最适合小学生和初中生使用的信息学奥林匹克竞赛辅导用书

聪明人的游戏

——信息学探秘

陈泰延　杨　鹏　骆丽璇　编著

实战篇

广东省出版集团
广东科技出版社

最适合小学生和初中生使用的信息学奥林匹克竞赛辅导用书

聪明人的游戏

——信息学探秘

（实战篇）

陈泰延　杨　鹏　骆丽璇　编著

广东省出版集团
广东科技出版社
·广州·

图书在版编目（CIP）数据

聪明人的游戏——信息学探秘（实战篇）/陈泰延，杨鹏，骆丽璇编著. —广州：广东科技出版社，2006.12
ISBN 7-5359-4255-5

Ⅰ.聪…　Ⅱ.①陈…②杨…③骆…　Ⅲ.信息学 — 普及读物　Ⅳ.G201-49

中国版本图书馆CIP数据核字（2006）第140793号

出版发行：广东科技出版社
（广州市环市东路水荫路11号　邮码：510075）
印　　刷：佛山市浩文彩色印刷有限公司
（南海区狮山科技工业园A区　邮码：528225）
规　　格：787mm×1 092mm　1/16　印张11.25　字数250千
版　　次：2006年12月第1版
2006年12月第1次印刷
定　　价：20.00元

序

信息学奥林匹克（Olympiad in Informatics,简称 OI）是旨在培养青少年创新能力和进行全面素质教育的一项重要活动。邓小平同志早在 1984 年就提出“计算机的普及要从娃娃抓起”，十多年来，我省已经形成了从 IOI（国际信息学奥林匹克）——NOI（全国信息学奥林匹克）——GDOI（广东省信息学奥林匹克）的系列性活动，对于促进广大青少年普及计算机知识，培养一大批出类拔萃的优秀计算机后备人才起到积极的作用。

开展信息学奥林匹克活动，让那些学有余力的青少年能够在课外继续学习计算机知识，培养自主学习能力、实践能力和创新能力。

大家知道，人的智力活动核心是思维，思维是人脑的主要功能，思维的广阔性、敏锐性、灵活性、深刻性和创造性的程度是衡量一个人能力发展水平的重要标志，训练思维能力旨在开发智力，用电脑帮助开发人脑是开展信息学奥林匹克活动的一个重要的特点。同时我们培养高素质人才，不仅仅看其智力因素，还要看其非智力因素：坚韧不拔的坚强意志，严谨求实的科学作风，力争上游的奋进精神，胜而不骄、败而不馁的良好心态，团结互助、友好合作的团队精神，这些都应当而且能够在信息学奥林匹克活动中得到培养。

多年来，广东省信息学奥林匹克竞赛活动以培养高素质人才作为开展活动的指导思想，坚持高标准、严要求，坚持课余培训的原则，鼓励参加信息学奥林匹克的同学全面发展，除了鼓励他们在参加信息学奥林匹克活动中，以“更快、更高、更强”的奥林匹克精神勉励自己以外，还建议他们在力所能及的情况下参加数学、物理等学科的奥林匹克竞赛，鼓励他们学好语文、英语等其他学科。由于有了上述明确的要求和目标，近年来，广东省所涌现出的一大批信息学奥林匹克竞赛选手，受到国内各知名大学的欢迎，他们进入大学后绝大多数都成为品学兼优、全面发展而又有特长的优秀学生。

由佛山市南海区具有丰富辅导经验的信息学教练员编著的这套教材，体现了新课程改革的理念和以学生为中心的教学思想。该教材将传统以知识罗列的组织方式，改为以极具趣味性的故事引入，通过以解决问题为核心，将知识分解渗透到问题解决的过程中，很好地解决了以往信息学教学枯燥难懂的难题，是一套适合小学、初中使用的难得的好教材。相信该教材的出版，将有利于中小学信息化教育活动的开展，有利于优秀信息学后备人才的培养。

广东省青少年信息学竞赛委员会主任
全国青少年信息学奥林匹克竞赛广东队总教练
中山大学信息科技学院计算机科学系教授
郭嵩山

2006 年 11 月

前　言

“信息学奥林匹克活动是聪明人的游戏”。郭嵩山教授的这句名言，无时不在鼓舞着我们。信息学奥林匹克活动的意义并不限于竞赛本身，它对开发青少年的智力潜能，培养更多的“聪明人”还具有独特的作用。

