

美 Richard J. Gaylord | Samuel N. Kamin | Paul R. Wellin 著

数学软件

邵勇译

Mathematica

高等教育出版社

入门



788879
346598679537676777
5786787890289
35642557898706795
363487687746734
4378677898
85687878928657
446534652678757
786687479868795
548697488347565
3476934687578669
677797896789
5437586745674
475786778754867897
6890789036799036893
5895834765743978
667594798769543670357
886753478798789064787
78509367543907657564
76972786786534798798784
88904-6587908609548790895
089456908784059869785678567587
598734785698790879654078006
4568769540706547845876609870698780
704776074576458746897569476457465
78765987644533473358765487659876587647543334647
45676587658765987698706856875678589857609857
8654376583687968365436873487568456734896735496876958769
098765897683547657346754397834786
98769654367853497657845847584876345676587645764536
46534765764587

TP312MA 328
G39

数学软件 Mathematica 入门

[美] Richard J. Gaylord | Samuel N. Kamin | Paul R. Wellin 著
邵 勇 译

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载，
也可到视听部复制



A0937850

高等教育出版社

图字:01-1998-2223号

Translation from the English language edition:

An Introduction to Programming with Mathematica by

Richard J. Gaylord, Samuel N. Kamin and Paul R. Wellin

Copyright © 1996 Springer - Verlag New York, Inc.

Published by TELOS, The Electronic Library of Science, Santa Clara, California

All Rights Reserved

数学软件 Mathematica 入门

中文版

© 高等教育出版社

版权所有

此中译本为 Springer - Verlag New York, Inc. 英文版原书在中华人民共和国境内的唯一合法译本

图书在版编目 (CIP) 数据

数学软件 Mathematica 入门 / (美) 盖洛德 (Gaylord, R. J.), (美) 卡明 (Kamin, S. N.), (美) 威廉 (Wellin, P. R.) 著; 邵勇译. —北京: 高等教育出版社, 2001

ISBN 7-04-008832-0

I. 数… II. ①盖…②卡…③威…④邵… III. 工程计算—应用软件, Mathematica IV. TB115

中国版本图书馆 CIP 数据核字 (2000) 第 77404 号

数学软件 Mathematica 入门

[美] Richard J. Gaylord Samuel N. Kamin Paul R. Wellin 著

邵 勇 译

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

邮政编码 100009

电 话 010-64054588

传 真 010-64014048

网 址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

经 销 新华书店北京发行所

印 刷 北京二二〇七工厂

开 本 787×1092 1/16

版 次 2001 年 1 月第 1 版

印 张 30

印 次 2001 年 1 月第 1 次印刷

字 数 680 000

定 价 33.60 元

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换

版权所有 侵权必究

前 言

我们大家都可从事的技术计算

计算机的出现已使研究工作的性质、科学和工程方面的教育发生了本质上的变化。实验科学家一般是使用计算机来收集和分析数据，而理论科学家使用计算机是从数值上和符号上对方程进行计算。对这两类科学家来说，计算机模拟研究已经成为一项必不可少的研究工具。

作为对这一变化的回应，技术教育也在渐渐改变自身，把计算机既作为讲授的内容，又作为展示技术内容的媒介引进到课堂教学中。

计算机在所有这一切领域中的应用正飞速地进行着，这是因为计算机硬件的发展已达到这样一种程度，即每个人都支付得起购买他（或她）自己的独立自足的计算机环境的费用。甚至有可能，大学或学院为筹集建立台式计算机实验室所需资金而做的努力，因学生完全可以携带着他们自己个人的功能强大的笔记本电脑来上课而变得不必要了。

可尽管硬件难关已被突破，但另一障碍却又阻止着这一技术上的计算能力的大众化传播。这一障碍就是由于一直比较缺乏强大而又界面友好的软件造成的。虽然存在有许多软件包可执行各种类型的技术计算工作，但提供其中包含有适合非专业程序员进行程序设计以满足他们的计算需要的程序设计语言的完全集成化的技术计算环境的软件却少得可怜。而 Mathematica 却提供了这样一种工作环境。它包含：

