

HP 35s 科学计算器

用户手册



i n v e n t

第一版

HP 部件编号 F2215AA-90011

注意事项

请登陆 www.register.hp.com 注册您的产品

本手册及所有示例提供的是当前的情况，如有更改，恕不另行通知。本公司对于本手册不做任何担保，包括但不限于对于适销性，用作特定用途的适用性及不侵权性的默认担保。

本公司对于因本手册的提供、性能或使用以及所含的示例而导致的相关事故错误或因之引起的损害不承担任何责任。

© 惠普公司 1988, 1990-1991, 2003, 2007, 版权所有。未经本公司事先书面许可，不得复制、改编或翻译本手册，除非是在版权法的许可范围内。

Hewlett-Packard Company
16399 West Bernardo Drive
MS 8-600
San Diego, CA 92127-1899
USA

出版历史

第一版

2007 年 2 月

目录

第一部分 基本操作

1. 开始	1-1
重要初步信息	1-1
打开及关闭计算器	1-1
显示器对比度	1-1
重要键盘及显示器信息	1-2
换档键	1-2
字母键	1-3
光标键	1-3
后退及清除	1-4
使用菜单	1-6
退出菜单	1-8
RPN 及 ALG 模式	1-9
取消键	1-11
显示屏及信号器	1-12
键入数字	1-15
使数字变为负数	1-15
10 的指数	1-15
认识输入光标	1-17
数字范围及溢出	1-17
进行算术计算	1-18
单个自变量或二进制运算	1-18
两个自变量或二进制运算	1-19
控制显示模式	1-21
数字中的句号及逗号 (.) (,)	1-23

复数显示格式 (x^y , $x+y$, $r\cdot$)	1-24
显示全 12 位精度	1-25
分数	1-26
输入分数	1-26
信息	1-27
计算器内存	1-28
检查可用内存	1-28
清除所有内存	1-29
2. RPN: 自动内存栈	2-1
什么是栈	2-1
X 和 Y 寄存器都在显示屏上	2-3
清除 X 寄存器	2-3
查阅栈	2-3
交换栈中的 X 和 Y 寄存器	2-4
算术 – 栈是如何进行的	2-5
ENTER 是如何工作的	2-6
如何清除栈	2-7
LAST X 寄存器	2-8
纠正带 LAST X 的错误	2-9
重新使用带 LAST X 的数字	2-10
RPN 模式下链计算	2-12
从括号表示由内向外解题	2-12
练习	2-14
计算顺序	2-14
更多练习	2-16
3. 将数据存入变量	3-1
保存并恢复数字	3-2
查看变量	3-4

使用内存目录	3-4
变量目录	3-4
带保存变量的算术	3-6
保存算术	3-6
恢复算术	3-7
用 x 与任何变量交换	3-8
变量 “I” 和 “J”	3-9
4. 真数函数	4-1
指数函数和对数函数	4-1
除法运算的商和余数	4-2
幂函数	4-2
三角学	4-3
输入 p	4-3
设置角模式	4-4
三角函数	4-4
双曲函数	4-6
百分比函数	4-6
物理常量	4-8
转换函数	4-10
矩形 / 极坐标转换	4-10
时间转换	4-13
角度转换	4-13
单位转换	4-14
概率函数	4-15
因素	4-15
伽玛	4-15
概率	4-15
数字部分	4-17

5. 分数	5-1
输入分数	5-1
显示屏中的分数	5-2
显示规则	5-2
精度指示器	5-3
修改分数显示	5-4
设置最大分母	5-4
选择分数格式	5-6
分数显示举例	5-8
对分数进行四舍五入	5-8
等式中的分数	5-9
程序中的分数	5-10
6. 输入并评估方程	6-1
方程的使用方法	6-1
方程运算概述	6-3
将方程输入方程列表	6-4
方程中的变量	6-4
方程中的数字	6-5
方程中的函数	6-5
方程中的括号表示	6-6
显示并选择方程	6-6
编辑 / 清除方程	6-8
方程类型	6-9
评估方程	6-10
用 ENTER 键进行评估	6-11
使用 XEQ 进行评估	6-12
响应方程提示	6-13
方程的语法	6-14

运算符优先级.....	6-14
方程函数.....	6-16
语法错误.....	6-19
对方程进行验证.....	6-19
7. 解方程.....	7-1
求根.....	7-1
解内置方程.....	7-6
了解并管理 SOLVE.....	7-7
验证结果.....	7-7
终止 SOLVE 计算.....	7-8
选择 SOLVE 的初始猜测值.....	7-8
欲知详情.....	7-12
8. 对方程进行积分.....	8-1
对方程进行积分 (Ú FN).....	8-2
积分精度.....	8-6
指定精度.....	8-6
解析精度.....	8-6
欲知详情.....	8-8
9. 带复数的运算.....	9-1
复栈.....	9-2
复运算.....	9-2
使用极性计数法中的复数.....	9-5
方程中的复数.....	9-7
程序中的复数.....	9-8
10. 矢量算术.....	10-1
矢量运算.....	10-1
矢量的绝对值.....	10-3

点积	10-4
两矢量夹角	10-5
方程中的矢量	10-6
程序中的矢量	10-7
从变量或寄存器创造矢量	10-8
11.基底转换、算术运算及逻辑	11-1
基底 2、8 和 16 中的算术运算	11-4
数字的表示	11-6
负数	11-6
数字范围	11-7
二进制数字窗口	11-8
使用程序和方程中的基底	11-8
12.统计运算	12-1
输入统计数据	12-1
输入单变量数据	12-2
输入双变量数据	12-2
纠正数据输入中的错误	12-2
统计计算	12-4
平均值	12-4
样本标准差	12-6
总体标准差	12-7
线性回归	12-7
数据精确度的限制	12-10
总和数值与统计寄存器	12-11
总和统计	12-11
访问统计寄存器	12-12

第二部分 编程

13.简单编程	13-1
设计程序	13-3
选择模式	13-3
程序边界 (LBL 和 RTN)	13-4
在程序中使用 RPN, ALG 及方程	13-4
数据输入及输出	13-5
输入程序	13-6
清除函数及后退键	13-7
程序中的函数名称	13-8
运行程序	13-10
执行程序 (XEQ)	13-10
程序检测	13-11
输入并显示数据	13-12
使用 INPUT 输入数据	13-13
用 VIEW 显示数据	13-15
使用方程显示信息	13-16
显示信息而不停止程序	13-18
停止或终止程序	13-19
对停止或暂停 (STOP, PSE) 进行编程	13-19
终止正在运行的程序	13-19
错误停止	13-19
对程序进行编辑	13-20
程序内存	13-21
查看程序内存	13-21
内存使用	13-22
程序目录 (MEM)	13-22

清除一个或多个程序	13-23
校验	13-23
非可编程函数	13-24
用 BASE 进行编程	13-25
在程序中选择基底模式	13-25
程序行中输入的数字	13-25
多项式表达式及霍纳法	13-26
14.编程方法	14-1
程序中的子程序	14-1
调用子程序 (XEQ, RTN)	14-1
迭套子程序	14-2
分路 (GTO)	14-4
编程 GTO 指令	14-5
使用键盘上的 GTO 指令	14-5
条件指令	14-6
比较测试 ($x \neq y$, $x \neq 0$)	14-7
标志	14-9
回路	14-16
条件回路 (GTO)	14-17
带计数器的回路 (DSE, ISG)	14-18
间接定址变量和标签	14-20
变量 "I" 和 "J"	14-20
间接地址 (I) 和 (J)	14-21
带 (I)/(J) 的程序控制	14-23
带 (I)/(J) 的方程	14-23
未命名间接变量	14-23
15.对程序进行求解和积分	15-1
对程序进行求解	15-1

使用程序中的 SOLVE	15-6
对程序进行积分	15-7
使用程序中的积分	15-10
求解及积分限制	15-11
16.统计程序	16-1
拟合曲线	16-1
正态及逆 / 正态分布	16-11
分组标准差	16-18
17.其余程序及方程	17-1
金钱的时间价值	17-1
素数发生器	17-7
矢量中的叉积	17-11

第三部分 附录及参考文献

A. 支持、电池及服务	A-1
计算器支持	A-1
常见问题答案	A-1
环境限值	A-2
更换电池	A-3
测试计算器运行	A-4
自检	A-5
保修	A-7
客户支援	A-8
监管信息	A-12
联邦通讯委员会通告	A-12
B. 用户内存及栈	B-1
管理计算器内存	B-1

重置计算器	B-2
清除内存	B-3
栈提升状态	B-4
禁用操作	B-5
中性操作	B-5
LAST X 寄存器状态	B-6
访问栈寄存器内容	B-7
C. ALG: 概述	C-1
关于 ALG	C-1
在 ALG 模式下进行两自变量算术运算	C-2
简单算术运算	C-2
幂函数	C-3
百分比计算	C-3
排列 / 组合	C-4
除法运算的商和余数	C-4
括弧计算	C-4
指数函数和对数函数	C-5
三角函数	C-6
双曲函数	C-6
数字部分	C-7
查阅栈	C-7
对方程进行积分	C-8
带复数的运算	C-8
基底 2、8 和 16 中的算术运算	C-10
输入统计双变量数据	C-11
D. 有关求解的更多信息	D-1
SOLVE 是如何求根的	D-1
解分结果	D-3

当 SOLVE 无法求出根时	D-8
四舍五入错误	D-13
E. 关于积分的更多信息	E-1
积分是如何评估的	E-1
可导致错误结果的条件	E-2
延长计算时间的条件	E-7
F. 信息	F-1
G. 运算索引	G-1
H. 索引	H-1

第一部分

基本操作

开始



请注意页边的这一符号。该符号标明了 **RPN** 模式下的示例或键盘按键，这些示例或键盘按键必须在 **ALG** 模式下以不同方式进行。

附录 C 说明了在 **ALG** 模式下使用您的计算器的方法。

重要初步信息

打开及关闭计算器

要想打开计算器，按 。ON 印在  键的底部上。

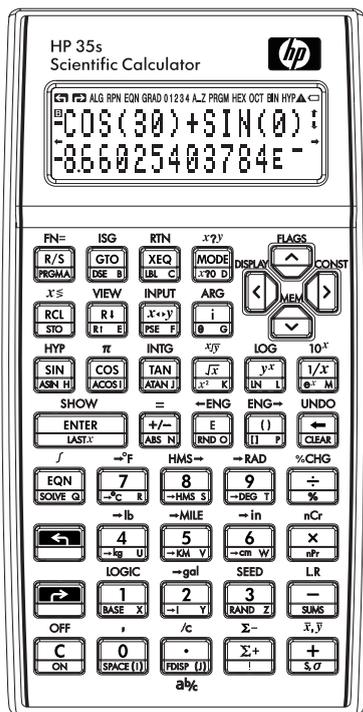
要想关闭计算器，按  。即先按下并释放  换挡键，然后按  键 (OFF 以红色印在其下)。由于本计算器具备连续记忆功能，将其关闭并不会影响任何您已经保存的信息。

为达到节能的效果，本计算器在经过 10 分钟不使用后会自动关闭。如果您在显示器上看到电量低指示 ()，请尽快更换电池。相关说明，请参阅附录 A。

显示器对比度

显示器对比度取决于以下因素：照明、视角以及对比度设置。要想提高或降低对比度，保持  键并按下  或  键。

重要键盘及显示器信息



换档键

每个键有 3 种功能，一个印在其表面，一个右移功能 (黄色) 及一个左移功能 (蓝色)。各换档键的功能名称分别以黄色及蓝色印在各键的上部及底部。按下某键以实现预期的功能前，请按下相应的换档键 (或)。例如，要想关闭计算器，按下并释放 换档键，然后按下 **C** 键。

按下  或  将打开显示器上方的相应  或  信号器符号。在您按下一个键之前，信号器将恒亮。要想取消某个换档键(并关闭信号器)，再次按下同一换档键。

字母键



如上所示，大多数键在其右下角显示一个字母。无论何时，当您需要输入某个字母(例如，一个变量或程序标签)时，显示器上即出现 **A..Z** 信号器，显示字母键以激活。

变量在第 3 章中讲述；标签也在第 13 中讲述。

光标键

4 个光标方向键中的任何一个都标有箭头。在本手册中，我们将用图形 , ,  及  来表示这些光标键。

后退及清除

您首先需要知道的是如何清除输入项，纠正数字以及将整个显示清除以便从头开始。

清除键

键	说明
	<p><i>后退。</i></p> <p>如果某项表达式正处在被输入的过程中，清除输入光标 (<i>_</i>) 左边的字符。否则，对于第 2 行中完成的表达式或计算结果，将以 0 取代该结果。另外，还将清除错误信息并退出菜单价。当计算器处于程序输入及方程输入模式时，的性能内似。</p> <ul style="list-style-type: none">■ 方程输入模式： 如果某项方程正处在被输入或编辑的过程中，清除仅靠插入光标左边的字符。如果该方程已输入 (无插入光标出现)，将清除整个方程。■ 程序输入模式： 如果某个程序行正处在被输入或编辑的过程中，将清除插入光标左边的字符；否则，如果该程序行已输入，将清除整个程序行。
	<p><i>清除或取消。</i></p> <p>将显示的数字清除为 0 或取消当前状态 (如菜单、提示、目录、方程输入或程序输入模式)。</p>

清除键 (续)

键	说明
 CLEAR	<p>清除菜单 (\times VARS ALL Σ STK CLVAR\times) 包含清除 x(X 寄存器中的数字)、所有直接变量、所有内存、所有统计数据、栈及间接变量的选项。</p> <p>如果您按 [3](3ALL) 键, 将显示一个新菜单 CLR ALL? Y N, 使您在清除存储器中的任何信息之前能够确认您的决定。</p> <p>在程序输入过程中, 3ALL 被 3PGM 所取代。如果您按 [3](3PGM), 将显示一个新菜单 CLR PGMS? Y N, 使您在清除所有的程序之前能够确认您的决定。</p> <p>在决定输入过程中, 3ALL 被 3EQN 所取代。如果您按 [3](3EQN) 键, 将显示 CLR EQN? Y N 菜单, 使您在清除所有的方程之前能够确认您的决定。</p> <p>当您选择 [6](CLVAR\times) 时, 该指令即粘贴至带 3 个占位符的命令。您必须在占位符空位中输入一个 3 位数。然后, 所有位址大于已输入位址的间接变量将被清除。例如, CLVA056 清除所有位址大于 56 的间接变量。</p>

使用菜单

HP 35s 的实际功能比您在键盘上看到的要多得多。这是因为键盘键中有 16 个是菜单键。总共有 16 个菜单，提供更多功能或更多功能的更多选项。

HP 35s 菜单

菜单名	菜单说明	章
	数字功能	
L.R.	$\hat{x} \hat{y} r m b$ 线性回归：曲线拟合及线性估计。	12
\bar{x}, \bar{y}	$\bar{x} \bar{y} \bar{x}w$ 统计 x 及 y 值的算术平均值；统计 x 值的加权平均值。	12
s, σ	$sx sy \sigma x \sigma y$ 样本标准差，总体标准差。	12
CONST	41 个物理常量值的访问菜单 — 参阅第 4–8 页“物理常量”。	4
SUMS	$n \Sigma x \Sigma y \Sigma x^2 \Sigma y^2 \Sigma xy$ 统计数据总和。	12
BASE	DEC HEX OCT BIN $d h o b$ 基底转换 (10 进制、16 进制、8 进制及 2 进制)。	12
INTG	SGN INT \div Rmd r INTG FP IP 符号值、整数除法、除法余数、最大整数、分数部分及整数部分。	4,C
LOGIC	AND XOR OR NOT NAND NOR 逻辑运算符	11

编程说明		
FLAGS	SF CF FS? 功能设置、清除及测试标志。	14
x?y	$\neq \leq < > \geq =$ X 及 Y 寄存器比较测试。	14
x?0	$\neq \leq < > \geq =$ X 寄存器及 0 比较测试。	14
其它功能		
MEM	VAR PGM 内存状态 (可用内存的字节数); 变量目录; 程序目录 (程序标签)。	1, 3, 12
MODE	DEG RAD GRAD ALG RPN 角模式及运算模式	4, 1
DISPLAY	FIX SCI ENG ALL. , 1,000 1000 $x \cdot i \cdot y$ $x + y \cdot i \cdot r \theta a$ 固定、科学、工程、全浮点数字显示格式; 基数符号选项 (或); 复数显示格式 (RPN 模式, 仅有 $x \cdot i \cdot y$ 及 $r \theta a$)	1
R↓ R↑	X Y Z T 查阅 ALG 模式中栈的功能 -X-、Y-、Z-、及 T- 寄存器	C
CLEAR	清除内存不同部分功能 — 参阅第 1-5 页表中的  CLEAR。	1, 3, 6, 12

使用菜单功能:

1. 按下菜单键显示一组菜单项目。
2. 按     将下划线移至您想要选择的项目。
3. 该项目被加上下划线时请按  键。

借由编号菜单项, 您可以在该项目被加上下划线时按下  或者只输入该项目的编号。

有些菜单，如常数及总和不只一页。输入这些菜单将打开 **▲** 或 **▼** 信号器。在这些菜单中，使用 **▶** 及 **◀** 光标键导航至当前菜单页上的一个项目；使用 **▼** 及 **▲** 键访问菜单中的前页及后页。

示例：

在本例中，我们使用显示菜单将数字的显示固定至 4 个小数位，然后计算 $6 \div 7$ 。用显示菜单返回至数字的全浮点显示结束本例。

键：	显示：	说明：
	0 0	初始显示
◀ DISPLAY	1FIX 2SCI 3ENG 4ALL	输入显示菜单
1 或 ENTER	FIX_	固定指令粘贴至第 2 行
4	0.0000 0.0000	固定至 4 个小数位
6 ENTER 7 ÷	0.0000 0.8571	进行除法运算
◀ DISPLAY 4	0 8.57142857143E-	返回至全精确度

各菜单可指引并帮助您执行多种功能。您既不需要记住计算器各种内置功能的名称，也无须搜索印在键盘上的各种功能。

退出菜单

无论何时，当您执行某项菜单功能时，菜单自动消失（如上例所示）。如果您想退出某项菜单而不执行一项功能，您可以有 3 种选择：

- 按  出现 2 级清除及内存菜单，一次一级。见第 1-5 页表中的  **CLEAR**。
- 按  或 **C** 取消任何其它菜单。

键:	显示:
1 2 3 . 5 6 7	123.5678_
8	
 DISPLAY	<u>1FIX</u> 2SCI ↓ 3ENG 4ALL
 或 C	123.5678_

- 按下其它任何菜单键用新菜单取代旧菜单。

键:	显示:
1 2 3 . 5 6 7	123.5678_
8	
 DISPLAY	<u>1FIX</u> 2SCI ↓ 3ENG 4ALL
 CLEAR	<u>1X</u> 2VARS ↓ 3ALL 4Σ
C	123.5678

RPN 及 ALG 模式

本计算器可设置为在 RPN(Reverse Polish Notation) 模式或 ALG(Algebraic) 模式下进行算术运算。

在逆波兰 (RPN) 模式下，计算的中间结果自动保存；因此，您无须使用标点符号。

在算术几何 (ALG) 模式下，您可按照标准运算顺序进行算术运算。

要想选择 **RPN** 模式：

按 **MODE** **5** (**5RPN**) 将计算器设置为 RPN 模式。当计算器处于 RPN 模式时，**RPN** 信号器亮起。

要想选择 **ALG** 模式：

按 **MODE** **4** (**4ALG**) 将计算器设置为 **ALG** 模式。当计算器处于 **ALG** 模式时，**ALG** 信号器亮起。

示例：

假设您想计算 $1+2=3$ 。

在 **RPN** 模式下，先输入第一个数字，然后按 **ENTER** 键，接着输入第二个数字，最后按下算术运算符键：**+**。

在 **ALG** 模式下，先输入第一个数字，然后按 **+** 键，接着输入第二个数字，最后按 **ENTER** 键。

RPN 模式	ALG 模式
1 ENTER 2 +	1 + 2 ENTER

在 **ALG** 模式下，结果及计算过程都显示。在 **RPN** 模式下，只显示结果而不显示计算过程。

注意



您既可以选择 **ALG** 模式，也可选择 **RPN** 模式进行计算。在整本手册中，页边上的“✓”显示 **RMN** 模式下的示例及键盘键在 **ALG** 模式下必须用不同的方法进行。附录 C 说明了 **ALG** 模式下您的计算器的使用方法。

取消键

取消键

取消键的操作取决于计算器上下文，但其作用主要是恢复某项删除的输入而非取消任何任意操作。有关在执行一项数字功能后恢复显示屏第 2 行的输入的详情，请参阅第 2 章 *Last X 寄存器*。在用  或  恢复后立即按  **UNDO**：

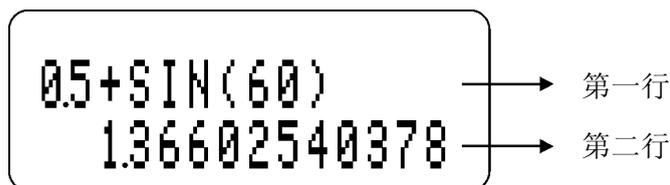
- 您已经删除的输入
- 方程模式下删除的方程
- 程序模式下删除的程序行

此外，您还可以用取消键恢复一个刚刚用清除菜单清除的寄存器数值。取消操作必须紧随删除操作进行；任何插入操作都将妨碍取消键恢复已经删除的对象。除了在删除后全部恢复输入项，取消键可在编辑输入项时使用。在编辑恢复以下项目时请按  **UNDO**：

- 您刚刚用  删除的表达式中的一个数字
- 一项您正在编辑但是被用  删除的表达式
- 您刚刚用  删除的方程或程序中的一个字符（在方程模式或程序模式下）

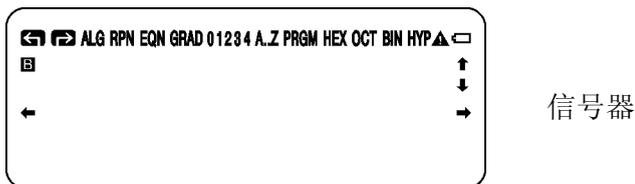
另外请注意，取消操作受可用内存的限制。

显示屏及信号器



显示屏由两行及信号器组成。

超过 14 个字符的输入项将滚动至左侧。在输入过程中，输入项在 ALG 模式下显示在第一行，在 RPN 模式下则显示在第二行。每一项计算都以多达 14 位数显示，包括 E 符号 (指数) 以及多达 3 位数的指数值。



显示屏上的符号，如上图所示，称为信号器。在显示屏上出现时，每一个符号都有特殊的意义。

HP 35s 信号器

信号器	意义	章
B	“ B (忙)”信号器在执行运算、方程及程序时出现。	
▲ ▼	在分数显示模式下(按  FDISP)，只有其中一个“ ▲ ”或“ ▼ ”信号的“ ▲▼ ”半会被打开，显示所示分子是否稍小于或大于其真实值。如果“ ▲▼ ”无任何部分亮起，将显示该分子的真实值。	5
	左移处于激活状态。	1
	右移处于激活状态。	1
RPN	逆波兰模式处于激活状态。	1, 2
ALG	算术几何模式处于激活状态。	1, C
PRGM	程序输入处于激活状态。	13
EQN	方程输入处于激活状态或计算器正在评估某个表达式或执行某个方程。	6
0 1 2 3 4	显示哪些标志已设置(标志 5 至 11 无信号器)。	14
RAD 或 GRAD	Radians 或 Grad 角模式已设置。DEG 模式(默认)无信号器。	4
HEX OCT BIN	显示有效基数。DEC(基数 10, 默认)无信号器。	11
HYP	反双曲函数处于激活状态。	4, C

HP 35s 信号器 (续)

信号器	意义	章
←, →	<p>第一行或第二行输入项的显示屏左边或右边的字符更多。所有这些信号器会同时出现, 表示某项输入显示屏的左边或右边有字符。第一行中字符有遗漏的输入项将显示显示一个省略符, 表示字符有遗漏。在 RPN 模式下, 使用  及  键滚动至某个输入项并查看首位字符。在 ALG 模式下, 使用   及   查看其余字符。</p>	1, 6
↑, ↓	<p> 及  键处于激活状态, 可用于查看方程列表, 变量目录, 程序行, 菜单页或程序目录中的程序。</p>	1, 6, 13
A..Z	<p>字母键处于激活状态。</p>	3
▲	<p>注意! 表示特殊情况或错误。</p>	1
	<p>电池电量低。</p>	A

键入数字

本计算器能够处理的最小及最大数值为 ± 9.999999999999999999 ⁴⁹⁹。如果计算结果超出此范围，故障信息“OVERFLOW”立即与  信号器一同出现。然后，溢出信息被与计算器能够显示的溢出边界最接近的数值所取代。计算器能够与 0 区分的最小数字为 $\pm 10^{-499}$ 。如果您输入这些数值之间的某个数字，输入时计算器将显示 0。同样地，如果计算的结果也在这些数值之间，结果将显示为 0。输入超出最大范围的数值将导致故障信息“INVALID DATA”，清除该故障信息将使您返回至先前输入项以进行纠正。

使数字变为负数

✓  键改变数字的符号。

- 要想输入一个负数，先输入该数字，然后按 。
- 在 ALG 模式，您可以在输入数字之前或之后按  键。
- 要想改变之前输入的数字的符号，按  即可（如果该数字有一个指数， 只影响尾数 — 该数字的非指数部分）。

10 的指数

显示屏上的指数

明显带有 10 的幂的数字（如 4.2×10^{-5} ）显示为字母 **E** 前加上 10 的指数。因此， 4.2×10^{-5} 输入后显示为 **4.2E-5**。

数值对于显示格式而言过大或过小的数字将自动以指数的形式显示。

例如，在 4 个小数位的 FIX 格式中，观察以下键盘键的效果：

键:	显示:	说明:
0 . 0 0	0.000062_	显示输入的数字。
0 0 6 2		
ENTER	0.0001	对数字进行四舍五入，使之适应显示格式。
0 . 0 0	4.2000E-5	自动使用科学记数法，否则，将不会出现有效数位。
0 0 4 2		
ENTER		

键入 10 的指数

E 键用于快速输入 10 的指数。例如，非但无需将 1 百万输入为 1000000，您只需要输入 **1** **E** **6** 即可。下例说明了输入过程以及计算器是如何显示结果的。

示例:

假设您想输入普朗克常数： 6.6261×10^{-34}

键:	显示:	说明:
6 . 6 2 6	0	输入尾数
1	6.6261_	
E	0	等同 $\times 10^x$
	6.621E	
3 4 +/- ENTER	6.621E ⁻³⁴	输入指数
	6.621E-34	

对于无倍数的 10 的幂，如上例中的 1 百万，按下 **1** **E** 键出现所需 10 的指数。

指数函数

要想计算 10 的指数 (基数 10 反对数), 使用 $\boxed{10^x}$ 。计算任何数字升幂 (取幂) 后的结果, 使用 $\boxed{y^x}$ (参阅第 4 章)。

认识输入光标

当您键入某个数字时, 光标 ($_$) 出现并在显示屏上闪烁。光标显示下一个数位的位置, 从而表明该数字不完整。

键:	显示:	说明:
$\boxed{1} \boxed{2} \boxed{3}$	123 $_$	输入未终止; 数字不完整。

如果您执行一项函数来计算结果, 光标消失, 原因是该数字完整 - 输入已终止。

✓ $\boxed{\sqrt{x}}$ 11.0905 输入终止。

按 $\boxed{\text{ENTER}}$ 终止输入。要想把两个数字分开, 键入第一个数字, 按 $\boxed{\text{ENTER}}$ 终止输入, 然后键入第二个数字

✓ $\boxed{1} \boxed{2} \boxed{3} \boxed{\text{ENTER}}$ 123.0000 一个完成的数字。

✓ $\boxed{4} \boxed{+}$ 127.0000 另外一个完成的数字。

如果输入未终止 (如有光标出现), $\boxed{\leftarrow}$ 后退清除最后一位数。如果输入终止 (无光标), $\boxed{\leftarrow}$ 发挥与 \boxed{C} 一样的作用清除整个数字。试试吧!

数字范围及溢出

计算器上可能的最小数为 $-9.9999999999 \times 10^{499}$, 而最大数则为 $9.9999999999 \times 10^{499}$ 。

- 如果计算得出的结果超过可能的最大数, 则返回 $-9.9999999999 \times 10^{499}$ 或 $9.9999999999 \times 10^{499}$ 并出现告警信息 **OVERFLOW**。

进行算术计算

HP 35s 可在 RPN 或 ALG 模式下运行。这些模式会影响表达式的输入方法。以下各节说明了单个自变量 (或一进制) 及两个自变量 (或二进制) 运算的输入差异。

单个自变量或二进制运算

HP 35s 科学计算器的有些数字运算要求单个数字输入, 如 $\boxed{1/x}$, $\boxed{x^2}$, $\boxed{\text{LN}}$ 及 $\boxed{\text{SIN}}$ 。取决于计算器是在 RPN 模式下或是 ALG 模式下, 这些单个自变量运算的输入方法也不同。在 RPN 模式下, 先输入数字, 然后进行运算。如果在输入数字后按下 $\boxed{\text{ENTER}}$ 键, 第一行上出现该数字, 第二行上显示结果。否则, 将只会在第二行上显示结果而第一行则保持不变。在 ALG 模式下, 先按下运算符, 然后显示屏上显示函数, 仅随其后就是一组省略号。先在省略号之间输入数字, 然后按下 $\boxed{\text{ENTER}}$ 键。表达式在第一行显示, 而结果则在第二行显示。以下例子说明了这些差异。

示例:

先在 RPN 模式下然后在 ALG 模式下计算 3.4^2 。

键:	显示:	说明:
MODE 5 (5RPN)		进入 PRN 模式 (必要时)
3 . 4	0 3.4	输入数字
x²	0 11.56	按下平方运算符
MODE 4 (4ALG)		切换至 ALG 模式
x²	SQ()	进入平方运算
3 . 4	SQ(3.4)	插入省略号之间的数字
ENTER	SQ(3.4) 11.56	按 ENTER 键查看结果

例中,平方运算符在键上表示为 **x²**,但显示为 SQ()。有多个单个自变量运算符在 ALG 模式下的显示与在键盘上出现不同(也与在 RPN 模式下出现不同)。这些运算如下表所列。

键	在 RPN 及 RPN 程序中	在 ALG, 方程及 ALG 程序中
x²	x^2	SQ()
√x	\sqrt{x}	SQRT()
e^x	e^x	EXP()
10^x	10^x	ALOG()
1/x	$1/x$	INV()

两个自变量或二进制运算

虽然其差异与单个自变量运算符的情况类似,两个自变量运算,如 **+**, **÷**, **y^x** 及 **nCr** 依据不同的模式,以不同的方式输入。在 RRN 模式下,先输入第一个数字,然后把第二个数字放入 x 寄存器中并调用两个自变量运算。在 ALG 模式下,有两种情况,一种使用传统的中缀符,另外一种使用一种更为面向函数的方法。以下例子说明了这些差异。

示例

先在 RPN 模式下，然后在 ALG 模式下计算 $2+3$ 及 $6C_4$ 。

键:	显示:	说明:
MODE 5 (5RPN) 2 ENTER 3	2 3_	切换至 RPN 模式 (必要时) 输入 2, 把 3 放如 x 寄存器。注意 3 后的光标闪烁; 不要按 ENTER 键!
+	0 5	按加法键查看结果。
6 ENTER 4	6 4_	输入 6, 然后把 4 放入 x 寄存器。
↵ nCr	5 15	按组合键查看结果。
MODE 4 (4ALG) 2 + 3 ENTER	2+3	切换至 ALG 模式 表达式及结果均显示。
↵ nCr	5	输入组合函数。
6 > 4	nCr(6,4)	输入 6, 然后移动编辑光标经过逗号, 最后输入 4。
ENTER	nCr(6,4) 15	按 ENTER 键查看结果。

在 ALG 模式下, 中缀运算符为 **+**, **-**, **×**, **÷** 及 **y^x**。另外两个自变量使用 f 形式 (x,y) 的函数符号, 其中 x 及 y 依次为第一及第二运算数。在 RPN 模式下, 两个自变量运算的运算数输入在栈上的 Y 及 X 中。即 y 是 y 寄存器中的数值, x 是 x 寄存器中的数值。

$y(\sqrt[y]{x})$ x 次根是本规则的一个例外。例如, 若想以 RPN 模式计算 $\sqrt[3]{8}$, 按 **8** **ENTER** **3** **↵** $\sqrt[y]{x}$ 。在 ALG 模式下, 等值运算作为 **↵** $\sqrt[y]{x}$ **3** **>** **8** **ENTER** 键入。

对于单个自变量运算, 其中一些两个自变量运算在 RPN 模式下的显示与在 ALG 模式下不同。下表对这些差异进行了总结。

键	在 RPN 及 RPN 程序中	在 ALG, 方程及 ALG 程序中
y^x	y^x	\wedge
$\sqrt[y]{x}$	$x \sqrt{y}$	XROOT(,)
INT \div	INT \div	IDIV(,)

对于交换运算, 如 $\boxed{+}$ 及 $\boxed{\times}$, 运算数的顺序不影响计算结果。如果您在 RPN 模式下以错误顺序错误输入运算数用于非交换 2 个自变量运算, 只需按下 $\boxed{x \leftrightarrow y}$ 键交换 x 和 y 寄存器中的内容即可。这一点在第 2 章中有详细说明 (见 "交换栈中 X 和 Y 寄存器" 一节)。

控制显示模式

所有的数字都以 12 位精度保存, 但是您可以通过显示菜单中的选项控制数字显示过程中所用数位的数量。按 $\boxed{\leftarrow}$ $\boxed{\text{DISPLAY}}$ 访问本菜单。前 4 个选项 (FIX, SCI, ENG 及 ALL) 控制数字显示过程中数位的数量。在某些复杂的内部计算过程中, 计算器将 15 位精度用于中间结果。显示的数字按照显示模式进行四舍五入。

固定小数格式 (FIX)

如果符合的话, FIX 格式显示小数位多达 11 位的数字 (11 位数在 "." 或 ",'" 基数标记的右边)。在提示 $\text{FIX}_$ 后, 键入要显示的小数位的数量。对于 10 个或 11 个小数位, 按 $\boxed{\cdot} \boxed{0}$ 或 $\boxed{\cdot} \boxed{1}$ 。

例如, 在数字 123,456.7089 中, "7"、"0"、"8" 及 "9" 为计算器设置为 FIX4 显示模式时您看到的小数位。

任何因太大 (10^{11}) 或太小 (10^{-11}) 而不能以当前小数位设置显示的数字都将自动以科学格式显示。

科学格式 (SCI)

SCI格式以科学计数法显示数字("·"或" ," 基数标记前一位数), 其指数中则包含多达 11 个小数位及 3 个有效数位。在提示 **SCI_** 后, 键入要显示的小数位数字。对于 10 个或 11 个小数位, 按  **0** 或  **1**。(数字的尾数部分将始终小于 10。)

例如, 在数字 **1.2346E5** 中, "2"、"3"、"4" 及 "6" 为计算器设置为 SCI 4 显示模式时您看到的小数位。字母 E 后的 "5" 为 10 的指数: 1.2346×10^5 。

如果您输入或计算的数字超过 12 位数, 额外的精度无法保证。

工程格式 (ENG)

ENG格式以一种与科学计数法类似的方式显示数字, 只是其指数为 3 的倍数("·"或" ," 基数标记前可以有多达 3 位数)。本格式对于指定使用 10^3 的倍数为单位的科学及工程计算 (如微-, 毫- 及千单位) 特别有用。

在提示及 **ENG_** 后, 在第一个有效数位后输入您希望的数位数字。对于 10 个或 11 个小数位, 按  **0** 或  **1**。

例如, 在数字 **123.46E3** 中, 当计算器设置为 ENG 4 显示模式时, "2", "3", "4" 及 "6" 为您看到的第一个有效数字后的有效数字。"E" 后的 3 则为 10 的指数 (3 的倍数): 123.46×10^3 。

按  **←ENG** 或  **ENG→** 将使被显示数字的指数显示以 3 的倍数改变, 而尾数则相应进行调整。

示例:

下例不但说明了使用数字 12.346E4 的工程格式的特性, 还显示了   ENG 及  ENG  函数的使用。本例使用 RPN 模式。

键:	显示:	说明:
 DISPLAY 3  EN	ENG_	选择工程格式
C)		
4	0.0000E0 0.0000E0	输入 4(第 1 位数后 4 个有效数位)
1 2 . 3 4 6	123.46E3	输入 12.346E4
E 4 ENTER	123.46E3	
  ENG 或	123.46E3	
 ENG 	123.46E3	
  ENG	123.46E3	使指数增大 3
	0.12346E6	
 ENG 	123.46E3	使指数缩小 3
	123.46E3	

ALL 格式 (ALL)

ALL 格式为默认格式, 其数字显示精度高达 12 位。如果所有的位数与显示屏不相符, 数字将自动以科学格式显示。

数字中的句号及逗号 (.) (,)

HP 35s 使用句号及逗号使数字更易读。您可以选择句号或逗号作为小数点(基数)。此外, 您还可以选择是否用千位分隔符将数字分为 3 个一组。下例说明了这些选项。

示例

输入数字 12,345,678.90，将小数点改为逗号。然后选择无千位分隔符。最后，返回至默认设置。本例使用 RPN 模式。

键:	显示:	说明:
 DISPLAY 4 (4AL		选择全浮点精度 (ALL 格式)
L)		
1 2 3 4 5	12,345,678.9	默认格式使用逗号作为千位分隔符，句号作为基数。
6 7 8 . 9	12,345,678.9	
ENTER		
 DISPLAY 6 (6,)	12.345.678.9 12.345.678.9	改为将逗号用于基数。注意千位分隔符自动变为句号。
 DISPLAY 8 (810	12345678.9	该为无逗号分隔符。
00)	12345678.9	
 DISPLAY 5 (5.)	12,345,678.9	返回至默认显示格式。
 DISPLAY 7 (71,	12,345,678.9	
000)		

复数显示格式 ($x+iy$, $x+yi$, $r\theta a$)

复数可以多种格式显示，如 $x+iy$ ， $x+yi$ 及 $r\theta a$ ，尽管 $x+yi$ 只在 ALG 模式下可用。在下例中，复数 $3+4i$ 以全部 3 种格式显示。

示例

分别以不同的模式显示复数 $3+4i$ 。

键:	显示:	说明:
MODE 4 (4RLG)		启用 ALG 模式
3 i 4 ENTER	$3i.4$	输入复数，显示为 $3i4$ ，这是默认格式。
← DISPLAY .	$3i.4$	改为 $x+yi$ 格式。
1 ($11x+y.i$)	$3+4i.$	
← DISPLAY .	$3i.4$	改为 $r\theta a$ 格式。半径为 5，角度约为
0 ($1\theta r\theta a$) 或	$5\theta 53.13\theta 1\theta 23542$	53.13° 。
← DISPLAY ^		
^ > ENTER		

显示全 12 位精度

改变所显示小数位的数字会影响您看到的数字，但不影响数字的内部表示。保存在计算器内的任何数字都有 12 位数。

例如，在数字 14.8745632019 中，当显示模式设置为 FIX 4 时，您之会看到 "14.8746"，而最后 6 位数 ("632019") 则存在于计算器内部。

要想临时全精度显示一个数字，按 **←** **SHOW**。只要您按住 **SHOW** 不放，计算器将显示该数字的尾数(但无指数)。

键:	显示:	说明:
4 5 ENTER 1 .	58.5000	显示 4 个小数位。
3 x		
← DISPLAY 2 (2SCI)	$5.85E1$	科学格式: 2 个小数位, 1 一个指数。
2		
← DISPLAY 3 (3ENG)	$58.5E0$	工程格式。
2		
← DISPLAY 4 (4ALL)	58.5	所有有效数位; 尾 0 舍去。

 DISPLAY  (FIX)	58.5000	4 个小数位，无指数。
		
	0.0171	58.5 的倒数。
 SHOW (按住不放)	170940170940	显示全精度直到您释放 SHOW

分数

HP 35s 允许您输入分数，进行分数运算并将其作为小数或分数显示。HP 35s 以 $a \frac{b}{c}$ 的形式显示分数，式中 a 为整数， b 和 c 为自然数。除此之外， b 为 $0 \leq b < c$ ， c 为 $1 < c \leq 4095$ 。

输入分数

分数可以随时输入到栈上：

1. 键入数字的整数部分，按  (第一个  使数字的整数部分与分数部分分开)
2. 键入分数的分子，然后再次按下 。第二个  将分子与分母分开。
3. 先键入分母，然后按下 **ENTER** 或某个功能键终止数位输入。数字或结果按照当前显示格式进行格式化。

 键下的 $a \frac{b}{c}$ 提醒您  键两次用于分数输入。

下例说明了分数的输入及显示。

示例

输入混合数 $12\frac{3}{8}$ 并以分子及小数的形式对其进行显示。然后输入 $\frac{3}{4}$ 并加上 $12\frac{3}{8}$ 。本例使用 RPN 模式。

键:	显示:	说明:
1 2 . 3	0 12.3	小数点按通常方式进行解析。
. 8	0.0000 12 3/8 _	第二次按下 . 时，显示屏切换至分数模式。
ENTER	12.3750 12.3750	输入时，数字以当前显示格式显示。
↶ FDISP	12 3/8 12 3/8	切换至分数显示模式。
. 3 . 4	12 3/8 0 3/4 _	进入 $\frac{3}{4}$ 。注意，之所以从 . 开始是因为无整数部分 (您可以输入 $0\frac{3}{4}$)。
+	0 13 1/8	$\frac{3}{4}$ 加 $12\frac{3}{8}$ 。
↶ FDISP	0 13.1250	返回至当前显示模式。

有关分数使用的详情，请参阅第 5 章 -" 分数 "。

信息

计算器显示 **▲** 信号器对错误状况做出响应。通常情况下，伴随该错误信号器会出现一条消息。

- 要想清除一条消息，按 **C** 或 **←**；在 RPN 模式下，您将返回至出现错误前一样的栈。在 ALG 模式下，您将返回至最后一个表达式，而编辑光标则停留在错误处以对其进行纠正。

- 虽然未输入键函数，任何其它键也可清除信息。

如果无任何信息显示但是出现  信号器，则说明您按下的是一个失效或无效键。例如，按   将显示 ，原因是第二个小数点在此情况下无任何意义。所有的显示信息都在附录 F-" 信息 " 中说明。

计算器内存

在 HP 35s 30KB 的内存中，您可以保存任何数据组合 (变量、方程或程序行)。

检查可用内存

按   显示以下菜单：

```
1VAR  2 PGM  
nnn   mm,mmm
```

式中

nnn 为所用间接变量的数量。

mm,mmm 为可用内存的字节数。

按  (1VAR) 显示直接变量的目录 (参阅第 3 章 - “查看 VAR 目录中的变量”)。

按  (2PGM) 显示程序目录。

1. 要想进入变量目录，按  (1VAR)；要想进入程序目录，按  (2PGM)。
2. 要想查阅目录，按  或 。
3. 要想删除一个变量或程序，在目录中对其进行查看时按下  。
4. 要想退出目录，按 。

清除所有内存

清除所有内存将清除所有您保存的数字、方程及程序。清除内存不影响模式及格式设置 (要想清除设置及数据, 参阅附录 B-" 清除内存 "。)

要想清除所有内存:

1. 按下 **4** (**4ALL**), 您将看到确认提示 **CLR ALL? Y N**, 以免以外清除内存。
2. 按下 **<** (**Y**) **ENTER**。

RPN: 自动内存栈

本章说明在 RPN 模式下自动内存栈中计算是如何进行的。虽然您不必阅读并了解本资料即可使用本计算器，但是了解本资料将大大增强您对于本计算器的使用，尤其是在编程的时候。

在第 2 部分 -" 编程 " 中，您将了解到栈是如何能够帮助您操作并整理程序数据的。

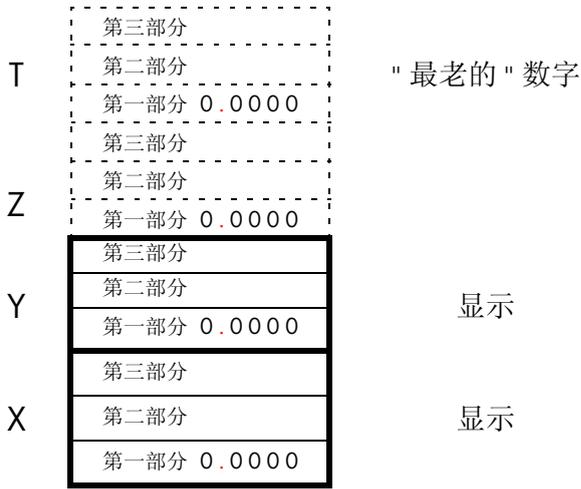
什么是栈

中间结果的自动保存即是 HP 35s 能够很容易地处理复杂计算且无括号表示的原因。自动保存的关键在于自动 RPN 内存栈。

HP 的运算逻辑依据的是一种清楚、无括号表示的数学逻辑，这种数学逻辑被称为波兰记数法，是由波兰逻辑学家 Jan Łukasiewicz(1878-1956) 提出的。

传统的算术几何计数法把运算符放在相关数字或变量之间，而 Jan Łukasiewicz 的计数法则把它们数字或变量的前面。为使栈的效率最佳，我们对该计数法进行了修改，指定数字后的运算符。因此，将这种计数法称为逆波兰式计数法，或简称为 RPN。

栈由 4 个被称为存储器的存储位置组成，这些存储位置上下迭在一起。这些寄存器，分别标记为 X、Y、Z 及 T，保存并操纵 4 个当前数字。" 最老的 " 数字保存在 T(顶部) 存储器中。栈是进行计算的工作区域。



X 寄存器中的数字最 "新": 该数字也是您在显示屏第 2 行中看到的数字。

每一个寄存器分为 3 部分:

- 真数或 1-D 矢量将占据第一部分; 这种情况下, 第二和第三部分为空。
- 复数或 2-D 矢量将占据第一和第二部分; 这种情况下, 第三部分为空。
- 3-D 矢量将占据第一、第二及第三部分。

在编程过程中, 栈用于进行计算, 临时保存中间结果, 在不同的程序及子程序间传递保存的数据 (变量), 接受输入并发送输出。

X 和 Y 寄存器都在显示屏上

在显示屏上显示菜单、信息、方程行或程序行之外的其它时间里，您在屏幕上看到的的就是 X 和 Y 寄存器。您可能已经注意到多个函数名称包含一个 x 或 y。

这并非巧合：这些字母指的是 X 和 Y 寄存器。 \leftarrow 10^x 使 X 寄存器中的数字升 10 次幂。

清除 X 寄存器

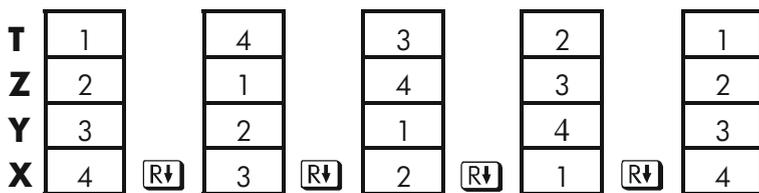
按 \rightarrow CLEAR 1(x) 始终将 X 寄存器清 0；该按键还可用于对本指令进行编程。相比之下，C 键则是区分上下文的。依据不同情况，该键清除或取消当前显示；只有在显示 X-寄存器时，该键才起于 \rightarrow CLEAR 1(x) 一样的作用。当 X-寄存器被显示且数字输入终止（无光标存在）时， \leftarrow 还起于 \rightarrow CLEAR 1(x) 一样的作用。

查阅栈

R↓ (向下滚动)

R↓ (向下滚动)使您能够通过向下"滚动"逐个查阅栈的全部内容。数字滚动通过x和寄存器时，你可以看到它们。

假设栈中填的是数字 1、2、3 和 4(按 1 ENTER 2 ENTER 3 ENTER 4)。
按 R↓ 四次使数字滚动返回至开始的地方。

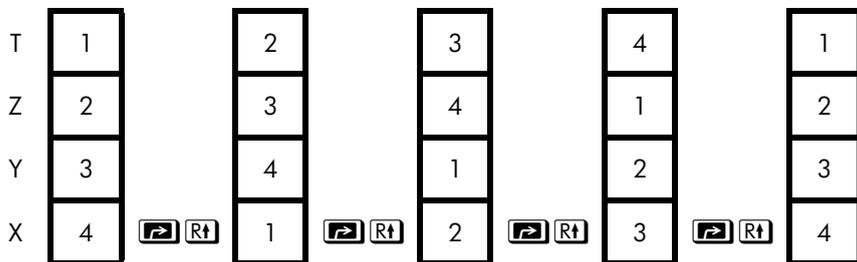


X寄存器中的内容转动至T寄存器中，而T寄存器中的内容则转动至Z寄存器中。注意只是寄存器的内容在转动，寄存器本身的位置保持不变且只显示X和Y寄存器的内容。

R↑ (向上滚动)

除了逐个将栈的内容向上“滚动”以外   (向上滚动) 键的功能与  内似。

X寄存器的内容转动至Y寄存器中，T寄存器的内容则转动至X寄存器中，依此类推。



交换栈中的 X 和 Y 寄存器

另外一个对栈的内容进行操作的键是  (x 交换 y)。该键可交换 X 寄存器和 Y 寄存器的内容而不影响栈的其它内容。按  两次恢复 X 和 Y 寄存器内容的原始顺序。

 功能主要用于交换计算中数字的顺序。

例如，计算 $9 \div (13 \times 8)$ 的一种方法：

按下        。

从左至右计算本表达式的键盘键为：

       。

注意

注意在任何特定时间栈中的数字都不会超过 4 个 - 无论何时，只要输入第 5 个数字，T 寄存器 (最上面的寄存器) 的内容将丢失。

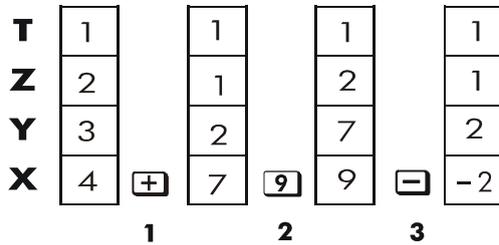


算术 – 栈是如何进行的

当新的数字进入 X 寄存器 (使栈上升) 且运算符将 X 寄存器和 Y 寄存器中的两个数字进行合并并在 X 寄存器中得出一个新数字时 (使栈下降), 栈的内容自动上下移动。

假设栈中填写的是数字 1、2、3 和 4, 看看计算时栈是如何使其内容上升和下降的

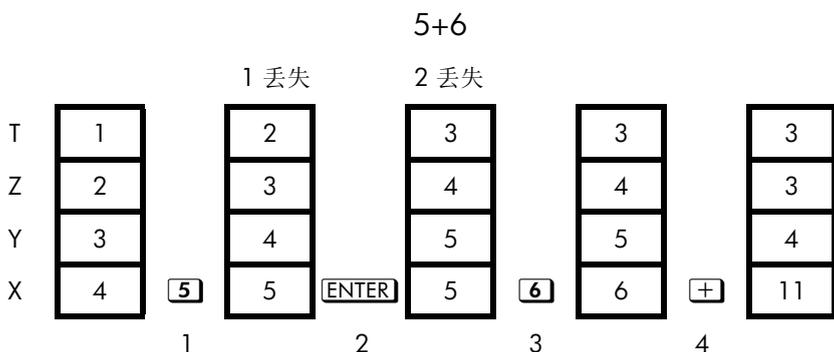
$$3+4-9$$



1. 栈使其内容 "下降"。T 寄存器 (最上面的寄存器) 复制其内容。
 2. 栈使其内容 "上升"。T 寄存器的内容丢失。
 3. 栈下降。
- 注意, 栈在上升时, 用 Z 寄存器的内容替换 T 寄存器 (最上面的寄存器) 的内容, 而 T 寄存器之前的内容则丢失。因此, 您可以看到栈的内存限制为 4 个数字。
 - 由于栈的自动移动, 在进行新的计算之前, 您无需清除 X 寄存器。
 - 大多数函数在下一个数字进入 X 寄存器时使栈做好提升其内容的准备。有关使栈提升禁用的函数列表, 请参阅附录 B。

ENTER 是如何工作的

您知道 **ENTER** 将两个依次输入的数字分开。就栈而言，它是如何做到这一点的呢？假定栈中再次填有数字 1、2、3 和 4，输入并使两个数字相加。



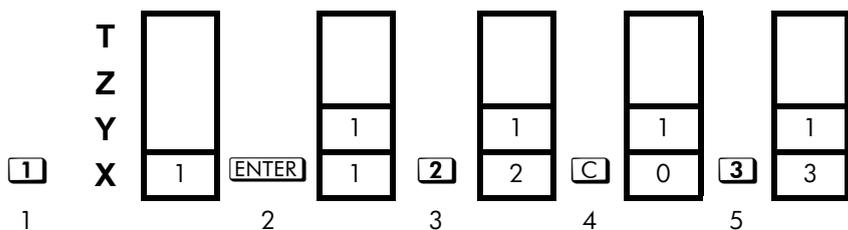
1. 使栈上升。
2. 使栈上升；复制 X 寄存器。
3. 不使栈上升。
4. 使栈下降，复制 T 寄存器。

ENTER 寄存器将 X 寄存器的内容复制到 Y 寄存器中。您键入(或恢复)的下一个数字将覆盖 X 寄存器中遗留的第一个数字的拷贝。其作用只是把两个顺序输入的数字分开。

您可以使用 **ENTER** 的复制作用迅速的将栈清除 按 **0** **ENTER** **ENTER** **ENTER**。所有的栈寄存器现在都包含 0。但是，需要注意的是在进行计算之前您无需将栈清除。

在一个行中两次使用某个数字

您可以其它有利的方式使用 **ENTER** 的复制功能。要想使某个数字与其自身相加，按 **ENTER** **+**。



1. 使栈上升
2. 使栈上升；复制 X 寄存器。
3. 覆盖 X 寄存器。
4. 用 0 覆盖清除 x。
5. 覆盖 x (替换 0。)

LAST X 寄存器

LAST X 寄存器是栈的伙伴，在执行最后一个数字函数前 (数字函数即从其它一个或多个数字，如 \sqrt{x} 得出结果的运算) 保持该数字。按 \rightarrow **LAST x** 将此值返回至 X 寄存器。

这种恢复 "Last x" 的能力主要有两种用途：

1. 纠正错误。
2. 在计算过程中重新使用某个数字。

有关在 LAST X 寄存器中保存 x 的函数的完整列表，请参阅附录 B。

纠正带 LAST X 的错误

单自变量函数错误

如果您执行错误的单自变量运算函数，使用  **LAST X** 恢复该数字，以便能够执行正确的函数。（如果您想要从栈中清除该错误结果，先按 **C**。）

由于  **%** 和  **%CHG** 不会使栈下降，您可以按照与单个自变量运算相同的方式从这些函数恢复。

示例：

假定您刚刚计算过 $\ln 4.7839 \times (3.879 \times 10^5)$ ，想要获得其平方根，但又错误按下 ，您无需从头开始。要想得出正确的答案，按  **LAST X** 。

双自变量运算错误

如果您在进行双自变量运算(如 **+**，**y^x** 或 **nCr**)时犯错，您可以通过  **LAST X** 及该双自变量运算的逆运算对错误进行纠正。

1. 按  **LAST X** 回复第二个数字 (x 刚好在运算前)。
2. 执行逆运算。此操作返回原先的第一个数字。第二个数字仍然在 LAST X 中。
然后：
 - 如果您使用的 *函数不正确*，再次按下  **LAST X** 恢复原始栈内容。现在，执行正确的函数。
 - 如果您使用的 *第二个数字不正确*，键入正确的第二个数字并执行该函数。

如果您使用的第一个数字不正确，键入正确的第一个数字，按  **LAST X** 恢复第二个数字，然后再次执行该函数（如果您想要从栈中清除该错误结果，先按 **C**。）

示例:

假设您在计算时犯错

$$16 \times 19 = 304$$

你可能会犯的错误有 3 种:

计算错误:

1 **6** **ENTER** **1**

9 **-**

1 **5** **ENTER** **1**

9 **x**

1 **6** **ENTER** **1**

8 **x**

错误:

函数错误

第一个数字错误

第二个数字错误

纠正:

↶ **LASTx** **+**

↶ **LASTx** **x**

1 **6** **↶** **LASTx** **x**

↶ **LASTx** **÷** **1** **9** **x**

重新使用带 LAST X 的数字

计算过程中, 您可以用 **↶** **LASTx** 来重新使用某个数字 (如某个常数)。记得在进行算术运算前再次输入该常数, 使该常数成为 X 寄存器中最后一个数字, 从而能够与 **↶** **LASTx** 一起保存并恢复。

示例:

计算 $\frac{96.704 + 52.3947}{52.3947}$

	T	<table border="1"><tr><td>t</td></tr></table>	t		<table border="1"><tr><td>t</td></tr></table>	t		<table border="1"><tr><td>t</td></tr></table>	t
t									
t									
t									
9 6	Z	<table border="1"><tr><td>z</td></tr></table>	z	5 2	<table border="1"><tr><td>z</td></tr></table>	z		<table border="1"><tr><td>t</td></tr></table>	t
z									
z									
t									
. 7	Y	<table border="1"><tr><td>96.7040</td></tr></table>	96.7040	. 3	<table border="1"><tr><td>96.7040</td></tr></table>	96.7040		<table border="1"><tr><td>z</td></tr></table>	z
96.7040									
96.7040									
z									
0 4	X	<table border="1"><tr><td>96.7040</td></tr></table>	96.7040	9 4	<table border="1"><tr><td>52.3947</td></tr></table>	52.3947	+	<table border="1"><tr><td>149.0987</td></tr></table>	149.0987
96.7040									
52.3947									
149.0987									
ENTER			7						

LAST	X	<table border="1"><tr><td>/</td></tr></table>	/		<table border="1"><tr><td>/</td></tr></table>	/	+	<table border="1"><tr><td>52.3947</td></tr></table>	52.3947
/									
/									
52.3947									

	T	<table border="1"><tr><td>t</td></tr></table>	t		<table border="1"><tr><td>t</td></tr></table>	t
t						
t						
	Z	<table border="1"><tr><td>z</td></tr></table>	z		<table border="1"><tr><td>t</td></tr></table>	t
z						
t						
	Y	<table border="1"><tr><td>149.0987</td></tr></table>	149.0987		<table border="1"><tr><td>z</td></tr></table>	z
149.0987						
z						
↶ LASTx	X	<table border="1"><tr><td>52.3947</td></tr></table>	52.3947	÷	<table border="1"><tr><td>2.8457</td></tr></table>	2.8457
52.3947						
2.8457						

LAST	X	<table border="1"><tr><td>52.3947</td></tr></table>	52.3947		<table border="1"><tr><td>52.3947</td></tr></table>	52.3947
52.3947						
52.3947						

键:

显示:

说明:

9 **6** **.** **7** **0** **4**

96.7040

输入第一个数字。

ENTER

5 **2** **.** **3** **9** **4**

149.0987

中间结果。

7 **+**

↶ **LASTx**

52.3947

在 **+** 前返回显示。

÷

2.8457

最终结果。

示例:

距离地球最近的两颗恒星分别为 Rigel Centaurus (距离为 4.3 光年) 和 Sirius (距离为 8.7 光年)。使用光速 c (每年 9.5×10^{15} 米) 将地球至这些恒星的距离转换为米

至 Rigel Centaurus: $4.3 \text{ yr} \times (9.5 \times 10^{15} \text{ m/yr})$ 。

至 Sirius: $8.7 \text{ yr} \times (9.5 \times 10^{15} \text{ m/yr})$ 。

键:	显示:	说明:
4 . 3 ENTER	4.3000	至 Rigel Centaurus 光年数。
9 . 5 E 1 5	9.5E15_	光速, c 。
X	4.0850E16	Meters 至 R. Centaurus 米数。
8 . 7 ↔ LASTx	9.5000E15	恢复 c 。
X	8.2650E16	至 Sirius 米数。

RPN 模式下链计算

在 RPN 模式下, 栈内容的自动上升和下降使您能够保留中间结果而无需对其进行保存或重新输入并使用括号表示。

从括号表示由内向外解题

例如, 评估 $(12 + 3) \times 7$ 。

如果您在纸上解答此题, 您需要首先计算 $(12 + 3)$ 的中间结果...