青少年信息学奥林匹克活动是发现和培养有潜质、有特长学生的重要阵地。事实证明，经过信息学活动的熏陶，不但能提升学生的思维能力和自主探究能力，形成坚毅的个性和科学的思维方法，还有利于学生科学素养、高尚人格、竞争意识和团队合作精神的培养。

在开展信息学活动的过程中，我们难以找到一本适合小学和初中入门学生使用的教材。现有的教材多拘泥于语言本身或是经典算法的直观呈现，专业性强、枯燥难懂，使不少信息学初学者望而却步，很快从信息学奥林匹克活动的队伍中消失。

原本其乐无穷的信息学奥林匹克活动为什么如此令人生畏？如何才能化解这一难题呢？经过长期的思考和实践，我们认为首先要从解决辅导教材入手，有了一本能令学生爱学、乐学并从中品味到信息学乐趣的好教材，就能吸引一大批学生加入到信息学奥林匹克活动中来，信息学奥林匹克活动才能方兴未艾。

为此，我们几位长期从事青少年信息学辅导工作的教练员，本着为信息学奥林匹克活动奉献绵薄之力的意愿，开始了这套教材的编著工作，并终能将《聪明人的游戏——信息学探秘》奉献给广大信息学爱好者。

本套辅导用书紧紧结合小学和初中学生的认知水平及心理特点编写，共分入门篇（待出版）和实战篇两册。入门篇从解决学生学习和生活中的一些简单问题入手，渗透利用Pascal语言编程解决问题的基本思想方法；实战篇注重一些更复杂问题的解决，重点在于程序设计的实现技能、数学模型的建立、算法和简单数据结构在问题解决中的应用。

本书打破了传统的知识组织方式，巧妙地将知识渗透到解题过程中，通过引例故事层层深入，以兴趣为起点，以活动为主线，螺旋上升地组织教学内容，针对有趣的问题来学习编程，从不同方面引导学生理解知识内涵，始终围绕问题解决这一核心开展学习活动。

全书力求化繁为简、描述准确、通俗流畅、图文并茂，尽量淡化专业性的概念，以趣味性的问题引领学生学习，让学生通过本教材真正找到学习信息学的乐趣。

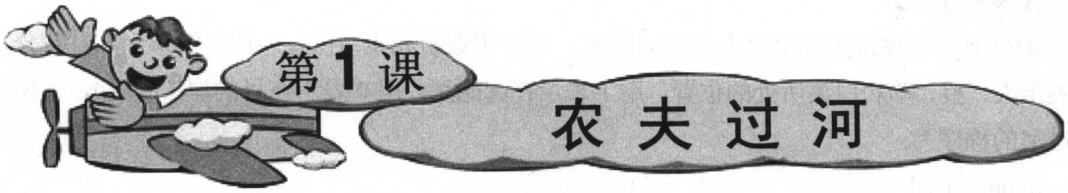
本书在正式出版之前，经佛山市南海区内部试用，取得了非常好的使用效果：全区开展信息学奥林匹克活动的小学和初中达到80%以上，小学到初中的信息学梯队建设创出新成效，在小学和初中涌现出了一大批信息学好苗子，在全国青少年信息学联赛中，南海的初中选手连年雄居全省之首，获一等奖人数连续两年均超过全省的四分之一。

在本书编写过程中，得到了广东省青少年信息学竞赛委员会主任、中山大学郭嵩山教授的不吝指导，也得益于全国信息学奥林匹克高级指导教师、全国著名信息学教练江涛老师的不少启发；同时，南海区广大教练为本书的编写提供了很多宝贵意见，在此一并致谢。

