中学信息学奥林匹克习题解析

张欣研 李冬梅 编著

北京大学出版社

中学信息学奥林匹克习题解析
张欣研 李冬梅 编著
北京大学

园丁带你进入电脑天地

中学信息学奥林匹克习题解析

张欣研　李冬梅　编著

北京大学出版社
北　京

内 容 提 要

本书以PASCAL语言为基础，系统地讲述了信息学奥林匹克竞赛试题的数学、搜索、动态规划解题方法及编程技巧，并给出了近两年许多信息学奥林匹克竞赛试题的解法。另外，书中还附列了许多珍贵的信息学竞赛资料。在附盘中，可以找到所有的程序和测试数据。

本书是为准备信息学竞赛的同学编写的，是一本不可多得的参考书，同时也会成为对计算机感兴趣的同学的良师益友。

图书在版编目（CIP）数据

中学信息学奥林匹克习题解析/张欣研编．—北京：北京大学出版社，1997
（园丁带你进入电脑天地）
ISBN 7-301-03444-X

Ⅰ．中… Ⅱ．张… Ⅲ．电子计算机-中学-竞赛题-解题 Ⅳ．G634.676

书　　名：中学信息学奥林匹克习题解析
著作责任者：张欣研　李冬梅
责 任 编 辑：郭佑民
标 准 书 号：ISBN 7-301-03444-X/TP·341
出 版 者：北京大学出版社
地　　址：北京市海淀区中关村北京大学校内　100871
电　　话：出版部 62752015　发行部 62559712　编辑部 62752032
排 印 者：北京经纬印刷厂印刷
发 行 者：北京大学出版社
经 销 者：新华书店
787×1092 毫米　16 开本　13.125 印张　330 千字
1997 年 7 月第一版　1997 年 7 月第一次印刷
定　　价：21.00 元

序

现代信息科学技术的迅速发展正在对人类社会产生难以估量的深远影响，信息社会将成为新世纪的一个标志。作为人类总体智慧的结晶，电脑加人脑会大大超过人脑。“计算机的普及要从娃娃抓起”是科教兴国的一项重要内容，是中华民族再度腾飞的准备，也是大批跨世纪拔尖人才脱颖而出的希望。

进入 90 年代，“娃娃要学电脑”已经成为舆论导向，随之而来的问题是：“娃娃怎样才能学好电脑”。我认为老师的指导十分重要，喜闻乐见的启蒙教材不可缺少。

《园丁带你进入电脑天地》是一套专为中学生编写的电脑普及丛书，内容十分丰富，如《中学计算机绘图教程》、《中学 FoxBASE 教程》、《中学电子表格教程》、《中学常用软件使用教程》、《中学 Internet 教程》、《中学 Windows 教程》、《中学计算机文字处理教程》、《中学 Pascal 教程》、《中学信息学奥林匹克习题解析》等都是目前中学生所希望学习和掌握的。从编写者的阵容看，多数是教学第一线的老师，他们了解读者的需求，熟悉青少年的认知规律，对所写内容有比较深入的了解，容易做到深入浅出；丛书是按课外读物的性质编写的，便于学生自学；从指导思想上注意教学法，突出启发性和实践性，强调学习这套丛书既要动脑，又要动手。

书应该是良师益友，给你一本好书，就好比帮你铺好了一条路，在这条路上迅跑就有了基础。但这是有条件的，因为电脑是实践性极强的学科，纸上谈兵，不动手实践是不可能学会的。理论联系实际，动手又动脑是学好电脑的必要条件。从这个意义上说，你要获得关于电脑的真知灼见，就要从自己敲键盘看屏幕做起。如果你光看书不动手，不管书写得多好你会感到越学越乏味，越学越难；反过来，如果你坚持动手，书上写的会使你感到贴心，解渴，学一章有一章的收获，做一道作业增添一分信心。那时，你就会感到越来越离不开电脑。电脑成了你的得心应手的工具，会帮你做许多事情，你的本事自然也就大得多了。这时，你就会深深地感觉到：学电脑入门并不难，深造也是办得到的。这也是编写这套丛书的园丁们所期望的，当然也包括我在内。

中国计算机学会普及委员会主任
国际信息学奥林匹克中国队总教练
清华大学计算机科学与技术系教授
吴文虎

1996 年 7 月 10 日

前　言

您是一个电脑初学者吗？站在琳琅满目的计算机图书前，您一定感受到当今计算机科学日新月异的发展给我们这个社会带来的巨大影响；您一定知道计算机在现代社会各个方面发挥着越来越大的作用。学习使用并掌握计算机简直是太重要了，所以，那么多的人，从小学生到老教师，从国家公务员到公司职员，从刚刚开始学习写作的人到著名的大作家都加入到电脑初学者的行列。然而，怎样才能轻松愉快、顺利地进入电脑天地呢？它一定是您心中常常在想的问题，它也是我们经常在思考和想要解决的问题。

