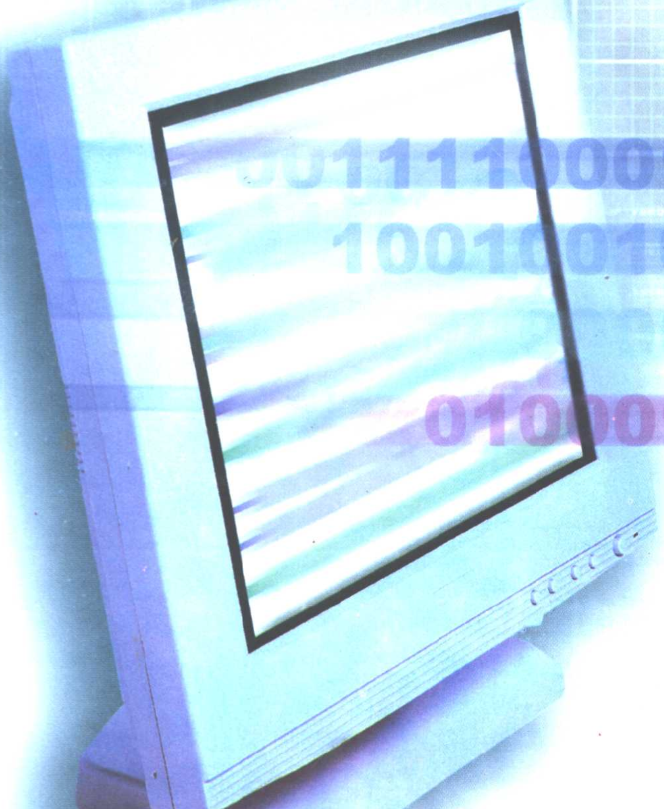
程序设计与问题求解

编著 朱国进 孙莉 朱明

ACM国际大学生程序设计竞赛基础及解题指导



东华大学出版社
（原中国纺织大学出版社）

程序设计与问题求解

ACM 国际大学生程序设计竞赛基础及解题指导

朱国进　孙　莉　朱　明　编著

東華大學 DONGHUA UNIVERSITY 重点资助教材建设项目

东 华 大 学 出 版 社
（原中国纺织大学出版社）

内容摘要

本书旨在为计算机素质教育和计算机教学改革提出和建设一种新概念程序设计教材，书中选用的问题全部来源于 ACM 国际大学生程序设计竞赛试题。其特点是：围绕应用环境中实际问题的求解过程来阐述和讲解程序设计思想方法和相关技术知识，向学生展示如何设计和选择合适的数据结构来表示实际问题中的处理对象，如何把一个实际问题转化成一个程序可计算的逻辑模型，以及如何考虑程序运行的效率来满足问题求解对时间的要求等。

本书适用的范围和对象更为广泛。它既可以用于程序设计基础课程，又可以用于 ACM 国际大学生程序设计竞赛基础的训练；不仅适合于计算机专业的学生，而且适合于非计算机专业的学生，同时还可以作为广大计算机程序设计兴趣爱好者的参考书籍。

图书在版编目(CIP)数据

程序设计与问题求解/朱国进，孙莉，朱明编著．
上海：中国纺织大学出版社，2002．5

ISBN 7-81038-305-1

Ⅰ．程… Ⅱ．①朱… ②孙…③朱… Ⅲ．程序设计
Ⅳ．TP311．1

中国版本图书馆 CIP 数据核字(2002)第 026051 号

责任编辑 邵静
封面设计 阿进

程序设计与问题求解
朱国进 孙 莉 朱 明 编著
东华大学出版社出版
(原中国纺织大学出版社)
(上海市延安西路 1882 号 邮政编码：200051)
新华书店上海发行所发所 常熟市大宏印刷厂印刷
开本：787×1092 1/16 印张：14．5 字数：345 千字
2002 年 5 月第 1 版 2002 年 5 月第 1 次印刷
印数：0001—3000
ISBN 7-81038-305-1/TP・19
定价：28．00 元

