BASIC

BASIC
CHENGXU
SHEJI
SHITI
JINGXUAN

程序设计试题精选

罗海鹏 编

广西教育出版社

BASIC

程序设计试题精选

罗海鹏 编

广西教育出版社

BASLC程序设计试题精选
罗海鹏　编著
☆
广西教育出版社出版
南宁市七一路7号
广西新华书店发行　灵山县印刷厂印刷
*
开本787×1092　1/32　印张4.75　字数100,000
1988年8月第1版　1988年8月第1次印刷
印数：1—5320册　定价：1.50元
ISBN　7-5435-0365-4/G·308

前　言

学习计算机程序设计，在一定意义上说，就象学数学一样，如果只学概念，不做习题，是不容易领会深透的，所学的知识也不能够巩固。学习计算机程序设计，有时也象学作文一样，如果只懂得一些文法，不认真地去读一定数量的佳作范文，同时自己动手多写多练，也是不可能写出好的文章来的。关于计算机程序设计的书已经出版较多，但不少初入门的同志仍然希望多得到一些难易适度的编写程序的题目，以便自己练习，或者看看别人是怎样编写的。

这本小册子就是基于上面这些想法而编写的，材料大多取自于我们组织的几十个计算机培训班的试题和几次程序设计竞赛的试题。为照顾不同水平的读者，每道题都给出了解答并附有注释。

书中所有的程序均在APPLE Ⅱ机上通过，对LASER 310机和COMX PC机它们也是基本可用的。这些程序适用于中、小学计算机辅导教师、计算机课外小组，它可以作为青少年计算机程序设计竞赛的辅导材料，也可提供电视大学有关专业的学员和各行各业想掌握一些计算机知识的同志们参考。书中所给出的程序当然不一定是最好的，希望读者能够给出更好的程序。

苏德富副教授对全书的内容进行了仔细的审查并提出了

很好的意见，庞中坚老师在APPLE Ⅱ机上验证了所有的程序并订正了其中的错误，借此机会表示衷心感谢。

作　者

一九八七年八月

目　录

引　言

初学程序设计的同志，虽然已了解了各种语句的功能，背下了各种语句的使用规则，但编起具体的程序来，往往还是不知从何入手。他们掌握不住对一个具体的问题来说该选用哪些语句，该把这些语句怎样衔接起来，该怎样使这些语句的组合刚好能满足题目的要求。本书给出了较多的难易适度的例子，读者最好不要急于看答案，而首先自己动手，编出程序，上机通过。然后再和书中的答案进行比较，也许读者的程序是更好的程序；当读者实在编不出时，才去参考答案，再自己动手去尝试。这样实践多了，自己也逐渐地会编程序了。我们以两个具体的例子来引入，讲一讲编程序中常出现的一些问题。第一个例子是我们在多次考试中使用过的题目。

例一、在半径为2米的圆铁片上，分别以1.9米，1.8米，…，0.1米为半径，切割出20个圆环（其中最小的一个退缩为圆，也可以认为是一个特殊的圆环）。求每个圆环的面积。

这个题目并不难，但根据我们组织的几次考试统计，编程序完全正确的人还不到参加考试人数的1/3。很多人感到无从下手，还有很多人虽然编出了程序，但其中仍有各种各样的错误。一个正确的解答可以是这样的。

解：

用循环语句来做这个问题，循环执行20次。每次用相邻

的大圆面积减去小圆面积，则得到圆环的面积。程序中出现的变量意义如下：

R：圆的半径；

S1：大圆的面积；

S2：小圆的面积；

S：圆环的面积。

程序一：

```
10  S1=3.1416*2*2（第一个圆环的外圆面积）
20  FOR  R=1.9  TO  0  STEP  -0.1
30  S2=3.1416*R*R（圆环的内圆面积）
40  S=S1-S2（圆环面积）
50  PRINT  S,（输出这个圆环面积）
60  S1=S2（下一个圆环的外圆面积）
70  NEXT  R
80  END
```

我们用中文写出对程序的注解，放在语句之后的圆括号里，这样做的目的，是使这个程序更容易被理解。本书中所有的程序，我们都照此办理。当把程序输入计算机时，略去括号和括号中的中文注释。

