

COCOL语言使用说明



COCOL (COntersational COmputing Language) 是日商中央电子株式会社所生产的 CEC—555 型电子计算机使用的一种计算语言。CEC—555 型电子计算机共配有四种语言, 即: 符号汇编语言 (包括 SAP—II, SAP—III, SAP—D 等), 对话型语言 (COCOL—71), FORTRAN 语言 (包括 4 K FORTRAN, 8 K FORTRAN 及 FORTRAN—D), COBOL 语言。

本说明书是针对对话语言 COCOL—71 而言的。由于厂方提供的说明书与实际提供的软件 (COCOL—71 程序纸带) 有一定的出入, 给使用带来了困难。我们经过一个时期的摸索, 根据在工作实践中的一些认识, 重新编写了 COCOL 语言的使用说明。

由于我们的水平不高, 工作实践也很有限, 因此, 难免会有不少的谬误之处, 希望同志们批评指正。

交通部科学研究院铁建所

1974·7·于北京

目 录

1. 引 言.....	(1)
2. 变 量.....	(2)
3. 算术运算.....	(3)
4. 指令S、T	(3)
5. 输出数据的格式.....	(6)
6. 间接指令.....	(9)
7. 指令D、G、Q、R	(10)
8. 指令A、I、G、F	(12)
9. 指令E、W、C	(18)
10. 错误的检查及修正, 指令M	(20)
11. 函 数.....	(24)
附录1. COCOL指令一览表	(27)
附录2. COCOL程序的操作	(28)
附录3. 错误信息.....	(30)
附录4. 例 题.....	(31)

1. 引 言

COCOL (CONversational COmputing Language的缩写) 是 CEC—555型电子计算机用的一种计算语言。它是一种适合于一般科学计算的人—机对话语言。

使用COCOL语言必需的设备: CEC—555型电子计算机; 电传打字机 (CS—1000型电传打字机或CS—100型电传打字机)。

使用COCOL语言需要的磁心存储器为4096单元 (字长12位)。

启动COCOL程序后, 程序控制打出

CONGRATULATIONS!!

YOU HAVE SUCCESSFULLY LOADED 'COCOL 1971' ON A
CEC—555 COMPUTER.

接着, 是关于选择方面的对话。COCOL打出

SHALL I RETAIN LOG, EXP, ATN ?;

使用者若需要函数 e^x , $\log_e x$ 及 $\arctan x$, 则答YES $\curvearrowright$ (这里的 $\curvearrowright$ 表示按键盘上的复归键, 也就是控制打字机字盘返回的键), 若不需要, 则答NO $\curvearrowright$ 。若答YES $\curvearrowright$, 则程序打出

PROCEED.

*

*号表示COCOL程序可以接受使用者的指令输入。若答NO $\curvearrowright$, 则COCOL还问

SHALL I RETAIN SINE, COSINE ?;

待使用者回答后才打出

PROCEED.

*

下面是上述初始对话的一个例子, 在这个例子中, 使用者需要正弦函数及余弦函数, 而不需要对数函数, 指数函数和反正切函数。

CONGRATULATIONS!!

YOU HAVE SUCCESSFULLY LOADED 'COCOL 1971' ON A CEC-555 COMPUTER.

SHALL I RETAIN LOG, EXP, ATN?: NO

SHALL I RETAIN SINE, COSINE?: YES

PROCEED.

*

2. 变 量

在COCOL语言中, 变量由一个或两个英文字母及数字来表示, 其中第一个必须是英文字母(不可以为F)。

例如:

A BI CL ID

这些都称之为变量名称。

变量还可以带有下标。下标用括号括起来, 写在变量名称的紧后面。下标允许用-2047至+2047的整数。也可以用一个算式来作为下标。

例如:

AI(I+3*J) X(5) C(N) SU(66)

若使用者书写的变量名称超过两个字, 则COCOL只识别最前面的两个。例如, SUM1和SUM2, DTA与DTB等, 在COCOL看来都是一样的, 前者都是SU, 而后两个都是DT。

3. 算术运算

COCOL能执行加法、减法、乘法、除法及指数运算。

符 号	数 学 表 示	COCOL 语 言
^ (注)	2^3	2 ^ 3
*	$2 \cdot 3$	2 * 3
/	$2 \div 3$	2 / 3
+	$2 + 3$	2 + 3
-	$2 - 3$	2 - 3

其中指数运算的指数值应该是正整数。

在一个算式中同时有几种运算时，则按上表的先后顺序来进行计算。其中+和-就先后顺序而言是同等的，也就是说，他们同时出现时，按自左至右的顺序进行计算。而乘法则应比除法优先进行。

例如：

$X/2*Y$ 是 $\frac{X}{2 \cdot Y}$ 而不是 $\frac{X \cdot Y}{2}$ 。

在算式中可以适当地使用括号：()、< >、[] (这三种括号在COCOL中是同等看待的)。在括号中的运算比括号外的运算优先进行。

4. 指 令 S、T

COCOL中的指令都以单个英文字母来表示。若指令还要附有

注) 指数运算在CS—1000型打字机上是用(^)来表示，而在CS—100型则是用(↑)。

内容，则在表示指令的字母与所附内容之间要用空格（按电传打字机上的SPACE键）隔开。空格在电传打字机上是不打印出来的（有时，为了便于说明问题，我们往往以“□”来表示空格）

