [, <i>listname2</i> , ..., <i>listname n</i> ]	将一个或多个 <i>listname</i> 的维数设置为 <b>0</b> 。	[STAT] EDIT <b>4:ClrList</b>	12-20
<b>ClrTable</b>	清除运算表中所有的值。	† [PRGM] I/O <b>9:ClrTable</b>	16-21

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>conj(value)</b>	返回一个复数或复数数组的共轭复数。	<b>[MATH]</b> <b>CPX</b> <b>1:conj(</b>	2-18
<b>Connected</b>	设置连线绘图方式；将所有 Y= 编辑器的图象样式重设为 \。	† <b>[MODE]</b> <b>Connected</b>	1-11
<b>CoordOff</b>	关闭光标坐标值的显示。	† <b>[2nd]</b> [FORMAT] <b>CoordOff</b>	3-14
<b>CoordOn</b>	开启光标坐标值的显示。	† <b>[2nd]</b> [FORMAT] <b>CoordOn</b>	3-14
<b>cos(value)</b>	返回一个实数、表达式或数组的余弦值。	<b>[COS]</b>	2-3
<b>cos<sup>-1</sup>(value)</b>	返回一个实数、表达式或数组的反余弦值。	<b>[2nd]</b> [COS <sup>-1</sup> ]	2-3
<b>cosh(value)</b>	返回一个实数、表达式或数组的双曲余弦值。	<b>[2nd]</b> [CATALOG] <b>cosh(</b>	15-10
<b>cosh<sup>-1</sup>(value)</b>	返回一个实数、表达式或数组的反双曲余弦值。	<b>[2nd]</b> [CATALOG] <b>cosh<sup>-1</sup>(</b>	15-10
<b>CubicReg</b> [ <i>Xlistname</i> , <i>Ylistname</i> , <i>freqlist</i> , <i>regequ</i> ]	用三次回归模型拟合频数为 <i>freqlist</i> 的数据 <i>Xlistname</i> 和 <i>Ylistname</i> ，并将回归方程保存到 <i>regequ</i> 。	<b>[STAT]</b> <b>CALC</b> <b>6:CubicReg</b>	12-26
<b>cumSum(list)</b>	返回 <i>list</i> 中元素累积求和的数组，从第 1 个元素开始。	<b>[2nd]</b> [LIST] <b>OPS</b> <b>6:cumSum(</b>	11-12
<b>cumSum(matrix)</b>	返回 <i>matrix</i> 元素的累积求和的矩阵。在返回矩阵中的每个元素都是 <i>matrix</i> 列从上到下的累积求和。	<b>[2nd]</b> [MATRIX] <b>MATH</b> <b>0:cumSum(</b>	10-15
<b>dbd(date1,date2)</b>	用实际天数计算法来计算 <i>date1</i> 和 <i>date2</i> 之间的天数。	<b>[APPS]</b> <b>1:Finance</b> <b>CALC</b> <b>D:dbd(</b>	14-14
<b>value</b> → <b>Dec</b>	用十进制格式来显示实数、复数、表达式、数组或矩阵。	<b>[MATH]</b> <b>MATH</b> <b>2:→Dec</b>	2-5

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项
<b>Degree</b>	设置角度方式。	† [MODE] <b>Degree</b> 1-11
<b>DelVar</b> <i>variable</i>	从内存中删除 <i>variable</i> 的内容。	† [PRGM] CTL <b>G:DelVar</b> 16-16
<b>DependAsk</b>	设置运算表以索取因变量的值。	† [2nd] [TBLSET] <b>Depend: Ask</b> 7-3
<b>DependAuto</b>	设置运算表来自动生成因变量的值。	† [2nd] [TBLSET] <b>Depend: Auto</b> 7-3
<b>det</b> ( <i>matrix</i> )	返回 <i>matrix</i> 的行列式。	[2nd] [MATRX] MATH <b>1:det(</b> 10-12
<b>DiagnosticOff</b>	设置诊断关闭方式; $r$ 、 $r^2$ 和 $R^2$ 不作为回归模型的结果来显示。	[2nd] [CATALOG] <b>DiagnosticOff</b> 12-23
<b>DiagnosticOn</b>	设置诊断开启方式; $r$ 、 $r^2$ 和 $R^2$ 作为回归模型的结果来显示。	[2nd] [CATALOG] <b>DiagnosticOn</b> 12-23
<b>dim</b> ( <i>listname</i> )	返回 <i>listname</i> 的维数。	[2nd] [LIST] OPS <b>3:dim(</b> 11-11
<b>dim</b> ( <i>matrixname</i> )	象数组一样, 返回 <i>matrixname</i> 的维数。	[2nd] [MATRX] MATH <b>3:dim(</b> 10-12
<i>length</i> → <b>dim</b> ( <i>listname</i> )	对新的或者已有的 <i>listname</i> 赋予新维数 ( <i>length</i> )。	[2nd] [LIST] OPS <b>3:dim(</b> 11-11
{ <i>rows,columns</i> }→ <b>dim</b> ( <i>matrixname</i> )	对新的或者已有的 <i>matrixname</i> 赋予新维数。	[2nd] [MATRX] MATH <b>3:dim(</b> 10-13
<b>Disp</b>	显示主屏幕。	† [PRGM] I/O <b>3:Disp</b> 16-19
<b>Disp</b> [ <i>valueA,valueB,</i> <i>valueC,...,value n</i> ]	显示每一个值。	† [PRGM] I/O <b>3:Disp</b> 16-19



函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项
<b>DispGraph</b>	显示图象。	† [PRGM] I/O <b>4:DispGraph</b> 16-20
<b>DispTable</b>	显示运算表。	† [PRGM] I/O <b>5:DispTable</b> 16-20
<i>value</i> → <b>DMS</b>	以 DMS 格式显示 <i>value</i> 。	[2nd] [ANGLE] ANGLE <b>4:→DMS</b> 2-24
<b>Dot</b>	设置点绘图方式；将所有 Y= 编辑器的图象样式重设为 `.`。	† [MODE] <b>Dot</b> 1-11
<b>DrawF</b> <i>expression</i>	在图象上绘制 <i>expression</i> （对于 <b>X</b> ）。	[2nd] [DRAW] DRAW <b>6:DrawF</b> 8-9
<b>DrawInv</b> <i>expression</i>	通过在 <b>y</b> 轴上确定 <b>X</b> 值和 <b>x</b> 轴上确定 <b>Y</b> 值来绘制 <i>expression</i> 的反函数图象。	[2nd] [DRAW] DRAW <b>8:DrawInv</b> 8-9
<b>:DS&lt;</b> ( <i>variable,value</i> ) <b>:commandA</b> <b>:commands</b>	<i>variable</i> 减少 1；如果 <i>variable</i> < <i>value</i> 则跳过 <b>commandA</b> 。	† [PRGM] CTL <b>B:DS&lt;</b> ( 16-15
<b>e^(power)</b>	返回以常数 <b>e</b> 为底的乘幂。 <i>power</i> 为幂指数。	[2nd] [e <sup>x</sup> ] 2-4
<b>e^(list)</b>	返回一个数组，元素为以常数 <b>e</b> 为底，以数组 <i>list</i> 的元素为幂指数的乘幂。	[2nd] [e <sup>x</sup> ] 2-4
Exponent: <i>value</i> <b>E</b> <i>exponent</i>	返回 <i>value</i> 乘以 10 的 <i>exponent</i> 次幂。	[2nd] [EE] 1-7
Exponent: <i>list</i> <b>E</b> <i>exponent</i>	返回数组，元素为数组 <i>list</i> 对应元素乘以 10 的 <i>exponent</i> 次幂。	[2nd] [EE] 1-7
Exponent: <i>matrix</i> <b>E</b> <i>exponent</i>	返回矩阵，元素为矩阵 <i>matrix</i> 对应元素乘以 10 的 <i>exponent</i> 次幂。	[2nd] [EE] 1-7
► <b>Eff</b> ( <i>nominal rate, compounding periods</i> )	计算实际利率。	[APPS] <b>1:Finance</b> CALC <b>C:►Eff</b> ( 14-13
<b>Else</b> 请参阅 <b>If:Then:Else</b>		

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>End</b>	标志 <b>For</b> 、 <b>If-Then-Else</b> 、 <b>Repeat</b> 或 <b>While</b> 循环结束。	† [PRGM] CTL <b>7:End</b>	16-11
<b>Eng</b>	设置工程显示方式。	† [MODE] <b>Eng</b>	1-10
<b>EquString(Y= var,Strn)</b>	将 Y= var 的内容转换为字符串并保存在 <b>Strn</b> 中。	[2nd] [CATALOG] <b>EquString(</b>	15-7
<b>expr(string)</b>	将 <i>string</i> 转换为表达式并执行它。	[2nd] [CATALOG] <b>expr(</b>	15-7
<b>ExpReg [Xlistname, Ylistname,freqlist,regequ]</b>	用指数回归模型拟合频数为 <i>freqlist</i> 的数据 <i>Xlistname</i> 和 <i>Ylistname</i> ，并把回归方程保存到 <i>regequ</i> 。	[STAT] CALC <b>0:ExpReg</b>	12-26
<b>ExprOff</b>	在 TRACE 期间关闭表达式的显示。	† [2nd] [FORMAT] <b>ExprOff</b>	3-14
<b>ExprOn</b>	在 TRACE 期间开启表达式的显示。	† [2nd] [FORMAT] <b>ExprOn</b>	3-14
<b>Fcdf(lowerbound, upperbound, numerator df, denominator df)</b>	对所指定的 <i>numerator df</i> (自由度) 和 <i>denominator df</i> 计算 <i>lowerbound</i> 和 <i>upperbound</i> 之间的 F 分布概率。	[2nd] [DISTR] DISTR <b>9:Fcdf(</b>	13-32
<b>Fill(value,matrixname)</b>	将 <i>value</i> 保存到 <i>matrixname</i> 内的每个元素中。	[2nd] [MATRIX] MATH <b>4:Fill(</b>	10-13
<b>Fill(value,listname)</b>	将 <i>value</i> 保存到 <i>listname</i> 内的每个元素中。	[2nd] [LIST] OPS <b>4:Fill(</b>	11-11
<b>Fix #</b>	设置定点十进制表示方式为 # 个小数位。	† [MODE] <b>0123456789</b> (选择一个)	1-10
<b>Float</b>	设置浮点十进制表示方式。	† [MODE] <b>Float</b>	1-10

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>fMax</b> ( <i>expression,variable, lower,upper</i> [, <i>tolerance</i> ])	返回表达式 <i>expression</i> 在下限 <i>lower</i> 和上限 <i>upper</i> 区间内取极大值时变量 <i>variable</i> 的值。给定的误差为 <i>tolerance</i> 。	$\boxed{\text{MATH}}$ MATH 7:fMax(	2-6
<b>fMin</b> ( <i>expression,variable, lower,upper</i> [, <i>tolerance</i> ])	返回表达式 <i>expression</i> 在下限 <i>lower</i> 和上限 <i>upper</i> 区间内取极小值时变量 <i>variable</i> 的值。给定的误差为 <i>tolerance</i> 。	$\boxed{\text{MATH}}$ MATH 6:fMin(	2-6
<b>fnInt</b> ( <i>expression,variable, lower,upper</i> [, <i>tolerance</i> ])	返回表达式 <i>expression</i> 对变量 <i>variable</i> 在指定下限 <i>lower</i> 和上限 <i>upper</i> 内的数值积分，指定误差为 <i>tolerance</i> 。	$\boxed{\text{MATH}}$ MATH 9:fnInt(	2-7
<b>FnOff</b> [ <i>function#</i> , <i>function#</i> ,..., <i>function n</i> ]	取消选择所有的 Y= 函数或指定的 Y= 函数。	$\boxed{\text{VARS}}$ Y-VARS On/Off 2:FnOff	3-8
<b>FnOn</b> [ <i>function#</i> , <i>function#</i> ,..., <i>function n</i> ]	选择所有的 Y= 函数或指定的 Y= 函数。	$\boxed{\text{VARS}}$ Y-VARS On/Off 1:FnOn	3-8
<b>:For</b> ( <i>variable,begin,end</i> [, <i>increment</i> ]) : <i>commands</i> : <b>End</b> : <i>commands</i>	执行 <i>commands</i> 到 <b>End</b> ，变量 <i>variable</i> 从 <i>begin</i> 开始，每次增加 <i>increment</i> 直到变量 <i>variable</i> > <i>end</i> 。	† $\boxed{\text{PRGM}}$ CTL 4:For(	16-11
<b>fPart</b> ( <i>value</i> )	返回实数、复数、表达式、数组或矩阵的分数部分。	$\boxed{\text{MATH}}$ NUM 4:fPart(	2-14 10-11
<b>Fpdf</b> ( <i>x,numerator df</i> , <i>denominator df</i> )	根据所指定的 <i>numerator df</i> (自由度) 和 <i>denominator df</i> 计算 <i>lowerbound</i> 和 <i>upperbound</i> 间的 F 分布概率。	$\boxed{2\text{nd}}$ [DISTR] DISTR 8:Fpdf(	13-32

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
$value \rightarrow \text{Frac}$	以最简分数形式显示实数、复数、表达式、数组或矩阵。	$\overline{\text{MATH}}$ $\text{MATH}$ $1:\text{Frac}$	2-5
<b>Full</b>	设置全屏方式。	$\dagger \overline{\text{MODE}}$ <b>Full</b>	1-12
<b>Func</b>	设置函数图象方式。	$\dagger \overline{\text{MODE}}$ <b>Func</b>	1-11
<b>GarbageCollect</b>	显示 <b>garbage collection</b> 菜单以允许清除未用的归档内存。	$\overline{[2\text{nd}]}$ $\overline{[\text{CATALOG}]}$ <b>GarbageCollect</b>	18-19
$\text{gcd}(valueA, valueB)$	返回 $valueA$ 和 $valueB$ (可以是实数或数组) 的最大公因子。	$\overline{\text{MATH}}$ $\text{NUM}$ $9:\text{gcd}()$	2-15
$\text{geometcdf}(p, x)$	对指定离散几何分布的成功概率 $p$ , 计算其在 $x$ (第一次成功时的试验次数) 处的累积概率。	$\overline{[2\text{nd}]}$ $\overline{[\text{DISTR}]}$ $\text{DISTR}$ <b>E:geometcdf()</b>	13-34
$\text{geometpdf}(p, x)$	对指定离散几何分布的成功概率 $p$ , 计算其在 $x$ (第一次成功时的试验次数) 处的概率。	$\overline{[2\text{nd}]}$ $\overline{[\text{DISTR}]}$ $\text{DISTR}$ <b>D:geometpdf()</b>	13-34
<b>Get(variable)</b>	从 CBL 或者 CBR 系统获得数据并将其保存在 $variable$ 中。	$\dagger \overline{[\text{PRGM}]}$ $\text{I/O}$ <b>A:Get()</b>	16-22
<b>GetCalc(variable)</b>	获得另一台 TI-83 Plus 上 $variable$ 的内容并将之保存到接收此内容的 TI-83 Plus 的 $variable$ 中。	$\dagger \overline{[\text{PRGM}]}$ $\text{I/O}$ <b>0:GetCalc()</b>	16-22
<b>getKey</b>	返回当前击键的键代码, 如果没有键按下, 则返回 <b>0</b> 。	$\dagger \overline{[\text{PRGM}]}$ $\text{I/O}$ <b>7:getKey</b>	16-21
<b>Goto label</b>	转移控制到 $label$ 。	$\dagger \overline{[\text{PRGM}]}$ $\text{CTL}$ <b>0:Goto</b>	16-14

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>GraphStyle</b> ( <i>function#</i> , <i>graphstyle#</i> )	为 <i>function#</i> 设置 <i>graphstyle#</i> 。	† [PRGM] CTL H:GraphStyle( 16-16	
<b>GridOff</b>	关闭网格格式。	† [2nd] [FORMAT] GridOff	3-14
<b>GridOn</b>	开启网格格式。	† [2nd] [FORMAT] GridOn	3-14
<b>G-T</b>	设置图象 — 运算表垂直拆分 屏幕方式。	† [MODE] G-T	1-12
<b>Horiz</b>	设置水平拆分屏幕方式。	† [MODE] Horiz	1-12
<b>Horizontal</b> <i>y</i>	在 <i>y</i> 处画出水平线。	[2nd] [DRAW] DRAW 3:Horizontal	8-6
<b>identity</b> ( <i>dimension</i> )	返回 <i>dimension</i> 行 × <i>dimension</i> 列的单位矩阵。	[2nd] [MATRIX] MATH 5:identity( 10-13	
<b>:If</b> <i>condition</i> <b>:commandA</b> <b>:commands</b>	如果 <i>condition</i> = 0 (为假), 则跳过 <i>commandA</i> 。	† [PRGM] CTL 1:If	16-10
<b>:If</b> <i>condition</i> <b>:Then</b> <b>:commands</b> <b>:End</b> <b>:commands</b>	如果 <i>condition</i> = 1 (为真), 则执行从 <b>Then</b> 到 <b>End</b> 的 <i>commands</i> 。	† [PRGM] CTL 2:Then	16-10
<b>:If</b> <i>condition</i> <b>:Then</b> <b>:commands</b> <b>:Else</b> <b>:commands</b> <b>:End</b> <b>:commands</b>	如果 <i>condition</i> = 1 (为真), 则执行从 <b>Then</b> 到 <b>Else</b> 的 <i>commands</i> , 如果 <i>condition</i> = 0 (为假), 则执行从 <b>Else</b> 到 <b>End</b> 的 <i>commands</i> 。	† [PRGM] CTL 3:Else	16-11
<b>imag</b> ( <i>value</i> )	返回复数或复数数组的虚数 (非实数) 部分。	[MATH] CPX 3:imag( 2-18	

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项
<b>IndpntAsk</b>	设置运算表以索取自变量的值。	† [2nd] [TBLSET] <b>Indpnt: Ask</b> 7-3
<b>IndpntAuto</b>	设置运算表以自动生成自变量的值。	† [2nd] [TBLSET] <b>Indpnt: Auto</b> 7-3
<b>Input</b>	显示图象。	† [PRGM] I/O <b>1:Input</b> 16-17
<b>Input</b> [ <i>variable</i> ] <b>Input</b> [" <i>text</i> ", <i>variable</i> ]	提示将值保存到 <i>variable</i> 中。	† [PRGM] I/O <b>1:Input</b> 16-18
<b>Input</b> [ <b>Strn</b> , <i>variable</i> ]	显示 <b>Strn</b> 并将输入的值保存到 <i>variable</i> 中。	† [PRGM] I/O <b>1:Input</b> 16-18
<b>inString</b> ( <i>string</i> , <i>substring</i> [, <i>start</i> ])	返回从 <i>start</i> 开始 <i>substring</i> 的第一个字符在 <i>string</i> 中的位置。	[2nd] [CATALOG] <b>inString(</b> 15-7
<b>int</b> ( <i>value</i> )	返回 ≤ 实数、复数、表达式、数组或矩阵的最大整数。	[MATH] NUM <b>5:int(</b> 2-14 10-11
<b>ΣInt</b> ( <i>pmt1</i> , <i>pmt2</i> [, <i>roundvalue</i> ])	计算分期付款计划表中 <i>pmt1</i> 和 <i>pmt2</i> 之间的利息总额, 结果舍入到 <i>roundvalue</i> 。	[APPS] <b>1:Finance</b> CALC <b>A:ΣInt(</b> 14-10
<b>invNorm</b> ( <i>area</i> [, $\mu$ , $\sigma$ ])	对给定的正态分布曲线下的 <i>area</i> 计算反正态累积分布函数, 该正态分布曲线由 $\mu$ 和 $\sigma$ 指定。	[2nd] [DISTR] DISTR <b>3:invNorm(</b> 13-30
<b>iPart</b> ( <i>value</i> )	返回实数、复数、表达式、数组或矩阵的整数部分。	[MATH] NUM <b>3:iPart(</b> 2-14 10-11

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项
<b>irr</b> ( <i>CF0,CFList[,CFFreq]</i> )	返回现金流转的净现值为零时的利率。	[APPS] 1:Finance CALC 8:irr( 14-8
<b>:IS&gt;</b> ( <i>variable,value</i> ) <b>:commandA</b> <b>:commands</b>	<i>variable</i> 增加 1；如果 <i>variable</i> > <i>value</i> 则跳过 <i>commandA</i> 。	† [PRGM] CTL A:IS>( 16-14
<b>Llistname</b>	标识紧接着的 1 到 5 个字符作为用户创建的数组名。	[2nd] [LIST] OPS B:l 11-16
<b>LabelOff</b>	关闭坐标轴标记。	† [2nd] [FORMAT] LabelOff 3-14
<b>LabelOn</b>	开启坐标轴标记。	† [2nd] [FORMAT] LabelOn 3-14
<b>Lbl label</b>	创建一个或两个字符的 <i>label</i> 。	† [PRGM] CTL 9:Lbl 16-14
<b>lcm</b> ( <i>valueA,valueB</i> )	返回 <i>valueA</i> 和 <i>valueB</i> （可以为实数或数组）的最小公倍数。	[MATH] NUM 8:lcm( 2-15
<b>length</b> ( <i>string</i> )	返回 <i>string</i> 中字符数。	[2nd] [CATALOG] length( 15-8
<b>Line</b> ( <i>X1,Y1,X2,Y2</i> )	从 ( <i>X1,Y1</i> ) 到 ( <i>X2,Y2</i> ) 画一条直线。	[2nd] [DRAW] DRAW 2:Line( 8-5
<b>Line</b> ( <i>X1,Y1,X2,Y2,0</i> )	将从 ( <i>X1,Y1</i> ) 到 ( <i>X2,Y2</i> ) 的直线去掉。	[2nd] [DRAW] DRAW 2:Line( 8-5