1. 内置的数学功能和图形功能。它既强大又灵活。
2. 一种实际上可用来无限扩展其功能的程序设计语言。这一语言是交互式的：既具有数值操作功能，又具有符号操作功能；广泛使用了模式匹配；支持许多计算机科学家使用他们喜欢的函数型程序设计风格（当然同时支持传统程序设计风格）。
3. 广泛的在线帮助功能。比如，函数浏览器，它是从 2.2 版开始加入的一个新特性（2.2 版之后又改为叫 Help Browser，功能类似），它使学习内置函数及获得它们的正确语法变得非常容易。
4. 把 Mathematica 与其他计算环境和其他语言联系到一起的功能。
5. 一种可使文本和图形同时出现在文档中的界面。

在本书中，我们的重点是 Mathematica 程序设计语言。既然已有许多书，包括 Stephen Wolfram 写的使用手册[Wolfram 1991]在内，讲述了 Mathematica 的各个不同的方面，那么本书就如何使用最基本的程序设计语言以使 Mathematica 功能得以充分发挥这一点做一文字上的讲解就是非常必要的了。本书是为初级程序员讲述 Mathematica 程序设计语言的第一本书，着重点放于 Mathematica 中最具效率且最具特色的两种程序设计风格：函数风格和变换风格（或规则库风格）。

本书的读者

本书为两类不同读者编写：

- 第一类为不曾有过任何程序设计经验的 Mathematica 用户。我们将从最基本的内容讲起，比如解释如何创建**嵌套函数调用** (nested function call)，如何创建**匿名**(anonymous)函数及**高阶**(higher-order)函数。还要讲到递归程序设计和迭代程序设计。

我们在初学者类别中还包括进那些曾使用传统语言如 Fortran, C, BASIC 和 Pascal 等进行过程设计的 Mathematica 用户。从许多实际例子可推知，这些程序设计者将会以一种过程型程序设计风格编写 Mathematica 程序，编出的代码像是，比如说，“Mathematica 中的 Fortran”。本书在使你理解如何用函数程序设计风格和规则库程序设计风格来构造 Mathematica 程序的基础上，让你懂得用这两种风格编写出更加简洁和更加有效的代码是完全可能的。

- 第二类为想要学习程序设计并打算从一种友好的和实用的语言开始的用户。有些语言，如 C, Fortran，需要写出太多行复杂的代码才能做一些有趣的事情，而另一些语言，如 LISP，却使用非自然语法，因此你又不易掌握。Mathematica 具有更加自然的语法，并且提供有高水平的内置操作，因此拿来即可做许多有趣的事情。

第二版中所做的一些改进

这一版有两个主要变化——一是在练习题和它们的解答方面，二是增加了有关应用的一章。另外，还有几百个小的变化，主要是有关内容重新组织方面的变化和其他一些非实质性的变化。

相对于上一版，这一版增加了几十个练习题，其中有些难度适中，而有些难度则比较大。另外，大多数练习题，本书给出了答案，并在需要时加写几句有关说明以对读者加以提示。本书有答案的练习题数量超过一半。而其他练习题的答案也大都可在随书附带的软盘中找到。

我们这版增加了有关应用 (application) 的一章，即第十一章，并把有关目录 (context) 和程序包 (package) 的一章改为第十二章。第十一章中的应用，范围极广，这就给人一种印象：Mathematica 可以用来解决各种各样在计算上极具复杂性的问题。第十二章（目录和程序包）进行了扩充，即增加了有关如何把选项及缺省值结合进程序包中的讨论。

在全书的某些地方，提及了将要在 Mathematica 2.2 版之后的版本中可能会出现的新特性。针对那些与 Mathematica 2.2 版及更早的版本有着明显不同的特性，我们特别地给出注解来说明这些不同之处。

本书第一版的读者发现的少量错误，在第二版中都进行了改正。我们欢迎读者多与我们交流和提出宝贵意见。

如何使用本书

有几种使用本书的方式：

- 作为有关程序设计基础课程的初级课本。大概最明显的用途是作为为学习自然科学、工程或数学的学生开设的计算机科学基础的课本。这些学生解决他们感兴趣的问题的同时，也就学会了程序设计。Mathematica 程序设计语言支持许多种程序设计风格，这些风格可通过转换，用于其他程序设计语言中；因此这里所学到的用 Mathematica 进行的程序技巧，也可以被移植到传统的程序设计语言中。

