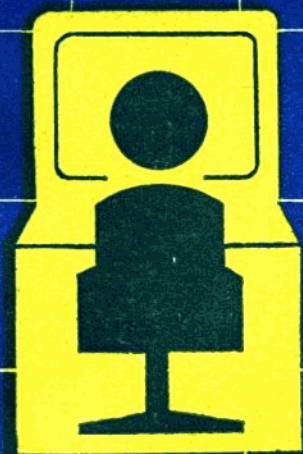
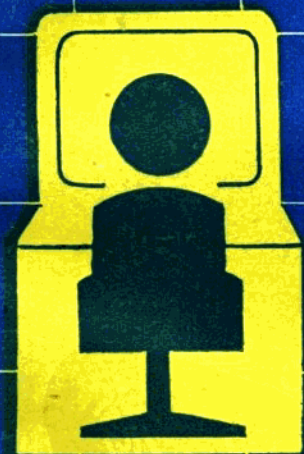


小学电算机知识 ③

xiaoxuediansuananjizhishi



青 岛 出 版 社

小学电算机知识

第 三 册

青 岛 市 教 育 局 编

青 岛 出 版 社

小学计算机知识
第三册

青岛市教育局编

*

青岛出版社出版发行

青岛印刷厂印刷

*

787×1092毫米 32开本 2印张 43千字

1987年7月第1版 1988年6月第2次印刷

印数 15901—29600

ISBN 7-5436-0037-4

G·28 定价：0.55元

前 言

科学的迅速发展，把电子计算机——科学技术知识与智慧的结晶——推进了少年儿童的生活领域，普及电子计算机“从娃娃抓起”已经成为一种现实需要。为此我们编写了一套《小学生电算机知识》，作为小学生电子计算机活动课试用教材。

第三册分15课，继续学习BASIC语言。各课既有侧重又保持连贯，力求做到知识性与趣味性相结合。

每课教学约为一课时，教学时应根据教材内容安排参观或实习。学完第三册需要一个学期。本书由井立才、孙寿铤同志编写。

由于经验不足，难免有些错误与缺点，欢迎提出宝贵意见，以便修改。

青岛市教育局

1987年5月

目 录

第一课	转子语句(GOSUB)和返回语句(RETURN)···	(1)
第二课	二维数组用途大·····	(4)
第三课	字符串函数(1)·····	(7)
第四课	字符串函数(2)·····	(11)
第五课	控制转向语句(ON...GOTO···)	(14)
第六课	控制转子语句(ON...GOSUB···)	(18)
第七课	自选打印格式语句(PRINT USING)·····	(21)
第八课	循环中的循环·····	(23)
第九课	计算机还是一位演奏家·····	(26)
第十课	控制字符、图形运动速度的游戏·····	(30)
第十一课	猜字母游戏·····	(32)
第十二课	打印机的使用·····	(34)
第十三课	打印月历·····	(37)
第十四课	录音机的使用·····	(39)
第十五课	怎样使用软盘驱动器·····	(42)
附加练习题	·····	(45)
练习答案	·····	(54)
附录一	ASCII代码表·····	(57)
附录二	BASIC语句及键盘命令一览表·····	(59)
附录三	键盘命令·····	(60)
附录四	使用计算机注意事项·····	(60)

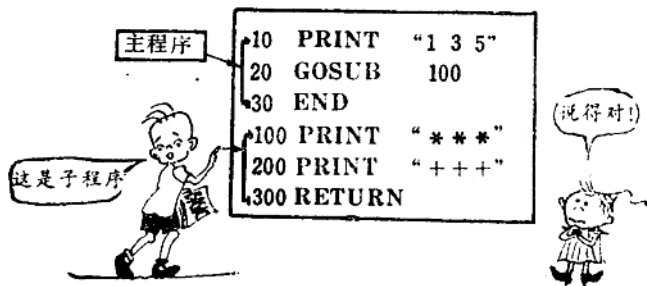
第一课 转子语句(GOSUB)和 返回语句(RETURN)

什么叫子程序？什么叫主程序？

有时，我们会遇到这样一种情况：需要对某一种运算进行多次重复。这样，在编制程序时便要书写多次，既麻烦又容易出错。为此，我们把需要重复运算的部分编成一段程序，这段程序就称为子程序。在整个程序运行过程中，当需要使用这段子程序时，就去调用它，非常方便。我们把调用子程序的程序称为主程序。

转子语句的意思是转去执行子程序，在程序中用GOSUB表示，读作：gōu sa bē。子程序的最后一行，称为返回语句，意思是返回主程序，用RETURN表示，读作：rǔi tèn。
一般格式：

行号	GOSUB	行号
行号	RETURN	



例题 1

```
主程序 { 10 PRINT "1 2 3 4 5"  
        20 GOSUB 100 (这是转子语句)  
        30 PRINT "6 7 8 9 10"  
        40 GOSUB 100  
        50 END  
子程序 { 100 PRINT "*****"  
        200 PRINT "—————"  
        300 RETURN (这是返回语句)
```