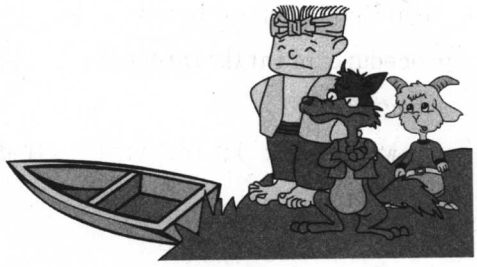
由于编者水平有限，书中定有不当和疏漏之处，恳请广大读者批评指正。

编　者

目　录



英国神学家阿尔奎恩里所著的《益智题》一书中有这样一道有趣的智力题：



有位农夫带着一只狼、一只羊、一棵白菜来到河边要过河。河边刚好有一条空着的小船，过河时，船很小仅能允许主人带一样东西，若带两样东西上船，船便会沉下去。另一方面，若没人照管，狼会吃掉羊，羊又将啃白菜，因此，狼和羊，羊和白菜在主人不在的情况下，是不可以放在一块的。问主人应该采取什么样的过河方案，才可以把狼、羊、白菜都安全地带到对岸去呢？

【分析】

① 要解答这个问题，我们要抓住两点考虑：一是农夫每次过河只能带一样东西；二是狼和羊、羊和白菜在农夫不在时不能同时在一块。

② 如果每次只从起始岸带东西到对岸，必定会发生农夫不在而狼与羊或羊与白菜同时在一块而导致羊或白菜被吃的后果。

③ 既然题目没有规定不准把已带过河的东西又带回来，我们当然可以让农夫把已带到河对岸的东西在返回起始岸时再带回来，以避免被吃掉。

【答案】

农夫可以按如下的方案过河：

(1) 人带羊过河；

(2) 人单独返回；

(3) 人带狼过河；

(4) 人带羊返回；

(5) 人带白菜过河；

(6) 人单独返回；

(7) 人带羊过河。

过河的方案可能不止一种，你还能帮助农夫找到其他的过河方案吗？

【参考程序】

如果要用计算机来模拟以上的过河方案，可以用数组元素 a[1]，a[2]，a[3]和 a[4]分别表示人、狼、羊和白菜所处的位置，用 1 表示在起始岸，用 0 表示在目的岸（对岸），则模拟过河的程序为：

```
program P1_1;
var i:integer;
 a:array[1..4] of byte;
procedure print(k:integer);     {显示某次过河或返回后,起始岸的状态}
  begin
    write(k,':'); for i:=1 to 4 do write(a[i]:6);
    writeln;
  end;
begin    {主程序}
  for i:=1 to 4 do a[i]:=1;     {置初始状态（全部在起始岸）}
  writeln('       man  wolf sheep vegetable'); {显示人、狼、羊和白菜的英文}
  print(0);     {显示未过河前的状态——全为 1}
  a[1]:=0;a[3]:=0;print(1);     {人带羊过河}
  a[1]:=1;print(2);     {人单独返回}
  a[1]:=0;a[2]:=0;print(3);     {人带狼过河}
  a[1]:=1;a[3]:=1;print(4);     {人带羊返回}
  a[1]:=0;a[4]:=0;print(5);     {人带白菜过河}
  a[1]:=1;print(6);     {人单独返回}
  a[1]:=0;a[3]:=0;print(7);  {人带羊过河后,a[1]～a[4]全为 0,表示已全部过河}
end.
```

试改写程序 P1_1，帮助农夫找出另一种过河的方案。

●算 法

“农夫过河”问题是一个典型的算法问题。什么是算法呢？**算法**就是为解决某一问题所采取的一系列方法和步骤，它是关于问题解决办法的精确描述。

利用程序解决问题时，通常要完成以下三方面的工作：

⑴ 组织数据：就是要考虑如何组织被处理的数据、中间结果和最终结果。数据的组织方式也称为数据结构。例如，利用一维数组 b 存放 20 个学生的体重，这就是利用数组存放数据的一种数据组织方式。