- 在有的课程中，Mathematica 可被用作研究其他技术课题的工具，那么本书可作为这种课程的辅助教材。在这些课程中，Mathematica 程序设计的原理在授课中随需随讲，但我们还是认为在学期的开始就有系统地介绍 Mathematica 程序设计是最好的。

- 也可作为自学用书，主要适用于那些需要充分利用 Mathematica 的程序设计功能的 Mathematica 用户及那些有兴趣了解 Mathematica 程序设计语言是如何工作的人员。

本书中的一些规定

本书中所有表示输入和输出的内容都以与一般正文不同的字体出现。出现在书中的 Mathematica 程序代码是这样，出现在行文中的例子，如 `Expand[(a+b)^4]` 也是这样。于是，比如说，输入行（也就是你从计算机键盘录入的内容）就这样显示：

```
In[1]:= 3 + 5
```

同时，所有的输出内容（即 Mathematica 经计算在你的计算机屏幕上显示出的结果）是以一种比输入内容稍浅一些的同字体显示出：

```
Out[1]= 8
```

你不必自己输入提示符 `In[1]:=` 或 `Out[1]=`；Mathematica 会自动为你显示。

本书中定义的所有程序，除被放于一张软盘中外，还以另外两种不同的方式供你查寻——一种是以程序名本身出现于书后索引中，另一种是出现在索引中的程序词条之中。所以，比如说，7.3 节中定义的 `runEncode` 函数，作为一个词条 `runEncode` 出现于索引中，同时，还出现于程序词条之中。

本书包含内容

我们大概应该首先从本书不包含的内容开始。未包含的内容是有关 Mathematica 中几百种内置操作的一览表和对它们的解释。要想查看这个表，[Wolfram 1991]肯定是必不可少的。

本书要教你如何去进行程序设计。我们假设你从未进行过计算机程序设计（尽管你可能已经是一名 Mathematica 用户），我们要在本书中引导你一步步地熟悉各种程序设计方法，这些方法你大概希望在 Mathematica 中得以使用。

各章内容安排如下：

第一章：预备知识。这一章包含使用 Mathematica 应用程序应具备的一些基本操作技能，比如，如何使用 Mathematica 界面，如何输入表达式。有经验的 Mathematica 用户阅读这章会非常轻松。

第二章：Mathematica 概述。对一点都没有接触过 Mathematica 的新手，我们编写本章，给出 Mathematica 的一些内置功能的实例，包括图形功能在内。与第一章一样，你若已是一名 Mathematica 用户，你也可轻松自如地读完本章。

第三章：表操作。本书核心内容从本章开始，即讨论表这一最为重要的数据类型。除数字外，大概表这一类型在程序设计中最为有用。（例如，我们的保龄球计分程序包含的基本上是表的操作。）我们在本章中将讨论如何用内置函数对表进行操作。

第四章：函数。本章继续第三章的讨论，引入嵌套函数调用和高阶函数，本章和上章中介绍的内置函数之间的结合，可用以解决许许多多的程序设计问题。

第五章：表达式的计算。在进一步讨论程序设计之前，我们认为有必要对 Mathematica 如何执行你的计算工作给以确切的解释。这就要求我们对基于规则的程序设计技术给予介绍。

第六章：条件函数定义。计算机的一个充满活力的属性便是可依据输入其中的内容的性质来做出决定以采取不同的行动。这一章讨论 Mathematica 程序员是如何将这种做决定的能力加进其程序中的。

第七章：递归。这一程序设计方法——其中一个函数的定义中包含这个函数本身——在 Mathematica 中应用极为广泛。

第八章：迭代。这又是一种程序设计方法。在其中，结果的获得则要通过重复进行同一过程指定的次数来达到。这一程序设计风格更是一种传统语言如 C 语言的风格而非 Mathematica 风格，尽管如此，它还是极为重要的。

第九章：数。Mathematica 包含有许多种类型的数。这一章讨论它们之间的相同处与不同处，并指出一些问题，这些问题在对非精确数和精确数进行计算时必定会遇到。

第十章：图形程序设计。这一章引进 Mathematica 图形的基本概念，并使用前面各章中的一些方法来创建基于图形的程序和解决本质上与图形关系更加密切的问题。

第十一章：应用。Mathematica 可用来解决大规模程序设计问题。本章中所含内容有：随机游动；用函数程序设计方法和基于规则的程序设计方法使生命游戏在 Mathematica 中得以实现；最后是小型图象描述语言的创建。

