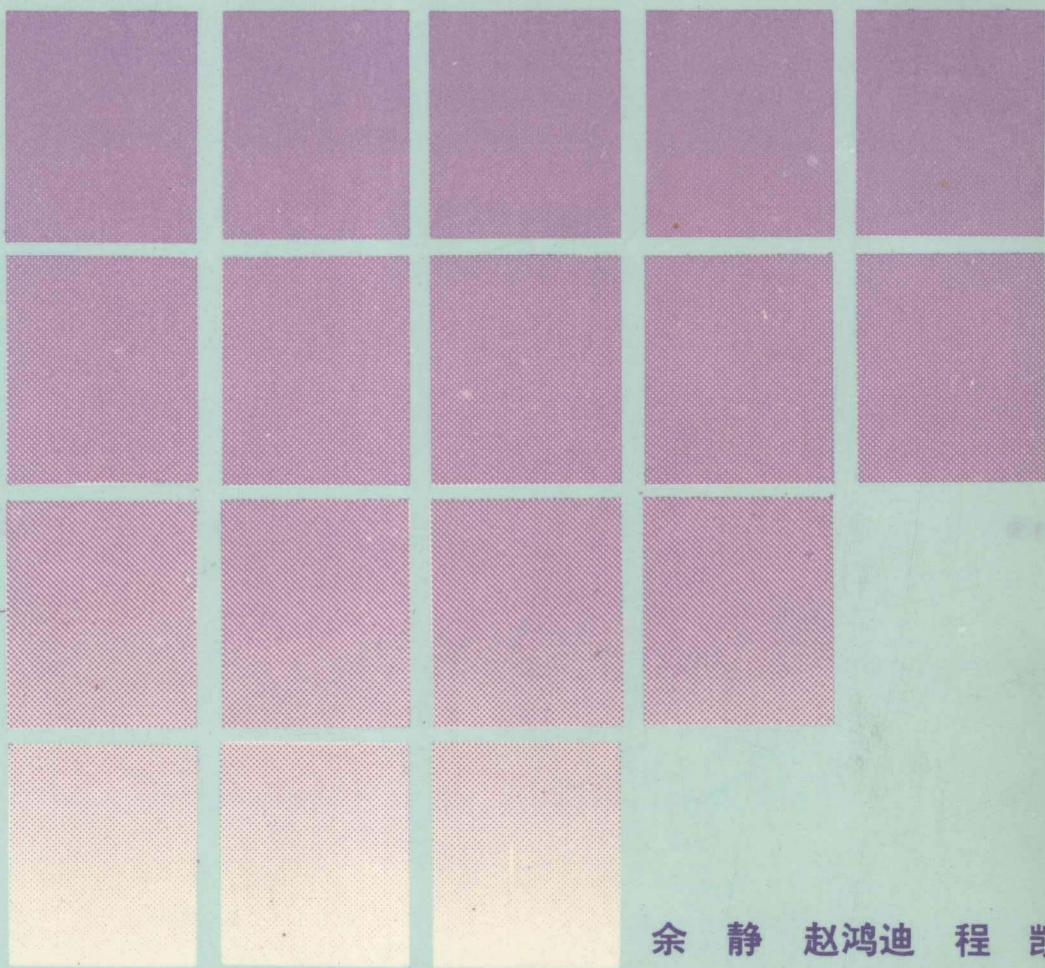


True BASIC

结构程序设计教程 (下册)



余 静 赵鸿迪 程 凯

东北工学院出版社

号 8 第字登录 (五)

True BASIC

结构程序设计教程

(下册)

余 静 赵鸿迪 程 凯 编著



1985年6月第1版
1985年6月第1次印刷
1001—3200册
ISBN 7-81008-578-9/TP·13
定价：1.50元

东北工学院出版社

(辽)新登字第8号

True BASIC

内 容 简 介

True BASIC 是 BASIC 创始人总结 BASIC 语言使用经验,吸取其他高级语言的优点,而发展的一种新型通用的结构式程序设计语言。对于有 BASIC 语言知识的读者,通过本书学习,可掌握结构式程序设计的内容和方法,反之,亦很容易掌握 BASIC 语言。