⑵ 设计算法：在确定数据结构的基础上，设计处理数据的方法与步骤。

⑶ 编写程序：根据数据结构和算法编写程序代码，反复调试和测试直到程序符合要求。

简单来说，就是：

程序=数据结构+算法

●算法的表示

描述算法的方法很多，常见的有自然语言、流程图、伪代码等。但不管用哪种方法描述，都要转换为程序，计算机才能处理。

1．自然语言

我们可用日常使用的自然语言（文字）来描述算法。但使用自然语言有时会显得繁琐冗长或不够严谨。

【例 1】任意给定一个整数 n，判断它是偶数还是奇数。

【分析】

根据数学的有关知识，能被 2 整除的整数就是偶数，否则就是奇数。我们可以将判断的算法用自然语言表示为：

⑴ 输入整数 n ；

⑵ 如果 n 除以 2 的余数是 0，n 就是偶数（显示有关信息），转⑷ ；

⑶ 否则 n 是奇数（显示有关信息）；

⑷ 结束。

2．流程图

采用流程图的方式来描述算法时，使用不同形状的图形代表不同的操作。用流程图描述的算法形象、直观，逻辑结构清晰，易于编写程序。

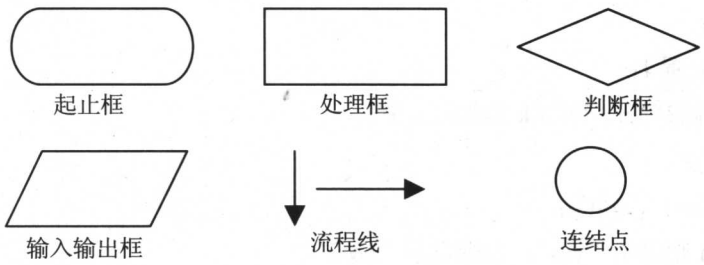


图 1-1 流程图常用的图形符号

对于例 1 中的算法，可用如图 1-2 所示的流程图表示。

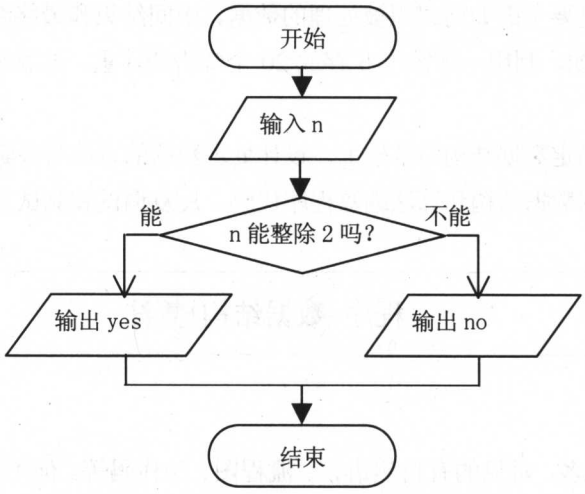


图 1-2 描述例 1 算法的流程图

一个完整的流程图要有一个开始框和一个结束框，分别表示算法的开始和结束。图形之间按照算法的过程用带箭头的流程线连接起来。

【例 2】判断一个整数 n 是否为完全数。试分别用自然语言、流程图表示求解的算法，并编写程序。

【分析】

① 根据数学知识，完全数是指其不包括本身的所有因数的和等于本身的整数。如 6 的因数有 1，2，3 和 6，由于 1+2+3=6，故 6 是完全数。

② 只要找出 n 的所有因数，再把它们（不包含 n）累加起来，看累加和是否等于 n，就能判断 n 是否为完全数了。

【算法描述】

用自然语言描述的算法如下：

⑴ 输入 n ；

⑵ s 置 0，k 置 1；

⑶ 如果 k 能整除 n，则将 k 累加到 s 中；

⑷ k 递增 1；

⑸ 如果 k≤n-1 ，转⑶；

⑹ 如果 s=n 则显示“yes”，否则显示“no”；

⑺ 结束。

用流程图描述的算法如图 1-3 所示。

【参考程序】

```
program P1_2;
var n,s,k:integer;
begin
  readln(n);
  s:=0;
 {存放各因子和的变量，先置为 0}
  for k:=1 to n-1 do
    if n mod k=0 then s:=s+k;
 {找出不含 n 本身的各个因数,加到 S 中}
  if s=n then writeln('yes')
       else writeln('no')
end.
```