序言

长期以来，有一种现象一直困扰着我们：一些计算机书面考试成绩优秀的学生往往很难写出一个正确的程序来解决应用环境中的实际问题。

近年来，由国际计算机学界著名的最具权威性的 ACM 学会（Association for Computer Machinery）主办的国际大学生程序设计竞赛（简称 ACM/ICPC）在国内蓬勃兴起，使得这一问题更为显见。ACM/ICPC 是世界各国大学生中最具影响力的国际计算机赛事，被誉为全球计算机世界的奥林匹克竞赛，竞赛试题具有丰富的实际背景，许多试题都是从计算机软硬件的实际开发过程中演变而来。作者所在的学校在以往几届比赛中都选派了最优秀的学生，但他们中大多数人在比赛时不能做对一题。赛后，许多参赛学生认为自己不可能解答出 ACM/ICPC 竞赛试题。这种情况在其他学校也存在。在每年一次的比赛中，都有许多学校的参赛队的成绩为零分。

针对这种状况，作者在一年前组织全校优秀的程序设计学生进行赛前集训。在集训的初期，作者发现参加集训的很多学生缺乏基本的程序设计思想和方法，并且在多数情况下不能把已学的程序设计知识用于解决应用性问题。产生这种情况的一个主要原因是：学生在学习程序设计课程时只是把注意力集中在程序设计语言上，而忽略了对程序设计思想和方法的学习，也很少考虑和尝试把所学的程序设计知识用于解决应用环境下的实际问题。这种情况有点类似于传统英语教学中只注重语法和句型练习，而忽视了在语言实际环境下的训练，导致了大多数非英语专业的学生学习了十几年英语却不能很好地用英语来表达自己的想法与别人进行交流。

基于这种认识，作者在赛前对学生集训时以求解应用环境下的实际问题为目标进行程序设计训练，结果得到了意想不到的收获：经集训后，2000 年参赛的两个队分别获得了第六名和第十二名，取得了历史性的突破，2001 年又分别在亚洲上海和新加坡两个赛区获得第七和第六名的好成绩。这份意外收获暗示着这样一个信息：传统的以讲解语言为主的程序设计教学法需要进行改革。值得一提的是，在这些获奖队员中只有一名是计算机专业的学生，其余都是非计算机专业的学生，并且除了一名学生以外都是进入大学以后才开始学习程序设计的。这说明以应用性问题求解为主的程序设计教学法具有推广的前景，它不仅适合于计算机专业的学生，而且也适合于非计算机专业的学生。

由此，作者认为基于问题求解的程序设计可以作为高校的一门选修课，称为“程序设计与问题求解”，以培养学生用程序设计来解决应用环境下实际问题的能力，使学生的程序设计水平在整体上上一个台阶。如果把传统的程序设计课程比作纯数学课程，那么基于问题求解的程序设计课程可以看成是应用数学课程。因此，后者将填补前者留下的空白，更适合于以应用为目的的程序设计学习者。

基于问题求解的程序设计课程是对传统计算机教学的挑战，它的开设及其配套教材的建设，对计算机教学改革和计算机素质教学将产生深远的影响。

有关计算机程序设计的教材有很多，如：PASCAL 语言程序设计，C 语言程序设计，C++

面向对象程序设计等等。这类教材有一个共同的特征，即都是围绕某一具体的程序设计语言的特点来编写的，教材中所举的例子多半是用来帮助读者理解相关语句的作用和使用规则。在目前的计算机程序设计课程中，使用的都是这类教材，其结果是：大多数学生更多地是在学习一种程序设计语言，而没有把学习重点放在程序设计本身上。

对于计算机应用专业和非计算机专业来说，开设计算机程序设计课程的主要目的应该是：使学生能够把所学的程序设计知识像数学那样作为一种工具来解答应用环境中的实际问题。这也是本书编著的目标之一。

本书旨在为计算机素质教育和计算机教学改革提出和建设一种新概念程序设计教材，这种新概念教材将围绕应用环境中实际问题的求解过程来阐述和讲解程序设计思想方法和相关技术知识，从而不仅能使学生更好地掌握程序设计技术，而且还有利于培养和提高学生用程序设计技术来解决实际问题的能力，为学生成为21世纪高科技创新人才铺设桥梁。