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项
<b>LinReg(a+bx)</b> [ <i>Xlistname</i> , <i>Ylistname</i> , <i>freqlist</i> , <i>regequ</i> ]	用线性回归模型拟合频数为 <i>freqlist</i> 的数据 <i>Xlistname</i> 和 <i>Ylistname</i> , 并将回归方程保 存到 <i>regequ</i> 。	[STAT] CALC <b>8:LinReg(a+bx)</b> 12-26
<b>LinReg(ax+b)</b> [ <i>Xlistname</i> , <i>Ylistname</i> , <i>freqlist</i> , <i>regequ</i> ]	用线性回归模型拟合频数为 <i>freqlist</i> 的数据 <i>Xlistname</i> 和 <i>Ylistname</i> , 并将回归方程保 存到 <i>regequ</i> 。	[STAT] CALC <b>4:LinReg(ax+b)</b> 12-25
<b>LinRegTTest</b> [ <i>Xlistname</i> , <i>Ylistname</i> , <i>freqlist</i> , <i>alternative</i> , <i>regequ</i> ]	执行线性回归和 <i>t</i> 检验。 <i>alternative=-1</i> 是 <, <i>alternative=0</i> 是 ≠, <i>alternative=1</i> 是 >。	† [STAT] TESTS <b>E:LinRegTTest</b> 13-24
<b>ΔList(list)</b>	返回一个元素为 <i>list</i> 中相邻元 素之差的数组。	[2nd] [LIST] OPS <b>7:ΔList(</b> 11-12
<b>List►matr(listname1,..., listname n,matrix-name)</b>	将每个指定 <i>listname</i> 内的元 素按列填充到 <i>matrixname</i> 中。	[2nd] [LIST] OPS <b>0&gt;List►matr(</b> 10-14 11-15
<b>ln(value)</b>	返回实数、复数、表达式或数 组的自然对数。	[LN] 2-4
<b>LnReg</b> [ <i>Xlistname</i> , <i>Ylistname</i> , <i>freqlist</i> , <i>regequ</i> ]	用对数回归模型拟合频数为 <i>freqlist</i> 的数据 <i>Xlistname</i> 和 <i>Ylistname</i> , 并将回归方程保 存到 <i>regequ</i> 。	[STAT] CALC <b>9:LnReg</b> 12-26
<b>log(value)</b>	返回实数、复数、表达式或数 组的对数。	[LOG] 2-4



函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>Logistic</b> [ <i>Xlistname</i> , <i>Ylistname</i> , <i>freqlist</i> , <i>regequ</i> ]	用逻辑回归模型拟合频数为 <i>freqlist</i> 的数据 <i>Xlistname</i> 和 <i>Ylistname</i> , 并将回归方程保存到 <i>regequ</i> 。	<b>[STAT]</b> <b>CALC</b> <b>B:Logistic</b>	12-27
<b>Matr ▶ list</b> ( <i>matrix</i> , <i>listnameA</i> ,..., <i>listname n</i> )	将 <i>matrix</i> 中每一列的元素作为每个 <i>listname</i> 的元素。	<b>[2nd]</b> [LIST] <b>OPS</b> <b>A:Matr ▶ list(</b>	10-14 11-16
<b>Matr ▶ list</b> ( <i>matrix</i> , <i>column#</i> , <i>listname</i> )	用 <i>matrix</i> 中所指定 <i>column#</i> 的元素填充 <i>listname</i> 。	<b>[2nd]</b> [LIST] <b>OPS</b> <b>A:Matr ▶ list(</b>	10-14 11-16
<b>max</b> ( <i>valueA</i> , <i>valueB</i> )	返回 <i>valueA</i> 和 <i>valueB</i> 中较大的一个。	<b>[MATH]</b> <b>NUM</b> <b>7:max(</b>	2-15
<b>max</b> ( <i>list</i> )	返回 <i>list</i> 中最大的实数或复数元素。	<b>[2nd]</b> [LIST] <b>MATH</b> <b>2:max(</b>	11-17
<b>max</b> ( <i>listA</i> , <i>listB</i> )	返回实数或复数数组, 元素为 <i>listA</i> 和 <i>listB</i> 中每对元素中较大的一个。	<b>[2nd]</b> [LIST] <b>MATH</b> <b>2:max(</b>	11-17
<b>max</b> ( <i>value</i> , <i>list</i> )	返回实数或复数数组, 元素为 <i>value</i> 和每个 <i>list</i> 元素中较大的一个。	<b>[2nd]</b> [LIST] <b>MATH</b> <b>2:max(</b>	11-17
<b>mean</b> ( <i>list</i> [, <i>freqlist</i> ])	返回频数为 <i>freqlist</i> 的数组 <i>list</i> 的平均值。	<b>[2nd]</b> [LIST] <b>MATH</b> <b>3:mean(</b>	11-17
<b>median</b> ( <i>list</i> [, <i>freqlist</i> ])	返回频数为 <i>freqlist</i> 的数组 <i>list</i> 的中位数。	<b>[2nd]</b> [LIST] <b>MATH</b> <b>4:median(</b>	11-17
<b>Med-Med</b> [ <i>Xlistname</i> , <i>Ylistname</i> , <i>freqlist</i> , <i>regequ</i> ]	用中位数-中位数模型拟合频数为 <i>freqlist</i> 的数据 <i>Xlistname</i> 和 <i>Ylistname</i> , 并将回归方程保存到 <i>regequ</i> 。	<b>[STAT]</b> <b>CALC</b> <b>3:Med-Med</b>	12-25
<b>Menu</b> ("title", <i>"text1"</i> , <i>label1</i> [,..., <i>"text7"</i> , <i>label7</i> ])	在程序执行期间, 生成多至有 7 个项的菜单。	<b>[PRGM]</b> <b>CTL</b> <b>C:Menu(</b>	16-15

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
$\min(\text{valueA}, \text{valueB})$	返回 $\text{valueA}$ 和 $\text{valueB}$ 中较小的一个。	$\boxed{\text{MATH}}$ NUM 6:MIN(	2-15
$\min(\text{list})$	返回 $\text{list}$ 中最小的实数或复数元素。	$\boxed{2\text{nd}}$ [LIST] MATH 1:MIN(	11-17
$\min(\text{listA}, \text{listB})$	返回实数或复数数组, 元素为 $\text{listA}$ 和 $\text{listB}$ 的每对元素中较小的一个。	$\boxed{2\text{nd}}$ [LIST] MATH 1:MIN(	11-17
$\min(\text{value}, \text{list})$	返回实数或复数数组, 元素为 $\text{value}$ 和每个 $\text{list}$ 元素中较小的一个。	$\boxed{2\text{nd}}$ [LIST] MATH 1:MIN(	11-17
$\text{valueA nCr valueB}$	返回每次从 $\text{valueA}$ 中取出 $\text{valueB}$ 的组合数。	$\boxed{\text{MATH}}$ PRB 3:nCr	2-21
$\text{value nCr list}$	返回一个数组, 元素是每次从 $\text{value}$ 中取出数组 $\text{list}$ 对应元素的组合数。	$\boxed{\text{MATH}}$ PRB 3:nCr	2-21
$\text{list nCr value}$	返回一个数组, 元素是每次从数组 $\text{list}$ 对应元素中取出 $\text{value}$ 的组合数。	$\boxed{\text{MATH}}$ PRB 3:nCr	2-21
$\text{listA nCr listB}$	返回一个数组, 元素是每次从数组 $\text{listA}$ 对应元素中取出数组 $\text{listB}$ 对应元素的组合数。	$\boxed{\text{MATH}}$ PRB 3:nCr	2-21
$\text{nDeriv}(\text{expression}, \text{variable}, \text{value}, [\epsilon])$	根据所指定的 $\epsilon$ , 返回 $\text{expression}$ 在 $\text{value}$ 处关于 $\text{variable}$ 的近似数值微分。	$\boxed{\text{MATH}}$ MATH 8:nDeriv(	2-7
$\text{Nom}(\text{effective rate}, \text{compounding periods})$	计算额定利率。	$\boxed{\text{APPS}}$ 1:Finance CALC B:Nom(	14-13
Normal	设置标准显示方式。	† $\boxed{\text{MODE}}$ Normal	1-10

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项
<b>normalcdf</b> ( <i>lowerbound</i> , <i>upperbound</i> [, $\mu$ , $\sigma$ ])	对所指定的 $\mu$ 和 $\sigma$ ，计算 <i>lowerbound</i> 和 <i>upperbound</i> 间的正态分布概率。	$\boxed{2\text{nd}}$ [DISTR] DISTR <b>2:normalcdf</b> ( 13-30
<b>normalpdf</b> ( <i>x</i> [, $\mu$ , $\sigma$ ])	对所指定的 $\mu$ 和 $\sigma$ 并根据指定的 <i>x</i> 值计算正态分布的概率密度函数。	$\boxed{2\text{nd}}$ [DISTR] DISTR <b>1:normalpdf</b> ( 13-29
<b>not</b> ( <i>value</i> )	如果 <i>value</i> $\neq 0$ ，则返回 <b>0</b> 。 <i>value</i> 可以是实数、表达式或数组。	$\boxed{2\text{nd}}$ [TEST] LOGIC <b>4:not</b> ( 2-26
<i>valueA</i> <b>nPr</b> <i>valueB</i>	返回每次从 <i>valueA</i> 中取出 <i>valueB</i> 的排列数。	$\boxed{\text{MATH}}$ PRB <b>2:nPr</b> 2-21
<i>value</i> <b>nPr</b> <i>list</i>	返回一个数组，元素是每次从 <i>value</i> 中取出数组 <i>list</i> 对应元素的排列数。	$\boxed{\text{MATH}}$ PRB <b>2:nPr</b> 2-21
<i>list</i> <b>nPr</b> <i>value</i>	返回一个数组，元素是每次从数组 <i>list</i> 对应元素中取出 <i>value</i> 的排列数。	$\boxed{\text{MATH}}$ PRB <b>2:nPr</b> 2-21
<i>listA</i> <b>nPr</b> <i>listB</i>	返回一个数组，元素是每次从数组 <i>listA</i> 对应元素中取出数组 <i>listB</i> 对应元素的排列数。	$\boxed{\text{MATH}}$ PRB <b>2:nPr</b> 2-21
<b>npv</b> ( <i>interest rate</i> , <i>CFO</i> , <i>CFL</i> <i>ist</i> [, <i>CFF</i> <i>req</i> ])	计算现金流入和流出的现值总和。	$\boxed{\text{APPS}}$ <b>1:Finance</b> CALC <b>7:npv</b> ( 14-9
<i>valueA</i> <b>or</b> <i>valueB</i>	如果 <i>valueA</i> 或 <i>valueB</i> $\neq 0$ ，则返回 <b>1</b> 。 <i>valueA</i> 和 <i>valueB</i> 可以是实数、表达式或数组。	$\boxed{2\text{nd}}$ [TEST] LOGIC <b>2:or</b> 2-26

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项
<b>Output</b> ( <i>row,column,"text"</i> )	从指定的 <i>row</i> 和 <i>column</i> 开始显示 <i>text</i> 。	† [PRGM] I/O <b>6:Output</b> ( 16-20
<b>Output</b> ( <i>row,column,value</i> )	从指定的 <i>row</i> 和 <i>column</i> 开始显示 <i>value</i> 。	† [PRGM] I/O <b>6:Output</b> ( 16-20
<b>Param</b>	设置参数图象方式。	† [MODE] <b>Par</b> 1-11
<b>Pause</b>	暂停程序的执行，直到按 [ENTER]。	† [PRGM] CTL <b>8:Pause</b> 16-13
<b>Pause</b> [ <i>value</i> ]	暂停程序的执行并显示 <i>value</i> ，直到按 [ENTER]。	† [PRGM] CTL <b>8:Pause</b> 16-13
<b>Plot#</b> ( <i>type,Xlistname,Ylistname,mark</i> )	为 <i>Xlistname</i> 和 <i>Ylistname</i> 定义 <i>type</i> 为 <b>Scatter</b> 或 <b>xyLine</b> 的统计图 <b>Plot#</b> ( <b>1</b> 、 <b>2</b> 或 <b>3</b> )，图的标记为 <i>mark</i> 。	† [2nd] [STAT PLOT] PLOTS <b>1:Plot1</b> ( <b>2:Plot2</b> ( <b>3:Plot3</b> ( 12-34
<b>Plot#</b> ( <i>type,Xlistname,freqlist</i> )	为频数是 <i>freqlist</i> 的 <i>Xlistname</i> 定义 <i>type</i> 为 <b>Histogram</b> 或 <b>Boxplot</b> 的 <b>Plot#</b> ( <b>1</b> 、 <b>2</b> 或 <b>3</b> )。	† [2nd] [STAT PLOT] PLOTS <b>1:Plot1</b> ( <b>2:Plot2</b> ( <b>3:Plot3</b> ( 12-34
<b>Plot#</b> ( <i>type,Xlistname,freqlist,mark</i> )	为频数是 <i>freqlist</i> 的 <i>Xlistname</i> 定义 <i>type</i> 为 <b>ModBoxplot</b> 的 <b>Plot#</b> ( <b>1</b> 、 <b>2</b> 或 <b>3</b> )，图的标记为 <i>mark</i> 。	† [2nd] [STAT PLOT] PLOTS <b>1:Plot1</b> ( <b>2:Plot2</b> ( <b>3:Plot3</b> ( 12-34
<b>Plot#</b> ( <i>type,datalistname,data axis,mark</i> )	使用 <i>mark</i> 在 <i>data axis</i> 上，为 <i>datalistname</i> 定义 <i>type</i> <b>NormProbPlot</b> 的 <b>Plot#</b> ( <b>1</b> 、 <b>2</b> 或 <b>3</b> ) <i>data axis</i> 可以是 <b>X</b> 或 <b>Y</b> 。	† [2nd] [STAT PLOT] PLOTS <b>1:Plot1</b> ( <b>2:Plot2</b> ( <b>3:Plot3</b> ( 12-34
<b>PlotsOff</b> [1,2,3]	取消选择所有的统计图，或者取消选择所指定的一个或多个统计图 ( <b>1</b> 、 <b>2</b> 或 <b>3</b> )。	[2nd] [STAT PLOT] STAT PLOTS <b>4:PlotsOff</b> 12-35
<b>PlotsOn</b> [1,2,3]	选择所有的统计图，或者选择所指定的一个或多个统计图 ( <b>1</b> 、 <b>2</b> 或 <b>3</b> )。	[2nd] [STAT PLOT] STAT PLOTS <b>5:PlotsOn</b> 12-35

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项
<b>Pmt_Bgn</b>	指定期初年金，即在每个支付期开始时支付。	[APPS] 1:Finance CALC F:Pmt_Bgn 14-14
<b>Pmt_End</b>	指定普通年金，即在每个支付期结束时支付。	[APPS] 1:Finance CALC E:Pmt_End 14-14
<b>poissoncdf(<math>\mu, x</math>)</b>	指定离散泊松分布的平均值 $\mu$ 计算其在 $x$ 处的累积概率。	[2nd] [DISTR] DISTR C:poissoncdf( 13-34
<b>poissonpdf(<math>\mu, x</math>)</b>	指定离散泊松分布的平均值 $\mu$ 计算其在 $x$ 处的概率。	[2nd] [DISTR] DISTR B:poissonpdf( 13-33
<b>Polar</b>	设置极坐标图象方式。	† [MODE] Pol 5-3
<i>complex value</i> ▶ <b>Polar</b>	在极坐标格式下显示 <i>complex value</i> 。	[MATH] CPX 7:▶Polar 2-19
<b>PolarGC</b>	设置极坐标图象坐标格式。	† [2nd] [FORMAT] PolarGC 3-13
<b>prgmname</b>	执行程序 <i>name</i> 。	† [PRGM] CTRL D:prgm 16-16
$\Sigma$ Prn( <i>pmt1</i> , <i>pmt2</i> [, <i>roundvalue</i> ])	计算分期付款计划表在 <i>pmt1</i> 和 <i>pmt2</i> 之间的本金总额，结果舍入到 <i>roundvalue</i> 。	[APPS] 1:Finance CALC 0:ΣPrn( 14-10
<b>prod(list[,start,end])</b>	返回 <i>start</i> 和 <i>end</i> 间 <i>list</i> 元素的积。	[2nd] [LIST] MATH 6:prod( 11-18
<b>Prompt variableA</b> [, <i>variableB</i> ,..., <i>variable n</i> ]	提示 <i>variableA</i> 的值，接着是 <i>variableB</i> 的值，等等。	† [PRGM] I/O 2:Prompt 16-19

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项
<b>1-PropZInt</b> ( $x, n$ [, <i>confidence level</i> ])	计算单比例 $z$ 的置信区间。	† [STAT] TESTS <b>A:1-PropZInt</b> ( 13-20)
<b>2-PropZInt</b> ( $x1, n1, x2, n2$ [, <i>confidence level</i> ])	计算双比例 $z$ 的置信区间。	† [STAT] TESTS <b>B:2-PropZInt</b> ( 13-21)
<b>1-PropZTest</b> ( $p0, x, n$ [, <i>alternative, drawflag</i> ])	计算单比例 $z$ 检验。 <i>alternative</i> =-1 是 <, <i>alternative</i> =0 是 ≠, <i>alternative</i> =1 是 >。 <i>drawflag</i> =1 绘制结果的图形, <i>drawflag</i> =0 计算结果。	† [STAT] TESTS <b>5:1-PropZTest</b> ( 13-14)
<b>2-PropZTest</b> ( $x1, n1, x2, n2$ [, <i>alternative, drawflag</i> ])	计算双比例 $z$ 检验。 <i>alternative</i> =-1 是 <, <i>alternative</i> =0 是 ≠, <i>alternative</i> =1 是 >。 <i>drawflag</i> =1 绘制结果图形, <i>drawflag</i> =0 计算结果。	† [STAT] TESTS <b>6:2-PropZTest</b> ( 13-15)
<b>Pt-Change</b> ( $x, y$ )	反转点 ( $x, y$ )。	[2nd] [DRAW] POINTS <b>3:Pt-Change</b> ( 8-15)
<b>Pt-Off</b> ( $x, y$ [, <i>mark</i> ])	使用 <i>mark</i> 消除点 ( $x, y$ )。	[2nd] [DRAW] POINTS <b>2:Pt-Off</b> ( 8-15)
<b>Pt-On</b> ( $x, y$ [, <i>mark</i> ])	使用 <i>mark</i> 绘出点 ( $x, y$ )。	[2nd] [DRAW] POINTS <b>1:Pt-On</b> ( 8-14)
<b>PwrReg</b> [ $Xlistname,$ $Ylistname, freqlist,$ $regequ$ ]	用幂回归模型拟合频数为 <i>freqlist</i> 的数据 <i>Xlistname</i> 和 <i>Ylistname</i> , 并将回归方程保 存到 <i>regequ</i> 。	[STAT] CALC <b>A:PwrReg</b> 12-27

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>Pxl-Change</b> ( <i>row,column</i> )	反转 ( <i>row,column</i> ) 处的象素; $0 \leq row \leq 62$ 与 $0 \leq column \leq 94$ 。	[2nd] [DRAW] POINTS <b>6:Pxl-Change</b> (	8-16
<b>Pxl-Off</b> ( <i>row,column</i> )	清除 ( <i>row,column</i> ) 处的象素; $0 \leq row \leq 62$ 与 $0 \leq column \leq 94$ 。	[2nd] [DRAW] POINTS <b>5:Pxl-Off</b> (	8-16
<b>Pxl-On</b> ( <i>row,column</i> )	绘出 ( <i>row,column</i> ) 处的象素; $0 \leq row \leq 62$ 与 $0 \leq column \leq 94$ 。	[2nd] [DRAW] POINTS <b>4:Pxl-On</b> (	8-16
<b>pxl-Test</b> ( <i>row,column</i> )	如果象素 ( <i>row, column</i> ) 是开 启的, 则返回 1; 如果是关闭 的, 则返回 0; $0 \leq row \leq 62$ 与 $0 \leq column \leq 94$ 。	[2nd] [DRAW] POINTS <b>7:pxl-Test</b> (	8-16
<b>P►Rx</b> ( <i>r,θ</i> )	对给出的极坐标 <i>r</i> 和 $\theta$ 或者极 坐标数组, 返回 <b>X</b> 。	[2nd] [ANGLE] ANGLE <b>7:P►Rx</b> (	2-24
<b>P►Ry</b> ( <i>r,θ</i> )	对给出的极坐标 <i>r</i> 和 $\theta$ 或者极 坐标数组, 返回 <b>Y</b> 。	[2nd] [ANGLE] ANGLE <b>8:P►Ry</b> (	2-24
<b>QuadReg</b> [ <i>Xlistname</i> , <i>Ylistname</i> , <i>freqlist</i> , <i>regequ</i> ]	用二次回归模型拟合频数为 <i>freqlist</i> 的数据 <i>Xlistname</i> 和 <i>Ylistname</i> , 并将回归方程保 存到 <i>regequ</i> 。	[STAT] CALC <b>5:QuadReg</b>	12-25
<b>QuartReg</b> [ <i>Xlistname</i> , <i>Ylistname</i> , <i>freqlist</i> , <i>regequ</i> ]	用四次回归模型拟合频数为 <i>freqlist</i> 的数据 <i>Xlistname</i> 和 <i>Ylistname</i> , 并将回归方程保 存到 <i>regequ</i> 。	[STAT] CALC <b>7:QuartReg</b>	12-26
<b>Radian</b>	设置角的弧度方式。	† [MODE] <b>Radian</b>	1-11
<b>rand</b> ( <i>numtrials</i> )	对指定的试验数 <i>numtrials</i> 返 回一个 0 和 1 之间的随机数。	[MATH] PRB <b>1:rand</b>	2-20
<b>randBin</b> ( <i>numtrials,prob</i> [, <i>numsimulations</i> ])	从指定的二项分布中生成并显 示一个随机实数。	[MATH] PRB <b>7:randBin</b> (	2-22

