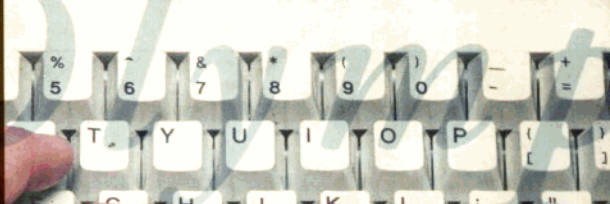


informatics



全国青少年信息学（计算机）奥林匹克丛书

# 全国青少年信息学 （计算机） 竞赛例题解析

中国继续教育联合学院推荐用书

李立新 主编  
陈树楷 吴文虎 王晓敏 审校

Computer

Computer



人民邮电出版社  
PEOPLE'S POSTS &  
TELECOMMUNICATIONS  
PUBLISHING HOUSE

Olympic

本书全面地介绍了计算机程序中常用的数据结构和算法,并结合全国和部分省市计算机奥林匹克信息学竞赛中的有关题目作了较全面的分析。

本书共分两部分,即基础部分和提高部分,每个部分按算法的性质分成若干章节。书中对每一个题目给出了详细的算法分析和程序设计方法并附有程序清单和运行结果。

本书可作为学校奥林匹克信息学(计算机)竞赛活动及广大中小学计算机教师继续教育用书,也可作为大专院校有关计算机课程的教学参考用书。

# 全国青少年信息学(计算机) 奥林匹克丛书编委会

顾 问： 陈树楷  
          吴文虎  
          潘懋德

主 任： 曹志超

副主任： 徐修存      李树岭      孙彦德

编 委： 李树奎      马启洲      李立新

          蒋新儿      蔡绍稷      王晓敏

          郭嵩山      张 明      孔 军

          吴再陵      张桂林      黄叶亭

          黄国洪      阮志远      吕晓春

今日世界,无论是科技竞争、经济竞争,还是综合国力竞争,归根到底是人才竞争。

世界各国都把科学技术作为综合国力竞争的经济热战主要武器,争先占领 21 世纪的技术制高点。各国政府越来越重视科学技术的作用,愈来愈重视人才的培养。许多国家相继制定了加强大学本、专科教育,抓好中小学教育和成人教育的措施,并提出了以做好继续教育为重要手段的人才第一战略,积极开发人才资源。

党中央和各级政府历来重视教育、培养人才工作。中央提出了科教兴国和可持续发展的战略方针,并一再强调科学教育的发展关系到全民素质的提高和中国现代化的前途。邓小平同志说过:“改革经济体制,最重要的、我最关心的,是人才。改革科技体制,我最关心的还是人才。”江泽民主席提出“严重的问题在于教育干部”深得人心,“要把培训教育作为一种开发,作为一种投入,是一种适应竞争需要的学习竞赛,要树立终身培训的观念。”已成为许多部门领导的共识。

随着我国社会主义市场经济的发展和与国际市场的逐步接轨,经济体制从传统的计划经济体制向社会主义市场经济体制转变,经济增长方式从粗放型向集约型转变这两个根本转变,急需大量专业人才。这使得人们对继续教育给与更大的重视,各类培训中心如雨后春笋般地冒了出来,既处于有利的发展机遇,又面临着十分繁重的任务。我国专业技术人员 2700 多万人,仅占我国人口总量的 2.3%,与经济建设关系密切的自然科学技术人员只占人口总量的 1.5%,比中等发达国家 80 年代的水平还差 2~3 倍。我国专业技术人员在产业间分布不合理,作为社会财富直接创造者的企业和农业领域尚未成为人才资源配置的重点,知识结构与现代经济发展需要差距较大,相当一部分专业技术人员老化,缺乏创新意识和创造能力,也缺乏市场经济知识与管理能力。高新技术人才、高级管理人才以及新兴行业的专业技术人员严重短缺,学术带头人年龄偏大。这种状况直接影响着我国国民经济与社会发展,直接影响着我国跨世纪宏伟目标的实现。要改变这种状况,继续教育起着十分关键的作用。总之,为实现我国社会主义发展宏伟战略目标,为造就千百万高素质的专业技术人员队伍,继续教育任重而道远。

信息科学技术学的需求更加突出。专业技术人员、广大教师都需要信息科学技术学科方面的继续教育,同时与对青少年计算机普及教育相联系,责任就更加重大。我们的工作符合专业技术人员的要求,符合市场经济对人才的需求,也符合现代化建设的需求。中国科协及其所属学会开展的学术交流、科学普及等各项工作,说到底,还是为了培养人才。从这一角度上看,直接培养人才的继续教育工作对学会来说,更是至关重要。