我们是长期工作在中学计算机教学第一线的教师。长期以来，我们一直关注着下面这些问题：近些年来，中小学越来越重视计算机课的教学和教材建设，许多学校为了提高学生的电脑知识水平，开设了不少计算机选修课，但没有合适的教材，这使得中学生渴望进一步学习提高的要求得不到满足；社会上为电脑初学者举办了各种类型的学习班，但缺少从电脑初学者角度出发编写的、受初学者欢迎的入门培训教材和提高培训教材；许多家庭购买了计算机，想买些书自学，计算机方面的书虽然很多，但多数系统性比较强，内容较深，不太适合自学，不易做到边看书，边操作，这种情况增加了自学者的难度。面对这些问题，北京大学出版社组织了北京大学附属中学、人民大学附属中学、北京四中、北京景山学校、北京农业大学附属中学、北京实验中学等重点中学的有经验的中学计算机高级教师进行了研讨并编写了这套《园丁带你进入电脑天地》丛书。

我们的目的是利用多年来积累的丰富的教学经验，利用对电脑初学者学习心理的了解，努力编好这套丛书，使中学电脑爱好者有一套理想的课外阅读书，使需要接受培训的电脑初学者有一套令人满意的培训教材，使电脑初学者有一套适合自学、可操作性强的计算机图书。

本丛书的特点是：

- 从学习者的认知规律出发，安排知识的顺序结构。
- 考虑教学需要，每节容量与一课时相适应。
- 集计算机基础知识与最新技术于一体，内容丰富。
- 突出实际操作，配有具体操作步骤及大量例题，并附有上机练习内容。
- 语言简明、流畅、生动，配有大量插图。

本丛书可作为中学生提高计算机水平的选修课教材或课外读物，也可作为电脑初学者用书或培训教材。因水平所限，难免会有一些不足之处，欢迎大家指正。

主编　李冬梅

1996 年 8 月

目　录

第一章　信息学竞赛的第一步

人类历史经历了漫长的历程，从原始人完全依赖自然（狩猎，耕种）到手工业发展，经过几千年才进入工业时代，进而又在几百年之内跨入电器时代。现在电器时代刚开始不过100年，可是又有一个时代已经悄悄地来到了我们的身边——这就是信息时代。

信息，我们总认为是一种看不见摸不着的东西，可是它确确实实在改变着我们的思想和生活。举例来说吧，现在哪儿看不着信息，商场里各种各样的售货信息，电台电视台里播放的经济信息，股市中时时刻刻传播着的股票信息，等等等等。可以这么说，在现在这样一个时代中，你若失去了信息，你便失去了一切。

信息学，便是随之而产生的学科。大家知道，计算机以它的高速、准确，成为处理信息的最佳设备。但光有硬件是不行的，还得有软件，这两者就像肉体和灵魂，缺一不可。而信息学，便是为计算机制造灵魂服务的。

可是，是否这一切离我们中学生都太远，简直是可望不可即呢？实际上，信息学已经来到了我们的身边。很多大城市的中学已经开设了计算机课，各级的信息学竞赛已经举办了多年，中国的中学生已经多次问鼎了国际信息学竞赛的金牌。这一切，不正说明信息学正在我们身边蓬勃发展吗？

那么，这一切是如何发展起来的呢？

1987年10月，在联合国教科文组织（UNESCO）在巴黎召开的第24届全体会议上，保加利亚的Sendov教授提出了开展国际中学生信息学竞赛的建议，于是1989年便在保加利亚举行了第一届国际信息学奥林匹克竞赛（International Olympiad in Informatics，简称IOI）。在这一届的比赛里，有13个国家参加，其中就有我们中国。这一点就是信息给我们带来的好处；相反，直到1991年底，美国才决定参加IOI，于是急匆匆在1992年组队出征德国波恩，尽管美国队的学生几乎是随便挑选出来的，但也取得了2块金牌的成绩，令人不得不对他们刮目相看。

在中国参加的各项世界中学生竞赛中，只有信息学是从举办的第一届开始参加的。所以在信息学竞赛中，我国几乎每年都成绩斐然（参见表1-1）。

表1-1　历届IOI概况

届次	年份	举办地点	参赛国家数	中国队情况				
				队员数	金	银	铜	名次
1	1989	保加利亚的索非亚	13	3	0	0	3	2
2	1990	白俄罗斯的明斯克	24	4	1	2	1	2
3	1991	希腊的雅典	25	3	2	1	0	1
4	1992	德国的波恩	50	4	3	1	0	1
5	1993	阿根廷的门多萨	46	4	1	1	2	5
6	1994	瑞典的斯德哥尔摩	52	4	3	0	1	2
7	1995	荷兰的阿姆斯特丹	52	5	3	1	1	1
8	1996	匈牙利的布达佩斯	59	4	4	0	0	1

综观历年的比赛，我们发现，除了第5届在阿根廷举办外，几乎都在欧洲举行，这是和欧洲IOI的水平较高有关的。那么，下几届在哪儿举办呢？参见表1-2。

表1-2　下几届IOI举办地点

届次	年份	举办国
9	1997	南非
10	1998	葡萄牙
11	1999	土耳其
12	2000	中国
13	2001	泰国
14	2002	韩国

以后几届IOI，已经分散到几乎世界各地了。尤其在2000年，我国将第一次承办IOI，这将是继中国承办数学、物理、化学奥林匹克之后的又一次盛会。关于中国举办IOI'2000，还有一段小插曲。IOI'1993期间，在阿根廷的门多萨召开的一次会议上，有几个国家被推荐为未来IOI的举办国，美国本来也想争办IOI'2000的，看到中国比自己更有竞争力（别忘了中国是IOI的元老），于是只有悄悄地放弃了这个念头。