为此，本书以应用环境中实际问题的求解过程为核心，向学生展示如何设计和选择合适的数据结构来表示实际问题中的处理对象，如何把一个实际问题转化成一个程序可计算的逻辑模型，以及如何考虑程序运行的效率来满足问题求解对时间的要求，从而引发学生对程序设计的兴趣和思考，开阔学生对问题求解的思路，激发学生在程序设计应用领域中的想象力和创造力，进而达到培养和提高学生用程序设计技术来解决实际问题能力的目的。

因此，本书的特色是以应用环境中实际问题的求解过程为核心。这种求解过程与一般的程序设计教材内容不同，通常涉及二百行左右的程序，而一般程序设计教材中所使用的举例程序大多数在二十行至五十行之间。两者数量上的这种差异隐含着两者之间存在着质的差别。设计一个有一定质量的二百行左右的程序，通常要把程序分成几段，每段一般不超过五十行，这就涉及模块化程序设计思想和方法。显然，这种方法对于五十行以下的程序不太适用。此外，相对于五十行以下的程序来说，二百行左右的程序更易于阐明程序可读性和程序易修改性等概念，更能够让学生掌握结构化程序设计方法和面向对象的程序设计概念。

本书的另一个特色是选用的问题全部来源于ACM国际大学生程序设计竞赛试题。这不仅能使程序设计课程瞄准国际水平，而且还可以为有兴趣参赛的学生打好扎实的基础。然而，本书与以程序设计竞赛试题解析为核心的课外读物不同，它强调程序设计基础知识的学习，这主要体现在：

1. 本书为每一个问题设计一系列由易至难的相关子问题，并通过对这些子问题的求解来逐步阐明与该问题相关的程序设计基础知识。

2. 本书将一些可以用相关程序设计基础知识来求解的问题编排在一起，从而有利于对相关程序设计基础知识的讨论和学习。

3. 本书的内容覆盖整个程序设计基础知识，内容的编排采用循序渐进的方式。

因此，本书适用的范围和对象更为广泛。它既可以用于程序设计基础课程，又可以用于ACM国际大学生程序设计竞赛基础的训练；不仅适合于计算机专业的学生，而且适合于非计算机专业的学生，同时还可以作为广大计算机程序设计兴趣爱好者的参考书籍。

作者

2002年5月

目　录

第1章　简单问题

1.1　求和问题

1.1.1 问题描述

你的任务是求出所有在1到n之间的数的和，包括1和n在内。

输入

输入一个简单的整数n，n的绝对值不能大于10000。

输出

输出所有在1和n之间的数的和（包括1和n在内）。

样例输入

-3

样例输出

-5

1.1.2 题目分析

这个题目是求和问题。由于是1到n之间的数，所以就是一个等差数列。可以利用初等数学中的等差数列求和原理 $s=(A_1+A_k)\times k/2$ 来解这道题，其中的s为数列前k项的和，A_1 是数列的第一项，A_k 是数列的第k项。

当n大于等于1时，求和公式为 $(1+n)\times n/2$。当n小于1时，数列的长度变成了 $|n|+2$（包括了0和1），求和公式变为 $(|n|+2)\times(n+1)/2$。

考虑计算中涉及 n^2，n的最大值为10000，n^2 达到 10^8，超出整数的范围，所以要将变量设为实型变量，以防止溢出。在输出时，采用不显示小数部分的输出格式。

该算法的时间复杂度为O(1)，即与n无关。

1.1.3 相关知识点

1. 标准场宽

标准场宽，即系统规定的数据输出的标准格式，数据在输出设备上所占的列数称为“场宽”，PASCAL语言对各类数据规定了标准场宽，一般微型机上的PASCAL语言对于场宽规定如表1-1所示。

2. 指定场宽

有两种用户自己定义场宽的方法：

1）指定单场宽

表 1-1 一般微型机上的 PASCAL 语言对于场宽规定

数据类型	标准场宽
整型	14
实型	16
布尔型	True 为 4，False 为 5
字符型	1
字符串	串长

指定场宽时，语句书写形式有所改变，指定单场宽的写语句形式为：

write（<输出项>：场宽）；

其中：场宽只能是整型表达式。若输出项的数据位小于指定场宽，数据输出时将“向右靠齐”；若数据位数大于指定场宽，则输出数据的实际位数。

2）指定双场宽

对于实型数据来讲，在单场宽中均以科学记数法输出，若希望以小数形式输出，可采用指定双场宽的方法。指定双场宽的写语句形式为：

write（<输出项>：场宽 1：场宽 2）；

其中：输出项的值必须是实型数据；场宽 1 为整个数据所占的宽度（包括符号位）；场宽 2 表示小数部分所占的宽度。场宽 1 和场宽 2 只能是整型表达式，且场宽 1 应大于场宽 2。例如：x=123.789，执行语句

```
writeln(x:7:3);
```