中国计算机学会有了10多年继续教育工作的实践,从1986年开办各类专门课程培训班,至今已有12年历史,累计开办了20多种培训班,接受培训人数约为10000人次。编辑出版丛书共5套26本,发行20多万册。还开展了《奥林匹克竞赛与计算机教学改革》、《计算机促进结构性教育改革》等软课题研究。

本套丛书的系列定名为:《全国青少年信息学(计算机)奥林匹克丛书·中国继续教育联合学院推荐用书》是有意义的。该丛书是在已使用了两年的内部讲义基础上,经进一步修改审定后出版,这是中国计算机学会普及工作委员会对质量审慎的表现。

我们期望着这套丛书能对广大各课师资参加计算机继续教育的学习有所帮助;能对于广大青少年参加全国计算机奥林匹克分区联赛有所帮助。我还希望读者能对这套丛书提出宝贵修改意见,指出不当之处。

最后,我还要感谢诸位顾问、编委和人民邮电出版社有关人员的辛勤工作,使得这套丛书能在极短的时间与广大读者见面。

中国科协副主席  
中国继续教育联合学院院长



1998年2月12日于北京

全国青少年信息学(计算机)奥林匹克竞赛活动是经中国科协、国家教委批准的一项全国性青少年学科竞赛活动,至今已成功地举办了两届。为了提高参赛青少年的计算机知识水平,全国分区联赛组委会编写了一本《分区联赛辅导教程》,经两年使用,现根据广大读者的要求,同时也为了配合中国科协、国家教委等部门联合开展的“园丁科技行动”,满足广大中小学计算机教师进行继续教育的迫切需要,对该教程进行了大幅度的扩充,形成了一套系列教材,本习题集是这套教材中的一本。

参加本习题集编写工作的有南京大学的蒋新儿教授、南京航空航天大学的李立新教授,同时,李申杰、章准平等同志参加了部分习题的调试工作。

江苏省科协青少部的王晓敏、夏军同志在本书出版过程中给予了很大的帮助,在此表示感谢。

编者

1998年2月

## 第一部分 基础部分

## ●第一章 PASCAL 程序设计基本方法一

1.1	ASCII 码字符串输出	3
1.2	求自然数 $n$ 的不同因数的个数	6
1.3	已知因数个数, 求 $n$	11
1.4	求一元多项式在 $X_0$ 处的值	22
1.5	NOI'95 第三题	25
1.6	统计输入的字符串中的字母频率	28
1.7	NOI'95 第一题	31
1.8	链表排序问答	36

## ●第二章 PASCAL 程序设计基本方法二

2.1	线性表基本操作单元	42
2.2	合并线性表	46
2.3	以基准数重排顺序	51
2.4	带头结点的线性链表基本操作单元	56
2.5	逆序合并链表	63
2.6	数组元素逆时针方向赋值	69
2.7	求幂集	79
2.8	自然数 $n$ 的拆分表达式	82
2.9	求 $n$ 的拆分数目	92

## 第二部分 提高部分

## ●第三章 自顶向下逐步求精

3.1	求最大数、最小数和平均值	101
3.2	Faibonacci 数列	102
3.3	求 2 个数的最大公因子	105
3.4	换钱问题	107
3.5	求素数	109

3.6	求完全数 .....	111
3.7	Nicomachus 定理及应用 .....	113
3.8	字符串统计问题 .....	114
3.9	字符串加密 .....	116
3.10	统计字符串中字符出现的频率 .....	118
3.11	数的排列 .....	120
3.12	求出数列的前 $n$ 项 .....	122
3.13	奇数幻方 .....	124
3.14	全排列问题 .....	126
<b>●第四章 回溯算法</b>		
4.1	骑士的游历 1 .....	130
4.2	骑士的游历 2 .....	134
4.3	迷宫问题 .....	140
4.4	砝码称重 .....	144
4.5	钱币问题 .....	146
4.6	四色问题 .....	148
4.7	无根树的编码 .....	151
4.8	背包问题(1) .....	155
<b>●第五章 动态规划</b>		
5.1	最短路径问题 .....	158
5.2	求最长不下降序列 .....	162
5.3	最小代价子母树 .....	165
5.4	背包问题(2) .....	171
5.5	四塔问题 .....	175
5.6	最小代价 .....	178
5.7	挖地雷 .....	182
<b>●第六章 多精度计算</b>		
6.1	多精度加法 .....	187
6.2	多精度减法 .....	190
6.3	多精度与单精度乘法 .....	194
6.4	多精度与多精度乘法 .....	196
6.5	多精度与单精度除法 .....	200
6.6	多精度与多精度除法(1) .....	202
6.7	多精度与多精度除法(2) .....	206
6.8	数塔问题 .....	210
6.9	计算 $e$ .....	220