在COCOL中，一行可以写几条指令，中间用分号“；”隔开。但每一行最多只能输入72个字符。使用者输完一行指令后，按复归键，则COCOL即开始执行这些指令，执行完后，再打出*号。

4.1 指令S (SET)

指令S是用来给定变量的值的。一个S指令，只能给定一个变量的值。

例如：

*S A = 3.14159; S B(3) = 428.77

*S C = 12*A + B(3)

*

4.2 指令T (TYPE)

指令T的意思是：请打印出后面变量或算式的值。

一个T指令，可以同时要求打印出几个变量或算式的值。此时，在这些变量或算式之间用逗号“，”分开。

例如：

*T 6.4318 + 8.1346

= 14.5664*

*

*S A = 6.5; S B(1) = 7.3

*T A, B(1), A + B(1)

= 6.5000 = 7.3000 = 13.8000*

*

指令T还可用来让程序印出指定的内容。在T后面若附有用引号围起来的字符（其中包括电传打字机可以印出的全部字符以及空格），则在执行该T指令时，COCOL将引号中的内容印出。

*T "COCOL PROGRAM"

COCOL PROGRAM*

由于指令T在打印出内容或结果的过程中，不能自动地进行复归（字盘返回）与改行，为此，需要应用控制符（!）与（#）。

指令T在执行中一旦遇见惊叹号（!），即自动地进行打字机的复归与改行。

*S A = 1; S B = 2; S D = 3

*T "ALPHA"A,! "BETA"B, " DELTA"D,!

ALPHA = 1.0000

BETA = 2.0000 DELTA = 3.0000

*

指令T在执行中一旦遇见记号（#），则仅进行复归，而不改行。

此外，还有一个控制符（\$）。指令T在执行中一旦遇见记号（\$），则将内存存储器中全部已定义的变量名称及其相应的值打印出来。（记号\$必须在一行的最后）

例如：

*T \$

A@(01) = 23.5000

B@(00) = 1.0000

C@(00) = 0.0000

V@(07) = 0.1230

B@(01) = 2.3000

ID(00) = 7.0000

PG(01) = 25.0000

PG(02) = 256.0000

*

这个例子表明，使用者已定义了变量为：

A(1), B, C, V(7), B(1), ID, PG(1), PG(2)。

这里，若变量名称只有一个字母，则程序自动在它后面添上一个@。

5. 输出数据的格式

COCOL语言中是以指令T来输出数据的。

使用者可以用如下的指令输入来规定输出数据的格式：

T_ %WX.YZ,

这里WX表示输出的全部位数，最多不能超过31。YZ表示小数点右边最多允许的位数。WX、YZ都是非负整数。若YZ为0（即小数点后面的数字不要），则可直接写为

T_ %WX,

在最初送入COCOL程序后，启动后即使对输出格式不作任何规定，COCOL自动地按%8.04的格式输出数据。例如：

*T 12.34

= 12.3400*

*

但是，一旦改变了输出格式之后，则在不再作新的格式规定之前，数据输出都按该改变了的格式进行。

例如：

*T %2,5

= 5*

*T 3*7

= 21*

*T %4.02, 12.22 + 2.37

= 14.59*

*T 1.234, 3, 34.56

= 1.23 = 3.00 = 34.56*

*

COCOL可以输出六位以上的数字，但实际上有效数字只取六位。

*S B = 1111.1111

*T %8.04, B

= 1111.1100*

*

要输出的数值的位数超过指定的输出格式范围时，若是已不可能用指定的格式来表示该数，则COCOL自动转为用浮点形式输出数值，但此时有效数字的个数还是由格式符号%后的数来定。例如：

*T %3,67823
= 0.678E+05*

否则，则在经过舍弃之后，还是用原指定的格式来表示。例如：

*T %3,23.45,2.3425
= 24 = 2*

要输出的数值比指定的输出格式范围小时，COCOL在数的前面置以空位，在数的后面则置0。

例如：

*T %7,12345
= 12345*

*T %8.04,016,0.016,0,7.00
= 16.0000 = 0.0160 = 0.0000 = 7.0000*

*

输出格式中规定了小数点右边最多允许的位数，例如%7.05，表示小数点后边最多可有五位，实际上，有时候并不输出五位。但是，输出的数的总位数还应该是7。

*T %7.05,12.345,123.456,12345.6,1234567
= 12.34500 = 123.4560 = 12345.60 = 1234570*

此外，为了处理较大的和较小的数值，使用者可以用指数形式（浮点形式，也叫E形式）来输出数据。

使用者只要作如下的指令输入

T_%,

则输出格式即为指数形式。

例如:

*T %,11,345.67

= 0.110000E+02 = 0.345670E+03*

*

这里的E+02, E+03, 即相当于 10^2 , 10^3 。小数点后给出六位数字。作为指数, 可以用-616到+616之间的整数。因此, COCOL可以表示的绝对值最大的数值为 0.999999×10^{616} 。