本书分两部分出版,互相联系又独立成册。第一部分以数值计算和事务管理为中心,介绍 True BASIC 基本语句与命令,掌握程序设计的基本内容;第二部分以图形语句和外设联接为实例,掌握图形设计技巧,完成图形软件设计和程序的编制。

本书内容由浅入深,简明扼要,适于作为大专院校教材,也可作为研究生和工程技术人员参考用书。

True BASIC
结构程序设计教程
(下 册)

余 静 赵鸿迪 程 凯 编著

东北工学院出版社出版发行 大连海运学院出版社印刷厂印刷
(沈阳·南湖) (大连·凌水桥)

开本: 787×1092 1/16 印张: 11 字数: 275 千字
1992年6月第1版 1993年3月第2次印刷
印数: 1001~2500 册

责任编辑: 王金邦 责任校对: 张德喜
封面设计: 唐敏智 版式设计: 秦 力

ISBN 7-81006-278-6/TP·12 定价: 7.60 元

大连理工大学出版社

序 言

BASIC 语言是程序设计语言中简单、易学、受用户欢迎的语言。20 世纪 70 年代以来,迅速发展的微型计算机,全部配有这种语言,使得 BASIC 成为世界上应用最广泛的一种计算机语言。但在利用内存空间、编辑功能以及实现程序结构化方面比不上 PASCAL 和 C 语言。BASIC 语言创始人 John G. Kemeny 和 Thomas E. Kurtz 总结了 BASIC 语言使用经验,吸取了其他高级语言的优点,引入了结构化程序设计的思想,对原有的 BASIC 语言作了重大修改,于 1985 年 11 月公布了新版本的 BASIC 语言,并命名为“True BASIC”(真正的 BASIC)。

True BASIC 保留了 BASIC 简单易学,使用方便的基本特点,删除了其中陈旧过时的内容,增加了编辑功能,无论对初学者还是有经验的程序设计人员,都有很强的吸引力,它的主要特点是:

1. 结构化程序设计语言

随着计算机科学的发展,“结构化程序设计”以其组织清晰,易编制,易阅读,普遍为软件工作者所接受。在较复杂的程序设计中,使用 BASIC 中的 GOTO, GOSUB, ON-GOTO 等转向语句,势必使程序执行走向紊乱,结构缠绕不清。这是 BASIC 语言不适于编制较为复杂软件的主要原因之一。True BASIC 语言可以不用 GOTO 语句,取消行号,同样完成程序中的各种转向;以内部函数和内部子程序作为模块化程序设计的辅助结构,使程序清晰易读。引用局部变量的外部函数和外部子程序易于实现模块化结构的程序设计。库文件和 CHAIN 语句又为大型程序的分解提供了方便。

2. 充分利用内存空间

BASIC 语言是按 64K 内存设计的。在普遍采用 640K 内存的微机上,它只用了十分之一内存,这是 BASIC 难于编制多功能大程序的原因之一。True BASIC 采用绝对地址的方法,允许使用全部内存,一般它可以编制 300K 以上的应用程序。

3. 强化的编辑功能

True BASIC 的屏幕编辑可利用功能键对整行内容、整个模块进行删除、复制和移动;也可利用专用命令(LOCATE, CHANGE, TRY 等)对程序中字符串进行显示、审核和修改。

4. 极强的图形功能

BASIC 语言实现图形缩放、移动和旋转,需要在数学公式的相对坐标上加相应的数值,所以编辑一段程序其工作量比重新绘制这个图形还多;而 True BASIC 简单的多,只要把原图形后边加上简单参数,就可实现图形缩放、平移和旋转。True BASIC 设有功能键,极容易实现图形的着色,还允许将屏幕分成几个独立的窗口,分别打开各窗口,在里面作图和书写文字。

5. 良好的通用性

True BASIC 是严格按照美国国家标准编写的,具有良好的可移植性。它还保留了与旧 BASIC 语言的兼容性,只要略加修改就可将 BASIC 程序转成 True BASIC 程序。

