青少年国际信息学（计算机）奥林匹克竞赛指导

——人工智能搜索与程序设计

●刘福生　王建德　编著

電子工業出版社

青少年国际信息学(计算机)奥林匹克竞赛指导

——人工智能搜索与程序设计

刘福生　王建德　编著

(京)新登字 055 号

内 容 提 要

本书采用循序渐进、逐步深入的方法，系统地介绍了人工智能搜索与程序设计的基础知识与编程技巧。全书分为程序设计知识、盲目搜索、启发性搜索和解题方法小结等四篇共十七章。每章都是从趣味解题入手，分析解题思路，归纳解题框架，而后给出一题多解。书中所选例题大多是国家队用来训练选手的题目和国际奥林匹克竞赛试题。

本书既是一本适合自学和授课的教材，也是一本历届奥林匹克题题解；既是辅导参赛选手的良好读物，又不失为中学计算机教师的参考手册，而且还可作为高等学校程序设计教学的辅助教材。

青少年国际信息学(计算机)奥林匹克竞赛指导
——人工智能搜索与程序设计

刘福生　王建德　编著

责任编辑：宋玉升　徐　轲

电子工业出版社出版（北京市万寿路）
电子工业出版社发行　各地新华书店经销
北京天利电子出版技术公司排版
北京市顺义县东辛山印刷厂印刷

开本：787×1092 毫米 1/16　印张：23.75　字数：520 千字
1993 年 4 月第 1 版　1993 年 4 月第 1 次印刷
印数：4000 册　　定价：12.80 元
ISBN7－5053－1868－3/TP・445

前　言

信息学(程序设计)竞赛，愈来愈受到各地区、各学校师生的青睐。特别是近几年，我国组队参加国际奥林匹克信息学竞赛并取得优异成绩以来，更加激励了全国青少年计算机爱好者、中学计算机教育者、青少年科技指导员以及广大关心青少年的人们，使他们对信息学的普及产生了浓厚的兴趣。目前，国际国内竞赛正日益进入人工智能搜索领域，各地师生迫切希望获得这方面的知识。我们本着"以提高读者科技素质为本，以增长学员参赛能力为衡"的原则，编纂了这本书。在两年的实践中，我们曾根据书中的内容，通过深入浅出的讲解，教以解题的程序步骤。一般中学生不仅能理解这些基础知识、掌握种种解题技巧，而且兴趣盎然。参加学习的同学，在知识和技能上都有所提高，不少人在国际、国内竞赛中，取得了优异成绩。第三、第四届国际奥林匹克信息学竞赛金牌得主杨云和同学就是学员中的佼佼者。

本书采用循序渐进、逐步深入的方法编写。从总体来讲是有步骤、有系统、由浅入深、由细到精地介绍《程序设计与人工智能搜索》的基础知识和解题技巧，但是每一章又都是从趣味解题入手，分析解题思路，归纳出解题框架，最后再运用这些框架去解决历届奥赛试题。本书试图努力提高读者的实践能力，摒弃泛泛讲授理论的做法，采取围绕编程实践来介绍各种概念和技巧。书中经常采用一题多解、反复比较的教法，尽量增长读者的见识，以使其能触类旁通、举一反三，藉以锻炼读者的应变能力，今后遇到任何试题时，不致于惊慌失措、一筹莫展。

书末附有奥林匹克竞赛试题精选，所以本书既是一本适合自学和授课的教材，也是一本历届奥赛试题题解；既是辅导参赛选手的良好读物，又不失为中学计算机教师的参考手册；高等学校也可用作 PASCAL 语言和程序设计教学的辅助读物。

我们怀有良好的愿望，但由于水平和经验不足，尽管已三易其稿，但疏漏还在所难免，尚祈学术前辈和广大读者不吝赐教。

本书在编写过程中，曾得到历届中国奥赛代表团总教练兼领队吴文虎教授的大力支持，审阅了全部书稿，提出了极其宝贵的修改建议，并为本书作序。我们特此向吴教授致以衷心的感谢。

国际奥林匹克信息学竞赛第三届、第四届金牌获得者、第二届铜牌获得者杨云和，历届各级竞赛优胜者杨天明、吴以群等同学，以及上海杨浦少科站章弘老师，上海师范专科学校何炜老师都为本书的编写提供了支持和帮助，谨向他们表示深深的谢意。