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>randInt</b> ( <i>lower,upper</i> [, <i>numtrials</i> ])	对指定的试验数 <i>numtrials</i> 生成并显示一个在指定 <i>lower</i> 和 <i>upper</i> 边界内的随机整数。	$\boxed{\text{MATH}}$ PRB <b>5:randInt</b> (	2-22
<b>randM</b> ( <i>rows,columns</i> )	返回一个 <i>rows</i> (1-99) × <i>columns</i> (1-99) 的随机矩阵。	$\boxed{2\text{nd}}$ [MATRX] MATH <b>6:randM</b> (	10-13
<b>randNorm</b> ( $\mu,\sigma$ , <i>numtrials</i> )	对指定的试验数 <i>numtrials</i> 从 $\mu$ 和 $\sigma$ 所指定的正态分布中生成并显示一个随机实数。	$\boxed{\text{MATH}}$ PRB <b>6:randNorm</b> (	2-22
<b>re<sup><math>\theta</math>i</sup></b>	设置极坐标的复数方式 ( <b>re<sup><math>\theta</math>i</sup></b> )。	† $\boxed{\text{MODE}}$ <b>re<sup><math>\theta</math>i</sup></b>	1-12
<b>Real</b>	设置仅当输入复数时才显示复数结果的方式。	† $\boxed{\text{MODE}}$ <b>Real</b>	1-12
<b>real</b> ( <i>value</i> )	返回一个复数或复数数组的实部。	$\boxed{\text{MATH}}$ CPX <b>2:real</b> (	2-18
<b>RecallGDB</b> <i>n</i>	恢复所有保存在图象数据库变量 <b>GDB</b> <i>n</i> 中的设置。	$\boxed{2\text{nd}}$ [DRAW] STO <b>4:RecallGDB</b>	8-20
<b>RecallPic</b> <i>n</i>	显示图象并增加保存在 <b>Pic</b> <i>n</i> 中的图。	$\boxed{2\text{nd}}$ [DRAW] STO <b>2:RecallPic</b>	8-18
<i>complex value</i> $\blacktriangleright$ <b>Rect</b>	在直角坐标系格式下显示 <i>complex value</i> 或数组。	$\boxed{\text{MATH}}$ CPX <b>6<math>\blacktriangleright</math>Rect</b>	2-19
<b>RectGC</b>	设置直角坐标系格式。	† $\boxed{2\text{nd}}$ [FORMAT] <b>RectGC</b>	3-13
<b>ref</b> ( <i>matrix</i> )	返回 <i>matrix</i> 的行梯形矩阵。	$\boxed{2\text{nd}}$ [MATRX] MATH <b>A:ref</b> (	10-15



函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>:Repeat</b> <i>condition</i> <i>:commands</i> <b>:End</b> <i>:commands</i>	如果 <i>condition</i> 为假, 则一直执行 <i>commands</i> 。	† [PRGM] CTL <b>6:Repeat</b>	16-12
<b>Return</b>	返回到调用程序。	† [PRGM] CTL <b>E:Return</b>	16-16
<b>round</b> ( <i>value</i> [, <i>#decimals</i> ])	返回舍入到 <i>#decimals</i> ( $\leq 9$ ) 位的数、表达式、数组或矩阵。	[MATH] NUM <b>2:round(</b>	2-13
<b>*row</b> ( <i>value</i> , <i>matrix</i> , <i>row</i> )	返回一个矩阵, 矩阵 <i>matrix</i> 的行 <i>row</i> 乘以 <i>value</i> 后仍保存在 <i>row</i> 中。	[2nd] [MATRIX] MATH <b>E:*row(</b>	10-16
<b>row+</b> ( <i>matrix</i> , <i>rowA</i> , <i>rowB</i> )	返回一个矩阵, 矩阵 <i>matrix</i> 的 <i>rowA</i> 和 <i>rowB</i> 相加后结果仍保存在 <i>rowB</i> 中。	[2nd] [MATRIX] MATH <b>D:row+(</b>	10-16
<b>*row+</b> ( <i>value</i> , <i>matrix</i> , <i>rowA</i> , <i>rowB</i> )	返回一个矩阵, 该矩阵是由矩阵 <i>matrix</i> 的 <i>rowA</i> 乘上 <i>value</i> , 再将其和 <i>rowB</i> 相加并把结果保存到 <i>rowB</i> 而生成的。	[2nd] [MATRIX] MATH <b>F:*row+(</b>	10-16
<b>rowSwap</b> ( <i>matrix</i> , <i>rowA</i> , <i>rowB</i> )	返回一个矩阵, 该矩阵是由矩阵 <i>matrix</i> 的 <i>rowA</i> 与 <i>rowB</i> 交换位置后生成的。	[2nd] [MATRIX] MATH <b>C:rowSwap(</b>	10-16
<b>rref</b> ( <i>matrix</i> )	返回一个 <i>matrix</i> 的简化行梯形矩阵。	[2nd] [MATRIX] MATH <b>B:rref(</b>	10-15
<b>R►Pr</b> ( <i>x</i> , <i>y</i> )	根据所给的直角坐标 <i>x</i> 和 <i>y</i> 或直角坐标的数组, 返回 <b>R</b> 。	[2nd] [ANGLE] ANGLE <b>5:R►Pr(</b>	2-24
<b>R►Pθ</b> ( <i>x</i> , <i>y</i> )	根据所给的直角坐标 <i>x</i> 和 <i>y</i> 或直角坐标的数组, 返回 <b>θ</b> 。	[2nd] [ANGLE] ANGLE <b>6:R►Pθ(</b>	2-24

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>2-SampFTest</b> [ <i>listname1</i> , <i>listname2</i> , <i>freqlist1</i> , <i>freqlist2</i> , <i>alternative</i> , <i>drawflag</i> ] (数据数组输入)	执行一个双样本 F 检验。 <i>alternative</i> =-1 是 <, <i>alternative</i> =0 是 ≠, <i>alternative</i> =1 是 >。 <i>drawflag</i> =1 绘制结果图形, <i>drawflag</i> =0 计算结果。	† <u>[STAT]</u> TESTS <b>D:2-SampFTest</b>	13-23
<b>2-SampFTest</b> <i>Sx1</i> , <i>n1</i> , <i>Sx2</i> , <i>n2</i> [, <i>alternative</i> , <i>drawflag</i> ] (概率统计输入)	执行一个双样本 F 检验。 <i>alternative</i> =-1 是 <, <i>alternative</i> =0 是 ≠, <i>alternative</i> =1 是 >。 <i>drawflag</i> =1 绘制结果图形, <i>drawflag</i> =0 计算结果。	† <u>[STAT]</u> TESTS <b>D:2-SampFTest</b>	13-23
<b>2-SampTInt</b> [ <i>listname1</i> , <i>listname2</i> , <i>freqlist1</i> , <i>freqlist2</i> , <i>confidence level</i> , <i>pooled</i> ] (数据数组输入)	计算一个双样本 <i>t</i> 的置信区间。 <i>pooled</i> =1 合并方差; <i>pooled</i> =0 不合并方差。	† <u>[STAT]</u> TESTS <b>0:2-SampTInt</b>	13-19
<b>2-SampTInt</b> $\bar{x}1$ , <i>Sx1</i> , <i>n1</i> , $\bar{x}2$ , <i>Sx2</i> , <i>n2</i> [, <i>confidence level</i> , <i>pooled</i> ] (概率统计输入)	计算一个双样本 <i>t</i> 的置信区间。 <i>pooled</i> =1 合并方差; <i>pooled</i> =0 不合并方差。	† <u>[STAT]</u> TESTS <b>0:2-SampTInt</b>	13-19
<b>2-SampTTest</b> [ <i>listname1</i> , <i>listname2</i> , <i>freqlist1</i> , <i>freqlist2</i> , <i>alternative</i> , <i>pooled</i> , <i>drawflag</i> ] (数据数组输入)	计算一个双样本 <i>t</i> 的检验。 <i>alternative</i> =-1 是 <, <i>alternative</i> =0 是 ≠, <i>alternative</i> =1 是 >。 <i>pooled</i> =1 合并方差; <i>pooled</i> =0 不合并方差。 <i>drawflag</i> =1 绘制结果图形; <i>drawflag</i> =0 计算结果。	† <u>[STAT]</u> TESTS <b>4:2-SampTTest</b>	13-13

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>2-SampTTest</b> $\bar{x}1, Sx1, n1,$ $\bar{x}2, Sx2, n2, [alternative,$ $pooled, drawflag]$ (概率统计输入)	计算一个双样本 $t$ 的检验。 $alternative=-1$ 是 $<$ , $alternative=0$ 是 $\neq$ , $alternative=1$ 是 $>$ 。 $pooled=1$ 合并方差; $pooled=0$ 不合并方 差。 $drawflag=1$ 绘制结果图 形; $drawflag=0$ 计算结果。	† [STAT] TESTS <b>4:2-SampTTest</b>	13-13
<b>2-SampZInt</b> ( $\sigma_1, \sigma_2,$ $[listname1, listname2,$ $freqlist1, freqlist2,$ $confidence level]$ ) (数据数组输入)	计算一个双样本 $z$ 的置信区间。	† [STAT] TESTS <b>9:2-SampZInt(</b>	13-18
<b>2-SampZInt</b> ( $\sigma_1, \sigma_2,$ $\bar{x}1, n1, \bar{x}2, n2$ $[confidence level]$ ) (概率统计输入)	计算一个双样本 $z$ 的置信区间。	† [STAT] TESTS <b>9:2-SampZInt(</b>	13-18
<b>2-SampZTest</b> ( $\sigma_1, \sigma_2,$ $[listname1, listname2,$ $freqlist1, freqlist2,$ $alternative, drawflag]$ ) (数据数组输入)	计算一个双样本 $z$ 的检验。 $alternative=-1$ 是 $<$ , $alternative=0$ 是 $\neq$ , $alternative=1$ 是 $>$ 。 $drawflag=1$ 绘制结果图形; $drawflag=0$ 计算结果。	† [STAT] TESTS <b>3:2-SampZTest(</b>	13-12
<b>2-SampZTest</b> ( $\sigma_1, \sigma_2,$ $\bar{x}1, n1, \bar{x}2, n2$ $[alternative, drawflag]$ ) (概率统计输入)	计算一个双样本 $z$ 的检验。 $alternative=-1$ 是 $<$ , $alternative=0$ 是 $\neq$ , $alternative=1$ 是 $>$ 。 $drawflag=1$ 绘制结果图形; $drawflag=0$ 计算结果。	† [STAT] TESTS <b>3:2-SampZTest(</b>	13-12
<b>Sci</b>	设置科学记数法显示方式。	† [MODE] <b>Sci</b>	1-10
<b>Select</b> ( $Xlistname,$ $Ylistname$ )	从散射图或 $xy$ 直线图 (只对 二者) 选择一个或多个特殊的 数据点, 然后将选定的数据点 保存到两个新数组 $Xlistname$ 和 $Ylistname$ 中。	[2nd] [LIST] OPS <b>8:Select(</b>	11-12

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>Send(variable)</b>	将 <i>variable</i> 中的内容发送到 CBL 或者 CBR 系统。	† [PRGM] I/O <b>B:Send(</b>	16-22
<b>seq(expression,variable, begin,end[,increment])</b>	返回数组, 数组元素为 <i>variable</i> 从 <i>begin</i> 到 <i>end</i> 并以 <i>increment</i> 增加所计算出的 <i>expression</i> 值。	[2nd] [LIST] OPS <b>5:seq(</b>	11-12
<b>Seq</b>	设置序列图象方式。	† [MODE] <b>Seq</b>	1-11
<b>Sequential</b>	设置顺序绘制函数图象方式。	† [MODE] <b>Sequential</b>	1-12
<b>SetUpEditor</b>	从统计数组编辑器中删除所有的数组名, 然后将 <b>L1</b> 到 <b>L6</b> 的数组名恢复到列 <b>1</b> 到 <b>6</b> 中。	[STAT] EDIT <b>5:SetUpEditor</b>	12-21
<b>SetUpEditor listname1 [listname2,..., listname20]</b>	从统计数组编辑器中删除所有的数组名, 然后设置统计数组编辑器从 <b>1</b> 列开始按指定的顺序显示一个或多个 <i>listnames</i> 。	[STAT] EDIT <b>5:SetUpEditor</b>	12-21
<b>Shade(lowerfunc, upperfunc[,Xleft,Xright, pattern,patres])</b>	在当前图象上对 <b>X</b> 绘制 <i>lowerfunc</i> 和 <i>upperfunc</i> , 并使用 <i>pattern</i> 和 <i>patres</i> 来遮蔽由 <i>lowerfunc</i> 、 <i>upperfunc</i> 、 <i>Xleft</i> 和 <i>Xright</i> 确定的区域。	[2nd] [DRAW] DRAW <b>7:Shade(</b>	8-10
<b>Shade<math>\chi^2</math>(lowerbound, upperbound,df)</b>	绘制由自由度 <i>df</i> 所指定的 $\chi^2$ 分布的密度函数, 并遮蔽 <i>lowerbound</i> 和 <i>upperbound</i> 间的区域。	[2nd] [DISTR] DRAW <b>3:Shade<math>\chi^2</math>(</b>	13-36

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>ShadeF</b> ( <i>lowerbound</i> , <i>upperbound</i> , <i>numerator df</i> , <i>denominator df</i> )	绘制由 <i>numerator df</i> 和 <i>denominator df</i> 所指定的 F 分布的密度函数，并遮蔽 <i>lowerbound</i> 和 <i>upperbound</i> 间的区域。	$\boxed{2nd}$ [DISTR] DRAW <b>4:ShadeF</b> (	13-37
<b>ShadeNorm</b> ( <i>lowerbound</i> , <i>upperbound</i> [, $\mu$ , $\sigma$ ])	绘制由 $\mu$ 和 $\sigma$ 所指定的正态密度函数，并遮蔽 <i>lowerbound</i> 和 <i>upperbound</i> 间的区域。	$\boxed{2nd}$ [DISTR] DRAW <b>1:ShadeNorm</b> (	13-35
<b>Shade_t</b> ( <i>lowerbound</i> , <i>upperbound</i> , <i>df</i> )	绘制由自由度 <i>df</i> 所指定的 t 分布的密度函数，并遮蔽 <i>lowerbound</i> 和 <i>upperbound</i> 间的区域。	$\boxed{2nd}$ [DISTR] DRAW <b>2:Shade_t</b> (	13-36
<b>Simul</b>	设置同时绘制函数图象方式。	† $\boxed{MODE}$ <b>Simul</b>	1-12
<b>sin</b> ( <i>value</i> )	返回实数、表达式或数组的正弦。	$\boxed{SIN}$	2-3
<b>sin<sup>-1</sup></b> ( <i>value</i> )	返回实数、表达式或数组的反正弦。	$\boxed{2nd}$ [SIN <sup>-1</sup> ]	2-3
<b>sinh</b> ( <i>value</i> )	返回实数、表达式或数组的双曲正弦。	$\boxed{2nd}$ [CATALOG] <b>sinh</b> (	15-10
<b>sinh<sup>-1</sup></b> ( <i>value</i> )	返回实数、表达式或数组的反双曲正弦。	$\boxed{2nd}$ [CATALOG] <b>sinh<sup>-1</sup></b> (	15-10

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>SinReg</b> [ <i>iterations</i> , <i>Xlistname</i> , <i>Ylistname</i> , <i>period</i> , <i>regequ</i> ]	迭代 <i>iterations</i> 次, 利用估计值 <i>period</i> 并用正弦回归模型拟合 <i>Xlistname</i> 和 <i>Ylistname</i> , 将回归方程保存到 <i>regequ</i> 。	[STAT] CALC <b>C:SinReg</b>	12-27
<b>solve</b> ( <i>expression</i> , <i>variable</i> , <i>guess</i> , <i>{lower,upper}</i> )	根据最初的估计值 <i>guess</i> 并在 <i>lower</i> 和 <i>upper</i> 范围内对 <i>variable</i> 求解 <i>expression</i> 。	† [MATH] MATH <b>0:solve(</b>	2-12
<b>SortA</b> ( <i>listname</i> )	按升序排序 <i>listname</i> 的元素。	[2nd] [LIST] OPS <b>1:SortA(</b>	11-10 12-20
<b>SortA</b> ( <i>keylistname</i> , <i>dependlist1</i> [, <i>dependlist2</i> , ..., <i>dependlist n</i> ])	按升序对 <i>keylistname</i> 的元素进行排序, 然后根据 <i>keylistname</i> 数组对每个 <i>dependlist</i> 进行排序。	[2nd] [LIST] OPS <b>1:SortA(</b>	11-10 12-20
<b>SortD</b> ( <i>listname</i> )	按降序排序 <i>listname</i> 元素。	[2nd] [LIST] OPS <b>2:SortD(</b>	11-10 12-20
<b>SortD</b> ( <i>keylistname</i> , <i>dependlist1</i> [, <i>dependlist2</i> , ..., <i>dependlist n</i> ])	按降序对 <i>keylistname</i> 的元素进行排序, 然后根据 <i>keylistname</i> 数组对每个 <i>dependlist</i> 进行排序。	[2nd] [LIST] OPS <b>2:SortD(</b>	11-10 12-20
<b>stdDev</b> ( <i>list</i> [, <i>freqlist</i> ])	返回频数为 <i>freqlist</i> 的 <i>list</i> 中元素的标准差。	[2nd] [LIST] MATH <b>7:stdDev(</b>	11-18
<b>Stop</b>	程序执行结束后返回到主屏幕。	† [PRGM] CTL <b>F:Stop</b>	16-16
Store: <i>value</i> → <i>variable</i>	将 <i>value</i> 保存在 <i>variable</i> 中。	[STO]	1-14
<b>StoreGDB</b> <i>n</i>	将当前图象保存到数据库 <b>GDB<sub>n</sub></b> 中。	[2nd] [DRAW] STO <b>3:StoreGDB</b>	8-19

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>StorePic</b> $n$	将当前图形保存到 <b>Pic</b> $n$ 中。	$\boxed{2nd}$ [DRAW] STO <b>1:StorePic</b>	8-17
<b>StringEqu</b> ( $string, Y= var$ )	将 $string$ 转换成方程并把它保存到 $Y= var$ 。	$\boxed{2nd}$ [CATALOG] <b>StringEqu</b> (	15-8
<b>sub</b> ( $string, begin, length$ )	返回字符串 $string$ 从 $begin$ 开始, 长度为 $length$ 的子串。	$\boxed{2nd}$ [CATALOG] <b>sub</b> (	15-9
<b>sum</b> ( $list[, start, end]$ )	返回 $list$ 中从 $start$ 到 $end$ 元素的和。	$\boxed{2nd}$ [LIST] MATH <b>5:sum</b> (	11-18
<b>tan</b> ( $value$ )	返回实数、表达式或数组的正切。	$\boxed{TAN}$	2-3
<b>tan<sup>-1</sup></b> ( $value$ )	返回实数、表达式或数组的反正切。	$\boxed{2nd}$ [TAN <sup>-1</sup> ]	2-3
<b>Tangent</b> ( $expression, value$ )	绘出与 $expression$ 在 $X=value$ 处相切的直线。	$\boxed{2nd}$ [DRAW] DRAW <b>5:Tangent</b> (	8-8
<b>tanh</b> ( $value$ )	返回实数、表达式或数组的双曲正切。	$\boxed{2nd}$ [CATALOG] <b>tanh</b> (	15-10
<b>tanh<sup>-1</sup></b> ( $value$ )	返回实数、表达式或数组的反双曲正切。	$\boxed{2nd}$ [CATALOG] <b>tanh<sup>-1</sup></b> (	15-10
<b>tcdf</b> ( $lowerbound, upperbound, df$ )	对所指定的自由度 $df$ 计算在 $lowerbound$ 和 $upperbound$ 之间的 $t$ 分布概率。	$\boxed{2nd}$ [DISTR] DISTR <b>5:tcdf</b> (	13-31
<b>Text</b> ( $row, column, text1, text2, \dots, text n$ )	在图象上从 ( $row, column$ ) 象素开始写 $text$ , 其中 $0 \leq row \leq 57$ 及 $0 \leq column \leq 94$ 。	$\boxed{2nd}$ [DRAW] DRAW <b>0:Text</b> (	8-12
<b>Then</b> 请参阅 <b>If:Then</b>			

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>Time</b>	设置对时间的序列图象。	† [2nd] [FORMAT] <b>Time</b>	6-8
<b>TInterval</b> [ <i>listname</i> , <i>frequist,confidence level</i> ] (数据数组输入)	计算 $t$ 的置信区间。	† [STAT] TESTS <b>8:TInterval</b>	13-17
<b>TInterval</b> $\bar{x},Sx,n$ [, <i>confidence level</i> ] (概率统计输入)	计算 $t$ 的置信区间。	† [STAT] TESTS <b>8:TInterval</b>	13-17
<b>tpdf</b> ( $x,df$ )	利用指定的自由度 $df$ 计算 $t$ 分布在指定的 $x$ 值处的概率密度函数 (pdf)。	[2nd] [DISTR] DISTR <b>4:tpdf(</b>	13-30
<b>Trace</b>	显示图象并进入 TRACE 方式。	[TRACE]	3-18
<b>T-Test</b> $\mu_0$ [ <i>listname</i> , <i>frequist,alternative</i> , <i>drawflag</i> ] (数据数组输入)	利用频数 <i>frequist</i> , 进行 $t$ 检验。 <i>alternative=-1</i> 是 <; <i>alternative=0</i> 是 ≠; <i>alternative=1</i> 是 >。 <i>drawflag=1</i> 绘制结果图形; <i>drawflag=0</i> 计算结果。	† [STAT] TESTS <b>2:T-Test</b>	13-11
<b>T-Test</b> $\mu_0, \bar{x},Sx,n$ [, <i>alternative,drawflag</i> ] (概率统计输入)	利用频数 <i>frequist</i> , 进行 $t$ 检验。 <i>alternative=-1</i> 是 <; <i>alternative=0</i> 是 ≠; <i>alternative=1</i> 是 >。 <i>drawflag=1</i> 绘制结果图形; <i>drawflag=0</i> 计算结果。	† [STAT] TESTS <b>2:T-Test</b>	13-11