输出结果为 123.789。当输出数据位数大于指定场宽时，整数部分将根据实际位数输出，小数部分则受场宽 2 的限制，舍去后面多余的位数。当小数部分位数小于场宽 2 时，则在小数后补 0。

例 1　输入两个不超过 10000 的整数，求它们的乘积，并以整数形式输出。

譬如，输入：10000　　9999

　　　输出：99990000

```
program  chengji(input,output);
var   a,b,sum:real;
begin
    read(a,b);
    sum:=a*b;
    writeln(sum:0:0);
end.
```

1.1.4 源程序代码

```
program acmz(input,output);
var
n:real;
sum:real;
```

```
begin
    read(n);
    if n>=1 then
        begin
            sum:=n*(n+1)/2;
        end;
    if n<1 then
        begin
            sum:=(abs(n)+2)*(1+n)/2;
        end;
    writeln(sum:0:0);
end.
```

1.2 阶乘问题

1.2.1 问题描述

GSM 网络最重要的部分被称为基本收发站(BTS)，由这些收发站形成的区域称为细胞，并且每一部电话都以最强的信号与 BTS 相连接（小而简化地看)。当然，收发站需要维护，技术人员们需要定期地检查它们的工作功能。

ACM 技术人员们最近面对一个非常有趣的问题，由于要去检查一组基本收发站 BTS，他们需要找到一条最短路径去检查所有给定的地点并返回中心公司楼。程序员们花了几个月的时间研究这个问题但没有结果，他们无法找到足够快的解决方案。不久以后，一位程序员在一篇学术文章中找到这个问题。不幸的是，他发现这个问题被称作“TSP 问题”而且很难解。如果我们要去检查 N 个 BTS，我们可以任何顺序参观，就有 N!个可能的路线，其数字表示形式称为阶乘，并可表示为 1×2×3×4×…×N 的乘积，因此，即使相对较小的 N 也会有一个非常大的阶乘数。

程序员们明白他们没有机会解出这道题。但因为他们已经接受了这个由政府下达的研究课题，他们必须继续该研究工作并至少得出一些结论，所以他们开始研究数的阶乘。

例如，定义函数 Z。对于任何给定的整数 N，Z(N)是指以十进制表示的 N!的末尾零的个数。他们发现这个函数是非递减函数。如果我们有两个数 $N_1<N_2$，那么 $Z(N_1) <= Z(N_2)$。这是因为我们不会在乘上任何一个整数的时候将末尾的零省去，只会得到更多的零。函数 Z 非常有趣，所以我们需要一个计算机程序来有效地确定 Z 的值。

输入说明

输入的第一行是一个单个的确定的正整数 T。他指明接下来的数字的个数。然后是 T 行，每一行包括一个确定的正整数 N，1 <= N <= 1000000000。

输出说明

对每一个数字 N，产生一行输出包含一个非负整数 Z(N)。

样例输入

6
3
60
100
1024
23456
8735373

样例输出

0
14
24
253
5861
2183837

1.2.2 题目分析

N!=1×2×3×…×N，此问题在于求把 N!分解成素数的乘积时，2 和 5 的幂是多少。末尾的零的个数等于 2 和 5 的指数中的较小的一个。故问题导致对从 1 到 1000 的整数中求是 2k 和 5l 型数倍数的数的个数。

例 1　10!=1×2×3×4×5×6×7×8×9×10=28×34×52×71=3628800，由于其中 5 上的指数为 2，所以乘积后的 0 为 2 个。

不超过 N 的正整数中是 5 的倍数的数共 A=[N/5]个（[ ]为取整符号例[2.5]=2），其中有 B=[A/5]个是 25 的倍数，B 个中还有 C=[B/5]个 125 的倍数……