编著者

一九九二年九月于上海

序　言

1987年保加利亚Sendov教授在UNESCO(联合国教科文组织)第24届全体会议上提出倡议:举办国际信息学奥林匹克,定名为International Olympiadin Informatics,简称IOI。1989年5月在保加利亚首都索菲亚举办了首届IOI,有13个国家的16个队46名青少年选手参赛,中国队的三名选手获三块铜牌,团体总分第二。1990年7月在苏联的明斯克举办第二届IOI,有26个国家的104名选手参赛,中国队的四名选手获一块金牌、二块银牌、一块铜牌,团体总分第二。1991年5月在希腊首都雅典举办第三届IOI,有23个国家的76名选手参赛,中国队的三名选手获二块金牌、一块银牌,总分跃居第一。

举办国际信息学奥林匹克的目的是,通过竞赛形式对有才华的青少年起到激励作用,促其能力得以发展;让青少年彼此建立联系,推动知识与经验的交流,促进合作与理解;宣传信息学这一新兴学科,给学校这一类课程增加动力与新的思路;建立教育工作者与专家之间的国际联系,推动学术思想交流。

信息学是一门新兴的学科,研究的内容涉及信息的来源、产生、获取、识别、转换、组织、存储、处理、检索、表达、评价等,还要研究与信息有关的理论。计算机科学属于信息学的组成部分,从1946年第一台电子计算机问世到今天,经历了40多年的发展,使人类从生产到生活发生了巨大的变化。以信息技术为主导将改变整个社会的经济活动方式,信息科学将在未来人才的知识结构中占据重要地位。目前世界强国都异常重视信息科学的普及教育,计算机与基础教育的结合已成为世界的趋势。IOI是这股潮流的一个标志。

从竞赛内容和形式来看,目前IOI属于编程竞赛,比试智力与利用计算机解题的能力。对于选手,除了要求基础知识学得扎实,要求掌握计算机程序设计语言、数据结构和有关算法外,并且还要求具有较强的上机编程能力。

竞赛是广大青少年喜闻乐见的一种活动形式。许多人都想在各种级别的赛场上一试身手,也都渴望有一个学习与提高的机会,盼望能有一本书来指导他们的学习活动。今天,由刘福生、王建德两位老师合著的辅导材料《青少年国际信息学奥林匹克竞赛指导——人工智能搜索与程序设计》和广大读者见面了。书中详细地介绍了人工智能搜索算法,这是解题中的一种非常有用的算法,有普遍意义。书中所选的例题大多是国家队用来训练选手的题目,或是对国际大赛试题的剖析,有相当的深度和难度。要学好人工智能搜索算法,需要具备先修的知识,譬如有关的程序设计语言(Pascal)、初步的数据结构与算法知识。对于这些知识,书中都作了系统、全面的介绍。编写本书的老师曾亲自培训过上海的选手,有丰富的经验。因此,这本辅导材料能够深入浅出,抓住要点,在思路与方法上给人以启迪,很有参考价值。

信息学竞赛的目的是为了在全国青少年中推动计算机的普及,使学生开阔眼界,扩大知识面,了解计算机在现代化社会中的战略地位,培养逻辑思维、创造性思维以及应用计算机解决实际问题的能力。普及是有层次的,普及与提高是相辅相成的。参加国际信息学奥林匹克和参加全国信息学奥林匹克属于较高层次。在这种高层次的普及活动中,因材施教的原则更为重要。这本辅导教材完全符合因材施教的原则。

信息学竞赛是智力与计算机解题能力的竞赛。中国的青少年业已在世界大赛中显露出

自己的才华。他们没有辜负党和人民的期望。社会主义祖国重视青少年的全面发展与科学素养的提高。中华民族有志气、有能力自立于世界民族之林。作为教师，我们有责任自觉地将普及现代科学技术知识与中华民族光辉灿烂的未来联系起来。最后这番话发自肺腑，愿与刘福生、王建德老师共勉，也希望广大青少年读者能够理解我们。