函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项
<b>tvm_FV</b> [( <i>N</i> , <i>I</i> %, <i>PV</i> , <i>PMT</i> , <i>P/Y</i> , <i>C/Y</i> )]	计算未来值。	[APPS] 1:Finance CALC 6:tvm_FV 14-8
<b>tvm_I%</b> [( <i>N</i> , <i>PV</i> , <i>PMT</i> , <i>FV</i> , <i>P/Y</i> , <i>C/Y</i> )]	计算年利率。	[APPS] 1:Finance CALC 3:tvm_I% 14-8
<b>tvm_N</b> [( <i>I</i> %, <i>PV</i> , <i>PMT</i> , <i>FV</i> , <i>P/Y</i> , <i>C/Y</i> )]	计算支付期数。	[APPS] 1:Finance CALC 5:tvm_N 14-8
<b>tvm_Pmt</b> [( <i>N</i> , <i>I</i> %, <i>PV</i> , <i>FV</i> , <i>P/Y</i> , <i>C/Y</i> )]	计算每次支付的金额。	[APPS] 1:Finance CALC 2:tvm_Pmt 14-7
<b>tvm_PV</b> [( <i>N</i> , <i>I</i> %, <i>PMT</i> , <i>FV</i> , <i>P/Y</i> , <i>C/Y</i> )]	计算现值。	[APPS] 1:Finance CALC 4:tvm_PV 14-8
<b>UnArchive</b>	将特定变量从用户数据归档区移至 RAM。 要归档变量, 请使用 <b>Archive</b> 。	[2nd] [MEM] 6:UnArchive 18-10
<b>uvAxes</b>	设置序列图象, 将 <b>u(n)</b> 绘制在 x 轴上、 <b>v(n)</b> 绘制在 y 轴上。	† [2nd] [FORMAT] uv 6-8
<b>uwAxes</b>	设置序列图象, 将 <b>u(n)</b> 绘制在 x 轴上、 <b>w(n)</b> 绘制在 y 轴上。	† [2nd] [FORMAT] uw 6-8
<b>1-Var Stats</b> [ <i>Xlistname</i> , <i>freqlist</i> ]	对 <i>Xlistname</i> 中频数为 <i>freqlist</i> 的数据进行一元分析。	[STAT] CALC 1:1-Var Stats 12-25
<b>2-Var Stats</b> [ <i>Xlistname</i> , <i>Ylistname</i> , <i>freqlist</i> ]	对 <i>Xlistname</i> 和 <i>Ylistname</i> 中频数为 <i>freqlist</i> 的数据进行二元分析。	[STAT] CALC 2:2-Var Stats 12-25
<b>variance</b> ( <i>list</i> [ <i>freqlist</i> ])	返回 <i>list</i> 中频数为 <i>freqlist</i> 的元素的方差。	[2nd] [LIST] MATH 8:variance( 11-18
<b>Vertical</b> <i>x</i>	在 <i>x</i> 处画一条垂直线。	[2nd] [DRAW] DRAW 4:Vertical 8-6
<b>vwAxes</b>	设置序列图象, 将 <b>v(n)</b> 绘制在 x 轴上、 <b>w(n)</b> 绘制在 y 轴上。	† [2nd] [FORMAT] vw 6-8
<b>Web</b>	设置序列图象, 如蛛网类型跟踪。	† [2nd] [FORMAT] Web 6-8

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>:While</b> <i>condition</i> <b>:commands</b> <b>:End</b> <b>:command</b>	当 <i>condition</i> 为真时, 执行 <i>commands</i> 。	† [PRGM] CTL 5:While	16-12
<b>valueA xor valueB</b>	只有 <i>valueA</i> 或 <i>valueB</i> = 0 才返回 1。 <i>valueA</i> 和 <i>valueB</i> 可以为实数、表达式或数组。	[2nd] [TEST] LOGIC 3:xor	2-26
<b>ZBox</b>	显示一个图象, 让您画一个方框定义新观察窗口, 然后更新窗口内容。	† [ZOOM] ZOOM 1:ZBox	3-20
<b>ZDecimal</b>	调整观察窗口使 $\Delta X=0.1$ 及 $\Delta Y=0.1$ , 然后以屏幕中心作为其原点显示图象屏幕。	† [ZOOM] ZOOM 4:ZDecimal	3-21
<b>ZInteger</b>	使用下列尺寸重新定义观察窗口: $\Delta X=1$ $Xscl=10$ $\Delta Y=1$ $Yscl=10$	† [ZOOM] ZOOM 8:ZInteger	3-22
<b>ZInterval</b> $\sigma, listname,$ <i>freqlist, confidence level</i> (数据数组输入)	计算 $z$ 置信区间。	† [STAT] TESTS 7:ZInterval	13-16
<b>ZInterval</b> $\sigma, \bar{x}, n$ <i>[, confidence level]</i> (概率统计输入)	计算 $z$ 置信区间。	† [STAT] TESTS 7:ZInterval	13-16
<b>Zoom In</b>	放大光标周围部分的图象。	† [ZOOM] ZOOM 2:Zoom In	3-21
<b>Zoom Out</b>	以光标位置为中心显示图象的较大部分。	† [ZOOM] ZOOM 3:Zoom Out	3-21

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>ZoomFit</b>	重新计算 <b>Ymin</b> 和 <b>Ymax</b> 使之包含所选函数在 <b>Xmin</b> 和 <b>Xmax</b> 之间的 <b>Y</b> 极小值和极大值, 然后重新绘制函数图象。	† <b>ZOOM</b> <b>ZOOM</b> <b>0:ZoomFit</b>	3-22
<b>ZoomRcl</b>	在自定义的观察窗口中绘制选择函数的图象。	† <b>ZOOM</b> <b>MEMORY</b> <b>3:ZoomRcl</b>	3-23
<b>ZoomStat</b>	重新定义观察窗口以便显示所有的统计数据点。	† <b>ZOOM</b> <b>ZOOM</b> <b>9:ZoomStat</b>	3-22
<b>ZoomSto</b>	立即保存当前的观察窗口。	† <b>ZOOM</b> <b>MEMORY</b> <b>2:ZoomSto</b>	3-23
<b>ZPrevious</b>	使用在执行上一条 <b>ZOOM</b> 指令以前所显示图象的窗口变量来重新绘制图象。	† <b>ZOOM</b> <b>MEMORY</b> <b>1:ZPrevious</b>	3-23
<b>ZSquare</b>	调整 <b>X</b> 或 <b>Y</b> 窗口设置使坐标系中的每个象素都有相同的宽度和高度, 然后更新观察窗口。	† <b>ZOOM</b> <b>ZOOM</b> <b>5:ZSquare</b>	3-21
<b>ZStandard</b>	立即重新绘制函数图象, 将窗口变量更改为默认值。	† <b>ZOOM</b> <b>ZOOM</b> <b>6:ZStandard</b>	3-22

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
<b>Z-Test</b> ( $\mu_0, \sigma$ , <i>listname</i> , <i>freqlist</i> , <i>alternative</i> , <i>drawflag</i> ) (数据数组输入)	利用频数 <i>freqlist</i> , 进行 $z$ 检验。 <i>alternative</i> =-1 是 <; <i>alternative</i> =0 是 $\neq$ ; <i>alternative</i> =1 是 >。 <i>drawflag</i> =1 绘制结果图形; <i>drawflag</i> =0 计算结果。	† [STAT] TESTS 1:Z-Test(	13-10
<b>Z-Test</b> ( $\mu_0, \sigma, \bar{x}$ , <i>n</i> [, <i>alternative</i> , <i>drawflag</i> ]) (概率统计输入)	进行 $z$ 检验。 <i>alternative</i> =-1 是 <; <i>alternative</i> =0 是 $\neq$ ; <i>alternative</i> =1 是 >。 <i>drawflag</i> =1 绘制结果图形; <i>drawflag</i> =0 计算结果。	† [STAT] TESTS 1:Z-Test(	13-10
<b>ZTrig</b>	立即重新绘制函数图象, 将窗口 变量更新为绘制三角函数的预 设置值。	† [ZOOM] ZOOM 7:ZTrig	3-22
阶乘: <i>value</i> !	返回 <i>value</i> 的阶乘。	[MATH] PRB 4:!	2-21
阶乘: <i>list</i> !	返回 <i>list</i> 元素的阶乘。	[MATH] PRB 4:!	2-21
度记数法: <i>value</i> <sup>°</sup>	解释 <i>value</i> 为度; 以 DMS 格式 表示度。	[2nd] [ANGLE] ANGLE 1:°	2-23
弧度: <i>angle</i> <sup>r</sup>	解释 <i>angle</i> 为弧度。	[2nd] [ANGLE] ANGLE 3:r	2-24
转置: <i>matrix</i> <sup>T</sup>	返回将 <i>matrix</i> 的每一元素 (列, 行) 转换到相应 (行, 列) 的位 置所形成的矩阵。	[2nd] [MATRX] MATH 2:T	10-12

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
$x^{\text{th}}\text{root}x\sqrt{\text{value}}$	返回值 $\text{value}$ 的 $x^{\text{th}}$ 方根。	$\overline{\text{MATH}}$ MATH 5: $x\sqrt{\phantom{x}}$	2-6
$x^{\text{th}}\text{root}x\sqrt{\text{list}}$	返回数组 $\text{list}$ 元素的 $x^{\text{th}}$ 方根。	$\overline{\text{MATH}}$ MATH 5: $x\sqrt{\phantom{x}}$	2-6
$\text{list}x\sqrt{\text{value}}$	返回一个数组，元素为对值 $\text{value}$ 开数组 $\text{list}$ 中每个元素方根。	$\overline{\text{MATH}}$ MATH 5: $x\sqrt{\phantom{x}}$	2-6
$\text{list}A^x\sqrt{\text{list}B}$	返回一个数组，元素为数组 $\text{list}B$ 中每个元素开对应数组 $\text{list}A$ 中元素方根。	$\overline{\text{MATH}}$ MATH 5: $x\sqrt{\phantom{x}}$	2-6
立方: $\text{value}^3$	返回实数或复数、表达式、数组或方阵的立方。	$\overline{\text{MATH}}$ MATH 3: $^3$	2-6 10-10
立方根: $\sqrt[3]{\text{value}}$	返回实数或复数、表达式或数组的立方根。	$\overline{\text{MATH}}$ MATH 4: $^3\sqrt{\phantom{x}}$	2-6
等于: $\text{value}A=\text{value}B$	如果 $\text{value}A = \text{value}B$ ，则返回 1；如果 $\text{value}A \neq \text{value}B$ ，则返回 0。 $\text{value}A$ 和 $\text{value}B$ 可以是实数或复数、表达式、数组或矩阵。	$\overline{2\text{nd}}$ [TEST] TEST 1:=	2-25 10-11
不等于: $\text{value}A\neq\text{value}B$	如果 $\text{value}A \neq \text{value}B$ ，则返回 1；如果 $\text{value}A = \text{value}B$ ，则返回 0。 $\text{value}A$ 和 $\text{value}B$ 可以是实数或复数、表达式、数组或矩阵。	$\overline{2\text{nd}}$ [TEST] TEST 2: $\neq$	2-25 10-11
小于: $\text{value}A<\text{value}B$	如果 $\text{value}A < \text{value}B$ ，则返回 1；如果 $\text{value}A \geq \text{value}B$ ，则返回 0。 $\text{value}A$ 和 $\text{value}B$ 可以是实数或复数、表达式或数组。	$\overline{2\text{nd}}$ [TEST] TEST 5:<	2-25

## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
大于: $valueA > valueB$	如果 $valueA > valueB$ , 则返回 1; 如果 $valueA \leq valueB$ , 则返回 0。 $valueA$ 和 $valueB$ 可以是实数或复数、表达式或数组。	$\boxed{2nd}$ [TEST] TEST 3:>	2-25
小于等于: $valueA \leq valueB$	如果 $valueA \leq valueB$ , 则返回 1; 如果 $valueA > valueB$ , 则返回 0。 $valueA$ 和 $valueB$ 可以是实数或复数、表达式或数组。	$\boxed{2nd}$ [TEST] TEST 6:≤	2-25
大于等于: $valueA \geq valueB$	如果 $valueA \geq valueB$ , 则返回 1; 如果 $valueA < valueB$ , 则返回 0。 $valueA$ 和 $valueB$ 可以是实数或复数、表达式或数组。	$\boxed{2nd}$ [TEST] TEST 4:≥	2-25
取倒数: $value^{-1}$	返回实数、复数或表达式的倒数。	$\boxed{x^{-1}}$	2-3
取倒数: $list^{-1}$	返回 $list$ 中元素的倒数。	$\boxed{x^{-1}}$	2-3
逆: $matrix^{-1}$	返回矩阵 $matrix$ 的逆矩阵。	$\boxed{x^{-1}}$	10-10
平方: $value^2$	返回值 $value$ 的平方。 $value$ 可以是实数、复数或表达式。	$\boxed{x^2}$	2-3
平方: $list^2$	返回数组, 其元素是 $list$ 元素的平方。	$\boxed{x^2}$	2-3
平方: $matrix^2$	返回矩阵 $matrix$ 的平方。	$\boxed{x^2}$	10-10
幂: $value^{power}$	返回 $value$ 的 $power$ 次幂。 $value$ 可以是实数、复数或表达式。	$\boxed{\wedge}$	2-3
幂: $list^{power}$	返回数组, 其元素是对应 $list$ 元素的 $power$ 次幂。	$\boxed{\wedge}$	2-3
幂: $value^{list}$	返回 $value$ 的 $list$ 数组中元素次幂。	$\boxed{\wedge}$	2-3

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
幂: $matrix^{power}$	返回 $matrix$ 的 $power$ 次幂。	$\boxed{\wedge}$	10-10
取负: $-value$	返回对实数或复数、表达式、数组或矩阵的取负。	$\boxed{-}$	2-4 10-9
10 的次幂: $10^{(value)}$	返回 10 的 $value$ 次幂。 $value$ 可以是实数、复数或表达式。	$\boxed{2nd} [10^x]$	2-4
10 的次幂: $10^{(list)}$	返回数组，元素是以数组 $list$ 中每个元素为指数对 10 的幂。	$\boxed{2nd} [10^x]$	2-4
平方根: $\sqrt{(value)}$	返回实数或复数、表达式或数组的平方根。	$\boxed{2nd} [\sqrt{ }]$	2-3
乘: $valueA * valueB$	返回 $valueA$ 乘以 $valueB$ 的结果。	$\boxed{\times}$	2-3
乘: $value * list$	返回 $value$ 和每个 $list$ 元素相乘后的数组。	$\boxed{\times}$	2-3
乘: $list * value$	返回每个 $list$ 元素和 $value$ 相乘后的数组。	$\boxed{\times}$	2-3
乘: $listA * listB$	返回数组 $listA$ 的元素乘以数组 $listB$ 的元素后的数组。	$\boxed{\times}$	2-3
乘: $value * matrix$	返回 $value$ 和矩阵 $matrix$ 的元素相乘后的矩阵。	$\boxed{\times}$	10-9
乘: $matrixA * matrixB$	返回矩阵 $matrixA$ 乘以矩阵 $matrixB$ 后的矩阵。	$\boxed{\times}$	10-9
除: $valueA / valueB$	返回 $valueA$ 除以 $valueB$ 的值。	$\boxed{\div}$	2-3
除: $list / value$	返回数组 $list$ 的元素除以 $value$ 后的数组。	$\boxed{\div}$	2-3
除: $value / list$	返回 $value$ 除以数组 $list$ 元素后的数组。	$\boxed{\div}$	2-3
除: $listA / listB$	返回数组 $listA$ 的元素除以数组 $listB$ 的元素后的数组。	$\boxed{\div}$	2-3

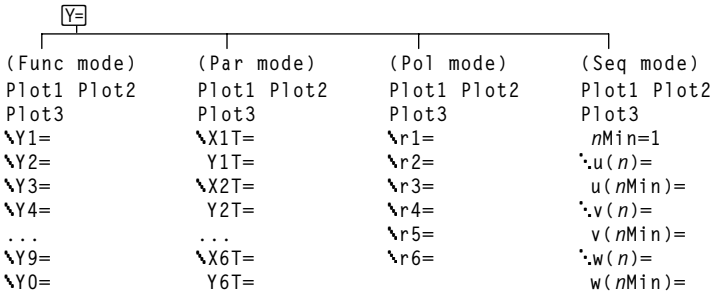
## 函数和指令表 (续)

函数或指令/ 参数	结果	单键或多键/ 菜单或屏幕/项	
加: $valueA+valueB$	返回 $valueA$ 加 $valueB$ 的值。	$\oplus$	2-3
加: $list+value$	返回 $list$ 中每个元素加上 $value$ 后的数组。	$\oplus$	2-3
加: $listA+listB$	返回数组 $listA$ 元素加上数组 $listB$ 元素后的数组。	$\oplus$	2-3
加: $matrixA+matrixB$	返回矩阵 $matrixA$ 元素加上矩阵 $matrixB$ 元素后的矩阵。	$\oplus$	10-9
合并: $string1+string2$	合并两个或多个字符串。	$\oplus$	15-6
减: $valueA-valueB$	$valueA$ 减 $valueB$ 。	$\ominus$	2-3
减: $value-list$	返回用 $value$ 减去数组 $list$ 元素后的数组。	$\ominus$	2-3
减: $list-value$	返回数组 $list$ 的元素减去 $value$ 后的数组。	$\ominus$	2-3
减: $listA-listB$	返回数组 $listA$ 元素减去数组 $listB$ 元素后的数组。	$\ominus$	2-3
减: $matrixA-matrixB$	返回矩阵 $matrixA$ 元素减去矩阵 $matrixB$ 元素后的矩阵。	$\ominus$	10-9
分记号: $degrees^{\circ}minutes'$ $seconds''$	用分来表示角度 $minutes$ 。	$\boxed{2nd}$ [ANGLE] ANGLE $2:'$	2-23
秒记号: $degrees^{\circ}minutes'$ $seconds''$	用秒来表示角度 $seconds$ 。	$\boxed{ALPHA}$ ["]	2-23



## TI-83 Plus 菜单图

TI-83 Plus 菜单图从键盘的左上角开始，并且按照键盘布局从左到右的顺序说明菜单。其默认值及设置如下所示。



### [2nd] [STAT PLOT]

STAT PLOTS  
 1:Plot1...Off  
     ↵ L1 L2 □  
 2:Plot2...Off  
     ↵ L1 L2 □  
 3:Plot3...Off  
     ↵ L1 L2 □  
 4:PlotsOff  
 5:PlotsOn

### [2nd] [STAT PLOT]

(PRGM editor)	(PRGM editor)	(PRGM editor)
PLOTS	TYPE	MARK
1:Plot1(	1:Scatter	1:□
2:Plot2(	2:xyLine	2:+
3:Plot3(	3:Histogram	3:•
4:PlotsOff	4:ModBoxplot	
5:PlotsOn	5:Boxplot	
	6:NormProbPlot	

### [WINDOW]

(Func mode)	(Par mode)	(Pol mode)	(Seq mode)
WINDOW	WINDOW	WINDOW	WINDOW
Xmin=-10	Tmin=0	θmin=0	nMin=1
Xmax=10	Tmax=π*2	θmax=π*2	nMax=10
Xscl=1	Tstep=π/24	θstep=π/24	PlotStart=1
Ymin=-10	Xmin=-10	Xmin=-10	PlotStep=1
Ymax=10	Xmax=10	Xmax=10	Xmin=-10
Yscl=1	Xscl=1	Xscl=1	Xmax=10
Xres=1	Ymin=-10	Ymin=-10	Xscl=1
	Ymax=10	Ymax=10	Ymin=-10
	Yscl=1	Yscl=1	Ymax=10
			Yscl=1

### [2nd] [TBLSET]

TABLE SETUP  
 TblStart=0  
 ΔTbl=1  
 Indpnt:Auto Ask  
 Depend:Auto Ask

### [2nd] [TBLSET]

(PRGM editor)  
 TABLE SETUP  
 Indpnt:Auto Ask  
 Depend:Auto Ask

## TI-83 Plus 菜单图 (续)

### ZOOM

ZOOM	MEMORY	MEMORY
1:ZBox	1:ZPrevious	(Set Factors...)
2:Zoom In	2:ZoomSto	ZOOM FACTORS
3:Zoom Out	3:ZoomRcl	XFact=4
4:ZDecimal	4:SetFactors...	YFact=4
5:ZSquare		
6:ZStandard		
7:ZTrig		
8:ZInteger		
9:ZoomStat		
0:ZoomFit		

### [2nd] [FORMAT]

(Func/Par/Pol modes)	(Seq mode)
RectGC PolarGC	Time Web uv vw uw
CoordOn CoordOff	RectGC PolarGC
GridOff GridOn	CoordOn CoordOff
AxesOn AxesOff	GridOff GridOn
LabelOff LabelOn	AxesOn AxesOff
ExprOn ExprOff	LabelOff LabelOn
	ExprOn ExprOff

### [2nd] [CALC]

(Func mode)	(Par mode)	(Pol mode)	(Seq mode)
CALCULATE	CALCULATE	CALCULATE	CALCULATE
1:value	1:value	1:value	1:value
2:zero	2:dy/dx	2:dy/dx	
3:minimum	3:dy/dt	3:dy/dθ	
4:maximum	4:dx/dt		
5:intersect			
6:dy/dx			
7:jf(x)dx			

### MODE

Normal Sci Eng  
 Float 0123456789  
 Radian Degree  
 Func Par Pol Seq  
 Connected Dot  
 Sequential Simul  
 Real a+bt re^θt  
 Full Horiz G-T

---

**[2nd] [LINK]****SEND**

1: All+...  
2: All-...  
3: Prgm...  
4: List...  
5: Lists to TI82...  
6: GDB...  
7: Pic...  
8: Matrix...  
9: Real...  
0: Complex...  
A: Y-Vars...  
B: String...  
C: Apps...  
D: AppVars...  
E: Group...  
F: SendId  
G: SendSW  
H: Back Up...