6. 解释和编译相融

大多数 BASIC 是解释性的,它具有从较短程序中立即获得运行结果的优点,但是对较大

程序运行时间长。如果为加快运行速度对 BASIC 进行编译,操作比较麻烦(退出 BASIC 状态,在 DOS 状态下用 BASCOM 和 LINK 软件进行编译),而且还要产生一系列中间文件(如目标文件),要占用磁盘空间。而 True BASIC 兼有二者的优点,可以先用解释程序进行调试,程序调好后,再用一条简单编译命令(COMPILE),立即将解释程序变为编译程序,从而提高了程序运行速度并增强了保密性。

此外,与 BASIC 语言比较,还有一些优点,如它有矩阵运算功能,数组量可比 BASIC 大约 4.4 倍等。但是从用户使用角度看,仍不够理想,它没有 BASIC 所设置打开通讯口的 OPEN "COM....."语句,因此在与绘图仪、数字化仪联接方面不如 BASIC 那样简便。

本书是为大学生在 IBM-PC 微机上学习 True BASIC 语言而编写的教材。全书共十四章分两部分出版,相互联系又独立成册。前七章以科学计算和数据处理为实例,介绍 True BASIC 常用语句和基本命令,掌握程序设计的基本知识,可用 30~40 学时讲授及上机操作。后七章以图形处理和外设通讯为实例,介绍了各种图形语句的使用和外部设备的联接,掌握图形设计技巧,完成图形软件的设计与程序编制,可用 30~40 学时讲授及上机操作。此部分可作为选修内容。

参加本书编写的有余静(一、七、十一、十二、十三、十四章)、赵鸿迪(四、五、六、八、九、十章)和程凯(二、三章),全书由余静审定。尤克、冯慎明和代丽燕参加了书中实例程序的调试。由于编著者水平有限,书中难免存在错误和不当之处,诚恳希望读者批评指正。

编者

1991 年 1 月

目 录

第八章 字符串应用和数据的输入、输出	1
第一节 字符串表达式及运算	1
一、字符串表达式	1
二、字符串比较	2
三、子字符串	3
第二节 字符串函数的应用	6
一、统一英文字母的大、小写	6
二、字符串自身连接	7
三、字符串的测定函数	7
四、字符串转换函数	8
第三节 格式化输入和输出	9
一、屏幕输出区域	9
二、光标	11
三、自定义格式输入、输出	12
第四节 应用举例	17
小 结	42
复习题	42
第九章 数组与矩阵运算	45
第一节 数组的输入、输出	45
一、MAT READ 语句	45
二、MAT PRINT 语句	46
三、MAT INPUT 语句	48
第二节 数组赋值与运算	51
一、MAT 赋值语句	51
二、MAT 加、减语句	54
三、MAT 乘语句	56
第三节 数组和矩阵函数	59
一、数组函数	59
二、矩阵函数	64
三、数组测定函数	67
第四节 应用举例	70
小 结	80
复习题	80

第十章 声响和音乐	83
第一节 PLAY 语句	83
一、格式和功能	83
二、举 例	85
第二节 SOUND 语句	88
一、格式和功能	88
二、举 例	88
小 结	92
复习题	92
第十一章 图 形	94
第一节 窗口和坐标设置语句	94
第二节 着色语句	96
第三节 图形语句	98
第四节 图形变换语句	102
第五节 动画语句	104
第六节 图形输入语句	105
小 结	106
复习题	106
第十二章 汇编语言与外设接口	107
第一节 用汇编语言写函数	107
第二节 用汇编语言写子程序	110
第三节 微机与绘图机的联接	112
一、利用打字机口作为绘图机通讯口	112
二、用汇编语言打开 RS-232C 通讯口传递绘图指令	114
第四节 与数字化仪的联接	117
小 结	121
复习题	121
第十三章 程序设计	123
第一节 程序的基本结构	123
一、程序设计追求的目标	123
二、True BASIC 语言的基本结构	123
三、结构化程序的构成	124
第二节 程序的逐步求精	124
第三节 模块在结构化程序设计中的运用	131
小 结	132