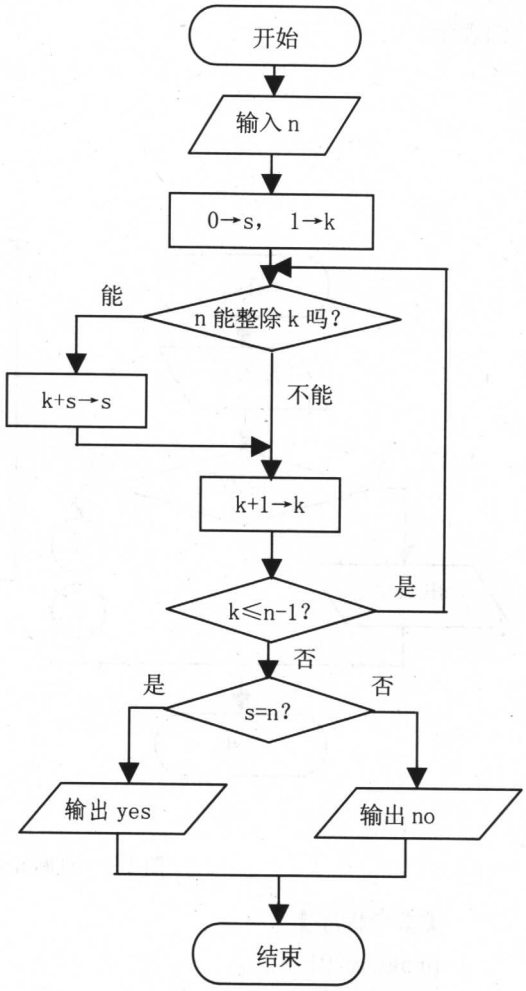


图 1-3　判断 n 是否为完全数的流程图

【例 3】任意给定一个大于 1 的整数 n，判断它是否为质数。试分别用自然语言、流程图表示问题求解的算法，并编写程序。

【分析】

根据质数的有关概念可知：除了 1 和它本身，不能被其他数整除的数就是质数（质数也叫素数）。2 是最小的质数。由此，我们得到用自然语言描述的算法如下：

⑴ 输入 n ；

⑵ 判断 n 是否等于 2，若 n=2，则显示“yes”表示 n 是质数，转⑷ ；

⑶ 依次检验 2～n-1 是不是 n 的因数，即能否整除 n 。若有这样的数，则显示 n 不是质数(“no”)；若没有这样的数，则显示 n 是质数（“yes”)；

