

1988年全国青少年 计算机程序设计竞赛 辅导及BASIC题解

吴文虎 范俊堂 编著



電子工業出版社

1988年全国青少年

计算机程序设计竞赛

辅导及BASIC题解

吴文虎 范俊堂 编著

电子工业出版社

内 容 简 介

本书为一九八八年全国青少年计算机程序设计竞赛辅导与题解。该书前半部分介绍结构化的BASIC程序设计思想,用一些实例来讲解如何在一般BASIC语言中实现程序结构化的思路、方法与技巧;后半部分对全国程序设计竞赛BASIC的试题作了详细的分析与解答;书末选编了部分省市的程序设计竞赛试题。

该书深入浅出,适于中等学校师生及初学者阅读参考,也是参加程序设计竞赛选手的必读材料。

1988年全国青少年
计算机程序设计竞赛
辅导及BASIC题解
吴文虎、范俊堂 编著
责任编辑:路 石

*

电子工业出版社出版(北京海淀区万寿路)
电子工业出版社发行 各地新华书店经销
北京密云华都印刷厂印装

*

开本: 787×1092毫米 1/32 印张: 6.625 字数: 143千字

1989年9月第一版 1989年9月第一次印刷

印数: 5000册 定价: 2.90元

ISBN 7-5053-0626-x/TP·99

前 言

1988年全国青少年计算机程序设计竞赛于1988年8月5日~10日在北京清华附中举行。28个省市自治区和解放军、石油系统共30个代表队120名中小学学生参加了竞赛,年龄最小的是9岁。经过笔试、上机、口试答辩,42人分别获得一、二、三等奖。在发奖大会上,老一辈科学家周培源发表了寄以深切期望和热情洋溢的讲话。

本届竞赛分BASIC组和LOGO组。BASIC组有89人参赛,获一等奖的5名,他们是柴海新(河南)、庄骏(北京)、陈果(湖北)、余智华(湖北)、薛晓岗(广东);获二等奖的10名;获三等奖的15名。LOGO组有31人参赛,获一等奖的2名,他们是王强(上海)、马琳(北京);获二等奖的4名;获三等奖的6名。

从竞赛的成绩看,青少年计算机教育和校外科技活动起步较早的省、市(如上海、北京、广东等)都有进步。边远地区的成绩都有不同程度的提高。这次BASIC组一等奖第一名由河南省选手柴海新夺魁,参赛的其他两名河南选手也都取得好成绩。这反映了河南的工作扎实,而且注意对学生的德育教育。湖北省、天津市、广西自治区的成绩都不错,湖北省BASIC组两名获一等奖。

竞赛期间,广西、湖北队介绍了他们是如何取得领导、家长和社会各界支持的;上海、北京队介绍了他们组织活动的

经验。教委、科协、学会组成有力的领导班子是推动青少年计算机知识教育的重要关键。通过计算机知识的学习，全面推动了文化课的学习，是取得家长支持的重要因素。编制软件程序结合教学管理和课程实验的深入，可以引起学校领导的兴趣，使献身于青少年计算机教育事业的积极分子不断涌现。

竞赛期间，组织了青少年同专家学者的座谈会。老一辈专家学者鼓励青少年好好学习知识，启发引导青少年学科学、爱科学、用科学。

1988年全国青少年计算机竞赛活动是成功的，赛出了水平，赛出了风格。

本届竞赛的根本目的是进一步推进和普及计算机教育。通过竞赛可选拔计算机的后备人才。1986年全国青少年计算机竞赛一等奖第一名获得者李劲同学，目前是清华大学无线电系二年级的学生，在同年级200多名同学中，李劲同学各门功课都名列前茅。