第十二章：目录和程序包。这是最后一章。它介绍了如何把 Mathematica 函数库组织成为一个个方便易用的单元的方法，这一个个单元被称作程序包。

软 盘

本书中的内容有些可以以电子形式获得。包含有大部分的例题、练习题和练习题解答的 Mathematica 笔记本被存于一张软盘中附于本书中，以便于读者在计算机中实际操作。文件名与书的章结构一致。比如，包含第一章（chapter 1）内容的笔记本的文件名为

Chap1.ma。

软盘中还包含有需要时即可加载的存贮了 Mathematica 程序的文件。这些文件（一般被叫做程序包（package）），都具有 .m 的扩展名。它们可以在你运行 Mathematica 后被使用，但首先要核实这些文件确实是放于 Mathematica 可以找得到的路径中（SetDirectory[" Path To files "]），然后把它们读进（<<file.m）。

所附软盘为一张 3.5 "、1.44 兆字节存贮量的高密度磁盘，它是以 DOS 兼容格式被格式化的。除可以被任何 Unix 机、NeXT 机或 Macintosh 机读出外，它可以被任何 IBM 兼容机读出。我们可以以不同的磁盘操作系统格式，如 DOS，Macintosh 及 Unix，获取这些文件。这就避免了文件转换的问题。磁盘中还包含有一个名为 Readme.txt 的文件，它提供了有关这一磁盘用途的更加详细的信息。

除书中所附软盘外，这些内容还可通过 Wolfram Research 的 MathSource 获得。MathSource 为一 Mathematica 材料的电子发行服务公司。要想获得有关 MathSource 的信息，可发一份 Email 到 mathsource@wri.com，在附言中加一句 help intro 即可。如果只想获取本书中的某些材料，那么附言中就加这样一句：find 0204938。MathSource 于是会发给你一个包含你要的材料文件，你便可下载它。intro 文件对 MathSource 进行了详细的描述，包括有关如何查询和获取其所提供的服务的说明。

有关本书内容的 Mathematica 笔记本和程序包还可以通过万维网（World-Wide Web）进行浏览，方法是把你的网络浏览器接到 <http://www.telospub.com>，这一站点中的内容会根据需要定期更新。

请读者注意的最后一点

本书讨论了 Mathematica 程序设计的基本方面，其实还有许多有关 Mathematica 及 Mathematica 程序设计语言的事情尚未讲到。要想获得这更多的内容，请查阅与 Mathematica 软件一同出售的参考手册：Mathematica, A System for Doing Mathematics by Computer[Wolfram 1991]。

MathSource 包含从科学和工程中一切领域引用来的成百个程序设计的例子和笔记本。为获取这些材料，你可以发 Email 给 mathsource@wri.com。有关如何进入 MathSource 中的这些材料的说明，你可以在随本书附上的软盘中找到。

对那些使用互联网的读者，存在一个专注于 Mathematica 的名为 comp.soft-sys.math.mathematica 的消息组（newsgroup）供你访问。另外，你还可得到一份这个消息组的发送名单，这一名单基本上是消息组的反映。为使自己也登录于这一发送名单中，可发一条信息给 mathgroup-request@christenser.cybernetics.net 进行申请。进一步，<http://www.wri.com/techsupport/> 的 FAQ 可以回答你一些普遍性的问题。

Mathematica in Education and Research 是一份由 TELOS/Springer-Verlag 出版的季刊，上面登载一些有关如何在课堂上使用 Mathematica 的文章和短文。它包含一些有关程序设计方面的固定栏目，还包含一个学生栏目。若要订阅，请寄信给：Mathematica in Education

and Research, *TELOS/Springer-Verlag*, 333 Meadowlands Parkway, Secaucus, NJ 07094 USA, 或发 Email 给 MathInEd@telospub.com。

Mathematica Journal 则是一份登载有关 *Mathematica* 的一切方面的文章。要想订阅, 请与 *Miller Freeman, Inc.*, 600 Harrison Street, San Francisco, CA; 415-905-2334 联系。

最后, 现如今已有 100 多种有关 *Mathematica* 本身或如何使用 *Mathematica* 去教授某一课程的书。本书末尾的参考文献中列出了其中的一些。要想获得一份完整的、最新的关于 *Mathematica* 的出版物的清单, 请与 *Wolfram Research* 联系。