## ●第七章 递归方法

7.1 钢板分割成小正方形问题 .....	225
7.2 用递归方法求解 8 皇后问题 .....	226
7.3 平面直线交点问题 .....	228
7.4 推广的哈夫曼编码 .....	230
7.5 表达式去括号 .....	238

## ●第八章 其他问题

8.1 过河问题 .....	247
8.2 行程问题(1) .....	256
8.3 行程问题(2) .....	260
8.4 求出方程 $x^n + y^n = s^n + t^n$ 的最小整数解 .....	263
8.5 最少转弯问题 .....	267
8.6 士兵排队问题 .....	270
8.7 逻辑集成电路 .....	276
8.8 平面地砖曲线问题 .....	281
8.9 $1 \times 2$ 的骨牌问题 .....	290
8.10 钢板切割零件问题 .....	304
8.11 工厂零件生产问题 .....	310
8.12 取数问题(1) .....	312
8.13 称球问题 .....	313
8.14 堆塔问题 .....	315
8.15 取数问题(2) .....	318

# 第一部分

## 基础部分



# PASCAL 程序设计

## 基本方法一

### 1.1 ASCII 码字符串输出

#### 一、问题描述

已知两个字符串及其长度,要求将这两个字符串按 ASCII 码顺序排列输出。

#### 二、算法设计

输入:字符串的长度可设为  $m, n$ ;存储的两个字符串可设为数组  $A[1..m], B[1..n]$ 。

判断方法:

$A[i]$  和  $B[i]$  比较 ( $i$  从 1 开始):

$A[i] > B[i]$             说明 B 在前, A 在后;

$A[i] < B[i]$             说明 A 在前, B 在后;

$A[i] = B[i]$             继续对  $A[i+1]$  和  $B[i+1]$  比较。

可能发生的情况是逐个字符比较时都相等,这时短的字符串便排列在前了;如果两个字符串不仅每个字符相同而且长度也相同,那么它们的排列顺序也无所谓谁在前,谁在后。

特殊的情况是字符串长度 ( $m$  或  $n$ )  $\leq 0$ , 这种情况可以作为出错处理。

#### 三、程序设计

比较两个字符串的顺序可用一个过程 `comp` 来实现,比较结果可借助于一个变量参数  $k$  来存储。

例如,可以约定:

$k = 2$             用于控制比较进行(循环控制变量)。

$k = -1$           表示 A 在 B 前,比较结束。

$k = 1$             表示 B 在 A 前,比较结束。

#### 四、程序清单

```

PROGRAM e101;
  CONST maxlen = 20;           {比较字符串长度上限设定}
  TYPE s = ARRAY[1..maxlen] OF char;
  VAR
    m, n, i, j, k; integer;
    a, b      :s;

  PROCEDURE comp(A, B: s; m, n; integer; VAR k: integer);
    VAR i: integer;

  BEGIN
    i := 1; k := 2;
    WHILE K = 2 DO

      BEGIN
        IF i = m + 1 THEN k := - 1
        ELSE
          IF i = n + 1 THEN k := 1
          ELSE BEGIN
            IF A[i] < B[i] THEN k := - 1;
            IF A[i] > B[i] THEN k := 1;
            i := i + 1
          END
        END

      END

    END;

  BEGIN { main }
    REPEAT
      writeln('Input length two string: ');
      readln(m, n);
      IF (m <= 0) or (n <= 0) THEN writeln('Length > 0 !!')
    UNTIL (m > 0) and (n > 0);
    writeln('Input first string: ');
    FOR j := 1 to m DO read(A[j]); readln;
    writeln('Input second string: ');
    FOR j := 1 to n DO read(B[j]); readln;
    comp(A, B, m, n, k);
    IF k = - 1 THEN BEGIN

```

```

FOR i:= 1 TO m DO write(A[i]); write(' ');
FOR i:= 1 TO n DO write(B[j]); writeln
END;
IF k = 1 THEN BEGIN
FOR i:= 1 TO m DO write(B[i]); write(' ');
FOR i:= 1 TO n DO write(A[j]); writeln
END;
readln
END.

```

## 五、运行结果 (下划线表示输入)

Input length two string:

0 -1

Length &gt; 0 !!

Input length two string:

3 0

Length &gt; 0 !!