今年竞赛中出现你追我赶的局面，在今后推广普及计算机知识教育中，一定会涌现出大量的计算机科学技术及其应用的后备人才。

目 录

前 言

第一章	结构化的流程图简介.....	(1)
第二章	从观察、归纳到解题算法.....	(15)
第三章	从逻辑思维到算法.....	(31)
第四章	字符串处理与查找.....	(46)
第五章	递推算法.....	(59)
第六章	递归算法及其实现方法.....	(78)
第七章	1988年全国青少年程序设计竞赛 BASIC 语言部分题解与分析.....	(96)
	一、笔试试题及解答.....	(96)
	二、上机试题及解答.....	(155)
附 录	部分省市的程序设计竞赛试题.....	(188)
	一、1988年北京市青少年尤美杯计算机程序设 计竞赛初试题.....	(188)
	二、1988年北京市青少年尤美杯计算机程序设 计竞赛初试上机题.....	(193)
	三、1988年北京市青少年尤美杯计算机程序设 计竞赛复赛试题.....	(196)
	四、1988年甘肃省中小学计算机程序设计竞赛 笔答试题.....	(202)

第一章 结构化的流程图简介

青少年学习电子计算机,从学习一门计算机语言(BASIC或LOGO语言)入手,就设备和师资条件来说,比较适合我国的国情。通过课堂教学和课外活动,青少年从小接触计算机这一先进工具,可起到开发智力、拓展思路、提高逻辑思维和分析解决问题的能力,这对我国青少年素质的提高,智力与能力的发展是有深远意义的。

学好BASIC语言,要从基本语句学习开始,扎扎实实地去逐步掌握程序设计的方法与技巧。为了编程,首先要研究算法,不仅数值运算(比如解联立方程、求素数等)要研究算法,非数值运算(比如字符处理、书刊检索等)也要研究算法。算法从广义上说就是解决问题的步骤,不过这些步骤都要充分考虑计算机的特点。为了描述算法,可以用许多工具,比如自然语言、伪代码、一般流程图和结构化流程图等。一般流程图在有关程序设计语言的书中都有介绍,不再重复,这里只想介绍结构化流程图,旨在帮助读者提高程序设计的质量和效率。

我们先举一个例子来说明怎样画结构化的流程图,以及它所包含的基本结构。

例1. 从键盘输入10个正数,求平均值和10个数中最大的数。

这道题的程序如下:

```

5  REM SAMPLE-1
10 TOTAL = 0
20 MAX = 0
30 FOR N = 1 TO 10
40 INPUT "A = "; A
50 IF A > MAX THEN MAX = A
60 TOTAL = TOTAL + A
70 NEXT N
80 AVERAGE = TOTAL / 10
90 PRINT "AVERAGE = "; AVERAGE
100 PRINT "MAX = "; MAX
110 END

```

与上述程序对应的结构化框图画在图1中。

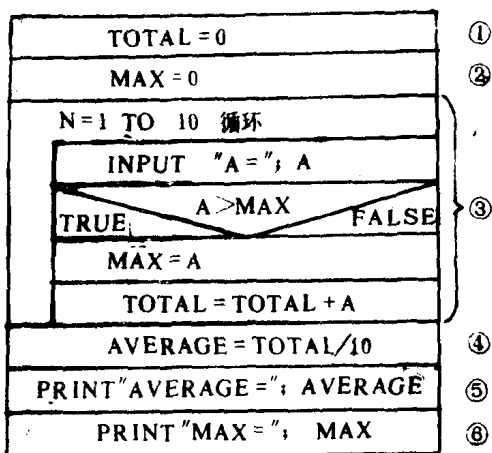


图1

从图1可见三种基本结构：

1. 顺序结构：指按顺序一步一步执行的结构。自上而

下顺序执行，如图2。每个矩形框是一种基本运算(也可能是系列运算)。两个以上的矩形框组成顺序结构。

图2中的①——对应程序的第10句；

②——对应20句；

③——对应30句到70句；

④——对应80句；

⑤——对应90句；

⑥——对应100句。

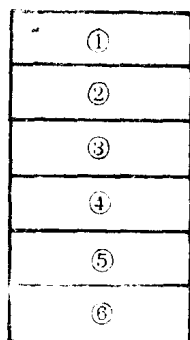


图2

2. 循环结构：见图3。

图3中的⑦为循环结构中的循环体。在这个例子中，循环体⑦对应40到60句。循环控制变量是由30句和70句来设定和控制的。这个循环结构的图形特点是：一个大矩形框中套一个小一些的矩形框，且有两边重合，小框表示循环体，大框上注明循环次数或循环条件。