本书制作情况

本书是从 *TEX* 源文件和 *Mathematica* 笔记本转换到 *LATEX* 而排版制作的。除少数几处明显提到 3.0 版本的新特性外, 全书总体上使用的是 *Mathematica* 2.2 版本。笔记本是通过使用可从 *MathSource* 获得的 *nb2tex* 转换器转换到 *LATEX* 文件来的。所有图形是用 *Mathematica* 制作的, 并通过使用得自 *epsf.sty* 这一类型文件中的选项的方法, 移置到 *TEX* 文件中。*PostScript* 文件是用 *dvips* 产生的, 并在出软片之前被分解成独立页的 *PostScript*。

鸣 谢

在本书的出版过程中, 我们有幸获得 *TELOS* 和 *Wolfram Research, Inc.* 这两家公司给予的宝贵帮助。

我们的出版商 *Allan Wylde*, 在我们向他提出要出版本书时, 立即意识到了此书的市场需求, 并且在本书长达九个月的编写过程中积极鼓励和支持我们的工作。我们在此对他在处理这项非传统项目当中显现出的敏锐目光和决断能力表示衷心的感谢。出版伙伴 *Kate McNally-Young*, 出版助理 *Keisha Sherbacoe* 及作品编辑 *Jan Benes*, 都在本出版项目上提供了许多有益的帮助。

Wolfram Research 公司的许多人都以不同的方式对本项目提供了大力帮助, 包括技术上的帮助, 提供新版 *Mathematica* 的第二拷贝的帮助, 原稿审读方面的帮助及对我们的努力报以喝彩的帮助。特别地, 我们还要感谢 *Stephen Wolfram*, *Prem Chawla*, *Joe Kaiping*, *Andre Kuzniarek*, *Ben Friedman*, *Mark Moline*, *Amy Young* 及 *Joe Grohens* 为我们提供的帮助。

原稿的审读者向我们提供了非常有用的意见和建议, 为此, 我们要向他们, *Claude Anderson*, *Bill Davis*, *Allan Hayes*, *Ralph Johnson*, *Jerry Keiper*, *John Novak*, *Jose Rial*, *Michael Schaferkotter*, *Bruce Smith* 及 *Doug Stein* 表示感谢。我们还要对第二版的审读者 *Claude Anderson*, *Paul C. Hoffman*, *Martin Maxey*, *Stephanie Rieser* 及 *Winthrop W. Smith* 表示感谢。

Richard J. Gaylord 感谢他的老师 *Shawn Sheridan* 教给了自己 *Mathematica* 和 *Macintosh*。

Samuel Kamin 感谢伊利诺思大学计算机科学系为我们提供了优越的工作环境及优秀

的员工。总之，他想感谢 Judy 和 Rebecca 对这项工作的热情支持和耐心帮助。

Paul Wellin 感谢他的妻子 Sheri 在这项工作中对他的支持、理解和给予的幽默开心。

最后，我们真诚希望读此书的读者对本书及所附软盘中材料提出批评指正，以便我们今后在新版中加以改正。

Richard J. Gaylord
(gaylord@ux1.cso.uiuc.edu)

Samuel N. Kamin
(kamin@cs.uiuc.edu)

Paul R. Wellin
(wellin@groucho.sonoma.edu)

1995 年 8 月

第一章 预备知识

Mathematica 是一个非常庞大和看似很复杂的系统。它包含了几百个可在科学、数学和工程中执行各种任务的函数。它具有其规则和语法都定义得很好的自己的语言。在本预备知识一章，我们将给出一些 Mathematica 用户需要知道的基本知识，例如如何启动 Mathematica，如何从中退出，如何输入简单的内容并获得相应结果，等等。另外，我们还将给出一点点有关这一系统内部结构的知识，为的是我们能够从这里开始一步步理解和掌握这本书最终要传授给你的东西——Mathematica 程序设计语言。

1.1 引言

Mathematica 被描述成一个高级的计算器是很恰当的。有了它，你可以输入复杂的公式并求得公式的值。

```
In[1]:= Sin[.86] - Log[1.23] (1 + .08/12)^12  
Out[1]= 0.533646
```