中国的确可以说是计算机奥林匹克的元老了。自1984年邓小平同志提出了“计算机的普及要从娃娃做起”之后，计算机普及教育全面拉开帷幕，各级的计算机竞赛也像雨后春笋一样应运而出。1984年8月，中国科协与原教育部委托中国计算机协会举办了首届全国青少年计算机程序设计竞赛，这就是中国信息学奥林匹克（National Olympiad in Informatics，简称NOI）的前身。

NOI到今年已经13岁了，它每年从来自全国近30个省、市、自治区的近100名身经百战的中学生选手中选出十几名幸运者去参加当年的冬令营，进一步选出4名选手代表中国参加IOI。

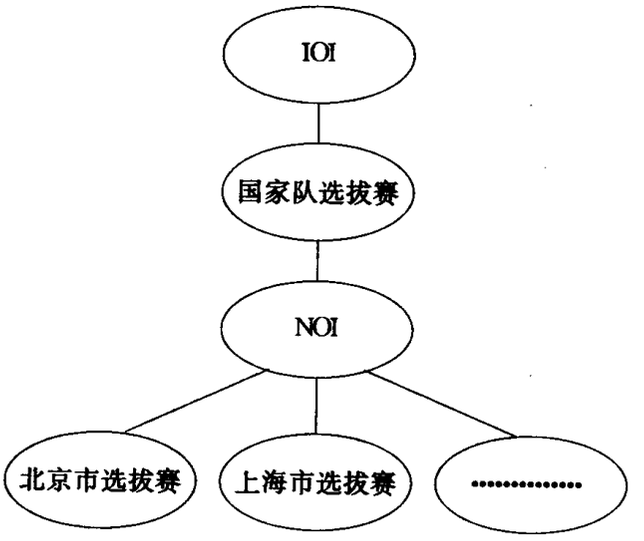

图1.1　中国的“竞赛树”

为什么中国能在IOI中取得那样好的成绩，这恐怕和我们身边很多植根奉献于教育事业的老师分不开，他们默默无闻，从最基层培养好苗子；也和计算机冬令营、集训队的教练分不开，他们一丝不苟，倾其所能培养金牌选手；特别要提到的是中国计算机集训队总教练吴

文虎老师，他以他严谨的作风，开放的视野，创造的精神带领着中国计算机事业未来的栋梁。

这一切，构筑了中国在国际信息学奥林匹克中的光辉战绩。

或许你会想，太难了，太难了，那么多的竞争，那么难的题目，怎么能够跃上最高峰。是的，登上最高峰的人太少了，但我们佩服那些敢于攀登最高峰的人们。

IOI 真的那么高不可攀吗？让我们看一看 1995 年 IOI 中的一道试题吧。

商店购物：

某商店每种商品都有一个价格。例如，一朵花的价格是 2 ICU（ICU 是信息学竞赛的货币单位）；一个花瓶的价格是 5 ICU。为了吸引更多的顾客，商店提供了特殊优惠价。

特殊优惠商品是把一种或几种商品分成一组，并降价销售。例如，3 朵花的价格不是 6 ICU 而是 5 ICU；2 个花瓶加 1 朵花是 10 ICU 而不是 12 ICU。

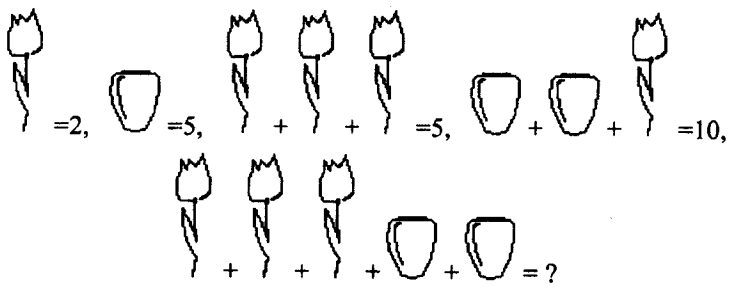

图 1.2　优惠价格销售

编一个程序，计算某个顾客所购商品应付的费用。要充分利用优惠价以使顾客付款最少。

这样的一道题，你能说它高不可攀吗？它几乎完全和我们的生活接上了轨，它来源于生活，却隐藏在生活中不引人注目的角落。当你注意观察的时候，你会发现，生活中到处充满着这样的问题。你是否已经跃跃欲试了。对了，这就是你的第一步。

1997 年 7 月，阔别许久的香港就要回归我们伟大的祖国了，就在它的下一个月——1997 年 8 月，NOI'1997 将坐阵香港，迎接来自全国各地的学生。同学们将会在香港度过一段难忘的时光。

公元 2000 年，第 12 届 IOI 将在中国北京举办，这是中国的一个机会，一个向世界展示自己计算机普及程度和水平的一个大好机会；这也是中国学生的一个机会，一个向世界展示自己的智慧和才华的机会。祝愿此次盛会圆满成功。