⑷ 结束。

上述算法中的第⑶点还可以进一步细化，你能将它详细写出来吗？

本例算法的流程图如图 1-4 所示。流程图的右边部分就是前面自然语言算法中第(3)点的详细表示。

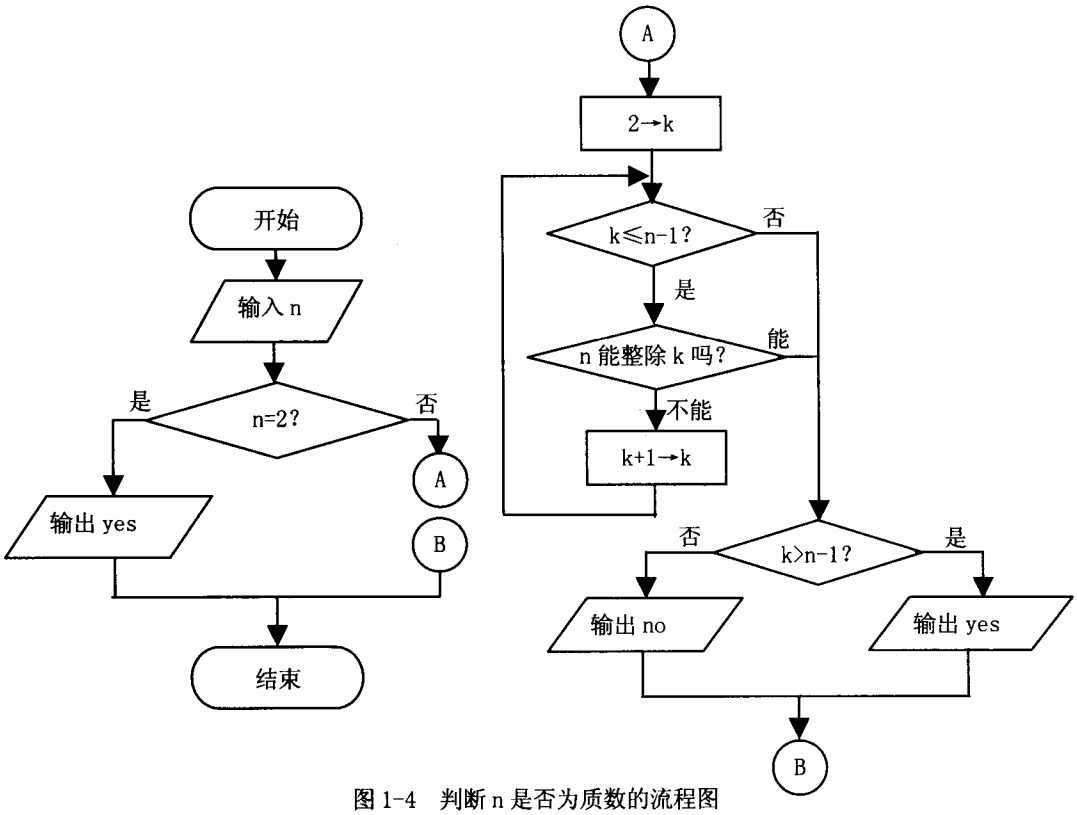


图 1-4　判断 n 是否为质数的流程图

【参考程序】

```
program P1_3;
var n,k:integer;
begin
  readln(n);
  if n=2 then writeln('yes')      {若 n=2,显示为质数}
  else
    begin     {k 从 2～n-1 不断除以 n，看是否能被整除}
     k:=2;
     while (k<=n-1) and (n mod k>0) do
        k:=k+1;     {当 k≤n-1 且不能整除 n 时，再试下一个 k}
      if k>n-1 then writeln('yes')      {如果k>n-1,说明没有任何一个k值能整除n}
        else writeln('no')
    end;
end.
```

1．找出“农夫过河”的其他方案，并编写程序来模拟过河的情况。

2．有 A，B，C，D 四个人，晚上要过一个独木桥（都是从一边走向另一边），4 个人必须用手电筒照明才能通过，但手电筒只有一个，而且独木桥只能同时允许 2 个人通过，已知 A 通过需要 1 分钟，B 通过需要 2 分钟，C 通过需要 5 分钟，D 通过需要 10 分钟，如果两个人同时走，过桥的时间是较慢的人的时间。问怎么走才能在 17 分钟内 4 个人都走过去?

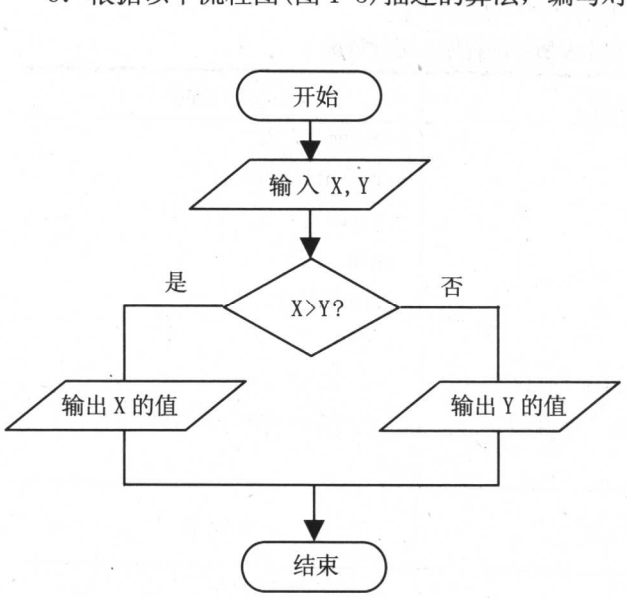
3．分别用自然语言和流程图描述“交换两个变量 **x** 和 **y** 中的数”的算法。