国际奥林匹克中国队总教练兼领队

吴文虎

1992年9月于清华园

目　录

第一篇　程序设计知识

第二篇　盲目搜索

第三篇　启发性搜索

第四篇　解题方法小结

第一篇　程序设计知识

第一章　PASCAL 语言

§1—1　一个简单例题

在详细介绍 Pascal 语言之前，首先让我们来看一个用 Pascal 语言编写例题：

例 1-1　在输入一个给定的整数 n 后，打印出所有不超过 n 的、其平方为回文的数。

所谓“回文”，指的是字符串两端的字符左右对称。例如 1，22，121，4224 等均是回文。

思路：输入一个整数 n，分别把 1…n 所有整数的平方值赋给变量 s，并按如下方法对 s 进行是否为回文的判别：

将 s 的各位数依个位、十位、百位…的顺序送入数组 d[1]，d[2]，d[3]…，然后设置两个指针 i，j，分别从左另两端开始逐个判别，若 d 两端的字符左右对称，则对应的 s 为回文数。按上述思路，直接写出程序如下：

```
program palindrome;                                          {程序首部}
const max=20;                                                {常量说明}
var n,m,i,j,s : longint; d : array[1..max]of integer;       {变量说明}
begin                                                        {主程序}
  read(n);
  for m := 1 to n do                                         {循环结构}
  begin
    s := m * m; j := 0;
    while s<>0 do                    {程序模块，将整数平方 s 的各位数值记入 d 数组}
      begin
        j := j+1; d[j] := s mod 10; s := s div 10;
      end;{while}
    i := 1;
    while (d[i]=d[j]) and (i<j)do    {程序模块，依次比较 d 两端各位数相等否}
      begin
        i := i+1; j := j-1;
      end;      {while}
    if i>=j then writeln (m,' ',m * m)
  end;{FOR}
end. {MAIN}
```

如果在计算机上运行这个程序，输入数据 100，程序将打印出

1　　1
2　　4
3　　9
11　121
22　484
26　676

共显示 6 行信息。每行的左侧是一个不超过 100 的数，右侧为对应的平方数。显然，这个平方数是回文数。

考虑到有些读者，可能对 PASCAL 语言还不太熟悉，现对 PASCAL 程序稍加说明。

概括一点说，程序由三大块组成：

① 程序首部
② 说明部分（又称数据节）
③ 语句部分（也叫编码节）

程序首部——程序的第一行就是首部，任何 PASCAL 程序的首部均以“program”开始，接着书写程序名。程序名由编程者选定。建议选择的程序名能反映程序的功能和有助于记忆。本例的程序名为“palindrome”（回文）。在某些版本中（如：Turbo PASCAL）首部还包含“编译指令”，有关编译指令的内容，有兴趣的读者可参阅有关资料。

说明部分——PASCAL 对程序中所涉及的对象都事先加以说明。例如，对变量要说明其数据类型，而在以后的程序中，对变量赋值时，表达式的值必须与它的数据类型相容，以帮助你避免在程序设计中可能发生的错误。在“§1—2　PASCAL 的程序结构”一节中我们将较详细地介绍说明部分。本例题第二行至第四行为说明部分，它包含以下两个内容：

常量说明　CONST max＝20；

其中 CONST 是保留字“常量”，max 是设计者给该常量起的名字，等号左边的数字 20 是常量的限定值，意思是定义一个常量 max＝20。

变量说明　Var n,m,i,j,s：longint；　d：array [1..max] of integer；

其中 Var 是“变量”保留字，n,m,i,j,s 是被定义的五个变量名，longint（长整数）是说明被定义的变量是长整数类型或简称长整型；说明中的另一个变量 d，则被定为整数（integer）数组（array）类型变量，其数组下标自 1 到 max，max 就是方才所定义的常量。

语句序列——PASCAL 的主程序部分，由一系列可执行程序语句组成，用保留字 begin 和 end. 加以界定。本例中读者清楚的看到说明部分后的 begin……end. {main}之间就是主程序（其中‘.’表示正文结束，而{main}是注释，此处标出，仅为了说明这个 end 对应主程序尾，以下情况亦同）。PASCAL 可以将对应一个复杂计算的一连串语句顺序连续，看成一条语句的形式，这种语句的形式称为复合语句，复合语句也用 begin……end 界定。例如本例中从 for 语句开始的 begin 到 end；{for}之间是一循环结构程序，此循环结构中又包含两个小的 while 程序模块，分别完成将整数平方值的各位数值记入 d 数组，以及通过比较 d 数组对称两端各位数值是否相等，来判别该平方值是否为回文。

程序的书写采用阶梯式，不同层次的程序模块语句的起写点，置于不同的列上，层次愈

低的模块语句起写点愈向右挪。各级程序错落有序，有利阅读理解，也便于分段分层调试。

为了便利读者阅读该程序，下面先将有关语句作一介绍：

1、输入语句 read 和 readln

功能：从键盘读入数据项，并把它存到变量中去，使该数据能在以后的计算中使用。

例如程序中的 read(n)表示从键盘读入某长整数赋给 n。

输入语句的一般形式为：

read(v1,v2,…,vn);

readln(v1,v2,…,vn);