**RECEIVE**

1: Receive

**[STAT]****EDIT**

1: Edit...  
2: SortA(  
3: SortD(  
4: ClrList  
5: SetUpEditor

**CALC**

1: 1-Var Stats  
2: 2-Var Stats  
3: Med-Med  
4: LinReg(ax+b)  
5: QuadReg  
6: CubicReg  
7: QuartReg  
8: LinReg(a+bx)  
9: LnReg  
0: ExpReg  
A: PwrReg  
B: Logistic  
C: SinReg

**TESTS**

1: Z-Test...  
2: T-Test...  
3: 2-SampZTest...  
4: 2-SampTTest...  
5: 1-PropZTest...  
6: 2-PropZTest...  
7: ZInterval...  
8: TInterval...  
9: 2-SampZInt...  
0: 2-SampTInt...  
A: 1-PropZInt...  
B: 2-PropZInt...  
C:  $\chi^2$ -Test...  
D: 2-SampFTest...  
E: LinRegTTest...  
F: ANOVA(

## TI-83 Plus 菜单图 (续)

---

### [2nd] [LIST]

NAMES	OPS	MATH
1:listname	1:SortA(	1:min(
2:listname	2:SortD(	2:max(
3:listname	3:dim(	3:mean(
...	4:Fill(	4:median(
	5:seq(	5:sum(
	6:cumSum(	6:prod(
	7:ΔList(	7:stdDev(
	8:Select(	8:variance(
	9:augment(	
	0:List▶matr(	
	A:Matr▶list(	
	B:⌂	

### [MATH]

MATH	NUM	CPX	PRB
1:▶Frac	1:abs(	1:conj(	1:rand
2:▶Dec	2:round(	2:real(	2:nPr
3: <sup>3</sup>	3:iPart(	3:imag(	3:nCr
4: <sup>3</sup> √(	4:fPart(	4:angle(	4:!
5:X√	5:int(	5:abs(	5:randInt(
6:fMin(	6:min(	6:▶Rect	6:randNorm(
7:fMax(	7:max(	7:▶Polar	7:randBin(
8:nDeriv(	8:lcm(		
9:fnInt(	9:gcd(		
0:Solver...			

### [2nd] [TEST]

TEST	LOGIC
1:=	1:and
2:≠	2:or
3:>	3:xor
4:≥	4:not(
5:<	
6:≤	

[2nd] [MATRX]

NAMES	MATH	EDIT
1:[A]	1:det(	1:[A]
2:[B]	2:T	2:[B]
3:[C]	3:dim(	3:[C]
4:[D]	4:Fill(	4:[D]
5:[E]	5:identity(	5:[E]
6:[F]	6:randM(	6:[F]
7:[G]	7:augment(	7:[G]
8:[H]	8:Matr▶list(	8:[H]
9:[I]	9>List▶matr(	9:[I]
0:[J]	0:cumSum(	0:[J]
	A:ref(	
	B:rref(	
	C:rowSwap(	
	D:row+(	
	E:*row(	
	F:*row+(	

[2nd] [ANGLE]

ANGLE
1:°
2:'
3:ʹ
4:▶DMS
5:R▶Pr(
6:R▶Pθ(
7:R▶Rx(
8:R▶Ry(

[PRGM]

EXEC	EDIT	NEW
1:name	1:name	1:Create New
2:name	2:name	
...	...	

[PRGM]

(PRGM editor)	(PRGM editor)	(PRGM editor)
CTL	I/O	EXEC
1:If	1:Input	1:name
2:Then	2:Prompt	2:name
3:Else	3:Disp	...
4:For(	4:DispGraph	
5:While	5:DispTable	
6:Repeat	6:Output(	
7:End	7:getKey	
8:Pause	8:ClrHome	
9:Lb1	9:ClrTable	
0:Goto	0:GetCalc(	
A:IS>(	A:Get(	
B:DS<(	B:Send(	
C:Menu(		
D:prgm		
E:Return		
F:Stop		
G:DelVar		
H:GraphStyle(		

## TI-83 Plus 菜单图 (续)

---

### 2nd [DRAW]

DRAW	POINTS	STO
1:ClrDraw	1:Pt-On(	1:StorePic
2:Line(	2:Pt-Off(	2:RecallPic
3:Horizontal	3:Pt-Change(	3:StoreGDB
4:Vertical	4:Pxl-On(	4:RecallGDB
5:Tangent(	5:Pxl-Off(	
6:DrawF	6:Pxl-Change(	
7:Shade(	7:pxl-Test(	
8:DrawInv		
9:Circle(		
0:Text(		
A:Pen		

### VARΣ

VARΣ	Y-VARS
1:Window...	1:Function...
2:Zoom...	2:Parametric...
3:GDB...	3:Polar...
4:Picture...	4:On/Off...
5:Statistics...	
6:Table...	
7:String...	

### VARΣ

(Window...)	(Window...)	(Window...)
X/Y	T/θ	U/V/W
1:Xmin	1:Tmin	1:u(nMin)
2:Xmax	2:Tmax	2:v(nMin)
3:Xscl	3:Tstep	3:w(nMin)
4:Ymin	4:θmin	4:nMin
5:Ymax	5:θmax	5:nMax
6:Yscl	6:θstep	6:PlotStart
7:Xres		7:PlotStep
8:ΔX		
9:ΔY		
0:XFact		
A:YFact		

---

VARs

(Zoom...)	(Zoom...)	(Zoom...)
ZX/ZY	ZT/Z $\theta$	ZU
1:ZXmin	1:ZTmin	1:Zu( <i>n</i> Min)
2:ZXmax	2:ZTmax	2:Zv( <i>n</i> Min)
3:ZXsc1	3:ZTstep	3:Zw( <i>n</i> Min)
4:ZYmin	4:Z $\theta$ min	4:Z <i>n</i> Min
5:ZYmax	5:Z $\theta$ max	5:Z <i>n</i> Max
6:ZYsc1	6:Z $\theta$ step	6:ZPlotStart
7:ZXres		7:ZPlotStep

VARs

(GDB...)	(Picture...)
GRAPH DATABASE	PICTURE
1:GDB1	1:Pic1
2:GDB2	2:Pic2
...	...
9:GDB9	9:Pic9
0:GDB0	0:Pic0

VARs

(Statistics...)	(Statistics...)	(Statistics...)	(Statistics...)	(Statistics...)
XY	$\Sigma$	EQ	TEST	PTS
1:n	1: $\Sigma x$	1:RegEQ	1:p	1:x1
2: $\bar{x}$	2: $\Sigma x^2$	2:a	2:z	2:y1
3:Sx	3: $\Sigma y$	3:b	3:t	3:x2
4: $\sigma x$	4: $\Sigma y^2$	4:c	4: $\chi^2$	4:y2
5: $\bar{y}$	5: $\Sigma xy$	5:d	5:F	5:x3
6:Sy		6:e	6:df	6:y3
7: $\sigma y$		7:r	7: $\hat{p}$	7:Q1
8:minX		8:r <sup>2</sup>	8: $\hat{p}1$	8:Med
9:maxX		9:R <sup>2</sup>	9: $\hat{p}2$	9:Q3
0:minY			0:s	
A:maxY			A: $\bar{x}1$	
			B: $\bar{x}2$	
			C:Sx1	
			D:Sx2	
			E:Sxp	
			F:n1	
			G:n2	
			H:lower	
			I:upper	

## TI-83 Plus 菜单图 (续)

---

### VARs

(Table...)	(String...)
TABLE	STRING
1:TblStart	1:Str1
2:ΔTbl	2:Str2
3:TblInput	3:Str3
	4:Str4
	...
	9:Str9
	0:Str0

### Y-VARS

(Function...)	(Parametric...)	(Polar...)	(On/Off...)
FUNCTION	PARAMETRIC	POLAR	ON/OFF
1:Y1	1:X1T	1:r1	1:FnoN
2:Y2	2:Y1T	2:r2	2:Fnoff
3:Y3	3:X2T	3:r3	
4:Y4	4:Y2T	4:r4	
...	...	5:r5	
9:Y9	A:X6T	6:r6	
0:Y0	B:Y6T		



---

[2nd] [DISTR]

DISTR

- 1:normalpdf(
- 2:normalcdf(
- 3:invNorm(
- 4:tpdf(
- 5:tcdf(
- 6: $\chi^2$ pdf(
- 7: $\chi^2$ cdf(
- 8:Fpdf(
- 9:Fcdf(
- 0:binompdf(
- A:binomcdf(
- B:poissonpdf(
- C:poissoncdf(
- D:geometpdf(
- E:geometcdf(

DRAW

- 1:ShadeNorm(
- 2:Shade\_t(
- 3:Shade $\chi^2$ (
- 4:ShadeF(

[APPS]

1:Finance

2:CBL/CBR

Finance

- CALC
- 1:TVM
- Solver...
- 2:tvm\_Pmt
- 3:tvm\_1%
- 4:tvm\_PV
- 5:tvm\_N
- 6:tvm\_FV
- 7:npv(
- 8:irr(
- 9:bal(
- 0: $\Sigma$ Prn(
- A: $\Sigma$ Int(
- B:Nom(
- C:Eff(
- D:dbd(
- E:Pmt\_End
- F:Pmt\_Bgn

VAR S

- 1:N
- 2:1%
- 3:PV
- 4:PMT
- 5:FV
- 6:P/Y
- 7:C/Y

CBL/CBR

- 1:GAUGE
- 2:DATA LOGGER
- 3:CBR
- 4:QUIT

[2nd] [MEM]

MEMORY  
 1:About  
 2:Mem Mgmt/Del...  
 3:Clear Entries  
 4:ClrAllLists  
 5:Archive  
 6:UnArchive  
 7:Reset...  
 8:Group

MEMORY  
 (Mem Mgmt/Del...)  
 RAM FREE 25631  
 ARC FREE 131069  
 1:All...  
 2:Real...  
 3:Complex...  
 4:List...  
 5:Matrix...  
 6:Y- Vars...  
 7:Prgm...  
 8:Pic...  
 9:GDB...  
 0:String...  
 A:Apps...  
 B:AppVars...  
 C:Group...

MEMORY (Reset...)

RAM  
 1:All RAM...  
 2:Defaults...

Resetting RAM erases  
 all data and programs  
 from RAM.

ARCHIVE  
 1:Vars...  
 2:Apps...  
 B:Both...

Resetting Both erases  
 all data, programs and  
 Apps from Archive.

ALL  
 1:All Memory...

Resetting ALL erases  
 all data, programs and  
 Apps from RAM and  
 Archive.

RAM

RESET RAM  
 1:No  
 2:Reset

Resetting RAM erases  
 all data and programs  
 from RAM.

RESET DEFAULTS  
 1:No  
 2:Reset

ARCHIVE

RESET ARC VARS  
 1:No  
 2:Reset

Resetting Vars erases  
 all data and programs  
 from Archive.

RESET ARC APPS  
 1:No  
 2:Reset

Resetting Apps erases  
 all Apps from Archive.

RESET ARC BOTH  
 1:No  
 2:Reset

Resetting Both erases  
 all data, programs and  
 Apps from Archive.

---

ALL  
└─┬─  
RESET MEMORY  
1:No  
2:Reset  
Resetting ALL will  
delete all data,  
programs & Apps from  
RAM & Archive.

#### MEMORY (GROUP...)

└─┬─  
GROUP UNGROUP  
1:Create New

#### MEMORY (UNGROUP...)

└─┬─  
1:*name*  
2:*name*  
...

#### [2nd] [CATALOG]

└─┬─  
CATALOG  
cosh(  
cosh<sup>-1</sup>(  
...  
Equ►String(  
expr(  
...  
inString(  
...  
length(  
...  
sinh(  
sinh<sup>-1</sup>(  
...  
String►Equ(  
sub(  
...  
tanh(  
tanh<sup>-1</sup>(

## 用户变量

TI-83 Plus 以不同方式使用下面所列变量。一些变量限于特定的数据类型。

变量 **A-Z** 和  **$\theta$**  定义为实数或复数，可以对它们进行存储。TI-83 Plus 在绘图期间会更新 **X**、**Y**、**R**、 **$\theta$**  和 **T**，因此要避免使用这些变量来保存非图象数据。

变量（数组名）**L1** 到 **L6** 限制用于数组，不能将其他类型的数据保存到它们中。

变量（矩阵名）**[A]** 到 **[J]** 限制用于矩阵，不能将其他类型的数据保存到它们中。

变量 **Pic1** 到 **Pic9** 以及 **Pic0** 限制用于图形，不能将其他类型的数据保存到它们中。

变量 **GDB1** 到 **GDB9** 以及 **GDB0** 限制用于图象数据库，不能将其他类型的数据保存到它们中。

变量 **Str1** 到 **Str9** 以及 **Str0** 限制用于字符串，不能将其他类型的数据保存到它们中。

除了系统变量之外，您可以直接或通过 **Y=** 编辑器将一些字符串、函数串、指令串或变量串保存到函数 **Y<sub>n</sub>**（**1** 到 **9** 以及 **0**）、**X<sub>nT</sub>/Y<sub>nT</sub>**（**1** 到 **6**）、**r<sub>n</sub>**（**1** 到 **6**）、**u(n)**、**v(n)** 和 **w(n)** 中。当函数计算时，决定串的有效性。

## 归档变量

可以将数据、程序或者任何变量从 **RAM** 存储到用户数据归档区内存，在这里，它们不会被用户在无意中进行编辑或删除。归档也使您可以释放 **RAM**，以便将其用于需要附加内存的变量。归档变量的名称前有一个“\*”号，它表示这些变量位于用户数据归档区。

## 系统变量

下面的变量必须是实数。可以对它们进行存储。由于 TI-83 Plus 会更改这些变量中的一些，例如，在执行 **ZOOM** 后。因此要避免使用这些变量来保存非图象数据。

- **Xmin**、**Xmax**、**Xscl**、 **$\Delta X$** 、**XFact**、**Tstep**、**PlotStart**、**nMin** 和其他的窗口变量。
- **ZXmin**、**ZXmax**、**ZXscl**、**ZTstep**、**ZPlotStart**、**Zu(nMin)** 和其他的 **ZOOM** 变量。

下面一些变量为 TI-83 Plus 所专用。不能对它们存储。

**n**、 **$\bar{x}$** 、**Sx**、 **$\sigma x$** 、**minX**、**maxX**、 **$\Sigma y$** 、 **$\Sigma y^2$** 、 **$\Sigma xy$** 、**a**、**b**、**c**、**RegEQ**、**x1**、**x2**、**y1**、**z**、**t**、**F**、 **$\chi^2$** 、 **$\hat{p}$** 、 **$\bar{x}1$** 、**Sx1**、**n1**、**lower**、**upper**、**r<sup>2</sup>**、**R<sup>2</sup>** 和其他统计变量。

本节包括 **Logistic** 和 **SinReg** 回归、**ANOVA**、**2-SampFTest** 和 **2-SampTTest** 的统计公式。

### Logistic

逻辑回归算法应用非线性递归最小二乘方法优化下面损失函数：

$$J = \sum_{i=1}^N \left( \frac{c}{1 + ae^{-bx_i}} - y_i \right)^2$$

这是离差的平方和，

式中：  
 $x$  = 自变量数组  
 $y$  = 因变量数组  
 $N$  = 数组维数

该方法试图递归估计常数  $a$ 、 $b$  和  $c$  使  $J$  尽可能地小。

### SinReg

正弦回归算法应用非线性递归最小二乘方法优化下面损失函数：

$$J = \sum_{i=1}^N [a \sin(bx_i + c) + d - y_i]^2$$

这是离差的平方和，

式中：  
 $x$  = 自变量数组  
 $y$  = 因变量数组  
 $N$  = 数组维数

该方法试图递归估计常数  $a$ 、 $b$ 、 $c$  和  $d$  使  $J$  尽可能地小。

### ANOVA(

**ANOVA F** 统计值是:

$$F = \frac{\text{Factor } MS}{\text{Error } MS}$$

---

组成 **F** 的均方 (*MS*) 是:

$$\text{Factor } MS = \frac{\text{Factor } SS}{\text{Factor } df}$$

$$\text{Error } MS = \frac{\text{Error } SS}{\text{Error } df}$$

---

组成均方的平方 (*SS*) 总和是:

$$\text{Factor } SS = \sum_{i=1}^I n_i(\bar{x}_i - \bar{x})^2$$

$$\text{Error } SS = \sum_{i=1}^I (n_i - 1)Sx_i^2$$

---

组成均方的自由度 *df* 是:

$$\text{Factor } df = I - 1 = \text{numerator } df \text{ for } F$$

$$\text{Error } df = \sum_{i=1}^I (n_i - 1) = \text{denominator } df \text{ for } F$$

---

式中:

- $I$  = 总体数
- $\bar{x}_i$  = 每个数组的平均数
- $Sx_i$  = 每个数组的标准差
- $n_i$  = 每个数组的长度
- $\bar{x}$  = 所有数组的平均数

---

## 2-SampFTest

以下是 **2-SampFTest** 的定义。

$Sx1, Sx2$  = 样本标准差分别有  $n_1-1$  和  $n_2-1$  个自由度  $df$ 。

$$F = \text{F-统计} = \left( \frac{Sx1}{Sx2} \right)^2$$

$df(x, n_1-1, n_2-1) = \text{Fpdf}()$  有自由度  $df$ 、 $n_1-1$  和  $n_2-1$

$p$  = 报告的  $p$  值

---

**2-SampFTest** 对于择一假设  $\sigma_1 > \sigma_2$ 。

$$p = \int_{F}^{\infty} f(x, n_1-1, n_2-1) dx$$

---

**2-SampFTest** 对于择一假设  $\sigma_1 < \sigma_2$ 。

$$p = \int_0^F f(x, n_1-1, n_2-1) dx$$

---

**2-SampFTest** 对于择一假设  $\sigma_1 \neq \sigma_2$ 。界限必须满足下式：

$$\frac{p}{2} = \int_0^{Lbnd} f(x, n_1-1, n_2-1) dx = \int_{Ubnd}^{\infty} f(x, n_1-1, n_2-1) dx$$

式中：

$[Lbnd, Ubnd]$  = 下限和上限

F 统计用作产生最小积分的边界。另一个边界被用作实现与前一个积分的相等关系。

### 2-SampTTest

以下是 **2-SampTTest** 的定义。自由度为  $df$  的双样本  $t$  统计是:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{S}$$

$S$  和  $df$  的计算依赖于方差是否被合并。如果方差没被合并:

$$S = \sqrt{\frac{Sx_1^2}{n_1} + \frac{Sx_2^2}{n_2}}$$

$$df = \frac{\left(\frac{Sx_1^2}{n_1} + \frac{Sx_2^2}{n_2}\right)^2}{\frac{1}{n_1 - 1} \left(\frac{Sx_1^2}{n_1}\right)^2 + \frac{1}{n_2 - 1} \left(\frac{Sx_2^2}{n_2}\right)^2}$$

---

否则:

$$Sx_p = \frac{(n_1 - 1)Sx_1^2 + (n_2 - 1)Sx_2^2}{df}$$

$$S = \sqrt{\frac{1}{n_1} + \frac{1}{n_2}} Sx_p$$

$$df = n_1 + n_2 - 2$$

$Sx_p$  是合并方差。



本节包括计算货币的时间价值、分期付款、现金流转、利率换算和日期间天数的金融公式。

## 货币的时间价值

$$i = [e^{(y \times \ln(x+1))}] - 1$$

式中:  $PMT \neq 0$

$$y = C/Y \div P/Y$$

$$x = (.01 \times I\%) \div C/Y$$

$C/Y$  = 每年的复合周期

$P/Y$  = 每年的支付期

$I\%$  = 每年的利率

$$i = (-FV \div PV)^{(1+N)} - 1$$

式中:  $PMT = 0$

计算  $i$  的迭代公式:

$$0 = PV + PMT \times G_i \left[ \frac{1 - (1+i)^{-N}}{i} \right] + FV \times (1+i)^{-N}$$

---

$$I\% = 100 \times C/Y \times [e^{(y \times \ln(x+1))} - 1]$$

式中:  $x = i$

$$y = P/Y \div C/Y$$

---

$$G_i = 1 + i \times k$$

式中:  $k = 0$  期末支付

$k = 1$  期初支付

---

$$N = \frac{\ln\left(\frac{PMT \times G_i - FV \times i}{PMT \times G_i + PV \times i}\right)}{\ln(1+i)}$$

式中:  $i \neq 0$

$$N = -(PV + FV) \div PMT$$

式中:  $i = 0$

---

货币的时间价值 (续)

$$PMT = \frac{-i}{G_i} \times \left[ PV + \frac{PV + FV}{(1+i)^N - 1} \right]$$

式中:  $i \neq 0$

$$PMT = -(PV + FV) \div N$$

式中:  $i = 0$

---

$$PV = \left[ \frac{PMT \times G_i}{i} - FV \right] \times \frac{1}{(1+i)^N} - \frac{PMT \times G_i}{i}$$

式中:  $i \neq 0$

$$PV = -(FV + PMT \times N)$$

式中:  $i = 0$

---

$$FV = \frac{PMT \times G_i}{i} - (1+i)^N \times \left( PV + \frac{PMT \times G_i}{i} \right)$$