T %,2^2047.1

= 0.161586E+616*

*

6. 间 接 指 令

所谓间接指令, 是指不是一按复归键就马上执行的指令。在一行的最左边打上一个行编号, 空一格后接着打指令, 则该行的指令在最后按复归键后即被存储起来, 而不立刻执行。存完后, COCOL在下一行打印出*号。

因此, 也可以说: 间接指令是前面带有行编号的指令。

行编号允许用01.01~31.99。点前面的数称为组编号。点前后都应是正整数。行编号不可用01.00, 02.00等等, 这些, 是用来表示整个组的所有行的。

例如:

*01.01 S A=1; S B=2

*01.03 S C=3

*01.05 T %1, A+B+C

*

*02.10 S X = 3.5

*02.40 T %, X*Y, X/Y

*02.25 S Y = 7.82

输入各行的先后次序，是可以任意的。

间接指令所组成的程序，就叫做间接程序。间接程序由指令 G 和 D 来控制执行。

7. 指令 D、G、Q、R

7.1. 指令 D (DO)

指令 D 是用来控制间接程序的执行的。被执行的间接程序，可以是一行，可以是一组，也可以是整个间接程序。

例如：

*D₁ (执行整个间接程序，自编号最小的开始，按编号的大小顺序往下执行)

*D₂ (执行第二组的全部指令，自编号最小的行开始)

*D_{3.12} (执行 03.10 行的指令)

执行完指定的间接程序中的行或组后，程序又回到 D 指令后面的指令。例如：

*

*01.10 T "A"

*01.20 T "C"; D 1.1

*01.30 T "B"

*D

ACAB*

*

*D 1.3

B*

*

执行整个程序时,先打印出字母A,接着打出字母C,接着是 D 1.1,也就是再印一个A,执行完01.10行后,程序回过来执行D 1.1指令后面的指令,即打印字母B。这就是整个程序的执行过程。

7.2. 指令G (GO)

后面不附行编号的指令G与单个的D作用相同。都是用来控制整个间接程序的执行的。

后面附有行编号的指令G (GO TO)将在下一章中介绍。

7.3. 指令Q (QUIT)

指令Q用于中止程序的执行。在程序执行中,一旦遇到指令Q,则程序不再继续往下执行,COCOL打出*号,向使用者表示可以接受新的输入。

7.4. 指令R (RETURN)

在用指令D执行某一行或某一组时,一旦遇到指令R,则程序立即返回,转向执行指令D后面的指令。

例如;

*01.05 S X = 1.1; S Y = 9.9

*01.20 S Z = 3; D 2

*01.30 S Z = 5; D 2.3

*01.40 Q

*

*02.10 T %, (X+Y) ^Z, !

*02.20 R

*02.30 T X^Z+Y^Z, !

*

*G

= 0.133100E+04

= 0.951006E+05

*

8. 指令A、I、G、F

8.1. 指令A (ASK)

指令A是在程序执行时的特定点，程序向使用者要求输入数据时用的指令。

例如：

*01.05 A X, Y, Z

*

COCOL 一执行01.05行，就打出冒号(：)。此时，使用者应输入变量X所相应的数值，紧接着打上一个分隔记号(逗号，空格或复归)，COCOL接着又打出冒号，等待使用者再输入变量Y的值，……直至A后面所附的变量名称都给了值为止。

例如：

*01.05 A X, Y, Z

*

***D 1.05**

:5, :4.3 :.67E-2

*

若是在未打分隔记号之前，发现数据打错了，则可以立即改正。为此，只要在已打的值后面打上一个下短横（_）注），接着重新打一个正确的数值。

例如：

***01.20 A X(1), X(2)**

***02.10 T %7.04, X(1), X(2)**

*

***D**

: 123.4 :234.5_34.567

= 123.4000 = 34.5670*

*

指令A还有一个功能，类似于指令T，在指令A后面若附有用引号围起来的字符，则在执行指令A时，COCOL把引号中的内容打印出来。

01.10 A "HOW MANY APPLES DO YOU HAVE? "AP

01.40 T %2, AP

01.70 Q

***G**

HOW MANY APPLES DO YOU HAVE? :34

= 34*

注）这是就CS—1000型电传打字机而言的，相当于CS—100型上的左箭头（←）。

*

此外, 控制符 (!), (#), (\$) 等, 对于指令A同样也是适用的。

01.20 A "A"A, ! "B"B, " C"C, ! !

01.50 T %, "A"A, ! "B"B, " C"C, !

01.70 Q

*

*D

A: 1.23

B: 456 C: 789E - 12

A = 0.123000E + 01

B = 0.456000E + 03 C = 0.789000E - 09

*

*A. \$

A@(00) = 0.123000E + 01

B@(00) = 0.456000E + 03

C@(00) = 0.789000E - 09

*

8.2. 指令I (IF)

指令I的一般表现形式为:

I, 空格, 用括号括起来的变量或数式, 以逗号分开的三个行编号。