Input length two string:

4 5

Input first string:

1234

{A}

Input second string:

12304

{B}

12304, 1234

{B, A}

Input length two string:

4 4

Input first string:

()-+

{A}

Input second string:

{}[ ]

{B}

()-+ , {}[ ]

{A, B}

Input length two string:

3 5

Input first string:

xyz

Input second string:

xyzab

xyz, xyzab

Input length two string:

5 5

Input first string:

abcde

Input second string:

abcde

abcde, abcde

Input length two string:

6 5

Input first string:

abcdef

Input second string:

abcde

abcde, abcdef

**注意:** 若将问题中的字符串理解为由英文字母组成的单词,那么输出的排列结果即为字典的顺序。要用这里设计的算法排列单词的字典顺序,算法中的字符串长度  $m, n$  的输入将会显得十分烦琐,应将长度隐含在单词的输入之中。例如:可以约定单词输入后再输入一个特别约定的字符作为结束输入一个单词的标记。

## 1.2 求自然数 $n$ 的不同因数的个数

### 一、问题描述

任给一个自然数  $n$ , 求出这个自然数不同因数的个数  $m$ 。

### 二、问题分析

下面列举出若干  $n$  和  $m$  的值先看一看:

$n = 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \dots\dots$

$m = 1 \ 2 \ 2 \ 3 \ 2 \ 4 \ 2 \ 4 \dots\dots$

显然,自然数 1 处于特殊地位;任何素数(大于 1,只有 1 和自己为自然数的因数)的因数个数都为 2;合数的因数个数  $> 2$ ,究竟多大则较难寻找出规律来。为此可以通过试除法来求出  $m$ 。

### 三、算法设计

当  $n = 1$  时  $m = 1$ ,是特例;

其它情况的处理如下:

```
m := 2;
for i := 1 TO k do
  if n mod i = 0 then m := m + 2;
```

其中

```
k := trunc(sqrt(n));
```

这里取  $m := m + 2$  是因为只要发现  $n$  的一个小于等于  $\sqrt{n}$  的因数,必然同时有一个大于等于  $\sqrt{n}$  的因数。但对  $n$  正好是完全平方数时,上述的  $m$  应减去 1。

#### 四、程序设计

程序中根据输入的  $n$ , 输出  $1 \sim n$  的相应  $m$  值的一张表, 以供对照用。

#### 五、程序清单

```
PROGRAM e102;
VAR
  n, i, j, k: longint;
  m: integer;

BEGIN {main}
  REPEAT
    write('Input N = ');
    readln(n);
    IF n <= 0 THEN writeln('N > 0!!!')
  UNTIL n > 0;
  writeln('N = ', 1; 10, ' M = ', 1);
  FOR j := 2 to n DO BEGIN
    m := 2;
    K := trunc(sqrt(j));
    FOR i := 2 TO k DO
      IF j MOD i = 0 THEN m := m + 2;
    IF j = sqr(k) THEN m := m - 1;
    writeln('N = ', j; 10, ' M = ', m);
  END
END.
```

#### 六、运行结果 (下划线表示输入数据)

Input N = <u>-1</u>	Input N = <u>20</u>
N > 0!!!	N = 1 M = 1
Input N = <u>1</u>	N = 2 M = 2
N = 1 M = 1	N = 3 M = 2



Input N = 10	N =	4	M = 3		
N =	1	M = 1	N =	5	M = 2
N =	2	M = 2	N =	6	M = 4
N =	3	M = 2	N =	7	M = 2
N =	4	M = 3	N =	8	M = 4
N =	5	M = 2	N =	9	M = 3
N =	6	M = 4	N =	10	M = 4
N =	7	M = 2	N =	11	M = 2
N =	8	M = 4	N =	12	M = 6
N =	9	M = 3	N =	13	M = 2
N =	10	M = 4	N =	14	M = 4
			N =	15	M = 4
			N =	16	M = 5
			N =	17	M = 2
			N =	18	M = 6
			N =	19	M = 2
			N =	20	M = 6

**注意:**

为了对输入的  $n$  仅输出一次  $m$ , 可将程序清单修改如下:

```
PROGRAM e102-1;
VAR
  n, i, j, k: longint;
  m: integer;
BEGIN {main}
  REPEAT
    write('Input N = ');
    readln(n);
    IF n <= 0 THEN writeln('N > 0!!')
  UNTIL n > 0;
  IF n = 1 THEN writeln('N = ', 1; 10, ' M = ', 1)
  ELSE BEGIN
    j := n;
    m := 2;
    K := trunc(sqrt(j));
    FOR i := 2 TO k DO
      IF j MOD i = 0 THEN m := m + 2;
      IF j = sqrt(k) THEN m := m - 1;
    writeln('N = ', j; 10, ' M = ', m);
  END
END.
```