式中:  $i \neq 0$

$$FV = -(PV + PMT \times N)$$

式中:  $i = 0$

---

---

## 分期付款

如果计算  $bal()$ ,  $pmt2 = npmt$

令  $bal(0) = RND(PV)$

从  $m = 1$  迭代到  $pmt2$

$$\begin{cases} I_m = RND[RND12(-i \times bal(m-1))] \\ bal(m) = bal(m-1) - I_m + RND(PMT) \end{cases}$$

则:

$$bal() = bal(pmt2)$$

$$\Sigma Prn() = bal(pmt2) - bal(pmt1)$$

$$\Sigma Int() = (pmt2 - pmt1 + 1) \times RND(PMT) - \Sigma Prn()$$

式中:

$RND$  = 舍入显示到所选择的小数位数

$RND12$  = 舍入到 12 位小数

余额、本金和利息依赖于 **PMT**、**PV**、**1%**、 $pmt1$  和  $pmt2$  的值。

### 现金流转

$$npv() = CF_0 + \sum_{j=1}^N CF_j(1+i)^{-S_j-1} \frac{(1-(1+i)^{-n_j})}{i}$$

$$\text{式中: } S_j = \begin{cases} \sum_{i=1}^j n_i & j \geq 1 \\ 0 & j = 0 \end{cases}$$

净现值依赖于初始的现金流转 ( $CF_0$ )、后来的现金流转 ( $CF_j$ )、每个现金流转的频数 ( $n_j$ ) 和指定利率 ( $i$ ) 的值。

---

$$irr() = 100 \times i, \text{ 式中 } i \text{ 满足 } npv() = 0$$

内部回报率取决于初始的现金流转 ( $CF_0$ ) 和后来的现金流转 ( $CF_j$ ) 的值。

---

$$i = I\% \div 100$$

---

### 利率换算

$$\blacktriangleright \text{Eff}() = 100 \times (e^{CP \times \ln(x+1)} - 1)$$

$$\text{式中: } x = .01 \times NOM \div CP$$

$$\blacktriangleright \text{Nom}() = 100 \times CP \times [e^{1 + CP \times \ln(x+1)} - 1]$$

$$\text{式中: } x = .01 \times EFF$$

$EFF$  = 实际利率

$CP$  = 复合周期

$NOM$  = 额定利率

---

---

## 日期间的天数

用 **dbd**( 函数, 可以输入或计算从 1950 年 1 月 1 日到 2049 年 12 月 31 日期间的日期。

---

**实际/实际天数算法** (假设每月的实际天数和每年的实际天数) :

$dbd$ ( (日期期间的天数) = 天数 II - 天数 I

$$\begin{aligned} \text{天数 I} &= (Y1-YB) \times 365 \\ &+ (MB \text{ 到 } M1 \text{ 的天数}) \\ &+ DT1 \\ &+ \frac{(Y1-YB)}{4} \end{aligned}$$

$$\begin{aligned} \text{天数 II} &= (Y2-YB) \times 365 \\ &+ (MB \text{ 到 } M2 \text{ 的天数}) \\ &+ DT2 \\ &+ \frac{(Y2-YB)}{4} \end{aligned}$$

式中:

$M1$  = 第一个日期的月  
 $DT1$  = 第一个日期的日  
 $Y1$  = 第一个日期的年  
 $M2$  = 第二个日期的月  
 $DT2$  = 第二个日期的日  
 $Y2$  = 第二个日期的年  
 $MB$  = 基月 ( 1 月 )  
 $DB$  = 基日 ( 1 )  
 $YB$  = 基年 ( 闰年后第一年 )

---

# B 通用信息

---

## 目录

电池信息 .....	2
疑难解答 .....	4
出错情况 .....	6
精度信息 .....	11
TI 产品维修和保修信息 .....	13

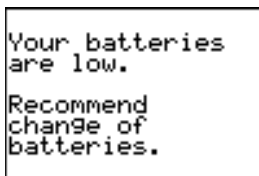
### 何时更换电池

TI-83 Plus 使用五节电池：四节 AAA 碱性电池和一节锂电池。在更换 AAA 电池时锂电池可为内存提供能量以保持内存信息。

当电池的电压级别下降到可用级别之下时，TI-83 Plus 在打开时显示以下信息：

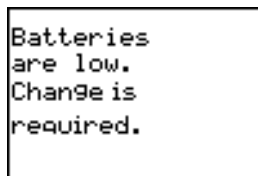
打开该设备时显示该信息。

试图下载应用程序时显示该信息。



```
Your batteries  
are low.  
Recommend  
change of  
batteries.
```

消息 A



```
Batteries  
are low.  
Change is  
required.
```

消息 B

显示消息 A 后，通常电池供电能力能维持一周或两周。（这一周或两周的时间是基于对碱性电池的测试得出，而其他类型的电池性能可能不同。）

更换电池前每次打开单元时，都显示供电不足的信息。如果在这两周内不更换电池，计算器可能自动关闭或无法开启，直到安装新电池。

如果显示消息 B，则必须立即更换电池以成功下载应用程序。

每三年或四年更换一次锂电池。

### 更换电池的影响

**不要**同时卸下这两种电池（AAA 和锂电池）。**不要**将电池的电能全部用完。如果您按照这个原则或页 B-3 中的步骤更换电池，可以在不丢失内存信息的情况下更换这两种电池之一。

---

## 电池的保护措施

在更换电池时，采取这些保护措施：

- 不要混用新电池和使用过的电池。不要将不同品牌（或同一品牌不同类型）的电池混合使用。
- 不要将充电电池和非充电电池混合使用。
- 安装电池时要根据极性（+ 和 -）图示安装。
- 在电池充电器中不要放置非充电电池。
- 对使用后的电池立即进行适当处理。不要将它们放到小孩能够得着的地方。
- 不要焚烧电池。

## 更换电池

要更换电池，请按下列步骤进行：

1. 关闭计算器。将滑动盖盖住键盘以避免不小心打开计算器。将计算器背面朝上。
2. 竖握计算器，用手指向下推电池盖顶部的插销，然后向您的方向推动盖子。

**注意：**要避免丢失内存中的信息，必须关闭计算器。不要同时卸下 AAA 电池和锂电池。

3. 同时更换四节 AAA 碱性电池，或者更换锂电池。
  - 要更换 AAA 碱性电池，将用过的四节 AAA 电池全部卸下并按照极性（+ 和 -）图将新电池装进电池盒。
  - 要更换锂电池，拧下锂电池盖上的螺钉，然后取下盖子。+（正极）向上安装新电池。盖上盖子，并用螺钉将其固定。请使用 CR1616 或 CR1620（或其他等价的）锂电池。
4. 盖上电池盒的盖子。如有必要，则打开计算器并调整显示的对比值（页 B-4 中步骤 1）。



### 疑难解答

要处理疑难点，请按下列步骤进行：

1. 如果屏幕无显示，可能需要调整对比度。  
要使屏幕变暗，则按一下 **[2nd]**，然后按住 **[▲]** 直到屏幕足够暗。  
要使屏幕变亮，则按一下 **[2nd]**，然后按住 **[▼]** 直到屏幕足够亮。
2. 如果显示出错误菜单，则按第 1 章所述的步骤进行。如有必要，关于具体错误的详细信息，请参阅页 B-5 到页 B-10。
3. 如果显示忙指示符（点划线），则图象或程序已暂停执行，TI-83 Plus 正在等待输入。按 **[ENTER]** 继续，或者按 **[ON]** 中断。
4. 如果显示棋盘板光标 (■)，则是您在提示中输入了最大数量的字符，或者内存已满。如果是内存已满，请：
  - 按 **[2nd]** **[MEM]** **2** 以显示 MEMORY MANAGEMENT DELETE 菜单。
  - 选择希望删除的数据类型，或选择 **1:All** 以获得所有类型的所有变量列表。在所显示的屏幕上会列出所选择类型的每个变量及每个变量所占用的字节数。
  - 按 **[▲]** 和 **[▼]** 以将选择光标 (▶) 移到希望删除的项目旁边，然后按 **[DEL]**。（第 18 章）。

- 
5. 如果计算器看起来根本不工作，请检查电池是否为新的，是否正确安装。请参阅页 B-2 和页 B-3 的电池信息。
  6. 在确定电池电源充足的情况下，如果计算器仍不工作，那么请依次尝试以下两种解决办法：
    - 按以下步骤下载计算器系统软件：
      1. 从计算器中取出一节电池，并在重装电池的过程中按住 **DEL** 键。这会强制计算器接受下载的系统软件。
      2. 使用 TI-GRAPH LINK™（可选）附件将计算器连接到个人计算机，以便将当前或新的软件代码下载到计算器。
    - 如果上述解决方法无效，请按以下步骤重置所有内存：
      1. 从计算器中取出一节电池，然后在重装电池的过程中按住 **CLEAR** 键。在继续按住 **CLEAR** 键时，同时按住 **ON** 键。当显示主屏幕时，松开这两个键。
      2. 按 **2nd** [MEM] 显示 MEMORY 菜单。
      3. 选择 **7:Reset** 以显示 RAM ARCHIVE ALL 菜单。
      4. 按 **▾ ▾** 以显示 ALL 菜单。
      5. 选择 **1:All Memory** 以显示 RESET MEMORY 菜单。
      6. 要继续进行重设，请选择 **2:Reset**。主屏幕上会出现 **MEM cleared** 的消息。

## 出错情况

当 TI-83 Plus 检测到错误时，会显示出 **ERR: 信息** 和出错菜单。第 1 章讲述了纠正错误的一般步骤。下表包含了每种错误类型、出错的可能原因和纠正建议。

错误类型	出错的可能原因和纠正建议
ARCHIVED	试图使用、编辑或删除一个已归档的变量。例如，如果 <b>L1</b> 已归档，则 <b>dim(L1)</b> 就会返回错误。
ARCHIVE FULL	试图归档一个变量，然而归档区已没有足够的空间来接收它。
ARGUMENT	函数或指令的参数个数不正确。请参阅附录 A 和有关章节。
BAD ADDRESS	试图发送或接收应用程序，但在传送过程中出现了错误（例如，电子干扰）。
BAD GUESS	<ul style="list-style-type: none"><li>在 <b>CALC</b> 运算中，所指定的 <b>Guess</b> 没在 <b>Left Bound</b> 和 <b>Right Bound</b> 之间。</li><li>对于 <b>solve(</b> 函数或方程求解器，所指定的 <i>估计值</i> 没在 <i>下限</i> 和 <i>上限</i> 之间。</li><li>估计值及其周围的点未定义。</li></ul> 检查函数的图象。如果方程有解，改变一下边界和/或最初的估计值。
BOUND	<ul style="list-style-type: none"><li>在 <b>CALC</b> 运算或者在用 <b>Select(</b> 中，定义的 <b>Left Bound &gt; Right Bound</b>。</li><li>在 <b>fMin(</b>、<b>fMax(</b>、<b>solve(</b> 或方程求解器中，输入了 <i>lower ≥ upper</i>。</li></ul>
BREAK	按 <b>[ON]</b> 键中断了程序的执行、中止了 <b>DRAW</b> 指令或者停止了表达式的求值。
DATA TYPE	所输入值或变量的数据类型错误。 <ul style="list-style-type: none"><li>对于函数（包括蕴含乘法）或指令，所输入参数的数据类型无效，例如要求输入实数的地方输入了复数。请参阅附录 A 和有关章节。</li><li>在编辑器中输入了不允许的类型，如在统计数组编辑器中将矩阵作为一个元素来输入。请参阅有关章节。</li><li>试图存储不正确的数据类型，如要在数组中存储矩阵。</li></ul>
DIM MISMATCH	试图执行引用了多个数组或矩阵的运算，但维数不匹配。
DIVIDE BY 0	<ul style="list-style-type: none"><li>试图除以零。在绘图期间不返回该错误。TI-83 Plus 考虑了图象上未定义值。</li><li>试图用垂直线进行线性回归。</li></ul>

错误类型	出错的可能原因和纠正建议
DOMAIN	<ul style="list-style-type: none"> <li>指定函数或指令的参数超出了有效范围。在绘图期间不返回该错误。TI-83 Plus 考虑图象上未定义值。请参阅附录 A 和有关章节。</li> <li>试图用 <math>-X</math> 进行对数回归或幂回归，或者用 <math>-Y</math> 进行指数回归或幂回归。</li> <li>试图用 <math>pmt2 &lt; pmt1</math> 来计算 <math>\Sigma Prn(</math> 或 <math>\Sigma Int(</math>。</li> </ul>
DUPLICATE	试图创建重复的群组名称。
Duplicate Name	因为在接收单元已经存在相同的变量名，所以要传送的变量不能被传送。
EXPIRED	试图运行具有有限试用期的应用程序，但该试用期已过期。
Error in Xmit	<ul style="list-style-type: none"> <li>TI-83 Plus 不能传送某一项。检查连接两个设备的线缆是否牢固，及接收设备是否为接收模式。</li> <li>在传送期间按 <math>\square</math> 中断了传送。</li> <li>试图执行从 TI-82 到 TI-83 Plus 的备份操作。</li> <li>试图从 TI-83 Plus 到 TI-82 传送数据（而不是 L1 到 L6）。</li> <li>在不使用 LINK SEND 菜单中的 5:Lists to TI82 的情况下，试图将数组 L1 到 L6 从 TI-83 Plus 传送到 TI-82。</li> </ul>
ID NOT FOUND	执行 SendID 命令，却找不到正确的计算器 ID 时，将会出现该错误。
ILLEGAL NEST	试图使用无效的函数来作为另一个函数的参数，如将用在表达式中的 seq( 函数作为参数用于 seq(。
INCREMENT	<ul style="list-style-type: none"> <li>seq( 的增量为 0 或者符号错误。在绘图期间不返回该错误。TI-83 Plus 允许在图象上有未定义的值。</li> <li>在 For( 循环中的增量为 0。</li> </ul>
INVALID	<ul style="list-style-type: none"> <li>试图在无效的地方引用变量或者使用函数。例如，Yn 不能引用 Y、Xmin、<math>\Delta X</math> 或 TblStart。</li> <li>试图引用从 TI-82 传送过来的但对于 TI-83 Plus 是无效的变量或函数。例如，用户可能将 Un-1 从 TI-82 传送到 TI-83 Plus，然后又试图引用它。</li> <li>在 Seq 方式下，没定义相位图方程就试图绘制相位图。</li> </ul>

错误类型	出错的可能原因和纠正建议
INVALID (cont.)	<ul style="list-style-type: none"> <li>在 <b>Seq</b> 方式下, 没有输入初始条件的正确数字就试图绘制递归序列。</li> <li>在 <b>Seq</b> 方式下, 试图引用除 <b>(n-1)</b> 或 <b>(n-2)</b> 以外的项。</li> <li>试图指定在当前图象方式下无效的图象样式。</li> <li>试图在没选定 (打开) 至少一个 <b>xy</b> 直线图或散射图的情况下使用 <b>Select()</b>。</li> </ul>
INVALID DIM	<ul style="list-style-type: none"> <li>所指定参数的维数不适合当前操作。</li> <li>所指定数组的维数不是 1 和 999 之间的整数。</li> <li>所指定矩阵的维数不是 1 和 99 之间的整数。</li> <li>试图求非方阵的逆。</li> </ul>
ITERATIONS	<ul style="list-style-type: none"> <li><b>solve()</b> 函数或方程求解器超过了允许迭代的最大次数。请检查函数的图象。如果方程有解, 则改变边界或最初的估计值, 或者改变这两个。</li> <li><b>irr()</b> 超过了允许迭代的最大次数。</li> <li>当计算 <b>I%</b> 时, 超过了迭代的最大次数。</li> </ul>
LABEL	<b>Goto</b> 指令中的标号未用程序中的 <b>Lbl</b> 指令定义。
MEMORY	<p>执行指令或函数时内存不足。在执行指令或函数前必须删除内存中的一些项 (第 18 章)。</p> <p>递归问题返回该错误, 例如画方程 <math>Y_1=Y_1</math> 的图象。</p> <p>用 <b>Goto</b> 跳出 <b>If/Then</b>、<b>For()</b>、<b>While</b> 或 <b>Repeat</b> 循环也可能返回该错误, 因为永远到达不了结束循环的 <b>End</b> 语句。</p>

错误类型	出错的可能原因和纠正建议
MemoryFull	<ul style="list-style-type: none"> <li>不能传送某一项，因为接收单元的可用内存不够。可以跳过这一项或者退出接收模式。</li> <li>在内存备份期间，接收单元的可用内存不足以接收发送单元内存中的所有项。一条信息表明了要做内存备份，发送单元必须删除的字节数。删除一些项后再试一试。</li> </ul>
MODE	试图存储另一种图象方式下的窗口变量或者试图在错误方式下执行指令，例如，在某种不是 <b>Func</b> 图象方式下执行 <b>DrawInv</b> 。
NO SIGN CHNG	<ul style="list-style-type: none"> <li><b>solve</b>( 函数或者方程求解器未发现符号改变。</li> <li>当 <b>FV</b>、<b>(N*PMT)</b> 和 <b>PV</b> 全 <math>\geq 0</math> 时，或者 <b>FV</b>、<b>(N*PMT)</b> 和 <b>PV</b> 全 <math>\leq 0</math> 时，试图计算 <b>I%</b>。</li> <li>当 <b>CFList</b> 和 <b>CFO</b> 全 <math>&gt; 0</math> 时，或者当 <b>CFList</b> 和 <b>CFO</b> 全 <math>&lt; 0</math> 时，试图计算 <b>irr</b>(。</li> </ul>
NONREAL ANS	在 <b>Real</b> 方式下，计算的结果为复数。在绘图期间不返回该错误。 <b>TI-83 Plus</b> 允许在图象上有未定义的值。
OVERFLOW	试图输入的数字或者已经计算出的数字超出了计算器的范围。在绘图期间不返回该错误。 <b>TI-83 Plus</b> 考虑了图象上未定义值。
RESERVED	试图不恰当使用系统变量。请参阅附录 A。
SINGULAR MAT	<ul style="list-style-type: none"> <li>奇异矩阵（行列式 = 0）作为 <b><math>\tau^{-1}</math></b> 的参数是无效的。</li> <li><b>SinReg</b> 指令或多项式回归产生了一个奇异矩阵（行列式 = 0），因为找不到解或解不存在。</li> </ul> <p>在绘图期间不返回该错误。<b>TI-83 Plus</b> 允许在图象上有未定义的值。</p>

## 出错情况 (续)

错误类型	出错的可能原因和纠正建议
SINGULARITY	在 <b>solve</b> ( 函数或方程求解器中表达式包含一个奇异点 (函数在此点无定义) )。检查函数的图象。如果方程有解, 则改变边界或最初的估计值, 或者这两个都改变。
STAT	试图进行统计计算, 而所用数组不合适。 <ul style="list-style-type: none"> <li>• 统计分析至少要有两个数据点。</li> <li>• <b>Med-Med</b> 在每个分隔项中必须至少要有三个点。</li> <li>• 当使用频数数组时, 它的元素必须为 <math>\geq 0</math>。</li> <li>• 对于一个直方图, <math>(X_{\max} - X_{\min}) / X_{\text{sc1}}</math> 必须是 <math>\leq 47</math>。</li> </ul>
STAT PLOT	当使用未定义数组的统计图在打开时, 试图显示图象。
SYNTAX	命令包含语法错误。寻找错置的函数、参数、括号或逗号的位置。请参阅附录 A 和有关的章节。
TOL NOT MET	指定了误差, 而算法不能按此误差返回一个精确的结果。
UNDEFINED	引用了当前没有定义的变量。例如, 引用了由于数组被编辑而不存在计算的统计变量, 或者引用了对当前计算是无效的变量, 如 <b>Med-Med</b> 之后的 <b>a</b> 。
VALIDATION	电子干扰导致链接失败或该计算器未被授权运行该应用程序。
VARIABLE	试图归档一个不能归档的变量, 或者试图取消归档一个应用程序或群组。 不能被归档的变量样例包括: <ul style="list-style-type: none"> <li>• 实数 <b>LRESID</b>、<b>R</b>、<b>T</b>、<b>X</b>、<b>Y</b>、<b>Theta</b>、<b>Vars</b> 下的统计变量、<b>STATISTICS</b> 菜单、<b>Yvars</b> 和 <b>AppldList</b>。</li> </ul>
VERSION	试图从另一台计算器接收不兼容的变量版本。
WINDOW RANGE	问题在于窗口变量。 <ul style="list-style-type: none"> <li>• 定义了 <math>X_{\max} \leq X_{\min}</math> 或 <math>Y_{\max} \leq Y_{\min}</math>。</li> <li>• 定义了 <math>\theta_{\max} \leq \theta_{\min}</math> 和 <math>\theta_{\text{step}} &gt; 0</math> (或者与之相反)。</li> <li>• 试图定义 <b>Tstep=0</b>。</li> <li>• 定义了 <math>T_{\max} \leq T_{\min}</math> 和 <math>T_{\text{step}} &gt; 0</math> (或者与之相反)。</li> <li>• 窗口变量太小或太大以致于不能正确地绘图。您可能试图缩小或放大窗口到超过了 <b>TI-83 Plus</b> 的数值范围的一点。</li> </ul>
ZOOM	<ul style="list-style-type: none"> <li>• 在 <b>ZBox</b> 中定义一点或线, 而不是框。</li> <li>• <b>ZOOM</b> 操作返回一个数学错误。</li> </ul>

### 计算精度

为提高精度，TI-83 Plus 内部采用的位数比显示的要多。存储在内存中的数值最多为 14 位并带有两位指数。

- 可以在窗口变量中最多存储 10 位数（对于 **Xscl**、**Yscl**、**Tstep** 和 **θstep** 来说最多为 12 位）。
- 显示的数值按指定的方式设置（最多是 10 位数和两位的指数）舍入。
- 在 **Float** 方式下 **RegEQ** 最多能显示 14 位。使用定点十进制表示设置而不用 **Float** 使 **RegEQ** 结果以指定的小数位数来舍入和保存。