4．判断一个整数 p 是否为 5 的倍数，若是则显示“yes!”，否则显示“no!”。请用自然语言和流程图分别描述解题算法，并写出程序。

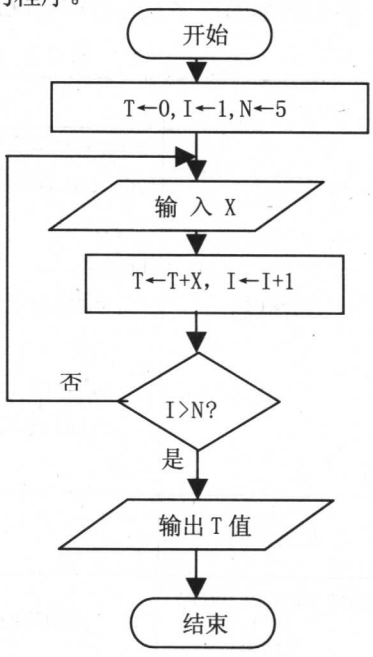
5．小雄家的邻居过来串门，看见小雄妈妈正在洗碗，便问她妈妈家里来了多少客人。小雄妈妈回答说：“每两个客人合用一个菜碗，每三个客人合用一个汤碗，每四客人合用一个饭碗，共用碗 65 只。”你能帮邻居算出有多少个客人吗?

请用自然语言和流程图分别描述解题算法，并写出程序。

6．根据以下流程图(图 1-5)描述的算法，编写对应的程序。



(1)



(2)

图 1-5　两个算法的流程图

第2课 为奥运作贡献

2008 年北京奥运会是奥运会历史上的一次盛举，是中国人的骄傲。小明由此引发了一个奇想，要向全国小学生发出倡议，建议每位同学都向奥运会捐出一些零用钱，为奥运会作点贡献。

小明倡议的捐款规则是这样的：

第 1 位捐款的同学只捐 1 分钱，第 2 位捐 2 分钱，第 3 位捐 3 分钱……第 n 位捐 n 分钱。

已知2005年全国共有小学生10864.07万人（即 108 640 700 人）。为了能算出最理想的情况下能为奥运会捐得多少钱（以元作单位），小明请来了班里的两位编程高手小朋和小友分别编程序来解决这个难题。

他们所编的程序及运行的情况怎么样呢？我们来看看。

表 2-1　小朋和小友所编的程序及运行情况

	小朋的程序	小友的程序

程序代码	program P2_1;
	var i,n:longint;
	  s:real;
	begin
	  readln(n);
	  s:=0;
	  for i:=1 to n do s:=s+i;
	  writeln(s/2:0:2,' yuan');
	end.
		program P2_2;
		var i,n:longint;
		  s:real;
		begin
		  readln(n);
		  s:=0;
		  s:=(1+n)/2*n;
		  writeln(s:0:2,' yuan');
		end.

运行时间	27.02 秒	0.05 秒
输入 108 640 700 时的结果	2950700477700000.00 yuan	2950700451300000.00 yuan

从表 2-1 可以看出，小朋的程序运行的时间要比小友的长得多，两个程序的计算结果也不同。为什么会这样呢？

【分析】

① 对比程序 P2_1 和 P2_2，我们会发现两个程序最大的差异是：小朋的程序采用将 108 640 700 位同学的捐款逐个累加来求总数的，程序中的 for 语句需要循环 108 640 700 次；而聪明的小友则先根据数学规律找出计算总和的公式，再根据公式只用一个赋值语句就把总数算出来了。这样，小友的程序自然就比小朋的程序快得多了！

② 小友是怎样把求总和的公式找出来的呢？

我们来看：

1=(1+1)÷2×1=1

1+2=(1+2)÷2×2=3

1+2+3=(1+3)÷2×3=6

1+2+3+4=(1+4)÷2×4=10

1+2+3+4+5=(1+5)÷2×5=15

1+2+3+4+5+6=(1+6)÷2×6=21

……

分析总和与前面各数的关系，可以归纳出一个求前 n 个自然数总和的公式，即：

1+2+3+…+n=(1+n)÷2×n

你能写出求前 n 个奇数之和的计算公式吗？

③ 单从程序看，小朋的程序虽然比较慢，但计算方法也是对的，为什么结果与小友的不一样呢？那是因为实型数经过多次运算后，产生了较大的误差，从而导致两个程序运行的结果不一致，小友所编程序的运行结果更准确一些。