3. 分支结构（又称选取结构）：见图4。

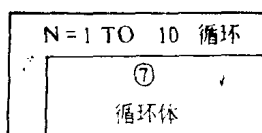


图3

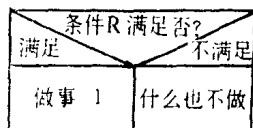


图4

图4中，倒三角表示分支条件，其左下方为满足该条件应执行的分支，右下方为不满足该条件应执行的分支。满足条件用True表示，不满足条件用False表示，可简写为T和F。

在本例中分支结构作为循环体的一个组成部分，或者说循环结构中嵌套着分支结构。

下面我们从本例中超脱出来，讲分支结构与循环结构更一般的表示。

分支结构有两种，一种已如图4所示，用IF...THEN...语句就可实现；另一种如图5，要用IF...THEN...ELSE语句，但这种语句只在True BASIC中有，而在一般BASIC版本中没有，在一般BASIC版本中可以这样处理：

```
10 IF R THEN做事1:GOTO 30
```

```
20 做事2
```

```
30 做事3
```

⋮

其中R表示分支条件。



图5

或者：

```
10 IF R THEN 50
```

```
20 做事2 (之一)
```

```
30 做事2 (之二)
```

```
40 做事2 (之三):GOTO 80
```

```
50 做事1 (之一)
```

```
60 做事1 (之二)
```

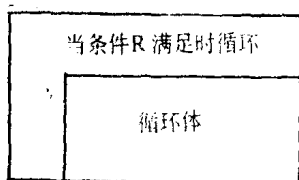
```
70 做事1 (之三)
```

```
80 做事3
```

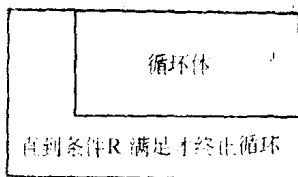
⋮

循环结构也有两种：一种是“当”型循环，一种是“直到”型循环。这两种结构略有不同，分别画在图6(a)和图6(b)中。“当”型的循环体框放在右下角；“直到”型放在左上角。

“当”的含意为：当条件R满足时就执行循环体，只有



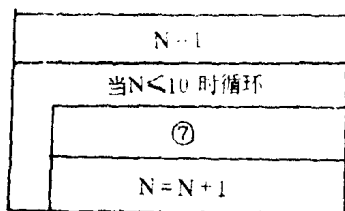
(a) “当”型循环



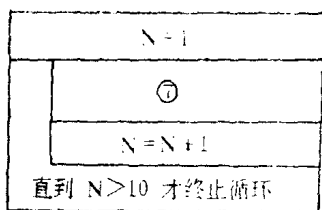
(b) “直到”型循环

图6

R条件得不到满足时才退出循环。仍用例1可将图3改画成图7(a)。在进入循环结构之前先将循环控制变量置一初值： $N=1$ ，然后再在原循环体⑦后增加一条将循环控制变量增值的语句： $N=N+1$ ，组成新的循环体。当 $N \leq 10$ 时重复执行新的循环体。在True BASIC中有专门的“当”语句(DO WHILE (条件) ...LOOP)，一般BASIC可以用



(a) “当”型



(b) “直到”型

图7

IF语句实现如下：

```

5  REM SAMPLE - 5
10 TOTAL = 0
20 MAX = 0
25 N = 1
40 INPUT "A = "; A
50 IF A > MAX THEN MAX = A

```

```

60  TOTAL = TOTAL + A
70  N = N + 1
75  IF N <= 10 THEN 40
80  AVERAGE = TOTAL / 10
90  PRINT "AVERAGE = "; AVERAGE
100 PRINT "MAX = "; MAX
110 END