### 图象精度

**Xmin** 表示最左边象素点的中心值，**Xmax** 表示最右边象素旁边点的中心值。（最右边象素是为忙指示器保留的。）**ΔX** 是两个相邻象素中心间的距离。

- 在 **Full** 屏幕方式下，**ΔX** 是用  $(Xmax - Xmin) / 94$  来计算的。在 **G-T** 拆分屏幕方式下，**ΔX** 是用  $(Xmax - Xmin) / 46$  来计算的。
- 如果在 **Full** 屏幕方式下从主屏幕或程序中输入 **ΔX** 的值，用  $Xmin + ΔX * 94$  计算出 **Xmax**。在 **G-T** 拆分屏幕方式下，用  $Xmin + ΔX * 46$  计算出 **Xmax**。

**Ymin** 表示最邻近底部象素点的中心值，**Ymax** 表示顶部象素点的中心值。**ΔY** 是两相邻象素中心间的距离。

- 在 **Full** 屏幕方式下，**ΔY** 是用  $(Ymax - Ymin) / 62$  来计算的。在 **Horiz** 拆分屏幕方式下，**ΔY** 是用  $(Ymax - Ymin) / 30$  来计算的。在 **G-T** 拆分屏幕方式下，**ΔY** 是用  $(Ymax - Ymin) / 50$  来计算的。
- 如果在 **Full** 屏幕方式下从主屏幕或程序输入 **ΔY** 的值，则用  $Ymin + ΔY * 62$  计算出 **Ymax**。在 **Horiz** 拆分屏幕方式下，用  $Ymin + ΔY * 30$  计算出 **Ymax**。在 **G-T** 拆分屏幕方式下，用  $Ymin + ΔY * 50$  计算出 **Ymax**。



图象精度 (续)

当选择 **Float** 方式时, 光标坐标值显示为八个字符数 (可能包括负号、小数点和指数)。用最精确的八位数来更新 **X** 和 **Y**。

用误差  $1E-5$  计算 **CALCULATE** 菜单上的 **minimum** 和 **maximum**; 用误差  $1E-3$  计算 **|f(x)dx**。因此, 所显示的结果可能没有精确到所全部显示的八位数。对于大多数函数, 至少精确到五位数。对于 **MATH** 菜单上的 **fMin()**、**fMax()** 和 **fnInt()** 以及 **CATALOG** 上的 **solve()** 来说, 可以指定它们的误差。

函数范围

函数	输入值范围
<b>sin</b> $x$ , <b>cos</b> $x$ , <b>tan</b> $x$	$0 \leq  x  < 10^{12}$ (弧度或度)
<b>sin</b> <sup>-1</sup> $x$ , <b>cos</b> <sup>-1</sup> $x$	$-1 \leq x \leq 1$
<b>ln</b> $x$ , <b>log</b> $x$	$10^{-100} < x < 10^{100}$
<b>e</b> <sup><math>x</math></sup>	$-10^{100} < x \leq 230.25850929940$
<b>10</b> <sup><math>x</math></sup>	$-10^{100} < x < 100$
<b>sinh</b> $x$ , <b>cosh</b> $x$	$ x  \leq 230.25850929940$
<b>tanh</b> $x$	$ x  < 10^{100}$
<b>sinh</b> <sup>-1</sup> $x$	$ x  < 5 \times 10^{99}$
<b>cosh</b> <sup>-1</sup> $x$	$1 \leq x < 5 \times 10^{99}$
<b>tanh</b> <sup>-1</sup> $x$	$-1 < x < 1$
$\sqrt{x}$ (实数方式)	$0 \leq x < 10^{100}$
$\sqrt{x}$ (复数方式)	$ x  < 10^{100}$
$x!$	$-0.5 \leq x \leq 69$ , 其中 $x$ 是 .5 的倍数

函数结果

函数	结果范围
<b>sin</b> <sup>-1</sup> $x$ , <b>tan</b> <sup>-1</sup> $x$	$-90^\circ$ 到 $90^\circ$ 或者 $-\pi/2$ 到 $\pi/2$ (弧度)
<b>cos</b> <sup>-1</sup> $x$	$0^\circ$ 到 $180^\circ$ 或者 $0$ 到 $\pi$ (弧度)

### 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。

e-mail 地址: **ti-cares@ti.com**

internet 地址: **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，请参阅随该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。

! (阶乘) 函数, 2-21

#SAMPLES

DATA LOGGER, 14-22

➔ **dim**(指定维数) 函数, 10-13, 11-11

➔ (存储指令), 1-15, A-28

**Fcdf**(, **Fpdf**) 函数, 13-32

▶**Dec** (小数) 函数, 2-5

▶**DMS** (度/分/秒) 函数, 2-24

▶**DMS** (度/分/秒) 输入表示, 2-23

▶**Eff**(实际利率) 函数, 14-13

**∫f(x)dx** (计算) 运算, 3-28

▶**Frac** (分数) 函数, 2-5

**ΣInt**(利息支付总额) 函数, 14-10

**ΔList**(函数, 11-12

▶**Nom**(额定利率) 函数, 14-13

▶**Polar** 函数, 2-19

**ΣPrn**(本金支付总额) 函数, 14-10

▶**Rect** (直角) 函数, 2-19

**ΔTbl** (运算表步长), 7-3

**χ<sup>2</sup>-Test** (平方检验), 13-22

**χ<sup>2</sup>cdf**(, **χ<sup>2</sup>pdf**) 函数, 13-31

**10<sup>^</sup>**(幂) 函数, 2-4

**1-PropZInt** (单比例 *z* 置信区间),  
13-20

**1-PropZTest** (单比例 *z* 检验), 13-14

**1-Var** 和 **2-Vars** 统计, 12-25

**2-PropZInt** (双比例 *z* 置信区间),  
13-21

**2-PropZTest** (双比例 *z* 检验), 13-15

**2-SampFTest** (双样本 *F* 检验), 13-23

**2-SampTInt** (双样本 *t* 置信区间), 13-19

**2-SampTTest** (双样本 *z* 检验),  
13-13

**2-SampZInt**(双样本 *z* 置信区间), 13-18

**2-SampZTest** (双样本 *z* 检验),  
13-12

**a+bi** 直角坐标复数方式, 1-12

**abs**(绝对值) 函数, 2-13, 2-19, 10-10

**and** (布尔) 运算符, 2-26

**ANGLE**

菜单, 2-23

运算, 2-23, 2-24

**angle** 函数, 2-19

**ANOVA**(单方向方差分析), 13-25

**ANOVA** 公式, A-52

**Ans** (最后结果)

存储区, 1-19

在表达式中使用, 1-19

Applications 菜单, 14-2, 14-16

**arcsine**, **arccosine**, **arctangent** 双  
曲函数, 15-10

**AsmComp**(编译汇编语言程序),  
16-24

**AsmPrgm**(汇编语言程序), 16-24

**augment**(合并) 函数, 10-14, 11-15

Automatic Power Down™ (APD™), 1-2

**AxesOff**, **AxesOn** 指令, 3-14

**bal**(分期付款) 函数, 14-10

**binomcdf**(, **binompdf**) 函数, 13-33

**Boxplot** (标准框图) 指令, 12-33

**C/Y** (每年的复利计算周期) 变量,  
14-15

**CALCULATE**

菜单, 3-25

运算, 3-25

**CATALOG**, 定义, 从其中选择项目,  
15-2

**CATALOG** 中的双曲函数, 15-10

**CBL/CBR** 应用程序

运行步骤, 14-2, 14-16

**Check RAM** (内存) 屏幕, 18-2

**Circle**(指令, 8-11

**Clear Entries** 指令, 18-5

**ClrAllLists** (内存) 指令, 18-5

**ClrDraw** 指令, 8-4

**ClrList** (清除数组) 函数, 12-20

**conj**(共轭) 函数, 2-18

**Connected** (绘图方式) 指令, 1-11

**CoordOff**, **CoordOn** 指令, 3-14

**cos**(, 余弦函数, 2-3

**cosh**(, **cosh**) 函数, 15-10

**CubicReg** (三次回归) 指令, 12-26

**cumSum**(累积和) 函数, 10-15,  
11-12

**DATA LOGGER**

选项, 14-21

**dbd**(日期期间的天数) 函数, 14-14

**Degree** 方式指令, 1-11

**DelVar** (删除变量内容) 指令, 16-16

**DependAsk**, **DependAuto** 指令, 7-3

**det**(行列式) 函数, 10-12

**DiagnosticOff**, **DiagnosticOn** 指令,  
12-23

**dim**(维数) 函数, 10-12, 10-13, 11-11

**DIRECTNS**

DATA LOGGER, 14-22

GAUGE, 14-20

**Disp** (显示) 指令, 16-19

- DispGraph** (显示图象) 指令, 16-20
- DispTable** (显示运算表) 指令, 16-20
- DISTR DRAW** (分布绘图) 菜单, 13-35
- DISTR** (分布函数) 菜单, 13-29
- Dot** (绘图方式) 指令, 1-11
- DRAW POINTS** 菜单, 8-14
- DRAW STO** (图形存储) 菜单, 8-17
- DrawF** 指令, 8-9
- DrawInv** 指令, 8-9
- DRAW** 菜单, 8-3
- DRAW** 操作
- 切线, 8-8
  - 清除图形, 8-4
  - 使用画笔注释图象, 8-13
  - 图象上的点, 8-14, 8-15
  - 图象上的函数和反函数, 8-9
  - 图象上的水平线和垂直线, 8-6
  - 图象上的线段, 8-5
  - 圆, 8-11
  - 准备画图, 8-3
- DS<** (减 1 并跳过) 指令, 16-15
- DuplicateName** 菜单, 19-7
- dy/dx** (计算) 运算, 3-28
- e<sup>^</sup>** (幂) 函数, 2-4
- Else** 指令。参阅 **If-Then-Else**
- End**
- DATA LOGGER**, 14-22
- End** 指令, 16-13
- Eng** (工程记数法) 方式, 1-10
- ENTRY** 键 (最后输入), 1-17
- EquString** (方程到字符串) 函数, 15-7
- Equation Operating System (EOS™)**, 1-23, 1-24
- expr** (字符串到表达式) 函数, 15-7
- ExpReg** (指数回归) 指令, 12-26
- ExprOff**, **ExprOn** 指令, 3-14
- Fill** (指令, 10-13, 11-11)
- FINANCE CALC** 菜单, 14-6
- FINANCE VARS** 菜单, 14-15
- Fix** (定点十进制) 方式, 1-10
- Float** (浮点十进制) 方式, 1-10
- fMax** (函数, 2-6)
- fMin** (函数, 2-6)
- fnInt** (函数, 2-7)
- FnOff**, **FnOn** 指令, 3-8
- For** (指令, 16-11)
- fPart** (分数部分) 函数, 10-11
- fPart** (小数部分) 函数, 2-14
- Full** (屏幕) 方式, 1-12
- Func** 方式, 1-11
- GAUGE**
- 选项, 14-19
  - 注释, 14-20
- gcd** (最大公约数) 函数, 2-15
- geomtcdf**, **geomtpdf** (函数, 13-34)
- Get** (从 CBL 中获取) 指令, 16-22
- GetCalc** (指令, 16-22)
- Goto** 指令, 16-14
- GraphStyleI** 指令, 16-16
- GridOff**, **GridOn** 指令, 3-14
- G-T** (图象 — 运算表) 拆分屏幕方式, 9-5, 9-6
- G-T** (图形 — 表) 拆分屏幕方式, 1-12
- Histogram** 指令, 12-32
- Horiz** (水平) 拆分屏幕方式, 1-12, 9-4
- Horizontal** (线) 指令, 8-7
- identity** (函数, 10-13)
- If** 指令
- If**, 16-10
  - If Then**, 16-10
  - If-Then-Else**, 16-11
- imag** (虚数分量) 函数, 2-18
- IndpntAsk**, **IndpntAuto** 指令, 7-3
- Input** 指令, 16-17
- inString** (字符串中) 函数, 15-7
- int** (最大整数) 函数, 2-14, 10-11
- intersect** (计算) 运算, 3-27
- INTRVL (SEC)**
- DATA LOGGER**, 14-22
- invNorm** (函数, 13-30)
- iPart** (整数部分) 函数, 2-14, 10-11
- irr** (内部回报率) 函数, 14-9
- IS>** (增 1 并跳过) 指令, 16-14
- L** (用户创建的数组变量) 指令, 11-16
- LabelOff**, **LabelOn** 指令, 3-14
- Lbl** (标记) 指令, 16-14
- lcm** (最小公倍数) 函数, 2-15
- LDCMT (GAUGE)**, 14-23
- LDIST**
- GAUGE**, 14-20, 14-23
  - LDREF (GAUGE)**, 14-20
- length** (函数, 15-8)
- Light** 探测器, 14-18
- Line** (指令, 8-5)
- LINK RECEIVE** 菜单, 19-7
- LINK SEND** 菜单, 19-6

**LinReg** (线形回归) 指令, 12-25  
**LinRegTTest** (线形回归 t 检验),  
13-24  
**LIST MATH** 菜单, 11-17, 11-18  
**List**( 函数, 11-12  
**ListMatr**( (数组到矩阵) 函数, 10-14,  
11-15  
**LISTS OPS** 菜单, 11-10  
**LLCMT (GAUGE)**, 14-23  
**LLIGHT**  
GAUGE, 14-20, 14-23  
**LLREF (GAUGE)**, 14-20  
**ln**( , 对数函数, 2-4  
**LnReg** (线形回归) 指令, 12-26  
**Logistic** 指令, 12-27  
**LTCMT (GAUGE)**, 14-23  
**LTEMP**  
GAUGE, 14-20, 14-23  
**LTREF (GAUGE)**, 14-20  
**LVCMT (GAUGE)**, 14-23  
**LVOLT**  
GAUGE, 14-20, 14-23  
**LVREF (GAUGE)**, 14-20  
**MATH CPX** (复数)  
菜单, 2-18  
运算, 2-18  
**MATH NUM** (数字)  
菜单, 2-13  
运算, 2-13  
**MATH PRB** (概率)  
菜单, 2-20  
运算, 2-20  
**MATH** 菜单, 2-5  
**Matrlist**( (矩阵到数组) 函数, 10-14,  
11-16  
**MATRIX MATH**  
菜单, 10-12  
操作, 10-12  
**max**( (最大值) 函数, 2-15, 11-17  
**maximum/minimum** (计算) 运算,  
3-27  
**mean**( 函数, 11-17  
**median**( 函数, 11-17  
**Med-Med** (中位数-中位数) 指令,  
12-25  
**MEMORY** 菜单, 18-2  
**Menu**( 指令, 16-15  
**min**( (最小值) 函数, 2-15, 11-17

**ModBoxplot** (改进式框图) 指令,  
12-32  
**nCr** (组合数) 函数, 2-21  
**nDeriv**( 函数, 2-7  
**normalcdf**( 函数, 13-30  
**normalpdf**( 函数, 13-29  
**Normal** 方式, 1-10  
**NormProbPlot** (正态概率曲线) 指令,  
12-33  
**not**( (布尔) 运算符, 2-26  
**nPr** (排列数) 函数, 2-21  
**npv**( (净现值) 函数, 14-9  
**or** (布尔) 运算符, 2-26  
**Output**( 指令, 9-6, 16-20  
**P>Rx**( , **P>Ry**( (极坐标到直角坐标的  
换算) 函数, 2-24  
**P/Y** (每年的支付周期) 变量, 14-15  
**Par** 方式, 1-11  
**Pause** 指令, 16-13  
**Pen** 指令, 8-13  
**PLOT**  
DATA LOGGER, 14-22  
**PlotsOff** , **PlotsOn** 指令, 12-35  
**Pmt\_Bgn** (期初支付) 函数, 14-14  
**Pmt\_End** (期末支付) 函数, 14-14  
**poissoncdf**( 函数, 13-34  
**poissonpdf**( 函数, 13-33  
**PolarGC** (图象极坐标) , 3-13  
**PoI** 方式, 1-11  
**PRGM CTL** (程序控制) , 16-9  
**PRGM EDIT** 菜单, 16-8  
**PRGM EXEC** 菜单, 16-8  
**PRGM I/O** (输入/输出) , 16-17  
**prgm** (执行) 指令, 16-16  
**prod**( (积) 函数, 11-18  
**Prompt** 指令, 16-19  
**Pt-Change**( 指令, 8-15  
**Pt-Off**( , **Pt-On** 指令, 8-15  
**Pt-On**( 指令, 8-14  
**PwrReg** (幂回归) 指令, 12-27  
**Pxl-Change**( 函数, 8-16  
**Pxl-Off**( , **Pxl-On**( 指令, 8-16  
**Pxl-Test**( 函数, 8-16  
**QuadReg** (二次回归) 指令, 12-25  
**QuartReg** (四次回归) 指令, 12-26  
**R>Pr**( , **R>Pθ**( (直角坐标到极坐标的换  
算) 函数, 2-24  
**Radian** 方式指令, 1-11  
**rand** (随机数) 函数, 使用, 2-20

- randBin**( (随机二项式) 函数, 2-22)  
**randInt**( (随机整数) 函数, 2-22)  
**randM**( (随机矩阵) 函数, 10-13)  
**randNorm**( (随机正态) 函数, 2-22)  
**RANGER** 程序, 14-18  
**RCL** (重新调用函数), 1-16  
**re<sup>θi</sup>** 极坐标复数方式, 1-12  
**Real**( (实数分量) 函数, 2-18)  
**RealTme**  
     DATA LOGGER, 14-22  
**Real** 方式, 1-12  
**RecallGDB** 指令, 8-20  
**RecallPic** 指令, 8-18  
**RectGC** (图象直角坐标), 3-13  
**ref**( (行梯形) 函数, 10-15)  
**Repeat** 指令, 16-12  
**RESET** 菜单, 18-6  
**Return** 指令, 16-16  
**Root**(nth) 函数, 2-6  
**round**( 函数, 2-13, 10-10)  
**row+**(, \*row(, \*row+ (矩阵行) 函数,  
     10-16)  
**rowSwap**( 函数, 10-16)  
**rref**( (简化的行梯形) 函数, 10-15)  
**Scatter** 指令, 12-31  
**Sci** (科学记数法) 方式, 1-10  
**Select**( 函数, 11-12, 11-13)  
**Send**( (发送到 CBL) 指令, 16-22)  
**Seq**( (序列) 函数, 11-12)  
**Seq** (序列) 图象方式, 1-11  
**Sequential** (绘图顺序) 方式, 1-12  
**SetUpEditor** 指令, 12-21  
**Shade $\chi^2$**  指令, 13-36  
**ShadeF**( 指令, 13-36)  
**Shade**( 指令, 8-10)  
**Shade\_t** 指令, 13-36  
**ShadeNorm**( 指令, 13-35, 13-36)  
**Simul** (同时) 方式, 1-12  
**sin**(, 正弦函数, 2-3)  
**sinh**(, **sinh**( 函数, 15-10)  
**SinReg** (正弦回归), 12-27, 12-28  
**solve**( 方程, 2-12)  
**Solver**, 2-8  
**Sonic** 探测器, 14-18  
**SortA**( (按升序排序) 函数, 11-10,  
     12-20)  
**SortD**( (按降序排序) 函数, 11-10,  
     12-20)  
**STAT CALC**, 12-24  
**STAT EDIT**, 12-20  
**STAT TESTS**  
      $\chi^2$ -TEST (平方检验), 13-22  
     **1-PropZInt** (单比例  $z$  置信区间),  
         13-20  
     **1-PropZTest** (单比例  $z$  检验),  
         13-14  
     **2-PropZInt** (双比例  $z$  置信区间),  
         13-21  
     **2-PropZTest** (双比例  $z$  检验),  
         13-15  
     **2-SampFTest** (双样本 F 检验),  
         13-23  
     **2-SampTInt** (双样本  $t$  置信区间),  
         13-19  
     **2-SampTTest** (双样本  $z$  检验), 13-13  
     **2-SampZInt** (双样本  $z$  置信区间),  
         13-18  
     **2-SampZTest** (双样本  $z$  检验), 13-12  
     **ANOVA**( (单向方差分析), 13-25)  
     **LinRegTTest** (线性回归  $t$  检验),  
         13-24  
     **TInterval**(单样本  $t$  置信区间), 13-17  
     **T-Test**, 13-11  
     **ZInterval** (单样本  $z$  置信区间),  
         13-16  
     **Z-Test**, 13-10  
**STAT TESTS** 菜单, 13-9  
**stdDev**( (标准差) 函数, 11-18)  
**Stop** 指令, 16-16  
**StoreGDB** 指令, 8-19  
**StorePic** 指令, 8-17  
**StringEqu**( (字符串到方程) 函数, 15-  
     8)  
**sub**( (子串) 函数, 15-8)  
**sum**( (和) 函数, 11-18)  
**TABLE SETUP** 屏幕, 7-3  
**tan**(, 正切函数, 2-3)  
**Tangent**( (线) 指令, 8-8)  
**tanh**(, **tanh**( 函数, 15-10)  
**TblStart** (运算表变量), 7-3  
**tcdf**( 函数, 13-31)  
**Temp** 探测器, 14-18  
**TEST LOGIC** (布尔), 2-26  
**TEST** (菜单关系), 2-25  
**Text**( 指令, 8-12, 9-6)  
**Then** 指令. 参阅 If-Then  
**TI-82** 和未解决的差异, 19-13  
**TI-83 Plus** 计算器