复习题.....	133
第十四章 程序调试与出错处理	134
第一节 程序的调试.....	134
第二节 出错处理.....	135
小 结.....	139
复习题.....	139
附 录	140
附录 1 命令语句索引	140
附录 2 库	140
附录 3 书写 DO 命令的程序	145
附录 4 OPTION 语句.....	147
附录 5 查看和修改存储器字节内容与测试内存空间	147
附录 6 压缩子程序	148
附录 7 True BASIC 2.03 版的改进	150
附录 8 从 BASIC 到 True BASIC 的转换软件	158
附录 9 EXE 文件生成软件——汇集程序	161
附录 10 True BASIC 与 DOS 系统的接口方法	163
附录 11 True BASIC 与其他执行程序 (.EXE 文件) 的接口方法	164
参考文献	166

第八章 字符串应用和数据的输入、输出

在计算机管理信息的过程中,字符型信息的存储、编辑、处理等有着多方面的应用。True BASIC 为字符串处理提供了多种函数。本章将讨论字符串表达式及字符串函数在程序设计中的应用。

在上册已经广泛使用了 True BASIC 所提供的标准输入和输出格式。本章将讨论屏幕控制及格式化输入、输出的语句,以满足程序设计中各种不同的需要。

第一节 字符串表达式及运算

一、字符串表达式

字符串表达式的运算符“&”,又叫连接符。

字符串表达式是由字符型常量(字符串)、字符型变量(简单变量或下标变量)、字符型函数等,用连接符“&”连接组成的表达式,运算结果是一个字符串。一个字符串、一个字符型函数,一个字符型变量,都是字符型表达式的简单形式。例如:

源程序及运行结果:

```
LET a$="abcd"
```

```
LET b$="efg"
```

```
LET c$=a$&b$
```

```
LET d$=c$&" "&a$&"**"&b$
```

```
PRINT "a$:";a$
```

```
PRINT "b$:";b$
```

```
PRINT "c$:";c$
```

```
PRINT "d$:";d$
```

```
END
```

```
RUN
```

```
a$:abcd
```

```
b$:efg
```

```
c$:abcdefg
```

```
d$:abcdefg abcd**efg
```

其中:

“abcd”,“efg”是字符串,即字符型常量;a\$,b\$,c\$,d\$等是字符型变量。它们都可以视为字符型表达式的简单形式。

由连接符组成的C\$&" "&a\$&"**"&b\$则是字符串表达式。

例如:输入姓名和年龄,再将“姓名:”和“年龄:”的中间插入四个空格,组成新的字符串输

出。

源程序及运行结果：

```
SET MODE "hires" ! 设汉字状态
```

```
OPEN #2:printer
```

```
FOR i=1 to 4
```

```
INPUT prompt "姓名:";name$
```

```
LINE INPUT prompt "年龄:";lag$
```

```
LET a$="姓名:"&name$&" "&"年龄:"&lag$
```

```
PRINT #2;a$
```

```
NEXT i
```

```
END
```

```
RUN
```

```
姓名:王 萧 年龄:28
```

```
姓名:李丽文 年龄:36
```

```
姓名:张文文 年龄:23
```

```
姓名:卢 准 年龄:45
```

二、字符串比较

两个字符串可以比较大小,比较的方法:从左向右,依次逐个比较每个字符的 ASCII 码值。如果两个字符串相等,必须是两个字符串中的字符个数相等,排列顺序一一对应。如果两个字符串不同,则按如下原则决定其大小:

1. 从左而右的比较中,一旦发现不等的字符,则认为 ASCII 码大的字符所属串为大,而不取决于字符串的长短。

例如:A 的 ASCII 码是 41,B 的 ASCII 码是 42,a 的 ASCII 码是 61,b 的 ASCII 码是 62,则"a">"AB",因为 a 的 ASCII 码大于 A 的 ASCII 码。