各位读者，你们准备从什么时候开始攀登信息学奥林匹克的最高峰呢？

从现在开始！

在攀登之前，希望读者做好这样的准备：去勤奋，去努力，去思考。记住，我们面前最大的困难就是自己，如果超越了自我，成功就会在眼前。

这本书就要带你——尊敬的读者漫游信息学竞赛的天地，在这个美丽的世界中，我们将一同领略数学天地、迷宫以及各种布满美妙风景的地点。

别再等待，加入我们的行列中来吧！

第二章　简单的尝试

“当我们使用苹果机上的BASIC时，国外已经开始使用先进的Pascal语言了。”国家队总教练吴文虎教授曾经这样感叹道。先进的东西就应该学习，本章，大家将会漫游在宽阔的Pascal园地里。也许你还有些迷惑，Pascal语言到底先进在哪儿，为什么几乎所有在信息学奥林匹克竞赛中获得金牌的选手都使用Pascal语言，下面就将解开你心中的疑团。

2.1　学点Pascal

Pascal语言作为国际信息学奥林匹克的首选语言，有其独特的优势。Pascal是一种结构化的语言，在它诞生之初，它的创始人沃思(Niclaus Wirth)就指出Pascal是一种易于进行编程练习的语言。的确如此，现在Pascal语言已成为世界上最常用的计算机语言之一，特别在青少年当中，更成为编程的标准语言。

如果你学过BASIC语言，那么，你会发现Pascal语言写的程序有一种崭新的感觉，一个典型的Pascal语言程序是这样的：

```
PROGRAM LITTLE_PROGRAM(INPUT, OUTPUT);
CONST
    MAX          = 100;
VAR
    MyData       : Array [1..Max] of Word;
    i, j, Min, Po    : Integer;

Begin
    Randomize;
    For i := 1 to Max do
        MyData[i] := Random(MAX);
    For i := 1 to MAX do
    Begin
        Min := Maxint;
        For j := i to MAX do
            if Min > MyData[j] then
            Begin
                Min := MyData[j];
                Po  := j;
            end; { if }
        MyData[Po] := MyData[i];
        MyData[i]  := Min;
    end; { For }
```

```
    Writeln('This is My Data :');
    For i := 1 to Max do
        Write(MyData[i] : 4);
end.  { Program }
```

从视觉上看，Pascal 程序很有层次感。程序的波澜起伏和编程者的思想一步一步深化相互对应。因此 Pascal 程序便于编者自己维护，也便于读程序的人领会。

□ 建议：读懂上述程序。

Pascal 语言有如下两种最基本的语句结构：

选择结构(if then else；Case of)

循环结构(For；Repeat Until；While do)

这是编程中最常用的语句，一定得记熟。

每一种语言都有自己的语法，Pascal 语言的语法是很严格的。在学习 Pascal 初期，一个程序编好运行时总有许多的 Errors(错误)，这是初学者很头疼的问题。例如每条语句后都应有一个分号，熟练了 BASIC 的人就会感觉不太习惯，因此学习 Pascal 的前期总爱出"';' Expected"(缺分号)的错误，解决的方法是多熟悉一段时间，以后就不会这样了。那么，Pascal 语言是不是很呆板呢？其实并不是这样，严格的语法，只是培养一种严谨的风格，使编程更有条理。

如何记住 Pascal 的语句呢？其实，Pascal 作为一种编程语言，为了便于人们记忆，很多的语句在英文中都有其实在的含义(在其他语言，如 BASIC 中也是如此)，例如：

关键字：

if	(如果)
then	(那么)
else	(其他)
Repeat	(重复)
Until	(直到)
While	(当…)
do	(做)
Begin	(开始)
end	(结束)
…	

过程与函数：

Inc	(增加)
Dec	(减少)
Read(ln)	(读)
Write(ln)	(写)
…	

因此，很多语句都可以用英文中的词语加以记忆。另外，Turbo Pascal 6.0 及以上版本提供了详尽的帮助，光标在任一条语句下时，按 Ctrl+F1 键，就可以得到关于此语句的帮助，英文好的读者就可以大受裨益了。

□ 建议：打开计算机，试一试 Pascal 的帮助功能，尝试去读一读 if 语句的帮助(见图 2.1)。

语句过关了，下面的问题就是如何组织起这些语句了。我们以一个简单的例子——求圆面积和周长开始吧。

一般的 Pascal 程序都是由 Program 开头的，这相当于一篇文章的题目，告诉读程序的人这是关于什么的程序。在我们这个例子里，以

```
PROGRAM CIRCLE(INPUT, OUTPUT);
```