例 2　10!= 1×2×3×4×5×6×7×8×9×10，其中 5 的倍数为 5，10，共[10/2]=2 个，这 2 个中有 25 的倍数[2/5]=0 个。所以，5 的指数为 2。

例 3　100!= 1×2×3×…×99×100，其中 5 的倍数为 5，10，…，95，100 共 20 个，20 个中有 25 的倍数 25，50，75，100 共 4 个，其中有 125 的倍数[4/5]=0 个。所以，5 的指数为 20+4=24。

所以，若 N!分解成素数的乘积，其中 5 的指数应为[N/5]+[[N/5]/5]+…+0。

因为 2n<5n，故在 N!中 2 的指数远远超过 5 的指数。

所以 N!末尾零的个数取决于 5 的指数。

1.2.3 源程序代码

```
program f(input,output);
var casecount,casepos:longint;
    count,data:real;
begin
```

```
    readln (casecount);
    for casepos:=1 to casecount do
        begin
            readln(data);
            count:=0;
            repeat
                data:=trunc(data/5);
                count:=count+data;
            until data=0;
            writeln(count);
        end;
end.
```

1.3 时钟夹角问题

1.3.1 问题描述

钟面上的时针和分针之间的夹角总是在 0°~180° 之间(包括 0° 和 180°)。举例来说，在十二点的时候两针之间的夹角为 0° 。而在六点的时候夹角为 180° ，在三点的时候为 90° 。本题要解决的是计算 12:00 到 11:59 之间任意一个时间的夹角。

输入

输入的数据包含各种各样的情况。每一组测试数据包含两个数字：第一个数字代表小时（大于 0 小于等于 12），第二个数字代表分（在区间 [0，59] 上）。两个数据皆为零则表示输入结束。

输出

对应每组测试数据，用常用格式显示时间以及这个时候时针和分针间的最小夹角，精确到小数点后一位。输出格式如下所示。

输入举例

12 0
12 30
6 0
3 0
0 0

期待的输出

At 12:00 the angle is 0.0 degrees.
At 12:30 the angle is 165.0 degrees.
At 6:00 the angle is 180.0 degrees.
At 3:00 the angle is 90.0 degrees.

1.3.2 题目分析

1. **模型**

分针转动的速度为时针的 12 倍。计算分针和时针之间的夹角只需分针离开 0 刻度的角度减去时针离开 0 刻度的角度即可（皆取顺时针方向），如果这个结果大于 180° 则取其补角。

若输入时间 3:20，设时针转过的角度为 a，分针转过的角度为 b，则有 a=360° ×(3/12)+360° ×(20/60)×(1/12)=100° ，b=360° ×(20/60)=120° 。所以两角度之间的夹角为 a 和 b 的差的绝对值|100° -120° |=20° 。当其大于 180° 时取补角。

若输入时间 3:50。设时针转过的角度为 a，分针转过的角度为 b。则有 a=360° ×(3/12)+360° ×(50/60)×(1/12)=115° 。b=360° ×(50/60)=300° 。所以两角度之间的夹角为 a 和 b 的差的绝对值|115° -300° |=185° 。当其大于 180° 时取补角 175° 。

2. **子问题说明**

例 1 输入一个时间 x:yy，求从零时刻按照顺时针方向到该时刻的分针转过的角度 b。

比如输入 3:20，输出 b=120.0

解 分针每走 60 格是 360° ，因此每一格的角度大小是 360° /60。这样读入一个分度 yy。角度就是 yy×360° /60。

例 2 输入一个时间 x:yy，求从零时刻按照顺时针方向到该时刻的时针转过的角度 a。

比如输入 3:20，输出 a=100.0

解 时针的转过的角度与钟点中的时刻度和分针转过的角度有关。首先，时针每走 12 格是 360° ，因此每一个准点 x 的度数就是 x×360° /12。其次，分针走 60 格，时针走 5 格，也就是时针的速度是分针的 5/60。那么如果分针走了 20 格，时针转过的角度就是 360 ×20/60×5/60。