你可以把得出的值存贮进来。

```
In[2]:= rent = 350  
Out[2]= 350
```

```
In[3]:= food = 175  
Out[3]= 175
```

```
In[4]:= heat = 83  
Out[4]= 83
```

```
In[5]:= rent + food + heat  
Out[5]= 608
```

你可以进行诸如对多项式进行因式分解、求函数的根、计算定积分和不定积分、画函数图形以及比这些多得多的操作。

```
In[6]:= Factor[x^2 - 1]
```

```
Out[6]= (-1 + x) (1 + x)
```

```
In[7]:= Solve[x^3 - 2x + 1 == 0, x]
```

```
Out[7]= {{x -> 1}, {x ->  $\frac{-1 - \sqrt{5}}{2}$ }, {x ->  $\frac{-1 + \sqrt{5}}{2}$ }}
```

```
In[8]:= Integrate[x^2.5 - 5.0911x, {x, 0, 20}]
```

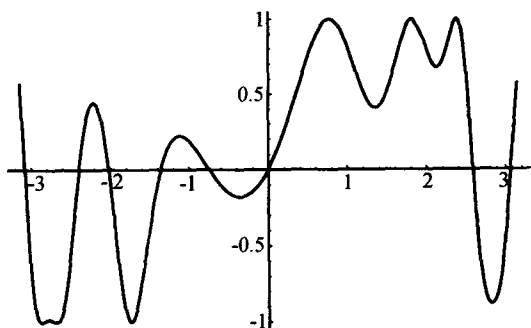
```
Out[8]= 9203.81
```

```
In[9]:= Integrate[1/(1 - x^3), x]
```

```
Out[9]= 
$$\frac{\frac{\text{ArcTan}\left[\frac{1 + 2x}{\sqrt{3}}\right]}{\sqrt{3}} - \frac{\text{Log}[1 - x]}{3} + \frac{\text{Log}[1 + x + x^2]}{6}}$$

```

```
In[10]:= Plot[Sin[x + Sqrt[2] Sin[x^2]], {x, -Pi, Pi}]
```



```
Out[10]= -Graphics-
```

Mathematica 具有方便的用户界面，在其中，你可以使用%符号容易地查找前一步计算的结果。

```
In[11]:= 2^64
```

```
Out[11]= 18446744073709551616
```

```
In[12]:= %
Out[12]= 18446744073709551616
```

你还可以使用 `Out[i]` 标志或等价的 `%i` 来查找任何一个较早计算出的结果。

```
In[13]:= Out[1]
Out[13]= 0.533646
```

```
In[14]:= %6
Out[14]= (-1 + x) (1 + x)
```

拥有一份由 Stephen Wolfram 编写的参考手册[Wolfram 1991]，或拥有一份列有 Mathematica 中包含的几百种操作的缩编的参考手册 ([Blachman 1991]或[Wolfram 1992])，那么，看来你就可以计算你想要计算的任何事情了。

但是，这一印象是错误的。还有更多类型的计算，它们也是完全有可能被写进一个单一程序中的。不管你是对计算保龄球得分感兴趣，还是求在一个环形圆纹曲面上进行的随机游动的均方距离，Mathematica 都没有提供给你某个内置函数去实现你的目的。Mathematica 真正拥有的——也是真正使它成为用途惊人的工具的——是用户自己去定义他们需要的函数的能力。这种能力被称作**程序设计** (programming)，其实它就是本书真正要传授给你的东西。

为使你对自己能做什么有个初步认识，这里提供了一个程序来计算保龄球比赛得分。已知条件是一个由每一球所得分数构成的表。

```
In[15]:= BowlingScore[pins_] := Module[{score},
  score[{x_, y_, z_}] := x + y + z;
  score[{10, y_, z_, r___}] := 10+y+z+score[{y, z, r}];
  score[{x_, y_, z_, r___}] := x+y+z+score[{z, r}] /;
    x + y == 10;
  score[{x_, y_, r___}] := x+y+score[{r}] /; x+y < 10;
  score[If[pins[[-2]] + pins[[-1]] >= 10, pins,
    Append[pins, 0]]]
```

下面为“大满贯”——12次击球均为全中——得分的计算。

```
In[16]:= BowlingScore[{10,10,10,10,10,10,10,10,10,10,10,10}]
Out[16]= 300
```