就是从键盘逐次读入数据，分别赋给变量 v1,v2,…,vn，它们的数据类型可以是整数型、实数型、字符型或字符串型，其类型在程序中由变量说明来定义。一次输入多个变量值时，read 语句要求数据之间用空格分隔，readln 语句要求用回车隔开。本例中各变量已定义为长整型。

2、输出语句 write 和 writeln

功能：把程序计算的结果，按适当的形式输出到屏幕。

例如程序中的 writeln(m,' ',m * m)，表示向屏幕输出变量 m 和 m * m 的值，两值之间用一个空格分隔开。

输出语句的一般形式为：

write(item1,item2,…,itemn);

writeln(item1,item2,…,itemn);

其中 itemi(1≤i≤n)表示输出项，它有两种形式：

①表达式：首先对表达式求值，然后输出值。表达式限定为整型、实型、字符型和布尔型。例如程序中分别输出表达式 m 和 m * m 的值。

②字符串：即前后用单引号(')括起来的任意字符组成的串。例如程序中输出由一个空格字符组成的字串' '，该字串位于表达式 m 和 m * m 之间。

输出时，所有项按列出顺序打印在一行上。若使用 write 语句，则光标置于同一行末项后；若使用 writeln 语句，则光标移到下一行。本程序的输出就是采用了换行方式。

为方便列表，用户可为每个输出项指定一个域宽。指定方法是输出项后跟上一个':'和一个指定域宽的整型表达式，即形式为"项：域宽"，如语句：

write('A=',a:3,',A+B',a+b:5);

若程序中 a 已赋值 5，b 已赋值 20，则输出字样为：

A=5,A+B=25

↑光标位置

3、注释语句

在程序中插入解释，提醒自己或别人注意某变量代表什么，某个语句或函数做些什么，这些解释叫注释。PASCAL 程序可插入任意多的注释。

注释语句由左花括号'{'开始，右花括号'}'结束。编译程序忽略花括号内的正文。

回过头来，让我们再来看上述程序。如果借助于右方的注释和语句本身的英文含义，我们就会感觉到 PASCAL 编写的程序，结构十分清晰。程序的执行顺序与程序正文的语句顺

序一致，其中每一个局部程序模块都反映了一个相对独立的性质，语句之间联系简单、自然，非常便于理解。

PASCAL 语言是 1971 年由 N. Wirth 专为教授结构化程序设计技巧而设计的语言。它具有如下特点：

①PASCAL 语言编写的程序能全面、清晰地体现结构化思想，方便阅读。

②PASCAL 语言，不仅提供了丰富的数据类型，而且还具有灵活的数据结构构造能力。

③PASCAL 语言不仅提供了过程和函数，而且还可定义局部变量，在调用者与被调用者之间可以传递参数。通过可以定义局部变量和传递参数的函数和过程，真正实现了模块独立性和程序模块化的要求。

④PASCAL 语言引入了可描述动态数据结构的指针类型，给程序提供了构造链表、树、图等递归数据结构的手段（有关数据结构和算法的知识将在第二、三章中介绍）。

由于 Pascal 语言精确地表达了程序设计的基本概念，具有小巧、简洁、连贯、精致、结构性好、表达能力强、实现功效高、调试方便、移植容易等优点，因此成为国际上最受欢迎、最广泛流行的程序设计语言之一。国际、国内青少年奥林匹克信息学竞赛上，使用该语言编程的人数为多。不少地方已将 PASCAL 语言列为高中的选修课。为了顺应这种趋势，使读者有更多的得益，本书将以 Turbo PASCAL 作为主线，渗透于每章节的讨论之中。

下面，我们将着重讨论 PASCAL 的程序结构、控制结构、数据类型和子程序。限于篇幅，讨论只能有选择、有重点地展开。读者若在学习中遇到本书未涉及的语言细节问题，请参阅有关 Turbo PASCAL 的技术资料。

§1－2　PASCAL 的程序结构

熟知 BASIC 的读者知道，BASIC 程序直接由若干程序行组成，对程序中的常量、行号及各种数据类型的变量或子程序不作预先定义，基本语句也不含其它语句成分，程序的结构性很差。而 PASCAL 语言按照结构程序设计的要求，严格规定了程序的书写格式，是一种良好的多节有序语言。其组成形式如下：

PROGRAM　程序名；
行号说明；
常量说明；
类型说明；　　— 说明部分
变量说明；
过程与函数说明；　　—分程序
begin
语句序列；　　— 语句部分
end.