$$(12 + 3) = 15$$

...然后用中间结果乘 7:

$$(15) \times 7 = 105$$

从括号表示的内部开始, 按照 HP 35s 上相同的方式对表达式进行评估。

键:	显示:	说明:
1 2 ENTER 3 +	15.0000	首先计算中间结果

您无需按 **ENTER** 保存这一中间结果以进行下一步运算, 原因是这是一个 *计算出来* 的结果, 自动保存。

键:	显示:	说明:
7 x	105.0000	按函数键得出答案。该答案可用于进行进一步计算。

现在，研究以下几个例子。谨记你只需按 **ENTER** 把按顺序输入的数字 (如在表达式开始部分的数字) 分开。运算本身 (**+**), (**-**), 等) 可将后续数字分开并保存中间结果。最后的结果首先保存并恢复以进行计算。

计算 $2 \div (3 + 10)$:

键:	显示:	说明:
3 ENTER 1 0 +	13.0000	首先计算 $(3 + 10)$ 。
2 x\leftrightarrowy \div	0.1538	把 2 放在 13 的 <i>前面</i> , 因此除是正确的: $2 \div 13$ 。

计算 $4 \div [14 + (7 \times 3) - 2]$:

键:	显示:	说明:
7 ENTER 3 x	21.0000	计算 (7×3) 。
1 4 + 2 -	33.0000	计算分母。
4 x\leftrightarrowy	33.0000	把 4 放在 33 <i>前面</i> 准备进行除法运算。
\div	0.1212	计算 $4 \div 33$, 即答案。

有多个括号表示的问题可以通过自动保存中间结果按相同方式解答。例如, 要想在纸上求解 $(3 + 4) \times (5 + 6)$, 您需要先计算 $(3 + 4)$, 然后计算 $(5 + 6)$, 最后乘上两个中间结果以得出答案。

用和 HP 35s 一样的方式解题, 但您无需记下中间答案 - 计算器已经替您记下它们。

键:	显示:	说明:
3 ENTER 4 +	7.0000	首先加 $(3+4)$
5 ENTER 6 +	11.0000	然后加 $(5+6)$
x	77.0000	然后一起乘以中间结果以获得最终答案。

练习

计算:

$$\frac{\sqrt{(16.3805 \times 5)}}{0.05} = 181.0000$$

解:

1 **6** **.** **3** **8** **0** **5** **ENTER** **5** **×** **\sqrt{x}** **.** **0** **5** **÷**

计算:

$$\sqrt{[(2+3) \times (4+5)]} + \sqrt{[(6+7) \times (8+9)]} = 21.5743$$

解:

2 **ENTER** **3** **+** **4** **ENTER** **5** **+** **×** **\sqrt{x}** **6** **ENTER** **7** **+** **8** **ENTER**
9 **+** **×** **\sqrt{x}** **+**

计算:

$$(10 - 5) \div [(17 - 12) \times 4] = 0.2500$$

解:

1 **7** **ENTER** **1** **2** **-** **4** **×** **1** **0** **ENTER** **5** **-** **$x \leftrightarrow y$** **÷**

或

1 **0** **ENTER** **5** **-** **1** **7** **ENTER** **1** **2** **-** **4** **×** **÷**

计算顺序

我们建议从最里面的括号表示向外进行链计算。然后，您也可以选择按从左向右的顺序解题。

例如，您已经计算:

$$4 \div [14 + (7 \times 3) - 2]$$

从最里面的括号 (7 × 3) 开始向外, 就如同您用笔和纸的计算方式一样。键盘键为 **7** **ENTER** **3** **×** **1** **4** **+** **2** **-** **4** **x↔y** **÷**。

如果您从左向右解题, 按

4 **ENTER** **1** **4** **ENTER** **7** **ENTER** **3** **×** **+** **2** **-** **÷**。

这种方法需要一个额外的键盘键。注意, 第一个中间结果仍然是最里面的括号表示 (7 × 3)。从左向右解一个题的优点就是您无需使用 **x↔y** 重新定位非交换函数的运算数。(**-** 和 **÷**)。

但是, 通常情况下, 优先使用的是第一种方法 (从最里面的括号表示开始), 原因是这种方法:

- 使用的键盘键更少。
- 需要的栈中寄存器也更少。

注意

如使用从左向右法, 确保一次需要的中间数字 (或结果) 不超过 4 个 (栈能够保存的数字不超过 4 个)。



以上例子, 如从左向右解, 则在某个时刻需要栈中所有的寄存器:

键:	显示:	说明:
4 ENTER 1 4	14.0000	将 4 和 14 作为中间数字保存在栈中。
ENTER		
7 ENTER 3	3_	此时, 栈里满是用于本计算的数字。
×	21.0000	中间结果。
+	35.0000	中间结果。
2 -	33.0000	中间结果。
÷	0.1212	最终结果。

更多练习

在 RPN 模式下做下列题目进行练习：

计算：

$$(14 + 12) \times (18 - 12) \div (9 - 7) = 78.0000$$

解：

$$\boxed{1} \boxed{4} \boxed{\text{ENTER}} \boxed{1} \boxed{2} \boxed{+} \boxed{1} \boxed{8} \boxed{\text{ENTER}} \boxed{1} \boxed{2} \boxed{-} \boxed{\times} \boxed{9} \boxed{\text{ENTER}} \boxed{7} \boxed{-} \boxed{\div}$$

计算：

$$23^2 - (13 \times 9) + 1/7 = 412.1429$$

解：

$$\boxed{2} \boxed{3} \boxed{\rightarrow} \boxed{x^2} \boxed{1} \boxed{3} \boxed{\text{ENTER}} \boxed{9} \boxed{\times} \boxed{-} \boxed{7} \boxed{1/x} \boxed{+}$$

计算：

$$\sqrt{(5.4 \times 0.8) \div (12.5 - 0.7^3)} = 0.5961$$

解：

$$\boxed{5} \boxed{\cdot} \boxed{4} \boxed{\text{ENTER}} \boxed{\cdot} \boxed{8} \boxed{\times} \boxed{\cdot} \boxed{7} \boxed{\text{ENTER}} \boxed{3} \boxed{y^x} \boxed{1} \boxed{2} \boxed{\cdot} \boxed{5} \boxed{x \leftrightarrow y} \boxed{-}$$
$$\boxed{\div} \boxed{\sqrt{x}}$$

或

$$\boxed{5} \boxed{\cdot} \boxed{4} \boxed{\text{ENTER}} \boxed{\cdot} \boxed{8} \boxed{\times} \boxed{1} \boxed{2} \boxed{\cdot} \boxed{5} \boxed{\text{ENTER}} \boxed{\cdot} \boxed{7} \boxed{\text{ENTER}} \boxed{3} \boxed{y^x} \boxed{-}$$
$$\boxed{\div} \boxed{\sqrt{x}}$$

计算：

$$\sqrt{\frac{8.33 \times (4 - 5.2) \div [(8.33 - 7.46) \times 0.32]}{4.3 \times (3.15 - 2.75) - (1.71 \times 2.01)}} = 4.5728$$

解:

4 ENTER 5 · 2 = 8 · 3 3 ×  LASTx 7 · 4 6 =
0 · 3 2 × ÷ 3 · 1 5 ENTER 2 · 7 5 = 4 · 3 ×
1 · 7 1 ENTER 2 · 0 1 × = ÷ \sqrt{x}

将数据存入变量

HP 35s 有 30KB 的内存，其中您可以保存数字、方程及程序。保存数字的地方称为变量，各个变量以字母 A 至 Z 命名（您可以选择字母提醒您所保存的内容，例如字母 B 代表 *银行余额*，C 代表 *光速*）。

示例：

本例先以 RPN 模式，然后以 ALG 模式向您展示在变量 A 中输入数值 3 的方法。

键：	显示：	说明：
MODE 5 (5 RPN)		切换至 RPN 模式 (必要时)
3	0.0000 3_	输入数值 (3)
STO	STO_	保存指令提示一个字母，注意 A...Z 信号器。
A	0.0000 3.0000	数值 3 保存在 A 中并返回至栈。
MODE 4 (4 ALG)		切换至 ALG 模式 (必要时)
3 STO A	3.0000 3▶A_	保存指令再次提示一个字母，A...Z 信号器出现。
ENTER	3▶A 3.0000	数值 3 保存在 A 中，结果放在第二行中。

在 ALG 模式下，你可以将表达式存到变量中；在这种情况下，表达式的数值保存在变量而非表达式本身中。

示例：

键：	显示：	说明：
$\boxed{1} \boxed{+} \boxed{3} \boxed{\div} \boxed{4}$	1+3÷4	输入表达式，然后按照前例进行。
$\boxed{\rightarrow} \boxed{STO} \boxed{G} \boxed{ENTER}$	1.7500	

每一个粉红色字母都与一个键及一个独一无二的变量相关(显示屏上的 **A..Z** 信号器确认这一点。)

注意变量 X、Y、Z 和 T 是与栈中 X 寄存器、Y 寄存器、Z 寄存器及 T 寄存器不同的存储位置。

保存并恢复数字

数字及矢量通过保存 ($\boxed{\rightarrow} \boxed{STO}$) 及恢复 (\boxed{RCL}) 指令的方式保存至带有字母的变量中或从中恢复。数字可以是真数或复数、小数或分数、基数 10 或其它 HP 35s 支持的数字。

把显示数字 (**X 寄存器**) 的拷贝保存至直接变量中：

按 $\boxed{\rightarrow} \boxed{STO}$ 字母键 \boxed{ENTER} 。

把数字的拷贝从直接变量中恢复至显示屏：

按 \boxed{RCL} 字母键 \boxed{ENTER} 。

示例：保存数字。

将阿伏伽德罗常数 (约为 6.0221×10^{23}) 存入 A 中。

键:	显示:	说明:
6 . 0 2 2 1	6.0221E23_	阿伏伽德罗常数。
E 2 3	6.0221E23▶R_	"▶" 提示变量。
▶ STO A	6.0221E23▶R	把阿伏伽德罗常数的拷贝存入 A 中。
ENTER	6.0221E23	这将使数字输入终止。
C	_	清除显示屏中的数字。
RCL	A..Z	A..Z 信号器打开
A ENTER	R=	将阿伏伽德罗常数从 A 拷贝至显示屏。
	6.0221E23	

要想恢复保存在变量中的数值，使用恢复指令。该指令的显示在 RPN 模式下与在 ALG 模式小稍有不同，如下例所示。

示例:

在本例中，我们恢复在前例中保存在变量 G 的 1.75 的数值。本例假定开始时 HP 35s 仍然在 ALG 模式下。

键:	显示:	说明:
RCL G ENTER	G	按 RCL 激活 A...Z 模式；无任何指令被粘贴至第一行中。
	1.7500	

在 ALG 模式下，恢复可以用来将变量粘贴至命令行的表达式中。假定我们想要估算 $15-2 \times G$ ，如上所示 $G=1.75$ 。

键:	显示:	说明:
1 5 - 2 x	15-2×G	
RCL G ENTER	11.5000	

然后，切换至 RPN 模式并恢复 G 的数值。

键:	显示:	说明:
MODE 5 (SRPN)		切换至 RPN 模式
RCL	RCL _	在 RPN 模式下, RCL 将命令粘贴至编辑行中。
G	1.7500 1.7500	无需按下 ENTER 。

查看变量

查看命令 (**VIEW**) 显示变量的数值而不将该数值恢复到 x 寄存器中。显示的形式为变量 = 数值。如果数字的位数过多而无法与显示屏相适应, 使用 **RIGHT** **RIGHT** 或 **LEFT** **LEFT** 查看遗漏的位数。要想取消查看显示, 按 **LEFT** 或 **C**。查看命令最常用于编程过程中, 但在您希望查看变量的数值而不对栈产生影响时也有用。

使用内存目录

内存目录 (**MEM**) 提供有关变量内存数量的信息。目录显示有如下几种格式:

```
1. VAR      2. PGM
nnn        mm, mmm
```

式中, *mm, mmm* 为可用内存的字节数, *nnn* 为所用间接变量的数量。

有关间接变量的更多详情, 请参阅第 14 章。

变量目录

默认情况下, 所有从 A 到 Z 的直接变量都包含数值 0。如果您在任何直接变量中保存一个非 0 数值, 该变量的数值可在变量目录 (**MEM** **1** (1VAR)) 中查看。

示例:

在本例中，我们先把 3 存入 C，4 存入 D，5 存入 E，然后通过 VAR 目录查看这些变量并将其清除。本例使用 RPN 模式。

键:	显示:	说明:
CLEAR (2VAR)		清除所有直接变量
S)		
STO C	4	把 3 存入 C，4 存入 D，5 存入 E。
STO D	5	
STO E		
MEM (1VAR)	C=	进入变量目录。
	3	

注意代表 和 键的 和 信号器处于激活状态，可帮助您滚动查看目录。如果分数显示模式处于激活状态，除非目录中只有一个变量，否则 和 信号器将不处于激活状态并显示精确度。我们返回至上例，说明查看 VAR 目录的方法。

	D=	向下滚动至下一个带非0数值的直接变量：D=4。
	4	
	E=	再次向下滚动将看到 E=5。
	5	

在变量目录中，对本例进行扩展，展示将变量数值清 0 的方法，从而有效删除当前数值。我们将删除 E。

CLEAR	C=	E 不再在变量目录中，原因是其数值为 0。下一个变量显示为 C。
	3	

假设您现在想要把 C 的数值复制到栈中。

	5	C值=3被复制到x寄存器中,而5则移动(根据先前定义 E=5)y 寄存器中。
	3	

要想随时离开变量目录，按 \leftarrow 或 C 。另外一种把变量清除的办法就是将数值 0 存入其中。最后，按下 \rightarrow CLEAR 2 (2VARS)，您可以清除所有的直接变量。如果所有的直接变量都包含数值 0，尝试进入变量目录将显示错误信息 “ALL VARS = 0”。

如果变量的数值因为位数太多而无法完全显示，您可以使用 \rightarrow 和 \leftarrow 查看遗漏的位数。

✓ 带保存变量的算术

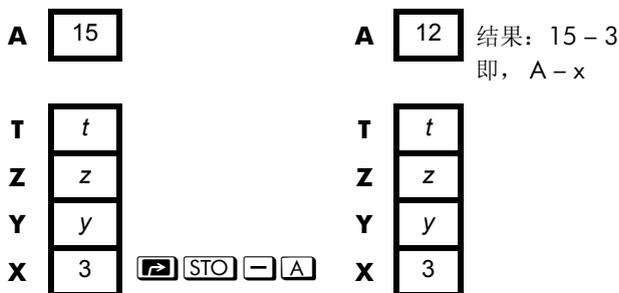
*保存算术*和*恢复算术*允许您用变量中保存的数字进行计算而*无需将该变量恢复至栈中*。计算使用 X 寄存器及指定变量的数字各一个。

✓ 保存算术

*保存算术*使用 \rightarrow STO $+$ ， \rightarrow STO $-$ ， \rightarrow STO \times 或 \rightarrow STO \div 在变量本身中进行算术运算并将结果保存在其中。保存算术使用 X 寄存器中的数值，不对栈产生影响。

新变量值 = 先前变量值 $\{+, -, \times, \div\}$ x。

例如，假定您想要用 X 寄存器中的数字 (显示为 3) 减去 A 中的数值 (15) 按 \rightarrow STO $-$ A 。现在 Now A = 12，而 3 仍然在显示屏上。

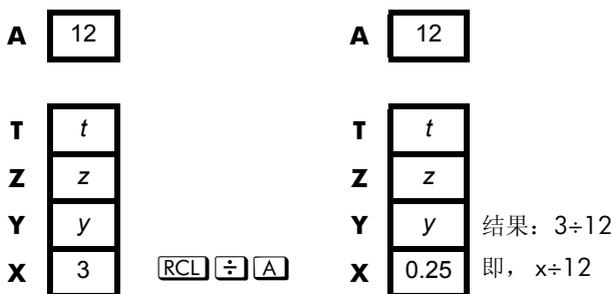


✓ 恢复算术

恢复算术用 RCL +, RCL -, RCL × 或 RCL ÷ 通过恢复的数字在 X 寄存器中进行算术运算并将结果留在显示屏中。只有 X 寄存器受到影响。变量中的数值保持不变, 而结果则取代 x 寄存器中的数值。

新 $x = \text{先前 } x \{+, -, \times, \div\}$ 。变量

例如, 假定您想要用 A 的数值 (12) 除以 X 寄存器中的数值 (显示为 3), 按 RCL ÷ A, 现在 $x=0.25$, 而 12 仍然在 A 中。恢复算术节省程序内存: 使用 RCL + A (一个指令) 所用的内存仅为 RCL A, + (两个指令) 的一半。



✓ 示例:

假定变量 D 、 E 和 F 包含数值 1、2 和 3，使用保存算术使 1 与这些变量相加。

键:	显示:	说明:
1 \rightarrow STO D	1.0000	将假设数值存入变量中。
2 \rightarrow STO E	2.0000	
3 \rightarrow STO F	3.0000	
1 \rightarrow STO		D 、 E 和 F 加一。
$+$ D \rightarrow STO	1.0000	
$+$ E \rightarrow STO		
$+$ F		
\leftarrow VIEW D	D=	显示当前数值 D 。
	2.0000	
\leftarrow VIEW E	E=	
	3.0000	
\leftarrow VIEW F	F=	
	4.0000	
\leftarrow	1.0000	清除查看显示；再次显示 X 寄存器。

根据上例，假定变量 D 、 E 和 F 包含数值 2、3 和 4，用 D 除 3、乘以 E 并将结果加上 F 。

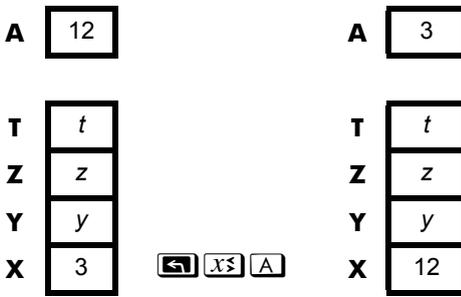
键:	显示:	说明:
3 RCL \div D	1.5000	计算 $3 \div D$ 。
RCL \times E	4.5000	$3 \div D \times E$ 。
RCL $+$ F	8.5000	$3 \div D \times E + F$ 。

用 x 与任何变量交换

\leftarrow $\text{X} \leftrightarrow$ 键允许您用 x (所显示的 X 寄存器) 的内容与任何变量的内容交换。执行本函数将不影响 Y -、 Z - 或 T 寄存器。

示例:

键:	显示:	说明:
1 2 STO A ENTER	12.0000	将 12 存入变量 A 中。
3	3_	显示 x。
↶ X↷ A	12.0000	交换 X 寄存器和变量 A 的内容。
↶ X↷ A	3.0000	交换 X 寄存器和变量 A 的内容。



变量 “I” 和 “J”

有两个变量您可以直接访问，即变量 I 和 J。虽然它们保存数值的方法和其它变量一样，变量 I 和 J 的独特之处在于它们可通过 (I) 和 (J) 指令用于参考其它变量，包括统计寄存器。(I) 见于 **0** 键上，而 (J) 则在 **◦** 键上。这是一种被称为间接定址的编程方法，详情见第 14 章 - “间接定址变量和标签”。

真数函数

本章讲述大多数在真数上进行计算的计算器函数，包括程序中使用的某些数字函数 (如 ABS, 绝对值函数等)。这些函数都成组讲述，如下：

- 指数函数和对数函数。
- 除法运算的商和余数。
- 幂函数。(x^y 及 $\frac{x}{y}$)
- 三角函数。
- 双曲函数。
- 百分比函数。
- 物理常量
- 坐标、角度及单位的转换函数。
- 概率函数。
- 数字部分 (变数函数)。

算术函数和计算在第 1 和第 2 章中讲述。高级数字运算 (求根、积分、复数、基数换算及统计) 在稍后各章讲述。本章中给出的所有例子都假定 HP 35s 处在 RPN 模式下。

✓ 指数函数和对数函数

先将数字放入显示屏中，然后执行函数 - 无需按 **ENTER**。

要想计算:	按:
自然函数 (基数为 e)	LN
常用对数 (基数为 10)	LOG
自然指数	e^x
常用指数 (反对数)	10^x

✓ 除法运算的商和余数

您可以用 INTG ($2INT\div$) 和 INTG ($3Rmdr$) 从两个整数的除法运算中分别得出整数商及整数余数。

1. 键入第一个整数。
2. 按 将第一个数字与第二个数字分开。
3. 键入第二个数字。(不要按 。)
4. 按下函数键。

示例:

要想显示 $58 \div 9$ 的商和余数

键:	显示:	说明:
	6.0000	显示商。
($2INT\div$)		
	4.0000	显示余数。
($3Rmdr$)		

✓ 幂函数

在 RPN 模式下, 要想计算 y 升 x 次幂的数字, 先键入 y x , 然后按 。
(若 $y > 0$, x 可以是任何数字; 若 $y < 0$, x 必须为正数。)

要想计算:	按:	结果:
152	$\boxed{1} \boxed{5} \boxed{\frac{\square}{\square}} \boxed{x^y}$	225.0000
106	$\boxed{6} \boxed{\frac{\square}{\square}} \boxed{10^x}$	1.000.000.0000
54	$\boxed{5} \boxed{\text{ENTER}} \boxed{4} \boxed{y^x}$	625.0000
2-1.4	$\boxed{2} \boxed{\text{ENTER}} \boxed{1} \boxed{\cdot} \boxed{4} \boxed{+/-} \boxed{y^x}$	0.3789
$(-1.4)^3$	$\boxed{1} \boxed{\cdot} \boxed{4} \boxed{+/-} \boxed{\text{ENTER}} \boxed{3} \boxed{y^x}$	-2.7440

在 RPN 模式下, 要想计算数字 y 的根 x (y 的 x 次根), 先键入 y $\boxed{\text{ENTER}}$ x , 然后按 $\boxed{\frac{\square}{\square}} \boxed{y^x}$ 。若 $y < 0$, x 必须为整数。

要想计算:	按:	结果:
$\sqrt{196}$	$\boxed{1} \boxed{9} \boxed{6} \boxed{\sqrt{x}}$	14.0000
$\sqrt[3]{-125}$	$\boxed{1} \boxed{2} \boxed{5} \boxed{+/-} \boxed{\text{ENTER}} \boxed{3} \boxed{\frac{\square}{\square}}$ $\boxed{y^x}$	-5.0000
$\sqrt[4]{625}$	$\boxed{6} \boxed{2} \boxed{5} \boxed{\text{ENTER}} \boxed{4} \boxed{\frac{\square}{\square}}$ $\boxed{y^x}$	5.0000
$-1.4\sqrt[3]{.37893}$	$\boxed{\cdot} \boxed{3} \boxed{7} \boxed{8} \boxed{9} \boxed{3} \boxed{\text{ENTER}} \boxed{1}$ $\boxed{\cdot} \boxed{4} \boxed{+/-} \boxed{\frac{\square}{\square}} \boxed{y^x}$	2.0000

三角学

输入 π

按 $\boxed{\frac{\square}{\square}} \boxed{\pi}$ 把前 12 位数放入 X 寄存器。

(所显示的数字取决于显示格式)。由于 $\boxed{\frac{\square}{\square}} \boxed{\pi}$ 是一个把的近似值返回到栈中的函数, 没有必要按 $\boxed{\text{ENTER}}$ 。

注意计算器不能够精确地表示 π , 因为 π 是一个超越数。

设置角模式

角模式指定三角函数中使用的角度该使用哪些量度单位。该模式不对已经存在的数字进行换算(见本章稍后的“换算函数”)。

$$360 \text{ 度} = 2\pi \text{ 半径} = 400 \text{ 度}$$

要想设置角模式,按 **MODE**。将显示一个菜单,从中您可以选择一个选项。

选项	说明	信号器
DEG	设置角度模式,该模式使用十进制角度而非十六进制角度(度、分、秒)	无
RAD	设置半径模式	RAD
GRAD	设置梯度模式	GRAD

✓ 三角函数

x 在显示屏上:

要想计算:	按:
x 的正弦值。	SIN
x 的余弦值。	COS
x 的正切值。	TAN
x 的余切值。	1/x ASIN
x 的余弦值。	1/x ACOS
x 的正切值。	1/x ATAN

注意



用无理数 π 进行的计算不能用计算器的15位内部精确度精确地加以表示。这一点在三角学中尤为明显。例如,计算得出的正弦 π (半径)不是0而是 -2.0676×10^{-13} ,这是一个接近0的很小的数字。

示例:

表示余弦 $(5/7)\pi$ 半径和余弦 128.57° 相等 (4 个有效数位)。

键:	显示:	说明:
MODE 2 (2RAD)		设置半径模式; RAD 信号器亮起。
. 5 . 7 ENTER	0.7143	小数格式的 $5/7$ 。
↵ π x COS	-0.6235	$\text{Cos } (5/7)\pi$ 。
MODE 1 (1DEG)	-0.6235	切换至角度模式 (无信号器)。
1 2 8 . 5 7	-0.6235	计算余弦 128.57° , 与余弦 $(5/7)\pi$
COS		相同。

编程说明:

使用反三角函数确定角度 θ 的方程式通常看起来是这样的:

$$\theta = \arctan (y/x)。$$

如 $x = 0$, 那么 y/x 不确定, 从而导致错误: **DIVIDE BY 0**。

双曲函数

x 在显示屏上:

要想计算:	按:
x 的双曲线正弦值 (SINH)。	HYP SIN
x 的双曲线余弦值 (COSH)。	HYP COS
x 的双曲线正切值 (TANH)。	HYP TAN
x 的双曲线余切值 (ASINH)。	HYP ASIN
x 的双曲线弧余弦值 (ACOSH)。	HYP ACOS
x 的双曲线正切值 (ATANH)。	HYP ATAN

✓ 百分比函数

百分比函数之所以特殊 (与 和 相比) 是因为它们在返回百分比计算 (在 X 寄存器中) 的结果时保留了基数的数值 (在 Y 寄存器中)。因此, 您可以使用基数和结果进行顺序计算而无需重新输入该基数。

要想计算:	按:
x% 的 y	y x %
从 y 到 x 的百分比变化。(y≠ 0)	y x %CHG

示例:

发现某总成本为 \$15.76 的销售税率为 6%。

使用 FIX 2 显示格式, 使成本相应进行四舍五入。

键:	显示:	说明:
 DISPLAY 1 (1FIX)		对显示的数字进行四舍五入至两个小数位。
2		
1 5 . 7 6 ENTER	15.76	
6  %	0.95	计算 6% 的销售税。
+	16.71	总成本 (底价 + 6% 的销售税)。

假定该成本为 \$15.76 的物品去年的价格为 \$16.12，那么今年的价格与去年的价格相比百分比变化为多少呢？

键:	显示:	说明:
1 6 . 1 2 ENTER	16.12	
1 5 . 7 6 	-2.23	与去年的价格相比，今年的价格下降了 2.2%。
%CHG		
 DISPLAY 1 (1FIX)	-2.2333	恢复 FIX 4 格式。
4		

注意 对于 %CHG 函数，两个数字的顺序很重要。顺序将影响该百分比变化是作为整数考虑还是作为负数考虑。



物理常量

常量 (CONST) 菜单中有 41 个常量菜单。您可以按  **CONST** 查看以下菜单。

CONST 菜单

项目	说明	数值
C	真空中光速	$299792458 \text{ m s}^{-1}$
G	标准重力加速度	9.80665 m s^{-2}
G	牛顿重力常数	$6.673 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$
Vm	理想气体的摩尔量	$0.022413996 \text{ m}^3 \text{ mol}^{-1}$
NA	阿伏加德罗常数	$6.02214199 \times 10^{23} \text{ mol}^{-1}$
R _∞	里德伯常数	$10973731.5685 \text{ m}^{-1}$
eV	电子电荷	$1.602176462 \times 10^{-19} \text{ C}$
m _e	电子质量	$9.10938188 \times 10^{-31} \text{ kg}$
m _p	质子质量	$1.67262158 \times 10^{-27} \text{ kg}$
m _n	中子质量	$1.67492716 \times 10^{-27} \text{ kg}$
m _μ	介子质量	$1.88353109 \times 10^{-28} \text{ kg}$
k	玻尔兹曼常数	$1.3806503 \times 10^{-23} \text{ J K}^{-1}$
h	普朗克常数	$6.62606876 \times 10^{-34} \text{ J s}$
ħ	2pi 以上的普朗克常数	$1.054571596 \times 10^{-34} \text{ J s}$
Φ ₀	磁通量子	$2.067833636 \times 10^{-15} \text{ Wb}$
a ₀	波尔半径	$5.291772083 \times 10^{-11} \text{ m}$
ε ₀	电气常数	$8.854187817 \times 10^{-12} \text{ F m}^{-1}$
R	摩尔气体常数	$8.314472 \text{ J mol}^{-1} \text{ K}^{-1}$
F	法拉第常数	$96485.3415 \text{ C mol}^{-1}$
u	原子质量常数	$1.66053873 \times 10^{-27} \text{ kg}$
μ ₀	磁性常数	$1.2566370614 \times 10^{-6} \text{ NA}^{-2}$
μ _B	玻尔磁子	$9.27400899 \times 10^{-24} \text{ J T}^{-1}$
μ _N	核磁子	$5.05078317 \times 10^{-27} \text{ J T}^{-1}$
μ _p	质子磁矩	$1.410606633 \times 10^{-26} \text{ J T}^{-1}$
μ _e	电子磁矩	$-9.28476362 \times 10^{-24} \text{ J T}^{-1}$
μ _n	中子磁矩	$-9.662364 \times 10^{-27} \text{ J T}^{-1}$
μ _μ	介子磁矩	$-4.49044813 \times 10^{-26} \text{ J T}^{-1}$

项目	说明	数值
r_e	经典电子半径	$2.817940285 \times 10^{-15} \text{ m}$
Z_0	特性真空阻抗	376.730313461Ω
λ_C	康普顿波长	$2.426310215 \times 10^{-12} \text{ m}$
λ_{Cn}	中子康普顿波长	$1.319590898 \times 10^{-15} \text{ m}$
λ_{Cp}	质子康普顿波长	$1.321409847 \times 10^{-15} \text{ m}$
α	精细结构常数	$7.297352533 \times 10^{-3}$
σ	斯蒂芬-玻尔兹曼常数	$5.6704 \times 10^{-8} \text{ W m}^{-2} \text{ K}^{-4}$
t	摄氏温度	273.15
atm	标准大气	101325 Pa
γ_P	质子磁旋比	$267522212 \text{ s}^{-1} \text{ T}^{-1}$
C1	第一辐射常数	$374177107 \times 10^{-16} \text{ W m}^2$
C2	第二辐射常数	0.014387752 m K
G_0	导电量子	$7.748091696 \times 10^{-5} \text{ S}$
e	自然对数的基数 (自然常数)	2.71828182846

参考: Peter J.Mohr and Barry N.Taylor, CODATA Recommended Values of the Fundamental Physical Constants: 1998, Journal of Physical and Chemical Reference Data, Vol.28, No.6, 1999 and Reviews of Modern Physics, Vol.72, No.2, 2000.

要想插入一个常数:

1. 将光标放在您想要插入常数的地方。
2. 按  **CONST** 显示物理常数菜单。
3. 按     (或者, 您可以按  **CONST** 访问下一页, 一次一页)滚动查看查看菜单, 直到您想要的常数被加上下划线, 然后按 **ENTER** 插入常数。

注意: 常数在用于表达式、方程式以及程序时, 应该用其名称而非数值来表示。

转换函数

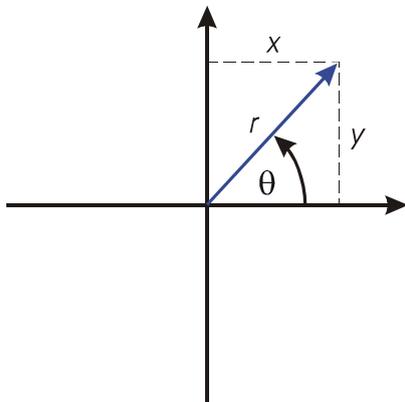
HP 35s 支持 4 种形式的转换。您可在以下项目间进行转换：

- 复数的矩形及极坐标格式
- 角度测量的角度、半径及梯度
- 时间的 10 进制格式及 16 进制格式 (以及角度)
- 各种支持的单位 (厘米 / 英寸, 千克 / 磅, 等)

除了矩形及极坐标转换外，每一种转换都与一个特定的键相关。键左移 (黄色) 以一种方式转换，而同一键右移 (蓝色) 则以另外一种方式转换。对于每一种转换，您输入的数字假定为用其它单位测量。例如，在使用 $\boxed{\rightarrow^{\circ}F}$ 将某个数字转换为华氏度数时，您输入的数字假定为一个用摄氏度数测量的温度。本章中的例子采用 RPN 模式。在 ALG 模式下，先输入函数，然后对数字进行转换。

矩形 / 极坐标转换

极坐标 (r, θ) 和矩形坐标 (x, y) 按照图示进行测量。角度 θ 使用当前角模式设置的单位。 θ 的计算结果为 -180° 和 180° ， $-\pi$ 和 π 半径或 -200 和 200 度。



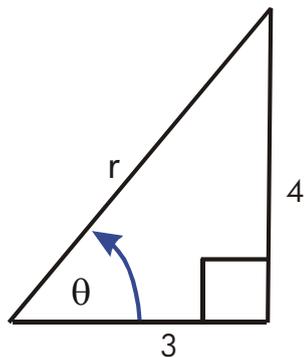
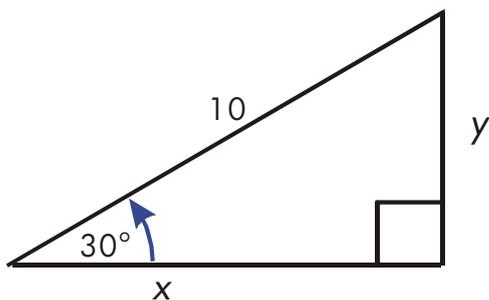
要想在矩形和极坐标进行转换：

表示复数的格式为模式设置。您可以以任何格式输入一个复数；输入后，该复数被转换模式设置确定的格式。以下所述为设置复数格式所需要的步骤：

1. 先按 $\boxed{\text{MODE}}$ $\boxed{1}$ (1 DEG)，然后在RPN模式下选择 $\boxed{9}$ ($9 \times i \cdot y$) 或 $\boxed{\cdot}$ $\boxed{0}$ ($1 \theta r \theta a$) (在ALG模式下，您还可以选择 $\boxed{\cdot}$ $\boxed{1}$ ($11 \times + y \cdot i$))
2. 输入您希望的坐标值 (x \boxed{i} y , x $\boxed{+}$ y \boxed{i} 或 r $\boxed{\text{RPN}}$ $\boxed{\theta}$ a)
3. 按 $\boxed{\text{ENTER}}$

示例： 极坐标 / 矩形转换

在一下直角三角形中，先找出左边三角形的边 x 和 y ，然后找右边三角形的斜边 r 和角度 θ 。



键：

$\boxed{\text{MODE}}$ $\boxed{1}$ (1 DEG)
 $\boxed{\text{MODE}}$ $\boxed{\text{DISP}}$ $\boxed{9}$ ($9 \times i \cdot y$)
 $\boxed{1}$ $\boxed{0}$ $\boxed{\text{RPN}}$ $\boxed{\theta}$ $\boxed{3}$ $\boxed{0}$
 $\boxed{\text{ENTER}}$

显示：

8.6603*i*5.0000

说明：

设置角度和复坐标模式。

将 $r \theta a$ (极坐标) 转换为 $x i y$ (矩形)。

\leftarrow DISPLAY \cdot 0
 (10r θ a)
 3 i 4 ENTER

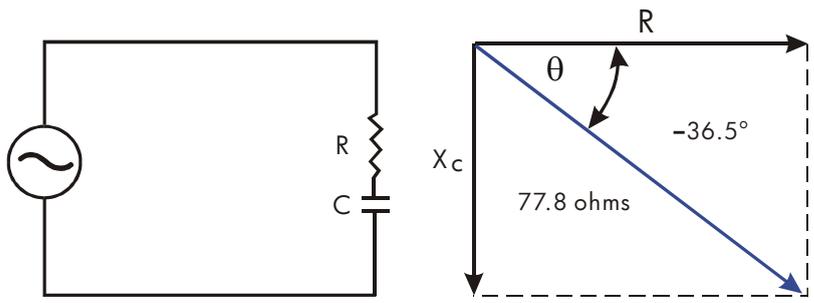
10.0000 θ 30.0000 设置复坐标模式。

5.0000 θ 53.1301 将 xiy(矩形) 转换为 r θ a(极坐标)。

示例： 矢量换算。

工程师 P.C. 波特认为，在所示的 RC 电路中，总阻抗为 77.8 欧姆且电压滞后电流达 36.5°。那么电路中电阻 R 以及电容电阻 X_C 的数值是多少呢？

使用所示的矢量图，阻抗等于极坐标幅度 r ，电压滞后等于角度 θ ，单位为度。当这些数值被转换为矩形坐标时， x 值得出 R ，单位为欧姆； y 值得出 X_C ，单位同样为欧姆。



键：

显示：

说明：

MODE 1 (1DEG)
 \leftarrow DISPLAY 9 (9x.i.v)
 7 7 \cdot 8 \rightarrow θ
 3 6 \cdot 5 \pm/\square
 ENTER

设置角度和复坐标模式。
 输入 θ ，电压滞后的度数。输入 r ，总阻抗的欧姆数。
 计算 x ，欧姆电阻， R 。
 计算 y ，欧姆电阻， X_C

77.8 θ -36.5

62.5401i-46.2772

时间转换

HP 35s 可在数字的 10 进制格式和 16 进制格式进行转换。这一点对于以度数测量的时间和角度特别有用。例如，在 10 进制格式下，以度数测量的角度表示为 D.ddd...，而在 16 进制格式下，这同一个角度则表示为 D.MMSSss，式中 D 为角度测量的整数部分，ddd... 为角度测量的分数部分，MM 为分钟的整数，SS 为秒数的整数部分，ss 为秒数的分数部分。

✓ 要想在 10 进制格式，小时、分钟和秒之间进行转换：

1. 输入您想要转换的数字
2. 按 \leftarrow \rightarrow HMS 转换为时间、角度、分钟及秒或按 \rightarrow \leftarrow HMS 转换回 10 进制格式。

示例：转换时间格式。

一小时的 1/7 是多少分钟，多少秒呢？使用 FIX 6 显示格式。

键：	显示：	说明：
\leftarrow DISPLAY 1 (1FIX)		设置 FIX 6 显示格式。
6	0.000000	作为纯小数的 1/7 小时。
. 1 . 7	0 1/7	
\rightarrow \leftarrow HMS	0.000000	等于 8 分钟 34.29 秒。
	0.083429	
\leftarrow DISPLAY 1 (1FIX)	0.000000	恢复 FIX 4 格式。
4	0.0834	

✓ 角度转换

当转换为半径时，x 寄存器中的数字假定为度数；当转换为度数时，x 寄存器中的数字假定为半径。

想在度数和半径之间转换角度：

示例

在本例中，我们将 30° 的角度测量转换为 $\pi/6$ 半径。

<p>键：</p> <p>3 0</p> <p>↵ →RAD</p>	<p>显示：</p> <p>0.0000</p> <p>30</p> <p>0.0000</p> <p>0.5236</p>	<p>说明：</p> <p>输入以度数表示的角度。</p> <p>转换为半径。结果读数为 0.5236，$\pi/6$ 的一个小数近似值。</p>
--	--	--

单位转换

HP 35s 的键盘上有 10 单位转换函数： \rightarrow kg, \rightarrow lb, \rightarrow °C, \rightarrow °F, \rightarrow cm, \rightarrow in, \rightarrow l, \rightarrow gal, \rightarrow MILE, \rightarrow KM

要想转换：	要想：	按：	显示结果：
1 lb	kg	1 ↵ →kg	0.4536 (千克)
1 kg	lb	1 ↵ →lb	2.2046 (磅)
32 °F	°C	3 2 ↵ →°C	0.0000 (°C)
100 °C	°F	1 0 0 ↵ →°F	212.0000 (°F)
1 in	cm	1 ↵ →cm	2.5400 (厘米)
100 cm	in	1 0 0 ↵ →in	39.3701 (英寸)
1 gal	l	1 ↵ →l	3.7854 (升)
1 l	gal	1 ↵ →gal	0.2642 (加仑)
1 MILE	KM	1 ↵ →KM	1.6093(KMS)
1 KM	MILE	1 ↵ →MILE	0.6214(里)

概率函数

✓ 因素

要想计算所显示的非负整数 x ($0 \leq x \leq 253$) 的因素, 按 $\boxed{\rightarrow} \boxed{!}$ (右移 $\boxed{\Sigma+}$ 键)。

✓ 伽玛

要想计算非复整数 x 的伽玛函数 $\Gamma(x)$, 键入 $(x - 1)$ 并按 $\boxed{\rightarrow} \boxed{!}$. $x!$ 函数计算 $\Gamma(x + 1)$ 。 x 的值不可以为负整数。

概率

✓ 组合

要想计算一次可取的 n 项目的组数 r , 先输入数 n , $\boxed{\leftarrow} \boxed{nCr}$, 然后输入 r (仅限非负整数)。同一组中同一项目出现的次数不得超过 1 次, 相同 r 项目的不同顺序不单独计数。

✓ 排列

要想计算一次可取的 n 项目的排列 r , 先如数 n , $\boxed{\rightarrow} \boxed{nPr}$, 然后输入 r (仅限非负整数)。同一排列中同一项目出现的次数不得超过 1 次, 相同 r 项目的不同顺序单独计数。

✓ 种子

要想在 x 中将数字作为随机数发生器的新种子保存, 按 $\boxed{\leftarrow} \boxed{SEED}$ 。

✓ 随机数发生器

要想在 $0 < x < 1$ 的范围内产生一个随机数, 按 $\boxed{\rightarrow} \boxed{RAND}$ 。(该数字是一个均布伪随机数序的一部分, 通过了 Knuth 博士的特殊试验。见《计算机编程艺术第 2 卷半数字算法》, 伦敦, Addison Wesley, 1981。)

随机函数使用一个种子来得出一个随机数。每一个得出随机数都成为下一个随机数的种子。因此，随机数的顺序可从相同的种子开始加以重复。您可以保存一个带种子函数的新种子。如果内存被清除，该种子重置为 0。种子为 0 将导致计算器产生其自己的种子。

示例： 人员组合。

某公司雇有 14 名女性员工和 10 名男性员工，正欲组建一个 6 人安全委员会。不同的人员组合可能会有多少种呢？

键：	显示：	说明：
2 4 ENTER 6	24	24 人一次 6 人分组。
6 nCr	6_ 134,596.0000	可能的组合总数。

如果随机选择雇员，委员会包括 6 位妇女的概率是多少呢？要想获得一件事情的概率，用总组合数除以该事件的组合数。

键：	显示：	说明：
1 4 ENTER 6	14	14 位妇女一次 6 人分组。
6 nCr	6_ 3,003.0000	委员会 6 位妇女的组合数。
X↔Y	134,596.0000	将组合总数放回到 X 寄存器中。
÷	0.0223	用总组合数除以妇女组合数，以获得包含所有妇女的组的概率。

数字部分

这些函数主要用于编程。

整数部分

- ✓ 要想去掉 x 的分数部分并用 0 替换，按  **INTG**  (**6IP**)。(例如，14.2300 的整数部分为 14.0000。)

分数部分

- ✓ 要想去掉 x 的整数部分并用 0 替换，按  **INTG**  (**5FP**)。(例如，14.2300 的分数部分为 0.2300。)

绝对值

要想用其绝对值替换 x 寄存器中的数字，按  **ABS**。对于复数和矢量，绝对值为：

1. $r\theta a$ 格式的复数为 r
2. xiy 格式的复数为 $\sqrt{x^2 + y^2}$
3. 矢量 $[A_1, A_2, A_3, \dots, A_n]$ 为 $|A| = \sqrt{A_1^2 + A_2^2 + \dots + A_n^2}$

自变量值

要想提取复数的自变量，使用  **ARG**。复数的自变量：

1. $r\theta a$ 格式为 a
2. xiy 格式为 $A \tan(y/x)$

符号值

- ✓ 要想显示 x 的符号，按  **INTG**  (**1SGN**)。如果 x 值为负，显示 -1.0000；如果为 0，显示 0.0000；如果为正，显示 1.0000。

最大整数

✓ 要想获得等于或大于给定数字的最大整数，按 $\boxed{\leftarrow}$ $\boxed{\text{INTG}}$ $\boxed{4}$ (4INTG)。

示例：

本例总结了许多抽出数字部分的运算。

要想计算：	按：	显示：
2.47 的整数部分	$\boxed{2}$ $\boxed{\cdot}$ $\boxed{4}$ $\boxed{7}$ $\boxed{\leftarrow}$ $\boxed{\text{INTG}}$ $\boxed{6}$ (6IP)	2.0000
2.47 的分数部分	$\boxed{2}$ $\boxed{\cdot}$ $\boxed{4}$ $\boxed{7}$ $\boxed{\leftarrow}$ $\boxed{\text{INTG}}$ $\boxed{5}$ (5FP)	0.4700
-7 的绝对值	$\boxed{7}$ $\boxed{+/-}$ $\boxed{\rightarrow}$ $\boxed{\text{ABS}}$	7.0000
9 的符号值	$\boxed{9}$ $\boxed{\leftarrow}$ $\boxed{\text{INTG}}$ $\boxed{1}$ (1SGN)	1.0000
等于或小于 -5.3 的最大整数	$\boxed{5}$ $\boxed{\cdot}$ $\boxed{3}$ $\boxed{+/-}$ $\boxed{\leftarrow}$ $\boxed{\text{INTG}}$ $\boxed{4}$ (4INTG)	-6.0000

RND函数($\boxed{\rightarrow}$ $\boxed{\text{RND}}$) 在内部将x四舍五入至显示格式指定的位数(内部数字用12位表示)。有关分数显示模式下 RND 的性能，请参阅第 5 章。

如果您得出的结果与示例不同，您可能不经意间修改了分数的显示方式。(见本章稍后部分 - 修改分数显示。)

下一话题包含更多有效和无效输入分数的例子。

显示屏中的分数

在分数显示模式下，数字先在内部作为小数进行评估，然后用允许的最精确分数进行显示。此外，精确度信号器显示与其 12 位小数值相比的任何不精确的方向。(大多数统计寄存器都属例外 - 它们始终作为小数显示。)

显示规则

您看到的分数可能会与您输入的不同。在默认状态下，本计算器根据以下规则显示分数。(要想修改显示规则，见本章稍后部分 - "修改分数显示")

- 数字包括整数部分，必要时还将包括一个真分数(分子小于分母)。
- 分母不大于 4095。
- 尽可能减分子。

示例:

以下为输入数值及显示结果的例子。为进行比较，还显示了内部 12 位数值。最后一栏中的 ▲ 和 ▼ 信号器说明如下。

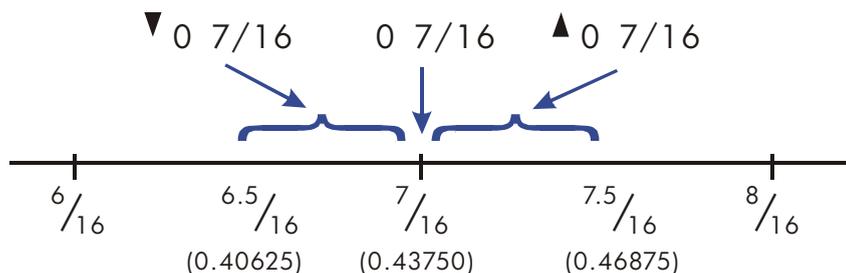
输入的数值	内部值	显示的分
$2\frac{3}{8}$	2.3750000000	$2\frac{3}{8}$
$14\frac{15}{32}$	14.4687500000	$14\frac{15}{32}$
$54/12$	4.5000000000	$4\frac{1}{2}$
$6\frac{18}{5}$	9.6000000000	$9\frac{3}{5}$
$34/12$	2.8333333333	$2\frac{5}{6}$ ▼
$15/8192$	0.00183105469	$0\frac{7}{3823}$ ▲
$12345678\frac{12345}{3}$	12349793.0000	12349793
$16\frac{3}{16384}$	16.0001831055	$16\frac{1}{4095}$

精度指示器

所显示的分数的精度由显示屏右边的 ▲ 和 ▼ 指示器指示。计算器把内部 12 位数的分数部分的数值与所显示的分数的数值进行比较：

- 如果无信号器亮起，则表示内部 12 位数的分数部分与所显示的分数的数值精确吻合。
- 如果 ▼ 亮起，则表示内部 12 位数值分数部分略小于所显示的分数——确切的分子比所显示的分子要低不超过 0.5。
- 如果 ▲ 亮起，则表示内部 12 位数值分数部分略大于所显示的分数——确切的分子比所显示的分子要高不超过 0.5。

本图显示了所显示的分数与临近数值的关系——▲ 表示确切的分子要“略高于”所显示的分子，而 ▼ 则表示确切的分子要“略低于”所显示的分子。



这一点在您修改分数的显示方式时尤其重要。(见本章稍后部分“修改分数显示”)。例如,如果您将5作为所有分数的分母,则 $2/3$ 显示为 $0\ 3\ 5\blacktriangle$,原因是确切分数接近 $3.3333/5$,“略高于” $3/5$ 。同样地, $-2/3$ 显示为 $-0\ 3\ 5\blacktriangle$,原因是真分子“略高于”3。

有些时候,有些信号器会在您不希望的时候亮起。例如,如果您输入 $2\ 2/3$,您将看到 $2\ 2\ 3\blacktriangle$,虽然那是您输入的确切数字。计算器始终把内部数值的分数部分与分数的12位数值进行比较。如果内部数值包含整数部分,其分数部分将包含不少于12位数,而且不于使用全部12位数的分数精确吻合。

修改分数显示

在默认状态下,计算器始终根据一定的规则显示分数。但是,您也可以按照自己希望的方式修改分数显示规则:

- 您可以设置所用的最大分母。
- 您可以从3种分数格式中任选一种。

以下几个话题将向您展示修改分数显示的方法。

设置最大分母

对于任何分数,其分母都是依据计算器中保存的最大数值进行选择的。如果您考虑的分数为 $a\ b/c$,则 $/c$ 对应于控制分母的数值。

$/c$ 值只定义分数显示模式下使用的最大分母—所用的特定分母由分数格式确定(在下一话题中探讨)。

- 要想设置最大分母值，输入数值，然后按 $\boxed{\leftarrow} \boxed{/c}$ 。分数显示模式将自动被禁用。您输入的数值将不超过 4095。
- 要想将 $/c$ 值恢复到 X 寄存器中，按 $\boxed{1} \boxed{\leftarrow} \boxed{/c}$ 。
- 要想将默认值恢复为 4095，按 $\boxed{0} \boxed{\leftarrow} \boxed{/c}$ 或将任何大于 4095 的数值作为最大分母输入。然后，分数显示模式将自动被禁用。

$/c$ 函数使用 X 寄存器中数字整数部分的绝对值，不修改 LAST X 寄存器中的数值。

如果所显示的分数的太长而与显示屏不相适应， \blacktriangleright 信号器将出现。您可以用 $\boxed{\rightarrow} \boxed{\leftarrow}$ 和 $\boxed{\rightarrow} \boxed{\triangleright}$ 逐页滚动，这样您将看该分数的其余部分。要想查看该数字的小数表示，按 $\boxed{\leftarrow}$ ，然后保持 $\boxed{\text{SHOW}}$ 不放。

示例：

本例说明了将最大分母设置为 3125 所需要的步骤，然后显示一个太长而无法完全显示的分数的。

键：	显示：	说明：
$\boxed{3} \boxed{1} \boxed{2} \boxed{5} \boxed{\leftarrow}$		将最大分母设置为 3125。
$\boxed{/c}$	0	注意分母中遗漏的位数。
$\boxed{1} \boxed{4} \boxed{\rightarrow} \boxed{e^x}$	1202604 888/31	
$\boxed{\triangleright}$	0 25	向右滚动将可看到分母的其余部分。

注意：

1. 在 ALG 模式下，您可以在第一行中输入一个表达式，然后按 $\boxed{\leftarrow} \boxed{/c}$ 。在此情况下，先对表达式进行评估，然后将结果用于确定最大分母。

2. 在ALG模式下，你可以将计算的结果用做/c函数的自变量。对于第2行中的数值，按   即可。第2行中的数值以分数格式显示，而其整数部分则被用来确定最大坟墓。
3. 您不可以将复数或矢量用作/c指令的自变量。将显示错误信息 "INVALID DATA"。

选择分数格式

本计算器有 3 种分数格式。所显示的分数始终为所选格式的规则内最为精确的分数。

- **最精确分数。**分数的任何分母都可高达至/c值，且尽可能减。例如，如果您正在研究带分数的数学概念，您可能希望任何分母都是可能的(/c值为4095)。这是默认的分数格式。
- **分母的因数。**分数只有属于/c值的因数的分母，且尽可能减。例如，如果您正在计算股票价格，您可能希望看到 $53 \frac{1}{4}$ 和 $37 \frac{7}{8}$ (/c值为8)。或如果/c值为12，可能的分母为2、3、4、6和12。
- **固定分母。**分数始终使用/c值作为分母，且不减。例如，如果您正在进行时间测量，您可能希望看到 $1 \frac{25}{60}$ (/c值为60)。

有 3 个标志控制分数格式。这些格式的数字编号为 7、8 和 9。每一个标志都清除或设置。其所有如下：

- 标志7 打开或关闭分数显示模式；清除 = 关闭，设置 = 打开。
- 标志8 在使用任何小于或等于/c值的数值或仅使用/c值的因数间切换；清除 = 使用任何数值，设置 = 仅使用/c值的因数。
- 标志9 仅在标志8被设置且在减或不减分数之间切换时起作用；清除=减，设置 = 不减。

标志 8 和 9 正确清除或设置后，您就可以获得如表中所示的 3 种分数格式：

要想获得本分数格式	修改这些标志:	
	8	9
最精确	清除	—
分母的因数	设置	清除
固定分母	设置	设置

您可以用以下所列的步骤修改标志 8 和 9 设置分数格式。(由于标志在程序中特别有用,其使用方法在第 14 章中详细讲述。)

- 按 **←** **FLAGS** 获取标志菜单。
- 要想设置标志,按 **1**(1SF) 并键入标志编号,如 8。
要想清除标志,按 **2**(2CF) 并键入标志编号。
要想查看一个标志是否已设置,按 **3**(3FS?) 并键入标志编号。按 **C** 或 **←** 清除 YES 或 NO 响应。)

示例:

本例使用数字 π 说明 3 种格式的分数的显示。本例假定分数显示格式处于激活状态,标志 8 处于其默认状态(被清除)。

键:	显示:	说明:
4 0 9 5 ←		将最大 /c 设置回其默认值。
/c		
← π	0	最精确格式
	3 16/113	标志 8= 清除。
← FLAGS 1 (1SF)	0	标志 8= 设置;
8	3 116/819	分母格式的因数: 819*5=4095
← FLAGS 1 (1SF)	0 0/4095	标志 9= 设置;
9	3 580/4095	固定分母格式
← FLAGS 2 (2CF)	0	返回至默认格式(最精确)
8 ← FLAGS 2 (2CF)	3 16/113	
9		

分数显示举例

下表说明数字 2.77 是如何以 3 种分数格式针对两个 /c 值显示的。

分数格式	2.77 是如何显示的	
	/c = 4095	/c = 16
最精确	2 77/100 (2.7700)	2 10/13▲ (2.7692)
分母的因数	2 1051/1365▲ (2.7699)	2 3/4▲ (2.7500)
固定分母	2 3153/4095▲ (2.7699)	2 12/16▲ (2.7500)

下表说明不同的数字是如何以 3 种分数格式针对 /c 值 16 显示的。

分数格式 *	输入的数字和显示的分数				
	2	2.5	2 2/3	2.9999	216/25
最精确	2	2 1/2	2 2/3▲	3▼	2 9/14▼
分母的因数	2	2 1/2	2 11/16▼	3▼	2 5/8▲
固定分母	2 0/16	2 8/16	2 11/16▼	3 0/16▼	2 10/16▲

* 对于 /c 值 16。

对分数进行四舍五入

如果分数显示模式处于激活状态，RND 函数将 X 寄存器中的数字转换为分数的最接近的小数表示。四舍五入按照当前的 /c 值及标志 8 和 9 的状态进行。如果函数与小数表示精确吻合，则精度指示器关闭。否则，精度指示器恒亮（见本章先前部分 -“精度指示器”）。

在方程或程序中，当分数显示模式处于激活状态时，由 RND 函数进行分数的四舍五入。

示例:

假设你有面积为 $56 \frac{3}{4}$ 英寸的空间想要等分为 6 段。假定您能够很方便地测量 $1/16$ 英寸增量，那么每一段的宽度是多少呢？累计的四舍五入误差又是多少呢？

键:	显示:	说明:
\leftarrow [FLAGS] [ENTER] [8]		设置标志 8
[1] [6] \leftarrow [1] [6]		建立 $1/16$ 英寸增量的分数格式(标志 8 和 9 应与前例相同)。
[5] [6] [.] [3] [.] [4]	$56 \frac{3}{4}$	将距离保存在 D 中。
\rightarrow [STO] [D]		
[6] [\div]	$9 \frac{7}{16} \blacktriangle$	各段的宽度要略大于 $9 \frac{7}{16}$ 英寸。
\rightarrow [RND]	$9 \frac{7}{16}$	将宽度四舍五入为本数值。
[6] [x]	$56 \frac{5}{8}$	6 个段的宽度。
[RCL] [D] [-]	$-0 \frac{1}{8}$	累计四舍五入误差。
\leftarrow [FLAGS] [2] (2CF) [8]	$-0 \frac{1}{8}$	清除标志 8。
\rightarrow [FDISP]	-0.1250	关闭分数显示模式。

等式中的分数

您可以在等式中使用分数。当一个等式被显示时，等式中所有的数值都以其输入时的形式显示。此外，分数显示模式可用于涉及等式的运算。

当您对比等式进行评估且收到要求输入变量值的提示时，您可以输入分数 - 各数值均以当前显示格式显示。

有关方程运算的更多详情，请参阅第 6 章。

程序中的分数

您可以像在方程中一样在程序中使用分数；数值均以其输入时的形式显示。

当您运行某个程序时，如果分数模式处于激活状态，所显示的数值都以分数显示模式显示。如果输入指令提示您输入数值，你可以输入分数。程序的结果用当前显示格式显示。

程序可通过使用 `/c` 函数和使用并清除标志 7、8 和 9 控制分数显示。见第 14 章“标志”。

有关程序运算的更多详情，请参阅第 13、14 章。

输入并评估方程

方程的使用方法

在 HP 35s，您可以以多种方式使用方程：

- 指定要评估的方程 (本章)。
- 指定方程求解未知数值 (第 7 章)。
- 指定函数进行积分 (第 8 章)。

示例：用方程进行计算。

假设您经常需要确定一段直管的体积。方程为

$$V = .25 \pi d^2 l$$

式中 d 为管道的内径， l 为其长度。

- ✓ 您可以重复键入计算。例如， $\boxed{\cdot} \boxed{2} \boxed{5} \boxed{\text{ENTER}} \boxed{\leftarrow} \boxed{\pi} \boxed{\times} \boxed{2} \boxed{\cdot} \boxed{5} \boxed{\rightarrow} \boxed{x^2} \boxed{\times} \boxed{1} \boxed{6} \boxed{\times}$ 计算长度为 16 英寸，直径为 2 1/2 的管道的体积 (78.5398 立方英寸)。但是，通过保存方程，您让 HP 35s“记住”了直径、长度和体积间的关系 – 因此，您可以重复多次使用该方程。

先将计算器置于方程模式下，然后使用以下键盘键键入方程：

键:	显示:	说明:
EQN	EQN LIST TOP	选择方程模式, 如 EQN 信号器所示。
RCL	或第 2 行中的当前方程	开始一个新方程, RCL 打开 A.Z 信号器, 使您能够输入变量名称。
V ← =	V=_	RCL V 输入 V
. 2 5	V=0.25_	数字输入使用 "_" 输入光标。
× ← π ×	V=0.25×π×_	数字以 × 结尾。
RCL D y^x 2	V=0.25×π×D^2_	y^x 输入 ^。
× RCL L	V=0.25×π×D^2×L_	
ENTER	V=0.25×π×D^2×L	终止并显示方程。
↵ SHOW	CK=49CA LN=14	显示方程的校和及长度, 这样您就能够检查键盘键。

通过把您的方程的校和及长度与示例中的校和及长度进行比较, 您可以验证您已经正确输入方程 (详情请参阅本章结尾处 -" 验证方程 ")。

对方程进行评估 (计算 V):