2. 虽然字符相同,但排列顺序不同时,两个字符串也不相等。例如:"AB"<"BA"。

3. 字符相同,排列顺序也相同时,则有后续字符的长串大于短串。如:"abAB"<"abABa**"。

例如:有一组英文书写的国家名字,按字母顺序排列打印,即按其 ASCII 码由小到大排列。

排序的方法很多,下面程序中是采用了冒泡排序法(交换排序)。即每次对两个相邻的数据比较,若不满足顺序要求则交换,否则不交换。

源程序及运行结果

```
OPEN #2:printer
```

```
DIM a$(7)
```

```
FOR i=1 to 7
```

```
READ a$(i)
```

```
PRINT #2;a$(i)&" "
```

```
NEXT i
```

```

PRINT #2:" "
PRINT #2:"-----"
FOR i=1 to 6
    FOR j=i+1 to 7
        IF a$(i)>a$(j) then
            LET s$=a$(i)
            LET a$(i)=a$(j)
            LET a$(j)=s$
        END IF
    NEXT j
NEXT i
FOR j=1 to 7
    PRINT #2;a$(j)&" ";
NEXT j
PRINT #2:" "
DATA "CHINA","FRANCE","KOREA","JAPAN","INDIA"
DATA "ENGLAND","CANADA"
END
RUN
CHINA FRANCE KOREA JAPAN INDIA ENGLAND CANADA
CANADA CHINA ENGLAND FRANCE INDIA JAPAN KOREA

```

三、子字符串

一个字符串的一部分,称为该字符串的子字符串或子串。可以对字符串的局部进行操作,得到一个新的字符串。

1. 提取子字符串。即从一个字符串中,取其中的一部分。
 a(m:n)$ (或 a(m;n)$)表示提取 $a$$ 中第 m 个字符开始到第 n 个字符为止的子串。

例如: a="ABCDEF"$,则 a(2:3)="BC"$, a(4:6)="DEF"$, a(2:2)="B"$

例如:输出由字母"ABCDEFGH"组成的等腰三角形。

源程序及运行结果:

```

OPEN #1:printer
LET a$="ABCDEFGH"
PRINT #1: tab(len(a$)+1);" *"
FOR i=1 to len(a$)
    LET c$=a$(1:i)
    CALL ch(c$,b$)
    PRINT #1: tab(1+len(a$)-i);b$;
    PRINT #1: a$(1:i)

```

```

NEXT i
END
SUB ch(a$,b$)
  LET b$=""
  LET kn=len(a$)
  FOR i=kn to 1 step -1
    LET b$=b$ & a$(i,i)
  NEXT i
  LET b$=b$ & a$(1:1)
END SUB
RUN

```

```

*
AAA
BAAAB
CBAAABC
DCBAAABCD
EDCBAAABCDE
FEDCBAAABCDEF
GFEDCBAAABCDEF
HGFCBAAABCDEF

```

2. 插入子字符串。可以把某个字符串作为子字符串插入到另一个字符串之首、尾、中间。也可以替换某几个字符。例如：

源程序及运行结果

```

OPEN #1:printer
LET a$="ABCDEF"
LET b$="ABCDEF"
LET c$="ABCDEF"
PRINT #1: "a$ : ";a$
LET a$(1:0)="abc"           ! 将"abc"插入 a$ 之首
LET b$(maxnum;0)="abc"      ! 将"abc"插入 b$ 之尾
LET c$(4:0)="abc"           ! 将"abc"插入 c$, 从其第 4 个字符位开始
PRINT #1: "-----"
PRINT #1: "a$ : ";a$
PRINT #1: "b$ : ";b$
PRINT #1: "c$ : ";c$
END
RUN
a$ : ABCDEF
-----

```

a \$: abcABCDEF

b \$: ABCDEFabc

c \$: ABCabcDEF

3. 修改字符串。利用子字符串插入的方法对某个字符串修改。如替换、删除等。例如：
源程序及运行结果

```
OPEN #1:printer
```

```
LET a $ ="ABCDEF"
```

```
LET b $ ="ABCDEF"
```

```
LET c $ ="ABCDEF"
```

```
LET d $ ="ABCDEF"
```

```
LET b $ (2;4) ="abc"
```