这样读入一个时刻 x:yy，那么时针的角度 a 就等于 x×360° /12+360° ×yy/60×5/60。

例 3 按时钟表示格式输出。

比如输入 12:0，输出 12:00

解 当输入的分针刻度是一位数时，要加一个 0 使其变为两位数。若变量 b 中存放输入的分针刻度，在输出时加如下判断 if b div 10=0 then write('0'，b) else write（b）。

1.3.3 源程序代码

```
program clock;
const
    infile='clock.in';
    outfile='clock.out';
var
    fin,fout:text;
    h,m:integer;
    a,b,r:real;
```

```
function calhour(hr,mt:integer):real;                {计算时针角度的函数}
begin
    calhour:=360/12*hr+360/60*mt/12;
end;

function calminute(m:integer):real;                  {计算分针角度的函数}
begin
    calminute:=360/60*m;
end;

begin
    assign(fin,infile);
    reset(fin);
    assign(fout,outfile);
    rewrite(fout);
    while not eof(fin) do begin
        readln(fin,h,m);                       {读入时和分}
        if (h=0) and (m=0) then break;         {同时为 0,输入结束}
            a:=calhour(h,m);                   {计算时针的刻度}
            b:=calminute(m);                   {计算分针的刻度}
            r:=abs(a-b);                       {求夹角}
            if r>180 then r:=360-r;            {如果大于 180°,则取补角}
                                               {以下为输出结果}
            write(fout,'At ',h,':');
            if m div 10 =0 then write(fout,'0',m)   {当是一位时补齐两位输出}
            else write(fout,m);
            writeln(fout,' the angle is ',r:0:1,' degrees.');
    end;
    close(fout);
    close(fin);
end.
```

1.4 有理化无理数问题

1.4.1 问题描述

有理数是指可以用两个整数的比来表示的数。对一个给定的质数 p，在数字定理中有一条基本的定理就是没有一个有理数等于 p 的平方根（缩写为 SQRT(p)）。这样一个数字称为无理数。我们还知道有一些有理数是接近于 SQRT（p）。

现在，给定一确定的整数 n，我们定义了一个集合 Q_n，集合中的元素由两个小于或等于 n 的整数的比来表示的。例如，集合 Q_4 包括 11 个数字{1/1，1/2，1/3，1/4，2/1，2/3，3/1，3/2，3/4，4/1，4/3}。2/2，2/4，3/3，4/2 和 4/4 不包括在内。这是因为它们分别等于 1/1，1/2，1/1，2/1 和 1/1。

你的任务是编写一程序来读入两个整数 p 和 n，而得到两个有理数 x/y 和 u/v，使得 u/v<SQRT(p)<x/y，并且在 Q_n 集合中没有其他元素介于 u/v 和 x/y 之间。当 n 大于 SQRT(p) 时，这样一对有理数总存在。

输入

输入包括多行，每行包括两个整数，一个质数 p 和一个整数 n，格式如下：

p n

两个数用空格隔开。假定 p 和 n 都小于 10000，并且 n 要大于 SQRT(p)。用一行两个零表示输入结束。

输出

在每一行中，输出由空格隔开的两个有理数，格式如下：

x/y u/v

这两个数必须是最简的。例如，6/14 和 15/3 是不被接受的。因为它们可以分别简化为 3/7 和 5/1。

输入举例

2 5

3 10

5 100

0 0

输出举例

3/2 4/3

7/4 5/3

85/38 38/17

1.4.2 题目分析

本题目要求对一个质数的平方根在一定范围内找出与它最接近的用分式表示的两个有理数，一个比它小，一个比它大。其中分子、分母的大小在[1…n]内。首先计算 p 的平方根，表示为 sp。然后对分母进行 1 到 n 的循环，对每一个分母 m，求得最接近 sp 的两个有理数，这两个有理数的分子为 z=trunc(m*p)和 z+1（z+1<10000），这样得到对分母 m 最接近 sp 的两有理数与 sp 的关系为 $\frac{z}{m} < sp < \frac{z+1}{m}$，最后从上述得到的有理数中，从小于 sp 的数中找到最大的那个数即为 $\frac{u}{v}$，从大于 sp 的数中找到最小的那个数即为 $\frac{x}{y}$。

举例说明：取 p=2，n=5，则得 $sp = \sqrt{2}$。通过对分母的 1 到 5 循环的得到分子如表