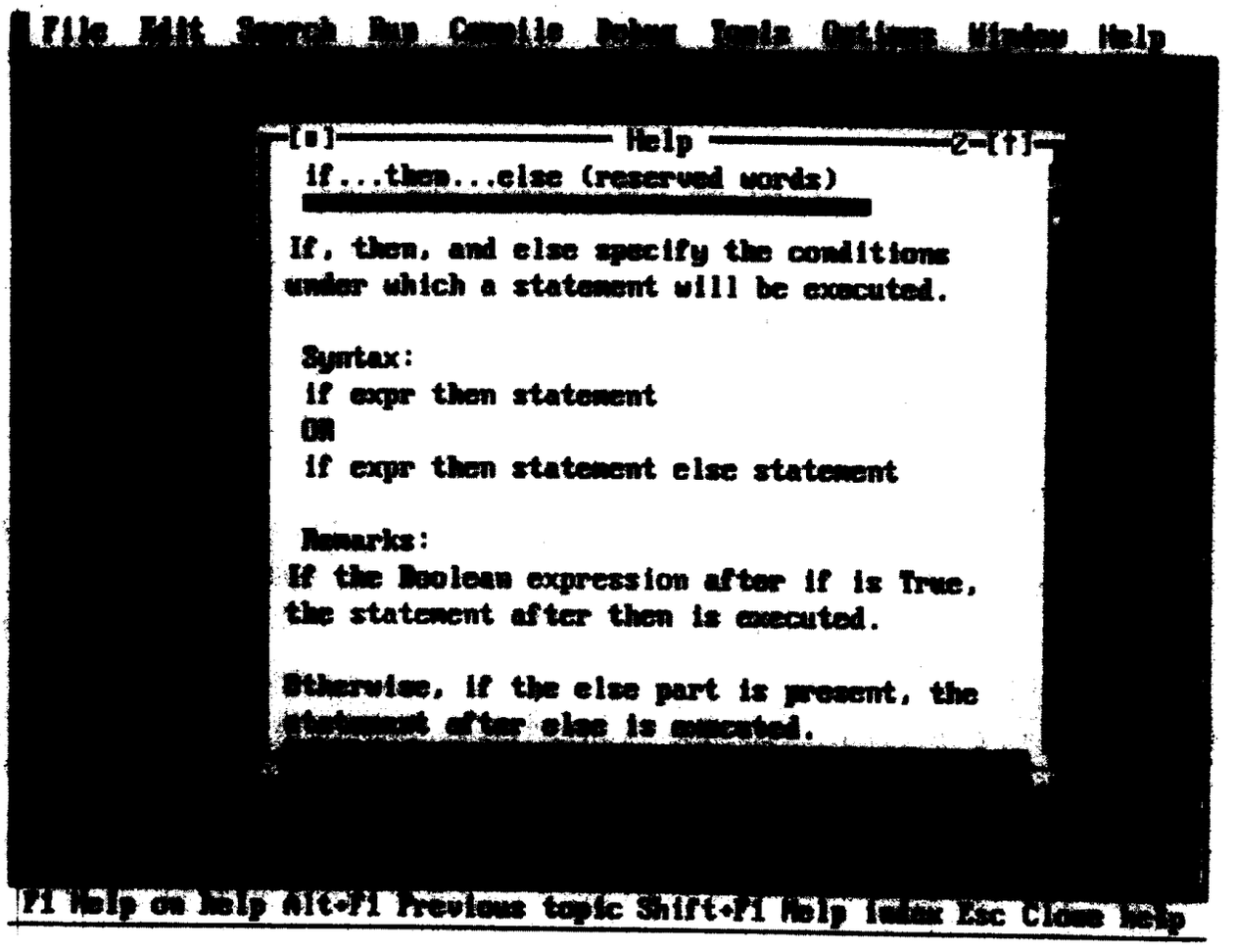

图 2.1　读一读帮助

开头就足够了。其中程序名后面括号中为标准输入 Input 和标准输出 Output，用逗号分隔，这些东西是老的 Pascal 版本留下来的，在我们所用的 Turbo Pascal 7.0 中可以不写。如果觉得开头这样还不详细，可以加上几句注释，如：

```
{ This Program can tell you the
  area and circumference of a circle. }
```

注释是以“{”开始，以“}”结束的一段文字，在程序中加注释可以增加程序的可读性。

```
Var Radius : Real;
```

Var 是 Pascal 的独特的一个词，是 Variable(变量) 的缩写，它是用来定义变量的。它后面可以跟很多形如“变量名列:变量类型”的项，中间用分号(；)分隔。其中变量名列是变量名的序列，变量名用冒号 (：) 分隔。变量类型有整型(Byte，Shortint，Integer，Word，Longint)、实型(Real，Single，Double 等)、字符串类型、指针类型、集合类型、枚举类型、子界类型以及自定义类型(用 TYPE 定义)等。需要注意的是变量名列中的变量应是同一种类型的。Var Radius : Real 说明了 Radius 是一个实型变量。表 2-1 列出各种类型的空间大小和应用范围。

然后正文开始

```
Begin
    Write('The Radius          of the circle is : ');
    Readln(Radius);
```

表 2-1　各种类型概况

类型	种类	应用范围	空间大小(单位:字节)
Byte	整数	0 至 255	1
Shortint	整数	−128 至 127	1
Integer	整数	−32768 至 32767	2
Word	整数	0 至 65535	2
Longint	整数	−2^ 31 至 2^ 31-1	4
Real	实数	2.9e-39 至 1.7e38(有效数字 11 至 12 位)	6
Single	实数*	1.5e-45 至 3.4e38(有效数字 7 至 8 位)	4
Double	实数*	5.0e-324 至 1.7e308(有效数字 15 至 16 位)	8
Extended	实数*	3.4e-4932 至 1.1e4932(有效数字 19 至 20 位)	10
Comp	实数*	−2^ 63 至 2^ 63-1(有效数字 19 至 20 位)**	8
^ Type	指针	任何类型	4
String	字符串	长度小于 256	256***
Set of ...	集合	元素可能总个数小于 256	元素可能总个数/8
(...,...,...)	枚举	枚举元素个数在长整型之内	由所属整型决定
C1..C2	子界	子界范围在长整型之内	由所属整型决定