以上例子说明，虽然编程解题的算法不是惟一的，但算法总有优劣之分。在编程解题时，我们要尽可能使用准确、快捷的算法；同时，尽可能对程序进行优化，提高程序的运行效率。

要提高程序的运行效率，一是要选择一种高效的算法，二是要对程序进行优化，尽量减少不必要的重复。

如何对程序进行优化呢？

1．避免重复计算

在程序中实际运算对象不变的情况下，不做重复运算。

(1) 优化常数计算。

将重复的常数计算提前一次完成，后面直接引用。如：

原程序段	优化后的程序段
for i:=1 to 1000 do s[i]=3200/12	w:=3200/12; for i:=1 to 1000 do s[i]=w

原程序段计算了 1000 次 3200/12 。优化的方法是只计算一次，存入一个变量中，在循环中将该变量直接赋给每一个数组元素。

(2) 优化表达式运算。

当变量的值没有改变时，多个表达式中相同的部分可抽取出来，提前进行计算，在原表达式直接引用这个中间结果进行运算。如：

原程序段	优化后的程序段

x:=t1+a*b/c;
y:=t2+a*b/c;
z:=t3+a*b/c*y;
	m:= a*b/c;
	x:=t1+m;
	y:=t2+m;
	z:=t3+m*y;

原程序段中三个赋值语句都出现了“a*b/c”，并且在此期间 a，b，c 的值没有改变，因此可以将“a*b/c”提前到第一个赋值语句前计算，将结果存入 m 中，以后再在出现“a*b/c”处直接引用 m。

2．尽量采用较快的算术运算

不同算术运算的执行效率相差很大，在可能的情况下尽量使用较快的运算。

表 2-2　各种算术运算的运算效率

运算	加	减	乘	除	乘方
效率	最基本	稍慢	慢	慢于乘法	很慢

(1) 以加代乘，如可用 I+I+I 代替 3*I。

(2) 以乘代除，如可用 a*0.2 代替 a/5。

3．避免重复判断

当循环体内有判断语句时，会不断重复判断，这将影响程序的执行效率。如有可能，尽量不使用判断。

例如：要统计全班 50 人中各分数段的人数（设分数为 1～100 分，以每 10 分为一个段），用 cn[1]～cn[10]分别存放各段人数。

程序段 1 在循环中使用了 10 个判断语句，逐一对分数进行分段判断；而程序段 2 则使用 trunc((sc-1)/10)+1 来确定每个分数段对应的数组下标，从而不需任何判断就能进行统计，这将使程序段 2 运行效率大大提高。

程序段 1	程序段 2

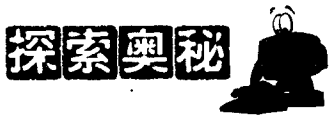
for k:=1 to 50 do
  begin
  read(sc);
  if sc in [1..10] then cn[1]:=cn[1]+1;
  if sc in [11..20] then cn[2]:=cn[2]+1;
  if sc in [21..30] then cn[3]:=cn[3]+1;
  ……
  end;
	for k:=1 to 50 do
	  begin
	  read(sc);
	  m:=trunc((sc-1)/10)+1;
	  cn[m]:=cn[m]+1;
	  end;

4. 减少循环次数

循环次数的多少往往对程序速度影响最大。例如，下面是“求 1000 以内 5 的倍数之和。”的两个程序段：

原程序段	优化后的程序段
for k:=1 to 1000 do if k mod 5=0 then s:=s+k;	for k:=1 to 200 do s:=s+5*k;

原程序段要循环 1000 次，循环中还要判断；而优化后的程序段只循环 200 次，比原程序段少循环 800 次，且无须进行判断，大大提高了运行效率。



【例 1】下面是“找出整数 n 的所有因数”的程序，请分析并对程序进行优化。

```
program P2_3;
var n,k:integer;
begin
  readln(n);
  for k:=1 to n do
    if n mod k=0 then write(k:5);writeln;
end.
```

【分析】

① 能够整除 n 的数就是 n 的因数。在程序 P2_3 中，用 1～n 之间的所有整数逐个检验是否能被 n 整除，如果能就显示该数。这样循环体中的 if 语句将被重复执行 n 次。

② 事实上，任何一个整数的因数都是成对出现的，只要找出其中一个因数，就能确定相应的另一个因数了。例如：