键:	显示:	说明:
ENTER	D? 数值	方程的右边提示输入变量。先提示输入 D; 数值为当前 D 值。
2 . 1 . 2	D? 2 1/2_	将 2 1/2 英寸作为分数输入。
R/S	L? 数值	保存 D, 提示输入 L; 数值为当前 L 值。
1 6 R/S	V= 78.5398	保存 L; 以立方英寸为单位计算 V 并将结果保存在 V 中。

方程运算概述

您创建的所有方程都保存在 *方程列表* 中。无论何时，只要您激活方程模式都可看到该列表。

您使用某些键进行涉及方程的运算。这些键在本文稍后部分详加说明。

在方程列表中显示方程时，一次显示两个方程。当前可用的方程在第 2 行中显示。

键	运算
	进入及退出方程模式。
	对所显示的方程进行评估。如果方程为 <i>赋值</i> ，对右边进行评估，并将结果保存在左边的变量中。如果方程为 <i>等式或表达式</i> ，像  一样计算其数值（见本章稍后部分 -“方程类别”）。
	对所显示的方程进行评估。计算其数值，如果存在“=”的话，用“-”替换“=”。
	求解所显示的方程，以获得您指定的未知变量。（请参阅第 7 章。）
	针对您指定的变量，对所显示的方程进行积分。（请参阅第 8 章。）
	删除当前方程或光标左边的元素。
 或 	开始编辑所显示的方程，只移动光标，不删除任何内容。
  或  	滚动当前方程显示列表。
 或 	向上或向下滚动查看方程列表。
  或  	跳至方程列表的顶部和底部。
 SHOW	显示所显示方程的校和(验证值)和长度(内存字节数)。
 UNDO	恢复最近删除的元素或方程。
	退出方程模式。

您还可以在程序中使用方程 — 这一点在第 13 章中探讨。

将方程输入方程列表

*方程列表*是您输入的所有方程的集合，保存在计算器的内存中。您输入的每一个方程都保存在方程列表中。

要想输入方程：

您可以随心所欲地创建方程 – 唯一的限制在于可用内存的数量。

1. 确认计算器处于起正常运行模式，通常显示屏中有一个数字。例如，您可以查看变量或程序的目录。
2. 按下 **[EQN]**。**EQN** 信号器显示方程模式处于激活状态，且显示一个来自方程列表的输入。
3. 开始输入方程。先前的显示被您输入的方程所取代 – 先前的方程不受影响。如果您想创建方程，按照要求按 **[←]** 或 **[↶] [UNDO]** 键。
4. 按 **[ENTER]** 终止方程并在显示屏中对其进行查看。方程自动保存在方程列表中 – 正好在您开始输入时显示的输入项后面。(如果您按的是 **[C]**，方程同样被保存，但是方程模式则被关闭)。

方程可包含变量、数字、矢量、函数及括号表示 – 它们将在以下的话题中进行说明。下例说明了这些元素。

方程中的变量

您可以使用方程中任何一个计算器的变量，从 **A** 至 **Z**，**(I)** 和 **(J)**。这些变量您想用多少次可以用多少次。(有关 **(I)** 和 **(J)** 的详细情况，见第 14 章 – 间接定址变量和标签。)

要想在方程中输入变量，按 **[RCL]** 变量。当您按下 **[RCL]** 时，**A..Z** 信号器向您展示您可以按下一个变量键在方程中输入其名称。

方程中的数字

您可以在方程中输入任何有效数字，包括基数 2、8 和 16、真数、复数以及分数。数字始终用 ALL 显示格式显示，这种格式可显示高达 12 个字符。

要想将一个数字输入到方程中，您可以使用标准数字输入键，包括 ,  和 。切勿使用  进行减法运算。

方程中的函数

您可以将很多 HP 35s 输入到方程中。本章稍后的“方程函数”给出了这些方程的完整列表。另外，附录 G“运算指数”也提供本信息。

当您输入方程时，您输入函数的方式和将其放入代数方程的方式是一样的：

- 在方程中，有些函数通常显示在其自变量之间，如 "+" 和 "÷"。对于这些中缀算符，以相同的顺序将其输入到方程中。
- 其它函数的函数名称后面通常有一个或多个自变量，如 "COS" 和 "LN"。对于这种前缀函数，将其输入函数发生处的方程中 - 您按下的键把一个左括号表示放在函数名称的后面，这样您就可以输入其自变量。
- 如果函数有两个或多个自变量，按   将其分开。

方程中的括号表示

您可以在方程中加入括号表示来控制运算进行的顺序。按 $\boxed{()}$ 插入括号表示 (详情请参阅本章稍后的 “运算符优先级”)。

示例: 输入方程。

输入方程 $r = 2 \times c \times (t - a) + 25$

键:	显示:	说明:
$\boxed{\text{EQN}}$	$V=0.25 \times \pi \times D^2 \times L$	显示方程列表中的最后一个方程。
$\boxed{\text{RCL}} \boxed{R} \boxed{\text{ON}} \boxed{=}$	$R=$	开始一个带变量 R 的新方程。
$\boxed{2}$	$R= 2$	输入数字
$\boxed{\times} \boxed{\text{RCL}} \boxed{C} \boxed{\times}$	$R=2 \times C$	输入中缀算符。
$\boxed{()}$	$R=2 \times C \times ($	输入带左括号表示的前缀函数。
$\boxed{\text{RCL}} \boxed{T} \boxed{-} \boxed{\text{RCL}}$		输入自变量和正确的括号表示。
$\boxed{A} \boxed{>} \boxed{+} \boxed{2} \boxed{5}$	$=2 \times C \times (T - A) + 25$	
$\boxed{\text{ENTER}}$	$R=2 \times C \times (T - A) + 25$	终止方程并对其进行显示。
$\boxed{\text{ON}} \boxed{\text{SHOW}}$	$CK=9E5F$ $LN=14$	显示其校和及长度。
\boxed{C}		退出方程模式。

显示并选择方程

方程列表包含两个内置方程 2×2 lin. solve 和 3×3 lin. 以及您输入的文件。您可以显示方程并选择一个进行计算。

要想显示方程：

1. 按下 **EQN**。这样将激活方程模式并打开 **EQN** 信号器。显示屏显示方程列表的一个输入项：
 - 如果方程指针位于列表的顶部，则为 **EQN LIST TOP**。
 - 当前方程 (您查看的最后一个方程)。
2. 按 **▲** 或 **▼** 滚动通过方程列表并查看每一个方程。列表在顶部和底部“wraps around”。**EQN LIST TOP** 标记列表的顶部。

查看长方程：

1. 如上所示显示方程列表中的方程。如果长度超过 14 个字符，则只显示 14 个字符。➡ 显示右边的字符更多。
2. 按 **→** 在开始时对方程进行编辑，或按 **←** 在结束时对其进行编辑。然后反复按 **←** 或 **→** 使光标一次一个字符移动通过方程。左边或右边有多个字符时，显示 **←** 和 **→**。
3. 按 **↶←** 或 **↷→** 通过显示屏滚动查看第 2 行中的长方程。

要想选择方程：

如上所示显示方程列表中的方程。第 2 行中显示的方程是用于所有方程运算的方程。

示例：查看方程。

查看您输入的最后一个方程。

键：	显示：	说明：
EQN	$R=2 \times C \times (T-A)+25$	显示方程列表中的当前方程。
→	$\underline{R}=2 \times C \times (T-A)+25$	激活方程左边的光标
ENTER ←	$=2 \times C \times (T-A)+25 \underline{\quad}$	激活方程右边的光标
C		退出方程模式。

编辑 / 清除方程

您可以编辑或清楚您正输入的方程。您还可以编辑或清除保存在方程列表中的方程。但是，您不可以编辑或清除两个内置方程： $2*2 \text{ lin. solve}$ 和 $3*3 \text{ lin. solve}$ 。如果您试图在两个内置方程之间放入一个新方程，该新方程将被放在方程 $3*3 \text{ lin. solve}$ 的后面。

要想编辑您正输入的方程：

1. 按 \leftarrow 或 \rightarrow 移动光标，使您能够在光标前插入字符。
2. 移动光标，反复按 \leftarrow 以删除不想要的数字或函数。在方程编辑行空白时按 \leftarrow 将不起作用，但是，在方程行空白时按 ENTER 会将该空白方程行删除。然后，显示屏将在方程列表中显示先前的输入项。
3. 按 ENTER (或 C) 将该方程保存在方程列表中。

要想编辑已保存的方程：

1. 显示需的方程，按 \rightarrow 激活方程开头处的光标或按 \leftarrow 激活方程结尾处的光标。(见上文 -“显示 / 选择方程”。)
2. 方程中的光标处于激活状态时，您就可以像输入新方程一样对方程进行编辑。
3. 按 ENTER (或 C) 将编辑过的方程保存在方程列表中，从而取代先前的版本。

编辑方程时使用菜单：

1. 对方程进行编辑时，选择设置菜单 (如 MODE , \leftarrow DISPLAY 或 \rightarrow CLEAR) 将结束方程的编辑状态。
2. 对方程进行编辑时，如选择插入或查看菜单 (如 L.R. , \leftarrow \bar{x}, \bar{y} , \rightarrow $\text{S.}\sigma$, \rightarrow SUMS , \rightarrow BASE , \leftarrow LOGIC , $\text{R}\downarrow$, \leftarrow MEM 和 \leftarrow CONST), 在插入项目后，方程仍将处于编辑模式。
3. 方程模式下，菜单 \bar{x}, \bar{y} , FLAGS , \rightarrow $\bar{x}^?0$ 被禁用。

要想清除保存的方程：

将方程列表上下滚动，直到所需的方程出现在显示屏的第 2 行中，然后按 \leftarrow 。

要想清除所有保存的方程：

在 EQN 模式下，按 \leftarrow [CLEAR]。选择 $\left[\frac{3}{3}\right]$ (3EQN)，显示 CLR EQN? Y N 菜单。
选择 \leftarrow (Y) [ENTER]。

示例：编辑方程。

去掉先前例子中的方程中的 25。

键：	显示：	说明：
[EQN]	R=2xCx(T-A)+25	在方程列表中显示当前方程。
\leftarrow	=2xCx(T-A)+25_	激活方程结尾处的光标。
\leftarrow \leftarrow \leftarrow	=2xCxCOS(T-A)_	删除数字 25。
[ENTER]	R=2xCx(T-A)	在方程列表中显示被编辑的方程的结尾部分。
[C]		退出方程模式。

方程类型

HP 35s 使用 3 种类型的方程：

- **等式。** 方程包含一个 "=", 左侧包含一个以上的变量。例如， $x^2 + y^2 = r^2$ 就是一个等式。
- **赋值。** 方程包含一个 "=", 左侧包含一个以上的变量。例如， $A = 0.5 \times b \times h$ 就是一个赋值。
- **表达式。** 方程不包含 "="。例如， $x^3 + 1$ 就是一个表达式。

当您用方程进行计算时，您可以使用任何一种方程 – 虽然方程类型会对其评估方法产生影响。当您解题以获得一个未知变量时，您可能会使用等式或赋值。当您对函数进行积分时，您可能会使用表达式。

评估方程

方程最有用的特点之一就是它们可以被评估以产生数值，也正是这一点使您能够用方程对结果进行计算。（此外，这一特点还使您能够解方程并对方程进行积分，如第 7、8 章所述）。

由于许多方程的两边被 "=" 分开，方程的基值即为方程两边数值之差。对于分计算，方程中的 "=" 实际上是作为 "-" 来处理的。该数值表示方程的平衡程度。

HP 35s 有两个键用于对方程进行评估，即 **ENTER** 和 **XEQ**。这两个键动作的不同之处仅在于它们对于赋值方程的评估方法不同：

- **XEQ** 返回方程的数值而不论方程的类型。
- **ENTER** 则仅在方程为赋值类型的方程时返回方程的数值。对于赋值式方程，**ENTER** 仅返回方程右边的数值，并将该数值"存入"左边的变量中 – 将数值保存在变量中。

下表所示为两种评估方程的方法。

方程类型	ENTER 的结果	XEQ 的结果
等式: $g(x) = f(x)$ 示例: $x^2 + y^2 = r^2$	$g(x) - f(x)$ $x^2 + y^2 - r^2$	
赋值: $y = f(x)$ 示例: $A = 0.5 \times b \times h$	$f(x)^*$ $0.5 \times b \times h^*$	$y - f(x)$ $A - 0.5 \times b \times h$
表达式: $f(x)$ 示例: $x^3 + 1$	$f(x)$ $x^3 + 1$	
* 同样把结果存入左边的变量 A 中, 例如。		

对方程进行评估:

1. 显示所需的方程。(见上文 -“显示 / 选择方程”。)
2. 按 **ENTER** 或 **XEQ**。方程提示输入每个所需变量的数值。(如果方程中某个数字的基数与当前基数不同, 计算器自动将结果变更为当前基数)。
3. 对于每一个提示, 输入所需数值:
 - 如果所显示的数值良好, 按 **R/S**。
 - 如果您想要一个不同的数值, 输入该数值并按 **R/S**。(另请参阅本章稍后部分 -“响应方程提示”。)

要想停止计算, 按 **C** 或 **R/S**。第 2 行中显示信息 **INTERRUPTED**。

方程评估不从栈中获取任何数值 - 仅使用方程中的数字和变量值。方程的数值返回到 X 寄存器中。

用 **ENTER** 键进行评估

如果一个方程显示在方程列表中, 你可以按 **ENTER** 对方程进行评估。(如果您正处在输入方程的过程中, 按 **ENTER** 将只会使方程结束, 而不对其进行评估。

- 如果方程为赋值式方程，将只对方程的右边进行评估。将结果返回到 X 寄存器并存入左边的变量，然后在显示屏上查看变量。实际上，**ENTER** 获得左边变量的数值。
- 如果方程为等式或表达式，则对整个方程进行评估，就如同对于 **XEQ** 一样。结果被返回到 X 寄存器中。

示例：评估带 **ENTER** 的方程。

使用本章开头处的方程求出直径 35 毫米，长度为 20 米的管道的体积。

键：	显示：	说明：
EQN	$V=0.25 \times \pi \times D^2 \times L$	显示所需的方程。
(按照要求 ^)		
ENTER	D? 2.5	开始对赋值式方程进行评估，使数值保存在 V 中。方程的右边提示输入变量。D 的当前值为 2.5。
3 5 R/S	L? 16	保存 D，提示输入 L，其当前数值为 16。
2 0 x 1 0 0		以毫米为单位保存 L；以立方毫米为单位保存 V，将结果保存在 V 中，显示 V。
0 ENTER	V=	
R/S	19.242,255,0033	
÷ 1 E 6	19.2423	将立方毫米改为升（但不修改 V）。
ENTER		

使用 **XEQ** 进行评估

如果一个方程显示在方程列表中，你可以按 **XEQ** 对方程进行评估。整个方程将被评估，无论方程类型如何。结果被返回到 X 寄存器中。

示例：评估带 **XEQ** 的方程。

使用前例的结果求出直径变为 35.5 毫米时管道体积的变化量。

键：	显示：	说明：
[EQN]	$V=0.25 \times P \times D^2 \times L$	显示所需的方程。
[XEQ]	V? 19,242,255.0033	开始对方程进行评估以求出其数值。提示所有的变量。
[R/S]	D? 35	保持同一 V，提示 D。
[3] [5] [.] [5]	L?	保存新 D，提示 L。
[R/S]	20,000	
[R/S]	-553,705.7051	保持同一 L；计算方程的数值 - 左右两边的不平衡量。
[÷] [1] [E] [6]	-0.5537	将立方毫米改为升。
[ENTER]		

方程的数值为旧体积 (根据 V) 减去新体积 (用新 D 值计算) - 因此旧体积比所示数量要小。

响应方程提示

当您对方程进行评估时，计算器提示您输入每个所需变量的数值。该提示给出变量名称及当前数值，如 **X?2,5000**。如果未命名的间接变量 (I) 或 (J) 在方程中，计算器将不会提示您输入其数值，原因是保存在未命名的间接变量中的当前数值将自动被使用。(参阅第 14 章)

- 要想保持数字不变，按 **[R/S]** 即可。

- 要想修改数字，输入新数字并按 **R/S**。新数字覆盖 X 寄存器中的旧数值。如果您喜欢的话，您还可以将数字作为分数输入。如果您需要计算某个数字，使用正常键盘计算，然后按 **R/S**。例如，您可以在 RPN 模式下按 2 **ENTER** 5 **yx** **R/S**，或在 ALG 模式下按 2 **yx** 5 **ENTER** **R/S**。在按 **ENTER** 前，表达式将在第 2 行中显示；在按 **ENTER** 后，结果将在第 2 行中显示。
- 要想取消提示，按 **C**。变量的当前数值仍然留在 X 寄存器中并显示在第 2 行的右边。如果您在数位输入过程中按 **C**，则数字将被清 0。再次按 **C** 取消提示。
- 要想显示被提示隐藏的数位，按 **←** **SHOW**。

在 RPN 模式下，每一个提示将变量值放入 X 寄存器并禁用栈提升。如果您在收到提示时输入数字，该数字替换 X 寄存器中的数值。当您按 **R/S** 时，栈提升被启用，使数值保存在栈上。

方程的语法

方程遵循某些决定其评估方法的惯例：

- 运算符是如何相互作用的。
- 那些函数在方程中有效。
- 方程的语法错误是如何进行校验的？

运算符优先级

方程中的运算符按某种使评估具有逻辑性和可预见性的顺序进行处理。

顺序	运算	示例
1	括号表示	(X+1)
2	函数	SIN(X+1)
3	幂 (x^y)	X^3
4	一元减 (\pm)	-A
5	乘除	X*Y, A÷B
6	加减	P+Q, A-B
7	等式	B=C

因此，举例而言，括号表示内的所有运算都是先于括号表示外的运算进行的。

示例：

方程	意义
$A \times B^3 = C$	$a \times (b^3) = c$
$(A \times B)^3 = C$	$(a \times b)^3 = c$
$A + B \div C = 12$	$a + (b/c) = 12$
$(A + B) \div C = 12$	$(a + b) / c = 12$
$\%CHG(T+12, A-6)^2$	$[\%CHG((t + 12), (a - 6))]^2$

方程函数

下表列举了方程中的有效函数。附录 G “运算指数” 也提供本信息。

LN	LOG	EXP	ALOG	SQ	SQRT
INV	IP	FP	RND	ABS	!
SGN	INTG	IDIV	RMDR		
SIN	COS	TAN	ASIN	ACOS	ATAN
SINH	COSH	TANH	ASINH	ACOSH	ATANH
→DEG	→RAD	HMS→	→HMS	%CHG	XROOT
→L	→GAL	→MILE	→KM	nCr	nPr
→KG	→LB	→°C	→°F	→CM	→IN
SEED	ARG	RAND	π		
+	-	×	÷	^	
sx	sy	σx	σy	\bar{x}	\bar{y}
\bar{x}_w	\hat{x}	\hat{y}	r	m	b
n	Σx	Σy	Σx^2	Σy^2	Σxy

为了方便，要求一个或两个自变量的前缀函数在您输入它们时显示一个左括号表示。

要求两个自变量的前缀函数包括 %CHG、XROOT、IDIV、RMDR、nCr 和 nPr。用逗号将两个自变量分开。

- ✓ 在方程中，XROOT 函数以与 RPN 用法相反的顺序获取其自变量。例如， $-8 \boxed{\text{ENTER}} 3 \boxed{\sqrt[3]{\quad}}$ 等同 $\text{XROOT}(3, -8)$ 。
- ✓ 另外两个自变量函数以用于 RPN 的 Y、X 顺序获取其自变量。例如， $28 \boxed{\text{ENTER}} 4 \boxed{\text{nCr}}$ 等同 $\text{nCr}(28, 4)$ 。

对于两个自变量函数，注意第 2 个自变量是否为负。这些都是有效方程：

%CHG(-X,-2)

%CHG(X,(-Y))

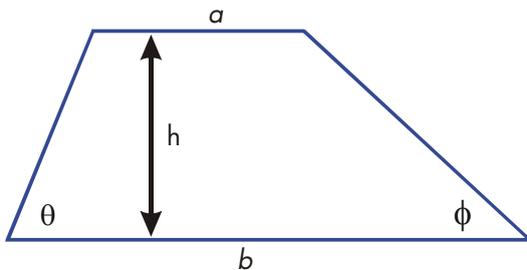
有 8 个方程函数的名称与其等值运算不同：

RPN 运算	方程函数
x^2	SQ
\sqrt{x}	SQRT
e^x	EXP
10^x	ALOG
$1/x$	INV
$\sqrt[y]{x}$	XROOT
y^x	^
INT÷	IDIV

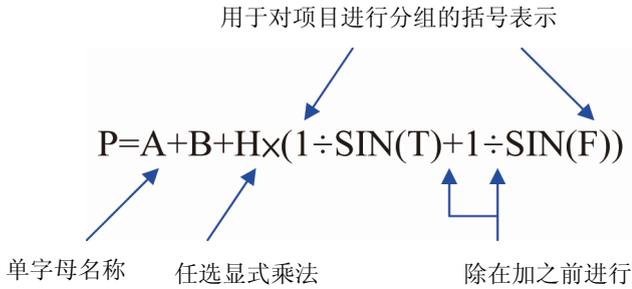
示例：梯形的周长。

以下方程计算梯形的周长。这是方程可能在本书中出现的方式：

$$\text{周长} = a + b + h \left(\frac{1}{\sin \theta} + \frac{1}{\sin \phi} \right)$$



以下方程遵循 HP 35s 的语法规则：



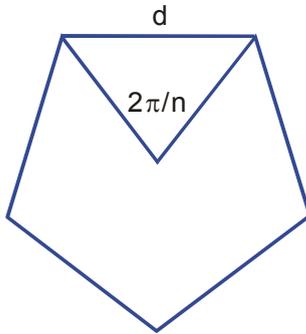
下一个方程同样遵循语法规则。本方程使用逆函数 $\text{INV}(\text{SIN}(T))$ 而非分数形式 $1 \div \text{SIN}(T)$ 。注意 SIN 函数“套在”INV 函数中 (INV 由 $\boxed{1/x}$ 输入。)

$$P=A+B+H \times (\text{INV}(\text{SIN}(T)) + \text{INV}(\text{SIN}(F)))$$

示例：多边形的面积。

正多边形 (带 n 条长度为 d 的边) 的面积方程为：

$$\text{面积} = \frac{1}{4} n d^2 \frac{\cos(\pi/n)}{\sin(\pi/n)}$$



您可以指定本方程为

$$A=0.25 \times N \times D^2 \times \text{COS}(\pi \div N) \div \text{SIN}(\pi \div N)$$

注意运算符和函数是如何共同给出所需的方程的。

您可以用以下键盘键将方程输入到方程列表中：

EQN **RCL** **A** **↵** **=** **·** **2** **5** **×** **RCL** **N** **×** **RCL** **D** **y^x** **2** **×**
COS **↵** **π** **÷** **RCL** **N** **>** **÷** **SIN** **↵** **π** **÷** **RCL** **N** **ENTER**

语法错误

在您对方程进行评估之前，计算器不对方程的语法进行校验。如果某个错误被删除，显示 **SYNTAX ERROR**，光标显示在第一个错误处。您必须对方程进行编辑以纠正错误（见本章先前部分 –“编辑/清除方程”。）

HP 35s 在方程评估前不对方程语法进行校验，使您能够创建实际上可能为信息的“方程”。这一点在程序中特别有用，如第 13 章所述。

对方程进行验证

当您在查看方程时 – 不是您输入方程的时，您可以按 **↵** **SHOW** 显示两样与方程有关的东西：方程的校和及其长度。按住 **SHOW** 键不放，使数值保留在显示屏中。

校和为唯一辨别本方程的 4 位 16 进制数值。如果您错误地输入方程，将不会有本校和。长度为方程所用计算器内存的字节数。

校和及长度允许您验证您输入的方程是否正确。您在例子中输入的方程校和及长度应与本手册中所示的数值吻合。

示例：方程的校和及长度。

求出本章开头处管道体积方程的校和及长度。

键:	显示:	说明:
EQN (按照要求 \wedge)	$V=0.25 \times \pi \times D^2 \times L$	显示所需的方程。
\leftarrow SHOW (保持不放)	CK=49CA LN=14	显示方程的校和及长度。
(释放)	$V=0.25 \times \pi \times D^2 \times L$	重新显示方程。
C		退出方程模式。

解方程

第 6 章讲述了使用 **ENTER** 求出赋值式方程左边变量数值的方法。当然，您也可以使用 SOLVE 求出任何类型的方程中任何变量的数值。

例如，考虑以下方程

$$x^2 - 3y = 10$$

如果您已知本方程中 y 的数值，那么 SOLVE 就可以解出未知数 x 。如果您已知 x 的数值，那么 SOLVE 就可以解出未知数 y 。这种方法对于 "文字题" 也一样有用：

$$\text{加成成本} \times \text{成本} = \text{价格}$$

如果您已知这些变量中的任何两个，那么 SOLVE 就可以计算出第 3 个变量的数值。

当方程只有一个变量或所有的变量 (其中一个除外) 的数值都已知时，求解 x 其实就是求方程的根。等式或赋值式方程精确平衡或表达式方程等于 0 时即存在方程的根。

求根

要想求得未知变量方程的根 (不包括内置方程):

1. 按 **EQN** 显示所需的方程，必要时，按照第 6 章 "把方程输入到方程列表中" 所述输入方程。

2. 先按 **[\rightarrow] SOLVE**，然后按键以获取未知变量。例如，按 **[\rightarrow] SOLVE X** 求解 x 。然后，方程提示输入方程中其它任何一个变量的数值。
3. 对于每一个提示，输入所需数值：
 - 如果所显示的数值正好是您需要的数值，按 **[R/S]**。
 - 如果您想要一个不同的数值，输入或计算该数值并按 **[R/S]**。(详情请见第 6 章“响应方程提示”。)

您可以按下 **[C]** 或 **[R/S]** 终止正在进行的计算。

根被求出后即保存在关系变量中，而变量数值则在显示屏上查看。此外，X 寄存器包含根，Y 寄存器包含先前的估计值或 0，而 Z 寄存器则包含根的数值 D 值(应为 0)。

某些复杂的数学情况下，因为无法求得定解，计算器显示 **NO ROOT FOUND**。见本章稍后部分“验证结果”和附录 D“当 SOLVE 无法求出根时”。

对于某些方程，在解方程之前提供未知变量的一个或两个*初始猜测值*会比较有帮助。这样做可加快计算速度，使答案直指物理解并在适当时求出一个以上的解。见本章稍后部分“选择 Solve 的初始猜测值”。

示例：解线性运动方程。

自由落体的运动方程为：

$$d = v_0 t + 1/2 g t^2$$

式中 d 为距离， v_0 为初始速度， t 为时间而 g 则为重力加速度。

输入方程：

键:	显示:	说明:
\rightarrow CLEAR 3 (3ALL)		清除内存。
\leftarrow (Y) ENTER		
EQN	3*3 lin. solve EQN LIST TOP	选择方程模式。
RCL D \leftarrow = RCL		开始方程。
V \times RCL T +	$D=V \times T +$	
\cdot 5 \times RCL G \times	$\leftarrow =V \times T + 0.5 \times G \times T^2$	
RCL T y^x 2		
ENTER	$D=V \times T + 0.5 \times G \times T^2$	终止方程，显示方程左端。
\leftarrow SHOW	CK=FB3C LN=15	校和及长度。

g (重力加速度) 作为变量包含在方程中，使您能够将其转换为不同的单位 (9.8 m/s^2 或 32.2 ft/s^2)。

计算物体从静止状态在 5 秒之内下落的米数。由于方程模式已经打开且所需的方程已经在显示屏上，您可以开始求解 D 值：

键:	显示:	说明:
\rightarrow SOLVE	SOLVE_	提示输入未知变量。
D	V? value	选择 D ; 提示输入 V 。
0 R/S	T? value	将 0 存入 V ; 提示输入 T 值。
5 R/S	G? value	将 5 存入 T ; 求解 G 值。
9 \cdot 8 R/S	SOLVING D= 122.5000	将 9.8 存入 G ; 求解 D 值。

尝试用同一个方程进行另外一种计算：物体从静止状态下落 500 米需要多长时间呢？

键:	显示:	说明:
EQN	$D = V \times T + 0.5 \times G \times T^2$	显示方程。
→ SOLVE T	D? 122.5	求解 T 值; 提示输入 D 值。
5 0 0 R/S	V? 0	将 500 存入 D; 求解 V 值。
R/S	G? 9.8	0 保留在 V 中; 提示输入 G 值。
R/S	SOLVING T= 10.1015	9.8 保留在 G; 中提示输入 T 值。

示例: 求解理想气体定律方程。

理想气体定律说明了理想气体的压力、体积、温度及数量 (摩尔量) 间的关系:

$$P \times V = N \times R \times T$$

式中, P 为压力 (单位为大气压或 N/m^2), V 为体积 (单位为升), N 为气体的摩尔量, R 为通用气体常数 (0.0821 升-atm/ 摩尔-K 或 8.314 J/ 摩尔-K), 而 T 则为温度 (开氏温度: $K = ^\circ C + 273.1$)。

输入方程:

键:	显示:	说明:
EQN RCL P ×	$P \times _$	选择方程模式, 开始方程。
RCL V ← =		
RCL N ×		
RCL R × RCL T	$P \times V = N \times R \times T _$	
ENTER	$P \times V = N \times R \times T$	终止并显示方程。
← SHOW	CK=EDC8 LN=9	校和及长度。

一个容量为 2 升的瓶内含有 0.005 摩尔温度为 24°C 度的二氧化碳气体。假定该气体的性能和理想气体一样，计算其压力。由于方程模式已经打开且所需的方程已经在显示屏上，您可以开始求解 P 值。

键:	显示:	说明:
\rightarrow SOLVE P	V? 数值	求解 P 值；提示输入 V 值。
2 R/S	N? 数值	将 2 存入 V；求解 N 值。
. 0 0 5 R/S	R? 数值	将 .005 存入 N；求解 R 值。
. 0 8 2 1 R/S	T? 数值	将 .0821 存入 R；求解 T 值。
2 4 + 2 7 3 .	T? 297.1000	计算 T 值(开氏温度)。
1 ENTER	SOLVING	
R/S	P= 0.0610	将 297.1 存入 T，以大气压为单位求解 P 值。

一个容量为 5 升的瓶内含有氮气。温度为 18°C 度时压力为 0.05 大气压。计算气体的密度 ($N \times 28 / V$ ，式中 28 为氮的分子重量)。

键:	显示:	说明:
EQN	$P \times V = N \times R \times T$	显示方程。
\rightarrow SOLVE N	P? 0.0610	求解 N 值；提示输入 P 值。
. 0 5 R/S	V? 2.0000	将 .05 存入 P；求解 V 值。
5 R/S	R? 0.0821	将 5 存入 V；求解 R 值。
R/S	T? 297.1000	保留先前的 R，提示输入 T。
1 8 ENTER 2 7 3	T? 291.1000	计算 T 值(开氏温度)。
. 1 +		

	R/S	SOLVING	将291.1存入T, 求解N值。
		N=	
✓	2 8 x	0.0105	以克为单位计算质量, $N \times 28$ 。
✓	RCL v ÷	0.2929	
		0.0586	以升每克为单位计算密度。

解内置方程

内置方程为: "2*2lin.solve" ($Ax+By=C$, $Dx+Ey=F$) 和 "3*3 lin.

Solve" ($Ax+By+Cz=D$, $Ex+Fy+Gz=H$, $Ix+Jy+Kz=L$)。如果您选择其中一个, **XEQ**, **ENTER** 和 **↵** 键将不起作用。按 **→** **SOLVE** 将要求2*2情况下的6个变量(A至F)或3*3情况下的12个变量(A至L)并用其求出2*2线性方程组的x, y值或3*3线性方程组的x, y和z值。结果将保存在变量x, y, 和z中。计算器将会检测到有无穷多解或无解的情况。

示例: 求解同步方程中的x, y值
$$\begin{cases} x+2y=5 \\ 3x+4y=11 \end{cases}$$

键:	显示:	说明:
EQN	3*3 lin. solve	输入方程模式。
↵	EQN LIST TOP	显示内置方程
→ SOLVE	2*2 lin. solve	
	A?	提示输入A值。
	数值	
1 R/S	B?	将1存入A; 求解B值。
	数值	
2 R/S	C?	将2存入B; 求解C值。
	数值	
5 R/S	D?	将5存入C; 求解D值。
	数值	

3 **R/S**

E?

将 3 存入 D；求解 E 值。

数值

4 **R/S**

F?

将 4 存入 E；求解 F 值。

数值

1 **1** **R/S**

X=



将 11 存入 F，并计算 x 和 y。

1.0000



▼

y=



y 值

2.0000



了解并管理 SOLVE

SOLVE 首先企图直接解方程以获得未知变量。如果这一企图失败，SOLVE 将其改为一个重复程序。该程序从利用未知变量的两个初始猜测值对方程进行评估开始。根据这两个初始猜测值的结果，SOLVE 产生另外一个更佳的猜测值。通过连续的重复，SOLVE 求出使方程的数值等于 0 的未知变量的数值。

当 SOLVE 对方程进行评估时，其评估方式和 **[XEQ]** 一样 – 方程中的任何 “=” 都被当作 “-” 对待。例如，理想气体方程作为 $P \times V - (N \times R \times T)$ 进行评估。这样就确保了等式或赋值式方程的根平衡以及表达式方程的根等于 0。

有些方程较之与其它方程更难解。在有些情况下，您需要输入初始猜测值以便求解。（见下文“选择 SOLVE 的初始猜测值”）。如果 SOLVE 无法求解，计算器会显示 **NO ROOT FND.**

有关 SOLVE 作用方式的详细情况，请参阅附录 D。

验证结果

SOLVE 计算结束后，您可以通过查看栈中左边的数值验证结果是否就是方程的解：

- X 寄存器(按 **[C]** 清除查看的变量)包含未知数的解(根)，即使方程评估为 0 的数值。

- ✓ ■ Y寄存器(按 **[RI]**)包含根的先前估计值或等于0。还数字应与X寄存器中数值一样。如果不一样,则根仅作为*近似值*返回,而X和Y寄存器中的数值给根加上括号。这些加括号的数字应靠在一起。
- ✓ ■ Z寄存器(再次按 **[RI]**)的根处包含方程的D值。对于确切的根,该数值应为0。如果不为0,给出根只是一个*近似值*,但是这一数字应接近0。

如果计算以 **NO ROOT FND** 结束,计算器不能在根上集中。(您可以按 **[C]** 或 **[←]** 清除消息,从而看到X寄存器的数值-根的最终估计值)。X和Y寄存器中的数值给最后被搜索以求根的间隔数加上括号。Z寄存器根的最终估计值处包含方程值。

- 如果X和Y寄存器数值相互不接近,或者Z寄存器的数值不接近0,X寄存器的估计值可能不是根。
- 如果X和Y寄存器数值相互接近,Z寄存器数值接近0,X寄存器的估计值可能是一个近似值。

终止 SOLVE 计算

要想终止计算,按 **[C]** 或 **[R/S]**,消息“**INTERRUPTED**”将出现。根的当前最佳估计值在未知变量中;使用 **[↵] [VIEW]** 对其进行查看而不扰乱栈,但是求解将无法恢复。

选择 SOLVE 的初始猜测值

两个初始猜测值来自:

- 未知变量中当前保存的数字。
- X寄存器中的数字(显示)。

无论您是否输入猜测值，这些来源都用于猜测值。如果您仅输入一个猜测值并将其存入变量中，第 2 个猜测值将为同一数值，因为显示屏也保存有您刚刚在变量中保存的数字 (如果是这种情况，计算器将对其中一个猜测值稍做修改，这样它将拥有两个不同的猜测值)。

输入您自己的猜测值有以下好处：

- 通过缩小搜索的范围，猜测值可缩短求解的时间。
- 如果数学解不只一个，猜测值可将 SOLVE 程序直接指向需要发答案或答案范围。例如，线性运动方程的

$$d = v_0 t + 1/2 g t^2$$

t 值可能有两个解。你可以通过输入近似猜测值将答案指向需要的解。

本例使用本章先前部分的方程，无需您在求解 T 前输入猜测值，原因是该例的头一部分，您保存 T 值并求解 D 值。留在 T 中的数值即为一个好 (物理) 值，因此在求解 T 值时将其用作一个猜测值。

- 如果某个方程不允许未知数的某些数值，猜测值可防止这些数值发生。例如，

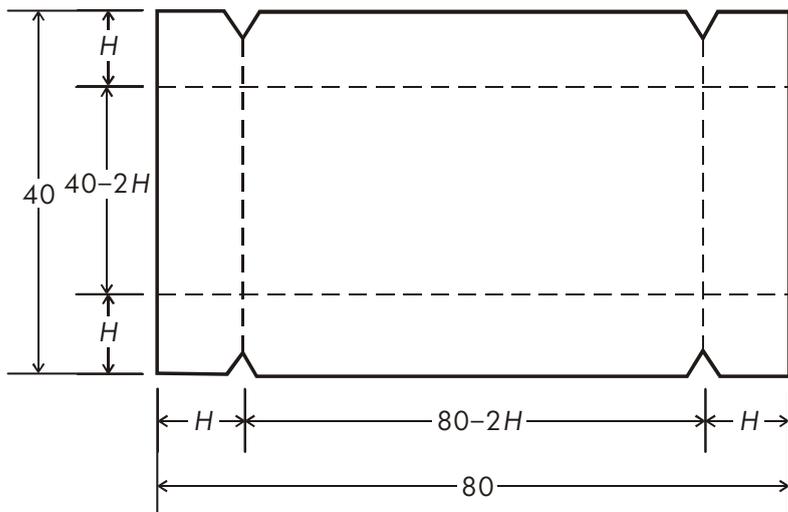
$$y = t + \log x$$

如果 $x \leq 0$ (消息 **NO ROOT FND**) 则导致错误。

在下例中，方程的根不只一个，但是猜测值可帮助您求出所需要的根。

示例：用猜测值求根。

使用一块长为 80 厘米，宽为 40 厘米的矩形板金制作体积为 7500 立方厘米的敞口箱。您需要求得给出指定体积的箱子高度（即沿着 4 条边中的一条的折迭量）。箱子高一些比矮一些好。



如果高度为 H ，箱子的长度为 $(80 - 2H)$ ，宽度为 $(40 - 2H)$ 。箱子体积为 V ：

$$V = (80 - 2H) \times (40 - 2H) \times H$$

您可以将其简化并输入为

$$V = (40 - H) \times (20 - H) \times 4 \times H$$

输入方程：

键：

显示：

说明：

EQN

$V = _$

选择方程模式，开始方程。

RCL **V** **↵** **=**

() **4** **0** **-**

RCL **H** **>**

$V = (40 - H) _$

X () 2 0 -	
RCL H >	$(40-H) \times (20-H)$ _
X 4 X RCL H	$H) \times (20-H) \times 4 \times H$ _
ENTER	$V = (40-H) \times (20-H)$ 终止并显示方程。
↵ SHOW	CK=49A4 校和及长度。 LN=19

高而窄或矮而平的箱子都可制成并达到所需的体积看来是合理的。由于箱子倾向于要高一些，较大的初始估计高度值是合理的。然而，若箱子高度大于 20 厘米，则在实际上是行不通的，原因在于板金的宽度仅为 40 厘米。因此，初始估计高度值在 10 到 20 厘米之间才是恰当的。

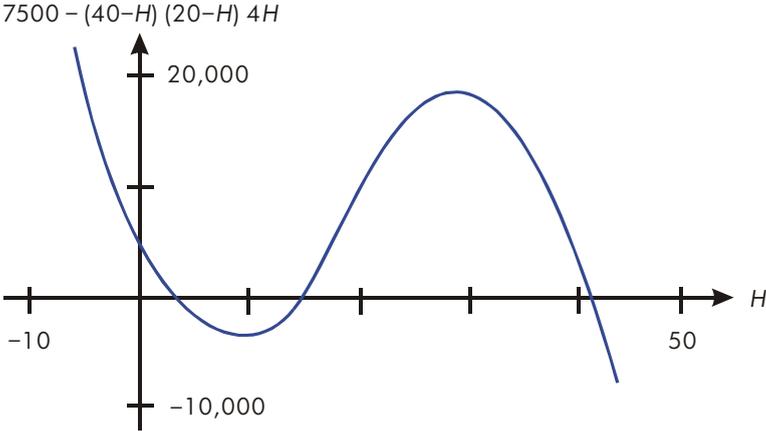
键:	显示:	说明:
C		退出方程模式。
1 0 ↵ STO H		保存上下限猜测值。
ENTER 2 0	20 _	
EQN	$V = (40-H) \times (20-H)$	显示当前方程。
↵ SOLVE H	V?	求解 H 值；提示输入 V 值。
	数值	
7 5 0 0 R/S	H= 15.0000	将 7500 存入 V；求解 H。

现在，通过查看根的先前估计值（在 Y 寄存器中）以及根处的方程值（在 Z 寄存器中）来检查根的质量，即其是否返回一个确切的根。

	键:	显示:	说明:
✓	R↓	15.0000	Y 寄存器中的数值刚好为最终结果前的估计值。由于它和解相同，该解为一个确切的根。
✓	R↓	0.0000	Z 寄存器的数值显示根处方程等于 0。

所需箱子的尺寸为 $50 \times 10 \times 15$ 厘米。如果您忽视高度的上限 (20 厘米) 并使用 30 厘米和 40 厘米的初始高度估计值, 您将得到一个高度 42.0256 厘米 — 典型无意义的一个根。如果您使用较小的初始估计值如 0 和 10 厘米, 您将得到一个高度 2.9774 厘米 — 从而产生一个不需要的矮而平的箱子。

如果您不知道该使用那些猜测值, 您可以使用图形来帮助了解方程的性能。对方程进行评估以获取未知数的多个数值。对于图上的每一点, 显示方程并按 **[XEQ]** — 在收到提示要输入 x 时, 输入 x - 坐标, 然后获取相应的方程数值 y - 坐标。对于上述问题, 您可以始终设置 $V = 7500$ 并改变 H 的数值以得出不同的方程数值。记住本方程的数值为方程左右两边之差。本方程的数值曲线图看起来如下。



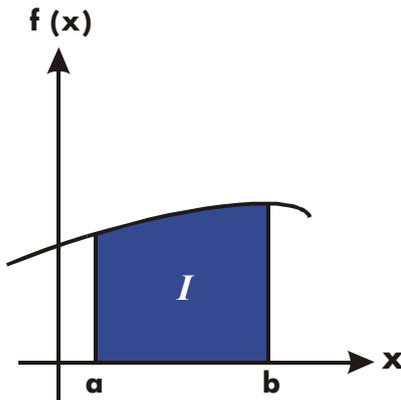
欲知详情

本章向您说明在广泛的应用上求解未知数或根。附录 D 包含更多关于 SOLVE 算法的作用方式, 结果解析方法, 无根时发生的情况以及能够导致错误结果的条件的信息。

对方程进行积分

许多算术、科学和工程问题要求计算函数的定积分。如果函数用 $f(x)$ 表示且积分间隔为 a 至 b ，则积分在数学上可以表示为

$$I = \int_a^b f(x) dx$$



数量 I 在集合上可以解释为与函数 $f(x)$ ， x -轴和极限 $x = a$ 及 $x = b$ 的图形临接的区域的面积 (只要 $f(x)$ 在整个积分间隔为非负)。

\int 运算 (\int FN) 针对特定变量 (\int FN d) 对当前方程进行积分。本方程可以有一个以上的变量。

对方程进行积分 (∫ FN)

对方程进行积分：

1. 如果定义积分函数的方程未保存在方程列表中，键入该方程(见第6章“将方程输入方程列表”)并退出方程模式。本方程通常只包含一个表达式。
- ✓ 2. 输入积分极限：先键入下限，按 **ENTER**，然后键入上限。
3. 显示方程 按 **EQN**，必要时，滚动查看方程列表(按 **▲** 或 **▼**) 显示所需的方程。
4. 选择积分变量：按 **←** **∫** 变量。这样将开始计算。

∫ 比计算器中其它任何运算使用的内存要多得多。如果执行 **∫** 导致出现 **MEMORY FULL** 信息，请参阅附录 B。

您可以按 **C** 或 **R/S** 进行积分运算，第 2 行中将显示信息“**INTERRUPTED**”，但是积分无法恢复。但是，有关积分的信息只有在计算正常结束时才会出现。

显示格式设置会影响为您的函数假定以及结果使用的精确度水准。积分更为精确，但在 **ALL** 中要求更长更高的 **FIX**、**SCI** 和 **ENG** 设置。结果的不确定性在 Y 寄存器中结束，将积分上限推入 T 和 Z 寄存器。详情请见本章稍后部分“积分精度”。

要想对带不同信息的同一方程进行积分：

- ✓ 要想使用相同的积分极限，按 **R↓** **R↓** 将其移入 X 和 Y 寄存器。然后从上表中步骤 3 开始。如果您想使用不同的极限，从第 2 步开始。

要想用不同的方程解另外一道题，以定义积分函数的方程从第 1 步开始。

示例：贝塞尔函数。

第一种数阶 0 的贝塞尔函数可以表示为

$$J_0(x) = \frac{1}{\pi} \int_0^{\pi} \cos(x \sin t) dt$$

解贝塞尔函数以获取 2 和 3 的 x 值。

输入定义积分函数的函数的表达式：

$$\cos(x \sin t)$$

键：	显示：	说明：
CLEAR 3		清除内存。
(3ALL) < (Y) ENTER		
EQN	3*3 lin. solve EQN LIST TOP	选择方程模式。
COS RCL X	COS(X)	方程类型。
X SIN	COS(XxSIN(
RCL T	COS(XxSIN(T)	
> >	COS(XxSIN(T))_	
ENTER	COS(XxSIN(T))	终止表达式，显示其左侧。
↵ SHOW	CK=E1EC LN=13	校和及长度。
C		退出方程模式。

现在，从 0 至 π ； $x = 2$ ，针对 t 对本函数进行积分。

键：	显示：	说明：
MODE 2 (2RAD)		选择半径模式。
0 ENTER ↵ π	3.1416	输入积分极限 (先输入下限)。
EQN	COS(XxSIN(T))	显示函数。
↵ ∫	∫ FN d_	提示输入积分变量。

	T	X?	提示输入 X 值。
	2 R/S	数值 INTEGRATING ∫ = 0.7034	x = 2。开始积分; 计算 $\int_0^{\pi} f(t)$ 的
✓	↶ π ÷	0.2239	结果。 $J_0(2)$ 的最终结果。

现在, 用相同的积积极限计算 $J_0(3)$ 。您必须重新指定积积极限 (0, π), 因为它们被后续 π 的除数推出栈。

键:

显示:

说明:

✓	0 ENTER ↶ π	3.1416	输入积积极限 (先输入下限)。
	EQN	$\text{COS}(X \times \text{SIN}(T))$	显示当前方程。
	↶ ∫	∫ FN d_	提示输入积分变量。
	T	X? 2.0000	提示输入 X 值。
	3 R/S	INTEGRATING ∫ = -0.8170	x=3。开始积分, 计算
✓	↶ π ÷	-0.2601	$\int_0^{\pi} f(t)$ 的结果。 $J_0(3)$ 的最终结果。

✓ **示例:** 正弦积分。

通讯理论中的某些问题 (例如, 通过理想网络的脉冲传输) 要求计算形式积分 (有些时候称为 *正弦积分*)

$$S_i(t) = \int_0^t \left(\frac{\sin x}{x} \right) dx$$

求出 $S_i(2)$ 。

输入定义积分函数的函数的表达式：

$$\frac{\sin x}{x}$$

如果计算器试图以 $x = 0$ ，积分的下限，评估本方程，将导致错误 (**DIVIDE BY 0**)。但是，积分算法通常不以任何一个积积极限对函数进行评估，除非积分间隔的终点彼此非常接近或采样点的数量非常大。

键:	显示:	说明:
EQN	3*3 lin. solve EQN LIST TOP	选择方程模式。
SIN RCL X	SIN(X)_	开始方程。
>	SIN(X)_	在这种情况下需要接近右括号表示。
÷ RCL X	SIN(X)÷X_	
ENTER	SIN(X)÷X	终止方程。
← SHOW	CK=0EE0 LN=8	校和及长度。
C		退出方程模式。

现在从 0 至 2 ($t = 2$) 针对 x (即 X) 对本函数进行积分。

键:	显示:	说明:
MODE 2 (2RAD)		选择半径模式。
0 STO X ENTER	2_	输入积分极限 (先输入下限)。
2		
EQN	SIN(X)÷X	显示当前方程。
← ∫ X	INTEGRATING ∫ = 1.6054	计算 $S_i(2)$ 的结果。

积分精度

由于计算器不能够精确计算积分的数值，只能以*近似值*取而代之。这种近似值的精度取决于用您的方程计算的积分函数的函数本身的精度。这会受计算器中的四舍五入误差及经验常数的精度影响。

带某些特征如最大值或非常快的振动的函数积分，其计算*可能会*不精确，但是这种可能性很低。能够导致问题的一般函数特征及其解决方法在附录 E 中探讨。

指定精度

显示格式的设施 (FIX, SCI, ENG 或 ALL) 决定积分计算的*精度*：所显示的数位越大，所计算积分的精度也越高 (计算积分所需的时间也就越长)。显示的位数越少，计算也越迅速，但是计算器会假定函数精确至指定数位的唯一数字。

要想指定积分的*准确度*，对显示格式进行设置，使显示屏仅显示*积分函数*数值中您认为准确的位数的数量。同一水准的准确度与精密度将反映在积分的结果中。

分数显示模式打开时 (标记 7 已设置)，精度由先前显示格式指定。

解析精度

计算积分后，计算器将该积分的结果的估计*不确定性*放入 Y 寄存器。按 $\boxed{X \leftrightarrow Y}$ 查看不确定性的数值。

例如，如果 $S_i(2)$ 为 1.6054 ± 0.0002 ，则 0.0002 为其不确定性。

示例：指定精度。

显示格式设置为 SCI 2，计算 $S_i(2)$ (取自前例) 表达式中的积分。

键：	显示：	说明：
 DISPLAY  2	1.61E0	设置带两个小数位的科学计数法，指明该函数精确至两个小数位。
(2SCI)  2		
  R↓  R↓	0.00E0 2.00E0	将积分极限由 Z 和 T 寄存器下拉至 X 和 Y 寄存器中。
EQN	SIN(X)÷X	显示当前方程。
  X	INTEGRATING ∫ = 1.61E0	积分近似至两个小数位。
	1.61E-2	积分近似值的不确定性。

积分为 1.61 ± 0.0161 。由于不确定性不会影响第 3 个小数位前的近似值，您可以认为近似值中显示的所有数位都是准确的。

如果近似值的不确定性超出您选择容忍的不确定性，您可以增大显示格式数位的数量并重复进行积分（只要 $f(x)$ 的计算仍然精确至显示屏上显示的数位数量）。一般而言，积分计算的不确定性针对显示格式中指定的每一个附加数位降低至原量的十分之一。

示例：改变准确度。

对于刚刚计算的 $S_i(2)$ 的积分，指定计算结果精确至 4 个而非 2 个小数位。

键:

显示:

说明:

 DISPLAY 

1.6079E-2

指定精确至 4 个小数位。前例的不确定性仍然在显示屏上。

(2SCI) 

0.0000E0

将积分极限由 Z 和 T 寄存器下拉至 X 和 Y 寄存器中。

2.0000E0



SIN(X)÷X

显示当前方程。

  X

INTEGRATING

计算结果。

∫ =

1.6054E0

注意不确定性大约为先前计算的 SCI 2 结果的不确定性的 1/100。

1.6056E-4



DISPLAY  (2SCI) 

0.0002

恢复 FIX4 格式。

MODE  (1DEG)

0.0002

恢复角度模式。

这一不确定性显示结果的正确性可能仅限于 3 个小数位。实际上，与本积分的实际数值相比，这一结果精确至 7 个小数位。由于结果的不确定性计算得比较保守，计算器的近似值在大多数情况下比其显示的不确定性要精确。

欲知详情

本章向您说明在广泛的应用范围内在 HP 35s 上使用积分的方法。附录 E 包含更多关于积分算法的作用方式、可能导致错误结果—延长计算时间的条件以及获取某个积分的当前近似值的详细信息。

带复数的运算

HP 35s 可使用算式中的复数

$$x+iy \quad x+y \cdot i \quad r\theta a$$

该算式包括复算术运算 (+、-、 \times 、 \div)、复三角学 (正弦、余弦和正切) 以及数学函数 $-z$ 、 $1/z$ 、 $z_1^{z_2}$ 、 $\ln z$ 和 e^z 。(式中, z_1 和 z_2 为复数)。

算式 $x+yi$ 仅在 ALG 模式下可用。

要想输入复数:

算式: $x+iy$

1. 输入真数部分。
2. 按下 i 。
3. 输入虚数部分。

算式: $x+y \cdot i$

1. 输入真数部分。
2. 按下 $+$ 。
3. 输入虚数部分。
4. 按下 i 。

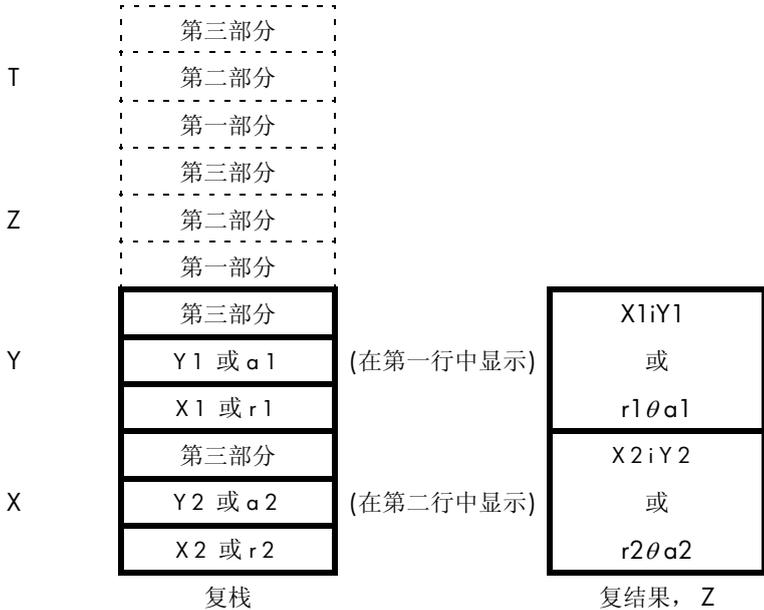
算式: $r\theta a$

1. 输入 r 的数值。
2. 按下 \rightarrow θ 。
3. 输入 θ 的数值。

除非另有说明, 本章中的例子全都采用 RPN 模式。

✓ 复栈

复数占据栈级的第一和第二部分。在 RPN 模式下，占据 X 寄存器第一和第二部分的复数在第二行中显示，而占据 Y 寄存器第一和第二部分的复数在第一行中显示。



复运算

像您在 ALG 和 RPN 模式下进行真实运算一样使用复运算。

✓ 要想进行一个复数的运算：

1. 如上文所述输入复数 z 。
2. 选择复函数。

一个复数 z 的函数

要想计算:	按:
改变符号, $-z$	
倒数, $1/z$	
自然对数, $\ln z$	
自然反对数, e^z	
Sin z	
Cos z	
Tan z	
绝对值, $ABS(z)$	
自变量值, $ARG(z)$	

✓ 要想进行两个复数的算术运算:

1. 如上文所述输入第一个复数 z_1 。
2. 如上文所述输入第二个复数 z_2 。
3. 选择算术运算:

两个复数 z_1 和 z_2 的算术运算

要想计算:	按:
加, $z_1 + z_2$	
减, $z_1 - z_2$	
乘, $z_1 \times z_2$	
除, $z_1 \div z_2$	
幂函数, $z_1^{z_2}$	

示例:

以下给出的是一些复数的三角学及算术运算的例子。

评估 $\sin(2i3)$

键:	显示:	说明:
DISPLAY 9 ($9 \times i \cdot w$)		设置显示格式。
2 i 3 SIN	9.1545 <i>i</i> -4.1689	结果为 9.1545 $i-4.1689$ 。

对表达式进行评估

$$z_1 \div (z_2 + z_3),$$

式中 $z_1 = 23 i 13$, $z_2 = -2i1$ $z_3 = 4 i - 3$

进行计算, 如

键:	显示:	说明:
DISPLAY 9 ($9 \times i \cdot w$)		设置显示格式
2 3 i 1 3 ENTER	23.0000 <i>i</i> 13.0000	ENTER z1
2 +/- i 1 ENTER	23.0000 <i>i</i> 13.0000 -2.0000 <i>i</i> 1.0000	ENTER z2
4 i 3 +/- +	-2.0000 <i>i</i> 1.0000 23.0000 <i>i</i> 13.0000	($z_2 + z_3$)。结果为 $2 i - 2$ 。
÷	2.0000 <i>i</i> -2.0000 23.0000 <i>i</i> 13.0000 2.5000 <i>i</i> 9.0000	$z_1 \div (z_2 + z_3)$ 。结果为 $2.5 i 9$ 。

评估 $(4 i - 2/5) \times (3 i - 2/3)$ 。

键:	显示:	说明:
DISPLAY 9 ($9 \times i \cdot w$)		设置显示格式
4 i . 2 . 5 +/-	4.0000 <i>i</i> -0.4000	输入 $4i-2/5$
ENTER	4.0000 <i>i</i> -0.4000	

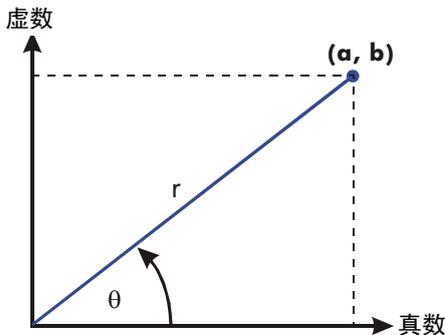
3 i . 2 . 3 +/=	4.0000 <i>i</i> -0.4000	输入 3i-2/3
	3 <i>i</i> -0.2/3	
x	11.7333 <i>i</i> -3.8667	结果为 11.7333i-3.8667

评估 e^{z-2} ，式中 $z = (1i 1)$ 。

键:	显示:	说明:
1 i 1 ENTER	1.0000 <i>i</i> 1.0000	输入 1i1, z^{-2} 的中间结果为 0i-5
2 +/= y^x	0.0000 <i>i</i> -5.0000	
□ e^x	0.8776 <i>i</i> -0.4794	最终结果为 0.8776i-0.4794。

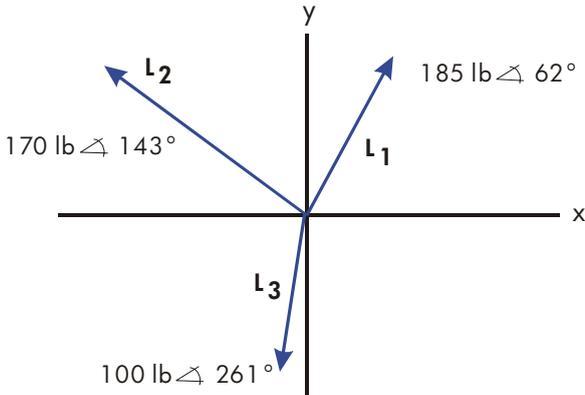
使用极性计数法中的复数

许多应用使用 *极性算式* 或 *极性计数法* 中的真数。这些算式使用成对的数字，复数也是如此，因此，你可以使用复运算用这些数字进行算术运算。



示例： 矢量加。

使以下 3 个载重量相加。



键:	显示:	说明:
MODE 1 (1DEG)		设置角度模式。
DISPLAY 0		设置复模式。
(10rθa)		
1 8 5 ↵ 0	185.0000062.0000	输入 L ₁
6 2 ENTER	185.0000062.0000	
1 7 0 ↵ 0	170.00000143.00...	输入 L ₂
1 4 3 ENTER	170.00000143.000➡	
1 0 0 ↵ 0	185.0000062.0000	输入 L ₃ 并使 L ₂ +L ₃ 相加
2 6 1 +	151.45290178.660➡	
+	178.93720111.148➡	加 L ₁ + L ₂ + L ₃ 。
↵ >	←9	滚屏查看答案的其余部分

您可以用复算式不同的数字进行复运算；但是，结果算式取决于 **DISPLAY** 菜单中的设置。

评估 $1i1+3\theta 10+5\theta 30$

键:	显示:	说明:
MODE 1 (1DEG)		设置角度模式。
← DISPLAY · 0		设置复模式。
(10rθa)		
1 i 1 ENTER	1.4142045.0000	输入 $1i1$
	1.4142045.0000	
3 → θ 1 0	3.0000010.0000	输入 $3\theta 10$
ENTER	3.0000010.0000	
5 → θ 3 0	1.4142045.0000	输入 $5\theta 30$ 并使 $3\theta 10$ 相加
+	7.8861022.5241	
+	9.2088025.8898	加 $1i1$, 结果为 $9.2088\theta 25.8898$