! 用"abc"替换 b \$ 的第 2、3、4 个字符

```
LET c $ (2;4) = ""
```

! 删除 c \$ 的第 2 至 4 个字符

```
LET d $ (2;3) = "   "
```

! 用 3 个空格替换 d \$ 的第 2、第 3 个字符

```
PRINT #1: "-----"
```

```
PRINT #1: "a $ : " ; a $
```

```
PRINT #1: "b $ : " ; b $
```

```
PRINT #1: "c $ : " ; c $
```

```
PRINT #1: "d $ : " ; d $
```

```
END
```

```
RUN
```

```
-----
```

```
a $ : ABCDEF
```

```
b $ : AabcEF
```

```
c $ : AEF
```

```
d $ : A   DEF
```

又如,输入一个英语句子,将其分解为单词并按每行一个单词输出。

源程序及运行结果:

```
SET MODE "hires
```

```
LET word $ = ""
```

```
INPUT prompt "输入一个英语短句:" ; test $
```

```
LET n=len(test $)
```

! len(test \$)测定 test \$ 串的长度函数

```
FOR i=1 to n
```

```
IF test $ (i;i) <> " " then
```

```
LET word $ (maxnum;0)=test $ (i;i) ! 其中,maxnum=len(word $)
```

```
ELSE
```

```
PRINT word $
```

```
LET word $ = ""
```

```
END IF
```

```
NEXT i
```

```
PRINT word $
```

```
END
```

```
RUN
```

输入一个英语短句:The computer can read English

The

computer

can

read

English

第二节 字符串函数的应用

在第五章的标准函数中,列出了字符串函数及功能。灵活地使用这些标准函数,可以进行多种多样的字符串处理。

一、统一英文字母的大、小写

lcase \$(a \$)函数,把a \$中的大写英文字母改为小写英文字母,其它符号不变。ucase \$(a \$)函数,把a \$中的小写英文字母改写为大写英文字母,其它符号不变。

如:lcase \$(“ABCdeFg12”) = “abcdefg12”

ucase \$(“ABCdeFg12”) = “ABCDEFG12”

用于英文字母输入时,控制程序运行或字符串比较,将是很方便的。例如:

```
LET a $ = “y”
```

```
DO WHILE p $ = “y”
```

```
INPUT “继续吗? (y/n):” a $
```

```
IF lcase $(a $) = “n” THEN
```

```
EXIT DO
```

```
ELSE
```

```
:
```

```
END IF
```

```
LOOP
```

其中 IF lcase \$(a \$) = “n”也可以换为:IF ucase \$(a \$) = “N”,效果相同。这样,操作时不必考虑输入字符的大小写。

其中,表达式:

lcase \$(a \$) = “n”

ucase \$(a \$) = “N”

各自等价于:

a \$ = “n” OR a \$ = “N”

二、字符串自身连接

repeat \$(a\$, n)由n个a\$组成一个长串。例如:

```
LET x$=repeat$("COM",3)
```

```
LET a$="COM"
```

```
LET y$=repeat$(a$,3)
```

x\$和y\$的值均为“COMCOMCOM”。

例如:输入任一个符号和一个数字n,将输出一个由n个该符号组成的正方形,其行数和列数为n。

```
INPUT prompt"图形符号:":a$
```

```
INPUT prompt"图形边长:":n
```

```
FOR i=1 TO n
```

```
PRINT repeat$(a$,n)
```

```
NEXT i
```

```
END
```

程序中的PRINT repeat(a\$,n)等价于:

```
FOR j=1 TO n
```

```
PRINT a$;
```

```
NEXT j
```

```
PRINT
```