```

“直到”型循环的“直到”二字的含意为：直到R条件满足时才退出循环，否则要重复执行循环体。例1中的循环如用“直到”型则将图3改为图7（b）。在True BASIC中有专门的“直到”语句（DO...LOOP UNTIL（条件）），一般BASIC也可用IF语句实现如下：

```

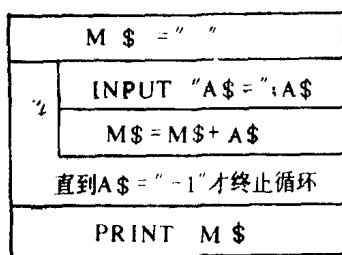
5   REM SAMPLE - 6
10  TOTAL = 0
20  MAX = 0
25  N = 1
30  IF N > 10 THEN 90
40  INPUT "A = "; A
50  IF A > MAX THEN MAX = A
60  TOTAL = TOTAL + A
70  N = N + 1
75  IF N <= 10 THEN 40
80  AVERAGE = TOTAL / 10
90  PRINT "AVERAGE = "; AVERAGE
100 PRINT "MAX = "; MAX
110 END

```

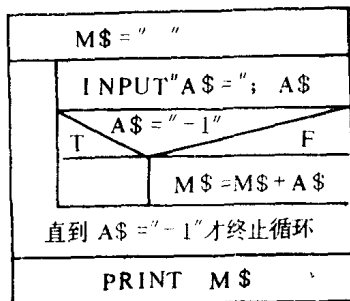
在多数情况下，“当”与“直到”型循环并不需要预先知道循环次数，或者说循环控制变量的取值可以不是循环次数，举例如下。

例2. 用键盘分若干次敲入字符或字符串，直到敲入标志“-1”为止，然后将敲入的全部字符显示出来(含“-1”)。

程序的流程图如图8 (a)。



(a) 显示字符串含“-1”



(b) 显示字符串不含“-1”

图8

程序如下：

```

5 REM SAMPLE-2
10 M$ = " "
15 INPUT "A$="; A$
20 M$ = M$ + A$
30 IF A$ <> "-1" THEN 15
40 PRINT M$
50 END

```

该题如果要求显示的全部字符中不含标志“-1”时，可将图8 (a) 改为图8 (b)。

程序如下：

```

5 REM SAMPLE-3

```

```

10 M$ = " "
20 INPUT "A$ = "; A$
30 IF A$ <> "-1" THEN
    M$ = M$ + A$ : GOTO 20
40 PRINT M$
50 END

```

程序中的30句既是分支结构中的判断条件，又是“直到”型循环的终止条件。

上面举的例子是“直到”型循环，下面举一个“当”型循环。

例3. 三尺之竿日截其半，请编程计算经多少天后竿长小于一寸。

程序框图如图9。

程序如下：

```

5  REM SAMPLE-8
10 L = 3 : N = 0
20 N = N + 1
30 L = L / 2
40 IF L >= 0.1 THEN 20
50 PRINT "N = "; N
60 END

```

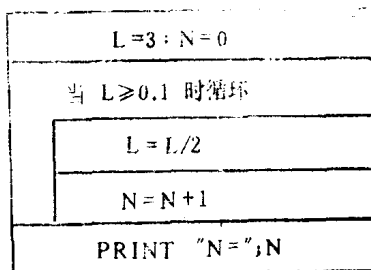


图9

在我们了解了“当”型和“直到”型循环的含意之后，很自然地可以想到，一般BASIC中的FOR...NEXT语句是用作计数循环的“当”型与“直到”型的结合。比如：

```

5  REM SAMPLE-9
10 FOR N = 1 TO 10

```

20 PRINT N

30 NEXT N

40 END

第10句与第30句合起来的作用可描述为：从 $N=1$ 开始循环，每循环一次循环控制变量 N 自动加1，“直到” $N>10$ 终止循环。或者说：“当” $1 \leq N \leq 10$ 时循环。FOR...NEXT给出了循环控制变量的初值、终值和步长，且循环控制量是自动增值（或减值）的，无需在循环体内安排增值（或减值）的语句，因此使用方便。不足之处是没有“当”和“直到”型循环那么灵活。在这里我们把FOR...NEXT看成是“当”与“直到”型循环结合的一种特定情况，是一种直接控制循环次数的情况。