你现在还看不懂这个程序——当然是这样，否则我们也就不写此书了！但有一点你应该记住，存在许多有用的计算问题，Mathematica 并不知道如何去解决。所以，为了充分利用 Mathematica 提供的许多种能力，你有时需要进行程序设计。本书的目的就是指示你如何做。

另一目的是教会你基本的程序设计原理。这些原理——与内置函数相结合，再使用条件函数、递归和迭代——可应用于所有其他程序设计语言。（但要记住，具体情况却有很大不同。）

1.2 Mathematica 的使用

这是最首要的。在你做任何有用事情之前，你首先需要知道如何启动 Mathematica 应用程序，如何退出 Mathematica 以及在遇到问题时如何解决它们。这些过程当然都与你正在使用的计算机系统有关。你应该阅读一下与你的 Mathematica 一同提供给你的与系统有关的具体说明；而如果在本书中所提供给你的指示并不适用于你的计算机系统时，你就要请求当地 Mathematica 专家的帮助了。

1.2.1 进入 Mathematica 和退出 Mathematica

Mathematica 有两种基本的用户操作界面类型：“文本”（或“命令行”）界面和“笔记本”（或“图形”）界面。下面的 1.4 节将进行详细的讨论，但因与现在的论述多少有些相关，故知道一下你用的是什么界面，对你是有好处的。在以后阅读了 1.4.1 和 1.4.2，你将在启动 Mathematica 之后立即说出你拥有的是什么界面。一般来说，如果你使用的是 Macintosh 机、运行 Windows 的 PC 机或大多数 Unix 工作站，你拥有的大概是笔记本界面；而如果你使用的是不运行 Windows 的 PC 机或 VAX 工作站，你拥有的大概就是“文本”界面了。

启动 Mathematica

如何在你的计算机上启动 Mathematica，将依赖于你拥有的是笔记本界面，还是需输入命令的命令行界面：

笔记本界面：选取 Mathematica 像标（见 1.4 节），然后双击鼠标。

文本界面：在提示符后输入 **math** 命令。

输入并执行

内容的输入完全照本书给出的那样。为使 Mathematica 计算你已输入的表达式：

笔记本界面时：在录入后，按 Shift-Return 或 Shift-Enter 组合键。

文本界面时：按 Enter 键。

注意，在文本界面时，每一行都将被计算，除非它不是一个完整的表达式（这将在下面几页之后详加说明）。在笔记本界面时，你所输入的每一行在键入 Shift-Enter 组合键之前都不会被计算；一旦键入 Shift-Enter 组合键，这些行就一起被执行。

退出 Mathematica

笔记本界面：从 File 菜单中选择 Quit 选项。

文本界面：在提示符后输入 Quit。

处理遇到的问题

每时每刻，你都有可能输入一些会使 Mathematica 在某些方面操作失常的内容，也许这种失常就是很长一段时间没有任何反应（比如，你不经心地要求 Mathematica 做一些非常困难的事情），也许，屏幕一页一页不停地显示你不会使用的信息。在这些情况下，你可以试着“中断”正在进行的计算。如何做却要依赖于你所使用的计算机的类型。

Macintosh 或 NeXT：按 Command-. 组合键（即 Command 和 . 的组合键），然后按 a.。

运行 Windows 的 PC：按 Alt-.（Alt 和 . 的组合键）。

不运行 Windows 的 PC：按 Ctrl-Break 组合键。

Unix：按 Ctrl-C 组合键，然后按 a，再按 Return 键。

在笔记本界面情况下，这些试图中断计算的尝试有时并不奏效。如果在等待了相当长一断时间后（比如，几分钟之后），Mathematica 看上去仍然阻塞不畅，你就只有“杀死核心”（kill the kernel）^①。这可以通过从 Action 菜单中选择 Quit/Disconnect Kernel 选项而办到。如果是 Mathematica 早于 2.2 版的版本，你就只能采取重新启动 Mathematica 的方法了。从 2.2 版本开始，核心本身可以在不杀死界面的前提下被重新启动，方法是先选择 Action 菜单下的 Kernels and Tasks... 选项，然后在出现的对话框中选择 New Kernel。

1.2.2 寻求帮助

如果你知道一个函数的名称，但却不太确信它可做些什么，那么了解它可做些什么的最容易的方法是输入 ?function 并执行。例如，想要了解 ParametricPlot 函数能做什么，你可以输入

```
In[1]:= ?ParametricPlot
ParametricPlot[{fx, fy}, {t, tmin, tmax}] produces a
```

^① 在试图杀死核心之前，一定要确信计算确实处于死循环之中，而不是确需很长时间才能完成的高强度计算。

parametric plot with x and y coordinates fx and fy generated as a function of t . `ParametricPlot[{{fx, fy}, {gx, gy}, ...}, {t, tmin, tmax}]` plots several parametric curves.