三、字符串的测定函数

对字符串的组成进行测定,可以得到一个数值。

len(a\$)测定a\$的长度,即a\$中字符总个数。例如:

```
LET a$="good morning"
```

```
LET x=len(a$)
```

```
LET y=len("good morning")
```

x和y的值均为12。

pos(a\$,b\$)查取a\$中与b\$相匹配的第一个字符位置序位值。例如:

```
LET a$="write"
```

```
LET b$="t"
```

```
LET x=pos("write",b$)
```

```
LET y=pos(a$,"t")
```

x和y的值都是4。

pos(a\$,b\$,n)与pos(a\$,b\$)函数功能相似,只是查取时从a\$的第n个字符开始。例如:

```
LET t$="True BASIC"
```

```
LET e$="ue"
```

```
LET f$="I"
```

```
LET t1=pos(t$,e$,1)
```

```
LET t2=pos(t$,f$,5)
```

```
LET t3=pos(t$,e$,5)
```

结果,t1 的值是 3,t2 的值是 5,t3 的值是 0,因为 t\$ 的第 5 个字符以后,再找不到“ue”串。其中,pos(t\$,e\$,1)等价于 pos(t\$,e\$),都是从第一个字符开始依次查找。

四、字符串转换函数

根据不同的需要,可以用转换函数进行字符串与数字及 ASCII 码之间的转换。

1. 字符串与数字转换

str\$(a)与 val(a\$)

val(a\$)把不含空格,由数符组成的串 a\$ 转换为与 a\$ 形状相同的数值。其中,可以有一个小数点,第一个字符可以是正、负号。

源程序及运行结果:

```
OPEN #1:printer
LET a$="-123"&"234"
LET a=val(a$)
LET b=val("-123.05"&"86")
LET c=val("1.986e+2")
PRINT #1:a,b,val("123.05"&"85"),c
RUN
-123.234 -123.059 123.058 198.6
```

str\$(a) 把数值 a 原形不变地转换为字符串数据,a,必须是数字型的。它是 val(a\$)的反函数。

例如:

源程序及运行结果:

```
OPEN #1:printer
LET a=-123+0.234
LET a$=str$(a)
LET b$=str$(-123.05+86)
LET c$=str$(1.986e+2)
PRINT #1:a$,b$,str$(123.05+85),c$
END
RUN
-122.766 -37.05 208.05 198.6
```

2. num\$(a)与 num(a\$)

num\$(a)把数 a 转换为由 IEEE8-字节表示法表示的字符串。该字节表示法在所有机器上都是一样的,共有 8 个字符。符号占一位,指数占 11 位,数据占 52 位,精度为 15 位。num\$(a)把数字 a 转换成一个字符串,常用于记录文件的书写。但转换的字符串不能直接阅读。

num(a\$)把一个 IEEE8-字节表示的字符串转换为可以阅读的数字,即原来数的形式。它

与 num\$(a)配合应用于记录文件的建立和调用中。

3. ord(a\$)与 chr\$(a)

ord(a\$)取 a\$ 串的 ASCII 码序值,其中 a\$ 只能是一位字符。例如:

源程序及运行结果:

```
OPEN #1:printer
PRINT #1:ord("a")
PRINT #1:ord("A")
PRINT #1:ord(" * ")
PRINT #1:ord("3")
PRINT #1:ord("~")
PRINT #1:ord("!")
END
RUN
```

97

65

42

51

94

33

chr\$(a) 当 $0 \leq a \leq 127$ 时,给出 ASCII 码 a 所对应的字符(见附录);当 $128 \leq a \leq 255$ 时,不同的机器给出不同的 ASCII 码值。当 $a > 255$ 时,采用 $\text{mod}(a, 256)$ 的方法,换算为 $0 \sim 255$ 范围内的数,再给出该 ASCII 码值所对应的字符。例如:

源程序及运行结果:

```
OPEN #1:printer
PRINT #1:chr$(97)
PRINT #1:chr$(65)
PRINT #1:chr$(30+35+255+1)
END
RUN
```

a

A

A

第三节 格式化输入和输出

本节讨论有关输出的控制语句和格式输出。

一、屏幕输出区域

在 PRINT 语句中,系统默认屏幕输出的每项数据长度占 16 位。屏幕的边界宽度为 80