其中

1、所有 PASCAL 程序都以 PROGRAM 作为第一个字。

2、说明部分的各个说明都以各自的保留字开始，以分号‘；’结束。

3、过程说明和函数说明都由各自的过程首部或函数首部以及随后的分程序构成，用于

定义函数和过程，其结构和主程序的结构类似(见§1－5　PASCAL 的子程序)。

4、分程序的语句部分是该分程序(它可以属于主程序、过程或函数说明)的执行部分。它以保留字 begin 开始，后面是一组语句，相邻语句之间以分号';'分隔，最后是保留字 end，实际上是一个复合语句。整个程序以圆点'.'结束，写在最后一个 end 之后。

5、由(3)、(4)可知，由于过程说明和函数说明都含分程序，分程序又可以有过程说明和函数说明，因此 PASCAL 程序是嵌套结构。

6、书写程序时，应按每个语句所属的层次阶梯式书写。相同的层次在相同列开始，使程序看起来层次分明。

一、说 明 部 分

用来对常量、行号、各种类型的变量以及函数和过程进行定义的语句，称为说明性语句。

(一)行号说明语句

BASIC 程序的每行语句，都有一个行号，而 PASCAL 程序中，行号并非必要，可以完全没有行号，也可以为满足灵活编程的需要而使用少量行号。之所以在程序中标定行号，仅为了与 GOTO 语句配合使用，以指向 GOTO 的去处。需要标定行号时，应该对行号进行定义，其格式如下：

Label　行号 1[,行号 2,…,行号 n];　　[]为任选项

其中行号由 0 到 999 的数字序列或者标识符组成。

例如

```
program max;
label 1,2;
……
begin
  ……
  if a>b then GOTO 1;
  max := b;
  GOTO 2;
1:max := a;
2:writeln(max);
  ……
end.
```

不恰当地使用 GOTO 语句，会破坏程序的结构化，从而导致混乱。为了有效地防止滥用 GOTO 语句，标号将被限制在同一过程、函数或结构语句的范围内。

(二)常量说明语句

给程序中使用的每一个常量命名，使用命名的常量可以增强程序的可读性和可维护性。

常量说明语句有两种形式：

1、无类型常量的说明

定义形式：

```
const  常量名=无类型常量;
```

无类型常量是真正的常量，Turbo Pascal 不允许改变其值。例如

```
const s='text';
  begin
    s:='AAAA'
  end.
```

因为程序在运行过程中要改变无类型常量 s 的值，所以程序出错。

2、有类型常量的说明

定义形式：

```
const  常量名：类型=类型常量;
```

对于有类型常量，可以重新被赋值。例如

```
const s:string[4]='text';
begin
  s:='AAAA'
end.
```

s 是一个长度为 4 的字串类型常量，初值为 'text'。由于它有类型，程序运行过程中可将其重新赋值为 'AAAA'。

有类型常量的说明要比无类型常量的说明复杂一些。各种类型常量，由于其类型不同，常量的性质、类型常量说明语句的书写格式也随之不同。我们将在"§1--4 PASCAL 的数据类型、类型常量、运算符、表达式"中详细介绍。

(三)类型说明

在 PASCAL 语言中，用户可以根据需要自己定义合适的数据类型。类型说明语句对每一个用户自定义的数据类型给予说明。

定义形式：

```
type  类型名=数据类型;
```

例如

```
type
  index=1..100;
  number=(one,two,three,four);
```

类型 index 是用户自定义的子界类型，该类型的变量可取从 1 到 100 区间内的任一个整数值。

类型 number 是用户的定义的枚举类型，该类型的变量仅可取 one，tow，three，four 四者之一。

(四)变量说明

对程序中的每个变量定义一个和它相联系的类型。它可以是系统提供的标准类型，也可以是用户通过 type 说明语句自己构造的类型，并且还可以规定类型变量的内存地址。变量说明语句的定义形式有两种。

1、一般变量说明

```
var  变量名表：类型;
```

例如

```
var
  a : integer;
  b : byte;
  c : char;
  d : boolean;
  e : number;
  f : index;
```

说明变量 a 是整数、变量 b 是字节型，分别取带符号位的 16 位整数和无符号位的 8 位整数。变量 c 是字符型，其值域是机器允许的字符集，变量 d 是布尔型，仅有 false 和 true 两个值，这些数据类型都是 PASCAL 的标准类型。而变量 e 是用户通过 type 自定义的枚举类型 number，变量 f 是用户自定义的子界类型 index(number 和 index 的类型定义见(三)类型说明)。