方程中的复数

您可以在方程中输入复数。显示某个方程时，所有的数字算式都如其输入时的状态一样显示，如 xiy 或 $r\theta a$

当您对方程进行评估且收到要求输入变量值的提示时，您可以输入复数。结果的数值和格式由显示设置控制。这和在 ALG 模式下进行计算是一样的。

包含复数的方程可以求解并进行积分。

程序中的复数

在程序中，您可以输入复数，如 $1i2+3010+5$

0 30 在程序中为：

程序行: (ALG 模式)	说明
F001 LBL F	开始程序
F002 1i2+3010+5030	
F003 RTN	

当您运行程序并收到 INPUT 指令要求您输入数值的提示时，您可以输入复数。结果的数值和格式由显示设置控制。

包含复数的程序也可求解并进行积分。

矢量算术

从数学的角度来看，矢量就是两个或多个排列成行或列的元素。

有两个或三个元件的物理矢量可用来表示诸如位置、速度、加速度、力、力矩、线性及角力矩、角速度及加速度等物理量。

要想输入变量：

1. 按  
2. 要想输入矢量的第一个数字。
3. 先按  ，然后输入 2-D 或 3-D 的第二个数字。
4. 先按  ，然后输入 3-D 矢量的第三个数字。

HP35s 不能处理拥有 3 个以上维度的矢量。

矢量运算

✓ 加 / 减：

需要两个操作数的矢量加 / 减的长度相同。如若试图加 / 减长度不同的矢量，将产生错误信息“INVALID DATA”。

1. 输入第一个矢量
2. 输入第二个矢量
3. 按  或 

✓ 计算 $[1.5, 2.2] + [-1.5, 2.2]$

键:	显示:	说明:
MODE 5 (5RPN)		切换至RPN模式(必要时)
▸ □ 1 · 5 ◀	[1.5000,-2.2000]	输入 [1.5,2.2]
, +/- 2 · 2	[1.5000,-2.2000]	
ENTER		
▸ □ +/- 1 · 5	[1.5000,-2.2000]	输入 [-1.5,2.2]
◀ , 2 · 2	[-1.5,2.2]	
+	0.0000	加两个矢量
	[0.0000,0.0000]	

计算 $[-3.4, 4.5] - [2.3, 1.4]$

键:	显示:	说明:
MODE 4 (4ALG)		切换至ALG模式
▸ □ +/- 3 · 4	[-3.4,4.5]	输入 [-3.4,4.5]
◀ , 4 · 5 >		
- ▸ □ 2 · 3	←[-3.4,4.5]-[2.3,1.4]	输入 [2.3,1.4]
◀ , 1 · 4		
ENTER	[-3.4,4.5]-[2.3,1.4]	减两个矢量
	[-5.7000,3.1000]	

✓ 纯量乘除:

1. 输入一个矢量
2. 输入一个标量
3. 按 **×** 进行乘法运算或 **÷** 进行除法运算

✓ 计算 [3,4]x5

键:	显示:	说明:
MODE 5 (5RPN)		切换至 RPN 模式
▢ ▢ 3 ↵ , 4	[3.0000,4.0000]	输入 [3,4]
ENTER	[3.0000,4.0000]	
5	[3.0000,4.0000] 5_	将 5 作为纯量输入
×	0.0000 [15.0000,20.0000]	进行乘法运算

计算 [-2,4]÷2

键:	显示:	说明:
MODE 4 (4ALG)		切换至 ALG 模式
▢ ▢ +/- 2 ↵	[-2,4]_	输入 [-2,4]
, 4 >		
÷ 2	[-2,4]÷2	将 5 作为纯量输入
ENTER	[-2,4]÷2 [-1.0000,2.0000]	进行除法运算

矢量的绝对值

在用于矢量时，绝对值函数 "ABS" 给出矢量的倍数。对于矢量 $A=(A_1, A_2, \dots, A_n)$,

其倍数定义为 $|A| = \sqrt{A_1^2 + A_2^2 + \dots + A_n^2}$ 。

1. 按 **▢** **ABS**
2. 输入一个矢量
3. 按 **ENTER**

例如：矢量 [5,12] 的绝对值：

▢ **ABS** **▢** **▢** **5** **↵** **,** **1** **2** **ENTER**。答案为 13。在 RPN 模式下：
MODE **5** (5RPN) **▢** **▢** **5** **↵** **,** **1** **2** **▢** **ABS**。

点积

函数 DOT 用于计算相同长度的两个矢量的点积。试图计算不同长度的两个矢量的点积将会导致错误信息 “INVALID DATA”。

对于 2-D 矢量 [A, B]、[C, D]，点积定义为 $[A, B] \cdot [C, D] = A \times C + B \times D$ 。

对于 3-D 矢量 [A, B, X]、[C, D, Y]，点积定义为 $[A, B, X] \cdot [C, D, Y] = A \times C + B \times D + X \times Y$

1. 输入第一个矢量
2. 按 **[X]**
3. 输入第二个矢量
4. 按 **[ENTER]**

注意：符号 **[X]** 在这里指的是 “点积” 而非 “叉积”。有关叉积的信息，请参阅第 17 章。

计算两个矢量 [1,2] 和 [3,4] 的点积。

键：	显示：	说明：
[MODE] [4] (4ALG)		切换至 ALG 模式
[F1] [1] [F2] [2]	[1,2]_	输入第一个矢量 [1,2]
[>]		
[X] [F1] [3] [F2] [4]	[1,2]×[3,4]	执行 [X] 以获取点积，输入第
[4]		二个矢量
[ENTER]	11.0000	两个矢量的点积为 11

✓ 计算两个矢量 [9,5] 和 [2,2] 的点积。

键：	显示：	说明：
[MODE] [5] (5RPN)		切换至 RPN 模式
[F1] [9] [F2] [5]	[9.0000,5.0000]	先输入第一个矢量 [9,5]，
[ENTER]	[9.0000,5.0000]	
[F1] [2] [F2] [2]	[9.0000,5.0000] [2,2]	然后输入第二个矢量 [2,2]

[X]

28.0000

按 **[X]** 以获取点积，两个矢量的点积为 28

两矢量夹角

两个矢量 A 和 B 之间的夹角可以求解为 $\theta =$

$$\text{ACOS}(A \cdot B / |A||B|)$$

求两矢量夹角: A=[1,0],B=[0,1]

键:	显示:	说明:
[MODE] [4] (4ALG)		切换至 ALG 模式
[MODE] [1] (1DEG)		设置角度模式
[F2] [ACOS]	ACOS()	弧余弦函数
[F2] [[]] [1] [)] [.] [0]	ACOS([1,0])	输入矢量 A[1,0]
[>]		
[X] [F2] [[]] [0] [)] [.]	ACOS([1,0]×[0,1])	输入矢量 B[0,1] 以获取 A 和 B 的点积
[1] [>]		
[÷] [F2] [ABS] [F2] [[]]	←[.]÷ABS([1,0])→	求出矢量 A[1,0] 的倍数
[1] [)] [.] [0] [>]		
[÷] [F2] [ABS] [F2] [[]]	←[1,0]÷ABS([0,1])→	求出矢量 B [0,1] 的倍数
[0] [)] [.] [1]		
[ENTER]	ACOS([1,0]×[0,1])	两矢量夹角为 90 度
	90.0000	

✓ 求两矢量夹角: A=[3,4],B=[0,5]

键:	显示:	说明:
[MODE] [5] (5RPN)		切换至 RPN 模式
[MODE] [1] (1DEG)		设置角度模式
[F2] [[]] [3] [)] [.] [4]	90	求出两矢量的点积
[ENTER] [F2] [[]] [0] [)]	20.0000	
[.] [5] [X]		

  3   4	20.0000	求出矢量的点积 [3,4]
 ABS	5.0000	
  0   5	5.0000	求出矢量的点积 [0,5]
 ABS	5.0000	
	20.0000	乘两矢量
	25.0000	
	90	除两矢量
	0.8000	
 ACOS	90	两矢量夹角为 36.8699 度
	36.8699	

方程中的矢量

矢量可以和真数一样用在方程及方程变量中。矢量可在提示输入变量时输入。

包含矢量的函数可以求解，但是，如果未知数为矢量的话，则解得结果的能力将有限。

包含矢量的方程可以进行积分，但是，方程的结果必须为一个真矢量，1-D 矢量或以 0 作为第二和第三元素的矢量。

程序中的矢量

矢量可以按与真数和复数一样的方式用在程序中

例如，程序中的 $[5, 6] + 2 \times [7, 8] \times [9, 10]$ 为：

	程序行:	说明:
G0001	LBL G	开始程序
G0002	$[5,6] + 2 \times [7,8] \times [9,10]$	$[5,6]$
G0003	RTN	

矢量可在提示输入变量值时输入。包含矢量的程序可用于求解和积分。

从变量或寄存器创造矢量

可以以运行或程序模式从间接寄存器创造包含内存变量内容、栈寄存器或数值的矢量。

在 ALG 模式下，按 $\boxed{\rightarrow} \boxed{II}$ 开始输入矢量。RPN 模式的作用方式与 ALG 模式类似，除了必须先按下 \boxed{EQN} 键，然后按下 $\boxed{\rightarrow} \boxed{II}$ 外。

要想输入一个包含保存在字母变量中的数值的元素，按 \boxed{RCL} 及变量字母。

要想输入一个来自栈寄存器的数值，按 $\boxed{R\downarrow}$ 键，用 $\boxed{\>}$ 或 $\boxed{\<}$ 键移动下划线符号，使之位于将要使用的栈寄存器下面，然后按 \boxed{ENTER} 。

要想输入一个由 I 或 J 寄存器中的数值间接表示的元素，先按 \boxed{RCL} ，然后按 (I) 或 (J) 两者中任何一个。

例如，要想在 RPN 模式下构建矢量 [C, REGZ, (J)]，先按 $\boxed{EQN} \boxed{\rightarrow} \boxed{II}$ ，然后按 $\boxed{RCL} \boxed{C} \boxed{\leftarrow} \boxed{,} \boxed{R\downarrow} \boxed{\>} \boxed{ENTER} \boxed{\leftarrow} \boxed{,} \boxed{RCL} \boxed{(J)} \boxed{ENTER}$ 。

基底转换、算术运算及逻辑

BASE 菜单 ( **BASE**) 允许您输入数值并强制以十进制、二进制和十六进制基底显示数字。

LOGIC 菜单 ( **LOGIC**) 提供对于逻辑功能的访问。

BASE 菜单

菜单标签	说明
DEC	<i>十进制模式</i> 。这是正常的计算器模式。
HEX	<i>十六进制模式</i> 。这种模式处于激活状态时，显示 HEX 信号器。数字均以十六进制格式显示。在 RPN 模式下，  ,  ,  ,  ,  和  键作为快捷键输入从 A 至 F 的数位。在 ALG 模式下，按  A、B、C、D、E 或 F 输入从 A 至 F 的数位。
OCT	<i>八进制模式</i> 。这种模式处于激活状态时，显示 OCT 信号器。数字均以八进制格式显示。
BIN	<i>二进制模式</i> 。这种模式处于激活状态时显示 BIN 信号器。数字均以二进制格式显示。如果某个数字的数位超过 12 个，   和   键将允许查看整个数字 (见本章稍后部分的 “长二进制数字窗口” 。)
d	位于数字结尾处意味着该数字是一个十进制数字。
h	位于数字的结尾处意味着该数字是一个十六进制数字，要想输入一个十六进制数字，输入 “h” 前的数字。

o	位于数字的结尾处意味着该数字是一个八进制数字。要想输入一个八进制数字，输入“o”前的数字。
b	位于数字的结尾处意味着该数字是一个二进制数字。要想输入一个二进制数字，输入“b”前的数字。

示例：转换数字的基底。

以下键盘键进行各种基底转换。

将 125.99_{10} 转换为十六进制、八进制和二进制数字。

键：	显示：	说明：
1 2 5 ↔ BASE	7Dh	将十进制数字转换为基底 16。
2 (2HEX)		
↔ BASE 3 (3OCT)	175o	基底 8。
↔ BASE 4 (4BIN)	1111101b	基底 2。
↔ BASE 1 (1DEC)	125.0000	

注意：使用非十进制基底时，仅数字的整数部分用于显示。分数部分保留（除非进行清除分数部分的运算）并在选择十进制基底时显示。

将 $24FF_{16}$ 转换为二进制基底。二进制数字的长度将超过 14 位数（最大显示）。

键：	显示：	说明：
↔ BASE 2 (2HEX)	24FFh	使用 U/x 键输入“F”。
2 4 U/x U/x ↔		
BASE 6 (6h)		

 **BASE** **4** (4BIN)

1001001111111111

整个二进制数字不符合。 ➡ 信

 **>**

←b

号器显示数字在右边继续。

显示数字的其余部分。整个数字为 10010011111111b。

 **<**

1001001111111111

➡ 再次显示前 14 位数。

 **BASE** **1** (1DEC)

9,471.0000

恢复基底 10。

您可以使用 **BASE** 菜单输入操作数后面的基底 n 符号 $b/o/d/h$ 以在任何基底模式下表示 2/8/10/16 基数。不带基底符号的数字为十进制数字。

注意:

在 ALG 模式下:

1. 结果的基底模式由当前的基底模式设置决定。
2. 如果无可用的命令行(第一行上无光标闪烁), 改变基底将会把第二行更新到新的基底中。
3. 按 **ENTER** 或改变基底模式后, 计算器将在结果后自动添加一个当前基底符号 $b/o/h$ 以在第二行中表示基底 2/8/16 数字。
4. 要想再次对表达式进行编辑, 按 **<** 或 **>**。

在 RPN 模式下:

当您在第二行中对数字进行编辑时, 先按 **ENTER**, 然后改变基底模式, 计算器将会转换第一行和第二行中的数字基底, 符号 $b/o/h$ 将被添加在数字后面以表示基底 2/8/16。

要想在第二行中查看下一屏幕的内容, 按  **<** 或  **>** 更换屏幕。

LOGIC 菜单

菜单标签	说明
AND	两个自变量的逻辑逐位“与”。 例如：AND (1100b,1010b)=1000b
XOR	两个自变量的逻辑逐位“异或”。 例如：XOR (1101b,1011b)=110b
OR	两个自变量的逻辑逐位“或”。 例如：OR (1100b,1010b)=1110b
NOT	返回自变量的一个补数。结果中的每一位数都是自变量中相应数位的补数。 例如：NOT(1011b)= 1111111111111111111111111111111110100b
NAND	两个自变量的逻辑逐位“非与”。 例如： NAND(1100b,1010b)=1111111111111111111111111111111111111110111b
NOR	两个自变量的逻辑逐位“非或”。 例如：NOR(1100b,1010b)= 1111111111111111111111111111111111111110001b

“与”、“或”、“异或”、“非”、“非与”以及“非或”可用作逻辑函数。分数、复数以及矢量自变量将被看作逻辑函数中的“INVALID DATA”。

基底 2、8 和 16 中的算术运算

您可以用任何基底中的 $\boxed{+}$, $\boxed{-}$, $\boxed{\times}$ 和 $\boxed{\div}$ 进行算术运算。HEX 模式下真正被灭活的功能键仅有 $\boxed{\sqrt{x}}$, $\boxed{e^x}$, $\boxed{\text{LN}}$, $\boxed{y^x}$, $\boxed{1/x}$ 和 $\boxed{\Sigma+}$ 。但是，您应该意识到由于数字的分数部分被去掉，大部分除算术运算之外的运算将不会得出有意义的结果。

基底 2、8 和 16 中的算术运算呈 2 的补数形式并只使用整数：

- 如果某个数字有分数部分，只有整数部分被用于进行算术计算。

■ 计算的结果始终为整数 (任何分数部分都被去掉)。

鉴于转换只改变数字的显示而不改变 X 寄存器中的实际数字,实际上由算术运算改变 X 寄存器中的数字。

如果运算的结果无法以有效的数位表示,显示屏将先显示 **OVERFLOW**, 然后显示可能的最大正数或负数。

示例:

以下为一些十六、八和二进制模式算术运算的例子:

$$12F_{16} + E9A_{16} = ?$$

键:

BASE **2** (2HEX)

显示:

FC9h

说明:

设置基底 16; **HEX** 信号器打开。
结果。

✓ **1** **2** **/x** **BASE** **6**
(6h) **ENTER** **y^x** **9** **SIN**
BASE **6** (6h) **+**

$$7760_8 - 4326_8 = ?$$

BASE **3** (3OCT)

7711o

设置基底 8; **OCT** 信号器打开。将所显示的数字转换为八进制。
结果。

✓ **7** **7** **6** **0** **BASE**
7 (7o) **ENTER** **4** **3** **2**
6 **BASE** **7** (7o) **-**

$$100_8 \div 5_8 = ?$$

✓ **1** **0** **0** **BASE** **7**
(7o) **ENTER** **5** **BASE**
7 (7o) **÷**

14o

结果的整数部分。

$$5A0_{16} + 1001100_2 = ?$$

✓ **BASE** **2** (2HEX) **5**
SIN **0** **BASE** **6**
(6h) **ENTER**

5A0h

设置基底 16; **HEX** 信号器打开。

- ↩ BASE 4 (4BIN) 1111111111111111 ➡ 二进制版本； ➡ 表示更多数位。由于最高位为 1，数字为负。

- ↩ > ◀ 111111111111101 ➡ 滚屏显示数字的其余部分。
- ↩ > ◀ 11011110b 显示最右边的窗口。
- ↩ BASE 1 (1DEC) -546.0000 十进制负数。

数字范围

36 位二进制数字的大小决定了可以十六 (9 位)、八 (12 位) 和二进制基底 (36 位) 表示的数字范围以及能够转换为这些其它基底的十进制 (11 位) 数字的范围。

基底转换的数字范围

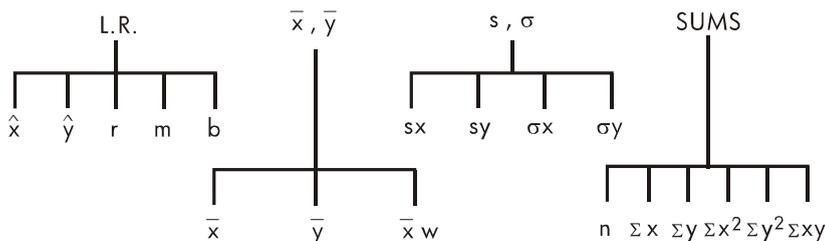
基底	倍数最大的正数	倍数最大的负数
十六进制	7FFFFFFFFh	800000000h
八进制	37777777777o	400000000000o
二进制	01111111111111111111111111111111 1111111111111111b	10000000000000000000000000000000 00000000000000b
十进制	34,359,738,367	-34,359,738,368

选择非十进制基底时，本范围之外的其它数字将无法输入。

统计运算

HP 35s 中的统计菜单提供对一组单 / 双变量数据 (真数) 进行统计分析的功能:

- 平均、样本及总体标准差。
- 线性回归及线性估计 (\hat{x} 和 \hat{y})。
- 加权平均数 (y 加权 x)。
- 总和统计: n , Σx , Σy , Σx^2 , Σy^2 及 Σxy 。



输入统计数据

单 / 双变量统计数据用 $\Sigma+$ (或 $\Sigma-$) 键以类似的方式输入。数据值作为总和统计在 6 个统计寄存器 (-27 至 -32) 累积。这 6 个统计寄存器的名称在 SUMS 菜单中显示 (按 Σ [SUMS], 查看 n Σx Σy Σx^2 Σy^2 Σxy)。

注意



在输入一组新统计数据之前, 务必要对统计寄存器进行清除 (按 Σ [CLEAR] 4 (4 Σ))。

输入单变量数据

1. 按 **[\rightarrow] [CLEAR] [4] (4 Σ)** 清除既有统计数据。
2. 键入各个 x 值, 按 **[$\Sigma+$]**。
3. 显示屏显示 n , 即当前累积的统计数据值的数量。

按 **[$\Sigma+$]** 实际上会把两个变量输入到统计寄存器中, 原因是 Y 寄存器中的既有数值作为 y 值累积。因此, 计算器将进行线性回归并向您显示基于 y 的数值, 即使您只输入 x 数据或即使您输入的 x 和 y 数值的数量不相等。虽然无错误出现, 其结果显然是无意义的。

要想在数字输入后立即将其恢复至显示屏, 按 **[\rightarrow] [LAST x]**。

✓ 输入双变量数据

输入输入为一对变量, 先输入依赖变量 (该变量对中的第二个变量), 按 **[ENTER]**, 然后输入独立变量 (该变量对的第一个变量), 按 **[$\Sigma+$]**。

1. 按 **[\rightarrow] [CLEAR] [4] (4 Σ)** 清除既有统计数据。
2. 先键入 y -值, 然后按 **[ENTER]**。
3. 键入相应的 x -值, 按 **[$\Sigma+$]**。
4. 显示屏显示 n , 即您已经累积的统计数据对的数量。
5. 继续输入 x , y 对, 每输入一项, n 就更新一次。

要想在 x -值输入后立即将其恢复至显示屏, 按 **[\rightarrow] [LAST x]**。

纠正数据输入中的错误

如果您在输入统计数据的时候出错, 删除错误数据并加入正确的数据。即使 x , y 对中只有一个数值是错误的, 您也必须将两个数值都删除然后重新输入。

要想纠正统计数据:

1. 重新输入不正确的数据, 但是不要按 $\Sigma+$, 应按 \leftarrow $\Sigma-$ 。这样将删除数值并使 n 递减。
2. 用 $\Sigma+$ 输入正确的数值。

✓ 如果不正确的数值就是刚刚输入的数值, 按 \leftarrow $LASTx$ 对其进行恢复, 然后按 \leftarrow $\Sigma-$ 将其删除。(不正确的 y 值仍旧在 Y 寄存器中, 其 x 值则保存在 $LAST X$ 寄存器中)。删除不正确的统计数据后, 计算器将在第一行中显示 Y 寄存器的数值并在第二行中显示 n 的数值。

示例:

在左边键入 x, y 数值, 然后如右所示进行纠正:

初始 x, y	纠正后的 x, y
20, 4	20, 5
400, 6	40, 6

键:

显示:

说明:

- | | | |
|--------------------------------------|----------|-------------------------|
| ✓ \leftarrow CLEAR 4 (4Σ) | 4.0000 | 清除既有统计数据。 |
| ✓ 4 ENTER 2 0 $\Sigma+$ | 1.0000 | 输入第一个新数据对。 |
| ✓ 6 ENTER 4 0 0 | 6.0000 | 显示屏显示 n , 您输入的数据对的数量。 |
| $\Sigma+$ | 2.0000 | 对的数量。 |
| \leftarrow LASTx | 6.0000 | 取会最后一个 x 值。最后的 y |
| | 400.0000 | 仍旧在 Y 寄存器中。 |
| \leftarrow $\Sigma-$ | 6.0000 | 删除最后的数据对。 |
| | 1.0000 | |
| ✓ 6 ENTER 4 0 $\Sigma+$ | 6.0000 | 重新输入最后的数据对。 |
| | 2.0000 | |
| ✓ 4 ENTER 2 0 \leftarrow | 4.0000 | 删除最前的数据对。 |
| $\Sigma-$ | 1.0000 | |

示例：平均值 (单变量)。

生产主管 May Kitt 想要确定特定工序所花费的平均时间。她于是随机挑选了 6 个人，在每个人实施该工序的时候对其进行观察并记下所需的时间 (以分钟为单位)：

15.5	9.25	10.0
12.5	12.0	8.5

计算平均时间 (将所有数据作为 x 值对待)。

键:	显示:	说明:
CLEAR 4 (4Σ)		清除统计寄存器。
1 5 . 5 $\Sigma+$	1.0000	输入第一个时间。
9 . 2 5 $\Sigma+$ 1 0		输入其余数据；总共 6 个数据点。
$\Sigma+$ 1 2 . 5 $\Sigma+$ 1	6.0000	
2 $\Sigma+$ 8 . 5 $\Sigma+$		
\bar{x} \bar{y} (\bar{x})	\bar{x} \bar{y} $\bar{x}\bar{y}$ 11.2917	计算完成该工序所需的平均时间。

示例：加权平均值 (双变量)。

某制造公司一年四次购买某部件。去年的采购量为：

部件单价 (x)	\$4.25	\$4.60	\$4.70	\$4.10
部件数 (y)	250	800	900	1000

求出本部件的均价 (针对采购量进行加权)。记得在 x (价格) 之前输入 y ，即加权数 (频率)。

键:	显示:	说明:
CLEAR 4 (4Σ)		清除统计寄存器。
2 5 0 ENTER 4 .		输入数据；显示 n 。
2 5 $\Sigma+$		
8 0 0 ENTER 4 .		
6 $\Sigma+$		
9 0 0 ENTER 4 .	900.0000	
7 $\Sigma+$	3.0000	

✓ **1 0 0 0 ENTER 4** 1,000.0000
1 Σ+ 4.0000
↶ \bar{x} \bar{y} > > (\bar{x}_w) \bar{x} \bar{y} \bar{x}_w
 4.4314

累积 4 个数据对。

计算针对采购量加权的均价。

样本标准差

样本标准差是用来衡量数据值关于平均值的分散度的。样本标准差假定数据是更大、更完整一组数据的采样并以 $n-1$ 作为除数进行计算。

- 按 **↶ σ_x** (σ_x) 获取 x 值的标准差。
- 按 **↶ σ_y >** (σ_y) 获取 y 值的标准差。

本菜单的 (σ_x) 和 (σ_y) 项在下一节“总体标准差”中讲述。

示例： 样本标准差。

使用上文“平均值”例中的同一工序时间，May Kitt 现在想要确定该工序的标准差时间 (s_x)。

15.5	9.25	10.0
12.5	12.0	8.5

计算时间的标准差 (将所有 x 值作为数据对待)。

键：	显示：	说明：
↶ CLEAR 4 (4Σ)		清除统计寄存器。
1 5 . 5 Σ+	1.0000	输入第一个时间。
9 . 2 5 Σ+ 1 0		输入其余数据；输入 6 个数据点。
Σ+ 1 2 . 5 Σ+ 1		
2 Σ+ 8 . 5 Σ+	6.0000	
↶ σ_x (σ_x)	$\Sigma x \Sigma y \delta x \delta y$ 2.5808	计算标准差时间。

总体标准差

总体标准差是用来衡量数据值关于平均值的分散度的。总体标准差假定数据构成一组完整的数据并以 n 作为除数进行计算。

- 按     (σ_x) 以获取 x 值的总体标准差。
- 按      (σ_y) 以获取 y 值的总体标准差。

示例：总体标准差。

Grandma Hinkle 的 4 个儿子已成年，身高分别为 170、173、174 和 180 厘米。求出其 4 个儿子身高的总体标准差。

键：	显示：	说明：
   (4Σ)		清除统计寄存器。
      		输入数据。累积 4 个数据点。
      		
 	4.0000	
    (σ_x)	$\Sigma x \Sigma y \Sigma x^2 \Sigma y^2$	计算总体标准差。
	3.6315	

线性回归

线性回归，LR(也称为*线性估计*)是一种用于求出与一组 x ， y 数据最为相符的直线的统计方法。

注意



为避免 STAT ERROR 信息，在执行 L.R. 菜单中的任何一个函数之前先输入您的数据。

L.R. (线性回归) 菜单

菜单键	说明
\hat{x}	根据计算所得与数据相符合的直线，针对给定假设 y 值估计 (预测) x 值。
\hat{y}	根据计算所得与数据相符合的直线，针对给定假设 x 值估计 (预测) y 值。
r	(x, y) 数据的相关系数。相关系数为 -1 到 $+1$ 之间的一个数字，用以衡量计算所得直线与数据的配合密切度。
m	计算所得直线的曲率。
b	计算所得直线的 y 交点。

- ✔ ■ 要想求出 x (或 y) 的估计值，键入 y (或 x) 的给定假设值，然后按 **◀** **L.R.** **▶** (\hat{x})(或 **◀** **L.R.** **>** (\hat{y}))。
- 要想求出定义与您的数据最相符合直线的数值，先按 **◀** **L.R.**，然后按 r ， m 或 b 。

示例：拟合曲线。

一种新大米品种的产量取决于其氮肥量。对于以下数据，确定其线性关系：相关系数，曲率以及 y -交点。

X , 所施氮肥量 (千克/亩)	0.00	20.00	40.00	60.00	80.00
Y , 谷物产量 (公吨/亩)	4.63	5.78	6.61	7.21	7.78

键：
▶ **CLEAR** **4** (**4Σ**)

显示：

说明：
清除所有先前的统计数据。

✓ **4** **.** **6** **3** **ENTER** **0**
 $\Sigma+$
 ✓ **5** **.** **7** **8** **ENTER** **2**
0 $\Sigma+$ 7.2100
 ✓ **6** **.** **6** **1** **ENTER** **4**
0 $\Sigma+$ 4.0000
 ✓ **7** **.** **2** **1** **ENTER** **6**
0 $\Sigma+$
 ✓ **7** **.** **7** **8** **ENTER** **8** 7.7800
0 $\Sigma+$ 5.0000
 \leftarrow **L.R.** **>** **>** (r) \hat{x} \hat{y} r m b
 0.9880
> \hat{x} \hat{y} r m b
 0.0387
> \hat{x} \hat{y} r m b
 4.8560

输入数据；显示 n 。

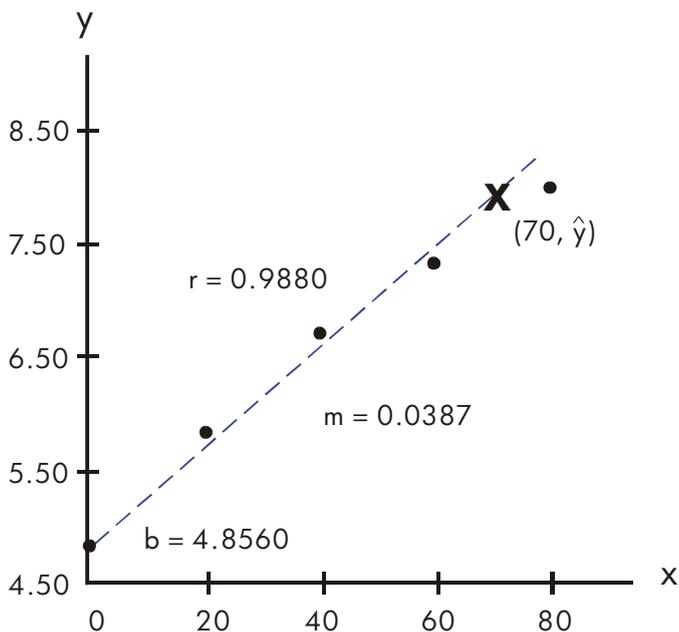
输入的 5 个数据对。

显示线性回归菜单。

相关系数；非常接近直线的数据。

直线斜率。

y 交点。



如果将 70 千克氮肥施加到稻田里情况会是怎样呢？根据上述统计数据计算谷物产量。

键:	显示:	说明:
✓ C 7 0	7.7800	输入假设的 x 值。
	70	
↶ L.R. ➤ (\hat{y})	\hat{x} \hat{y} r m b	预计产量，单位为吨 / 亩。
	7.5615	

数据精确度的限制

由于计算器使用的是有限精确度，因四舍五入而导致的计算限制也就顺理成章了。以下是两个例子：

对相互接近的大数进行规格化

计算器可能无法计算数据值差异量相对较小的变量的标准差和线性回归。为避免出现这种情况，将各个数值作为与中心值（如平均值）的差值输入对数据进行规格化。对于正规 x 值，此差值加回至 \bar{x} ， \hat{x} 和 \hat{y} 的计算，且必须对 b 进行调整。例如，如果你的 x 值为 7776999、7777000 及 7777001，您应该将数据作为 -1, 0 及 1 输入；然后加回 7777000 与 \bar{x} 和 \hat{x} 。对于 b ，则加回 $7777000 \times m$ 。要想计算 \hat{y} ，务必提供一个小于 7777000 的 x 值。

如果您的 x 和 y 值的倍数相差太大，会出现类似的不准确情况。同样，对数据进行规格化可避免这一问题。

已删除数据的影响

执行 **↶** **Σ-** 将不会删除在统计寄存器中可能由于原始数值产生的任何四舍五入误差。除非与正确的数据相比，该错误数据的倍数特别大，否则，该误差将不会很严重。在这种情况下，清除并重新输入所有的数据是比较可取的。

总和数值与统计寄存器

统计寄存器是存储器中 6 个独一无二保存 6 个总和数值的位置。

总和统计

按 **☐** **SUMS**，您将访问统计寄存器的内容：

- 按 **(n)** 恢复累及数据组的数量。
- 按 **>** **(Σx)** 恢复 x 值的总和。
- 按 **>** **>** **(Σy)** 恢复 y 值的总和。
- 按 **>** **>** **>** **(Σx²)**, **>** **>** **>** **>** **(Σy²)** 及 **>** **>** **>** **>** **>** **(Σxy)** 恢复 x 和 y 值的平方总和与积总和，除了计算器提供的计算外，这些数值在进行其它统计计算时也有用。

如果您已经输入统计数据，您将会看到统计寄存器的内容。按 **☐** **MEM** **1** **(1VAR)** **ENTER**，然后使用 **▲** 和 **▼** 查看统计寄存器。

示例： 查看统计寄存器。

用 **Σ+** 在统计寄存器中保存数据对 (1, 2) 和 (3, 4)。然后查看所保存的统计数值。

键：	显示：	说明：
☐ CLEAR 4 (4Σ)		清除统计寄存器。
✓ 2 ENTER 1 Σ+	2.0000	保存第一个数据对 (1, 2)。
	1.0000	
✓ 4 ENTER 3 Σ+	4.0000	保存第二个数据对 (3, 4)。
	2.0000	
☐	n=	↑ 显示 VAR 目录，查看 n 寄存器。
MEM 1 (1VAR)	2.0000	↓

\wedge	$\Sigma xy =$ 14.0000	↑ 查看 Σxy 寄存器。 ↓
\wedge	$\Sigma y^2 =$ 20.0000	↑ 查看 Σy^2 寄存器。 ↓
\wedge	$\Sigma x^2 =$ 10.0000	↑ 查看 Σx^2 寄存器。 ↓
\wedge	$\Sigma y =$ 6.0000	↑ 查看 Σy 寄存器。 ↓
\wedge	$\Sigma x =$ 4.0000	↑ 查看 Σx 寄存器。 ↓
\wedge	$n =$ 2.0000	↑ 查看 n 寄存器。 ↓
C	4.0000 2.0000	退出 VAR 目录。

访问统计寄存器

HP35s中统计寄存器的分布如下表所示。总和寄存器在表达式、方程和程序中应以名称而非数字表示。

统计寄存器

寄存器	数字	说明
n	-27	累积数据对的数量。
Σx	-28	累积 x 值的总和。
Σy	-29	累积 y 值的总和。
Σx^2	-30	累积 x 值的平方总和。
Σy^2	-31	累积 y 值的平方总和。
Σxy	-32	累积 x 和 y 值积和。

您可以通过将您希望的寄存器数字 (-27 至 -32) 存入 *I* 或 *J*, 然后保存总和 (数值 **STO** **(I)** 或 **(J)**) 来载入带总和的统计寄存器。同样地, 您可以按 **↵** **VIEW** **(I)** 或 **(J)** (或 **RCL** **(I)** 或 **(J)**) 查看 (或恢复) 寄存器数值 - 显示屏标上寄存器名称。SUMS 菜单包含恢复寄存器数值的函数。详情请见第 14 章“间接定址变量及标签”。

第二部分

编程

简单编程

本手册第一部分向您介绍了您能够*手动*使用的函数及运算，即按键进行各个运算。而且，您会看到使用方程各重复计算而无需每次都使用所有键盘键的方法。

在第二部分中，您将会了解到使用*程序*进行重复计算的方法 - 这些计算可能会涉及更多的输入或输出或更多复杂的逻辑。程序使您能够完全按照您所希望的方式进行重复运算和计算。

在本章中，您将了解到对一系列运算进行编程的方法。在下一章 "编程方法" 中，您将了解到有关子程序及条件指令的信息。

示例：简单程序。

要想求得半径为 5 的圆的面积，您可以使用

公式 $A = \pi r^2$ ，按

RPN 模式： 5 x^2 \leftarrow π \times

ALG 模式： 5 y^x 2 \times \leftarrow π ENTER

获得本圆形的结果 78.5398。

但是，如果您想获得许多不同圆形的*面积*，情况又会如何呢？

无需每次都重复给定的键盘键 (只针对不同的半径改变 "5")，您只需将可重复的键盘键放入程序即可：

RPN 模式
 0001 $\times 2$
 0002 π
 0003 \times

ALG 模式
 0001 $SQ(x) \times \pi$

这一非常简单的程序假设程序开始运行时半径的数值在 X 寄存器中。它对面积进行计算，并将其留在 X 寄存器中。

在 RPN 模式下，要想将本程序输入到程序存储器中，应采取以下步骤：

键：	显示：	说明：
(在 RPN 模式下)		
CLEAR 3		清除内存。
(3ALL) (Y)		
PRGM		激活程序输入模式 (PRGM 信号器打开)。
	PRGM TOP	重置程序指针到 PRGM TOP。
	0001 $\times 2$	(半径) ²
	0002 π	
	0003 \times	面积 = πx^2
PRGM		退出程序输入模式。

试着运行本程序求出半径为 5 的圆的面积：

键：	显示：	说明：
(在 RPN 模式下)		
		这样将会把程序设置在其开头部分。
5	78.5398	答案！

在 ALG 模式下，要想把本程序输入到程序存储器中，应采取以下步骤：

键：	显示：	说明：
(在 ALG 模式下)		
CLEAR 3		清除内存。
(3ALL) (Y)		

PRGM

激活程序输入模式 (**PRGM** 信号器打开)。

GTO **.** **.**

PRGM TOP

重置程序指针到 **PRGM TOP**。

x² **RCL** **X** **>** **X**

0001 SQ(X)×π 面积 = πx^2

π

PRGM

退出程序输入模式。

试着运行本程序求出半径为 5 的圆的面积：

键：
(在 **ALG** 模式下)

显示：

说明：

GTO **.** **.**

这样将会把程序设置在其开头部分。

5 **STO** **X** **ENTER**

5.0000

将 5 存入 X

5.0000

R/S

78.9358 答案!

我们将继续使用上述程序获取圆的面积对编程概念和方法进行说明。

设计程序

以下话题向您展示您可以将哪些指令放入程序。您在程序中放入的内容将会影响您在对其进行查看时其出现方式以及当您对其进行运行时其运行方式。

选择模式

以 **RPN** 模式创建并保存的程序应以 **RPN** 模式编辑并执行，而以 **ALG** 模式创建并保存的程序或步骤应以 **ALG** 模式编辑并执行。否则，结果会不正确。

程序边界 (LBL 和 RTN)

如果您想在程序存储器中保存一个以上的程序，程序需要一个标签来标记其开头部分 (如 **R001 LBL A**) 以及返回来标记其结尾 (如 **R005 RTN**)。

注意直线数字获取一个 **A** 与其标签相匹配。

程序标签

程序及程序片段 (称为子程序) 应从标签开始。要想对标签进行记录，按：

 **LBL** 字母-键

标签为一个从 **A** 至 **Z** 的单个字母。字母键的使用方式与其用于变量 (如第三章所述) 的方式一样。您不可以超过一次以上对标签进行分配 (这将导致信息 **DUPLICAT.LBL**)，但是标签可以使用变量使用的同一字母。

存储器中个别程序 (最上面的程序) 不带任何标签是可能的。但是，相临程序之间需要一个标签使之相互区隔开来。

程序的行数不得超过 999。

程序返回

程序及子程序应以返回指令结束。键盘键为：

 **RTN**

程序停止运行时，最后的 **RTN** 指令将程序指针返回至 **PRGM TOP**，即程序存储器的顶部。

在程序中使用 RPN，ALG 及方程

您可以按与在键盘键上进行计算一样的方式在程序中进行计算。

- 使用 RPN 运算 (与栈一起使用, 如第二章所述)。
- 使用 ALG 运算 (如附录 C 所述)。
- 使用方程 (如第 6 章所述)。

前例使用一系列的 RPN 运算计算圆的面积。反过来, 您也可以使用程序中的方程。(本章稍后部分有举例)。许多程序实际就是RPN及方程的结合, 利用了两者的优点。

RPN 运算的优点

使用较少的记忆体。
执行起来更快。

方程及 ALG 运算的优点

读写更为容易。
能够自动提示。

当程序执行包含方程的程序行时, 方程按照与 **XEQ** 评估方程列表中的方程相同的方式进行评估。对于程序评估, 方程中的 "=" 实际上是作为 "-" 来处理的。(对于赋值式方程不存在 **ENTER** 的可编程等同物 - 除了将表达式作为方程写入, 然后用 STO 将数值存入变量外)。

对于两种类型的计算, 您可以用 RPN 指令对输入、输出及程序流进行控制。

数据输入及输出

对于需要一个以上输入或返回一个以上输出的程序, 您可以决定让程序输入并返回信息的方式。

对于输入, 您可以用 INPUT 指令提示输入变量数值, 用方程提示输入其变量值或者对事先输入到栈上的数值进行取值。

对于输出，您可以用 VIEW 指令显示变量，显示从方程中获得的信息，在第一行中显示过程，在第二行中显示程序结果或者将未标记的数值留在栈上。

这些都在下一章 " 输入并显示数据 " 中讲述。

输入程序

按  **PRGM** 使计算器进入并退出程序输入模式 - 即打开 / 关闭 **PRGM** 信号器。程序输入模式下的键盘键都作为程序行保存在存储器中。每一个指令 (或命令) 或表达式都占用一个程序行。在 **ALG** 模式下，您可以在程序中直接输入表达式。

将程序输入到存储器中：

1. 按  **PRGM** 激活程序输入模式。
2. 按 **GTO**   显示 **PRGM TOP**。这样将把 *程序指针* 设置到任何其它程序前的一个未知点。输入程序行时，程序行都加在所有其它程序行之前。

如果您不需要任何其它可能会在存储器中的程序，按  **CLEAR** **3** (**3PGM**) 清除程序内存。要想确认您想删除所有的程序，按  (**Y**) **ENTER** 信息后的 **CLR PGMS? Y_N**。

3. 给程序一个 *标签*-从 A 至 Z 的单个字母。按  **LBL** 字母。选择一个可提醒您相关程序的字母，如 "A" 表示 " 面积 "。

如果显示信息 **DUPLICAT.LBL**，使用一个不同的字母。或者，您可以清除既有程序 - 按  **MEM** **2** (**2PGM**)，使用  或  找出标签，然后按  **CLEAR** 和 **C**。

4. 要想将程序指令作为计算器运算加以记录，按与您手动进行运算相同的键。记住许多函数并不在键盘上出现，但必须用菜单存储。要想在程序行中输入方程，请参阅以下说明。

5. 用 **返回** 指令结束程序，在程序运行后将程序指针设置回 **PRGM TOP**。按 **↩** **RTN**。
6. 按 **C** (或 **↵** **PRGM**) 取消程序输入。

程序行中的数字完全和您输入它们时一样保存并用 **ALL** 或 **SCI** 格式显示。(如果一个长数字在显示屏上被截短，按 **↩** **SHOW** 查看所有数位。)

在程序行中输入方程：

1. 按 **EQN** 激活方程输入模式。**EQN** 信号器打开。
2. 想您在方程列表中输入方程一样将方程输入到程序行中。详情请见第六章。使用 **←** 纠正输入错误。
3. 按 **ENTER** 终止方程并显示其左侧。(该方程并不成为方程列表的一部分。)

输入方程后，您可以按 **↩** **SHOW** 查看其校验和及长度。按住 **SHOW** 键不放，使数值保留在显示屏中。

对于长方程，**➡** 和 **⬅** 信号器显示滚屏可用于本程序行。您可以用 **↵** **⬅** 和 **↵** **➡** 使显示屏滚动。

清除函数及后退键

在程序输入期间注意这些特殊条件。

- **C** 始终取消程序输入，从不将数字清 0。
- 在程序行查看状态下，**←** 删除当前程序行，**⬅**/**➡** 开始编辑状态。在程序行编辑状态下，**←** 删除光标前的字符。
- 要想队函数进行编程以删除 X 寄存器，使用 **↵** **CLEAR** **1** (**1x**)。

当您在程序中添加或清除程序行时，必要情况下，**GTO** 及 **XEQ** 陈述自动更新。

例如：

```

A001 LBL A
A002 2+3
A003 1+2
A004
GTO A003

```

现在，清除程序行 A002，程序行 A004 变为 "A003 GTO A002"。

程序中的函数名称

程序行中使用的函数的名称不一定和其键上、菜单中或方程中的函数名称相同。程序中使用的名称通常是比能够与键或菜单相适应的名称要更为完整的缩写。

示例：输入带标签的程序。

以下键盘键删除先前的圆面积方程并输入一个包括标签和返回指令的新方程。如果您在输入时出现错误，先按  删除当前程序行，然后重新输入正确的程序行。

键：	显示：	说明：
(在 RPN 模式下)		
 PRGM		激活程序输入模式 (PRGM 打开)。
 CLEAR 		清除所有的程序内存
(3PGM)  (Y)	PRGM TOP	
ENTER		
 LBL 	A001 LBL A	给本程序调度程序加上标签 A(表示“面积”)。
 x^2	A002 $\times 2$	输入三个程序行。
 π	A003 π	
 \times	A004 \times	
 RTN	A005 RTN	结束程序。
 MEM  (2PGM)	LBL A LN=15	显示标签 A 以及以字节为单位的程序长度。

← SHOW

CK=DAF1
LN=15

程序的校验和及长度。

C C

取消程序输入 (**PRGM** 信号器关闭)。

不同的校验和意味着程序的输入和这里给出的程序不完全一样。

示例： 输入带方程的程序。

以下程序使用方程而不是像前一个程序一样用 RPN 运算计算圆的面积。

键:	显示:	说明:
(在 RPN 模式下)		
→ PRGM GTO ·	PRGM TOP	激活程序输入模式；设置程序指针至存储器顶部。
·		
→ LBL E	E001 LBL E	给本程序调度程序加上标签 E (表示 "方程")。
→ STO R	E002 STO R	将半径存入变量 R。
EQN ← π		选择方程输入模式；输入方程；返回至程序输入模式。
x RCL R		
y^x 2 ENTER	E003 $\pi \times R^2$	
← SHOW	CK=7E5B LN=5	
← RTN	E004 RTN	结束程序。
← MEM 2 (2PGM)	LBL E LN=17	显示标签 E 以及以字节为单位的程序长度。
← SHOW	CK=2073 LN=17	程序的校验和及长度。
C C		取消程序输入。

运行程序

要想运行或执行程序，程序输入不得处于激活状态（无程序行数字显示；**PRGM** 关闭）。按 **C** 取消程序输入模式。

执行程序 (XEQ)

按 **XEQ** 标签执行用该字母标记的程序：

要想从头执行程序，按 **XEQ** 标签 **ENTER**。例如，按 **XEQ** **A** **ENTER**。显示屏将显示“**XEQ A001**”，程序执行将从标签 A 的顶部开始。

您也可以按 **XEQ** 标签行数字，如 **XEQ** **A** **0** **0** **5** 从另外一个位置开始执行程序。

如果存储器中只有一个程序，您还可以在把指针移至程序行的顶部并按 **R/S**（运行/停止）键对程序进行执行。程序运行时，**PRGM** 信号器显示，**B** 信号器打开。

必要时，在执行程序前输入数据。

示例：

运行标有字母 A 和 E 的程序以求得三个半径分别为 5、2.5 和 2π 的不同圆的面积。在执行程序 A 或 E 之前，记得要输入半径。

键：	显示：	说明：
(在 RPN 模式下)		
5 XEQ A ENTER	RUNNING 78.5398	先输入半径，然后开始运行程序 A。显示计算所得出的面积。
2 . 5 XEQ E	19.6350	用程序 E 计算第二个圆的面积。
ENTER		
2 ← π ×		计算第三个圆的面积。
XEQ A ENTER	124.0251	

程序检测

如果您知道程序中有错误，但又不确定错误的位置，那么逐步执行就不愧是一种对程序进行检测的好方式。在使用前对长而复杂的程序进行检测也是一个好主意。通过逐步（一次一行）对程序进行执行，您将可以看到每执行一个程序行后的结果，这样您就可以验证正确结果也已知的数据的进程。

1. 至于例行程序执行，确认程序输入处于非激活状态（PRGM 信号器关闭）。
2. 将程序指针设置到程序的开头处（即，位于其LBL指令处）。该指令移动程序指针而不开始程序执行。
3. 按下并保持 不放。这样将显示当前程序行。当您释放 ，该程序行被执行。然后，显示该执行的结果（在 X-寄存器中）。
要想移动前面的程序行，您可以按 。无程序执行发生。
4. 程序指针移动至下一行。重复步骤三，直到您找出错误（出现错误结果）或到达程序的结尾处。

如果程序输入处于激活状态，那么 或 将只会改变程序指针而不执行程序行。在程序输入期间按下光标键不放将使程序行自动滚动。

示例：程序检测。

逐步执行标记有字母 A 的程序。将半径 5 用于测试数据。开始之前，检查程序输入模式处于 *非*激活状态：

键：	显示：	说明：
(在 RPN 模式下)		
<input type="checkbox"/> 5 <input type="checkbox"/> GTO <input type="checkbox"/> A <input type="checkbox"/> ENTER	5.0000	将程序计数器移动至标签 A。
<input type="checkbox"/> (按下不放) (释放)	A001 LBL A 5.0000	
<input type="checkbox"/> (按下不放) (释放)	A002 $\times 2$ 25.0000	平方输入。

 (按下不放)(释放)	A003 π 3.1416	π 值。
 (按下不放)(释放)	A004 \times 78.5398	25π 。
 (按下不放)(释放)	A005 RTN 78.5398	程序结束。结果正确。

输入并显示数据

计算器的 **变量**用于保存数据输入、中间结果以及最终结果。(变量,如第三章所述,有 A 至 Z 的字母标识,但是变量名称与程序标签毫无关系。)

在程序中,您可以用以下方式获取数据:

- 从 INPUT 指令中获得数据。该指令提示输入变量数值。(这不是最常用的方法。)
- 从栈中获取数据。(您可以用 STO 将数值存入变量供日后使用。)
- 从已经保存有数值的变量中获取数据。
- 从自动方程提示(由已设置的标志 11 启用时)。(如果您使用方程的话,这种方法也比较好。)

在程序中,您可以用以下方式显示信息:

- 使用 VIEW 指令。该指令显示变量的名称和数值。(这不是最常用的方法。)
- 在栈上-只有 X 和 Y 寄存器中的数值可见。(您可以用 PSE 看 X 和 Y 寄存器一秒。)
- 在显示的方程中(由已设置的标志 10 启用时)。(“方程”通常为一条信息,而非真实的方程。)

这些输入和输出方法中有一部分在以下话题中讲述。

使用 INPUT 输入数据

输入指令 ( INPUT 变量) 停止运行程序并显示提示要求输入给定变量。该显示包括变量的当前数值，如

```
R?  
0.0000
```

式中

"R" 为变量的名称，

"?" 信息提示，而

0.0000 则为保存在变量中的当前数值。

按  (运行 / 停止) 恢复程序。您键入的数值覆盖 X 寄存器的内容并存入给定变量。如果您未修改所显示的数值，该数值保留在 X 寄存器中。

带 INPUT 指令的圆面积程序看起来如下：

RPN 模式	ALG 模式
A001 LBL R	A001 LBL R
A002 INPUT R	A002 INPUT R
A003 \times^2	A003 $SQ(R)\times\pi$
A004 π	A004 RTN
A005 \times	
A006 RTN	

要想在程序中使用 INPUT 函数：

1. 决定您需要的数据数值并指定其名称。

(以圆面积为例，唯一需要的输入项为半径，我们将其指定为 R)。

2. 在程序的开头部分，针对每一个您需要其数值的变量插入 INPUT 指令。在程序的稍后部分，当您写如需要给定数值的那部分计算时，插入 **RCL** 变量指令使该数值返回到栈中。

由于 INPUT 指令也将您刚刚输入的数值留在 X 寄存器中，您无需在稍后恢复变量。需要时，您可以输入并使用该数值。这样您大概可以节省一些内存空间。但是，对于比较长的程序，将所有的数据输入在前面，然后在需要时恢复单个的变量会比较简单一些。

另外记住程序使用者在程序停止，等待输入时也可进行计算。这样就可以改变栈的内容，从而会影响程序将要进行的下一个计算。因此，程序不应假设 X、Y 及 Z 寄存器的内容在 INPUT 指令前后会是相同的。如果您在开始时收集所有的数据，然后在需要时将其恢复以进行计算，此举将防止栈的内容刚好在计算前被修改。

响应提示：

当您运行程序时，它将在每一次输入时停止并提示您输入该变量，如 **R?0.0000**。所显示的数值（以及 X 寄存器的内容）将为 R 的当前内容）。

- 要想保持数字不变，按 **R/S** 即可。
- 要想修改数字，输入新数字并按 **R/S**。新数字覆盖 X 寄存器中的旧数值。如果您喜欢的话，您还可以将数字作为分数输入。如果您需要计算某个数字，使用正常键盘计算，然后按 **R/S**。例如，您可以在 RPN 模式下按 **2** **ENTER** **5** **y^x** **R/S**，或在 ALG 模式下按 **2** **y^x** **5** **ENTER** **R/S**（在您按 **ENTER** 之前，表达式在第二行中显示。在您按 **ENTER** 之后，表达式的结果将取代第二行中显示的表达式并保存在 X 寄存器中）。

- 要想取消 **INPUT** 提示, 按 **C**。变量的当前数值仍然在 X 寄存器中。如果您按 **R/S** 恢复程序, 则重复被取消的 **INPUT** 提示。如果您在数位输入过程中按 **C**, 则数字将被清 0。再次按 **C** 取消 **INPUT** 提示。

用 **VIEW** 显示数据

编程后的 **VIEW** 指令 (**VIEW** 变量) 使正在运行的程序停止, 并辨别给定变量的内容, 如

```
R=
78.5398
```

这*只是显示*, 并不把数字复制到 X 寄存器。如果分数显示模式处于激活状态, 数值作为分数显示。

- 按 **ENTER** 把本数字复制到 X 寄存器。
- 如果本数字的宽度超过 14 个字符, 如二进制数字、复数及矢量数字, 按 **→** **←** 和 **→** **→** 显示其余部分。
- 按 **C** (或 **←**) 清除 **VIEW** 显示并显示 X 寄存器。
- 按 **→** **CLEAR** 清除所显示变量的内容。

按 **R/S** 使程序继续。

如果您不希望程序停止, 请参阅下文 " 显示信息而不停止程序 "。

例如, 参见第 16 章 " 正态及逆 / 正态分布 " 的程序。位于 T 子程序结尾处的程序行 T015 及 T016 显示 X 的结果。另外注意, 本程序的这一 **VIEW** 指令前有一个 **RCL** 指令。 **RCL** 指令没有必要, 但是比较方便, 因为它使 **VIEWed** 变量返回到 X 寄存器中, 从而可用于手动计算。(在查看 **VIEW** 显示时按 **ENTER** 将产生同样的效果)。第 16 和 17 章中的其它应用程序也可确保 **VIEWed** 变量在 X 寄存器中。

使用方程显示信息

方程在评估前无法检查其语法有效与否。这就意味着您可以将几乎任何*顺序*的字符作为方程输入到方程中，就如同您输入几乎*任何*方程一样。在任何程序行上，按 **EQN** 开始方程。按数字及数学键获取数字及符号。按每个字母前的 **RCL**。按 **ENTER** 结束程序。

如果标志 10 已设置，则方程被*显示*而非被*评估*。这就意味着您可以将任何您输入的信息作为方程显示。(标志在第 14 章中详细探讨)。

显示信息时，程序停止 - 按 **R/S** 恢复执行。如果所显示信息的长度超过 14 个字符，显示信息时  信号器打开。然后，您可以使用   和   使显示屏滚动。

如果您不希望程序停止，请参阅下文 "显示信息而不停止程序"。

示例：程序中的 **INPUT**、**VIEW** 及信息。

写入一个方程求给定半径及高度的圆柱的表面面积及体积。将程序标记为 **C** (表示圆柱) 并使用变量 **S** (表面面积)，**V** (体积)，**R** (半径)，及 **H** (高度)。使用以下公式

$$V = \pi R^2 H$$

$$S = 2\pi R^2 + 2\pi RH = 2\pi R (R + H)$$

键：	显示：	说明：
(在 RPN 模式下)  PRGM 		程序输入，清除程序内存。
CLEAR 3 (3PGM)	PRGM TOP	
 <Y> ENTER		
 LBL C	C001 LBL C	给程序加上标记。
 INPUT R	C002 INPUT R	
 INPUT H	C003 INPUT H	提示输入半径及高度的指令。

键：
(在 RPN 模式下)

EQN \leftarrow π \times
 RCL R \int^x 2 \times
 RCL H ENTER

\leftarrow SHOW

STO V
 EQN 2 \times \leftarrow
 π \times RCL R \times
 () RCL R +
 RCL H ENTER

\leftarrow SHOW

STO S
 \leftarrow FLAGS 1
 (1SF) \cdot 0
 EQN RCL V
 RCL O RCL L
 \leftarrow SPACE + \leftarrow
 SPACE RCL A
 RCL R RCL E
 RCL A ENTER

\leftarrow FLAGS 1

(2CF) \cdot 0
 \leftarrow VIEW V
 \leftarrow VIEW S
 \leftarrow RTN
 \leftarrow MEM 2

(2PGM)
 \leftarrow SHOW

C C

显示：

C004 $\pi \times R^2 \times H$
 CK=74FE
 LN=7
 C005 STO V

C006 $2 \times \pi \times R \times (R + H)$
 CK=19B3
 LN=11
 C007 STO S

C008 SF 10

C009 VOL + ARE \rightarrow

C010 CF 10
 C011 VIEW V
 C012 VIEW S
 C013 RTN

LBL C
 LN=67
 CK=97C3
 LN=67

说明：

计算体积。

方程的校验和及长度。

将体积保存在 V 中。
 计算表面面积。

方程的校验和及长度。

将表面面积保存在 S 中。
 设置标志 10 显示方程。

显示方程中的信息。

清除标志 10。

显示体积。

显示表面面积。

结束程序。

显示标签 C 以及以字节为单位的
 程序长度。

程序的校验和及长度。

取消程序输入。

现在，求半径为 $2\frac{1}{2}$ 厘米，高度为 8 厘米的圆柱的体积及表面面积。

键:	显示:	说明:
(在 RPN 模式下)		
XEQ C ENTER	R?	开始执行 C；提示输入 R(显示 R 中的任何数值)。
2 . 1 . 2	value	
2 . 1 . 2	H?	将 $2\frac{1}{2}$ 作为分数输入。提示输入 H。
R/S	value	
8 R/S	VOL + AREA	所显示的信息。
R/S	V=	体积，单位为立方厘米。
	157.0796	
R/S	S=	表面面积，单位为平方厘米。
	164.9336	

显示信息而不停止程序

通常，程序会在显示带 VIEW 的变量或方程信息时停止。您通常必须按 **R/S** 恢复执行。

必要时，您可以在显示信息时使程序继续。如果下一个程序行 - 在 VIEW 指令或查看的方程后 - 包含一个 PSE(暂停)指令，显示将信息从程序执行在经历一秒钟的暂停后继续。在这种情况下，不允许滚动显示屏或键盘输入。

显示被其它显示运算清除，在标志 7 被设置时由 RND 运算清除(四舍五入为分数)。

按 **↵** **PSE** 在程序中输入 PSE。

VIEW 及 PSE 程序行 - 或方程及 PSE 程序行在您一次一个程序行执行程序时作为单个运算处理。

停止或终止程序

对停止或暂停 (STOP, PSE) 进行编程

- 在程序输入期间按 **R/S** (运行/停止) 插入一个 STOP 指令。这样将显示 X 寄存器的内容并停止程序运行，直到您按键盘上的 **R/S** 将其恢复。您可以用 STOP 而非 RTN 以使程序结束而不将程序指针返回至存储器的顶部。
- 在程序输入期间按 **▶** **PSE** 插入一个 PSE (暂停) 指令。这样将暂停正在运行的程序并显示 X 寄存器的内容约一秒钟 – 以下情况除外。如果 PSE 仅随 VIEW 指令或显示的方程 (标志 10 已设置) 之后，则显示变量或方程 – 显示在经历 1 秒钟的暂停之后继续。