这个问题在数学上是这样的：

大圆面积：$S1=\pi R^2$；

小圆面积：$S2=\pi(R-0.1)^2$；

圆环面积：$S=S1-S2$。

20次的使用上面的公式，每次把半径R减少0.1（开始时R=2），就可以求出20个圆环的面积了。可以省略的计算是每次所算出的小圆的面积，即是下一次的大圆面积。

有些人也能编写出类似的程序，但往往容易产生一些句法错误，比如：

• 漏写了 STEP－0.1；

• 在 STEP－0.1中，漏写了负号；

也有些人产生了一些简单的逻辑错误，比如：

• 把输出语句放在循环以外，这样就只能求得最后一个圆环了；

• 循环多做了一次，或者少做了一次。

有些人在编程序时也会出现一些较复杂的逻辑错误。对于上面这道题，一个逻辑上错误的程序如下：

程序二：

```
10  S=3.1416*2*2
20  FOR R=1.9 TO 0 STEP -0.1
30  S1=3.1416*R*R
40  S=S-S1
50  PRINT S,
60  NEXT R
70  END
```

这个程序想省掉一个变量 S2，用 S代表圆环面积并代表大圆面积，S1代表小圆面积。输出的第一个圆环的面积是正确的，但第二个，第三个，…，就都是不正确的了，因为圆环面积和大圆面积是不一样的。这样的错误比较隐蔽，有时较难查出。

有的人编出的程序也是正确的，但程序中有冗余的东西，因此显得烦琐了。下面就是一个这样的程序。

程序三：

```
10  FOR  R=2  TO  0  STEP  -0.1
20  S1=3.1416*R*R
30  S2=3.1416*(R-0.1)*(R-0.1)
40  S=S1-S2
50  PRINT  S,
60  NEXT  R
70  END
```

程序三完成了和程序一同样的工作，但在程序三中，做乘法：4×20=80次，做加法：3×20=60次。而在程序一中，做乘法：2+2×20=42次，做加法：20次。因此，程序三烦琐了。

有些人编程序时喜欢动不动就使用数组。这道题也有很多人用数组来编写，程序是类似这样的。

程序四：

```
10  DIM  S(20), R(20)
20  R(1)=2
30  S(1)=3.1416*R(1)*R(1)
40  FOR  I=2  TO  20
50  R(I)=R(I-1)-0.1
60  S(I)=3.1416*R(I)*R(I)
70  S=S(I-1)-S(I)
80  PRINT  S,
90  NEXT  I
100  PRINT  3.1416*0.1*0.1(最后一个圆环)
110  END
```

由于使用了数组元素，程序执行时要根据下标的值去寻

找每一个数组元素，速度就慢得多了。只有当若干个数必须同时参加某个运算时，才使用数组。如果是计算一项输出一项时，完全可以用简单变量代替数组元素，这样可提高程序执行的效率。

圆环的计算过程还可以进一步简化，数学推导如下：

$$S=\pi R^2-\pi(R-0.1)^2$$
$$=\pi(R+(R-0.1))(R-(R-0.1))$$
$$=\pi(2R-0.1)\times 0.1$$
$$=0.62832R-0.031416\ (\text{把}\pi\text{换成}3.1416)$$

根据这个公式可以编出下面的程序：

程序五：

```
10  FOR  R=2  TO  0.1  STEP  -0.1
20  S=0.62832*R-0.031416
30  PRINT  S,
40  NEXT  R
50  END
```

在程序五里，乘法只需做20次，加法也只需做20次，这是一个最简洁的程序了。但对这个程序如果不做详细的解释，就使人不容易看懂。故笔者认为，一般说还是用程序一好些。本书中的程序也大多是以当前程序编制的最重要的指标——“清晰”为出发点来考虑的。

读别人的程序时，最害怕碰到GOTO语句。一个并不太长的程序，如果有了几个GOTO语句，也常常会给理解它带来很多困难。所以我们在编程序时，应避免过多地使用GOTO语句。用一个例子来详述这一问题。