再有, 如果你不能确定某个命令的正确拼法 (比如 `Integrate`), 你可以输入下面的形式以显示所有以 `Int` 开头的内置函数。

```
In[2]:= ?Int*
Integer                InterpolationOrder
IntegerDigits          Interrupt
IntegerQ               Intersection
Integrate              Interval
InterpolatingFunction  IntervalIntersection
InterpolatingPolynomial IntervalMemberQ
Interpolation          IntervalUnion
```

使用笔记本界面的读者可选择 **Action** 菜单中的 **Complete Selection** 选项。这会给出同上面例子中相似的列表。这些读者还可以使用命令完功能与模板功能来寻找你只了解其一部分又想了解其更多情况的函数的信息。

从 2.2 版本开始, **Mathematica** 具有笔记本界面的版本, 在其帮助系统中增加了一个名叫 **Function Browser** (后改为 **Help Browser**) 的很有用的功能。1.6.1 中将详加讨论。

1.2.3 输入内容的句法

当首次启动 **Mathematica** 时, 你会注意到它会为你提供 `In` 和 `Out` 提示符, 而你就不必自己再输入它们了。具有笔记本界面的用户将在输入的内容被执行后才看得到这些提示符, 而具有命令行界面的用户, **Mathematica** 自动提供这样的提示符, 等待你的输入:

```
In[1]:= 39/13
Out[1]= 3
```

你对 **Mathematica** 的输入由**表达式** (expression) 构成, 这些表达式分为数值表达式和非数值表达式。数值表达式的输入形式与通常数学中的一样, 如前面所示。但是, 既然你不能占用多行来输入一个分式, 你就不得不使用斜线号 `/` 代表除号, 如上面我们所做的那样。同样地, 使用符号 `^` 代表取幂:


```
In[2]:= 2^5
```

```
Out[2]= 32
```

Mathematica 允许你把两个表达式并置，以表示这两个表达式的相乘运算，如在数学中那样，但 **Mathematica** 还提供了使用星号 $*$ 来达到同样目的的替代方法，即计算机程序设计中常使用的比较传统的方法。

```
In[3]:= 2 5
```

```
Out[3]= 10
```

```
In[4]:= 2*5
```

```
Out[4]= 10
```

Mathematica 的操作仍按照与数学中相同的运算先后次序(precedence)进行。具体来说，乘法运算和除法运算比加法运算和减法运算有优先权，所以说， $3+4*5$ 等于 23 而不等于 35。

函数的写法与数学书中的写法差不多，不同之处是这里的函数要以大写字母开头，且函数的变元被括在一对方括号中：

```
In[5]:= Factor[x^5 - 1]
```

```
Out[5]= (-1 + x) (1 + x + x2 + x3 + x4)
```

几乎所有的内置函数都以全名方式写出，如上例所示。(例外情况包括一些熟知的缩写形式，如进行微分操作的 **D**，求平方根操作的 **Sqrt**，取对数的操作 **Log** 及求矩阵的行列式的操作 **Det**，等等。)以全名方式写出函数名称这种做法在你并不特别确信一函数就是执行某一特别操作时会变得非常有用。例如，如果我们想要 **Mathematica** 计算一复数的共轭复数，那么进行这一操作的函数应该怎样写，比较合理的猜测应是这样：

```
In[6]:= Conjugate[3 + 4 I]
```

```
Out[6]= 3 - 4 I
```

我们使用一对方括号 $[]$ 把函数的变元括起来，那么由一对大括号 $\{ \}$ 括起来的就表示一个表或值的分布范围；**Mathematica** 拥有非常强大的表操作功能，这要在下面第三章中介绍。表还可以作为变元的补充内容，出现在函数的变元序列之后，比如，在 **Plot** 函数和 **Integrate** 函数中。