* 只有当打开 N 编译开关时才能使用(见下一节);

* * Comp 类型只能处理整数;

* * * 字符串后没有给出最长长度时为 256,给出最长长度时为最长长度+1。

```
Writeln('The Area                    of the circle is :',Pi * Radius * Radius);
Writeln('The Circumference           of the circle is :',2 * Pi * Radius);
end.
```

Pascal 语言中每一个“Begin”都和一个“end”相对应,其中最后一个“end”后应该加一个“.”,表示程序结束。

可以开始运行了,按 Ctrl+F9 键,根据提示输入半径,就可以得到结果了。是不是很轻松啊?

程序已经完美了吗?不妨输入半径 −1,得到了什么结果?

原程序仍然能得出结果,而实际上半径小于 0 的圆是不存在的,因此程序需要修改。

□ 建议:修改上述程序,使其知道输入错误(即能判错)。

2.2　熟悉 Pascal

在编写 Pascal 程序的时候,面对着花花绿绿的屏幕,你是否感觉到不知所措;在修改程序的时候,你是否感觉到困难重重;在运行程序的时候,你是否感觉到你的程序总不能给你一个满意的结果。如果你碰到了上述问题,此节将帮你解开心中的疑团。

在上一节中,我们已经接触了 Turbo Pascal 的集成开发环境(IDE),也许你还未感到 IDE 的方便与实用,那么看一看 IDE 最上行的菜单吧,试一试应用其中的一些功能(例如 File 下的 Open, Edit 下的 Find),如果有机会还可以用 IDE 写几篇英文文章,去好好的熟悉它。

如果你准备开始编程,应先使你的编程工具达到最优的状态,请检查如下几条:

● 自己编的程序,最好放在自己的专用源程序目录中,否则东一个西一个,很难保证安

全。这儿,Program, Source 都是很好的目录名。

● Options 菜单下的 Environment 子菜单中的 Editor ...,确保其中的 Auto indent mode 处于[X] 状态,这样方便你程序的层次化,如果不是这样,用方向键使其变亮,再按空格键就可以了,选完后按 Enter 键。

在你编辑的过程中,最好养成经常存盘的习惯,记住存盘按 F2 键就可以了。存盘的好处是,不会由于电源故障或机器死住而懊悔不已。

程序编好了,接着就该运行了。要确保以下几条,以除后顾之忧。

● Options 菜单下的 Compile ... 中有很多选项,用 Tab 键和光标移动键使 Runtime errors 中的[ ]都变成[X],这样可以减轻你调试的负担。

● 还是 Options 菜单下的 Compile ...,有一项 Memory ...,里面可以填三个数字,它们依次分别为 Stack Size(栈内存值),Low heap limit(堆下限),High heap limit(堆上限)。一般都将它们稳定为 65520,0,655360 为最佳。

● Compiler菜单中,Destination 选项为你选择目标程序的存放位置,问你是在内存中编译还是编译成 EXE 文件运行。当然刚开始运行程序时一般应选 Memory,因为这样既快又不会在你的硬盘上造成一堆垃圾,等到需要编成 EXE 文件运行或程序已经成熟时再改成 Disk 也不迟。

● 最后,所有选择都已完成,该把选择的结果存盘了。在 Options 菜单中,选择 Save TURBO. TP,下一次进入的时候,编程调试环境就很好了。

然后就是调试了,这可是件伤脑筋的事。不过面对各种各样的错误,得先有必胜的信心才行,回想一下,程序是自己编的,总不能被自己的程序吓倒吧!

□ 经验:编完一个程序,先通读一遍,就像自己是一位读者,来拜读作家的手稿。这样很容易发现诸如 i 错写成 j 这样的"笔误"。

在编程中,错误一般分为语法错误和逻辑错误。语法错误通常会在按 F9 编译的时候发现,这时屏幕上方会出现醒目的 Errors,光标也会停在出错的地方,因此语法错误很快就会解决。但是一旦程序编译通过了,一句忠告:先按 F2 存盘,再运行吧!因为你现在面临的可能是导致机器崩溃的逻辑错误。

这并不是危言耸听,那些编译通过的程序,有时怎么修改也仍然和预期的结果不一致,这是因为有些逻辑错误太隐蔽了,一个字母敲错,一个符号不对,以至于语句的顺序弄颠倒,都有可能导致逻辑错误。这些无处不在的逻辑错误,使编程的人伤透了脑筋,有时甚至调试程序的时间比编程序的时间还要长。

逻辑错误既然那么可怕,有没有办法发现这些逻辑错误呢? Turbo Pascal 系统给我们提供了良好的调试环境,可以 Trace(一步一步执行)、Go to cursor(执行到光标处)、Breakpoint(可以让程序在任何地方停下来)、Watch(可观察几乎所有量的值)等等。下面我们就来看一看运用它们调试程序的威力。

● Trace

在编完程序后,我们通常按 Ctrl+F9 去运行我们的程序。然而,对于编程新手来说,并不是所有的程序都能一次通过,因此我们需要用 Trace(一步一步执行)调试我们的程序。

Trace 分为两种,一种叫 Step Over(F8),另一种叫 Trace into(F7),这两种有什么区别呢?