这个程序中，第10行到第50行是主程序，100行到300行是子程序。第20行是转去调用子程序的转子语句，300行是使程序从子程序返回到主程序的第30行执行的返回语句，40行又是一个转去调用子程序的转子语句，执行完子程序又返回50行，程序结束。

请看上面程序的运行：

RUN

1 2 3 4 5

—————

6 7 8 9 10

—————

例题 2

主程序	100	FOR	X = 1	TO	5
	200	GOSUB	600		
	300	PRINT	X; S		
	400	NEXT	X		
	500	END			
子程序	600	S = 0			
	700	FOR	J = 1	TO	X
	800	S = S + J			
	900	NEXT	J		
	1000	RETURN			
	RUN				
	1		1		
	2		3		
	3		6		
	4		10		
	5		15		

程序中第100行到500行是主程序，600行到1000行是子程序，主程序共调用了5次子程序。

练习：

1. 运行下面程序，看是什么结果

```

10  FOR  I = 0  TO  2
20  GOSUB 500
30  NEXT  I
40  FOR  I = 1  TO  0 STEP  -1
50  GOSUB 500
60  NEXT  I

```

```

70  END
500  FOR  J = - I  TO  I
510  PRINT  TAB(10-I);  " * ";
520  NEXT  J
530  PRINT
540  RETURN

```

2. 试编程序打印下列图案

(1)

```

      *
    ***
  *****
*****

```

(2)

```

      ****
     ****
    ****
   ****
  ****
 ****

```

第二课 二维数组用途大

为进一步了解数组的用途，我们引入二维数组的概念。
什么是二维数组呢？

由双下标变量组成的数组叫二维数组。如 $A(X, Y)$ 、 $B(5, 10)$ 就是二维数组。其中的 X 和 Y 必须预先赋值。

原来是这样，
我明白啦！

二维数组的格式：

行号 DIM x(3,5)

这是一个二维数组，
3和5规定数组的
大小，也可以用变
量表示。



例如：数组 $X(3, 3)$ ，它表示 X 数组中共存有 $3 \times 3 = 9$ 个数，在计算机内的存放是先存行后存列。

例如：有三个学生的语文、数学、自然考试成绩分别是：

		语 文	数 学	自 然
李	明	70	95	81
王	红	90	100	80
张	勇	88	92	75

如果用数组存放，以上成绩可以用数组 $X(3, 3)$

$$X \begin{Bmatrix} 1, 1 & 1, 2 & 1, 3 \\ 2, 1 & 2, 2 & 2, 3 \\ 3, 1 & 3, 2 & 3, 3 \end{Bmatrix} = X \begin{Bmatrix} 70 & 95 & 81 \\ 90 & 100 & 80 \\ 88 & 92 & 75 \end{Bmatrix}$$

编写一个程序，求出每个同学的平均成绩：

```
10 DIM X(3, 3), Y(3)
20 FOR I = 1 TO 3
30 FOR J = 1 TO 3
40 READ X(I, J)
50 Y(I) = Y(I) + X(I, J)
60 NEXT J
70 PRINT Y(I)/3
80 NEXT I
90 DATA 70, 95, 81, 90, 100, 80, 88, 92,
      75
100 END
      RUN ↵
      82
      90
      85
```

上面程序中，X数组定义了九个内存单元，为了存放每个同学的各门成绩，通过读数语句分别送入，并计算出结果。

练习：

1. 编写全班学生三门功课的平均成绩。
2. 编一个九九表并显示结果。

第三课 字符串函数(1)

以前我们学习了字符串和字符串变量,为了更深入地学习 BASIC 语言,以便解决更多的实际问题,我们来学习字符串函数。

什么是字符串函数? BASIC 语言规定了一些特定的字符串函数,它可以改变字符串的排列和转换,我们称它为字符串函数。这一课我们先学习 ASC、CHR\$、STR\$ 和 VAL 四个字符串函数。

1. ASC 函数

一般格式:

ASC (字符串变量)

ASC 是 ASCII 的前三个字母。ASCII 是英语“美国信息交换标准代码”的缩写。计算机所用的每一个字符,都有其对应的 ASCII 代码(见附录一)。例如 A 的 ASCII 代码为十进制数 65, 1 的 ASCII 代码为十进制数 49。字符的 ASCII 代码最大是 225。ASC 的功能是把字符串变量的第一个字符转换为与其对应的 ASCII 代码。

例如:

```
5  A$ = "ABCD"
10 PRINT "A = ", ASC(A$)
20 D$ = "1 2 3"
30 PRINT "I = ", ASC(D$)
40 END
```

```
RUN↵
```

```
A = 65
```

```
I = 49
```

这里，65和49分别为A和1的ASCII代码。

2. CHR\$函数

一般格式：

CHR\$ (十进制数)