功能, 17, 18  
键盘, 2, 3  
**Time** (格式) 指令, 6-8  
**TInterval** (单样本  $t$  置信区间), 13-17  
**tpdf**( 函数, 13-30  
**Trace** 指令, 3-19  
**T-Test** 指令, 13-11  
**TVM** (货币的时间价值) 变量, 14-15  
**tvm\_FV** (TVM 未来值) 函数, 14-8  
**tvm\_I%** (TVM 利率) 函数, 14-8  
**tvm\_N** (TVM # 支付周期) 函数, 14-8  
**tvm\_Pmt** (TVM 支付) 函数, 14-7  
**tvm\_PV** (TVM 现值) 函数, 14-8  
UNITS  
    **DATA LOGGER**, 14-22  
    **GAUGE**, 14-19  
**uv** 坐标轴, **uw** 坐标轴格式指令, 6-8  
**value** (计算) 运算, 3-25  
**variance**( 函数, 11-18  
**VARS STRING** 菜单, 15-4  
**Vertical** (线) 指令, 8-7  
**Volt** 探测器, 14-18  
**vw** 坐标轴格式指令, 6-8  
**web** 图, 序列图象, 6-11  
**Web** 指令, 6-8  
**While** 指令, 16-12  
**Xfact** 和 **Yfact**, 缩放因子, 3-24  
**xor** (布尔) 异或运算符, 2-26  
**xyLine** 指令, 12-31  
**Y=** 编辑器中的函数  
    取消选择, 3-7  
    显示, 3-5  
**Y=** 编辑器中的图象样式图标, 3-9  
**Ymax**  
    **DATA LOGGER**, 14-22  
**Ymin**  
    **DATA LOGGER**, 14-22  
**ZBox**, 3-20  
**ZDecimal**, 3-21  
**zero** (计算) 运算, 3-26  
**ZInteger**, 3-22  
**ZInterval** (单样本  $z$  置信区间), 13-16  
**Zoom In**, **Zoom Out**, 3-21  
**ZOOM MEMORY** 菜单, 3-23  
**Zoom** (函数图象), 3-20  
**ZoomFit**, 3-22  
**ZoomRcl**, 3-23  
**ZoomStat**, 3-22

**ZoomSto**, 3-23  
**ZPrevious**, 3-23  
**ZSquare**, 3-21  
**ZStandard**, 3-22  
**Z-Test** 指令, 13-10  
**ZTrig**, 3-22

## - B -

保存  
    用 **Input** 保存变量值, 16-18  
保存函数图象, 3-3  
备份内存, 19-16  
编程  
    编辑程序, 16-7  
    程序, 16-7  
    程序控制指令, 16-9  
    创建新程序, 16-4  
    存储于 **eqn** 的方程, 2-12  
    调用子程序, 16-23  
    定义程序, 16-4  
    管理内存, 16-4  
    拷贝和重命名程序, 16-8  
    求解器编辑器中的变量值, 2-9  
    删除程序, 16-4  
    输入命令, 16-6  
    执行程序, 16-6  
    中断程序, 16-6  
编辑键, 击键。结果, 1-8  
便捷绘图, 3-15  
变量  
    **Vars** 和 **Y-Vars** 菜单, 1-22  
    **Y=** 编辑器, 7-4  
    重新调用值, 1-16  
    检验和区间输出, 13-28  
    矩阵名称, 10-3  
    名称和定义项, 1-13, 1-14  
    数组名称, 11-3  
    统计, 12-29  
    显示和存储值, 1-15  
    用户和系统, A-50  
    在方程求解器中, 求解, 2-10  
    在求解器编辑器中, 编辑, 2-9  
    字符串, 15-4, 15-5  
表达式  
    定义, 1-6  
    输入, 1-6  
表达式中的 **Y=** 函数, 计算, 3-6  
表盘图 (**GAUGE**), 14-19

## - B (续) -

布尔运算符, 2-26

## - C -

菜单

Vars 和 Y-Vars, 1-22

入门, 4

使用和滚动, 1-20

退出, 1-21

选择项目, 1-21

在菜单中移动, 4

菜单图, A-39

参数方程

编辑, 4-5

在 Y= 编辑器中定义, 4-5

在 Y= 编辑器中选择和取消选择, 4-5

参数图象

窗口变量和 Y-Vars 菜单, 4-6

定义和显示, 4-4

跟踪, 4-7

计算运算, 4-8

设置窗口变量, 4-5

设置图象方式, 4-4

设置图象格式, 4-6

缩放运算, 4-8

显示图象, 4-6

选择图象样式, 4-4

研究自由移动光标, 4-7

移动跟踪光标, 4-8

插入显示光标, 1-5

拆分屏幕方式

**G-T** 方式, 9-5

**Horiz** 方式, 9-6

从键盘设置, 9-3

从主屏幕或程序设置, 9-6

程序流向, 控制, 16-9

充满显示光标, 1-5

重新调用

变量值, 1-16

图象图形, 8-18

重新设置

TI-83 Plus, 5

内存, 18-6

传送

传送到其他, 19-10

传送数组到 TI-82, 19-12

传送项目到另一个单元, 19-8

从 TI-82 到, 19-13

错误原因, 19-8

## - C (续) -

停止传送, 19-8

选择发送项, 19-6

从另一个单元接收项目, 19-7

从主屏幕和程序选择函数, 3-8

存储

变量, 1-15

多个 ENTRY 值, 1-18

结果, 1-19

图象数据库 (GDBs), 8-19

图象图形, 8-17

存储函数图象。参阅 保存函数图象

存储指令。参阅 →

错误消息, B-6

错误原因

传送, 19-8

清除, 1-2

诊断和修正错误, 1-25

## - D -

大写字母显示光标, 1-5

递归序列, 6-6

第二功能显示光标, 1-5

电池

保护措施, B-2

电量不足消息, 1-3

更换方法, B-3

更换时间, B-2

类型, 1-2

替换时间, 1-3

对比度设置。参阅 调节显示对比度

多个 ENTRY 值, 存储, 1-18

## - F -

发送。参阅 传送

反

函数, 10-10

三角函数, 2-3

方程求解器, 2-8

方程有多根, 2-12

方式

复数, 2-16, 2-17

由程序设置, 1-9

方式设置

**Connected** (绘图方式), 1-11

**Degree**, 1-11



- F (续) -

**Dot** (绘图方式), 1-11  
**Eng** (工程记数法), 1-10  
**Fix** (定点十进制), 1-10  
**Float** (浮点十进制), 1-10  
**Full** (屏幕), 1-12  
**Func**, 1-11  
**G-T** (图形 — 表), 1-12  
**Horiz** (水平), 1-12  
**Normal**, 1-10  
**Par**, 1-11  
**Pol**, 1-11

**Radian**, 1-11

**Real**, 1-12

**Sci** (科学记数法), 1-10

**Seq**, 1-11

**Sequential**, 1-12

**Simul** (同时), 1-12

复数, 1-12

弧度, 2-24

角度, 2-24

访问以前输入, 1-17

非递归序列, 6-5

分布函数

**binomcdf**(, 13-33, A-4

**binompdf**(, 13-33, A-4

$\chi^2$ **cdf**(, 13-31, A-3

**df**( $\chi^2$ **p**, 13-31, A-4

$\chi^2$ **pdf**, 13-32, A-8

**Fpdf**(, 13-32, A-9

**geometcdf**(, 13-34, A-10

**geometpdf**(, 13-34, A-10

**invNorm**(, 13-30, A-12

**normalcdf**(, 13-30, A-17

**normalpdf**(, 13-29, A-17

**poissoncdf**(, 13-34, A-19

**poissonpdf**(, 13-33, A-19

**tcdf**(, 13-31, A-29

**tpdf**(, 13-30, A-30

分期付款

公式, A-58

计算计划表, 14-10

负函数, 2-4

复数方式, 1-12, 2-16, 2-17

复数结果, 解释, 2-16

- G -

公式

**ANOVA**, A-52

分期付款, A-58

货币的时间价值, A-57

利率换算, A-59

日期期间的天数, A-60

双样本**F**检验, A-53

双样本 **t** 统计, A-54

现金流转, A-59

关系运算, 2-25, 10-11

- H -

函数, 定义, 1-7

函数和指令, 全部列表, A-2

函数图象

$\Delta x$  和  $\Delta y$  窗口变量, 3-12

存储数值到窗口变量, 3-12

定义, 3-3

覆盖图象上的函数, 3-16

改变窗口变量, 3-11

跟踪, 3-18, 3-19

观察窗口, 3-11

绘制一族曲线, 3-16

检查和改变方式设置, 3-4

精度, 3-17

设置窗口变量, 3-11, 3-12

设置方式, 3-4

设置格式, 3-13, 3-14

设置图象样式, 3-9, 3-10

使用快速缩放, 3-19

显示, 3-3, 3-11, 3-15, 3-16

显示和改变格式设置, 3-13

研究自由移动光标, 3-17

阴影, 3-10

由程序设置方式, 3-4

在 **Y=** 编辑器中编辑, 3-5

在 **Y=** 编辑器中定义, 3-5

在 **Y=** 编辑器中定义, 主屏幕, 程序,  
3-6

暂停或中止绘图, 3-15

合并字符串, 15-6

弧度方式指令, 2-24

## - H (续) -

换算函数

- ▶**Dec** (小数) 函数, 2-5
- ▶**DMS** (度/分/秒) 函数, 2-24
- ▶**Frac** (分数) 函数, 2-5
- ▶**Polar** 函数, 2-19
- ▶**Rect** (直角) 函数, 2-19
- Equ**▶**String**( (方程到字符串), 15-7
- List**▶**matr**( (数组到矩阵) 函数, 10-14, 11-15
- Matr****list**( (矩阵到数组) 函数, 10-14, 11-16
- P**▶**Rx**, **P**▶**Ry** (极坐标到直角坐标), 2-24
- R**▶**Pr**, **R**▶**θ** (直角坐标到极坐标), 2-24
- String**▶**Equ**( (字符串到方程), 15-8

回归模型

- 分析显示方式, 12-23
- 功能, 12-22
- 自动回归方程, 12-22
- 自动离差数组功能, 12-22

货币的时间价值

- 公式, A-57
- 计算, 14-7

## - J -

极坐标, 复数, 2-17

极坐标方程

- 定义, 5-4
- 在 **Y=** 编辑器中选择和取消选择, 5-4

极坐标图象

- 窗口变量和 **Y-Vars** 菜单, 5-5
- 跟踪, 5-6
- 计算运算, 5-6
- 设置图象方式, 5-3
- 设置图象格式, 5-5
- 缩放运算, 5-6
- 显示 **Y=** 编辑器, 5-3
- 显示图象, 5-5
- 选择窗口变量, 5-4
- 选择图象样式, 5-3
- 研究自由移动光标, 5-6
- 移动跟踪光标, 5-6

计算

- 货币的时间价值, 14-7
- 利息换算, 14-13
- 现金流转, 14-9

## - J (续) -

计算方程的顺序。参阅 **Equation Operating System (EOS)**

继续表达式, 1-19

检验输出变量, 13-28

键盘

- 布局, 2, 3
- 数学运算, 2-3, 2-4

角度方式指令, 2-24

阶乘。参阅 **!** (阶乘) 函数

金融操作

- TVM Solver** (货币的时间价值) 函数, 14-5, 14-6

分期付款, 计算, 14-10

计算日期期间的天数, 14-14

利息换算, 计算, 14-13

现金流转, 计算, 14-9

支付方法, 定义, 14-14

精度信息

- 函数范围和结果, B-12
- 函数图象, 3-17
- 计算和图象, B-11

矩阵

编辑矩阵元素, 10-6

变量, 10-3

从内存删除, 10-4

定义, 10-3

反函数, 使用, 10-10

访问元素, 10-8

关系运算, 10-11

观察矩阵元素, 10-5

拷贝矩阵, 10-8

幂函数, 使用, 10-10

使用 **dim**( 创建矩阵和重新定义矩阵维数, 10-13

使用数学函数, 10-9

维数, 10-3

显示矩阵, 10-8

显示元素, 10-4

行运算, 10-15, 10-16

选择, 10-3

在表达式中输入和使用, 10-7

## - K -

开启和关闭

**TI-83 Plus**, 1-2

**Y=** 编辑器中的统计图, 3-7

象素, 8-16

- K (续) -

科学记数法, 输入数字, 1-7

- L -

来自程序的图象函数, 15-8

立方和立方根函数, 2-6

利率换算公式, A-59

连接

连接到 CBL 系统, 19-4

连接两个单元, 19-4

链接。参阅链接操作

链接操作

TI-82 到, 19-4

链接到个人电脑或 Macintosh, 19-4

链接两个单元, 19-4

使用线缆链接两个单元, 19-4

- M -

忙指示器, 1-4

命名数组, 11-3

- N -

内存

备份, 19-16

传送时内存不足, 19-7

检查可用内存, 18-2

清除数组, 18-5

清除项目, 18-5

删除项目, 18-4

显示 Check RAM 屏幕, 18-2

- Q -

切线, 画, 8-8

清除

错误原因, 1-2

图形, 8-4

项, 1-18

- R -

日期期间的天数公式, A-60

- S -

三角函数数学运算, 2-3  
设置

拆分屏幕方式, 9-3

程序中的方式, 1-9

从程序设置图象样式, 3-10

从主屏幕或程序设置拆分屏幕方式,  
9-6

从主屏幕或程序设置运算表, 7-3

显示对比度。参阅调节显示对比度  
设置方式, 1-9-12

输入

表达式和指令, 1-6

方程求解器中的表达式, 2-8, 2-9

复数, 2-16, 2-17

计算式, 6-8

科学记数法表示的数, 1-7

求解器编辑器中的变量值, 2-9

输入显示光标, 1-5

数据采集

开始, 14-25

停止, 14-26

数据采集方式, 14-18

数学运算, 键盘, 2-3, 2-4

数值序列, 和与积, 11-18

数组

LDCMT, 14-23

LDIST, 14-20, 14-23

LDREF, 14-20

LLCMT, 14-23

LLIGHT, 14-20, 14-23

LLREF, 14-20

LTCMT, 14-23

LTEMP, 14-20, 14-23

LTREF, 14-20

LVCMT, 14-23

LVOLT, 14-20, 14-23

LVREF, 14-20

从内存删除, 11-4

从数组中拆离公式, 11-8

从图中选择数据点, 11-13, 11-14

存储和显示, 11-4

访问元素, 11-4

画一组曲线, 11-4

将公式与数组名相连, 11-7, 11-8

## - S (续) -

拷贝数组, 11-4  
 命名, 11-3  
 输入名称, 11-6  
 用于数学运算, 2-3  
 在表达式中使用, 11-9  
 在数学函数中使用, 11-9  
 双样本 F 检验公式, A-53  
 双样本  $t$  统计公式, A-54  
 水平和图象 — 运算表方式下的象素, 9-6  
 缩放操作  
   序列图象, 6-10  
 缩放光标, 3-20  
 缩放内存, 主屏幕或程序, 3-24  
 缩放因子, 3-24  
 缩放运算  
   参数图象, 4-8  
   极坐标图象, 5-6

## - T -

条形图 (GAUGE), 14-19  
 调节显示对比度, 1-3  
 统计变量, 12-29  
 统计分布函数。参阅分布函数  
 统计分析  
   建立, 12-10  
   数据点, 出现频率, 12-24  
   在程序中进行统计计算, 12-30  
   在程序中使用统计数据, 12-30  
 统计绘图, 12-31  
   **Boxplot** (标准框图), 12-33  
   **Histogram**, 12-32  
   **ModBoxplot** (改进式框图), 12-32  
   **NormProbPlot** (正态概率曲线), 12-33  
   **Scatter**, 12-31  
   **xyLine**, 12-31  
   定义, 12-34  
   定义观察窗口, 12-36  
   跟踪统计图, 12-36  
   开启和关闭统计图, 12-35  
   数组中的数据, 步骤, 12-31  
   显示其他统计图编辑器, 12-35  
   在程序中定义, 12-37, 12-38  
   在程序中显示, 12-38  
 统计数组编辑器  
   编辑数组元素, 12-13

## - T (续) -

编辑由公式生成的数组元素, 12-16  
 拆离数组名中的公式, 12-16  
 创建数组名, 12-12  
 环境, 观察和编辑元素, 12-18  
 环境, 观察和输入名称, 12-19  
 恢复数组名, 12-12  
 将公式与数组名相连, 12-14  
 切换环境, 12-17  
 清除数组元素, 12-12  
 删除数组, 12-12  
 生成公式的数组名, 12-15  
 输入数组名, 12-11  
 显示, 12-10  
 图象, 删除和重新调用, 8-18  
 图象数据库 (GDB)  
   重新调用, 8-20  
   定义, 8-19  
   删除, 8-20  
 推理统计学。参阅 STAT TESTS  
   **data** 或 **stats**, 选择, 13-7  
   变量, 计算方法, 13-28  
   合并选项, 选择, 13-8  
   忽略置信区间, 13-8  
   计算检验结果, 13-8  
   计算置信区间, 13-8  
   检验输出变量, 13-28  
   区间输出变量, 13-28  
   输入参数值, 13-7  
   输入描述, 13-26, 13-27  
   图象检验结果, 13-8  
   显示和使用, 13-6  
   择一假设, 选择, 13-7  
 退出菜单, 4

## - X -

显示  
 CATALOG 中的函数和指令, 15-6  
 变量值, 1-15  
 菜单, 4  
 程序中的数值和消息, 16-19  
 对比度。参阅调节显示对比度  
 光标, 类型, 1-5  
 函数图象, 3-15  
 类型, 1-4  
 使用 **Input** 指令的图象, 16-17  
 输入和结果, 1-4  
 运算表, 7-5

- X (续) -

主屏幕, 1-4  
现金流转公式, A-59  
线段, 绘制, 8-5  
相位图. 参阅 序列图象. 参阅 序列  
  图象  
序列图象  
  CALC (计算) 运算, 6-10  
  TI-83 Plus 和 TI-82, 6-15  
  web 图, 6-11  
  递归序列, 6-6  
  定义函数, 6-5  
  定义和显示, 6-3  
  非递归序列, 6-5  
  跟踪, 6-9  
  函数, 6-3  
  设置  
    窗口变量, 6-7  
    图象方式, 6-3  
    图象格式, 6-8  
    坐标轴格式, 6-8  
  缩放操作, 6-10  
  显示 Y= 编辑器, 6-4  
  显示图象, 6-8  
  相位图, 6-13, 6-14  
  选择和取消选择函数, 6-4  
  选择图象样式, 6-4  
  选择坐标轴组合, 6-8  
  研究自由移动光标, 6-9  
  用 web 图作为收敛性样例, 6-12  
选择  
  Vars 或 Y-Vars 菜单中的名称, 1-22  
  Y= 编辑器中的函数, 3-7  
  菜单项,  
  从 CATALOG 中选择字符串函数,  
  15-6  
  从图中选择数据点, 11-13, 11-14

- Y -

样例

  其他

    食肉动物 — 猎物模型, 序列图  
    象, 6-13  
    食肉动物 — 猎物模型, 序列图  
    象, 6-14  
    用 web 图举例说明收敛性, 6-12  
  入门  
    定义函数, 9

- Y (续) -

定义数值运算表, 10  
改变观察窗口, 12  
函数的根, 求解, 7-2  
画切线, 8-2  
极坐标玫瑰线, 图象, 5-2  
计算复利, 14-4  
抛掷硬币模型, 2-2  
求摆锤长度与周期, 12-2  
求计算的最大值, 16  
森林和数木, 序列图象, 6-2  
生成数组序列, 11-2  
使用二次公式, 6-8  
投资汽车, 14-3  
图形细化, 15  
显示和跟踪图形, 13, 14  
线性方程组, 求解, 10-2  
向另一单元发送变量, 19-2  
小球的轨迹, 参数图象, 4-2, 4-3  
研究单位圆, 9-2  
圆柱体积, 找到程序, 16-2, 16-3  
运算表细化, 11  
总体平均身高, 13-2, 13-3, 13-4,  
  13-5  
应用  
  参数方程, 17-12, 17-13  
  抵押贷款, 计算和绘图, 17-18  
  绘制不等式, 17-5  
  绘制单位圆和三角曲线, 17-10  
  绘制分段函数, 17-4  
  绘制蛛网吸引点, 17-8  
  计算未付的贷款余额, 14-11, 14-  
  12  
  求解非线性方程组, 17-6  
  求曲线间面积, 17-11  
  使用程序创建 Sierpinski 三角形,  
  17-7  
  使用程序估计系数, 17-9  
  使用框图比较实验结果, 17-2,  
  17-3  
  微积分基本原理, 17-14, 17-15  
  正多边形面积, 17-16, 17-17  
一行多个输入项, 1-6  
疑难解答, B-4  
以前输入  
  访问, 1-17  
  执行, 1-18  
应用. 参阅 样例, 应用

- Y (续) -

运算表

从主屏幕或程序清除, 7-6

显示, 7-5, 7-6

运算表变量

定义, 7-3

定义和编辑因变量, 7-4

- Z -

在方程求解器中求解变量, 2-10, 2-11

在图象上放置文本, 8-12

遮蔽分布指令

**Shade $\chi^2$** (, 13-36

**ShadeF**(, 13-36

**Shade\_t**(, 13-36

**ShadeNorm**(, 13-35, 13-36

遮蔽图象区域, 8-10

直角坐标, 复数, 2-17

指令, 定义, 1-7

中断计算, 1-7

主屏幕, 返回, 1-4

注释

带文本的图象, 8-12

使用 **Pen** 指令的图象, 8-13

转置 (T) 矩阵函数, 10-12

字符串

**EquString**(, 15-7

**expr**(, 15-7

**inString**(, 15-7

**length**(, 15-8

**StringEqu**(, 15-8

**sub**(, 15-8

**VARS STRING** 菜单, 15-4

保存和显示变量, 15-5

变量, 15-4

定义, 15-3

输入, 15-3

显示, 15-6

最大值 (GAUGE), 14-19

最后输入, 1-17, 1-18

最小值 (GAUGE), 14-19