Step Over 运行程序的时候，过程和函数都是直接跳过的，它不会运行到过程和函数内部，这样的好处是节省了我们调试的时间，尤其在我们已经知道过程和函数已经正确的情况下，我们往往选择它。

Trace into 就和前一种不同了，它才是真正的 Trace，几乎所有的过程和函数（除了 Turbo Pascal 标准的过程和函数或者没有源代码的 Unit 中的过程和函数），它都一丝不苟地一步一步运行。这种功能，在我们刚开始调试时很有用处。

如图 2.2 就显示了这两个功能的区别。

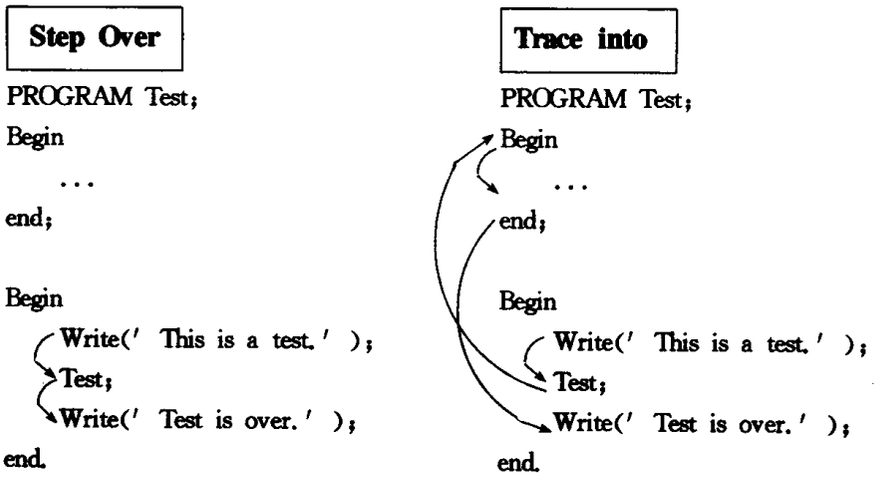

图 2.2 Step Over和Trace into的区别

Trace 的另一个功能，还可以使我们对程序的走向了解清楚。所以在看别人编的程序时，应用这一功能，可以起到事半功倍的效果。

当然，Trace 也不是万能的。例如在调试递归程序时，用 Step Over 经常不能完整地运行一个过程或函数（它总是在递归的不同层次中乱跳），这主要是由于它功能的局限。不过总而言之，如果能适当地运用 Trace，一定会对我们的调试提供极大方便的。

● Go to cursor

运行程序时，在某一处发现了 Runtime Error（运行错误），就可以运用这个功能了。它可以让程序运行到光标处停下来，这样我们就可以用其他的调试功能发现错误所在了。

这种调试的用法是，把光标停在我们要程序运行到的地方，按 F4 就行了。它的好处就是快速便捷。

● Breakpoint

Go to cursor 有一个局限，就是它同时只能让程序在一个地方停下来，于是 Breakpoint 就来帮忙了。Breakpoint 可以同时定义许多暂停点，无论程序运行到哪个暂停点都会停下来，停下之后我们就可以进行各项调试了。

它的用法颇为简单，只要用 Ctrl+F8 将某一处的背景变成红色，这一点就成为暂停点了。以后按 Ctrl+F9 运行到这儿就会停在这儿了，直到你再一次按 Ctrl+F8 将暂停点撤销为止。

● Watch

在调试程序的过程中，如果能知道某些变量及表达式的值，对我们肯定是很有帮助的。Watch 就担当了这个角色。

按 Ctrl+F7 可以弹出 Add Watch 对话框（如图 2.3），我们键入一个变量名或表达式后，

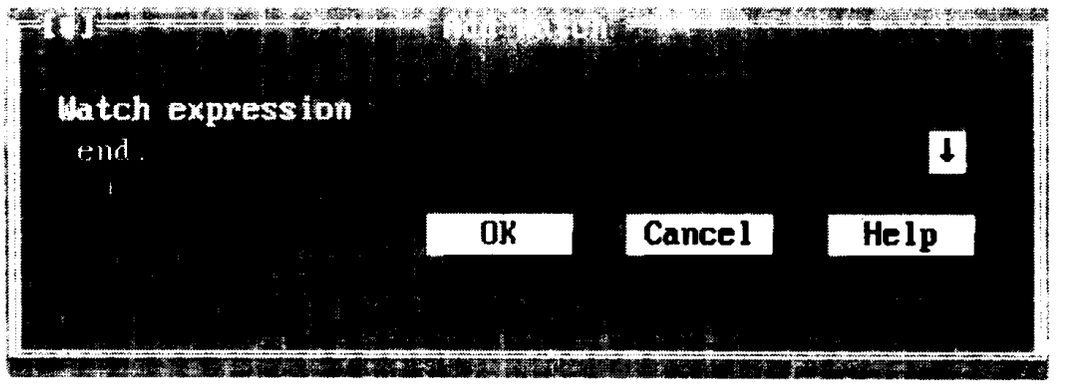

图 2.3　Add Watch 的对话框

它就可以让我们时时监视我们要观察的内容。在 Watches 窗口中我们可以翻看我们要观察的东西，可以左右移动来看很长的数组的项；按 Enter 键可以编辑观察的内容，Insert 键可以添加观察的内容，Delete 键可以删去不再想观察的内容。Watch 的特点就是方便及实用。