终止正在运行的程序

您可以随时按 **C** 或 **R/S** 终止正在运行的程序。程序在停止前完成其当前指令。按 **R/S** (运行/停止) 恢复程序。

如果您先终止程序，然后按 **XEQ**, **GTO** 或 **◀** **RTN** 您就不能恢复带 **R/S** 的程序。重新执行程序 (**XEQ** 标签行数字)。

错误停止

如果程序运行过程中出现错误，程序执行停止，显示屏上出现错误信息。(附录 F 提供有信息及条件的列表。)

要想查看包含致错指令的程序中的行，按 **▶** **PRGM**。程序将在该点停止。(例如，它可能是一个 \div 指令，致使非法除以 0)。

对程序进行编辑

您可以通过插入、删除及编辑程序行对程序内存中的程序进行修改。如果一个程序行中包含方程，您可以对方程进行编辑。

要想输出程序行：

1. 选择相关的程序或子程序，按  或  确定必须修改的程序行的位置。按下光标键不放使显示屏继续滚动。
2. 删除您想要修改的程序行—直接按  (取消函数处于激活状态)。然后，指针移至 *前* 一行。(如果您想删除一个以上的连续程序行，从该组中 *最后* 一行开始)。
3. 键入新指令 (如有)。这样将替换您删除的指令。
4. 退出程序输入 ( 或  **PRGM**)。

要想插入一个程序行：

1. 找出并显示您想要插入程序行的点之 *前* 的程序行。
2. 键入新指令；新指令插入在当前显示的程序行 *后*。

例如，如果您想要在程序的程序行 A004 和 A005 之间插入一个新程序行，您需要先显示程序行 A004，然后键入一个或多个指令。随后的程序行，从原始程序行 A005 开始，均向下移动并重新相应进行编号。

要想对程序行中的操作数、表达式或方程进行编辑：

1. 找出并显示您想要编辑的程序行。
2. 按  或  开始对程序行进行编辑。这样将打开“_”编辑光标，但不删除程序行中的任何内容。
 -  键激活程序行左边的光标
 -  键激活程序行结束的光标
3. 移动光标“_”并反复按  删除不需要的数字或函数，然后重新输入程序行的其余部分。(按  后，取消功能处于激活状态)

注意:

1. 程序行中的光标处于激活状态时， 或  键将被禁用。
2. 当您程序行进行编辑(光标处于激活状态)且程序行为空时，使用  将不起作用。如果您想清除程序行，按 ，则程序行将被清除。
3. 您可以用   和   键查看长程序行而不对其进行编辑。
4. 在 ALG 模式下， 不可以用作函数，但是可用于对程序行进行验证。
5. 无论其输入模式如何，方程都可在任何模式下进行编辑。

程序内存

查看程序内存

按   使计算器进入并退出程序输入 (PRGM 信号器打开，显示程序行)。当程序输入模式处于激活模式时，显示程序内存的内容。

程序内存从 **PRGM TOP** 开始。程序行的列表是环行的，因此，您可以从底部向顶部移动程序指针并按相反方向移动。当程序输入处于激活状态时，有四种方法可改变程序指针 (所显示的程序行):

-   和   允许您按标签逐个移动。如果未定义标签，将移至程序的底部或顶部。
- 要想一次移动不只一行 ("滚动")，继续按住  或  不放。
- 按    将程序指针移动至 **PRGM TOP**。
- 按   标签 *nnn* 移动至特定的程序行。

如果程序输入模式处于非激活状态 (如果无程序行显示), 您也可以按 **GTO** 标签行数字移动程序指针。

取消程序输入模式并不会改变程序指针的位置。

内存使用

如果在程序输入期间您遇到信息 **MEMORY FULL**, 则意味着程序内存中无足够的空间可用于您刚刚试图输入的程序行, 您可以通过清除程序和其它数据以提供更多可用空间。见下文 " 清除一个或多个程序 " 或附录 B 的 " 管理计算器内存 "。

程序目录 (MEM)

程序目录是所有带所用内存字节数以及与之相关的程序行的程序标签的列表。按 **MEM** **2** (2PGM) 显示程序目录, 按 **↓** 或 **↑** 在列表内移动。您可以用本目录:

- 回顾程序内存中的标签以及各个加有标签的程序或子程序的内存成本。
- 执行加有标签的程序 (显示标签时按 **XEQ** 或 **R/S**)。
- 显示加有标签的程序。 (显示标签时按 **PRGM**)。
- 删除特定的程序。 (显示标签时按 **CLEAR**)。
- 查看与给定程序片段相关的校验和 (按 **SHOW**)。

目录向您展示各个加有标签的程序片段所用内存的字节数。程序通过程序标签加以辨识:

```
LBL C  
LN=67
```

式中, 67 为程序所用内存的字节数。

清除一个或多个程序

要想从内存中清除特定程序

1. 按 **◀** **MEM** **2** (**2PGM**) **ENTER** 并显示 (使用 **▼** 和 **▲**) 程序的标签。
2. 按下 **▶** **CLEAR**。
3. 按 **C** 取消目录或 **◀** 退出。

要想从内存中清除所有的程序：

1. 按 **▶** **PRGM** 显示程序行 (**PRGM** 信号器打开)。
2. 按 **▶** **CLEAR** **3** (**3PGM**) 清除程序内存。
3. 信息 **CLR PGMS? Y N** 提示您确认。按 **◀** (**Y**) **ENTER**。
4. 按 **▶** **PRGM** 取消程序输入。

清除所有的内存 (**▶** **CLEAR** **3** (**3ALL**)) 也将清除所有的程序。

校验

校验和是给各个程序标签极其相关程序行 (直到下一个标签) 的一个独一无二的十六进制数值。该数字用于和您在程序内存中键入的某个现有程序的已知校验和进行比较。如果已知校验和与您计算器显示的校验和是一致的,则表示您已经正确地输入所有的程序行。要想查看您的校验和:

1. 按 **◀** **MEM** **2** (**2PGM**) **ENTER** 获取程序标签的目录。
2. 必要时,用光标键显示相应的标签。
3. 按下并保持 **◀** **SHOW** 不放以显示 **CK= 校验和**及 **LN= 长度**。

例如，想要查看当前程序的校验和 (" 圆柱 " 程序)：

键：	显示：	说明：
(在 RPN 模式下)		
MEM 2	LBL C	显示标签 C。该标签所用内存 字节数为 67。
2PGM ENTER	LN=67	
SHOW	CK=97C3	校和及长度。
(按下不放)	LN=67	

如果您的校验和与本数字不相吻合，则意味着您未正确地输入本程序。

您将会看到第 16 和 17 章中提供的所有应用程序都包括带标签子程序的校验和数值，使您能够验证您程序输入的准确度。

除此之外，程序中的每一个方程都有一个校验和。见本章稍前部分的 " 在程序行中输入方程 "。

非可编程函数

以下 HP 35s 函数是*不可以*编程的：

CLEAR 3 (3PGM)	
CLEAR 3 (3ALL)	标签行数字
, , ,	
, ,	
	6 (6CLVAR:×)

用 BASE 进行编程

您可以用  **BASE** 对指令进行编程以修改基底模式。这些设置在程序中的作用方式与其作为函数在键盘上执行一样。这就允许您写入接受四中基底之任何一种的数字的程序，以任何基底进行算术运算并以任何基底显示结果。

在写入使用 10 之外的其它基底的数字的程序时，设置基底模式作为计算器的当前设置，并在程序中将其设置为指令。

在程序中选择基底模式

将 BIN、OCT 或 HEX 指令插入到程序的开头处。通常您应该在程序的结尾处加入一个 DEC 指令，以便程序结束时计算器的设置可恢复至十进制模式。

程序中修改基底模式的指令通常可决定 *程序执行期间及之后* 输入的解析方式及输出的情形，但在您输入程序行时不影响程序行。

程序行中输入的数字

在开始程序输入之前，应先设置基底模式。基底模式的当前设置决定程序的结果。

某个信号器将会告诉您哪一种基底是当前设置。比较以下十进制和非十进制模式程序行。所有十进制和非十进制数字在计算器的显示屏上都左对齐。

十进制模式已设置：

```
      :  
      :  
PRGM  
A009 BIN  
A010 10  
      :  
      :
```

二进制模式已设置：

```
      :  
      :  
PRGM BIN  
A009 BIN  
A010 10b  
      :  
      :
```

十进制数字可省略
基底符号 "d"。

二进制数字应加
上基底符号 "b"

多项式表达式及霍纳法

有些表达式，如多项式表达式，将同一变量多次用于其解。例如，表达式

$$Ax^4 + Bx^3 + Cx^2 + Dx + E$$

使用四次变量 x 。用 RPN 运算计算该表达式的程序可以从变量中反复恢复保存的 x 。

示例：

用 RPN 运算写入一个程序计算 $5x^4 + 2x^3$ ，然后以 $x=7$ 对其进行评估。

键:	显示:	说明:
(在 RPN 模式下)		
PRGM GTO		
. .	PRGM TOP	
PRGM LBL A	A001 LBL A	
INPUT X	A002 INPUT X	
5	A003 5	5
RCL X	A004 RCL X	
4	A005 4	
y^x	A006 y^x	x^4
X	A007 \times	$5x^4$
RCL X	A008 RCL X	
3	A009 3	
y^x	A010 y^x	x^3
2	A011 2	
X	A012 \times	$2x^3$
+	A013 +	$5x^4 + 2x^3$
RTN	A014 RTN	
MEM 2	LBL A	显示标签 A。该标签所用内存字节数为 46。
(2PGM)	LN=46	校和及长度。
SHOW	CK=EA18	
	LN=46	
C C		取消程序输入。

现在，以 $x=7$ 对本多项式进行评估。

键:	显示:	说明:
(在 RPN 模式下)		
XEQ A ENTER	X?	提示输入 x。
	数值	
7 R/S	12,691.0000	结果。

对于任何方程这一程序的一种比较普遍的形式

$Ax^4 + Bx^3 + Cx^2 + Dx + E$ 为:

```
A001 LBL A
A002 INPUT A
A003 INPUT B
A004 INPUT C
A005 INPUT D
A006 INPUT E
A007 INPUT X
A008 RCL X
A009 RCL× A
A010 RCL+ B
A011 RCL× X
A012 RCL+ C
A013 RCL× X
A014 RCL+ D
A015 RCL× X
A016 RCL+ E
A017 RTN
```

校和及长度: 9E5E 51

编程方法

第 13 章讲述了编程的基础知识。本章将探讨一些更为复杂但也更有用的编程方法

- 通过分离并给专门用于特定任务的那部分程序加上标记用子程序对程序进行简化。子程序的使用还可缩短必须不只一次执行一系列步骤的程序。
- 使用条件指令 (比较和标志) 确定应使用哪些指令或子程序。
- 使用带计数器的回路执行一系列指令一定次数。
- 使用间接定址访问使用相同程序指令的不同变量。

程序中的子程序

程序是由一个或多个 *子程序* 组成的。子程序是实现特定任务的功能单元。复杂的程序需要子程序将任务分组并分开。这样就使程序更易写、读、理解并修改。

子程序通常以标签开始并以使程序 / 子程序执行结束的指令如 RTN 或 STOP 结尾。

调用子程序 (XEQ, RTN)

子程序 就是一个从其它程序 *调用* (或由其它程序执行) 的程序并在子程序结束时 *返回* 至同一程序。

- 如果您打算让计算器内存中只有一个程序，您可以不同的标签将子程序分开。如果您打算让计算器内存中有不只一个程序，最好让子程序成为主程序标签的一部分，从特定的行数字开始。
- 子程序本身可以调用其它子程序。

本章中的流程图使用以下记数法：

A005 GTO B001 → ① 程序执行从本行分路至其它标记为 ← ① ("从 1") 的行数字。

B001 LBL B ← ① 程序执行从其它标记为 → ① ("至 1") 的行数字分路至本行。

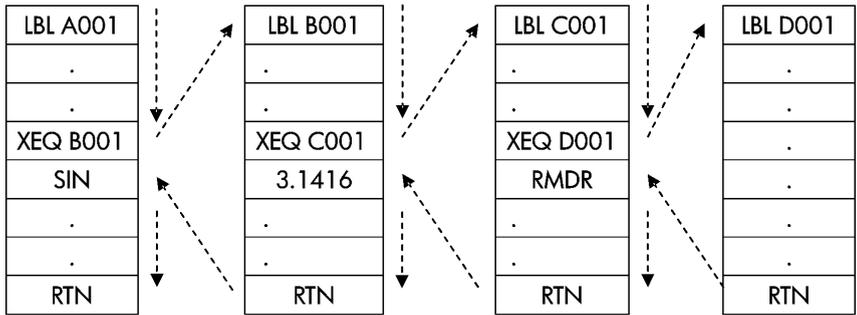
下例向您展示调用子程序修改您输入的的数字的符号。行 **D003 XEQ E001** 从子程序 D 调用的子程序 E 改变数字的符号。子程序 E 以 RTN 指令结束，该指令将程序执行发回至位于程序行 D004 处的子程序 D(保存并显示结果)。见以下流程图。

D001 LBL D		从这里开始。
D002 INPUT X		
D003 XEQ E001	→ ①	调用子程序 E。
D004 STO X	← ②	从这里返回。
D005 VIEW X		
D006 RTN		
E001 LBL E	← ①	开始子程序。
E002 +/-		改变数字的符号
E003 RTN	→ ②	返回至子程序 D。

迭套子程序

一个子程序可以调用另外一个子程序，而这另外一个子程序又可以调用其它子程序。子程序的这种“迭套”-即在另外一个子程序内调用子程序-仅限于 20 级深的子程序栈(不算最上面的程序级)。迭套子程序的运算如下所示：

主程序
(顶级)



程序结束

试图执行迭套超过 20 级的子程序将会导致 XEQ OVERFLOW 错误。

示例：迭套子程序。

以下子程序，标记为 S，计算作为更大程序中更大计算之一部分的表达式的数值。

$$\sqrt{a^2 + b^2 + c^2 + d^2}$$

该子程序调用另外一个标记为 Q 的子程序 (迭套子程序) 进行重复平方和加法运算。这样就通过使程序比没有子程序时要短来节省内存。

在 RPN 模式下，

S001 LBL S		从这里开始子程序。
S002 INPUT A		输入 A。
S003 INPUT B		输入 B。
S004 INPUT C		输入 C。
S005 INPUT D		输入 D。
S006 RCL D		恢复数据。
S007 RCL C		
S008 RCL B		
S009 RCL A		
S010 x ²		A ² 。
S011 XEQ Q001 → ①		A ² + B ² 。
② → S012 XEQ Q001 → ③		A ² + B ² + C ²
④ → S013 XEQ Q001 → ⑤		A ² + B ² + C ² + D ²
⑥ → S014 √ x		$\sqrt{A^2+B^2+C^2+D^2}$
S015 RTN		返回至主子程序。
Q001 LBL Q	← ①③⑤	迭套子程序
Q002 x<>y		
Q003 x ²		
Q004 +		加 x ² 。
②④⑥ ← Q005 RTN		返回至子程序 S。

分路 (GTO)

如同我们见到的子程序的情况一样，通常将程序执行转移至程序的一部分要比转移至下一个程序行要好。这就称作分路。

无条件分路使用 GTO(去往) 指令对特定程序行进行分路 (标签和行数字)。

编程 GTO 指令

GTO 标签指令 (按 **GTO** 标签行程序) 将正在运行的程序的执行转移至特定的程序行。程序继续从新位置运行, 绝不自动返回至其起点, 因此, 不把 GTO 用于子程序。

例如, 考虑第 16 章中的 "曲线拟合" 程序。GTO Z 001 指令将三个独立初始子程序中的任意一个从执行分路至 LBL Z, 也就是作为进入程序中心部分的共同入口点的子程序。

S001 LBL S		可以从这里开始。
.		
.		
S004 GTO Z001	→①	分路至 Z001。
L001 LBL L		可以从这里开始。
.		
.		
L004 GTO Z001	→①	分路至 Z001。
E001 LBL E		可以从这里开始。
.		
.		
E004 GTO Z001	→①	分路至 Z001。
Z001 LBL Z	←①	分路到这里。
.		
.		

使用键盘上的 GTO 指令

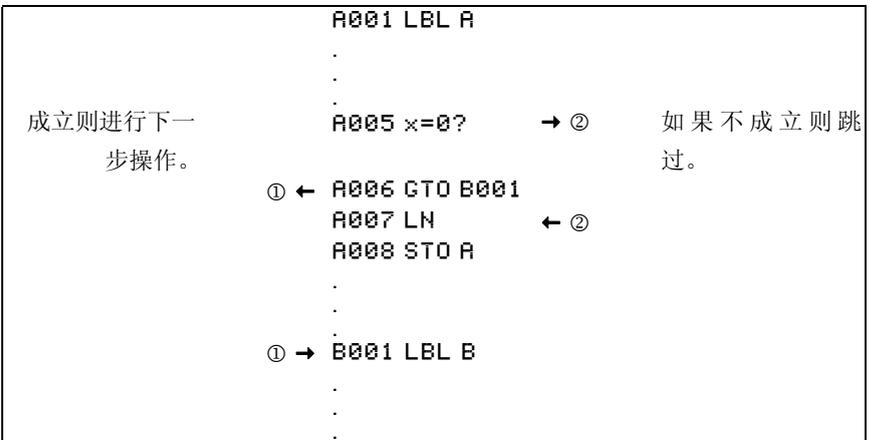
您可以使用 **GTO** 将程序指针移动至特定的标签行数字而不开始程序执行。

- 至 PRGM TOP: **GTO** **·** **·**。
- 至特定的行数字: **GTO** 标签行数字(行数字 < 1000)。例如, **GTO** **·** **A** **0** **0** **5**。例如, 按 **GTO** **A** **0** **0** **5**。显示屏将显示 "GTO A005"。
- 如果您想要查看标签的第一行, 例如, A001: **GTO** **A** **ENTER** (按下并保持不放), 显示屏将显示 "GTO A001"。

条件指令

改变程序执行的另外一种方法就是 *条件测试*, 即在命题不成立时对比较两数字并跳过下一个程序指令的成立 / 不成立测试。

例如, 如果程序行 A005 上的条件指令为 $x=0?$ (即, x 等于 0 吗?), 那么程序将拿 X 寄存器的内容与 0 进行比较。如果 X 寄存器的确包含 0, 则程序行继续滚动至下一行。如果 X 寄存器不包含 0, 则程序跳过下一行, 从而分路至程序行 A007。这一规则通称为 "成立则执行"。



上例子指出了一种常用于条件测试的方法; 紧随测试 (仅在 "成立" 的情况下执行) 之后的程序行是至另外一个标签的 *分路*。因此, 测试的静效应就是在某些情况下分路至不同的子程序。

条件指令有三类:

14-6 编程方法

- 比较测试。这些测试比较 X 和 Y 寄存器或 X 寄存器和 0。
- 标志测试。这些测试检查标志的状态，可以是已设置或清除。
- 回路计数器。这些通常用于按特定次数进行回路。

比较测试 (x?y, x?0)

有 12 种比较可用于编程。按  **x?y** 或  **x?0** 显示两类测试其中一类的菜单：

- **x?y** 用于比较 x 和 y 的测试。
- **x?0** 用于比较 x 和 0 的测试。

记住 x 指的是 X 寄存器中的数字，而 y 则指的是 Y 寄存器中的数字。这些测试不比较变量 X 和 Y。您可以使用 **x?y** 和 **x?0** 比较两个数字。如果其中一个数字非真数，将返回错误信息 **INVALID DATA**。

选择比较的类型，然后按菜单键找出您想要的条件指令。

测试菜单

x?y	x?0
≠ for x ≠ y?	≠ for x≠0?
≤ for x≤y?	≤ for x≤0?
< for x<y?	< for x<0?
> for x>y?	> for x>0?
≥ for x≥y?	≥ for x≥0?
= for x=y?	= for x=0?

如果您从键盘执行某个条件测试，计算器将显示 **YES** 或 **NO**。

例如，如果 $x = 2$ ， $y = 7$ ，测试 $x < y$ 。

键:

显示:

在 RPN 模式下 $\boxed{7} \boxed{\text{ENTER}} \boxed{2} \boxed{\leftarrow} \boxed{x?y} \boxed{>} \boxed{>} \boxed{(<)} \boxed{\text{ENTER}}$ YES

在 ALG 模式下 $\boxed{7} \boxed{x\leftrightarrow y} \boxed{2} \boxed{\leftarrow} \boxed{x?y} \boxed{>} \boxed{>} \boxed{(<)} \boxed{\text{ENTER}}$ YES

示例:

第 16 章 " 正态及正 / 逆态分布 " 程序在子程序 T 中使用 $x < y?$ 条件。

程序行: (RPN 模式)	说明
.	
.	
.	
T009 ÷	计算 X <i>猜测值</i> 的纠正值。
T010 STO+ X	加上纠正值得出一个新的 X <i>猜测值</i> 。
T011 ABS	
T012 0.0001	
T013 x<y?	测试看纠正值是否有效。
T014 GTO T001	如果纠正值有效, 返回至回路的开头处。 如果纠正值无效则继续。
T015 RCL X	
T016 VIEW X	显示 X 的计算值。
.	
.	
.	

程序行 T009 计算 X *猜测值* 的纠正值。程序行 T013 拿计算所得纠正值的绝对值于 0.0001 进行比较。如果该数值小于 0.0001 (" 成立则进行 "), 程序执行程序行 T014 ; 如果该数值等于或大于 0.0001 , 程序行跳至 T015。

标志

标志就是状态指示符。状态或者为*已设置 (成立)*，或者为*清除 (不成立)*。*标志测试*实际是另外一种遵循“成立则进行”规则的条件测试：如果受测标志状态为设置，程序执行直接进行；如果该标志的状态为清除，则程序执行跳过一行。

标志的意义

HP 35s 有 12 个标志，编号为 0 至 11。所有的标志都可通过键盘或程序指令加以设置、清除及测试。全部 12 个标志的默认状态为*清除*。附录 B 中探讨的三键内存清除操作将所有的标志清除。标志不受  **CLEAR**  **(3)** **(3ALL)**  **(Y)**  **ENTER** 的影响。

- **Flags 0、1、2、3 和 4** 的意义未事先指定。即，其状态意义将由您在给定程序中对其所下定义决定。（见下例。）
- **标志 5**，其状态为设置时，将在程序内发生溢出时终止程序并显示 **OVERFLOW** 和 。溢出是由结果超过计算器能够处理的最大数而引起的。可能的最大数替换为溢出结果。如果标志 5 的状态为设置，虽然程序最终停止时 **OVERFLOW** 会简短显示，带溢出的程序也不会终止。
- **标志 6** 是在发生溢出 **TOO BIG** 时由计算器*自动*设置的（虽然您自己也可以设置标志 6）。标志 6 无作用，但是可以加以测试。此外，在程序中使用非二进制基底时，标志 6 也针对程序中的 **TOO BIG** 进行设置。

标志 5 和 6 允许您控制程序运行期间发生的溢出条件。设置标志 5 会使程序在紧随导致溢出的程序行之后的程序行处停止。通过在程序中测试标志 6，您就可以在发生溢出时随时对程序流程或结果进行修改。

- **标志 7、8 和 9** 控制分数的显示。标志 7 还可从键盘加以控制。如在分数显示模式打开或关闭时按  **FDISP**，标志 7 也将被设置或清除。

标志状态	分数控制标志		
	7	8	9
清除 (默认)	分数显示关闭，以当前显示格式显示真数。	分数分母不大于 /c 值。	将分数减小至最小形式。
设置	分数显示打开，将真数作为分数显示。	分数分母为 /c 值的因数。	无分数减少。(仅在标志 8 被设置时使用。)

■ **标志 10** 控制方程的程序执行：

标志 10 处于清除状态时 (默认状态)，正在运行的程序中的方程自动被评估且结果被放到栈上。

标志 10 处于设置状态时，正在运行的方程中的方程显示为信息，导致其作用起来像 VIEW 陈述一样：

1. 程序执行停止。
 2. 程序指针移至下一程序行。
 3. 方程显示不对栈产生影响。您可以按  或  清除显示。按其它任何键执行该键的函数。
 4. 如果下一程序行为一个 PSE 指令，经历一秒钟的暂停后程序执行继续。
- 标志 10 的状态仅由键盘上 SF 和 CF 运算的执行或程序中的 SF 和 CF 陈述控制。

■ **标志 11** 在执行程序中的方程时控制提示 – 它在键盘执行期间不影响自动提示：

标志 11 处于清除状态时 (默认状态)，程序中方程的评估、 SOLVE 及 \int FN 继续进行而不终止。方程中各个变量的当前数值在每次遇到变量时自动恢复。INPUT 提示不受影响。

标志 11 处于设置状态时，在方程中首次遇到时，计算器提示输入各个变量。无论变量在方程中出现的次数如何，每个变量只提示一次。求解时，不提示未知数；而在积分时，则不提示积分的数值。提示使程序执行停止。如果  是您对于提示的唯一响应，按  用您键入的变量值或变量的显示 (当前) 值恢复计算。

标志 11 在程序中的方程评估、 SOLVE 或 \int FN 之后自动清除。标志 11 的状态也是由键盘上 SF 和 CF 运算的执行或程序中的 SF 和 CF 陈述控制。

设置标志信号器

标志 0、1、2、3 和 4 的显示中有在相应标志处于设置状态时打开的信号器。0、1、2、3 或 4 存在与否可让您随时了解这 5 个标志中的任何一个是否处于设置状态。但是，标志 5 至 11 并无此种状态显示。这些标志的状态可以通过执行键盘上的 FS? 指令确定。(见下文 "使用标志"。)

使用标志

按  **FLAGS** 显示 FLAGS 菜单: SF CF FS?

在选择好您想要的函数后，计算器会提示您输入标志数 (0-11)。例如，按  **FLAGS** **1**(1SF) **0** 设置标志 0，按  **FLAGS** **1**(1SF) **◦** **0** 设置表示 10，按  **FLAGS** **1**(1SF) **◦** **1** 设置标志 11。

FLAGS 菜单

菜单键	说明
SF n	设置标志。设置标志 n 。
CF n	清除标志。清除标志 n 。
FS? n	标志是够处于设置状态? 测试标志 n 的状态。

标志测试即和比较测试一样影响程序执行的条件测试。The FS? n 测试给定标志是否处于设置状态。如果是，则程序中下一程序行将被执行。如果不是，则下一程序行将被跳过。这就是 "成立则进行" 规则，见本章先前部分 "条件指令" 所述。

如果您用键盘对标志进行测试，计算器将显示 "YES" 或 "NO"。

在程序中确认您将要测试的条件是否从一种未知状态开始是一个比较好的做法。当前的标志设置取决于先前被运行的程序是如何对其进行处理的。例如，您不可以假定任何给定标志处于设置状态以及该标志只会在程序中的某个东西对其进行设置时被设置。在出现可能对其进行设置的条件之前，您应该通过清除标志对此加一确认。请参阅以下示例。

示例：使用标志。

程序行： (RPN 模式)	说明：
S001 LBL S	
S002 CF 0	清除标志 0，即 $\ln X$ 指示符。
S003 CF 1	清除标志 1，即 $\ln Y$ 指示符。
S004 INPUT X	提示并保存 X
S005 FS? 0	如果标志 0 处于设置状态 ...
S006 LN	... 取 X 输入的自然对数
S007 STO X	标志测试后将该数值存入 X
S008 INPUT Y	提示并保存 Y
S009 FS? 1	如果标志 1 处于设置状态 ...
S010 LN	... 取 Y 输入的自然对数
S011 STO Y	标志测试后将该数值存入 Y
S012 VIEW X	显示数值
S013 VIEW Y	显示数值
S014 RTN	
校和及长度：16B3 42	

如果您写入程序行 S002 CF0 和 S003 CF1(如上所示)，标志 0 和 1 被清除，因此程序行 S006 和 S010 不取 X 和 Y 输入的自然对数。

如果您用 SF0 和 CF1 替换程序行 S002 和 S003，则标志 0 被设置，因此。程序行 S006 取 X 输入的自然对数。

如果您用 CF0 和 SF1 替换程序行 S002 和 S003，则标志 1 被设置，因此。程序行 S010 取 Y 输入的自然对数。

如果您用 SFO 和 SF1 替换程序行 S002 和 S003，标志 0 和 1 被设置；因此，程序行 S006 和 S010 取 X 和 Y 输入的自然对数。

用上述程序查看标志的使用方法

键:	显示:	说明:
(在 RPN 模式下)		
XEQ S ENTER	X? 数值	执行标志 S；提示输入 X 值
1 R/S	Y? 数值	将 1 存入 X；提示输入 Y 值
1 R/S	X= 1.0000	将 1 存入 X；在标志测试后显示 X 值。
R/S	Y= 1.0000	在标志测试后显示 Y 值

您可以尝试一下其它情况。尝试后应记得按 **↶** **FLAGS** **2** (**2CF**) **0** 和 **↶** **FLAGS** **2** (**2CF**) **1** 清除标志 1 和 0。

示例：控制分数显示。

以下程序让您发挥计算器的分数显示性能。本程序提示并将您的输入用于分数及分母 (\sqrt{c} 值)。此外，本程序还包括 3 个分数显示标志 (7、8 和 9) 以及 " 信息显示 " 标志 (10) 的使用示例。

本程序中的信息作为 **MESSAGE** 列举并作为方程输入：

1. 按 **EQN** 设置方程输入模式 (**EQN** 信号器打开)。
2. 按 **RCL** 字符获取信息中的每一个阿尔法字符；按 **↷** **SPACE** 获取每个空格符。
3. 按 **ENTER** 将信息加入到当前程序行中并结束方程输入模式。

程序行:
(RPN 模式)

说明:

F001	LBL F	开始分数程序。
F002	CF 7	清除 3 个分数标志。
F003	CF 8	
F004	CF 9	
F005	SF 10	显示信息。
F006	DEC	选择二进制基底。
F007	INPUT V	提示输入数字。
F008	INPUT D	提示输入分母 (2 - 4095)。
F009	RCL V	先显示信息, 然后显示二进制数字。
F010	DECIMAL	
F011	PSE	
F012	STOP	
F013	RCL D	
F014	/C	设置 /c 值及标志 7。
F015	RCL V	
F016	MOST PRECISE	先显示信息, 然后显示分数。
F017	PSE	
F018	STOP	
F019	SF 8	设置标志 8。
F020	FACTOR DENOM	先显示信息, 然后显示分数。
F021	PSE	
F022	STOP	
F023	SF 9	设置标志 9。
F024	FIXED DENOM	先显示信息, 然后显示分数。
F025	PSE	
F026	STOP	
F027	GTO F001	查看程序的开头部分。

校和及长度: BE54 123

使用上述程序查看分数显示的不同形式：

键：	显示：	说明：
(在 RPN 模式下) [XEQ] [F] [ENTER]	V? 数值	执行标签 F；提示输入分数 (V)。
[2] [.] [5] [3] [R/S]	D? 数值	将 2.53 存入 V；提示输入分母 (D)。
[1] [6] [R/S]	DECIMAL 16.0000 2.5300	将 16 作为 /c 值保存。先显示信息，然后显示二进制数字。
[R/S]	MOST PRECISE 2 8/15 ▼ 2 8/15	信息显示分数格式(分母不大于 16)，然后显示分数。▼ 显示分子 " 略小于 "8。
[R/S]	FACTOR DENOM 2 1/2 ▲ 2 1/2	信息显示分数格式 (分母是 16 的因数)，然后显示分数。
[R/S]	FIXED DENOM 2 8/16 ▲ 2.5300	信息显示分数格式 (分母为 16)，然后显示分数。
[R/S] [C] [↩] [FLAGS]	2.5300	停止程序，清除标志 10
[2] (2CF) [.] [0]	2.5300	

回路

向后分路—即分路至先前程序行中的一个标签—使得能够超过一次以上执行程序中的某个部分。这被称为 *回路*。

```

D001 LBL D
D002 INPUT M
D003 INPUT N
D004 INPUT T
D005 GTO D001
    
```

本子程序就是一个不定回路的例子，可用于收集初始数据。在输入 3 个数值之后，将由您按 **XEQ** 标签行数字来终止本回路以执行其他子程序。

条件回路 (GTO)

当您想要在满足某项条件之后再继续进行运算而又不知道回路本身需要重复多少次时，您可以创建一个带条件测试和 GTO 指令的回路。

例如，以下子程序使用回路按常量 B 对数字 A 进行递减，直到最后得出的 A 小于或等于 B 。

程序行： (RPN 模式)	说明：
S001 LBL S	
S002 INPUT A	
S003 INPUT B	
S004 RCL A	恢复 A 比记住其在栈中的位置要容易。
S005 RCL- B	计算 $A - B$ 。
S006 STO A	用新结果替换原先的 A 。
S007 RCL B	恢复常量进行比较。
S008 x<y?	$B < \text{新 } A$?
S009 GTO S004	是：重复减的回路。
S010 VIEW A	否：显示新 A 。
S011 RTN	

校和及长度：2737 33

带计数器的回路 (DSE, ISG)

当您想要按特定次数执行回路时,用  **ISG** (增量,如大于则跳过)或  **DSE** (减量,如小于或等于则跳过)条件函数键。每次在程序中执行回路函数时,计算器自动对保存在变量中的某个计数器值进行递增或递减。计算器将当前计数器值与最终计数器值进行比较,然后根据结果继续或退出回路。

对于递减回路,使用  **DSE** 变量

对于递增回路,使用  **ISG** 变量

这些函数在 BASIC 中作为 FOR-NEXT 回路实现相同的作用:

FOR 变量 = 初始-变量 TO 最终变量 STEP 增量

.
. .
. . .

NEXT 变量

DSE 指令就像是一个带负增量的 FOR-NEXT 回路。

在按下 ISG 或 DSE( **ISG** 或  **DSE**) 的移动键后,计算器将提示您输入一个包含回路控制数(如下所述)的变量。

回路控制数

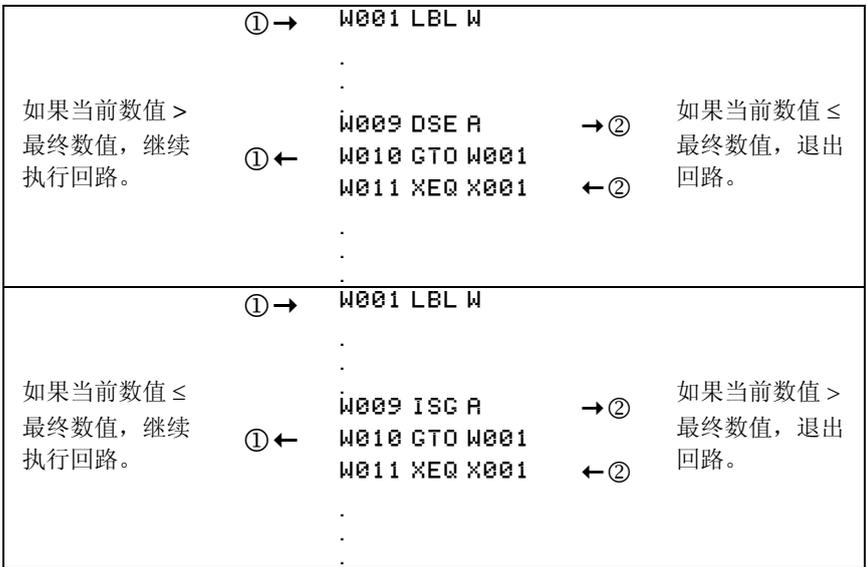
指定的变量应包含一个回路控制数 $\pm\text{cccccc}.\text{fffi}$, 其中:

- $\pm\text{cccccc}$ 为当前计数器值(1至12位)。该数值随着回路执行而改变。
- fff 为最终计数器值(必须为三位)。该值不随回路执行而改变。 fff 的非指定值假定为 000。

- ii 为递增及递减间隔(必须为两位或非指定)。该值不发生改变。 ii 的非指定值假定为 01(按 1 递增或递减)。

给定回路控制数 $cccccc.ffffi$, DSE 递减至 $cccccc - ii$, 比较新 $cccccc$ 与 fff , 并在本 $cccccc \leq fff$ 使程序执行跳过下一程序行。

给定回路控制数 $cccccc.ffffi$, ISG 使 $cccccc$ 递增至 $cccccc + ii$, 比较新 $cccccc$ 与 fff 并在本 $cccccc > fff$ 时使程序执行跳过下一程序行。



例如, ISG 回路控制数 0.050 指的是: 从 0 开始计数, 由小到大数至 50, 然后按每个回路使数字增大 1。

如果回路控制数为复数或矢量, 将使用真数部分或第一部分来对回路进行控制。

以下程序使用 ISG 在 RPN 模式下执行 10 次回路。回路计数器 (1.010) 保存在变量 Z 中。头 0 和尾 0 可以舍掉。

```
L001 LBL L
L002 1.01
L003 ST0 Z
L004 ISG Z
L005 GTO L004
L006 RTN
```

先按 **XEQ** **L** **ENTER**，然后按 **↩** **VIEW** **Z** 看回路控制数现在是否为 11.0100。

间接定址变量和标签

*间接定址*是一种用在高级编程过程中指定变量或标签而*不事先指定究竟哪一个*的方法。这是在程序运行时确定的，因此，取决于程序的中间结果 (或输入)。

间接定址使用 4 个不同的键：**I**，**(I)**，**J** 和 **(J)**。

这些键可用于许多以 A 至 Z 作为变量或标签的函数。

- *I*和*J*是内容可以指向其它变量的的变量，像其它任何变量(A至Z)一样保存数字。
- (I)和(J)为提供方向指示的编程函数，"使用I或J中的数字确定要定址哪一个变量或标签"。
这是一个*间接地址*。(A至Z为*直接地址*)。

I 和 **(I)** 一起用于创建间接地址，而这一点也适用于 **J** 和 **(J)**。

就其本身而言，(I) 或 (J) 既未被定义 ((I) 或 (J) 中无数字)，也未受到控制 (使用留在 I 或 J 中的任何数字)。

变量 "I" 和 "J"

您可以像保存、恢复和操作其它变量的内容一样保存、恢复和操作 I 或 J 的内容。您甚至可以求解 *I, J* 值并用 I 或 J 进行积分。下列函数可以使用变量 "i" (变量 J 情况相同)。

STO I
RCL I
STO +, -, ×, ÷ I
RCL +, -, ×, ÷ I

INPUT I
VIEW I
∫ FN d I
SOLVE I

DSE I
ISG I
x < > I

间接地址 (I) 和 (J)

许多使用 A 至 Z(作为变量或标签) 的函数都可以使用 (I) 或 (J) 间接指向 A 至 Z(变量或标签) 或统计寄存器。函数 (I) 或 (J) 变量 I 至 J 中的数值确定哪一个变量、标签或寄存器要定址。其方法如下表所示。

如果 I/J 包含:	那么 (I)/(J) 将定址:
-1	变量 A 或标签 A
.	.
.	.
.	.
-26	变量 Z 或标签 Z
-27	n 寄存器
-28	Σx 寄存器
-29	Σy 寄存器
-30	Σx^2 寄存器
-31	Σy^2 寄存器
-32	Σxy 寄存器
0	未命名间接变量开始
.	.
.	.
.	.
800	最大地址为 800
I<-32 或 I>800 或变量未确定	错误: INVALID (I)
J<-32 或 J>800 或变量未确定	错误: INVALID (J)

INPUT(I)、INPUT(J)、VIEW(I) 以及 VIEW(J) 给显示标记上间接定址变量或寄存器的名称。

SUMS 菜单使您能够从统计寄存器恢复数值。但是，您必须使用间接定址进行其它运算，如 STO，VIEW 及 INPUT。

下列函数可以使用 (I) 或 (J) 作为地址。对于 FN=(I) 或 (J) 指向一个标签；对于所有其它函数 (I) 或 (J) 指向一个变量或寄存器。

STO(I)/(J)

RCL(I)/(J)

STO +, -, ×, ÷, (I)/(J)

RCL +, -, ×, ÷, (I)/(J)

X<>(I)/(J)

FN=(I)/(J)

INPUT(I)/(J)

VIEW(I)/(J)

DSE(I)/(J)

ISG(I)/(J)

SOLVE(I)/(J)

∫ FN d(I)/(J)

您不可以对未命名的变量或统计寄存器进行求解或积分。

带 (I)/(J) 的程序控制

由于 I 的内容每次可随程序的运行而发生改变 – 甚至是在同一程序的不同部分中，程序指令如 STO (I) 或 (J) 可以不同次数将数值保存至不同的变量。例如，STO (-1) 表示将数字保存在变量 A 中。这样就通过不确定究竟需要哪个变量或程序标签 (直到程序开始运行为止) 来实现操作的灵活性。

间接定址对于计数和回路控制非常有用。变量 I 或 J 所起的作用就像是一个目录一样，保存包含函数 DSE 和 ISG 回路控制数的变量的地址。

带 (I)/(J) 的方程

您可以用方程中的 (I) 或 (J) 间接指定变量。注意 <I> 或 <J> 表示变量 I 或 J (间接参考数) 中数字指定的变量，但是变量 I 或 J 和 <I> 或 <J> (其中使用的用户括号表示而非 (I) 或 (J) 键) 指的是变量 I 或 J。

未命名间接变量

将一个正数放入变量 I 或 J 允许您访问高达 801 个间接变量。下例向您说明其使用方法。

程序行:
(RPN 模式)

说明:

A001 LBL R
A002 100
A003 STO I
A004 12345
A005 STO (I)

指定存储位址范围 "0-100" 并将 "12345" 保存到位址 100 中。

A006 150
A007 STO I
A008 67890
A009 STO (I)

将 "67890" 保存到位址 150 中。未定义的间接存储范围现在为 "0-150"。

A010 100
A011 STO I
A012 0
A013 STO (I)

将 0 保存到间接寄存器 100 中。未定义的范围仍旧为 "0-150"。

A014 170
A015 STO I
A016 RCL(I)
A017 RTN

显示 "INVALID (I)", 原因是位址 "170" 未定义。

注意:

1. 如果您想要从未定义的存储地址恢复数值, 将显示错误信息 "INVALID (I)"。(请参阅 A014)
2. 计算器将变量 0 的内存分配给非 0 变量。在使用变量后, 重要的是要将 0 存入变量以释放内存。每一个被分配的间接寄存器使用的程序内存为 37 字节。
3. 最多 800 个变量。

对程序进行求解和积分

对程序进行求解

第 7 章中，您了解到输入方程的方法 – 方程显示被添加到方程列表中，然后求解以获取任何变量。您还可以输入一个计算函数的 *程序*，然后对 *其* 求解以获取任何变量。这一点在您正求解的方程随某些条件而发生变化或需要重复进行计算时尤为有用。

求解编程函数：

1. 输入定义函数的程序 (见下文 " 针对 SOLVE 写入程序 ")。
2. 选择要求解的程序：按   标签。(如果您重新对同一程序求解，可以跳过这一步骤)。
3. 求解未知数：按   变量。

注意，如果您正求解的是编程函数，则需要 FN= ；如果您正求解的是方程列表中的方程，则不需要。

要想停止计算，按  或 ，第二行中将出现信息 **INTERRUPTED**。根的当前最佳估计值在未知变量中。使用   对其进行查看而不对栈产生干扰。要想恢复计算，按 。

针对 **SOLVE** 写入程序：

程序可以使用方程及 ALG 或 RPN 运算，取决于哪一种组合最为方便。

1. 从 **标签**处开始程序。本标签用以识别您希望 SOLVE 对其进行评估的函数 (**FN=** 标签)。
2. 加入各个变量的 INPUT 指令, 包括未知数。INPUT 指令使您能够求解多变量函数中的任何变量。未知数的 INPUT 被计算器所忽略, 因此, 对于每一个变量 (包括未知数), 您只需要写入一个包含一个 *单独* INPUT 指令的程序。

如果您不加入任何 INPUT 指令, 程序将使用保存在变量中或在收到方程提示时输入的数值。

3. 输入指令对函数进行评估。
 - 对于编程为多线 RPN 或 ALG 序列的函数, 其形式必须为根为 0 的表达式。如果您的方程为 $f(x) = g(x)$, 您的程序应计算 $f(x) - g(x)$ 。默认为 "=0"。
 - 编程为方程的函数可以是任何类型的方程 - 等式、赋值或表达式。方程由程序进行评估, 且其数值在解处为 0。如果您想要方程提示输入变量数值而不包括 INPUT 指令, 确认标志 11 处于设置状态。
4. 输入带 RTN 的程序。程序执行应以 X 寄存器中函数的数值结束。

示例: 用 ALG 编程。

用 ALG 运算写入一个程序, 求解 "理想气体定律" 方程中的任何未知数。方程为:

$$P \times V = N \times R \times T$$

式中

P = 压力 (大气压或 N/m^2)。

V = 体积 (升)。

N = 气体的分子量。

R = 通用气体常数

(0.0821 升-atm/ 摩尔 - 开氏温度或 8.314 J/ 摩尔开氏温度)。

T = 温度 (开氏温度 ; $K = ^\circ C + 273.1$)。

要想开始，把计算器置于程序模式下。必要时，将程序指针定位在程序内存的顶部。

键:	显示:	说明:
(在 ALG 模式下)		
PRGM		设置程序模式。
	PRGM TOP	

输入程序:

程序行: (ALG 模式)	说明:
G001 LBL G	辨别编程函数。
G002 INPUT P	保存压力 P
G003 INPUT V	保存体积 V
G004 INPUT N	保存气体的分子量 N
G005 INPUT R	保存气体常数 R
G006 INPUT T	保存温度 T 。
G007 $P \times V = N \times R \times T$	按 压力 \times 体积 = 分子 \times 气体常数 \times 温度。
G008 RTN	结束程序。

校和及长度: F425 33

按 取消程序输入模式。

使用程序 "G" 求 24°C 下体积为 2 升的瓶中 0.005 摩尔二氧化碳的压力。

键:	显示:	说明:
(在 ALG 模式下)		
FN=		选择 "G"– 程序。SOLVE 进行评估以求得未知变量的数值。
SOLVE	V?	选择 P ; 提示输入 V 值。
	value	
	N?	将 2 存入 V ; 求解 N 值。
	value	

. 0 0 5 R/S
 . 0 8 2 1
 R/S
 2 4 + 2 7
 3 . 1 ENTER
 R/S

R?
 value
 T?
 value
 T?
 297.1000
 SOLVING
 P=
 0.0610

将 .005 存入 N；求解 R 值。
 将 .0821 存入 R；求解 T 值。
 计算 T 值。
 将 297.1 存入 T；求解新 P 值。
 压力为 0.0610 大气压。

示例： 用方程编程。

写入程序，用方程求解 " 理想气体定律 "。

键:	显示:	说明:
(在 RPN 模式下)		
PRGM	PRGM TOP	选择程序输入模式。将程序指针移动到程序列表的顶部。
. .	H001 LBL H	给程序加上标记。
LBL H	H002 SF 11	启用方程提示。
FLAGS 1		
(1SF) . 1		对方程进行评估，清除标志 11。(
EQN		校和及长度：EDC8 9)。
RCL P		
RCL V =		
RCL N		
RCL R		
RCL T ENTER	H003 P×V=N×R×T	
RTN	H004 RTN	结束程序。
C	0.0610	取消程序输入模式。

程序的校验和及长度：DF52 21

现在，计算温度从前例中下降 10°C 时二氧化碳的压力变化。

15-4 对程序进行求解和积分

键:
(在 RPN 模式下)

STO **L**

FN= **H**

SOLVE **P**

R/S

R/S

R/S

ENTER **1** **0** **=**

R/S

RCL **L** **=**

显示:

0.0610

0.0610

V?

2.0000

N?

0.0050

R?

0.0821

T?

297.1000

T?

287.1000

SOLVING

P=

0.0589

-0.0021

说明:

保存先前的压力。

选择程序 "H"。

选择变量 P ; 提示输入 V 值。

2 保留在 V 中; 提示输入 N 值。

.005 保留在 N 中; 提示输入 R 值。

.0821 保留在 R 中; 提示输入 T 值。

计算新 T 值。

将 287.1 存入 T ; 求解新 P 值。

计算温度从 297.1 K 下降至 287.1 K 时气体的压力变化值 (结果为负表示压力下降)。

使用程序中的 SOLVE

您可以将 SOLVE 运算作为程序的一部分使用。

如果适当的话，在执行 SOLVE 变量指令之前加入或提示输入初始猜测值 (加入到未知变量及 X 寄存器中)。两个求解未知变量方程的指令在程序中出现为：

FH= 标签

SOLVE 变量

编程 SOLVE 指令不产生加有标记的显示 (变量 = 数值)，因为这可能不是您程序的有效输出 (即，在对其进行显示前，您可能希望用该数字进行进一步的计算)。如果您确实希望显示本结果，在 SOLVE 指令后加一个 VIEW 变量指令。

如未求出未知变量的解，则跳过下一程序行 (根据 " 成立则进行 " 规则。请参阅第 14 章)。程序于是应该处理未求出根的情况，如选择新的初始估计值或改变输入值。

示例：程序中的 SOLVE。

以下内容节选自一个允许您按 **[XEQ]** X 或 Y 求解 x 或 y 的程序。

程序行:
(RPN 模式)

说明:

X001 LBL X	X 设置。
X002 24	X 目录。
X003 GTO L001	分路至主子程序。
校和及长度: 62A0 11	
Y001 LBL Y	Y 设置。
Y002 25	Y 目录。
Y003 GTO L001	分路至主子程序。
校和及长度: 221E 11	
L001 LBL L	主子程序。
L002 STO I	将目录存入 I。
L003 FN= F	定义要求解的程序。
L004 SOLVE(I)	求解相应的变量。
L005 VIEW(I)	显示解。
L006 RTN	结束程序。
校和及长度: D45B 18	
F001 LBL F	计算 $f(x,y)$ 。按照要求加入 INPUT 或方程提示。
.	
.	
.	
F010 RTN	

对程序进行积分

在第 8 章中, 您了解到输入方程 (或表达式) 的方法 – 将其加入到方程列表中, 然后针对任何变量进行积分)。您还可以输入计算函数的程序, 然后针对任何变量对其进行积分。这一点在您正求解的方程随某些条件而发生变化或需要重复进行计算时尤为有用。

对编程函数进行积分:

1. 输入定义积分函数的函数的程序。(见下文 "针对 f FN 写入程序")。

2. 选择定义要积分函数的程序: 按 \leftarrow $\boxed{\text{FN=}}$ 标签 (如果您是在对同一程序重新进行积分, 您可以跳过这一步骤)。
3. 输入积分极限: 先键入 *下限*, 按 $\boxed{\text{ENTER}}$, 然后键入 *上限*。
4. 选择积分变量, 开始进行计算: 按 \leftarrow $\boxed{\int}$ 变量。

注意, 如果您正求解的是编程函数, 则需要 FN= ; 如果您正求解的是方程列表中的方程, 则不需要。

您可以按 $\boxed{\text{C}}$ 或 $\boxed{\text{R/S}}$ 停止正在进行的积分计算, 第二行中将出现信息 **INTERRUPTED**。但是, 计算无法恢复。在计算正常结束前, 将不会出现有关积分的信息。

在积分计算正在进行时按 $\boxed{\text{XEQ}}$ 将取消 $\boxed{\int}$ $\boxed{\text{FN=}}$ 运算。在这种情况下, 您应该从头开始再次开始 $\boxed{\int}$ $\boxed{\text{FN=}}$ 。

写如 \int **FN** 的程序:

程序可以使用方程, **ALG** 或 **RPN** 运算, 取决于哪一种组合最为方便。

1. 从 *标签*处开始程序。本标签用以识别您想要对其进行积分的函数 (**FN=** 标签)。
2. 加入各个变量的 **INPUT** 指令, 包括积分的变量。**INPUT** 指令使您能够针对多变量函数中的任何变量进行积分。积分变量的 **INPUT** 被计算器所忽视, 因此对于每一个变量 (包括积分的变量) 您只需要些入一个包含 *单独* **INPUT** 指令的程序。

如果您不加入任何 **INPUT** 指令, 程序将使用保存在变量中或在收到方程提示时输入的数值。

3. 输入指令对函数进行评估。

- 变成多行RPN或ALG序列的函数必须计算您想要对其进行积分的函数数值。
- 编程为方程的函数通常包含在指定积分函数的表达式中—虽然它可以是任何类型的方程。如果您想要方程提示输入变量数值而不包括 INPUT 指令，确认标志 11 处于设置状态。

4. 输入带 RTN 的程序。程序执行应以 X 寄存器中函数的数值结束。

示例：用方程编程。

第 8 章例中的正弦积分函数为

$$Si(t) = \int_0^t \left(\frac{\sin x}{x} \right) dx$$

该函数可通过对定义积分函数的程序进行积分来加以评估：

```

S001 LBL S      对函数进行定义。
S002 SIN(X)÷X  作为表达式的函数。(校验和及长度：0EE0 8)。
S003 RTN       结束子程序
程序的校验和及长度：D57E 17
  
```

输入本程序，从 0 to 2 ($t = 2$)。针对 x 对正弦积分函数进行积分。

键：	显示：	说明：
(在 RPN 模式下)		
MODE 2 (2RAD)		选择半径模式。
↶ FN= S		选择标签 S 作为积分函数。
0 ENTER 2	2_	输入积分上限和下限。
↶ / ⊗	INTEGRATING	从 0 至 2 对函数进行积分；显示结果。
	∫ =	
	1.6054	
MODE 1 (1DEG)	1.6054	恢复角度模式。

使用程序中的积分

积分可以从程序进行执行。记得要在执行积分前加入或提示输入积分极限并记住准确度及执行时间是由程序运行时的显示格式控制的。两个积分指令在程序中出现的为：

FN= 标签

∫ FN d 变量

编程 **∫ FN** 指令并不产生加有标记的显示 (**∫ = 数值**)，因为这可能不是您程序的有效输出 (即，在对其进行显示前，您可能想要用该数字进行进一步的计算)。如果您确实要显示本结果，添加一个 **PSE**( **PSE**) 或 **STOP**( **STOP**) 指令显示 X- 寄存器中 **∫ FN** 指令之后的结果。

如果 **PSE** 指令在每次重复进行积分或求解期间紧随所显示的方程 (标志 10) 之后，方程将显示 1 秒钟，方程执行将继续直到每次重复结束为止。在显示方程期间，不允许滚动屏幕或进行键盘输入。

示例：程序中的 **∫ FN**。

第 16 章 " 正态及正 / 逆态分布 " 程序包含通常密度函数方程的积分。

$$\frac{1}{\sqrt{2\pi}} \int_M^D e^{-\frac{(D-M)^2}{S}} dD.$$

$e^{((D-M) \div S)^2 \div 2}$ 函数由标记为 **F** 的子程序进行计算。其它子程序提示输入未知数值并进行其它计算以求得 **Q(D)**，也即通常曲线的上尾部面积。积分本身从子程序 **Q** 进行设置和执行：

Q001	LBL Q	
Q002	RCL M	恢复积分下限。
Q003	RCL X	恢复积分上限。(X = D.)
Q004	FN= F	指定函数。
Q005	∫ FN d D	用虚拟变量 D 对普通函数进行积分。

求解及积分限制

SOLVE 变量和 ∫ FN d 变量指令不能够调用包含其它 SOLVE 或 ∫ FN 指令的子程序。即，这些指令中无论哪一个都不能递归使用。例如，试图计算复积分将导致 ∫ (∫ FN) 错误。另外，SOLVE 和 ∫ FN 都不能调用包含 FN= 标签指令的子程序；如果试图这么做，将返回 SOLVE ACTIVE 或 ∫ FN ACTIVE 错误。SOLVE 不能调用包含 ∫ FN 指令(产生 SOLVE (∫ FN) 错误)的子程序，就如同 ∫ FN 不能够调用包含 SOLVE 指令(产生 ∫ (SOLVE) 错误)的子程序。

程序中的 SOLVE 变量和 ∫ FN d 变量指令使用计算器中 20 个待返回子程序之一。(请参阅第 14 章 "迭套子程序"。)

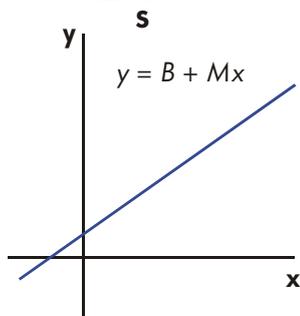
统计程序

拟合曲线

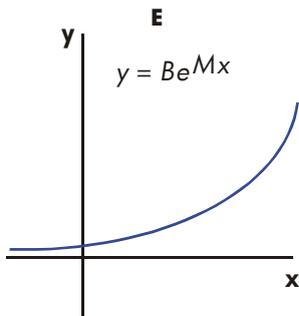
本程序可用于适 4 个方程模型之中的一个与您的数据相适应。这些模型为直线、对数曲线，指数曲线及幂曲线。程序先接受两个或多个 (x, y) 数据对，然后计算相关系数 r 以及两个回归系数 m 和 b 。程序包含计算估计值 \hat{x} 及 \hat{y} 的子程序。(有关这些数值的定义，见第 12 章 "线性回归"。)

曲线及相关方程的样本在下文显示。HP 35s 的内部回归函数用于计算回归系数。

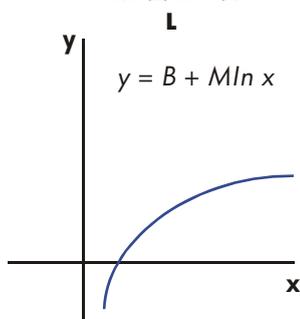
直线拟合



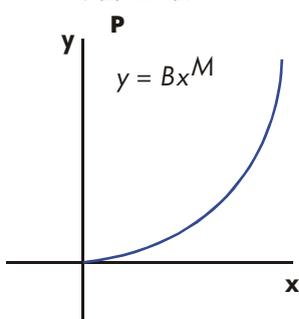
指数曲线拟合



对数曲线拟合



幂曲线拟合



要想使对数曲线拟合， x 的数值必须为正。要想使指数曲线拟合， y 的数值必须为正。要想使幂曲线拟合， x 和 y 的数值都必须为正。如果在这些情况下输入的是负数，将导致 LOG(NEG) 错误。

倍数较大但差异相对较小的数据数值可导致精确度问题，倍数差异较大的数据数值也是如此。请参阅第 12 章 "数据精确度限制"。

程序列举:

程序行: (RPN 模式)	说明
S001 LBL S	本子程序设置直线模型的状态。
S002 CF 0	清除标志 0, 即 ln X 指示符。
S003 CF 1	清除标志 1, 即 ln Y 指示符。
S004 GTO Z001	分路至共同入口点 Z。
校和及长度: 8E85 12	
L001 LBL L	本子程序设置对数模型的状态。
L002 SF 0	设置标志 0, ln X 的指示符。
L003 CF 1	清除标志 1, 即 ln Y 指示符。
L004 GTO Z001	分路至共同入口点 Z。
校和及长度: AD1B 12	
E001 LBL E	本子程序设置指数模型的状态。
E002 CF 0	清除标志 0, 即 ln X 指示符。
E003 SF 1	设置标志 1, ln Y 的指示符。
E004 GTO Z001	分路至共同入口点 Z。
校和及长度: D6F1 12	
P001 LBL P	本子程序设置幂模型的状态。
P002 SF 0	设置标志 0, ln X 的指示符。
P003 SF 1	设置标志 1, ln Y 的指示符。
校和及长度: 3800 9	
Z001 LBL Z	定义所有模型的共同入口点。
Z002 CLΣ	清除统计寄存器。按  CLEAR  (4Σ)
Z003 0	针对第一个输入将回路计数器设置为 0。
校和及长度: 8611 10	
W001 LBL W	定义输入回路的开始。
W002 1	调整回路计数器 1 提示输入。
W003 +	
W004 STO X	将回路计数器存入 X, 使之出现时提示 X。
W005 INPUT X	显示带提示的计数器并保存 X 输入。

程序行:
(RPN 模式)

说明

W006 FS? 0	如果标志 0 处于设置状态 ...
W007 LN	... 取 X 输入的自然对数。
W008 STO B	保存纠正子程序的数值。
W009 INPUT Y	提示并保存 Y。
W010 FS? 1	如果标志 1 处于设置状态 ...
W011 LN	... 取 Y 输入的自然对数。
W012 STO R	
W013 RCL B	
W014 $\Sigma+$	在统计寄存器中将 B 和 R 作为 x,y 数据对累积。
W015 GTO W001	另外一个 X, Y 数据对的回路。

校和及长度: 9560 46

U001 LBL U	定义“取消”子程序的开始。
U002 RCL R	恢复最新的数据对。
U003 RCL B	
U004 $\Sigma-$	从统计累积中删除本数据对。
U005 GTO W001	另外一个 X, Y 数据对的回路。

校和及长度: A79F 15

R001 LBL R	定义输出子程序的开始
R002 r	计算相关系数。
R003 STO R	将其保存在 R 中。
R004 VIEW R	显示相关系数。
R005 b	计算系数 b。
R006 FS? 1	如果标志 1 处于设置状态, 取 b 的自然对数。
R007 e ^x	
R008 STO B	将 b 存入 B。
R009 VIEW B	显示数值。
R010 m	计算系数 m。
R011 STO M	保存 m 在 M 中。
R012 VIEW M	显示数值。

校和及长度: 850C 36

Y001 LBL Y	计算估计 (预计) 回路的开始。
------------	--------------------

程序行: (RPN 模式)	说明
Y002 INPUT X	显示, 提示并在发生改变时将 x 值存入 X 。
Y003 FS?0	如果标志 0 处于设置状态 ...
Y004 GTO K001	分路至 K001
Y005 GTO M001	分路至 M001
Y006 STO Y	将 \hat{y} - 值存入 Y 。
Y007 INPUT Y	显示, 提示并在发生改变时将 y 值存入 Y 。
Y008 FS?0	如果标志 0 处于设置状态 ...
Y009 GTO O001	分路至 O001
Y010 GTO N001	分路至 N001
Y011 STO X	针对下一回路将 \hat{x} 存入 X 。
Y012 GTO Y001	另外一个估计值的回路。
校和及长度: C3B7 36	

A001 LBL A	本子程序计算执行模型的 \hat{y} 。
A002 RCL M	
A003 RCL \times X	
A004 RCL+ B	计算 $\hat{y} = MX + B$ 。
A005 RTN	返回至调用子程序。
校和及长度: 9688 15	

G001 LBL G	本子程序计算执行模型的 \hat{x} 。
G002 RCL Y	
G003 RCL- B	
G004 RCL \div M	计算 $\hat{x} = (Y - B) \div M$ 。
G005 RTN	返回至调用子程序。
校和及长度: 9C0F 15	

B001 LBL B	本子程序计算对数模型的 \hat{y} 。
B002 RCL X	
B003 LN	
B004 RCL \times M	
B005 RCL+ B	计算 $\hat{y} = M \ln X + B$ 。
B006 RTN	返回至调用子程序。

程序行:
(RPN 模式)

说明

校和及长度: 889C 18

H001 LBL H 本子程序计算对数模型的 \hat{x} 。
H002 RCL Y
H003 RCL - B
H004 RCL ÷ M
H005 e^x 计算 $\hat{x} = e^{(Y - B) \div M}$
H006 RTN 返回至调用子程序。

校和及长度: 0DBE 18

C001 LBL C 本子程序计算指数模型的 \hat{y} 。
C002 RCL M
C003 RCL × X
C004 e^x
C005 RCL × B 计算 $\hat{y} = Be^{MX}$ 。
C006 GTO M005 分路至 M005

校和及长度: 9327 18

I001 LBL I 本子程序计算指数模型的 \hat{x} 。
I002 RCL Y
I003 RCL ÷ B
I004 LN
I005 RCL ÷ M 计算 $\hat{x} = (\ln(Y \div B)) \div M$ 。
I006 GTO N005 查看 N005

校和及长度: 7219 18

D001 LBL D 本子程序计算幂模型的 \hat{y} 。
D002 RCL X
D003 RCL M
D004 y^x
D005 RCL × B 计算 $Y = B(X^M)$ 。
D006 GTO K005 查看 K005

校和及长度: 11B3 18

J001 LBL J 本子程序计算幂模型的 \hat{x} 。

程序行:
(RPN 模式)

说明

J002 RCL Y
J003 RCL ÷ B
J004 RCL M
J005 1/x

J006 y^x 计算 $\hat{x} = (Y/B)^{1/M}$

J007 GTO 0005 查看 0005

校和及长度: 8524 21

K001 LBL K 确定是否应该运行 D001 或 B001

K002 FS?1 如果标志 1 处于设置状态 ...