上述几种数据类型称为序数类型。序数类型的每一个值对应一个序数。约定整型值、字节型值的序数为其本身；字符型值的序数为字符的 ASCII 码值；布尔型值 false 的序数为 0，true 为 1；枚举类型值为定义它时的枚举序号，约定第 1 个枚举值为 0，其余依次类推。子界类型的序数是该值在原类型的序数。

对于序数类型，可定义以下标准函数。

ord(x)—值 x 的序数；

pred(x)—比值 x 的序数少 1 的那个值。当 x 的序数已是最小时，函数值无定义；

succ(x)—比值 x 的序数大 1 的那个值。当 x 的序数已是最大时，函数值无定义。

例如设变量 a 的值为 99；变量 b 的值为 1；变量 c 的值为 'A'；变量 d 的值为 false；变量 e 的值为 one；变量 f 的值为 100。则

```
ord(a)=99,    ord(c)=65,    ord(d)=0;
pred(b)=0,    pred(f)=99,   pred(a)=98;
succ(c)='B',  succ(e)=two,  succ(f)=101;
```

为了后面叙述的方便，我们先简单例举几种数据类型。有关 PASCAL 数据类型的详细讨论，将集中在"§1—4　PASCAL 的数据类型、类型常量、运算符和表达式"一节中展开。

2、绝对变量说明

```
var  变量名表 : 类型 absolte 子句;
```

通过 absolute 子句说明该类型变量是占内存中一个特定地址的绝对变量。absolute 子句有二种形式：

① absolute　段地址值 : 段内偏移量

限定变量所在的段地址和偏移量，两整数在范围 $0000～$FFFF(0～65535)之间。

例如：

```
var
  crtmode : byte absolute $0040 : $0049;
```

说明 crtmode 是字节类型，该变量所在的段址为 $0040，偏移量为 $0049。

② absolute　变量名 2

说明变量名表中的变量与变量 2 占据同样的内存地址。

例如

```
var
  str : string[32];
    strlen : byte absolute str;
```

第一个变量说明定义变量 str 为长度 32 的字串类型；第二个变量说明限定两变量 strlen 和 str 有同一开始地址。因为字串变量 str 的第一个字节包含字串的动态长度，所以变量 strlen 将包含 str 的实际长度。

程序中的每一个变量，根据定义它的变量说明语句的位置，决定其性质和作用范围。

在主程序的说明部分说明的变量称为全局变量，全局变量的作用域定义为整个程序。运行时它们占据内存的数据段。数据段最大为 65520 字节。如果需要超过 65520 字节的全局数据，则应分配更大的结构作为动态变量（见“§1－4　PASCAL 的数据类型、类型常量、运算符和表达式”）。

在过程和函数内说明的变量称作局部变量。一个局部变量只有在定义它的过程和函数内有效。运行时它们占据内存的栈段。每调用过程和函数一次，就分配其局部变量进栈，这些局部变量被安排在栈顶。每退出过程和函数一次，栈顶的局部变量出栈。程序执行的任何时候，需进栈的局部变量的存储单元数不能超过堆栈容量。堆栈容量一般为 16384 字节，可通过系统（$s 编译指令）调整为 1024～65520 字节中的任何值。

在 BASIC 语句中，只有全局变量。

（五）过程与函数说明

定义主程序需要调用的过程（函数）。过程（函数）说明包括过程（函数）首部说明和过程（函数）体说明。

Ⅰ、过程说明

为完成一个操作的子程序作算法过程的说明。过程说明的定义形式有两种：

① procedure　过程名；　　{过程首部}

　　过程体；

② procedure　过程名（形式参数表）　{过程首部}

　　过程体；

其中，过程体的说明有三种形式：

1、分程序

分程序结构如同主程序结构的分程序部分一样，具有说明部分和语句部分。在语句部分调用本过程，就是递归调用。

2、超前说明 forward

有时，一个过程在被定义之前就需要引用它，由于过程的定义说明位于过程引用之后因此编译程序在遇到该过程名时，不能立即知道它的意义，不能检查这样的引用是否合法。为了解决这个上、下文的自然衔接问题，可以通过 forward 语句，在这个过程被引用之前，先预告该过程的过程名和要求参数的情况。在 forward 过程说明之后的某处再加上过程的定义