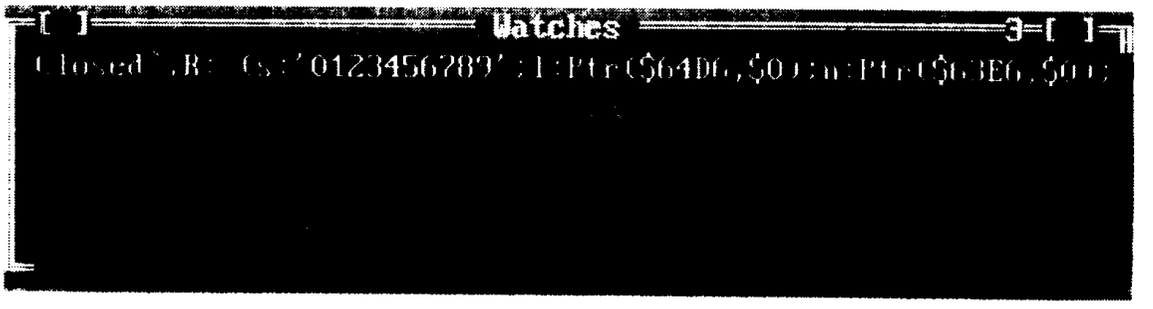

图 2.4　Watches 窗口

在如同图 2.4 的 Watches 窗口中，我们还可以用一些格式来进行某些特殊的观察。例如有一个数组 D[10,10]，如果想观察从 D[1,1]到 D[1,10]这 10 项，就可以在 Add Watch 中输入 D[1],10 来达到这个目的。后面加上的",10"是指要观察接下来的 10 项。这样的格式还有很多，列出如下：

C	用字符显示
D	用十进制数字显示
Fn	实数用前 n 位数字显示
Hx, H, X, $	用十六进制数字显示
M	显示内存映像
P	用 Segment : offset 显示指针
R	显示结构时同时显示结构中的域名

我们给出一个例子来看一看各种格式的用法。

```
Type
    NamePtr = ^ NameRec;
    NameRec = record
      Next  : NamePtr;
      Count : Integer;
      Name  : string[31];
```

```
    end;
Var
    List : array[1..10] of Integer;
    P    : NamePtr;
```

我们在 Watches 中将得到这些：

List :	(10,20,30,40,50,60,70,80,90,100)
List[6], 3X :	$3C, $46, $50
P :	PTR($3EA0, $C)
P, P :	3EA0:000C
P^ :	(PTR($3EF2, $2),412,'John')
P^.Next^, RX :	(NEXT:NIL;COUNT:$19C;NAME:'Joe')
Mem[$40:0], 8M :	F8 03 F8 02 00 00 00 00
Mem[$40:0], 8MD :	248 3 248 2 0 0 0 0

这么多格式，并不一定都要记住，记下几个常用的就可以了。

此外，Turbo Pascal 系统还有很多的调试功能，例如 Evaluate and Modify（计算和修改），Call Stack（调用的栈）等等，这些都是很有用的。将它们协调地利用起来，一定会提高你的程序调试技巧的。

调试结束了，运行通过了，是否编程就结束了呢？其实还没完。

你不想使程序更快速点吗？编译开关可以帮你这个忙。你可能对编译开关还不太了解，这儿先介绍一下。编译开关是一种通知 Pascal 系统如何编译程序的工具，例如，Pascal 很强大的范围检查，在调试时给我们很大的帮助，但调试过后，它就没有什么用了，因此我们可以关闭它，使程序的运行速度加快。事实证明，适当地调整编译开关，可以使程序的运行速度提高 20%至 30%左右。

在 Options 菜单的 Compile ... 里可以很直观地调整编译开关，那么，如何在程序中调整呢？我们可以使用 $ 指令。$ 指令的格式是这样的：

{$Switch1+(或-),Switch2+(或-), ...}

其中 Switch 是编译开关指令，之后的+代表打开，-代表关闭，开关之间用逗号“,”分隔（注意不要有空格），只有打开编译开关才能使它起作用。

常用的编译开关指令有这样几种：

D：调试消息开关；

E：软件实数运算开关；

I：输入输出检查；

L：变量等信息开关；

N：扩展实数运算开关；

Q：运算溢出检查；

R：范围检查；

S：栈空间检查；

G：286 代码开关；

M：内存分配开关。

例如，如果我们调试结束，想使程序以快一点的速度运行，就可以将编译开关 R,S 等关