K003 XEQ D001 执行 D001

K004 XEQ B001 执行 B001

K005 GTO Y006 查看 Y006

校和及长度: 4BFA 15

M001 LBL M 确定是否应该运行 C001 或 A001

M002 FS?1 如果标志 1 处于设置状态 ...

M003 XEQ C001 执行 C001

M004 XEQ A001 执行 A001

M005 GTO Y006 查看 Y006

校和及长度: 1C4D 15

O001 LBL O 确定是否应该运行 J001 或 H001

O002 FS?1 如果标志 1 处于设置状态 ...

O003 XEQ J001 执行 J001

O004 XEQ H001 执行 H001

O005 GTO Y011 查看 Y011

校和及长度: 0AA5 15

N001 LBL N 确定是否应该运行 I001 或 G001

N002 FS?1 如果标志 1 处于设置状态 ...

N003 XEQ I001 执行 I001

N004 XEQ G001 执行 G001

N005 GTO Y011 查看 Y011

校和及长度: 666D 15

使用的标志:

如果 X 输入需要自然对数, 则标志 0 处于设置状态。如果 Y 输入需要自然对数, 则标志 1 处于设置状态。

如果标志 1 设置在子程序 N 中, 则执行 I001。如果标志 1 处于清除状态, 则执行 G001。

程序指令:

1. 键入程序调度程序; 完成后按 **[C]**。
2. 先按 **[XEQ]**, 然后按: 选择您想要拟合的曲线类型:
 - **[S]** **[ENTER]** 如果是直线;
 - **[L]** **[ENTER]** 如果是对数曲线;
 - **[E]** **[ENTER]** 如果是指数曲线; 或
 - **[P]** **[ENTER]** 如果是幂曲线。
3. 键入 x 值, 然后按 **[R/S]**。
4. 键入 y 值, 然后按 **[R/S]**。
5. 对于每一个数据对重复步骤 3 和 4。如果您发现您在步骤 3 中按下 **[R/S]** 后出错 ($Y?$ 数值提示仍然可见), 再次按 **[R/S]** (显示 $X?$ 数值提示) 并按 **[XEQ]** **[U]** **[ENTER]** 取消 (移除) 最后一个数据对。如果您发现您在执行步骤 4 后出错, 按 **[XEQ]** **[U]** **[ENTER]**。无论是这两种情况的哪一种, 都应在步骤 3 那里继续。
6. 键入所有的数据之后, 按 **[XEQ]** **[R]** **[ENTER]** 查看相关系数 R 。
7. 按 **[R/S]** 查看回归系数 B 。
8. 按 **[R/S]** 查看回归系数 M 。
9. 按 **[R/S]** 查看 \hat{x} , \hat{y} 估计子程序的 $X?$ 数值提示。
10. 如果您想要根据 x 对 \hat{y} 进行估值, 在收到 $X?$ 数值提示时键入 x , 然后按 **[R/S]** 查看 \hat{y} ($Y?$)。
11. 如果您想要根据 y 对 \hat{x} 进行估值, 按 **[R/S]** 直到您看到 $Y?$ 数值提示, 键入 y , 然后按 **[R/S]** 查看 \hat{x} ($X?$)。
12. 有关更多估计值的情况, 请参阅步骤 10 或 11。
13. 对于新情况, 请参阅步骤 2。

使用的变量:

B	回归系数 (直线的 y 交点); 也用于刻划。
M	回归系数 (直线的曲率)。
R	相关系数; 也用于刻划。
X	输入数据时数据对的 x 值; 预计 \hat{y} 时的假设 x 值或给定假设 y 值时的 \hat{x} (x 估计值)。
Y	输入数据时数据对的 y 值; 预计 \hat{x} 时的假设 y 值或给定假设 x 值时的 \hat{y} (y 估计值)。
统计寄存器	统计累积及计算。

示例 1:

使直线与以下数据拟合。键入第三个数据对时故意犯错, 然后用取消子程序对错误进行纠正。另外, 针对 x 值 37 估计 y 值。针对 y 值 101 估计 x 值。

X	40.5	38.6	37.9	36.2	35.1	34.6
Y	104.5	102	100	97.5	95.5	94

键:	显示:	说明:
(在 RPN 模式下)		
XEQ S ENTER	X? 1.0000	开始直线子程序。
4 0 . 5 R/S	Y? 数值	输入数据对的 x 值。

1 0 4 . 5	X?	输入数据对的 y 值。
R/S	2.0000	
3 8 . 6 R/S	Y?	输入数据对的 x 值。
	104.5000	
1 0 2 R/S	X?	输入数据对的 y 值。
	3.0000	

现在，故意输入 379 而非 37.9，这样您就可以了解如何对错误输入进行纠正。

键:	显示:	说明:
(在 RPN 模式下)		
3 7 9 R/S	Y?	输入数据对错误的 x 值。
	102.0000	
R/S	X?	恢复 X? 提示。
	4.0000	
XEQ U ENTER	X?	删除最后一个数据对。现在开始进行正确的数据输入。
	3.0000	
3 7 . 9 R/S	Y?	输入数据对正确的 x 值。
	102.0000	
1 0 0 R/S	X?	输入数据对的 y 值。
	4.0000	
3 6 . 2 R/S	Y?	输入数据对的 x 值。
	100.0000	
9 7 . 5 R/S	X?	输入数据对的 y 值。
	5.0000	
3 5 . 1 R/S	Y?	输入数据对的 x 值。
	97.5000	
9 5 . 5 R/S	X?	输入数据对的 y 值。
	6.0000	
3 4 . 6 R/S	Y?	输入数据对的 x 值。
	95.5000	
9 4 R/S	X?	输入数据对的 y 值。
	7.0000	
XEQ R ENTER	R=	计算相关系数。
	0.9955	

R/S	B=	33.5271	计算回归系数 B。
R/S	M=	1.7601	计算回归系数 M。
R/S	X?	7.0000	提示输入假设 x 值。
3 7 R/S	Y?	98.6526	将 37 存入 X, 计算 \hat{y} 。
1 0 1 R/S	X?	38.3336	将 101 存入 Y, 并计算 \hat{x} 。

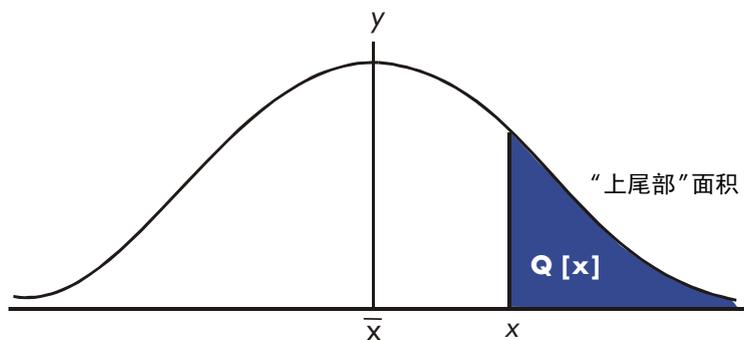
示例 2:

重复例 1(使用相同的数据)进行对数、指及幂曲线拟合。下表提供给您各类曲线的开始执行标签几结果(相关/回归系数及 x 和 y 估计值)。每次您运行程序进行不同的曲线拟合时您都需要重新输入数据数值。

	对数	指数	幂
开始:	XEQ L ENTER	XEQ E ENTER	XEQ P ENTER
R	0.9965	0.9945	0.9959
B	-139.0088	51.1312	8.9730
M	65.8446	0.0177	0.6640
Y(\hat{y} 当 X=37)	98.7508	98.5870	98.6845
X(\hat{x} 当 Y=101)	38.2857	38.3628	38.3151

正态及逆 / 正态分布

正态分布常用于关于某个平均值模拟随机变量的作用。该模型假设样本分布关于标准差为 S 的平均值 M 对称并接近下文所示钟形曲线的形状。给定 x 值, 本程序计算从样本数据中随机选择数值要更高的几率。这被称为上尾部面积 Q(x)。本程序还提供倒数: 给定 Q(x) 值, 计算相应的 x 值。



$$Q(x) = 0.5 - \frac{1}{\sigma\sqrt{2\pi}} \int_{\bar{x}}^x e^{-((x-\bar{x})/\sigma)^2/2} dx$$

本程序使用 HP 35s 的内置积分功能对普通频率曲线的方程进行积分。倒数用牛顿法获得，反复搜索得出给定几率 $Q(x)$ 的 x 值。

程序列举:

程序行: (RPN 模式)	说明
S001 LBL S	本子程序对正态分布程序进行初始化。
S002 0	保存默认平均值。
S003 STO M	
S004 INPUT M	提示输入并保存平均值 M 。
S005 1	保存默认标准差数值。
S006 STO S	
S007 INPUT S	提示输入并保存标准差 S 。
S008 RTN	停止显示标准差数值。
校和及长度: 70BF 26	
D001 LBL D	本子程序计算 $Q(X)$ 给定 X 。
D002 INPUT X	提示并保存 X 。
D003 XEQ Q001	计算上尾部面积。
D004 STO Q	将数值存入 Q 使 VIEW 函数能够对其进行显示。
D005 VIEW Q	显示 $Q(X)$ 。
D006 GTO D001	计算另外一个 $Q(X)$ 的回路。
校和及长度: 042A 18	
I001 LBL I	本子程序计算 X 给定 $Q(X)$ 。
I002 INPUT Q	提示输入并保存 $Q(X)$ 。
I003 RCL M	恢复平均值。
I004 STO X	将平均值保存为 X 的猜测值, 称为 $X_{\text{猜测值}}$ 。
校和及长度: A970 12	
T001 LBL T	本标签定义重复回路的开始。
T002 XEQ Q001	计算 $(Q(X_{\text{猜测值}}) - Q(X))$ 。
T003 RCL - Q	
T004 RCL X	
T005 STO D	
T006 R↓	
T007 XEQ F001	在 $X_{\text{猜测值}}$ 处计算衍生数。
T008 RCL ÷ T	
T009 ÷	计算 $X_{\text{猜测值}}$ 的纠正值。

程序行:
(RPN 模式)

说明

T010 STO+ X 加上纠正值得出一个新的 X *猜测值*。
T011 ABS
T012 0.0001
T013 x<y? 测试看纠正值是否有效。
T014 GTO T001 如果纠正值有效, 返回至回路的开头处。如果纠正值无效
 则继续。
T015 RCL X
T016 VIEW X 显示 X 的计算值。
T017 GTO I001 计算另外一个 X 的回路。
校和及长度: EDF4 57

Q001 LBL Q 本子程序计算上尾部面积 $Q(x)$ 。
Q002 RCL M 恢复积分下限。
Q003 RCL X 恢复积分上限。
Q004 FN= F 选择 LBL F 定义的函数进行积分。
Q005 J FN d D 用虚拟变量 D 对普通函数进行积分。
Q006 2
Q007 π
Q008 \times
Q009 $\sqrt{\times}$
Q010 RCL \times S 计算 $S \times \sqrt{2\pi}$ 。
Q011 STO T 针对逆子程序临时保存结果。
Q012 \div
Q013 +/-
Q014 0.5
Q015 + 加上曲线下一半的面积, 因为我们以平均值作为下限进行
 积分。
Q0016 RTN 返回至调用子程序。
校和及长度: 8387 52

F001 LBL F 本子程序计算普通函数 $e^{-((X-M)\div S)^2\div 2}$ 的积分函数。
F002 RCL D
F003 RCL- M

程序行:
(RPN 模式)

说明

F004 RCL ÷ S
F005 ×2
F006 2
F007 ÷
F008 +/-
F009 e^x
F010 RTN 返回至调用子程序。
校和及长度: B3EB 31

使用的标志:

无。

备注:

本程序的准确度取决于显示设置。对于 ± 3 标准差范围内的输入，显示 4 个或更多有效数字对于大多数应用而言是足够的。

全精度时，输入极限变为 ± 5 标准差。计算时间由于所显示数位的数量降低而大幅缩短。

在子程序 Q，常数 0.5 可以用 2 和 $\frac{1}{x}$ 替换。

如果您对于倒数性能不感兴趣的话，则无需键入倒数子程序 (在子程序 I 和 T 中)。

程序指令:

1. 键入程序调度程序；完成后按 **C**。
2. 按下 **XEQ** **S** **ENTER**。
3. 在提示输入 M 值后，键入总体平均值，按 **R/S** (如果平均值为 0，则只按 **R/S** 即可)。

4. 在提示输入 S 后，键入总体标准差并按 **[R/S]**。(如果标准差为 1，则只需按 **[R/S]** 即可。)
5. 要想以给定 $Q(X)$ 计算 X ，跳至这些指令的步骤 9。
6. 要想以给定 X 计算 $Q(X)$ ，按 **[XEQ] [D] [ENTER]**。
7. 在收到提示后，键入 X 值，按 **[R/S]**。显示结果 $Q(X)$ 。
8. 要想用相同的平均值和标准差针对新 X 计算 $Q(X)$ ，按 **[R/S]**，执行步骤 7。
9. 要想以给定 $Q(X)$ 计算 X ，按 **[XEQ] [I] [ENTER]**。
10. 收到提示后，键入 $Q(X)$ 值，按 **[R/S]**。显示结果 X 。
11. 要想用相同的平均值和标准差针对新 $Q(X)$ 计算 X ，按 **[R/S]**，执行步骤 10。

使用的变量：

D	虚拟积分变量。
M	总体平均值，默认值 0 。
Q	与上尾部面积相对应的几率。
S	总体标准差，默认值 1 。
T	临时用于将数值 $S \times \sqrt{2\pi}$ 传递到反程序中的变量。
X	输入定义上尾部面积左侧的数值。

示例 1:

您的好朋友告诉您说您的盲约智商为 " 3σ "。您对此是这么解释的：这个人比除了高出平均值 3 个标准差的人之外的当地人要更聪明。

假定您认为当地人的盲约数为 10,000。那么有多少人会在 " 3σ " 这一组呢？由于这一问题是以标准差来进行论述的，使用 M 的默认值 0 和 S 的默认值 1 。

键：	显示：	说明：
(在 RPN 模式下) [XEQ] [S] [ENTER]	$M?$ 0.0000	开始初始化子程序。

R/S

S?

接受 M 的默认值 0。

R/S

1.0000

接受 S 的默认值 1。

XEQ **D** **ENTER**

1.0000

开始分配程序并提示输入 X 。

X?

数值

3 **R/S**

Q=

输入 X 值 3, 开始计算 $Q(X)$ 。显示比在平均值三个标准差以内的每个人都要聪明的人数比例。

0.0013

1 **0** **0** **0** **0** 13.4984

乘以总数。显示符合标准的当地人的盲约近似数。

X

由于您的朋友有时候会夸大其辞, 您决定看看 " 2σ " 盲约会有多稀罕。注意按 **R/S** 即使程序重新运行。

键:
(在 RPN 模式下)

显示:

说明:

R/S

X?

恢复程序。

2 **R/S**

3.0000

输入 X 值 2, 计算 $Q(X)$ 。

Q=

1 **0** **0** **0** **0**

0.0228

针对修订后的估计值乘以总数。

227.5012

X

示例 2:

一组考分的平均值为 55。标准差为 15.3。假定标准正态曲线充分模拟分布状态, 那么一个随机选择的学生得分至少为 90 的几率会是多少呢? 若要 10% 的学生及格分数会是多少呢? 若要 20% 的学生不及格分数又会是多少呢?

键:	显示:	说明:
(在 RPN 模式下)		
XEQ S ENTER	M?	开始初始化子程序。
	0.0000	
5 5 R/S	S?	保存平均值 55。
	1.0000	
1 5 . 3 R/S	15.3000	保存标准差 15.3。
XEQ D ENTER	X?	开始分配程序并提示输入 X。
	数值	
9 0 R/S	Q=	输入 X 值 90, 计算 Q(X)。
	0.0111	

因此, 我们预计仅约 1% 的学生得分会超过 90。

键:	显示:	说明:
(在 RPN 模式下)		
XEQ I ENTER	Q?	开始反子程序。
	0.0111	
0 . 1 R/S	X=	将 0.1 (百分之十) 存入 Q(X), 计算 X。
	74.6077	
R/S	Q?	恢复反子程序。
	0.1000	
0 . 8 R/S	X=	将 0.8 (百分之百减百分之二十) 存入 Q(X), 计算 X。
	42.1232	

分组标准差

分组数据的标准差 S_{xy} 为数据点 x_1, x_2, \dots, x_n 的标准差, 其发生正整数频率为 f_1, f_2, \dots, f_n 。

$$S_{xy} = \sqrt{\frac{\sum x_i^2 f_i - \frac{(\sum x_i f_i)^2}{\sum f_i}}{(\sum f_i) - 1}}$$

本程序允许您输入数据、纠正输入项并计算标准差及分组数据加权平均值。

程序列举：

程序行： (ALG 模式)	说明
S001 LBL S	开始分组标准差程序。
S002 CLΣ	清除统计寄存器 (-27 至 -32)。
S003 0	
S004 STO N	清除计数 N。
校和及长度：E5BC 13	
I001 LBL I	输入统计数据点。
I002 INPUT X	将数据点存入 X。
I003 INPUT F	将数据点频率存入 F。
I004 1	输入增量 N。
I005 STO B	
I006 RCL F	恢复数据点频率 f_i 。
校和及长度：3387 19	
F001 LBL F	累积总合。
F002 -27	
F003 STO I	保存目录寄存器 -27。
F004 RCL F	
F005 STO+(I)	在寄存器 -27 中更新 $\sum f_i$ 。
F006 RCL× X	$x_i f_i$
F007 STO Z	
F008 -28	
F009 STO I	保存目录寄存器 -28。
F010 RCL Z	
F011 STO+(I)	在寄存器 -28 中更新 $\sum x_i f_i$ 。
F012 RCL× X	$x_i^2 f_i$
F013 STO Z	
F014 -30	保存目录寄存器 -30。
F015 STO I	
F016 RCL Z	

程序行:
(ALG 模式)

说明

F017 STO+(I)	在寄存器 -30 中更新 $\sum x_i^2 f_i$ 。
F018 RCL B	
F019 STO+ N	增量 (或减量) N 。
F020 RCL N	
F021 RCL F	
F022 ABS	
F023 STO F	
F024 VIEW N	闲书数据对的当前数量。
F025 GTO I001	找出标签行数字 I 进行下一数据输入。
校和及长度: F6CB 84	

G001 LBL G	计算分组数据统计。
G002 s \times	分组标准差。
G003 STO S	
G004 VIEW S	显示分组标准差。
G005 \bar{x}	加权平均值。
G006 STO M	
G007 VIEW M	显示加权平均值。
G008 GTO I001	返回输入更多数据点。
校和及长度: DAF2 24	

U001 LBL U	取消数据输入错误。
U002 -1	输入减量 N 。
U003 STO B	
U004 RCL F	恢复最后一个数据频率输入。
U005 +/-	改变 f_i 的符号。
U006 STO F	
U007 GTO F001	调整计数及总和。
校和及长度: 03F4 23	

使用的标志:

无。

程序指令:

1. 键入程序调度程序; 完成后按 **[C]**。
2. 按 **[XEQ] [S] [ENTER]** 开始输入新数据。
3. 键入 x_i 值 (数据点), 按 **[R/S]**。
4. 键入 f_i 值 (频率), 按 **[R/S]**。
5. 查看所输入数据点的数量后按 **[R/S]**。
6. 针对每一个数据点重复步骤 3 至 5。

如果您发现您在步骤 4 中按下 **[R/S]** 后出现数据输入错误 (x_i 或 f_i), 先按 **[XEQ] [U] [ENTER]**, 然后再按 **[R/S]**。然后, 返回至步骤 3 输入正确数据。

7. 输入最后一个数据对后, 按 **[XEQ] [G] [ENTER]** 计算并显示分组标准差。
8. 按 **[R/S]** 显示分组数据的加权平均值。
9. 要想加入额外数据点, 按 **[R/S]**, 继续从步骤 3 开始执行。
要开始一个新问题, 从步骤 2 开始。使用的变量:

X	数据点。
F	数据输入频率。
N	数据对计数器。
S	分组标准差。
M	加权平均值。
i	用于间接定址正确的统计寄存器的目录变量。
寄存器 -27	总和 Σf_i 。
寄存器 -28	总和 $\Sigma x_i f_i$ 。
寄存器 -30	总和 $\Sigma x_i^2 f_i$ 。

示例:

输入以下数据, 计算分组标准差。

分组	1	2	3	4	5	6
x_i	5	8	13	15	22	37
f_i	17	26	37	43	73	115

键:	显示:	说明:
(在 ALG 模式下)		
XEQ S ENTER	X? 数值	提示输入第一个 x_i 。
5 R/S	F? 数值	将 5 存入 X, 提示第一个 f_i 。
1 7 R/S	N= 1.0000	将 17 存入 F; 显示计数器。
R/S	X? 5.0000	提示输入第二个 x_i 。
8 R/S	F? 17.0000	提示输入第二个 f_i 。
2 6 R/S	N= 2.0000	显示计数器。
R/S	X? 8.0000	提示输入第三个 x_i 。
1 4 R/S	F? 26.0000	提示输入第三个 f_i 。
3 7 R/S	N= 3.0000	显示计数器。

您错将 x_3 值 13 输入为 14。执行子程序 U 取消输入错误:

XEQ U ENTER	N= 2.0000	删除错误数据; 显示修改后的计数器。
R/S	X? 14.0000	提示输入新的第三个 x_i 。
1 3 R/S	F? 37.0000	提示输入新的第三个 f_i 。
R/S	N= 3.0000	显示计数器。
R/S	X? 13.0000	提示输入新的第四个 x_i 。
1 5 R/S	F? 37.0000	提示输入新的第四个 f_i 。

4 **3** **R/S**

R/S

2 **2** **R/S**

7 **3** **R/S**

R/S

3 **7** **R/S**

1 **1** **5** **R/S**

XEQ **G** **ENTER**

R/S

C

N=
4.0000
X?
15.0000
F?
43.0000
N=
5.0000
X?
22.0000
F?
73.0000
N=
6.0000
S=
11.4118
M=
23.4084
23.4084

显示计数器。

提示输入新的第五个 x_i 。

提示输入新的第五个 f_i 。

显示计数器。

提示输入新的第六个 x_i 。

提示输入新的第六个 f_i 。

显示计数器。

计算并显示6个数据点的分组标准差 (sx)。

计算并显示加权平均值 (\bar{x})。

清除 VIEW。

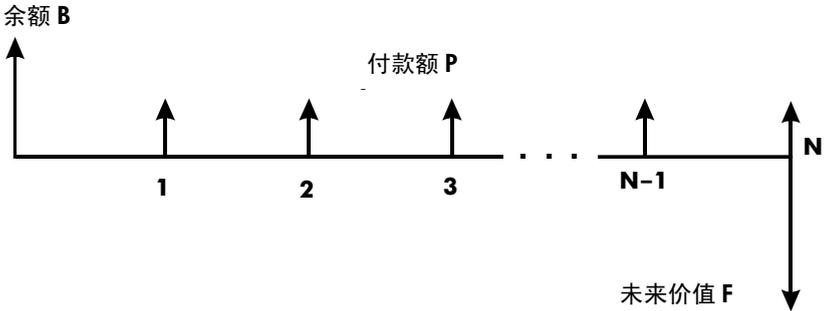
其余程序及方程

金钱的时间价值

给定 " 金钱的时间价值方程 "(TVM) 中五个数值中的任意四个, 您就可以解出第 5 个数值。本方程可用于多种金融应用领域, 如消费及住房贷款, 储蓄帐户等等。

TVM 方程为:

$$P \left[\frac{1 - (1 + I/100)^{-N}}{I/100} \right] + F(1 + (I/100))^{-N} + B = 0$$



现金价值符号 (余额 B ; 付款额 P ; 未来价值 F) 与现金流动方向相对应。您收到的款项的符号为正号, 而您支付款项的符号则为负号, 注意任何问题都可以从里哪个个方面加以考虑。借出人和借款人用相反的符号看待同一问题。

方程输入:

键入本方程:

$$P \times 100 \times (1 - (1 + I \div 100)^{-N}) \div I + F \times (1 + I \div 100)^{-N} + B$$

键:	显示:	说明:
(在 RPN 模式下)		
[EQN]	EQN LIST TOP	选择方程模式。
[RCL] [P] [×] [1] [0] [0]	或当前方程 P×100_	开始输入方程。
[×] [()] [1] [-]	P×100×(1-_)	
[()] [1] [+]	P×100×(1-(1+_)	
[RCL] [1] [÷] [1] [0] [0]	←0×(1-(1+I÷100)_	
[>] [y^x]	←(1-(1+I÷100)^_	
[+/-] [RCL] [N] [>]	←(1+I÷100)^-N)_	
[÷] [RCL] [1] [+]	←100)^-N)÷I+F×_	
[×]		
[()] [1] [+]	←^-N)÷I+F×(1+I)_	
[RCL] [1]	←I+F×(1+I÷100)_	
[÷] [1] [0] [0] [>]	←×(1+I÷100)^-N)_	
[y^x] [+/-] [RCL] [N]	←(1+I÷100)^-N+B_	
[+]		
[RCL] [B]		
[ENTER]	P×100×(1-(1+I÷	终止方程。
[☞] [SHOW] (按住不放)	CK=CEFA	校和及长度。
	LN=41	

备注:

TVM方程要求I必须为非0以避免DIVIDE BY 0错误。如果您正求解I值且不清楚其当前数值,在开始 SOLVE 计算 (**[☞] [SOLVE] [1]**) 前按 **[1] [☞] [STO] [1]**。

您被提示输入数值的顺序取决于您正求解的变量值。

SOLVE 指令:

1. 如果您的首个 TVM 计算是要解出利率 i ，按 **1** **→** **STO** **1**。
2. 按下 **EQN**。必要时，按 **↑** 或 **↓** 滚动查看方程列表，直到您看到 TVM 方程为止。
3. 进行以下 5 个运算中的任何一个：
 - a. 按 **→** **SOLVE** **N** 计算复利计算期。
 - b. 按 **→** **SOLVE** **I** 计算定期利息。

对于按月付款，返回的结果 I 为月利率 i ；按 **12** **×** 查看年利率。
 - c. 按 **→** **SOLVE** **B** 计算贷款或储蓄帐户的初始余额。
 - d. 按 **→** **SOLVE** **P** 计算定期付款。
 - e. 按 **→** **SOLVE** **F** 计算贷款的未来价值或余额。
4. 收到提示时键入 4 个未知变量的数值；按每个数值后的 **R/S**。
5. 当您按下最后一个 **R/S** 时，未知变量的数值即被计算并显示出来。
6. 要想计算一个新的变量，或重新用不同数据计算同一变量，返回至步骤 2。

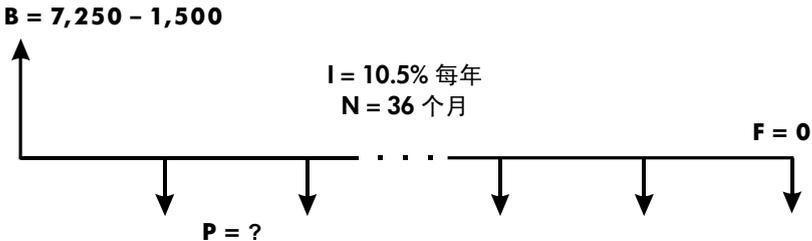
SOLVE 在本应用中在无初试猜测值的情况下也能有效发挥作用。

使用的变量:

- N 复利计算期的数量。
- i 以百分比表示的定期利率。(例如,如果年利率为 15%,每年的付款次数为 12 次,则定期利率 i 为 $15 \div 12 = 1.25\%$ 。)
- B 贷款或储蓄帐户的初始余额。
- P 定期付款。
- F 储蓄帐户或贷款余额的未来价值。

示例:

第一部分。您以按月计算年利率 10.5%, 贷款 3 年 (36 个月) 买车。车的购买价为 7250 美元。定金为 \$1500 美元。



键:	显示:	说明:
(在 RPN 模式下)		
DISPLAY 1 (FIX) 2		选择 FIX 2 显示格式。
EQN (必要时 ▼)	$P \times 100 \times (1 - (1 + I)^{-N}) \div I \rightarrow$	显示 TVM 方程的最左侧部分。
→ SOLVE P	$I?$	选择 P ; 提示输入 I 值。
	数值	
1 0 . 5 ENTER	$I?$	将您的年利率投入转换为同等
1 2 ÷	0.88	月利率。
R/S	$N?$	将 0.88 存入 I ; 求解 N 值。
	数值	
3 6 R/S	$F?$	将 36 存入 N , 计算 F 。
	数值	

0 **R/S**

B?

将 0 存入 F ；提示输入 B 值。

7 **2** **5** **0** **ENTER**

数值

B?

计算 B 初始贷款余额。

1 **5** **0** **0** **-**

5,750.00

R/S

SOLVING

将 5750 存入 B ；计算每月付款额 P 。

P=

-186.89

计算答案为负，原因是贷款是从借款人的角度来看待的。借款人收到的款项（初始余额）为正，而支付的款项则为负。

第二部分。哪一种利率会使每月付款额减少 10 美国呢？

键:	显示:	说明:
(在 RPN 模式下)		
[EQN]	$P \times 100 \times (1 - (1 + I)^{-N})$	显示 TVM 方程的最左侧部分？
[F2] [SOLVE] [1]	P? -186.89	选择 I；提示输入 P 值。
[F2] [RND]	P? -186.89	将付款额四舍五入至两个小数位。
[1] [0] [+]	P? -176.89	计算新的付款额。
[R/S]	N? 36.00	将 -176.89 存入 P；提示输入 N。
[R/S]	F? 0.00	将 36 保留在 N 中；提示输入 F。
[R/S]	B? 5,750.00	将 0 保留在 F 中；提示输入 B。
[R/S]	SOLVING I= 0.56	将 5750 保留在 B 中；计算月利率。
[1] [2] [X]	6.75	计算年利率。

第三部分。使用计算得出的利率 (6.75%)。假定两年后您将车卖掉，您的欠款余额将会是多少呢？换句话说，两年后的未来余额将会是多少呢？

注意第二部分的利率 I 不为 0，因此在计算新 I 时，您不会收到 **DIVIDE BY 0** 错误。

键:	显示:	说明:
(在 RPN 模式下)		
[EQN]	$P \times 100 \times (1 - (1 + I)^{-N})$	显示 TVM 方程的最左侧部分。
[F2] [SOLVE] [F]	P? -176.89	选择 F；提示输入 P 值。
[R/S]	I? 0.56	保留 P；提示输入 I。
[R/S]	N? 36.00	0.56 保留在 I 中；提示输入 N 值。

2 **4** **R/S**

B?
5,750.00

将 24 存入 N；求解 B 值。

R/S

SOLVING
F=
-2,047.05

将 5750 保留在 B 中；计算未来余额 F。符号再次为符，表示您必须付清这笔款项。

← **DISPLAY** **1**

设置 FX4 显示格式。

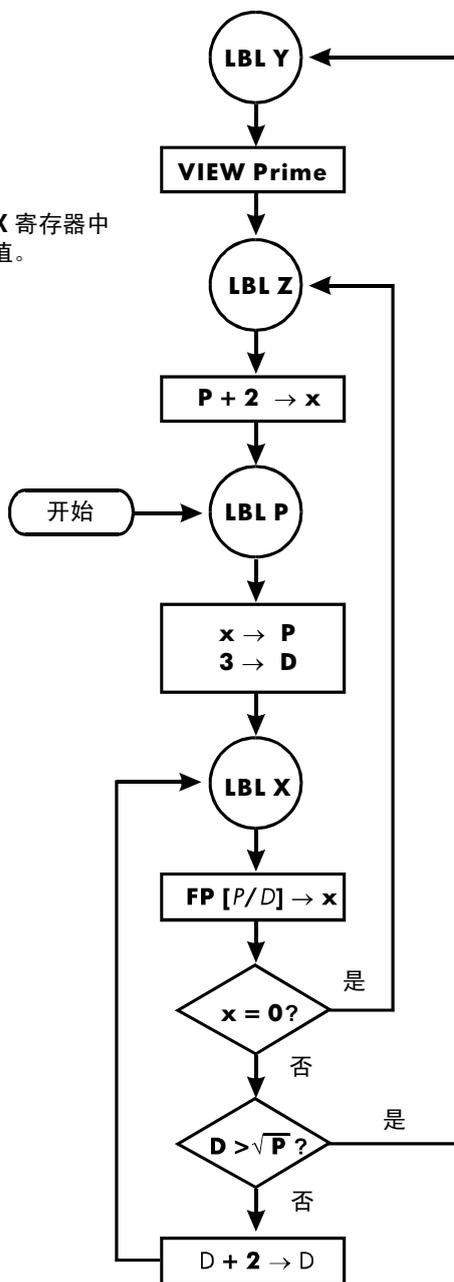
(1FIX) **4**

素数发生器

本程序接受任何大于 3 的正整数。如果数字为素数 (不能被其本身和 1 整除)，则程序将返回输入值。如果输入值不为素数，则程序将返回大于输入值的第一个素数。

程序通过逐一尝试所有可能的因数来辨别非素数。如果某个数字不是素数，程序将加 2(确认该数值仍然为奇数) 并进行测试看其是否已找到一个素数。这一过程将一直持续到找到一个素数。

注意: x 为 X 寄存器中的数值。



程序列举:

程序行: (ALG 模式)	说明
Y001 LBL Y	本子程序显示素数 P 。
Y002 VIEW P	
校和及长度: 2CC5 6	
Z001 LBL Z	本子程序使 P 加 2。
Z002 2+ P	
校和及长度: EFB2 9	
P001 LBL P	本子程序保存 P 的输入值。
P002 LASTx▶ P	
P003 FP(P÷2)	
P004 x<>y	
P005 0	
P006 x=y?	测试偶数输入
P007 1+P▶P	如果输入偶数则对 P 进行递增。
P008 3▶D	将 3 存入测试除数 D
校和及长度: EA89 47	
X001 LBL X	本子程序对 P 进行测试, 看其是否为素数。
X002 FP(P÷D)	求出 $P \div D$ 的分数部分。
X003 x=0?	测试余数为 0(不是素数)。
X004 GTO Z001	输入数字不是素数, 试一试下一种可能性。
X005 SQRT(P)	
X006 x<>y	
X007 D	
X008 x>y?	测试看看是否所有可能的因数都已试过。
X009 GTO Y001	如果所有的因数都已试过, 分路至显示子程序。
X010 2+D▶D	
X011 GTO X001	分路至带新因数的测试潜素数。
校和及长度: C6B5 53	

使用的标志:

无。

程序指令:

1. 键入程序调度程序; 完成后按 **C**。
2. 键入一个大于 3 的正整数。
3. 按 **XEQ** **P** **ENTER** 使程序运行。将显示素数 P 。
4. 要想查看下一个素数, 按 **R/S**。

使用的变量:

P 素数值及潜素数值。
 D 用于测试当前 P 值的除数。

备注:

未进行确保输入值大于 3 的测试。

示例:

789 后的第一个素数是多少? 下一个素数是多少?

键:	显示:	说明:
(在 ALG 模式下)		
7 8 9 XEQ	P=	计算 789 后的下一个素数。
P ENTER	797.0000	
R/S	P=	计算 797 后的下一个素数。
	809.0000	

矢量中的叉积

下例向您展示用程序函数计算叉积的方法。

叉积：

$$\mathbf{v}_1 \times \mathbf{v}_2 = (YW - ZV)\mathbf{i} + (ZU - XW)\mathbf{j} + (XV - YU)\mathbf{k}$$

式中

$$\mathbf{v}_1 = X\mathbf{i} + Y\mathbf{j} + Z\mathbf{k}$$

及

$$\mathbf{v}_2 = U\mathbf{i} + V\mathbf{j} + W\mathbf{k}$$

程序行： (RPN 模式)

```
R001 LBL R
R002 INPUT X
R003 INPUT Y
R004 INPUT Z
R005 GTO R001
校和及长度：D82E 15
```

说明

定义矩形输入 / 显示子程序的开始。
显示或接受 X 的输入值。
显示或接受 Y 的输入值。
显示或接受 Z 的输入值。
至 R001 输入矢量

```
E001 LBL E
E002 RCL X
E003 STO U
E004 RCL Y
E005 STO V
E006 RCL Z
E007 STO W
E008 GTO R001
校和及长度：B6AF 24
```

定义矢量输入子程序的开始。
将 X、Y 及 Z 中的数值分别复制到 U、V 和 W 中。
至 R001 输入矢量

程序行:
(RPN 模式)

说明

C001 LBL C	定义叉积子程序的开始。
C002 RCL Y	
C003 RCL× W	
C004 RCL Z	
C005 RCL× V	
C006 -	计算 $(YW - ZV)$, 即 X 组分。
C007 STO A	
C008 RCL Z	
C009 RCL× U	
C010 RCL X	
C011 RCL× W	
C012 -	计算 $(ZU - WX)$, 即 Y 组分。
C013 STO B	
C014 RCL X	
C015 RCL× V	
C016 RCL Y	
C017 RCL× U	
C018 -	
C019 STO Z	保存 $(XV - YU)$, 即 Z 组分。
C020 RCL A	
C021 STO X	保存 X 组分。
C022 RCL B	
C023 STO Y	保存 Y 组分。
C024 GTO R001	至 R001 输入矢量
校和及长度: 838D 72	

示例:

计算两个矢量 $v_1=2i+5j+4k$ 和 $v_2=i-2j+3k$ 的叉积

键:	显示:	说明:
XEQ R ENTER	X?	运行 R 子程序输入矢量值
1 R/S	v?	value 输入 x 组分的 v2
2 +/- R/S	z?	value 输入 y 组分的 v2
3 R/S	X?	value 输入 z 组分的 v2
XEQ E ENTER	X?	1 运行 E 子程序在 U、V 及 W 变量 中交换 v2
2 R/S	v?	1 输入 x 组分的 v1
5 R/S	z?	-2 输入 y 组分的 v1
4 R/S	X?	3 输入 z 组分的 v1
XEQ C ENTER	X?	2 运行 C 子程序计算
R/S	v?	23 叉积的 x 组分
R/S	z?	-2 计算叉积的 y 组分
		-9 计算叉积的 z 组分

第三部分

附录及参考文献

支持、电池及服务

计算器支持

您可以从 HP 的计算器支持部获得与您的计算器使用方法有关问题的答案。HP 的经验表明许多用户在使用 HP 的产品时遇到类似的问题，因此，本公司提供以下 "常见问题答案" 一节。如果您未找到您的问题答案，请与 A-8 页上所列的计算器支持部联系。

常见问题答案

Q: 本人该如何确定计算器是否正常工作？

A: 请参阅第 A-5 页，其中描述了计算器的诊断自检。

Q: 本人数字里的小数点为逗号而非句号，本人该如何恢复句号呢？

A: 使用  **DISPLAY**  (5.) 函数 (第 1-23 页)。

Q: 本人该如何修改显示屏上小数位的数量呢？

A: 使用  **DISPLAY** 菜单 (第 1-21 页)。

Q: 本人该如何将内存全部或部分清除？

A:  **CLEAR** 显示清除菜单，允许您清除 x(X 寄存器中的数字)、所有的直接变量、所有的内存、所有的统计数据、所有的栈级以及所有的间接变量

Q: 数字中的 "E" (例如， $2.51E-13$) 表示什么？

A: 10 的指数; 即 2.51×10^{-13} 。

Q: 计算器显示信息 **MEMORY FULL**, 本人该怎么做?

A: 开始前, 您必须清除一部分内存。(请参阅附录 B。)

Q: 为什么计算 π 半径的正弦 (或正切) 值时会显示一个非常小的数字而非 0?

A: π 无法用计算器的 12 位精确度准确表示。

Q: 为什么使用三角函数时本人得到的答案是错误的?

A: 您必须确定计算器使用的是正确的角模式 (**MODE** **1DEG**, **2RAD** 或 **3GRD**)。

Q: 显示菜单中的 *信号器* 代表什么?

A: 信号器代表计算器的某种状态信息。见第一章 "信号器"。

Q: 数字作为分数显示。本人该如何获取小数呢?

A: 按  **FDISP**。

环境限值

要想保证产品可靠性, 应遵守以下温度及湿度限值:

- 工作温度: 0~45°C (32~113°F)。
- 存储温度: -20~65°C (-4~149°F)。
- 工作及存储湿度: 相对湿度 90%, 最高温度 40°C (104°F)。

更换电池

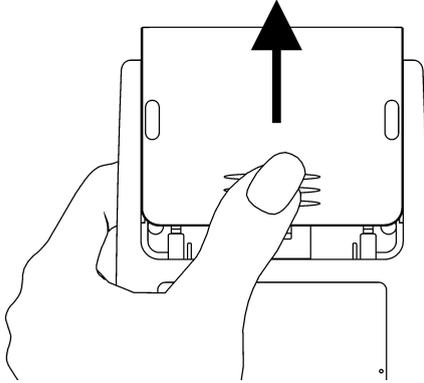
计算器由两个 3V 锂电池 CR2032 提供电源。

出现低电量信号器 () 时应尽快更换电池。如果电池信号器打开且显示屏变的昏暗，您可能会丢失数据。如果数据丢失，则会显示 **MEMORY CLEAR** 信息。

将电池取下后，应在两分钟种将新电池换上以免丢失保存的信息。(打开电池盖前，请将新电池就手准备好。)

要想安装电池：

- 1.** 将两个新纽扣电池就手准备好。避免接触电池的端子 - 只可触摸其边缘。
- 2.** 确认计算器处于关闭状态。在整个充电过程完成之前，请勿再次按下打开 ()。如果电池取下后计算器仍处于打开状态，连续内存的内容将被擦除。
- 3.** 将计算器翻个面，滑动电池盖将其取下。



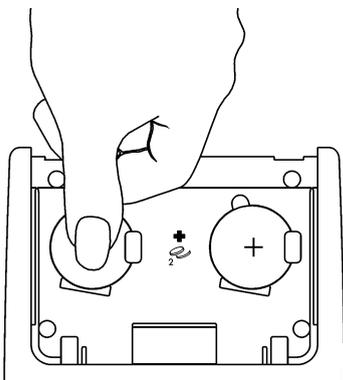
- 4.** 要想防止内存损失，切勿将两个旧电池同时取下。务必一次取下并更换一个电池。

警告



不要将电池切断、刺破或者在将其放入火中处理。电池会爆裂或爆炸，从而释放出有毒化学物质。

5. 安上新 CR2032 锂电池，确认正极符号 (+) 朝上。



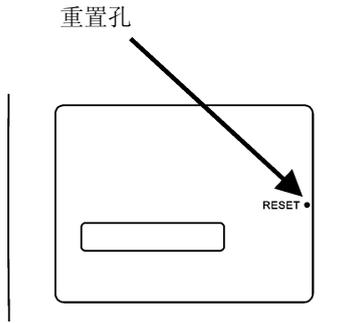
6. 依据步骤 4 至 5 取下并安上另外一个电池。确认每个电池的正极符号 (+) 朝外。
7. 更换电池盒盖。
8. 按下 **C**。

测试计算器运行

使用以下导则确定计算器是否正常工作。每执行一步即对计算器进行测试，看其运行是否已恢复。如果您的计算器需要维修，请参阅第 A-8 页。

- 计算器无法打开 (步骤 1-4) 或在您输入按键 (步骤 1-3) 时无响应:
 1. 重置计算器。按下 **C** 键不放，然后按下 **GTO**。可能需要多次重复这些重置键盘键。
 2. 擦除内存。按下并保持 **C** 不放，然后按下并保持 **R/S** 和 **i** 不放。当您释放这三个键时，内存将被清除并显示 MEMORY CLEAR 信息。

3. 取下电池 (见 "更换电池"), 轻按紧贴计算器中两个电池触点的电池。更换电池, 打开计算器。计算器应显示 **MEMORY CLEAR**。
4. 如果计算器仍然不对键盘键做出响应, 使用一个薄而尖的物体按重置孔。所保存的数据保持完好。



如果这些步骤未能恢复计算器运行, 则需要对其进行维修。

- 如果计算器对键盘键做出响应, 但是您怀疑计算器出了故障:
 1. 进行下一节说述的自检。如果计算器未通过自检, 则需要对其进行维修。
 2. 如果计算器通过自检, 您可能在操作计算器时出错。重新阅读本手册相关部分并查阅 "常见问题答案"(第 A-1 页)。
 3. 联系第 A-8 页上的计算器联系部。

自检

如果显示器可以打开, 但是计算器运行似乎不正常, 进行以下诊断自检。

1. 按下 **[C]** 键不放, 同时按 **[XEQ]**。
2. 按任意键 8 次, 观察各种不同的显示模式。在把键按 8 次后, 计算器先显示版权信息 © 2007 HP DEV CO. L. P., 然后显示信息 **KBD 01**。
3. 按以下顺序按键:

R/S → **GTO** → **XEQ** → **MODE** → **^** → **<** → **>** → **RCL** → **R↓** →
x↔y → **i** → **∇** → **SIN** → **COS** → **TAN** → **√x** → **y^x** → **1/x** →
ENTER → **+/-** → **E** → **()** → **←** → **EQN** → **7** → **8** → **9** → **÷** → **↵** → **4**
 → **5** → **6** → **x** → **↵** → **1** → **2** → **3** → **-** → **C** → **0** → **.** →
Σ+ → **+**

- 在您以正确顺序按下各键且其正常工作后，计算器显示 **KBD**，其后紧跟 2 位数字。(计算器以 16 进制基底对按键进行计数)。
- 如果您不按正确顺序按键或某个键工作不正常，下一个键盘键显示故障信息(见步骤 4)。

4. 诊断自检产生两个结果之中的一个：

- 计算器若通过自检则显示 **35S-OK**。至步骤 5。
- 计算器如果未通过自检，则显示 **35S-FAIL**，其后紧跟一个一位数字。如果您因为不按顺序按键而收到信息，重新设置计算器(按下 **C** 不放，然后按 **GTO**) 并再次进行自检。如果您按正确顺序按键，但还是收到本信息，重复自检对结果进行验证。如果计算器再次未能通过自检，则需要对其进行维修(见第 A-8 页)。在将计算器寄出进行维修时，请连同计算器一起附上故障信息。

5. 要想退出自检，重置计算器(按下 **C** 不放，然后按 **GTO**)。

按 **C** 和 **MODE** 开始在出厂时采用的连续自检。您可以按下任意键终止本出厂测试。

保修

HP 35s 科学计算器；保修期：12 个月

1. 惠普向终端使用客户担保，自购买本公司的产品以后，在上述指定期限内因为原料和工艺出现的缺陷，惠普将会提供免费的硬件，附件以及替代供应品。如果惠普在担保期限内收到这种缺陷的通知，它将有权选择，或者维修或者替换证实确实存在缺陷的产品。替换的产品可能是新的，或者和新的相差不多。
2. 惠普向您担保，本公司的软件自购买以后，在上述指定担保期限内，只要合理的安装和使用，不会由于原材料和工艺的缺陷而停止执行程序的指令。如果惠普在担保期限内收到此类缺陷的通知，它将替换由于此类缺陷而不能执行程序指令的软件媒体。
3. 惠普不能担保其产品的运行将一直连续或者不会出错。如果惠普未能在一个合理的时间范围内，维修或者替换任何符合担保条件的产品，您将有权持购买证明退回产品获得退款。
4. 惠普的产品可能会装有与新部件同等功效的部分改制部件或者可能被偶然使用过。
5. 担保不适用于由下列原因引起的缺陷 (a) 不当或者不充备的保养或校准，(b) 软件，界面连接，非惠普提供的部件或者供应品，(c) 未经授权的修改或者错用，(d) 针对产品公开发表的环境说明书以外的操作，或者 (e) 不适当的场所准备或保养。
6. 惠普没有任何书面或口头的其它明确担保。在当地法律允许的范围内，任何隐性的担保或商品满意品质的条件或对某种特殊目的的合适度，都仅限于上述明确担保期间。一些国家，地区或者省份不允许对隐性担保的时间限制，因此上述限制或排除性要求对您可能并不适用。该担保赋予您特定的法律权利，您也可以拥有其他因国家，地区和省份不同而有所变化的权利。

7. 在当地法律允许的范围内，本担保声明中的法律救济是您唯一的独享的救济。除上述之外，无论在任何事件中，以及不管是否基于合同，侵权，或者其他原因，惠普或它的供应者都不再对数据的丢失或者直接的，特殊的，偶然的，派生的（包括利益和数据流失），以及其他损失负有责任。一些国家，地区或者省份不允许对于偶然或派生损失的排除或限制性要求，所以上述限制或排除可能并不适用于您。
8. 惠普产品和服务的唯一担保由伴随这些产品和服务的明确担保声明所规定。惠普不负责技术性或者校对性错误或者本处省略的错误。

对于在澳大利亚和新西兰交易的消费者而言：本声明中的担保条款，除了法律允许的范围之外，并没有排除，限制或者修改而是包括适用于本产品出售给你的强制性法定权利。

客户支援

AP

国家：	电话号码
澳洲	1300-551-664 或 03-9841-5211
中国	010-68002397
香港	2805-2563
印尼	+65 6100 6682
日本	+852 2805-2563
马来西亚	+65 6100 6682
纽西兰	09-574-2700

菲律宾	+65 6100 6682
新加坡	6100 6682
南韩	2-561-2700
台湾	+852 2805-2563
泰国	+65 6100 6682
越南	+65 6100 6682

EMEA

国家:	电话号码
澳洲	01 360 277 1203
比利时	02 620 00 86
比利时	02 620 00 85
捷克	296 335 612
丹麦	82 33 28 44
芬兰	09 8171 0281
法国	01 4993 9006
德国	069 9530 7103
希腊	210 969 6421
荷兰	020 654 5301
爱尔兰	01 605 0356
意大利	02 754 19 782
卢森堡	2730 2146
挪威	23500027
葡萄牙	021 318 0093
俄国	495 228 3050
南非	0800980410
西班牙	913753382
瑞典	08 5199 2065
瑞士 (法国)	022 827 8780
瑞士 (德国)	01 439 5358
瑞士 (意大利)	022 567 5308
英国	0207 458 0161

国家:	电话号码
安哥拉	1-800-711-2884
安地卡及巴布达	1-800-711-2884
阿根廷	0-800- 555-5000
阿鲁巴群岛	800-8000 ♦ 800-711-2884
巴哈马	1-800-711-2884
巴贝多	1-800-711-2884
百慕达	1-800-711-2884
玻利维亚	800-100-193
巴西	0-800-709-7751
唯京群岛	1-800-711-2884
开曼群岛	1-800-711-2884
库拉索群岛	001-800-872-2881 + 800-711-2884
智利	800-360-999
哥伦比亚	01-8000-51-4746-8368 (01-8000-51- HP INVENT)
哥斯大黎加	0-800-011-0524
多明尼克	1-800-711-2884
多明尼加共和国	1-800-711-2884
厄瓜多尔	1-999-119 ♦ 800-711-2884 Andinatel(安地那电话公司) 1-800-225-528 ♦ 800-711-2884 Pacifitel(太平洋电话公司)
萨尔瓦多	800-6160
法属安地列斯群岛	0-800-990-011 ♦ 800-711-2884
法属圭亚那	0-800-990-011 ♦ 800-711-2884
格瑞纳达	1-800-711-2884
哥德洛普岛	0-800-990-011 ♦ 800-711-2884
瓜地马拉	1-800-999-5105
盖亚那	159 ♦ 800-711-2884

海地	183 ♦ 800-711-2884
宏都拉斯	800-0-123 ♦ 800-711-2884
牙买加	1-800-711-2884
马提尼克	0-800-990-011 ♦ 877-219-8671
墨西哥	01-800-474-68368 (800 HP INVENT)
蒙特塞略	1-800-711-2884
荷属安地列斯群岛	001-800-872-2881 ♦ 800-711-2884
尼加拉瓜	1-800-0164 ♦ 800-711-2884
巴拿马	001-800-711-2884
巴拉圭	(009) 800-541-0006
秘鲁	0-800-10111
波多黎各	1-877 232 0589
圣露西亚	1-800-478-4602
圣文森	01-800-711-2884
圣基茨和尼维斯	1-800-711-2884
圣马丁	1-800-711-2884
苏利南莫河	156 ♦ 800-711-2884
千里达托贝哥	1-800-711-2884
特克斯和凯科斯群岛	01-800-711-2884
美国维尔京群岛	1-800-711-2884
乌拉圭	0004-054-177
委内瑞拉	0-800-474-68368 (0-800 HP INVENT)

NA

国家:	电话号码
加拿大	800-HP-INVENT
美国	800-HP INVENT

请登陆 <http://www.hp.com> 查看最新的服务及支持信息。

规则信息

联邦通讯委员会

根据 FCC 条例第 15 节，本产品已经过测试并且符合 B 级数字装置的限制。这些限制设计的目的就是防止在居住安装中的有害干扰而提供合理保护。本产品会产生电波，如果没有根据说明正确安装会对通信产生干扰。然而，这并不能完全保证在特定安装时干扰不会发生。如果在开启和关闭该产品时产生对收音和电视接收装置的干扰，使用者可以通过下列方式调整：

- 重新调整天线的方向或者重新放置天线。
- 隔离该产品和其它接收器。
- 该产品不要与其它接收器连接在同一个出口上。
- 请咨询经销商或相关技术人员。

更正

未经惠普公司许可，FCC 要求使用者不得改变，修改该装置。否则取消使用者使用权。

针对带 FCC 标识产品的“一致性声明”，仅限于美国

本设备部分依从 FCC 规定的第 15 节。操作符合以下两种条件：(1) 本设备不会造成有害干扰，并且(2) 必须接受任何受到的干扰，包括可能导致不需要操作的干扰。

与产品有关，但与本申请无关的问题，请寄至：

Hewlett-Packard Company

P. O. Box 692000, Mail Stop 530113

Houston, TX 77269-2000

有关本 FCC 声明的问题，请寄至：

Hewlett-Packard Company

P. O. Box 692000, Mail Stop 510101

Houston, TX 77269-2000

or call HP at 281-514-3333

验证产品请参见产品上的部件、序列或者模式号码。

加拿大通告

本 B 类数字设备符合加拿大致干扰设备条例的所有要求。

A-12 支持、电池及服务

Avis Canadien

Cet appareil numérique de la classe B respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

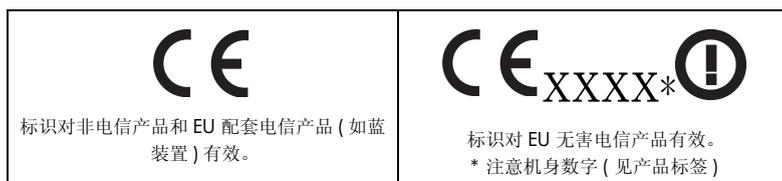
欧盟规则通告

该产品符合下列 EU 指示：

- 低电压指令 2006/95/EC
- EMC Directive 2004/108/EC

这些指示的符合表明与欧洲标准 (European Norms) 相符，可见惠普公司发表的关于产品或产品组的 EU 一致条例。

条例符合信息可见产品标识：



Hewlett-Packard GmbH, HQ-TRE, Herrenberger Strasse 140, 71034 Boeblingen, Germany

日本通告

この装置は、情報処理装置等電波障害自主規制協議会 (VCCI) の基準に基づくクラス B 情報技術装置です。この装置は、家庭環境で使用することを目的としていますが、この装置がラジオやテレビジョン受信機に近接して使用されると、受信障害を引き起こすことがあります。

取扱説明書に従って正しい取り扱いをしてください。

欧共同体用户在个人家庭中的废弃设备处理



产品上或包装上的该符号表示该产品废弃后应另行处理。您应该将废弃设备送交到专门的电子产品回收点，用于废弃物循环使用处理。废弃产品的回收利用节约资源，保护人类健康以及环境卫生。要了解废品回收的更多信息请联系所在城市相关部门或垃圾回收服务站或产品购买处。

过氯酸物质 - 适用特殊处理

本计算器的内存备用电池可能含有过氯酸物质并在加州回收及处理时需要采取特殊处理。

产品中有毒有害物质或元素的名称及含量 根据中国《电子信息产品污染控制管理办法》

部件名称	有毒有害物质或元素					
	铅 (Pb)	汞 (Hg)	镉 (Cd)	六价铬 (Cr(VI))	多溴联苯 (PBB)	多溴二苯醚 (PBDE)
PCA	X	○	○	○	○	○
外观壳子/字键	○	○	○	○	○	○

○：表示该有毒有害物质在该部件所有均质材料中的含量均在SJ/T11363-2006 标准规定的限量要求以下。
X：表示该有毒有害物质至少在该部件的某一均质材料中的含量超出SJ/T11363-2006 标准规定的限量要求。

表中标有“X”的所有部件都符合欧盟 RoHS 法规
“欧洲议会和欧盟理事会 2003 年 1 月 27 日关于电子电器设备中限制使用某些有害物质的 2002/95/EC 号指令”