CHR\$函数和ASC函数的功能恰好相反，它可把ASCII代码转换成与其对应的字符。

例如，要把66和50这两个ASCII代码，转换为与其对应的字符时，CHR\$()就可大显身手了。这个程序如下：

```
10 PRINT CHR$(66), CHR$(50)
```

```
20 END
```

```
RUN↵
```

```
B      2
```

下面的程序，可显示全部ASCII代码所对应的字符，请上机练习。

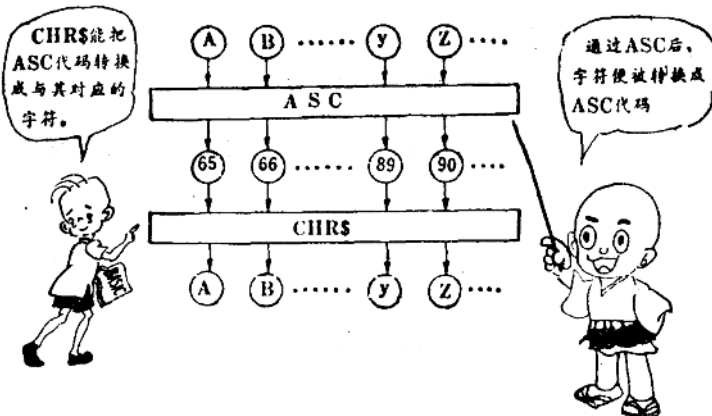
```
10 FOR I = 0 TO 255
```

```
20 PRINT CHR$(I)
```

```
30 NEXT I
```

```
40 END
```

```
RUN↵
```



3 STR\$()函数

一般格式:

STR\$ (数)(或表达式)

STR\$()的功能是可把变量的数字转换成字符性的数。这种字符性的数不是数值, 它不能进行算术运算。

例如:

```
10 A = 12
20 C$ = STR$(A)
30 PRINT C$
40 END
RUN
12
```

这里, 12是字符, 而不是数值12。

4. VAL函数

一般格式:

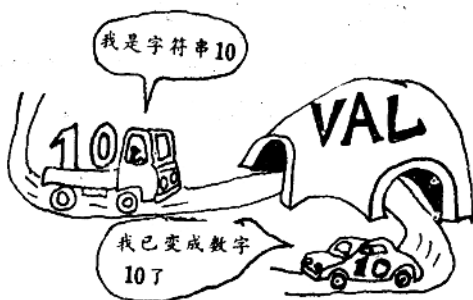
VAL 数字串(或字符变量)

VAL函数与STR\$函数的功能恰恰相反,它的功能是把字符串变成数值。

例如:

```
10 A$ = "1 2 3"  
20 B$ = "4 5 6"  
30 C = VAL(B$ + A$)  
40 PRINT C  
50 END  
RUN
```

456123 (这里的456123是数字)



练习:

1. 什么是ASC函数和CHR\$函数? 功能是什么?
把A、B、C变为ASCII代码; 把50、60、70变成ASCII码所对应的字符。

2. 什么是STR\$函数和VAL函数? 功能是什么?

把字符串 40、50、60变成数值。

第四课 字符串函数(2)

在这一课里, 我们再学习四个字符串函数。

1. INKEY\$函数

一般格式:

字符串变量 = INKEY\$

INKEY是英语“键盘输入”的意思, 读作: “yinkei”, INKEY\$ 函数的功能是把从键盘输入的字符送给指定的字符变量, 不管输入的是字符还是数字, 计算机都把它当作字符对待。变量只接受第一次按的字符, 其他均不理睬。

例如: 10 A\$ = INKEY\$

20 PRINT "A B C D", A\$

30 GOTO 10

40 END

RUN

如果打入E, 屏幕上就显示出: A B C D E; 打入F, 则显示: A B C D F……, 按什么字符, 在A B C D后边就显示什么字符。

2. LEFT\$()函数

一般格式:

LEFT\$ (字符串或字符串变量, 数值)

LEFT\$()函数的功能是对字符串或字符串变量从左边算起截取指定个数的字符, 用来组成新的字符串。

例如: 10 A\$ = "BASIC—LASER—310"

20 B\$ = LEFT\$(A\$, 5)

30 C\$ = LEFT\$(A\$, 11)

40 PRINT A\$, B\$, C\$

50 END

RUN,↵

BASIC—LASER—310

BASIC

BASIC—LASER

3. RIGHT\$()函数

一般格式:

RIGHT\$ (字符串或字符串变量, 数值)

RIGHT是英文“右”的意思。RIGHT\$()函数的功能是对字符串或字符串变量自右至左截取规定个数的字符, 字符的排列顺序不变。

例如: 10 A\$ = "A B C D E F G H I J K"

20 B\$ = RIGHT\$(A\$, 6)

30 PRINT A\$

40 PRINT B\$

50 END

RUN,↵

A B C D E F G H I J K