例二、由键盘输入A，B，C三个数，编一个程序，把

这三个数按其值的大小顺序打印出来。

解：

在下面的程序一中增加了变量D、E，D装A、B中的较大数，E装A、B中的较小数。

程序一：

```
10  INPUT A, B, C
20  IF A<B THEN 60 (对COMX PC机来说，THEN后的GOTO不能省略)
30  D=A  (把大数送到D中)
40  E=B  (把小数送到E中)
50  GOTO 80
60  D=B  (把大数送到D中)
70  E=A  (把小数送到E中)
80  IF D>C THEN 110
90  PRINT C, D, E  (从大到小打印出这三个数)
100 GOTO 180
110 PRINT D, (此时D是最大数)
120 IF C>E THEN 160 (两个小的再相比)
130 PRINT E, (打印出第二大的数)
140 PRINT C  (打印出最小的数)
150 GOTO 180
160 PRINT C, (打印出第二大的数)
170 PRINT E  (打印出最小的数)
180 END
```

这个程序共使用了6个GOTO语句，使得整个程序在

结构上显得不够清晰，读起来有一定的困难。其实，这个程序完全可以编得简练一些。

程序二：

```
10  INPUT  A, B, C
    (20行～30行，用A装A、B中较大的数，用B装A、B中较小的数)
20  IF  A >= B  THEN  40
30  T = A: A = B: B = T   (T是用于A、B交换的临时工作单元)
    (40行～50行，用A装A、C中较大的数，用C装A、C中较小的数)
40  IF  A >= C  THEN  60
50  T = A: A = C: C = T
    (60行～70行，用B装B、C中较大的数，用C装B、C中较小的数)
60  IF  B >= C  THEN  80
70  T = B: B = C: C = T
80  PRINT  A, B, C
90  END
```

这个程序仅用了3个GOTO语句，这些GOTO语句都是向下转，没有转回头的，而且GOTO的范围都很小，故在结构上还是比较清晰的。对这个问题来说，程序再稍微变动一下，可以完全避开GOTO语句。

程序三：

```
10  INPUT  A, B, C
20  IF  A < B  THEN  T = A: A = B: B = T
```

（A装A、B中较大的数，B装A、B中较小的数）

30 IF A<C THEN T=A:A=C:C=T

（A装A、C中较大的数，C装A、C中较小的数，这时A已经是A、B、C中的最大数）

40 IF B<C THEN T=B:B=C:C=T

（B装B、C中较大的数，C装B、C中较小的数）

50 PRINT A，B，C （输出大、中、小三数）

60 END

本书中的程序，都希望尽量少用GOTO语句；如果用了GOTO语句，也希望尽量使它的跳动范围较小，仍能基本保持程序的清晰可读。

1、相差最小的素数

由键盘输入正整数M，N（M<N），求M，N之间的所有素数中相差最小的两个。

解：

判断I是不是素数，可用2去除它，3去除它，…，一直除到I-1，如果都除不尽则I是素数。好一点的方法是一直除到I/2，更好的方法是一直除到SQR（I）。I不可能有超过SQR（I）的因子了，因为如果有，则除得的商必比SQR（I）要小，那么在除到SQR（I）之前应该已经用这个数试除过I，而把I判断为合数了。

程序清单：

10 INPUT M，N

```
20  P1=0:P2=0 (P1、P2将装相邻的两个素数)
30  FOR I=M TO N
40  IF I=1 THEN 130 (1不是素数，让它过去)
50  IF I=2 THEN 90 (2是素数，记下来)
60  FOR J=2 TO SQR(I)+1 (60行～80行，判断I是不是素数)
70  IF I/J=INT(I/J) THEN 130 (在COMX PC机上不能省略这里的GOTO)
80  NEXT J
90  IF P1=0 THEN P1=I:Q=P1:GOTO 130
100 IF P2=0 THEN P2=I:R=P2:T=P2-P1:GOTO 130
    (T是两个相邻素数之差，Q、R是相差最小的一对素数)
110 P1=P2:P2=I
120 IF P2-P1<T THEN T=P2-P1:Q=P1:R=P2
130 NEXT I
140 PRINT Q, R
150 END
RUN
? 2000, 3000
2027        2029
```

如果相差最小的素数不只一对，则我们仅求出了第一