注：环保使用期限的参考标识取决于产品正常工作的温度和湿度等条件

用户内存及栈

本附录讲述

- 用户内存的分配及要求，
- 重置计算器而不影响内存的方法，
- 清除 (清理) 所有的用户内存并重置系统默认值的方法，及
- 影响栈提升的操作。

管理计算器内存

HP 35s 的 30KB 用户内存可供您对所保存的数据 (变量、方程及程序行) 进行任意组合。SOLVE, ∫ FN 及统计计算也需要用户内存。(∫ FN 运算运行起来特别 “昂贵”)。

所有保存的数据都保留到您明确将其清除为止。信息 **MEMORY FULL** 意味着当前无足够内存可用于您刚刚试图进行的操作。您需要清除部分 (或全部) 用户内存。例如，您可以：

- 清除任一或所有方程 (见第六章 “编辑并清除方程”)。
- 清除任一或所有程序 (见第十三章 “清除一个或多个程序”)。
- 清除所有用户内存 (按  **CLEAR** **3** (**3ALL**))。

要想查看可用内存有多少，按  **MEM**。显示屏显示可用内存字节数。

要想查看方程列表中特定方程的内存要求：

1. 按 **[EQN]** 激活方程模式。(将显示 **EQN LIST TOP** 或当前方程的左侧。)
2. 必要时，滚动查看方程列表(按 **[^]** 或 **[v]**)，直到您找到所需的方程为止。
3. 按 **[↵] [SHOW]** 查看方程的校验和(十六进制)及长度(单位为字节)。例如，**CK=382E LN=41**。

要想查看特定程序的总内存要求：

1. 按 **[↵] [MEM] [2] (2PGM)** 显示方程列表中的第一个标签。
2. 滚动查看方程列表(按 **[^]** 或 **[v]** 直到您找到所需的方程标签及大小为止)。例如，**LBL F LN=57**。
3. 可选操作：按 **[↵] [SHOW]** 查看程序的校验和(十六进制)及长度(单位为字节)。例如，**CK=9CC9 LN=57**。

要想查看程序中某个方程的内存要求：

1. 显示包括方程的程序行。
2. 按 **[↵] [SHOW]** 查看校验和及长度。例如，**CK=AB71 LN=15**。

重置计算器

如果计算器不响应键盘键或其工作出现异常，常识重新对其进行设置。重置计算器将终止当前计算并取消程序输入、数字输入、正在运行的程序、**SOLVE** 计算、**f FN** 计算，**VIEW** 显示或 **INPUT** 显示。所保存的数据保持完好。

要想重新设置计算器，按下 **[C]** 不放，然后按 **[GTO]**。如果您无法重新对计算器进行设置，尝试安装新电池。如果计算器无法重新设置或正常工作，您应该尝试用下一节所述的特殊程序清除内存。

如果计算器仍然不对键盘键做出响应，使用一个薄而尖的物体按重置孔。

计算器在摔到地上或电源中断时可自行重新进行设置。

清除内存

常用的内存清除方法就是按 **☞** **CLEAR** **3** (**3ALL**)。但是，另外有一种强大的清除程序可重置附加信息并在键盘工作不正常时特别有用。

如果计算器未能响应键盘键而您又无法通过对其重新进行设置或更换电池来恢复操作，尝试以下内存清除程序。这些键盘键可清除所有的内存、重新对计算器进行设置并恢复所有格式及模式的初始默认设置（如下所示）：

1. 按下并保持 **C** 键不放。
2. 按下并保持 **R/S**。
3. 按下 **i**。（您将同时按三个键）。当将释放全部三个键后，如果操作成功，显示屏将显示 **MEMORY CLEAR**。

目录	清除全部	内存清除 (默认)
角模式	不变	角度
基底模式	不变	小数
对比度设置	不变	中等
小数点	不变	“.”
千分位	不变	“1,000”
分母 (/c 值)	不变	4095
显示格式	不变	FIX 4
标志	不变	清除
复模式	不变	xiy
分数显示模式	不变	关闭
随机数种子	不变	0
方程指针	方程列表顶部	方程列表顶部
方程列表	清除	清除
FN= 标签	空	空
程序指针	程序顶部	程序顶部
程序内存	清除	清除
栈提升	启用	启用
栈寄存器	清 0	清 0
变量	清 0	清 0
间接变量	未定义	未定义
逻辑	不变	RPN

计算器摔在地上或电源中断时，计算器内存可能会被无意清除。

栈提升状态

4 个栈寄存器始终存在，而且栈始终呈现一种 *栈提升状态*。也就是说，当把下一个数字放入 X 寄存器时，栈提升始终是关于其性能 *启用或禁用* 的。(请参阅第 2 章 -“自动内存栈”)

除了以下两个列表中的函数外，其它所有函数都支持栈提升。

禁用操作

5种操作 **ENTER**, **Σ+**, **Σ-**, **↵ CLEAR 1** (1X)和 **↵ CLEAR 5** (5STK)都不支持栈提升。在这5种禁用操作中的任何一种之后键入的数字将X寄存器中的当前数字覆盖。Y、Z及T寄存器保持不变。

此外,当 **C** 和 **←** 起与 CLx 一样的作用时,它们也不支持栈提升。

INPUT函数不支持栈提升就如同它终止提示程序一样(因此,任何输入的数字都将覆盖 X 寄存器),但是它在程序恢复时支持栈提升。

中性操作

以下操作不影响栈提升的状态:

DEG, RAD, GRAD	FIX, SCI, ENG, ALL	DEC, HEX, OCT, BIN	CLVARS
PSE	SHOW	RADIX . RADIX ,	CLΣ
OFF RCL +	R/S 和停止	^ 和 v	C * 和 ← *
MEM 1 (1VAR)**	MEM 2 (2PGM)**	GTO . .	GTO . 标签 nnn
EQN	FDISP	错误	PRGM 和程序输入
切换二进制窗口	数字输入	xiy rθa	UNDO
* 除非是在用作 CLx 时。			
**包括显示目录时的所有操作, {VAR} ENTER 和 {PGM} XEQ 除外。它们都支持栈提升。			

LAST X 寄存器状态

以下操作以 RPN 模式将 x 存入 LAST X 寄存器:

$+, -, \times, \div$	$\sqrt{x}, x^2,$	$e^x, 10^x$
LN, LOG	$y^x, \sqrt[x]{y}$	$1/x, \text{INT}\div, \text{Rmdr}$
SIN, COS, TAN	ASIN, ACOS, ATAN	$\hat{x} \hat{y}$
SINH, COSH, TANH	ASINH, ACOSH, ATANH	IP, FP, SGN, INTG, RND, ABS
%, %CHG	$\Sigma+, \Sigma-$ HMS \rightarrow , \rightarrow HMS	RCL+, -, \times, \div \rightarrow DEG, \rightarrow RAD
nCr nPr	!	ARG
CMPLEX $+, -, \times, \div$	CMPLEX $e^x, \text{LN}, y^x, 1/x$	CMPLEX SIN, COS, TAN
\rightarrow kg, \rightarrow lb	\rightarrow $^{\circ}\text{C}$, \rightarrow $^{\circ}\text{F}$	\rightarrow cm, \rightarrow in
\rightarrow l, \rightarrow gal	\rightarrow KM \rightarrow MILE	

注意 /c 不影响 LAST X 寄存器。

恢复 - 算术顺序 **[X] [RCL] [+]** 变量将 x 存入 LASTx 而 **[X] [RCL] 变量 [+]** 则将恢复的数字存入 LASTx。

在 ALG 模式下, LAST X 寄存器与栈相伴; 它保存作为最后一个表达式结果的数字并支持以 ALG 模式使用先前的表达式结果。

访问栈寄存器内容

保存在 4 个栈寄存器 X、Y、Z 和 T 中的数值都可用 REGX、REGY、REGZ 和 REGT 指令在方程或程序中以 RPN 模式访问。

要想使用这些指令，先按 **[EQN]**。然后按 **[R]** 在显示屏上产生一个菜单，显示 X-、Y-、Z- 和 T- 寄存器。按 **[>]** 或 **[<]** 将移动下划线符号，显示当前选择的寄存器。按 **[ENTER]** 将把指令放入恢复所选栈寄存器数值供进一步使用的程序或方程。这些指令显示为 REGX、REGY、REGZ 及 REGT。

例如，通过先按 **[EQN]**，然后按指令 REGX x REGY x REGZ x REGT 输入的程序行将对 4 个栈寄存器中数值的积进行计算并将结果放入 X 寄存器。计算器将把先前的 X、Y 及 Z 数值留在 Y、Z 及 T 的栈寄存器中。

用这种方式有效使用栈中的数值是可能的，但在 HP 35s 上则不一定能够实现。

ALG: 概述

关于 ALG

本附录总结了 ALG 模式部分特有的功能，包括，

- 两变量算术
- 指数函数和对数函数 ( 10^x ,  LOG,  e^x ,  LN)
- 三角函数
- 数字部分
- 查阅栈
- 带复数的运算
- 对方程进行积分
- 基底 2、8 和 16 中的算术运算。
- 输入统计双变量数据。

按 **MODE** **4** (**4ALG**) 将计算器设置为 ALG 模式。当计算器处于 ALG 模式时，ALG 信号器亮起。

在 ALG 模式下，运算按以下顺序进行。

1. 括号表达中的表达式。
2. 因数 (!) 函数要求您在按  前输入数值。
3. 要求在按函数键，如 COS, SIN, TAN, ACOS, ASIN, ATAN, LOG, LN, x^2 , $1/x$, \sqrt{x} , π , $\sqrt[3]{x}$, %, RND, RAND, IP, FP, INTG, SGN, nPr, nCr, %CHG, INT \div , Rmdr, ABS, e^x , 10^x 单位转换后输入数值的函数。

4. $\sqrt[n]{y}$ 和 y^x 。
5. 一进制减 +/-
6. \times, \div
7. $+, -$
8. $=$

在 ALG 模式下进行两自变量算术运算

对于用 ALG 模式进行算术运算的探讨取代以下受 ALG 模式影响的部分。两自变量算术运算受 ALG 模式的影响：

- 简单算术运算
- 幂函数 (y^x , $\sqrt[n]{y}$)
- 百分比计算 ($\%$ 或 \rightarrow $\%CHG$)
- 排列 / 组合 (\leftarrow nCr , \rightarrow nPr)
- 除法运算的商和余数 (\leftarrow $INTG$ 2 ($2INTG\div$), \leftarrow $INTG$ 3 ($3Rmdr$))

简单算术运算

以下为一些简单算术运算的例子。

在 ALG 模式下，先输入第一个数字，接着按运算符 ($+$, $-$, \times , \div)，再输入第二个数字，最后按 ENTER 键。

要想计算：

12 + 3

按：

$\boxed{1} \boxed{2} \boxed{+} \boxed{3} \boxed{\text{ENTER}}$

显示：

12+3
15.0000

12 - 3

$\boxed{1} \boxed{2} \boxed{-} \boxed{3} \boxed{\text{ENTER}}$

12-3

9.0000

12 \times 3

$\boxed{1} \boxed{2} \boxed{\times} \boxed{3} \boxed{\text{ENTER}}$

12 \times 3

36.0000

12 \div 3

$\boxed{1} \boxed{2} \boxed{\div} \boxed{3} \boxed{\text{ENTER}}$

12 \div 3

4.0000

幂函数

在 ALG 模式下，要想计算升至 x 次幂的数字 y ，先键入 y y^x x ，然后按 $\boxed{\text{ENTER}}$ 。

要想计算：	按：	显示：
12^3	$\boxed{1} \boxed{2} \boxed{y^x} \boxed{3} \boxed{\text{ENTER}}$	$12^{\wedge}3$ 1,728.0000
$64^{1/3}$ (立方根)	$\boxed{\leftarrow} \boxed{y^x} \boxed{3} \boxed{\>} \boxed{6} \boxed{4} \boxed{\text{ENTER}}$	$\text{XROOT}(3,64)$ 4.0000

百分比计算

百分比函数。 $\boxed{\%}$ 键将数字除以 100。

要想计算：	按：	显示：
200 的 27%	$\boxed{\leftarrow} \boxed{\%} \boxed{2} \boxed{0} \boxed{0} \boxed{\>} \boxed{2} \boxed{7} \boxed{\text{ENTER}}$	$\%(200,27)$ 54.0000
200 减 27%	$\boxed{2} \boxed{0} \boxed{0} \boxed{-} \boxed{\leftarrow} \boxed{\%} \boxed{2} \boxed{7} \boxed{\>} \boxed{2} \boxed{7} \boxed{\text{ENTER}}$	$200-\%(200,27)$ 146.0000
25 加 12%	$\boxed{2} \boxed{5} \boxed{+} \boxed{\leftarrow} \boxed{\%} \boxed{2} \boxed{5} \boxed{\>} \boxed{1} \boxed{2} \boxed{\text{ENTER}}$	$25+\%(25,12)$ 28.0000

要想计算：	按：
$x\%$ 的 y	$\boxed{\leftarrow} \boxed{\%} y \boxed{\>} x \boxed{\text{ENTER}}$
从 y 到 x 的百分比变化。 ($y \neq 0$)	$\boxed{\leftarrow} \boxed{\%CHG} y \boxed{\>} x \boxed{\text{ENTER}}$

示例：

假定该成本为 \$15.76 的物品去年的价格为 \$16.12，那么今年的价格与去年的价格相比百分比变化为多少呢？

键：	显示：	说明：
$\boxed{\leftarrow} \boxed{\%CHG} \boxed{1} \boxed{6} \boxed{\cdot}$ $\boxed{1} \boxed{2} \boxed{\>} \boxed{1} \boxed{5} \boxed{\cdot}$ $\boxed{7} \boxed{6} \boxed{\text{ENTER}}$	$\%CHG(16.12,15.7$ -2.2333	与去年的价格相比，今年的价格下降了 2.2%。

排列 / 组合

示例： 人员组合。

某公司雇有 14 名女性员工和 10 名男性员工，正欲组建一个 6 人安全委员会。不同的人员组合可能会有多少种呢？

键：	显示：	说明：
$\left[\leftarrow \right]$ $\left[nCr \right]$ $\left[2 \right]$ $\left[4 \right]$ $\left[\rightarrow \right]$	$nCr(24,6)$	可能的组合总数。
$\left[6 \right]$ $\left[\text{ENTER} \right]$	134,596.0000	

除法运算的商和余数

您可以用 $\left[\leftarrow \right]$ $\left[\text{INTG} \right]$ $\left[2 \right]$ ($2\text{INTG} \div$) 及 $\left[\leftarrow \right]$ $\left[\text{INTG} \right]$ $\left[3 \right]$ (3Rmdr) 得出涉及两个整数的除法运算的商或余数。

$\left[\leftarrow \right]$ $\left[\text{INTG} \right]$ $\left[2 \right]$ ($2\text{INTG} \div$) 整数 1 $\left[\rightarrow \right]$ 整数 2。 $\left[\text{ENTER} \right]$

$\left[\leftarrow \right]$ $\left[\text{INTG} \right]$ $\left[3 \right]$ (3Rmdr) 整数 1 $\left[\rightarrow \right]$ 整数 2。 $\left[\text{ENTER} \right]$

示例：

要想显示 $58 \div 9$ 的商和余数

键：	显示：	说明：
$\left[\leftarrow \right]$ $\left[\text{INTG} \right]$ $\left[2 \right]$ ($2\text{INTG} \div$)	$\text{IDIV}(58,9)$	显示商。
$\left[5 \right]$ $\left[8 \right]$ $\left[\rightarrow \right]$ $\left[9 \right]$ $\left[\text{ENTER} \right]$	6.0000	
$\left[\leftarrow \right]$ $\left[\text{INTG} \right]$ $\left[3 \right]$ (3Rmdr)	$\text{RMDR}(58,9)$	显示余数。
$\left[5 \right]$ $\left[8 \right]$ $\left[\rightarrow \right]$ $\left[9 \right]$ $\left[\text{ENTER} \right]$	4.0000	

括弧计算

当您想要推迟至输入更多数字后再对中间结果进行计算，应使用括弧。例如，假设您想要计算：

$$\frac{30}{85-12} \times 9$$

当您键入 **3 0 ÷ (8 5 - 1 2) × 9** 时，计算器将计算出结果 -107.6471。但是，这不是您想要的结果。要想把除法运算推迟至85减12之后再进行，应使用括弧：

键：	显示：	说明：
3 0 ÷ () 8 5 -	30÷(85-)	未进行任何计算。
1 2)	30÷(85-12)	计算 85 - 12。
× 9	30÷(85-12)×9	计算 30/73。
ENTER	30÷(85-12)×9 3.6986	计算 30/(85 - 12) × 9。

您可以省去左括弧前的乘号 (×)。默认乘法运算在方程模式下不可用。例如，表达式 $2 \times (5 - 4)$ 可以输入为 **2 () 5 - 4**，无需在 2 和左括弧之间加上 **×** 键。

指数函数和对数函数

要想计算：	按：	显示：
自然函数 (基数为 e)	LN 1 ENTER	LN(1) 0.0000
常用对数 (基数为 10)	LOG 1 0 ENTER	LOG(10) 1.0000
自然指数	e^x 2 ENTER	EXP(2) 7.3891
常用指数 (反对数)	10^x 2 ENTER	ALOG(2) 100.0000

三角函数

假设角度的单位为 **MODE** **1** (1DEG)。

要想计算:	按:	显示:
x 的正弦值。	SIN 3 0 ENTER	SIN(30) 0.5000
x 的余弦值。	COS 6 0 ENTER	COS(60) 0.5000
x 的正切值。	TAN 4 5 ENTER	TAN(45) 1.0000
x 的余切值。	2nd ASIN 1 ENTER	ASIN(1) 90.0000
x 的余弦值。	2nd ACOS 0 ENTER	ACOS(0) 90.0000
x 的正切值。	2nd ATAN 0 ENTER	ATAN(0) 0.0000

双曲函数

要想计算:	按:
x 的双曲线正弦值 (SINH)。	2nd HYP SIN , 键入数字, 按 ENTER
x 的双曲线余弦值 (COSH)。	2nd HYP COS , 键入数字, 按 ENTER
x 的双曲线正切值 (TANH)。	2nd HYP TAN , 键入数字, 按 ENTER
x 的双曲线余切值 (ASINH)。	2nd HYP 2nd ASIN , 键入数字, 按 ENTER
x 的双曲线弧余弦值 (ACOSH)。	2nd HYP 2nd ACOS , 键入数字, 按 ENTER
x 的双曲线正切值 (ATANH)。	2nd HYP 2nd ATAN , 键入数字, 按 ENTER

数字部分

要想计算:	按:	显示:
2.47 的整数部分	$\left[\leftarrow \right]$ INTG $\left[6 \right]$ (6IP) $\left[2 \right]$ $\left[\cdot \right]$ $\left[4 \right]$ $\left[7 \right]$ ENTER	IP(2.47) 2.0000
2.47 的分数部分	$\left[\leftarrow \right]$ INTG $\left[5 \right]$ (5FP) $\left[2 \right]$ $\left[\cdot \right]$ $\left[4 \right]$ $\left[7 \right]$ ENTER	FP(2.47) 0.4700
-7 的绝对值	$\left[\rightarrow \right]$ ABS $\left[+/\- \right]$ $\left[7 \right]$ ENTER	ABS(-7) 7.0000
9 的符号值	$\left[\leftarrow \right]$ INTG $\left[1 \right]$ (1SGN) $\left[9 \right]$ ENTER	SGN(9) 1.0000
等于或小于 -5.3 的最大整数	$\left[\leftarrow \right]$ INTG $\left[4 \right]$ (4INTG) $\left[+/\- \right]$ $\left[5 \right]$ $\left[\cdot \right]$ $\left[3 \right]$ ENTER	INTG(-5.3) -6.0000

查阅栈

$\left[\rightarrow \right]$ 或 $\left[\rightarrow \right]$ $\left[\rightarrow \right]$ 键槽在显示屏上产生一个菜单 - X, Y, Z, T 寄存器, 让您查看栈的全部内容。 $\left[\rightarrow \right]$ 和 $\left[\rightarrow \right]$ $\left[\rightarrow \right]$ 键槽的不同之处在于显示屏上下划线的位置。按 $\left[\rightarrow \right]$ $\left[\rightarrow \right]$ 显示 T 寄存器上的下划线, 按 $\left[\rightarrow \right]$ 则显示 Y 寄存器上的下划线。

按 $\left[\rightarrow \right]$ 显示以下菜单:

X Y Z T

数值

按 $\left[\rightarrow \right]$ $\left[\rightarrow \right]$ 显示以下菜单:

X Y Z T

数值

您可以按 $\left[\rightarrow \right]$ 和 $\left[\rightarrow \right]$ $\left[\rightarrow \right]$ (以及 $\left[\rightarrow \right]$ 或 $\left[\leftarrow \right]$) 查看栈的全部内容并将其恢复。它们将显示为 REGX, REGY, REGZ 或 REGT (取决于那部分栈被恢复) 并可以用于进行进一步计算。

ALG 模式下 X-、Y-、Z-、T- 寄存器的数值与在 RPN 模式下相同。在进行正常计算、求解、编程或积分之后，当您在 ALG 及 RPN 逻辑模式之间进行切换时，4 个寄存器的数值将和在 RPN 或 ALG 模式下相同。

对方程进行积分

1. 键入方程 (请参阅第 6 章 "将方程输入到方程列表中")，然后退出方程模式。
2. 输入积分极限；键入下限后按 $\int_{x \leftrightarrow y}$ ，然后键入上限。
3. 显示方程 按 EQN ；必要时，滚动查看方程列表 (按 \uparrow 或 \downarrow) 显示所需的方程。
4. 选择积分变量：按 \leftarrow \int 变量。这样将开始计算。

带复数的运算

要想输入复数：

算式： $x + iy$

1. 输入真数部分。
2. 按下 i 。
3. 输入虚数部分。

算式： $x + y \cdot i$

1. 输入真数部分。
2. 按下 $+$ 。
3. 输入虚数部分。
4. 按下 i 。

算式： $r \theta a$

1. 输入 r 的数值。
2. 按下 \angle θ 。
3. 输入 θ 的数值。

要想进行一个复数的运算：

1. 选择函数。
2. 输入复数 z 。
3. 按 **ENTER** 计算。
4. 计算结果将在第二行中显示，显示形式则为您在 **MODE** 中设置的显示形式。

要想进行两个复数的算术运算：

1. 输入第一个复数 z_1 。
2. 选择算术运算。
3. 输入第二个复数 z_2 。
4. 按 **ENTER** 计算。
5. 计算结果将在第二行中显示，显示形式则为您在 **MODE** 中设置的显示形式。

以下是一些复数的例子：

示例：

评估 $\sin(2+3i)$

键：	显示：	说明：
2nd DISPLAY 9 ($9 \times i \cdot v$)		设置显示形式
SIN 2 + 3 i	$\text{SIN}(2+3i)$	
ENTER	$\text{SIN}(2+3i)$ $9.1545i-4.1689$	结果为 $9.1545i-4.1689$

示例：

对表达式进行评估

$$z_1 \div (z_2 + z_3),$$

式中 $z_1 = 23 + 13i$, $z_2 = -2 + i$, $z_3 = 4 - 3i$

键:	显示:	说明:
\leftarrow [DISPLAY] [.] [1] $(10 \times + \nu i)$ $()$ [2] [3] [+] [1] [3] [i] [>] [÷] [()] [+/-] [2] [+] [i] [+] [4] [-] [3] [i] [ENTER]	$\leftarrow i \div (-2 + i + 4 - 3i)$ $(23 + 13i) \div (-2 + \dots$ $2.5000 + 9.0000i$	设置显示形式。 结果为 $2.5000 + 9.0000 i$

示例:

评估 $(4 - 2/5 i) \times (3 - 2/3 i)$

键:	显示:	说明:
$()$ [4] [-] [.] [2] [.] [5] [i] [>] [x] [()] [3] [-] [.] [2] [.] [3] [i] [ENTER]	$\leftarrow 5i) \times (3 - 0.2/3i)$ $(4 - 0.2/5i) \times (3 - \dots$ $11.7333i - 3.8667$	结果为 $11.7333 i - 3.8667$

基底 2、8 和 16 中的算术运算

以下为一些十六、八和二进制模式算术运算的例子:

示例:

$$12F_{16} + E9A_{16} = ?$$

键:	显示:	说明:
\rightarrow [BASE] [2] (2HEX)		设置基底 16。HEX 信号器 打开。

$\boxed{1} \boxed{2} \boxed{RCL} \boxed{F} \boxed{\rightarrow}$
 $\boxed{BASE} \boxed{6} \boxed{(6h)} \boxed{+} \boxed{RCL}$
 $\boxed{E} \boxed{9} \boxed{RCL} \boxed{A} \boxed{\rightarrow}$
 $\boxed{BASE} \boxed{6} \boxed{(6h)} \boxed{ENTER}$

$12Fh + E9Ah$ 结果。
 $FC9h$

$$7760_8 - 4326_8 = ?$$

$\boxed{\rightarrow} \boxed{BASE} \boxed{3} \boxed{(3OCT)}$

$12Fh + E9Ah$ 设置基底 8。OCT 信号器
 7711_0 打开。

$\boxed{7} \boxed{7} \boxed{6} \boxed{0} \boxed{\rightarrow} \boxed{BASE}$

$7760_0 - 4326_0$ 将所显示的数字转换为八
 3432_0 进制。

$\boxed{7} \boxed{(7_0)}$

$\boxed{-} \boxed{4} \boxed{3} \boxed{2} \boxed{6} \boxed{\rightarrow}$

$\boxed{BASE} \boxed{7} \boxed{(7_0)} \boxed{ENTER}$

$$100_8 \div 5_8 = ?$$

$\boxed{1} \boxed{0} \boxed{0} \boxed{\rightarrow} \boxed{BASE} \boxed{7}$

$\boxed{(7_0)} \boxed{\div} \boxed{5} \boxed{\rightarrow}$

$100_0 \div 5_0$ 结果的整数部分
 14_0

$\boxed{BASE} \boxed{7} \boxed{(7_0)} \boxed{ENTER}$

$$5A0_{16} + 10011000_2 = ?$$

$\boxed{\rightarrow} \boxed{BASE} \boxed{2} \boxed{(2HEX)}$

$5A0h +$ 设置基底 16；HEX 信号
 器打开。

$\boxed{5} \boxed{RCL} \boxed{A} \boxed{0} \boxed{\rightarrow}$

$\boxed{BASE} \boxed{6} \boxed{(6h)} \boxed{+}$

$\boxed{1} \boxed{0} \boxed{0} \boxed{1} \boxed{1} \boxed{0} \boxed{0} \leftarrow A0h + 10011000b$

$\boxed{0} \boxed{\rightarrow} \boxed{BASE} \boxed{8} \boxed{(8b)}$

$ENTER$ $5A0h + 10011000b$ 十六进制基底中的结果。

$\boxed{\rightarrow} \boxed{BASE} \boxed{1} \boxed{(1DEC)}$

$5A0h + 10011000b$ 恢复十进制基底。
 $638h$
 $1,592.0000$

输入统计双变量数据

在 ALG 模式下，记得以相反顺序 ($y \boxed{x \leftrightarrow y}$) x 或 $y \boxed{ENTER} x$) 输入 (x, y) ，使 y 在 Y 寄存器中结束而 x 在 X 寄存器中结束。

- 按 $\boxed{\rightarrow} \boxed{CLEAR} \boxed{4} \boxed{(4\Sigma)}$ 清除既有统计数据。
- 先键入 y 值，然后按 $\boxed{x \leftrightarrow y}$ 。

- 键入相应的 x 值，按 $\boxed{\Sigma+}$ 。
- 显示屏显示 n ，即您已经累积的统计数据对的数量。
- 继续输入 x ， y 对，每输入一项， n 就更新一次。

如果您想要删除刚刚输入的错误数值，按 $\boxed{\leftarrow}$ $\boxed{\Sigma-}$ 。在删除错误的统计数据后，计算器将在第一行中显示最后输入的统计数据 (显示屏上最上一行) 并在第二行中显示 n 的数值。如果没有统计数据，计算器将在第 2 行中显示 $n=0$ 。

示例:

在左边键入 x ， y 数值后，如右所示进行纠正:

初始 x, y	纠正后的 x, y
20, 4	20, 5
400, 6	40, 6

键:	显示:	说明:
$\boxed{\rightarrow}$ $\boxed{\text{CLEAR}}$ $\boxed{4}$ (4 Σ)		清除既有统计数据。
$\boxed{4}$ $\boxed{x\leftrightarrow y}$ $\boxed{2}$ $\boxed{0}$ $\boxed{\Sigma+}$	20 $\Sigma+$ 1.0000	输入第一个新数据对。
$\boxed{6}$ $\boxed{x\leftrightarrow y}$ $\boxed{4}$ $\boxed{0}$ $\boxed{0}$ $\boxed{\Sigma+}$	400 $\Sigma+$ 2.0000	显示屏显示 n ，您输入的数据对的数量。
$\boxed{\rightarrow}$ $\boxed{\text{LAST}x}$	LAST x 400.0000	取会最后一个 x 值。最后的 y 仍在 Y 寄存器中。
$\boxed{\leftarrow}$ $\boxed{\Sigma-}$	400 $\Sigma-$ 1.0000	删除最后的数据对。
$\boxed{6}$ $\boxed{x\leftrightarrow y}$ $\boxed{4}$ $\boxed{0}$ $\boxed{\Sigma+}$	40 $\Sigma+$ 2.0000	重新输入最后的数据对。
$\boxed{4}$ $\boxed{x\leftrightarrow y}$ $\boxed{2}$ $\boxed{0}$ $\boxed{\leftarrow}$ $\boxed{\Sigma-}$	20 $\Sigma-$ 1.0000	删除最前的数据对。
$\boxed{5}$ $\boxed{x\leftrightarrow y}$ $\boxed{2}$ $\boxed{0}$ $\boxed{\Sigma+}$	20 $\Sigma+$ 2.0000	重新输入最初的数据对。统计寄存器中仍有总共两个数据对。

线性回归

线性回归或 L.R.(也称为*线性估计*)是一种求出与一组 x, y 数据最相适应的直线的统计方法。

- 要想求出 x (或 y) 的估计值, 键入 y (或 x) 的给定假设值, 按 **ENTER**, 然后按 **◀** **L.R.** **(\hat{x})**(或 **◀** **L.R.** **▶** **(\hat{y})**)。
- 要想求出定义与您的数据最相符合直线的数值, 先按 **◀** **L.R.**, 然后按 **(r)**, **(m)** 或 **(b)**。

有关求解的更多信息

本附录提供有关第 7 章中 SOLVE 运算之外的 SOLVE 运算的信息。

SOLVE 是如何求根的

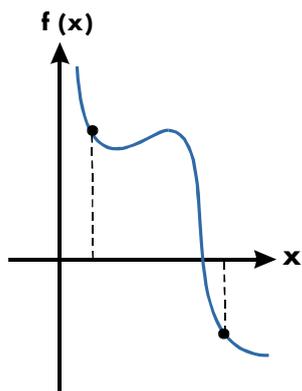
SOLVE 首先企图直接解方程以获得未知变量。如果这一企图失败，SOLVE 将其改为一个重复程序。重复运算是要分别执行指定的方程。方程返回的数值为未知变量 x 的函数 $f(x)$ 。 $f(x)$ 为以未知变量 x 定义的函数的数学速记)。SOLVE 从未知变量 x 的估计值开始，函数 $f(x)$ 每连续执行一次时对其进行完善。

如果函数 $f(x)$ 的任意两个连续估计值的符号相反，则 SOLVE 假定 function $f(x)$ 在两个估计值之间至少一个位置处与 x 轴相交。这一区间在求出根之前系统性变窄。

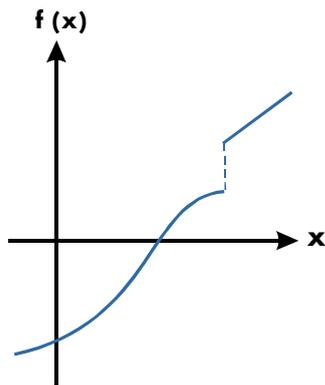
SOLVE 若要求根，则根必须在计算器的数字范围内；而且，如果出现反复搜索，函数必须以数学的方式加以定义。如果满足以下条件中的任何一种或多种，只要有根存在（在溢出范围内），SOLVE 就始终可以求根：

- 两个估计值得出的 $f(x)$ 数值符号相反，且函数图在两个估计值之间至少一个位置处与 x 轴相交（下图 a）。
- $f(x)$ 始终随 x 增大而增大或减小（下图 b）。
- $f(x)$ 图处处呈现凹状或凸状（下图 c）。

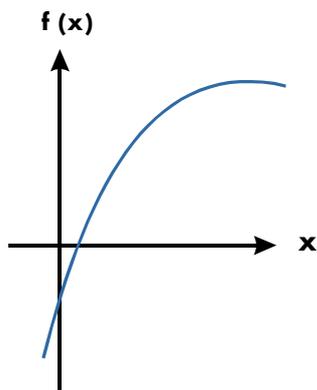
- $f(x)$ 有一个或多个局部最小值时，各自单独出现在 $f(x)$ 的相邻根之间 (下图 b)。



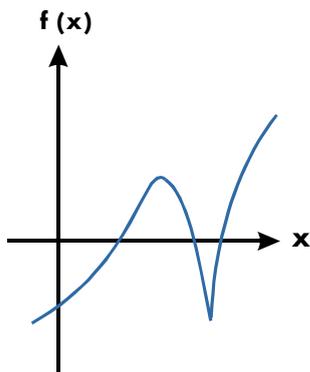
a



b



c



d

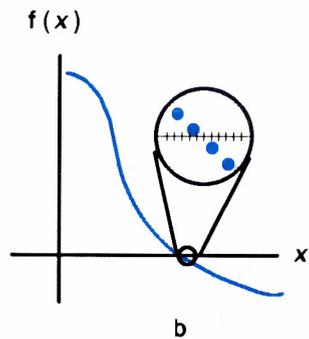
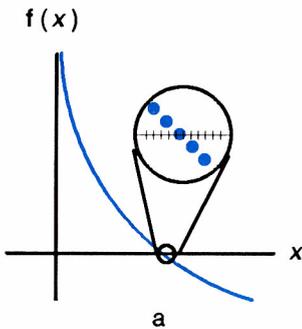
可以求根的函数

在大多数情况下，计算所得的根为方程理论、不确定精确根的准确估计值。“理想”的解为 $f(x) = 0$ 。但是，非常小的非 0 $f(x)$ 值通常可以接受，原因是它可能是由有限 (12 位) 精度的近似数引起的。

解分结果

在以下两种情况中的任何一种情况下， SOLVE 运算都将得出一个解：

- SOLVE 运算求得一个 $f(x)$ 等于 0 的估计值。(见下图。)
- SOLVE 运算求得的估计值 $f(x)$ 不等于 0，但是计算所得的根为一个靠近函数图与 x 轴相交位的 12 位数字(见下图 b)。这种情况在两个最终估计值相临(即，在 12 为数中它们相隔 1)且函数的数值对于一个估计值为正而对于另外一个估计值则为负时发生。或者，这两个估计值为 $(0, 10^{-499})$ 或 $(0, -10^{-499})$ 。在大多数情况下， $f(x)$ 将相对靠近 0。



- ✓ 要想获得更多有关结果的信息，按 $\boxed{R\downarrow}$ 查看根 (x) 的先前估计值，该数值留在 Y 寄存器中。再次按 $\boxed{R\downarrow}$ 查看 $f(x)$ 值，该数值留在 Z 寄存器中。如果 $f(x)$ 等于 0 或相对较小，极有可能是解已经求出。但是，如果 $f(x)$ 相对较大，在对结果进行解析时，您需要谨慎行事。

示例：单根方程。

求方程的根：

$$-2x^3 + 4x^2 - 6x + 8 = 0$$

将方程作为表达式输入：

键:	显示:	说明:
EQN		选择方程模式。
+/- 2 x		输入方程。
RCL X y^x 3		
+ 4 x		
RCL X y^x 2		
- 6 x RCL X		
+ 8 ENTER	$-2x^3+4x^2-6$ →	
↵ SHOW	CK=B9AD LN=18	校和及长度。
C		取消方程模式。

先则，解方程以求根：

键:	显示:	说明:
0 ↵ STO X	10_	根的初始猜测值。
ENTER 1 0		
EQN	$-2x^3+4x^2-6$ →	选择方程模式；显示方程的左侧。
↵ SOLVE X	SOLVING X= 1.6506	求解 X；显示结果。
✓ ↵	1.6506	两个最终估计值相同，精确至 4 个小数位。
✓ ↵	-4.0000E-11	$f(x)$ 非常小，因此近似值是一个很好的根。

示例： 双根方程。

求抛物线方程的两个根：

$$x^2 + x - 6 = 0.$$

将方程作为表达式输入：

键:	显示:	说明:
EQN		选择方程模式。
RCL X y^x 2 +		输入方程。
RCL X - 6	X^2+X-6	
ENTER		
↵ SHOW	CK=3971 LN=7	校和及长度。
C		取消方程模式。

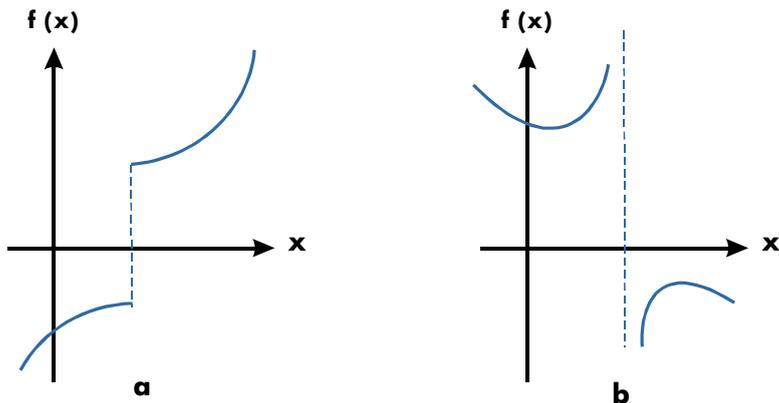
现在，解方程以求出其正 / 负根：

键:	显示:	说明:
0 ↵ STO X	10_	您的正根初始猜测值。
ENTER 1 0		
EQN	X^2+X-6	选择方程模式；显示方程。
↵ SOLVE X	SOLVING X= 2.0000	用初始猜测值 0 和 10 计算正根。
↵ R↓	2.0000	两个最终估计值相同。
↵ ↵ SHOW	0.000000000000	$f(x) = 0$.
0 ↵ STO X	-10_	您的负根初始猜测值。
ENTER 1 0 +/-		
EQN	X^2+X-6	重新显示方程。
↵ SOLVE X	SOLVING X= -3.0000	用初始猜测值 0 和 -10 计算负根。
↵ ↵ ↵ SHOW	0.000000000000	$f(x) = 0$.

某些情况需要特殊对待：

- 如果函数图出现与 x 相交的断点，则 SOLVE 运算返回一个接近该断点的数值（见下图 a）。在这种情况下， $f(x)$ 可能比较大。

- $f(x)$ 的数值在函数图改变符号的位置可能会趋于无穷 (见下图 b)。这种情况称为“极”。由于 SOLVE 运算决定两个相邻 x 数值的符号修改, SOLVE 运算返回可能的根。但是, $f(x)$ 的数值会相对比较大。如果“极”发生在一个以 12 位精确表示的 x 值处, 该数值将导致计算停止, 同时出现错误信息。



特殊情况：间断与极

示例：间断函数。

求方程的根：

$$IP(x) = 1.5$$

输入方程：

键：	显示：	说明：
EQN		选择方程模式。
INTG 6 (6IP)		输入方程。
RCL X > ↵ =		
1 . 5 ENTER	IP(X)=1.5	
↵ SHOW	CK=D2C1 LN=9	校和及长度。
C		取消方程模式。

现在，解方程以求根：

键：	显示：	说明：
0 → STO X	5_	您的初始根猜测值。
ENTER 5		
EQN	IP(X)=1.5	选择方程模式；显示方程。
→ SOLVE X	SOLVING	以猜测值 0 和 5 求根。
	X=	
	2.0000	
← SHOW	1.99999999999	显示根，精确至 11 个小数位。
✓ ↵ ← SHOW	2.00000000000	先前的估计值略大一些。
✓ ↵	-0.5000	f(x) 相对较大。

注意最后两个估计值以及 $f(x)$ 的相对较大数值间的差异。问题在于，不存在 $f(x)$ 等于 0 的 x 数值。但是， $x = 1.99999999999$ 时，有一个临近 x 数值得出相反的 $f(x)$ 符号。

示例：

求方程的根

$$\frac{x}{x^2 - 6} - 1 = 0$$

x 接近 $\sqrt{6}$ 时， $f(x)$ 变为一个非常大的正数或负数。

将方程作为表达式输入。

键：	显示：	说明：
EQN		选择方程模式。
RCL X ÷ ()		输入方程。
RCL X y^x 2		
← 6 >		
← 1 ENTER		
	X÷(X^2-6)-1	

◀ **SHOW**

CK=7358

校和及长度。

LN=11

C

取消方程模式。

现在，解方程以求根。

键：

显示：

说明：

2 **.** **3** **↵**

您的初始根猜测值。

STO **X** **ENTER**

2.7_

2 **.** **7**

EQN

$X \div (X^2 - 6) - 1$

选择方程模式；显示方程。

↵ **SOLVE** **X**

NO ROOT FND

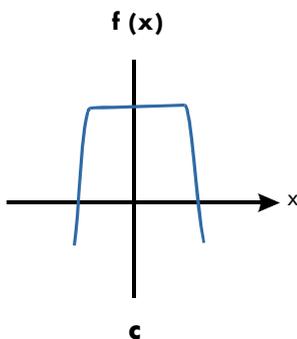
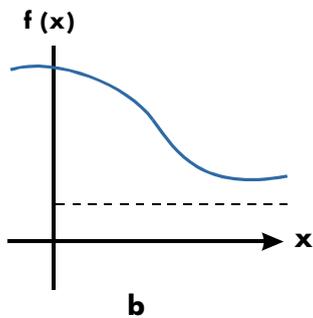
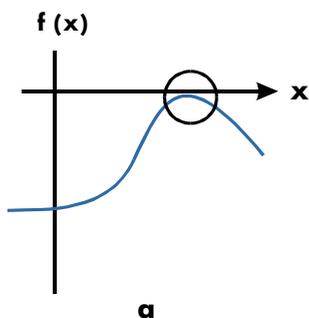
未求出 $f(x)$ 的根。

当 **SOLVE** 无法求出根时

有些时候，**SOLVE** 会求不出根来。以下情况导致信息 **NO ROOT FND**。

- 搜索在局部最小值或最大值附近终止 (见下图 a)。
- 搜索停止是因为 **SOLVE** 在水平渐近线上运行。水平渐近线是一个 $f(x)$ 对于广泛的 x 数值基本恒定的区域 (见下图 b)。
- 搜索集中在函数的局部 "平" 区 (见下图 c)。

在这类情况下，栈中的数字将和执行 **SOLVE** 运算前的数值一样。



未求出根的情况

示例： 相对最小值。

计算本抛物线方程的根：

$$x^2 - 6x + 13 = 0.$$

$x = 3$ 其数值最小。

将方程作为表达式输入：

键：

显示：

说明：

EQN

RCL **X** **y^x** **2**

- **6** **x** **RCL** **X**

+ **13** **ENTER**

$X^2 - 6 \times X + 13$

选择方程模式。
输入方程。

[↶] [SHOW]

CK=EC74
LN=10

校和及长度。

[C]

取消方程模式。

现在，解方程以求根：

键：

显示：

说明：

[0] [↶] [STO] [X]

您的初始根猜测值。

[ENTER] [1] [0]

10_

[EQN]

X^2-6xX+13

选择方程模式；显示方程。

[↶] [SOLVE] [X]

NO ROOT FND

初始值为 0 和 10，搜索失败

示例：渐近线。

求方程的根

$$10 - \frac{1}{X} = 0$$

将方程作为表达式输入。

键：

显示：

说明：

[EQN]

选择方程模式。

[1] [0] [-] [1/x]

输入方程。

[RCL] [X] [ENTER]

10-INW(X)

[↶] [SHOW]

CK=6EAB
LN=9

校和及长度。

[C]

取消方程模式。

[.] [0] [0] [5] [↶]

您的正根猜测值。

[STO] [X] [ENTER] [5]

5_

[EQN]

10-INW(X)

选择方程模式；显示方程。

[↶] [SOLVE] [X]

X=

用猜测值 0.005 和 5 求解 x 值。

0.1000

0.1000

先前的估计值相同。

0.000000000000

f(x) = 0

注意观察您用负数作为猜测值时发生的情况：

键：	显示：	说明：
\pm/\square 1 \rightarrow STO \square X ENTER	-1.0000	您的负根猜测值。
\pm/\square 2 EQN	10-INV(X)	选择方程模式；显示方程。
\rightarrow SOLVE \square X	X= 0.1000	求解 X；显示结果。

示例： 求方程的根。

$$\sqrt{[x \div (x + 0.3)]} - 0.5 = 0$$

将方程作为表达式输入：

键：	显示：	说明：
EQN		选择方程模式。
$\sqrt{\square}$ RCL \square X \div () RCL \square X + \cdot 3 > > - \cdot 5 ENTER	SQRT(X÷(X+0.3)) \rightarrow	输入方程。
\leftarrow SHOW	CK=9F3B LN=19	校和及长度。
C		取消方程模式。

先试着求正根：

键：	显示：	说明：
0 \rightarrow STO \square X ENTER 1 0 EQN	10_ SQRT(X÷(X+0.3)) \rightarrow	您的正根猜测值。
\rightarrow SOLVE \square X	X= 0.1000	选择方程模式；显示方程的左侧。 用猜测值 0 和 10 计算根。

然后试着输入猜测值 0 和 -10 求负根。注意函数未针对 0 和 -0.3 间的 x 数值进行定义，因为那些数值得出的分母为正，但是分子为负，造成负平方根。

键:	显示:	说明:
<code>0</code> <code>↵</code> <code>STO</code> <code>X</code>		
<code>ENTER</code> <code>+/-</code> <code>1</code> <code>0</code>	<code>-10</code>	
<code>EQN</code>	<code>SQRT(X÷(X+0.3))</code> <code>▶</code>	选择方程模式；显示方程的左侧。
<code>↵</code> <code>SOLVE</code> <code>X</code>	<code>NO ROOT FND</code>	未求出 $f(x)$ 的根。

示例：局部“平”区。

求方程的根

$$f(x) = x + 2 \text{ if } x < -1,$$

对于 $-1 \leq x \leq 1$ (局部平区), $f(x) = 1$,

$$f(x) = -x + 2 \text{ if } x > 1.$$

在 RPN 模式下，将函数作为程序输入：

```
J001 LBL J
J002 1
J003 2
J004 RCL+ X
J005 x<y?
J006 RTN
J007 4
J008 -
J009 +/-
J010 x>y?
J011 R↓
J012 RTN
```

校和及长度：9412 39

用初始猜测值 10^{-8} 和 -10^{-8} 求 X 值。

键:	显示:	说明:
(在 RPN 模式下)		
1 E 8		输入猜测值。
+/- ▸ STO X 1	$-1E-8$	
+/- E 8 +/-		
↵ FN= J	$-1.00000E-8$	选择程序 "J" 作为函数。
▸ SOLVE X	$X=$	求解 X; 显示结果。
	-2.00000	

四舍五入错误

计算器的有限 (12 位) 精确度会因为四舍五入而导致错误, 从而对 SOLVE 和积分的重复解产生负面影响。例如,

$$[(|x| + 1) + 10^{15}]^2 - 10^{30} = 0$$

无根是因为 $f(x)$ 始终大于 0。但是, 给定猜测值 1 和 2, SOLVE 由于四舍五入错误而返回答案 1.0000。

四舍五入错误还可导致 SOLVE 不能将根求出来。方程

$$|x^2 - 7| = 0$$

在 $\sqrt{7}$ 处有一个根。但是, 不存在和 $\sqrt{7}$ 完全相等的 12 位数字, 因此, 计算器绝不会使方程等于 0。另外, 函数从不发生改变, 符号 SOLVE 返回信息 **NO ROOT FND**。

关于积分的更多信息

本附录提供更有有关第 8 章中积分之外的积分的信息。

积分是如何评估的

积分运算使用的算法 $\int_{a}^{b} f(x) dx$ 通过计算积分区间内多个 x (称为采样点) 值下函数数值的加权平均来支来计算函数 $f(x)$ 的积分。对于任何这种采样过程, 其结果准确度取决于所考虑采样点的数量: 一般而言, 采样点越多, 准确度越高。如果 $f(x)$ 可以在无穷个采样点去进行评估, 算法通过忽略被评估函数 $f(x)$ 的不确定性引起的限制, 始终提供确切的答案。

在无穷个采样点对函数进行下去可以永远进行下去。但是, 这毫无必要, 原因是计算所得积分的最大准确度受计算所得函数数值的准确度的限制。只需使用确定数量的采样点, 算法就可以计算考虑 $f(x)$ 固有不确定性而获得的准确度一样准确的积分。

本积分算法首先只考虑几个样本点, 得出相对不怎么精确的近似值。如果这些近似值达不到 $f(x)$ 的准确度允许的准确度, 算法以更大数量的采样点进行重复。这些重复每次使用大约两倍的采样点继续进行, 直到所得出的近似值达到考虑 $f(x)$ 固有不确定性而获得的准确度。

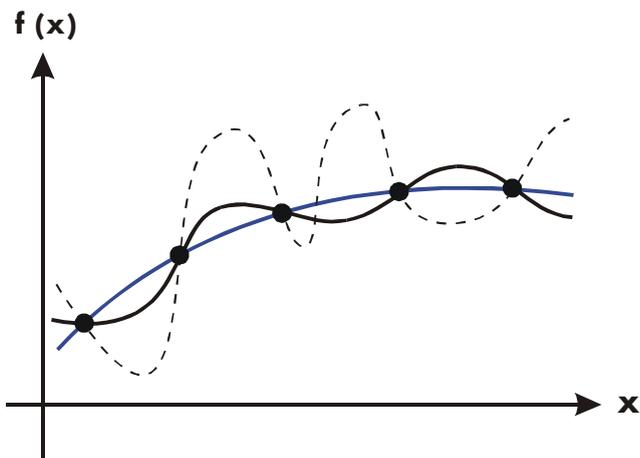
如第 8 章所述，最终近似值的不确定性是为一个从显示各种中获得的数字，指定了函数的不确定性。在每一次的重复结束时，算法将重复期间计算的近似值与先前两次重复期间计算的近似值进行比较。如果这 3 个近似值以及其它两个近似值之间的差异小于最终近似值可容忍的不确定性，计算将结束，将当前近似值留在 X 寄存器中并将其不确定性留在 Y 寄存器中。

这三个连续近似值的误差 - 即实际积分与近似值之间的差异全都大于近似值本身之间的差异是极有可能的。因此，最终结果的误差将小于其不确定性 (只要 $f(x)$ 变得不要太快)。虽然我们无法得知最终结果的误差，该误差是极有可能超过所显示的近似值不确定性。换句话说，Y 寄存器中的不确定性估计值即为近似值与实际积分间差异几乎肯定的 " 上限 "。

可导致错误结果的条件

虽然 HP 35s 中的积分算法在某些情况下是最适用的积分算法，和其它所有的数字积分算法一样，它也可能会给您错误的答案。*发生这种情况的可能性很低*。这种算法设计用于以几乎任何平滑的函数得出准确的结果。只有是表现非常奇怪的函数才存在得出错误答案的实质风险。这种函数很在与实际物理状态相关的问题中出现；如有出现，则可以直接进行辨识和处理。

不幸的是，由于算法对于 $f(x)$ 的全部了解仅限于其相同采样点的数值，算法无法区分 $f(x)$ 与其它任何在所有采样点与 $f(x)$ 吻合的函数。这种情况在下文进行说明 (积分区间的一部分上)，向您展示其图形包括许多共同采样点的函数。



利用这一采样点数字，算法将计算任何所显示函数积分的同一近似值。显示有兰色及黑色实线的函数的实际积分大致都相同，因此，如果 $f(x)$ 是这些函数中的一个的话，近似值将会相当准确。然而，由于显示有虚线的函数的实际积分与其它函数的实际积分大不相同，因此，如果 $f(x)$ 就是这一函数，当前近似值将相当不准确。

算法通过在一个及多个采样点对函数进行采样逐步了解到函数的一般表现。如果函数在某个区域的浮动值和积分区间其余部分的表现不同，在某个重复时算法将有可能将该浮动值删除。出现这种情况时，采样点的数值将增大直到连续重复得出考虑最为迅速，*但也最有特色的浮动值的存在的近似值。*

例如，考虑近似值

$$\int_0^{\infty} x e^{-x} dx.$$

由于您正在用数字方法对本积分进行评估，您可能会认为您应该将积分上限表示为 10^{499} ，该数字实际上就是计算器中您能够输入的最大数字。

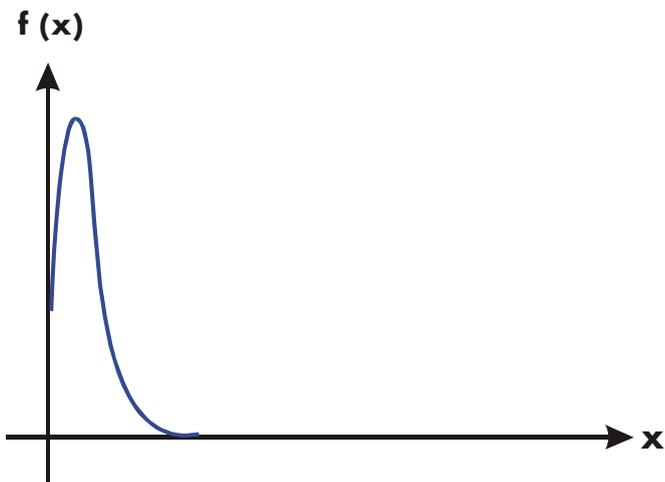
试试看会发生何种情况。输入函数 $f(x) = xe^{-x}$ 。

键:	显示:	说明:
EQN		选择方程模式。
RCL X X e^x	$X \times EXP ()$	输入方程。
+/- RCL X ENTER	$X \times EXP (- X)$	方程结束。
↵ SHOW	CK=2FE6 LN=9	校和及长度。
C		取消方程模式。

将显示格式设置为 SCI 3, 指定积分的上下 / 限为 0 和 10^{499} , 然后开始积分。

键:	显示:	说明:
✓ ↵ DISPLAY 2 (2SCI)		指定精度水准及积分极限。
3 ENTER 1 E 4	1E499_	
9 9		
EQN	$X \times EXP (- X)$	选择方程模式; 显示方程。
↵ / X	INTEGRATING	积分近似值。
	∫ = 0.000E0	

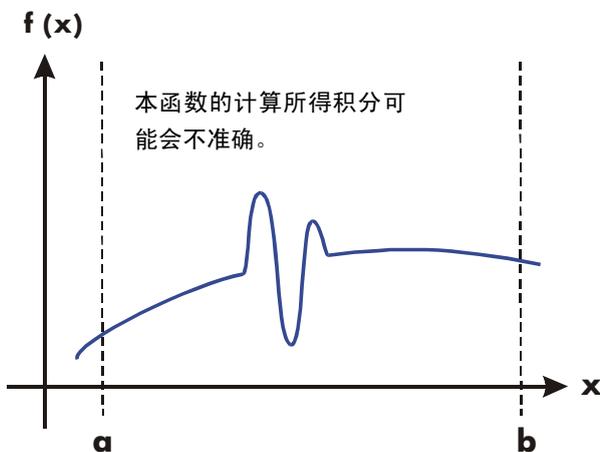
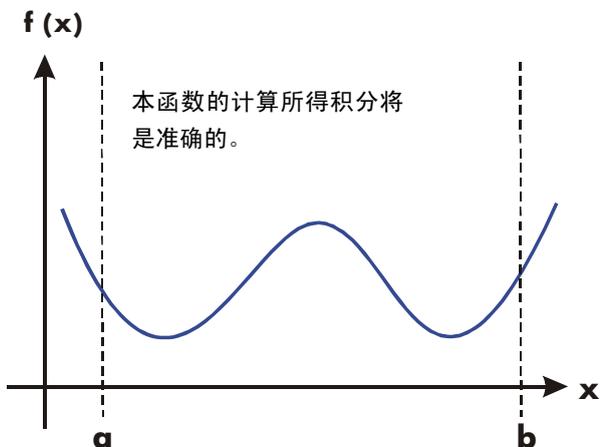
计算器返回的答案明显是错误的, 原因是 0 和 ∞ 之间 $f(x) = xe^{-x}$ 的实际积分实际为 1。但是问题并不在于 ∞ 表示为 10^{499} , 因为 0 和 10^{499} 之间 $f(x) = xe^{-x}$ 的实际积分非常接近 1。从积分区间上 $f(x)$ 的图形就可以明显看出答案不正确的原因。



图形为一个非常接近原点的峰值。由于刚好无任何采样点发现该峰值，算法假定 $f(x)$ 在整个积分区间内保持等于 0。即使您通过以 SCI 11 或 ALL 格式对积分进行计算从而增大采样点的数量，在这一特定区间上对这一特定函数进行积分时无任何附加采样点会发现该峰值。（对于解决这类问题的更好方法，参见下一话题“延长计算时间的条件”）。

幸运地是，由于显示此种异常现象的函数（通常在其它任何地方不反映函数特点的浮动）非同寻常，您不大可能在不知情的情况下对其进行积分。能够导致错误结果的函数可以通过该函数及其低阶衍生数在整个积分区间的变化以一种很简单的方式加以辨别。从根本上来说，函数或其衍生数的变化越快，计算停止的也就越慢，而所得出的近似数也就越不可靠。

注意函数 (或其低阶衍生数) 变化迅速程度是由积分区间的宽度决定的。对于给定数量的采样点, 与仅仅局限在区间的一小部分上相比, 有 3 个浮动值的函数 $f(x)$ 在这些变量在大部分积分区间上分布时可以由其样本以一种更好的方式加以特征化。(这两种情况如以下两副图所示)。考虑到作为一种函数摆动类型的变化或浮动, 精彩的标准为摆动时间与积分区间之比; 该比数越大, 计算完成的也就越快, 所得出的近似值也就越可靠。



在很多情况下，您将足够熟悉您要对其进行积分的函数，从而了解该函数有无任何相对于积分区间的快速摆动。如果您不熟悉该函数，且怀疑它可能导致问题，您可以用您为了该目的而写入的方程或程序对方程进行评估来迅速划出一些点。

如果因为某种原因使您在获得一个积分近似值后怀疑其有效性，您可以用一个简单的程序对其进行验证。将积分区间进一步划分为两个或多个相临子区间。对每一个子区间上的函数进行积分，然后加上所得出的近似值。这样就使函数在一组全新的采样点处被采样，从而更有可能揭示任何先前被隐藏的峰值。如果初始近似值有效，它将等于各子区间上近似值之和。

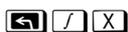
延长计算时间的条件

在前例中，算法得出的结果不正确，原因是它从未检测到函数的峰值。之出现这种情况是因为函数的变化速度相对于积分区间的宽度太快。如果积分区间的宽度较小，您将会得到正确的结果；但是，如果区间太宽，所需的计算时间也就会很长。

考虑一个积分区间足够宽以至与需要大量的计算时间，但也不是宽的无法正确计算的积分。注意由于 $f(x) = xe^{-x}$ 和 x 接近 ∞ 一样非常迅速地接近 0，该函数在较大 x 数值处对于积分的贡献是可以忽略不计的。因此，您可以通过用一个不似 10^{499} 那么大的数字，例如 10^3 替换 ∞ ，即积分上限对积分进行评估。

在本新积分极限内重新进行先前的积分问题：

键：	显示：	说明：
0 ENTER 1 E	1E3_	新上限。
3		
EQN	$X \times EXP(-X)$	选择方程模式；显示方程。



INTEGRATING

积分。(计算需要 1 至 2 两分钟。)

$\int =$

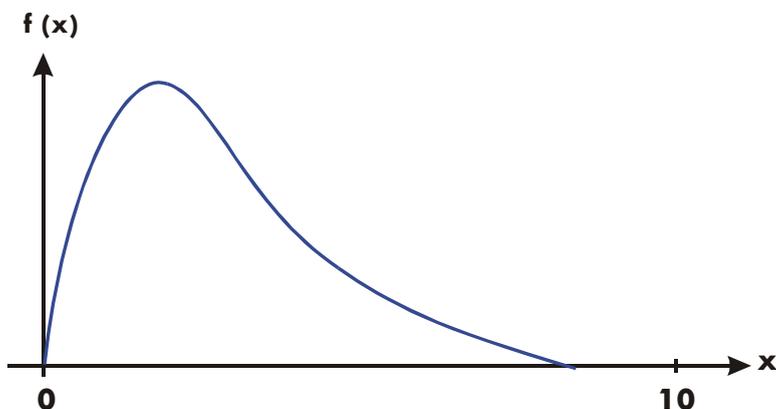
1.000E0



1.000E-3

近似值的不确定性。

这就是正确的答案，但其所花费的时间很长。要想弄清楚为什么会这样，比较 $x = 0$ 和 $x = 10^3$ 之间的函数图(它看起来和前例中显示的函数图几乎是一样的)与 $x = 0$ 和 $x = 10$ 之间的函数图：