除了顺序结构、分支结构和循环结构之外，还有一个用得较多的是多分支转子程序结构，如图10所示。在BASIC语言中用ON...GOSUB语句实现多分支转子程序结构。下面举一个应用多分支转子程序结构的例子。

据N的取值选择执行			
N=0	N=1	N=2	N=3
退出	子程序1	子程序2	子程序3

图10

例4. 任意输入10个数，然后由键盘控制选做下列三种运算或退出该程序。要求：

键入“1”：输出10个数的总和；

键入“2”：输出10个数的平均值；

键入“3”：输出10个数由大到小的排序结果；

键入“4”：退出。

这个题目可选用“菜单”技术，主程序框图如图11所示。先用循环结构输入10个任意数（图11上未详细画出，读者可自己加上），然后用了个“直到”型循环的结构。循环体的第一项是显示一个供读者选择的所谓“菜单”。菜单上面说明了选“1”是求总和；选“2”是求平均值；选“3”是排序；选“4”是退出（参见后面列出的程序）。接着是键盘输入语句，让读者选择自己要做的项目。之后是一个多分支转子程序结构。如键入“1”转求总和子程序；如键入“2”转求平均值子程序；如键入“3”转排序子程序；如键入“4”则退出。

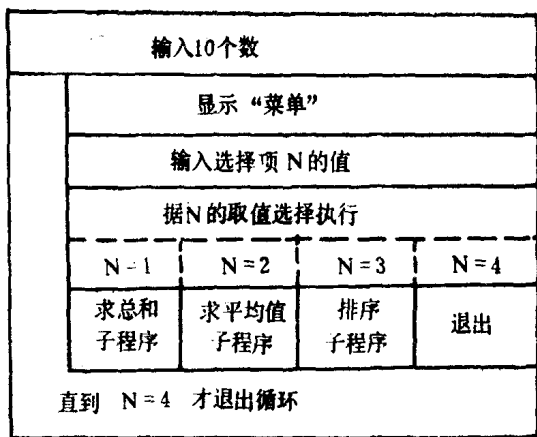


图11

图12画出了求10个数总和的子程序框图。

图13画出了求10个数平均值的子程序框图。

图14画出了用“起泡”法排序的子程序框图，这个框图

的主体是一个“直到”型循环结构。这里P是标志， $P=0$ 标志着数组中的数按降序排好，如果真的 $P=0$ 说明大功告成，则循环可以终止。这个“直到”型循环的循环体第一项是预置 $P=0$ ，意思是假设排好了顺序，然后再检验是否真的排好了，如真的排好了，P会始终保持0不变，否则P会由0变1。这件事由一个循环结构完成，让数组D(I)的相邻两个

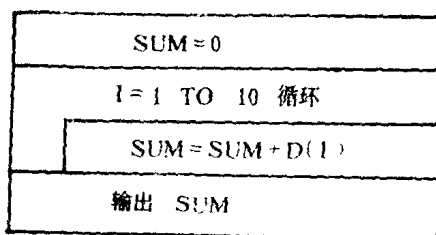


图12

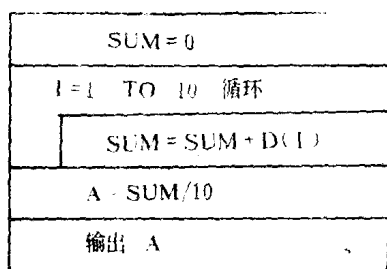


图13

元素比大小，如前一个小于后一个则两数交换，同时让标志 $P=1$ ，表示检验中发现了没有排好序的地方。这种比较共作9次，所以设计成I = 1 TO 9 循环。当这个内层循环作完之后，只要 $P=1$ ，外层循环（“直到”型循环）就不可能终止，还得进入循环体重新检验，直到完全排好为止。接下来