您将会看到本函数仅在较小的 x 值处精彩。 x 值较大时，函数不精彩，原因是它以一种可以预测的方式逐步、平滑地降低。

算法以较高的样本密度对函数进行采样，直到连续近似值之前的差异变的足够小为止。对于函数不精彩区域中的窄区间，获得这一临界密度所需的时间要少。

要想获得相同的采样点密度，较大区间上所需采样点的数量要比较小区间上所需采样点的数量小得多。因此，在较大区间上需要多次进行重复以获得相同准确度的近似值，从而计算需要更多时间的积分。

由于计算时间取决于函数比较精彩区域中取得特定采样点密度的速度，当积分区间包括大多数函数不精彩的区域时，任何函数的积分计算时间都将被延长。幸运的是，如果您必须计算这样一和积分，您可以对问题进行修改，从而大幅缩短计算时间。两个这类方法对积分区间及变量转换进行进一步划分。这些方式使您能够修改函数或积分极限，从而使积分函数在积分区间的表现更佳。

信息

计算器通过显示信息对某些条件或键盘键做出响应。  符号提醒您注意该信息。对于有效条件，信息在您将其清除之前一直保留。按  或  清除该信息，将显示先前的显示内容。按任何其它键也可清除该信息，但是该键的功能将不予执行。

 FN ACTIVE	一个正在运行的程序试图在积分计算正在进行时选择程序标签 (FN= 标签)。
 (J FN)	一个正在运行的程序试图在另外一个积分计算正在进行时对程序进行积分 (J FN α 变量)。
 (S SOLVE)	一个正在运行的程序试图在积分计算正在进行时解方程。
ALL VARS=0	变量目录 (   (1 VAR)) 显示未保存数值。
BAD GUESS	您在针对变量求解方程时设置一个错误的猜测数字 (和复数或矢量一样)。
CALCULATING	计算器正在执行可能会需要一段时间的函数。
CLR ALL? Y N	允许您验证清除存储器中的所有内容。
CLR EQN? Y N	允许您验证清除您正在编辑的方程。(仅在方程输入模式下发生。)
CLR PGMS? Y N	允许您验证清除存储器中的 <i>所有程序</i> 。(仅在程序输入模式下发生。)
DIVIDE BY 0	试图除以 0(如果 Y 寄存器包含 0, 则包括  (%CHG))。

DUPLICAT, LBL	试图输入已经存在的另外一个程序调度程序的程序标签。
EQN LIST TOP	显示方程内存的 " 顶部 "。内存机制为环形，因此， EQN LIST TOP 也是方程内存中最后一行之后的行。
INTEGRATING	计算器正在计算方程或程序的积分。这可能会需要一段时间。
INTERRUPTED	正在运行的 CALCULATE、SOLVE 或 f FN 运算通过在 ALG、RPN、EQN 或 PGM 模式下按 C 或 R/S 而终止。
INVALID DATA	数据错误。 <ul style="list-style-type: none"> ■ 试图保存或计算错误数据。 ■ 试图在 $r > n$，非整数 r 或 n 或 $n \geq 10^{16}$ 的条件下计算排列或组合。 ■ 试图在统计数据中保存复数或矢量。 ■ 试图保存包含位数大于允许最大基底 n 位数的基底 n 数字。 ■ 试图用 $x \leftrightarrow y$ 运算在统计寄存器中保存无效数据。 ■ 试图比较复数或矢量。 ■ 试图使用带非法自变量的三角或双曲线函数： <ul style="list-style-type: none"> ■ 带 x 的 TAN，90° 的奇数倍。 ■ $x < -1$ 或 $x > 1$ 时，ACOS 或 ASIN。 ■ 带 $x \leq -1$ 的 HYP ATAN；或 $x \geq 1$。 ■ 带 $x < 1$ 的 HYP ACOS。
INVALID VAR	试图在解方程时输入无效变量名称。
INVALID x!	试图在 x 为负整数的情况下进行因数或伽玛运算。

INVALID $\psi \times$	指数错误： <ul style="list-style-type: none"> ■ 试图将 0 升 0 次幂或负数次幂。 ■ 试图将负数升非整数次幂。 ■ 试图用一个负真部分将复数 $(0 + i0)$ 升一定次幂。
INVALID (I)	试图用无效间接值 (I 未定义) 进行运算。
INVALID (J)	试图用无效间接值 (J 未定义) 进行运算。
LOG(0)	试图取 0 或 $(0 + i0)$ 的对数。
LOG(NEG)	试图取负数的对数。
MEMORY CLEAR	所有的拥护内存都已被擦除 (见 B-3 页)。
MEMORY FULL	计算器无足够内存可用以进行计算 (见附录 B)。
NO	测试条件检查的条件不成立。(只有从键盘执行时才会发生。)
NONEXISTENT	试图参考一个不存在的带 GTO , XEQ 或 FN 的程序标签 (或行标签)。注意错误 NONEXISTENT 可能意味着 <ul style="list-style-type: none"> ■ 您明确 (从键盘上) 调用一个不存在的程序标签或 ■ 您调用的程序指向另外一个不存在的标签。 <p>积分结果不存在。</p>
NO LABELS	程序目录 ( MEM 2 (2PGM)) 显示未保存任何程序标签。
NO SOLUTION	针对本线性方程组未求出任何根。
MULT SOLUTION	针对本线性方程组求出了多个根。
NO ROOT FND	SOLVE (包括 EQN 和 PGM 模式) 无法用当前初始猜测值求出方程的根 (见第 D-8 页)。这些条件包括: 猜测值不佳、未求出根, 兴趣点以及左右不等。程序中执行的 SOLVE 运算不会产生这一错误, 而相同的条件则导致其跳过下一程序行 (指令 SOLVE 变量后的行)。

OVERFLOW	警告 (立刻显示)；结果的数值太大，计算器无法处理。计算器以当前显示格式返回 $\pm 9.999999999999999E499$ 。(见第 1-17 页 "数字及溢出的范围")。本条件设置标志 6。如果标志 5 处于设置状态，溢出的附加作用就是使正在运行的程序停止并将信息留在显示屏上，直到您按某个键。
PRGM TOP	显示程序内存的 "顶部"。内存机制为环形，因此， PRGM TOP 也是程序内存中最后一行之后的行。
RUNNING B	计算器正在运行方程或程序 (非 SOLVE 或 \int FN 子程序)。
SELECT FN	试图在未选择程序标签的情况下执行 SOLVE 变量或 \int FN d 变量。这种情况回在您头一次在信息 MEMORY CLEAR 后使用 SOLVE 或 \int FN 或当前标签不再存在时发生。
SOLVE ACTIVE	一个正在运行的程序试图在 SOLVE 运算进行时选择程序标签 (FN= 标签)。
SOLVE(SOLVE)	一个正在运行的程序试图在 SOLVE 运算进行时对程序进行积分。
SOLVE(\int FN)	一个正在运行的程序试图在 SOLVE 运算进行时对方程进行积分。
SOLVING	计算器正解方程或程序以求出其根。这可能会需要一段时间。
SQRT(NEG)	试图计算负数的平方根。
STAT ERROR	统计错误： <ul style="list-style-type: none"> ■ 试图在 $n = 0$ 的条件下进行统计计算。 ■ 试图在 $n = 1$ 的条件下计算 s_x, s_y, \hat{x}, \hat{y}, m, r 或 b。 ■ 试图只以 x 数据 (所有 y 值等于 0) 计算 r, \hat{x} 或 $\bar{x}w$。 ■ 试图在所有 x-值相等的条件下计算 \hat{x}, \hat{y}, r, m 或 b。
SYNTAX ERROR	对表达式，方程、 SOLVE 或 \int 进行评估期间检测到语法错误。按 ← 或 C 清除错误信息并允许您对错误进行纠正。

TOO BIG	数字的数值太大，无法转换为 HEX、OCT 或 BIN 基底。数字必须在该范围内 $-34,359,738,368 \leq n \leq 34,359,738,367$ 。
XEQ OVERFLOW	正在运行的程序尝试 21 st 迭套 XEQ 标签。(多达 20 个子程序可以进行迭套)。由于 SOLVE 和 \int FN 各使用一级，它们也会导致这一错误。
YES	测试指令检查的条件成立。 (只有从键盘执行时才会发生。)

自检信息:

35S-OK	自检及键盘检测通过。
35S-FAIL <i>n</i>	自检及键盘检测未通过，需要对计算器进行维修。
© 2007 HP DEV CO. L. P.	自检成功完成后显示版权信息。

运算索引

本节使您在恰当时迅速参阅所有函数、运算及其公式。列举按函数名称依字母顺序进行。本名称是程序行中使用的名称。例如，名为 **FIX n** 的函数作为  **DISPLAY**  (**IFIX**) 执行。

不可编程函数的名称在键框中。例如，。

非字母及希腊字符均在所有字母前字母化。前面加有箭头的函数名称（例如，**→DEG**）如果箭头不在函数名称前面，也进行字母化。

最后一列，标记为 *，指的是表格结尾处的注释。

名称	键及说明	页码	*
+/-	 改变数字的符号。	1-15	1
+	 加。返回 $y + x$ 。	1-19	1
-	 减。返回 $y - x$ 。	1-19	1
×	 乘。返回 $y \times x$ 。	1-19	1
÷	 除。返回 $y \div x$ 。	1-19	1
^	 幂。显示一个指数。	6-16	1
	删除最后键入的数字，清除 x ；清除菜单；擦除方程中最后键入的函数；删除方程；删除程序步骤。	1-4 1-8 6-3 13-7	
	显示目录中先前的输入项；移动至方程列表中的前一个方程；将程序指针移动至前一步。	1-28 6-3 13-11 13-20	

名称	键及说明	页码	*
	显示目录中的下一输入项；移动至方程列表中的下一方程；将程序指针移动至下一行（在程序输入期间），执行当前的程序行（不在程序输入期间）。	1-28 6-3 13-11 13-20	
或	使光标移动，不删除任何内容。	1-14	
或	滚屏显示左边和右边更多数位；显示方程或二进制数字的其余部分；查看 CONST 和 SUMS 菜单中的下一菜单项。	1-11 6-4 11-8	
	在程序模式下查看方程的最顶行或最后一个标签的第一行。	6-3	
	在程序模式下查看方程的最底行或最后一个标签的第一行。	6-3	
,	将函数的两个或三个自变量分开。	6-5	1
1/x	倒数。	1-18	1
10 ^x	公共指数。	4-2	1
%	返回 10 升 \times 次幂后的数值。 百分比。	4-6	1
%CHG	返回 $(y \times x) \div 100$ 。 百分比变化。	4-6	1
π	返回 $(x - y)(100 \div y)$ 。 返回近似值 3.14159265359(12 位)。	4-3	1
$\Sigma+$	将 (y, x) 累计到统计寄存器中。	12-2	
$\Sigma-$	从统计寄存器移除 (y, x) 。	12-2	
Σx	(Σx) 返回 x 值的和。	12-11	1

名称	键及说明	页码	*
Σx^2	SUMS $\>$ $\>$ $\>$ (Σx^2) 返回 x 值平方的和。	12-11	1
Σxy	SUMS $\>$ $\>$ $\>$ $\>$ $\>$ (Σxy) 返回 x 和 y 值积的和。	12-11	1
Σy	SUMS $\>$ $\>$ (Σy) 返回 y 值的和。	12-11	1
Σy^2	SUMS $\>$ $\>$ $\>$ $\>$ (Σy^2) 返回 y 值平方的和。	12-11	1
σx	S,σ $\>$ $\>$ (σx) 返回 x 值的总体标准差： $\sqrt{\sum (x_i - \bar{x})^2 \div n}$	12-7	1
σy	S,σ $\>$ $\>$ $\>$ (σy) 返回 y 值的总体标准差： $\sqrt{\sum (y_i - \bar{y})^2 \div n}$	12-7	1
\int FN d 变量	(\int FN d $_$) 变量 用 Y 寄存器中积分变量的下限以及 X 寄存器中积分变量的上限对所显示的方程或 FN= 选择的程序进行积分。	8-2 15-7	
()	括弧。按 $\>$ 退出括弧进行进一步计算。	6-6	1
[]	：用于进行矢量运算的矢量符号。	10-1	1
θ	：用于进行复数运算的复数符号。	9-1	1
A through Z	变量命名变量的数值。	6-4	1

名称	键及说明	页码	*
ABS	 ABS 绝对值。 返回 $ x $ 。	4-17	1
ACOS	 ACOS 余弦值。 返回 $\cos^{-1} x$ 。	4-4	1
ACOSH	 HYP  ACOS 双曲线弧余弦。 返回 $\cosh^{-1} x$ 。	4-6	1
 4 (4PLG)	激活代数几何模式。	1-9	
ALOG	 10^x 常用指数。 返回 10 升指定次幂后的数值 (反对数)。	6-16	1
ALL	 DISPLAY 4 (4ALL) 显示所有有效数位。可能必须向右滚动 ( >) 以查看全部数位。	1-23	
AND	 LOGIC 1 (1AND) 逻辑运算符	11-4	1
ARG	 ARG 用其自变量“ θ ”将复数替换。	4-17	1
ASIN	 ASIN 弧正弦 返回 $\sin^{-1} x$ 。	4-4	1
ASINH	 HYP  ASIN 双曲线余切值。 返回 $\sinh^{-1} x$ 。	4-6	1
ATAN	 ATAN 弧正切。 返回 $\tan^{-1} x$ 。	4-4	1
ATANH	 HYP  ATAN 双曲线弧正切 返回 $\tanh^{-1} x$ 。	4-6	1

名称	键及说明	页码	*
b	L.R. (b) 返回回归线的 y 交点: $\bar{y} - m\bar{x}$ 。	12-11	1
b	BASE (8b) 显示二进制数字	11-2	1
BASE	显示基底转换模式。	11-1	
BIN	BASE (4BIN) 选择二进制 (基底为 2) 模式。	11-1	
	打开计算器; 清除 x ; 清除信息及提示; 取消菜单; 取消目录; 取消方程输入; 取消程序输入; 停止方程执行; 停止正在运行的程序。	1-1 1-4 1-8 1-29 6-3 13-7 13-19	
$/c$	分母。 将所显示分数的分母极限设置为 x 。 如果 $x = 1$, 显示当前 $/c$ 值。	5-4	
$\rightarrow^{\circ}\text{C}$	$\rightarrow^{\circ}\text{C}$ 将 $^{\circ}\text{F}$ 转换为 $^{\circ}\text{C}$ 。	4-14	1
CF n	FLAGS (2CF) n 清除标志 n ($n = 0$ 至 11)。	14-12	
CLEAR	显示菜单清除数字或部分内存; 从 MEM 目录清除所显示的变量或程序;	1-5 1-28	
CLEAR (3ALL)	清除所显示的方程。	1-29	
CLEAR (3PGM)	清除所有保存的数据、方程及程序。	13-23	
CLEAR (3EQN)	清除所有的程序 (计算器在程序模式下)。	13-7	
CLΣ	CLEAR (4Σ) 清除统计寄存器。	12-1	

名称	键及说明	页码	*
CLVARS	CLEAR 2 (2VARS) 将所有变量清 0。	3-6	
CLx	CLEAR 1 (1X) 将 x (X 寄存器) 清 0。	2-3 2-7 13-7	
CLVARx	CLEAR 6 (6CLVARx) 将所有位址大于 x 位址的间接变量清 0。	1-4	
CLSTK	CLEAR 5 (5STK) 将所有的栈级清 0。	2-7	
→CM	→cm 将英寸转换为厘米。	4-14	1
nCr	nCr 一次所取项目 n 的组合 r。 返回 $n! \div (r! (n - r)!)$ 。	4-15	1
COS	COS 余弦。 返回 $\cos x$ 。	4-3	1
COSH	HYP COS 双曲线余弦。返回 $\cosh x$ 。	4-6	1
CONST	对 41 个物理常数进行评估。	4-8	
d	BASE 5 (5d) 显示一个二进制数字	11-1	1
DEC	BASE 1 (1DEC) 选择十进制模式。	11-1	
DEG	1 (1DEG) 选择度数角模式。	4-4	
→DEG	→DEG 半径至度数。 返回 $(360/2\pi) x$ 。	4-13	1
DISPLAY	显示菜单设置显示格式、基数 (· 或 ,)、千分位及复数的显示格式。	1-21	

名称	键及说明	页码	*
DSE variable	 DSE 变量 减量, 若等于或小于则跳过。对于保存在变量中的控制数 ccccccc.fff <i>i</i> , 用 ccccccc (计数器数值) 减 <i>i</i> (增量值), 如果结果 \leq fff (最终数值), 跳至下一程序行。	14-18	
 E	开始输入指数, 使“E”与所输入的数字相加。显示紧跟一个 10 的幂。	1-15	1
ENG <i>n</i>	 DISPLAY  (3ENG) <i>n</i> 选择第一个数位后带 <i>n</i> 个数位的工程显示 ($n = 0$ 至 11)。	1-22	
 ENG 和  ENG	使所显示数字的指数显示以 3 的倍数发生变化。	1-22	
 ENTER	将两个按顺序键入的数字分开; 完成方程输入; 对所显示的方程进行评估 (适当的话保存结果)。	1-19 6-4 6-11	
ENTER	 将 <i>x</i> 复制到 Y- 寄存器中, 将 <i>y</i> 提升到 Z- 寄存器中, <i>z</i> 提升到 T- 寄存器中, 然后丢弃 <i>t</i> 。	2-6	
 EQN	激活或取消 (开关) 程序输入模式。	6-3 13-7	
e^x	  自然指数。 返回升 <i>x</i> 次幂后的 <i>e</i> 值。	4-1	1
EXP	  自然指数。 返回升指定次幂后的 <i>e</i> 值。	6-16	1
$\rightarrow^{\circ}\text{F}$	  $\rightarrow^{\circ}\text{F}$ 将 $^{\circ}\text{C}$ 转换为 $^{\circ}\text{F}$ 。	4-14	1
 FDISP	打开 / 关闭分数显示模式。	5-1	
FIX <i>n</i>	  (1FIX) <i>n</i> 选择带 <i>n</i> 个小数位的固定显示: $0 \leq n \leq 11$ 。	1-21	

名称	键及说明	页码	*
FLAGS	显示要设置 / 清除的菜单并对标志进行测试。	14-12	
FN = 标签	FN= 标签 选择加有标记的程序作为当前函数 (由 SOLVE 及 [FN 使用)。	15-1 15-7	
FP	INTG 5 (5FP) _x 的分数部分。	4-17	1
FS? <i>n</i>	FLAGS 3 (3FS?) <i>n</i> 如果标志 <i>n</i> (<i>n</i> = 0 至 11) 处于设置状态, 执行下一个程序行。如果标志 <i>n</i> 处于清除状态, 跳至下一程序行。	14-12	
→GAL	→gal 将升转换为加仑。	4-14	1
GRAD	MODE 3 (3GRD) 设置百分度角模式。	4-4	
标签 <i>nnn</i>	将程序指针设置至程序 <i>nnn</i> 的程序行 <i>nnn</i> 。	13-21	
	将程序指针设置在 PRGM TOP。	13-21	
h	BASE 6 (6h) 显示十六进制数字	11-1	1
HEX	BASE 2 (2HEX) 选择十六进制 (基底 16) 模式。	11-1	
HYP	显示双曲线函数的 HYP_ 前缀。	4-6	
→HMS	→HMS 小时至小时、分钟及秒。 将 <i>x</i> 从小时 - 分钟 - 秒格式转换为小数分数。	4-13	1
HMS→	HMS→ 小时, 分, 秒至小时。 将 <i>x</i> 从小时 - 分钟 - 秒格式转换为小数分数。	4-13	1
	用于输入复数。	9-2	1

名称	键及说明	页码	*
(I)/(J)	$\boxed{\text{RCL}} \boxed{(I)} / \boxed{(J)}, \boxed{\text{STO}} \boxed{(I)} / \boxed{(J)}$. 其字母与变量 I/J 中保存的数值相对应的变量数值。	6-4 14-21	1
→IN	$\boxed{\leftarrow}$ $\boxed{\rightarrow}$ in 将厘米转换为英寸。	4-14	1
IDIV	$\boxed{\leftarrow}$ $\boxed{\text{INTG}} \boxed{2} (\div \text{INT})$ 得出涉及两个整数的除法运算的商。	6-16	1
INT÷	$\boxed{\leftarrow}$ $\boxed{\text{INTG}} \boxed{2} (\div \text{INT})$ 得出涉及两个整数的除法运算的商。	4-2	1
INTG	$\boxed{\leftarrow}$ $\boxed{\text{INTG}} \boxed{4} (\div \text{INTG})$ 求等于或小于给定数字的最大整数。	4-18	1
INPUT 变量	$\boxed{\leftarrow}$ $\boxed{\text{INPUT}}$ 变量 将变量恢复至 X- 寄存器, 显示变量名称及数值, 停止程序执行。按 $\boxed{\text{R/S}}$ (恢复程序执行) 或 $\boxed{\nabla}$ (执行当前程序行) 将您的输入存入变量。(仅在程序中使用。)	13-13	
INV	$\boxed{1/x}$ 自变量的倒数。	6-16	1
IP	$\boxed{\leftarrow}$ $\boxed{\text{INTG}} \boxed{6} (\div \text{IP})$ x 的整数部分。	4-17	1
ISG 变量	$\boxed{\leftarrow}$ $\boxed{\text{ISG}}$ 变量 增量, 如果更大则跳过。 对于保存在变量中的控制数 cccccc.ffff, 使 ii (增量值) 与 cccccc (计数器数值) 相加; 如果结果 > fff (最终数值), 跳过下一程序行。	14-18	
→KG	$\boxed{\rightarrow}$ $\boxed{\rightarrow}$ kg 将磅转换为千克。	4-14	1
→KM	$\boxed{\rightarrow}$ $\boxed{\rightarrow}$ KM 将英里转换为公里。	4-14	1
→L	$\boxed{\rightarrow}$ $\boxed{\rightarrow}$ l 将加仑转换为升。	4-14	1

名称	键及说明	页码	*
LASTx	LASTx 返回 LAST X 寄存器中保存的数字。	2-8	
→LB	→lb 将公克转换为磅。	4-14	1
LBL 标签	LBL 标签 给程序加上单个字母标签供 XEQ、GTO 或 FN= 运算参考。(仅在程序中使用)。	13-3	
LN	LN 自然函数。 返回对数 e^x 。	4-1	1
LOG	LOG 常用对数。 返回 $\log_{10} x$ 。	4-1	1
L.R.	显示线性回归的菜单。	12-4	
m	L.R. > > > (m) 返回回归线的曲率： $[\Sigma(x_i - \bar{x})(y_i - \bar{y})] \div \Sigma(x_i - \bar{x})^2$	12-7	1
→MILE	→MILE 将公里转换为里。	4-14	1
MEM	显示可用内存的数量及目录菜单。	1-28	
MEM 2 (2PGM)	开始程序目录。	13-22	
MEM 1 (1VAR)	开始变量目录。	3-4	
MODE	显示菜单设置 ALG, RPN 或角模式。	1-7 4-4	
n	SUMS (n) 返回数据点的组数。	12-11	1
NAND	LOGIC 5 (5NAND) 逻辑运算符	11-4	1
NOR	LOGIC 6 (6XOR) 逻辑运算符	11-4	1

名称	键及说明	页码	*
NOT	[LOGIC] 4 (4NOT) 逻辑运算符	11-4	1
o	[BASE] 7 (7o) 显示八进制数字	11-2	1
OCT	[BASE] 3 (3OCT) 选择八进制 (基底 8) 模式。	11-1	
OR	[LOGIC] 3 (3OR) 逻辑运算符	11-4	1
[OFF]	关闭计算器。	1-1	
nPr	[nPr] 一次所取 n 项目的排列数 r 。返回 $n! \div (n-r)!$ 。	4-15	1
[PRGM]	启活或取消 (开关) 程序输入模式。	13-6	
PSE	[PSE] 暂停。 短时停止程序执行以显示 x 、变量或方程, 然后恢复。(仅在程序中使用)。	13-18 13-19	
r	[L.R.] (r) 返回 x -和 y -值之间的关系系数: $\frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \times (y_i - \bar{y})^2}}$	12-7	1
rθa	[DISPLAY] 1 0 (1θrθa) 修改复数的显示。	1-25	
RAD	[MODE] 1 (1RAD) 选择半径角模式。	4-4	
→RAD	[→RAD] 度数至半径。 返回 $(2\pi/360) x$ 。	4-13	1
RADIX ,	[DISPLAY] 6 (6,) 选择逗号作为基数标记 (小数点)。	1-23	
RADIX .	[DISPLAY] 5 (5.) 选择句号作为基数标记 (小数点)。	1-23	

名称	键及说明	页码	*
RANDOM	 RAND 执行RANDOM函数。返回一个 0 至 1 范围内的随机数。	4-15	1
RCL 变量	 RCL 变量 恢复。 将变量复制到 X- 寄存器中。	3-7	
RCL+ 变量	  变量 返回 $x +$ 变量。	3-7	
RCL- 变量	  变量。 返回 $x -$ 变量。	3-7	
RCLx 变量	  变量。 返回 $x \times$ 变量。	3-7	
RCL÷ 变量	  变量。 返回 $x \div$ 变量。	3-7	
RMDR	 INTG  (3Rmdr) 得出涉及两个整数的除法运算的余数。	6-16	1
RND	 RND 四舍五入。 在 FIX n 显示模式下将 x 四舍五入至 n 个小数位, 在 SCI n 或 ENG n 显示模式下四舍五入至 $n+1$ 个或在分数显示模式下四舍五入至最接近所显示分数的小数位。	4-18 5-8	1
RPN	MODE  (5RPN) 逆波兰式记数法。	1-9	
RTN	 RTN 返回。 标记程序的结尾; 程序指针返回至顶部或调用子程序。	13-4 14-1	

名称	键及说明	页码	*
R↓	<p>[R] 向下滚动。</p> <p>在 RPN 模式下将 t 移至 Z- 寄存器, z 移至 Y- 寄存器, y 移至 X- 寄存器, x 移至 T- 寄存器。</p> <p>在 ALG 模式下显示 X、Y、Z 及 T 菜单对栈进行查看。</p>	2-3 C-7	
R↑	<p>[R] [R] 向上滚动。</p> <p>在 RPN 模式下将 t 移至 X- 寄存器, z 移至 T- 寄存器, y 移至 Z- 寄存器, x 移至 Y- 寄存器。</p> <p>在 ALG 模式下显示 X、Y、Z 及 T 菜单对栈进行查看。</p>	2-3 C-7	
[R] [S.σ]	显示标准差菜单。	12-4	
SCI n	<p>[←] [DISPLAY] [2] (2SCI) n</p> <p>选择带 n 个小数位的科学显示。($n = 0$ 至 11)。</p>	1-22	
SEED	<p>[←] [SEED] 用种子数 X 重新开始随机数顺序。</p>	4-15	
SF n	<p>[←] [FLAGS] [1] (1SF) n</p> <p>设置标志 n ($n = 0$ 至 11)</p>	14-12	
SGN	<p>[←] [INTG] [1] (1SGN) 显示 x 的符号。</p>	4-17	1
[←] [SHOW]	完整显示 x (或当前程序行中的数字) 的尾数(全部 12 位数); 显示方程及程序的十六进制校验和及小数字节长度。	6-19 13-23	
SIN	<p>[SIN] 正弦。</p> <p>返回 $\sin x$。</p>	4-3	1

名称	键及说明	页码	*
SINH	[HYP] [SIN] 双曲线正弦。 返回 $\sinh x$ 。	4-6	1
[SOLVE] 变量	[SOLVE] 变量 在变量及 x 中使用初始猜测值求解所显示的方程或 $FN=$ 选择的程序。	7-1 15-1	
[SPACE]	方程输入期间加入空格符。	14-14	1
SQ	[x²] 自变量的平方。	6-16	1
SQRT	[√x] x 的平方根。	6-16	1
STO 变量	[STO] 变量 保存。将 x 存入变量。	3-2	
STO + 变量	[STO] [+] 变量 将变量 + x 存入变量。	3-6	
STO - 变量	[STO] [-] 变量 将变量 - x 存入变量。	3-6	
STO × 变量	[STO] [x] 变量 将变量 × x 存入变量。	3-6	
STO ÷ 变量	[STO] [÷] 变量 将变量 ÷ x 存入变量。	3-6	
STOP	[R/S] 运行 / 停止。 在当前程序行开始程序执行；停止一个正在运行的程序，显示 $X-$ 寄存器。	13-19	
[SUMS]	显示总和菜单。	12-4	
sx	[S.O] ($\bar{s}x$) 返回 $x-$ 值的样本标准差： $\sqrt{\sum (x_i - \bar{x})^2 \div (n - 1)}$	12-6	1

名称	键及说明	页码	*
sy	(\bar{s}_y) 返回 y - 值的样本标准差: $\sqrt{\sum (y_i - \bar{y})^2 \div (n - 1)}$	12-6	1
TAN	正切。返回 $\tan x$ 。	4-3	1
TANH	双曲线正切。 返回 $\tanh x$ 。	4-6	1
VIEW 变量	变量 显示加有标签的变量内容而需将数值恢复到栈中。	3-4 13-15	
	对所显示的方程进行评估。	6-12	
XEQ 标签	标签 执行按标签辨别的程序。	14-1	
x ²	x^2 x 的平方。	4-2	1
\sqrt{x}	x 的平方根。	4-2	1
$\sqrt[x]{y}$	$\sqrt[x]{y}$ y 的 x 次根。	4-2	1
\bar{x}	\bar{x} (\bar{x}) 返回 x 值的平均值: $\Sigma x_i \div n$ 。	12-4	1
\hat{x}	\hat{x} (\hat{x}) 给定 X- 寄存器中的 y - 值, 根据回归线返回 x - 估计值 $\hat{x} = (y - b) \div m$ 。	12-11	1
!	! 因数(或伽玛)。返回 $(x)(x - 1) \dots (2)(1)$, or $\Gamma(x + 1)$ 。	4-15	1
XROOT	$\sqrt[x]{y}$ 自变量 ₂ 的自变量 ₁ 根	6-16	1
$\bar{x}w$	\bar{x} \bar{x} $\bar{x}w$ 返回 x 值的加权平均值: $(\Sigma y_i x_i) \div \Sigma y_i$ 。	12-4	1
\bar{x} \bar{x}	显示菜单(算术平均值)菜单。	12-4	

名称	键及说明	页码	*
$x \leftrightarrow \text{variable}$	  x 交换。 用 x 与变量进行交换。	3-8	
$x \leftrightarrow y$	 x 交换 y 将 x 移至 Y-寄存器, y 移至 X-寄存器。	2-4	
  $x?y$	显示 "x?y" 比较测试菜单。	14-7	
$x \neq y$	  $x?y$ (\neq) 如果 $x \neq y$, 执行下一个程序行; 如果 $x=y$, 跳过下一程序行。	14-7	
$x \leq y?$	  $x?y$  (\leq) 如果 $x \leq y$, 执行下一个程序行; 如果 $x > y$, 跳过下一程序行。	14-7	
$x < y?$	  $x?y$  ($<$) 如果 $x < y$, 执行下一个程序行; 如果 $x \geq y$, 跳过下一程序行。	14-7	
$x > y?$	  $x?y$  ($>$) 如果 $x > y$, 执行下一个程序行; 如果 $x \leq y$, 跳过下一程序行。	14-7	
$x \geq y?$	  $x?y$  (\geq) 如果 $x \geq y$, 执行下一个程序行; 如果 $x < y$, 跳过下一程序行。	14-7	
$x = y?$	  $x?y$  ($=$) 如果 $x = y$, 执行下一个程序行; 如果 $x \neq y$, 跳过下一程序行。	14-7	
  $x?0$	显示 "x?0" 比较测试菜单。	14-7	
$x \neq 0?$	  $x?0$ (\neq) 如果 $x \neq 0$, 执行下一个程序行; 如果 $x=0$, 跳过下一程序行。	14-7	

名称	键及说明	页码	*
$x \leq 0?$	 $x?0$  (\leq) 如果 $x \leq 0$, 执行下一个程序行; 如果 $x > 0$, 跳过下一程序行。	14-7	
$x < 0?$	 $x?0$   ($<$) 如果 $x < 0$, 执行下一个程序行; 如果 $x \geq 0$, 跳过下一程序行。	14-7	
$x > 0?$	 $x?0$    ($>$) 如果 $x > 0$, 执行下一个程序行; 如果 $x \leq 0$, 跳过下一程序行。	14-7	
$x \geq 0?$	 $x?0$     (\geq) 如果 $x \geq 0$, 执行下一个程序行; 如果 $x < 0$, 跳过下一程序行。	14-7	
$x = 0?$	 $x?0$      ($=$) 如果 $x = 0$, 执行下一个程序行; 如果 $x \neq 0$, 跳过下一程序行。	14-7	
XOR	 LOGIC  ($2 \times \text{OR}$) 逻辑运算符	11-4	1
xiy	 DISPLAY  ($9 \times i.y$) 改变复数的显示。	4-11	
$x+yi$	 DISPLAY   ($11x+y.i$) 改变复数的显示。仅限 ALG 模式。	1-25	
\bar{y}	 $\bar{x}\bar{y}$  (\bar{y}) 返回 y 值的平均值。 $\Sigma y_j \div n$ 。	12-4	1
\hat{y}	 L.R.  (\hat{y}) 给定 X- 寄存器中的 x - 值, 根据回归线返回 y - 估计值: $\hat{y} = m x + b$ 。	12-11	1

名称	键及说明	页码	*
y^x	y^x 幂。 返回升 x 次幂后的 y 值。	4-2	1

注意：

1. 函数可在方程中使用。

索引

符号

∫ FN。 见积分

10 的指数 1-15, 1-16

% 函数 4-6

 1-15

 (在分数中) 1-26

π 4-3, A-2

▲ ▼ 信号器

在分数中 5-2, 5-3

↔ 信号器

二进制数字 11-8

方程 6-7, 13-7

。 见后退键

_。 见数字输入光标

。 见积分

  信号器 1-3

 信号器 1-1

A

A...Z 信号器 1-3, 3-2, 6-4

ALG

与方程相比 13-4

在程序中 13-4

ALL 格式。 见显示格式

设置 1-23

在程序中 13-7

在方程中 6-5

B

BIN 信号器 11-1

C

%CHG 变化自变量 4-6, C-3



退出菜单 1-4, 1-8

退出目录 1-4

运算 1-4

打开 / 关闭 1-1

调整对比度 1-1

清除 X 寄存器 2-3, 2-7

清除信息 1-4

取消 VIEW 3-4

取消提示 1-4

/c 值 5-4

停止 SOLVE 7-8, 15-1

停止积分 8-2, 15-8

退出程序模式 13-7

退出方程模式 6-3, 6-4

终止程序 13-19

取消提示 6-14, 13-15

/c 值 B-4, B-6

D

DSE 14-18

E



使数字分开 1-17, 2-6

栈操作 2-6

复制数字 2-6

复制查看到的变量 13-15

结束方程 6-4, 6-7, 13-7

评估方程 6-10, 6-11

清除栈 2-6

 (指数) 1-16

ENG 格式 1-22

EQN 信号器

在程序模式下 13-7

方程列表 6-4, 6-7

F

FDISP

- 不可编程 5-10
- 开关显示模式 5-1, A-2
- 开关标志 14-9

FIX 格式 1-21

FN=

- 在程序中 15-6, 15-10
- 对函数进行积分 15-8
- 求解程序 15-1

G

GTO

- 求程序行 13-22, 14-5
- 求程序标签 13-10, 13-22, 14-5
- 求程序顶部 13-6, 13-21, 14-6

Grandma Hinkle 12-7

GTO 14-4, 14-17

H

HEX 信号器 11-1

I

i 3-9, 14-20

(i) 14-20, 14-21, 14-23

ISG 14-18

J

j 3-9, 14-20, 14-21

(j) 14-20

L

LAST X 寄存器 2-8, B-6

LASTx 函数 2-8

Łukasiewicz 2-1

M

MEM

- 变量目录 1-28

查看内存 1-28

程序目录 1-28, 13-22

O

OFF 1-1

OCT 信号器 11-1, 11-4

P

π A-2

PRGM TOP 13-4, 13-7, 13-21, F-4

PSE

使程序暂停 13-19, 15-10

防止程序停止 14-11

R

R/S

停止 SOLVE 7-8, 15-1

停止积分 8-2, 15-8

运行程序 13-22

编辑提示 6-11, 6-14, 7-2, 13-15

终止程序 13-19

恢复程序 13-16, 13-19

RCL 3-2, 13-14

RCL 算术运算 3-7

RPN

与方程相比 13-4

原点 2-1

在程序中 13-4

R \downarrow 和 R \uparrow 2-3, C-7

S

SHOW

数字数位 1-25, 13-7

提示数位 6-14

程序校验和 13-22, B-2

程序长度 13-22, B-2

方程校验和 6-19, B-2

方程长度 6-19, B-2

SPACE 14-14

SCI 格式。见显示格式

设置 1-22

在程序中 13-7

SOLVE

舍弃 D-13

使用 7-1

停止 7-2, 7-8

未求出根 7-8, 15-6, D-8

无限制 15-11

在程序中 15-6

栈上的结果 7-2, 7-7, D-3

最小或最大 D-8

初始猜测值 7-2, 7-7, 7-8, 7-12,

15-6

多个根 7-9

恢复 15-1

极 D-6

间断 D-5

检查结果 7-7, D-3

目的 7-1

平区域 D-8

评估程序 15-2

评估方程 7-1, 7-7

其作用方式 7-7, D-1

STO 3-2, 13-12

T

TVM 17-1

T 寄存器 2-5

X

XEQ

运行程序 13-10, 13-22

评估方程 6-10, 6-12

X 根自变量 6-17

X 寄存器

受提示的影响 6-14

显示 2-3

与 Y 交换 2-4

与变量交换 3-8

在程序暂停期间 13-19

在程序中清除 13-7

栈的部分 2-1

不受 VIEW 影响 13-15

不清除 2-5

测试 14-7

带变量的算术运算 3-6

清除 1-5, 2-3, 2-7

2 画

十进制模式。见基底模式

十六进制数字。见十六进制数字

6 画

至。见 GTO

舍弃

SOLVE D-13

统计 12-10

分数 5-8

积分 8-6

三角函数 4-4

7 画

十六进制数字

输入 11-1

算术 11-4

转换为 11-2

范围 11-7

时间格式 4-13

矢量

在程序中 10-7

在方程中 10-6

叉积 17-11

- 坐标转换 4-12, 9-5
- 从变量或寄存器创建矢量 10-8
- 点积 10-4
- 加 / 减 10-1
- 绝对值 10-3
- 两矢量夹角 10-5
- 使栈滚动 2-3, C-7
- 输入
 - 始终提示 14-11
 - 输入程序数据 13-12
 - 响应 13-14
 - 在 SOLVE 程序中 15-2
 - 在积分函数中 15-8
- 输入光标
 - 意义 1-17
 - 后退 1-4
- 数学
 - 一般程序 1-18
 - 栈操作 2-5, 9-2
 - 真数 4-1
 - 中间结果 2-12
 - 长计算 2-12
 - 复数 9-1
 - 计算顺序 2-14
- 数字
 - E 1-15, A-1
 - 输入 1-15, 1-16, 11-1
 - 四舍五入 4-18
 - 素数 17-7
 - 缩短 11-6
 - 显示所有数位 1-25
 - 显示格式 1-21, 11-6
 - 在程序中 13-7
 - 在方程中 6-5
 - 保存 3-2
 - 编辑 1-4, 1-17

- 真数 4-1
- 重新使用 2-6, 2-10
- 大 / 小 1-15, 1-17
- 逗号及句号 1-23, A-1
- 范围 1-17, 11-7
- 分数 1-26, 5-1
- 复数 9-1
- 负 1-15, 9-3, 11-6
- 改变符号 1-15, 9-3
- 恢复 3-2
- 基底 10-1, 13-25
- 记性算术计算 1-18
- 交换 2-4
- 精确度 D-13
- 内部表示 11-6
- 清除 1-4, 1-5, 1-17
- 求出分布 4-17
- 数字中的 E 1-15, 1-22, A-1
- 双变量统计 12-2
- 双曲函数 4-6, C-6

8 画

- 四舍五入
 - 数字 4-18
 - 分数 5-8, 13-18
- 素数发生器 17-7
- 算术
 - 十六进制 11-4
 - 一般程序 1-18
 - 八进制 11-4
 - 栈操作 2-5, 9-2
 - 中间结果 2-12
 - 长计算 2-12
 - 二进制 11-4
 - 计算顺序 2-14
- 随机数 4-15, B-4
- 提示

- 输入 13-12, 13-14, 15-2, 15-8
 - 显示隐藏的数位 6-14
 - 响应 6-13, 13-14
 - 影响栈 6-14, 13-14
 - 编程方程 14-11, 15-1, 15-9
 - 方程 6-13
 - 清除 1-4, 6-14, 13-15
 - 体积单位 4-14
 - 体积换算 4-14
 - 条件测试 14-6, 14-7, 14-9, 14-12, 14-17
 - 停止 13-19
 - 停止算术运算 3-6
 - 统计
 - 双变量数据 12-2
 - 运算 12-1
 - 单变量数据 12-2
 - 分布 16-11
 - 分组数据 16-18
 - 计算 12-4
 - 拟合曲线 12-8, 16-1
 - 统计数据。见统计寄存器
 - 输入 12-1
 - 双变量 12-2
 - 变量和 12-11
 - 初始化 12-2
 - 单变量 12-2
 - 精确度 12-10
 - 纠正 12-2
 - 清除 1-5, 12-2
 - 统计变量之和 12-11
 - 统计菜单 12-1, 12-4
 - 统计寄存器。见统计数据
 - 无分数 5-2
 - 包含总和 12-1, 12-11, 12-12
 - 查看 12-11
 - 初始化 12-2
 - 存储 12-12
 - 纠正数据 12-2
 - 清除 1-5
- ## 9 画
- 尾数 1-25
 - 未来余额 (财务) 17-1
 - 温度
 - 换算单位 4-14
 - 计算器极限 A-2
 - 问题 A-1
 - 问题答案 A-1
 - 物理常量 4-8
 - 显示
 - 显示 X- 寄存器 2-3
 - 调整对比度 1-1
 - 显示格式
 - 设置 1-21, A-1
 - 影响四舍五入 4-18
 - 影响积分 8-2, 8-6, 8-7
 - 逗号及句号 1-23, A-1
 - 默认 B-4
 - 现金流 17-1
 - 线性回归 (估计) 12-8, 16-1
 - 相关系数 12-8, 16-1
 - 小数点 A-1
 - 校验和
 - 程序 13-22
 - 方程 6-19, 13-7, 13-24
 - 信息
 - 显示 13-16, 13-18
 - 响应 1-27, F-1
 - 在方程中 13-16
 - 总结 F-1
 - 清除 1-4
 - 信号器 A-3

- 阿尔法 1-3
- 标志 14-12
- 电池 1-1, A-3
- 电量低 1-1, A-3
- 换档键 1-2
- 列表 1-13

10 画

- 虚拟部分 (复数) 9-1, C-8
- 样本标准差 12-6
- 溢出
 - 设置响应 14-9, F-4
 - 标志 14-9, F-4
 - 测试发生率 14-9
 - 计算结果 1-17, 11-5
- 因数函数 4-15
- 优先级 (方程运算符) 6-14
- 有关计算器的帮助信息 A-1
- 余弦 (三角函数) 4-4, 9-3, C-6
- 余额 (财务) 17-1
- 语法 (方程) 6-14, 6-19, 13-16
- 阿尔法字符 1-3

11 画

- 八进制数字。见数字
- 输入 11-1
- 算术 11-4
- 转换为 11-2
- 范围 11-7
- 百分比函数 4-6
- 半径
 - 转换为角度 4-14
 - 角度单位 4-4, A-2
- 运行程序 13-10
- 保存算术 3-6
- 贝塞尔函数 8-3
- 比较测试 14-7

- 变化百分比函数 4-6

变量

- 输入名称 1-3
- 数字保存 3-1
- 显示所有数位 13-15
- 与 X 交换 3-8
- 与栈分开 3-2
- 在程序中 13-12, 15-1, 15-7
- 在方程中 6-3, 7-1
- 保存 3-2
- 查看 3-4, 13-15, 13-18
- 查看时清除 13-15
- 程序输出 13-15, 13-18
- 程序输入 13-14
- 从方程保存 6-12
- 存储栈寄存器内容 B-7
- 多项式 13-26
- 恢复 3-2, 3-4
- 积分 8-2, 15-7, C-8
- 间接定址 14-20, 14-21
- 名称 3-1
- 目录 1-28, 3-4
- 内部算术运算 3-6
- 清除 1-28
- 清除全部 1-5
- 求解 7-1, 15-1, 15-6, D-1
- 变量目录 1-28, 3-4
- 标志
 - 设置 14-12
 - 未分配 14-9
 - 信号器 14-12
 - 意义 14-9
 - 溢出 14-9
 - 运算 14-12
 - 测试 14-9, 14-12
 - 方程提示 14-11

- 方程评估 14-11
- 分数显示 14-10
- 默认状态 14-9
- 清除 14-12
- 标准差
 - 正态分布 16-11
 - 分组数据 16-18
 - 计算 12-6, 12-7
- 标准差菜单 12-6, 12-7
- 表达式方程 6-10, 6-11, 7-1
- 栈提升。见栈
 - 运算 2-5
 - 不影响 B-5
 - 禁用 B-4
 - 默认状态 B-4
 - 启用 B-4
- 栈。见栈提升
 - ENTER** 的作用 2-6
 - 受提示的影响 6-14, 13-14
 - 用常数填充 2-7
 - 与变量分开 3-2
 - 与变量交换 3-8
 - 运算 2-1, 2-5, 9-2
 - 不受 VIEW 影响 13-15
 - 查阅 2-3, C-7
 - 长计算 2-12
 - 程序输出 13-12
 - 程序输入 13-12
 - 程序计算 13-14
 - 大小限制 2-4, 9-2
 - 方程用途 6-11
 - 复数 9-2
 - 滚动 2-3, C-8
 - 寄存器 2-1
 - 交换 X 和 Y 2-4
 - 目的 2-1, 2-2
- 真数
 - 运算 4-1
- 真数部分 (复数) 9-1
- 整数部分函数 4-17
- 正态分布 16-11
- 正弦 (三角学) 4-4, 9-3, A-2, C-6
- 正切 (三角学) 4-4, 9-3, A-2, C-6
- 直角坐标到极坐标转换 4-10, 9-5
- 执行程序 13-10
- 指数函数 1-17, 4-1, 9-3, C-5
- 指数曲线拟合 16-2
- 不确定性 (积分) 8-2, 8-6
- 质量换算 4-14
- 猜测值 (对于 SOLVE) 7-2, 7-7, 7-8, 7-12, 15-6
- 中间结果 2-12
- 种子 (随机数) 4-15
- 重置计算器 A-4, B-2
- 菜单
 - 使用 ... 的举例 1-8
 - 退出 1-4, 1-8
 - 一般运算 1-6
 - 列表 1-6
- 菜单键 1-6
- 测试菜单 14-7
- 查看
 - 停止程序 13-15
 - 无栈影响 13-15
 - 显示变量 3-4
 - 显示程序数据 13-15, 13-18, 15-6
- 转换函数 4-10
- 子程序
 - 程序的部分 14-1
 - 调用 14-1
 - 迭套 14-2, 15-11
- 自检 (计算器) A-5

字母键 1-3
总体标准差 12-7
常数 (填充栈) 2-7
组合 4-15
最大整数 4-18
最佳拟合回归 12-7, 16-1, C-13
长度换算 4-14
坐标
 转换 4-10

12 画

成立则进行 14-6, 15-6
乘 / 除 10-2
除法运算的商和余数 4-2
窗口 (二进制数字) 11-8
程序
 RPN 运算 13-4
 删除 1-28
 删除行 13-20
 删除方程 13-7, 13-20
 删除全部 1-5
 设计 13-3, 14-1
 使用 SOLVE 15-6
 使用积分 15-10
 输入 13-6
 数字 13-7
 数据输出 13-5, 13-14, 13-18
 数据输入 13-5, 13-13, 13-14
 提示输入数据 13-12
 停止 13-14, 13-16, 13-19
 显示长数字 13-7
 信息 13-16, 13-18
 行数字 13-22
 移动通过 13-11
 运行 13-10
 在端部返回 13-4
 暂停 13-19

编辑 1-4, 13-7, 13-20
编辑方程 13-7, 13-20
变量 13-12, 15-1, 15-7
标志 14-9, 14-12
执行 13-10
不停止 13-18
中断 13-19
逐步进行 13-11
测试 13-11
插入程序行 13-6, 13-20
子程序 14-1
长度 13-22, 13-23, B-2
错误 13-19
对于 SOLVE 15-1, D-1
对于积分 15-7
方程 13-4, 13-7
方程提示 14-11
方程评估 14-11
方法 14-1
分数 5-8, 13-15, 14-9
函数不允许 13-24
恢复 13-16
回路 14-16, 14-17
回路计数器 14-18
间接定址 14-20, 14-21, 14-23
目的 13-1
内存使用 13-22
程序输入模式 1-4, 13-6
程序标签
 输入 13-4, 13-6
 输入名称 1-3
 校验和 13-23
 移动至 13-22
 执行 13-10
 查看 13-22
 分路至 14-2, 14-4, 14-16

- 复制 13-6
- 间接定址 14-20, 14-21, 14-23
 - 目的 13-4
- 清除 13-6
- 程序指针 13-6, 13-11, 13-19, 13-21, B-4
- 程序目录 1-28, 13-22
- 程序。见程序标签
 - ALG 模式 13-4
 - 条件测试 14-7, 14-9, 14-12, 14-17, 15-6
 - 校验和 13-22, 13-23, B-2
 - 比较测试 14-7
 - 调用子程序 14-1, 14-2
 - 分路 14-2, 14-4, 14-6, 14-16
 - 基底模式 13-25
 - 计算 13-13
 - 目录 1-28, 13-22
 - 清除 13-6, 13-22, 13-23
 - 清除全部 13-6, 13-23
- 错误
 - 纠正 2-8, F-1
 - 清除 1-4
- 错误信息 F-1
- 打开及关闭 1-1
- 代数几何模式 1-9
- 单位转换 4-14
- 单变量统计 12-2
- 单步执行 13-11
- 等式方程 6-9, 6-11, 7-1
- 电池 1-1, A-3
- 迭套子程序 14-2, 15-11
- 定址
 - 间接 14-20, 14-21, 14-23
- 逗号（数字中） 1-23, A-1

13 画

- 度数（角度单位） 4-4, A-2
- 对数函数 4-1, 9-3, C-5
- 对数曲线拟合 16-2
- 对比度调整 1-1
- 对计算器进行测试 A-4, A-5
- 多项式 13-26
- 二进制数字。见数字
 - 输入 11-1
 - 算术 11-4
 - 查看所有数位 11-8
 - 转换为 11-2
 - 范围 11-7
 - 滚动 11-8
- 方程
 - TVM 方程 17-1
 - 删除 1-5, 6-9
 - 使用 6-1
 - 输入 6-4, 6-8
 - 数值 6-10, 6-11, 7-1, 7-7, 13-4
 - 数字 6-5
 - 提示输入数值 6-11, 6-13
 - 无根 7-8
 - 显示 6-6
 - 校验和 6-19, 13-7, 13-24
 - 与 ALG 模式相比 13-4
 - 与 RPN 相比 13-4
 - 语法 6-14, 13-16
 - 运算符优先级 6-14
 - 在程序中 13-4, 13-7, 13-24, 14-11
 - 在程序中删除 13-20
 - 在程序中输入 13-7
 - 在程序中显示 13-16, 13-18, 14-11
 - 在程序中编辑 13-7, 13-20

- 造作总结 6-3
- 保存变量数值 6-12
- 编辑 1-4, 6-8
- 变量 6-3, 7-1
- 栈的用途 6-11
- 长 6-7
- 长度 6-19, 13-7, B-2
- 作为应用 17-1
- 程序中的提示 14-11, 15-1, 15-8
- 带 (I)/(J) 14-23
- 多个根 7-9
- 根 7-1
- 滚动 6-7, 13-7, 13-16
- 函数 6-5, 6-16, G-1
- 基底模式 6-5, 6-11, 13-25
- 积分 8-2
- 及分数 5-9
- 控制评估 14-11
- 括号表示 6-6, 6-15
- 类别 6-9
- 内存 13-16
- 评估 6-10, 6-11, 6-12, 7-7, 13-4, 14-11
- 求解 7-1, D-1
- 方程列表
 - EQN 信号器 6-4
 - 显示 6-6
 - 在方程模式下 6-3
 - 造作总结 6-3
 - 编辑 6-8
 - 加 6-4
- 方程列表顶部 6-7, F-2
- 方程模式
 - 退出 1-4, 6-3
 - 显示方程列表 6-3
 - 在程序输入期间 13-7
- 后退 1-4, 6-8
- 开始 6-3, 6-7
- 分数
 - 舍弃 5-8
 - 设置格式 5-6, 14-10, 14-14
 - 输入 1-26
 - 四舍五入 5-8
 - 无统计寄存器 5-2
 - 显示 5-2, 5-4, A-2
 - 标志 14-9
 - 分母 1-26, 5-4, 14-10, 14-14
 - 格式 5-6
 - 及程序 5-10, 13-15, 14-9
 - 及方程 5-9
 - 减 5-2, 5-6
 - 精度指示器 5-2, 5-3
- 分数显示模式
 - 设置 5-1, A-2
 - 影响 VIEW 13-15
 - 影响四舍五入 5-8
- 分数部分函数 4-17
- 分组标准差 16-18
- 分路 14-2, 14-16, 15-7
- 分母
 - 设置最大值 5-4
 - 范围 1-26, 5-2
 - 控制 5-4, 14-10, 14-14
- 符号 (数字) 1-15, 9-3, 11-6
- 符号值 4-17
- 符号惯例 (财经) 17-1
- 赋值式方程 6-9, 6-11, 6-12, 7-1
- 复数
 - 输入 9-1
 - 运算 9-2
 - 在栈上 9-2
 - 查看 9-2

- 自变量值 4-17
- 坐标系统 9-5
- 付款 (财务) 17-1
- 负数 1-15, 9-3, 11-6
- 改变数字符号 1-15, 9-3
- 概率
 - 正态分布 16-11
- 函数 4-15
- 根函数 4-3
- 根。见 SOLVE
 - 未发现 7-8, D-8
 - 在程序中 15-6
 - 程序 15-1
 - 多个 7-9
 - 方程 7-1
 - 检查 7-7, D-3

14 画

- 暂停。见 PSE
- 估计值 (统计) 12-8, 16-1
- 故障查找 A-4, A-5
- 滚动
 - 二进制数字 11-8
 - 方程 6-7, 13-7, 13-16
- 函数
 - 双自变量 1-18, 2-9, 9-4
 - 显示屏上的名称 13-8
 - 在方程中 6-5, 6-16
 - 真数 4-1
 - 不可编程 13-24
 - 单自变量 1-18, 2-9
 - 列表 G-1
- 函数最小值 D-8
- 函数最大值 D-8
- 函数的不一致性 D-5
- 函数的极 D-6
- 后退键

- 删除程序行 13-20
- 退出菜单 1-4, 1-8
- 运算 1-4
- 方程输入 1-4
- 清除 X 寄存器 2-3, 2-7
- 清除信息 1-4
- 取消 VIEW 3-4

换算

- 时间格式 4-13
- 数字基底 10-1, 11-1
- 温度单位 4-14
- 质量单位 4-14
- 长度单位 4-14
- 坐标 4-10
- 角度单位 4-13
- 角度格式 4-13
- 换档键 1-3
- 恢复算术 3-7
- 回归 (线性) 12-7, 16-1
- 回路 14-16, 14-17
- 回路计数器 14-18, 14-23
- 霍纳法 13-26

15 画

- 模式。见角模式、基底模式、方程模式、分数显示模式及程序输入模式
- 基数标记 A-1
- 基底
 - 设置 11-1
 - 算术 11-4
 - 影响显示 11-6
 - 转换 11-2
 - 程序 11-8, 13-25
 - 默认 B-4
- 基底模式
 - 设置 13-25
 - 编程 13-25

- 方程 6-5, 6-11, 13-25
 - 默认 B-4
 - 积分
 - 使用 8-2, C-8
 - 所需时间 8-6, E-7
 - 停止 8-2, 15-8
 - 显示格式 8-2, 8-6, 8-7
 - 限制 15-11
 - 在程序中 15-10
 - 变量 8-2, C-8
 - 栈上的结果 8-2, 8-6
 - 转换变量 E-9
 - 准确度 8-2, 8-6, E-1
 - 子区间 E-7
 - 极限 8-2, 15-8, C-8, E-7
 - 结果不确定性 8-2, 8-6, E-2
 - 困难函数 E-2, E-7
 - 目的 8-1
 - 内存使用 8-2
 - 评估程序 15-7
 - 其作用方式 E-1
 - 积分极限 8-2, 15-8, C-8
 - 极坐标到直角坐标转换 4-10, 9-5
 - 计算器
 - 有关 ... 的问题 A-1
 - 重新设置 A-4, B-2
 - 测试操作 A-4, A-5
 - 打开及关闭 1-1
 - 调整对比度 1-1
 - 短接 A-5
 - 环境限值 A-2
 - 默认设置 B-4
 - 计算器的湿度极限值 A-2
 - 夹角
 - 换算单位 4-13
 - 换算格式 4-13
 - 两矢量间 10-5
 - 默认单位 4-4, A-2
 - 加权数换算 4-14
 - 加权平均值 12-4
 - 间接定址 14-20, 14-21, 14-23
 - 键
 - 移动 1-3
 - 阿尔法 1-3
 - 字母 1-3
 - 交点（拟合曲线）12-8, 16-1
 - 角度
 - 转换为半径 4-14
 - 角度单位 4-4, A-2
 - 角模式 4-4, A-2, B-4
 - 借出人（财务）17-1
 - 借款人（财务）17-1
 - 珈玛函数 4-15
 - 金钱的时间价值 17-1
 - 金融计算 17-1
- ## 16 画
- 精确度（数字）1-25, D-13
 - 绝对值（真数）4-17
 - 款项（财务）17-1
- ## 17 画
- 括号表示
 - 算术运算中 2-12
 - 在方程中 6-5, 6-6, 6-15
 - 利息（财务）17-3
 - 连续内存 1-1
 - 链计算 2-12
 - 两个补数 11-4, 11-6
 - 流程图 14-2
- ## 18 画
- 逻辑
 - AND 11-4

NAND 11-4
NOR 11-4
NOT 11-4
OR 11-4
XOR 11-4

19 画

幂信号器 1-1, A-3
幂函数 1-17, 4-2, 9-3
幂曲线拟合 16-2

20 画

模式菜单
 角模式 4-4
目录
 使用 1-28
 退出 1-4
 变量 1-28, 3-4
 程序 1-28, 13-22

21 画

内存
 用途 B-1
 栈 2-1
 程序 13-21, B-2
 大小 1-28, B-1
 关闭时保持 1-1
 可用数量 1-28
 满 A-1
 清除 1-5, 1-29, A-1, A-4, B-1, B-3
 清除统计寄存器 12-2
 清除变量 1-28
 清除程序 1-28, 13-6, 13-22
 清除方程 6-9
内存已满 B-1, F-3
内存清除 A-4, B-3, F-3
拟合曲线 12-8, 16-1
逆 / 正态分布 16-11

逆双曲线函数 4-6
逆函数 9-3
逆三角函数 4-4, C-6

22 画

排列 4-15
平均数 (统计)
 正态分布 16-11
 计算 12-4
平均值菜单 12-4
期限 (数字) 1-23, A-1
清除
 X 寄存器 2-3, 2-7
 数字 1-17
 统计寄存器 12-2
 一般信息 1-4
 变量 1-28
 程序 1-29, 13-23
 方程 6-9
 内存 1-29, A-1
清除菜单 1-5
清除内存 A-4, B-3
曲率 (曲线拟合) 12-8, 16-1
三角函数 4-4, 9-3, C-6

