

计算机实用工具丛书二

常用微机 通用软件使用手册

下 册

张 宁

徐启甫

郭文华

李修华

杨学军

编

中国计算机函授学院

前 言

随着微型计算机应用的深入普及和推广，越来越多的人迫切需要了解 and 掌握计算机的操作与使用，有些已从事计算机方面工作的同志，更希望能在使用的基礎上多掌握一点计算机技术，以便能自己开发一些专用的、小型的应用软件。为了帮助这些同志尽快实现自己的愿望，使有志于计算机应用开发的同志早出成果，我们编写了这本《常用微机通用软件使用手册》，旨在为广大的计算机爱好者和计算机应用人员提供一些实用的有价值的资料，帮助大家去掌握一些常用软件的使用，掌握一些软件使用和编程的技巧。书中列举了大量的例子，以使读者易于理解和仿效。

本书较详细地介绍了目前国内最流行的两种机型——IMB PC及其兼容机和APPLE II及其兼容机上常用的一些软件及其使用。全书共分三篇：第一篇主要介绍两种机型上使用的多种操作系统，包括PC DOS、CC DOS、CP/M和苹果DOS等；第二篇主要介绍多种程序设计语言，包括宏汇编语言、各种高级语言等；第三篇主要介绍各种用途的实用程序，包括中文文字处理软件、汉字表格软件、汉字数据库管理系统和AutoCAD绘图软件包等。

本书在编写过程中，力求通俗易懂、叙述清楚。书中的例子绝大部分都已在相应的机器上调试和运行，对广大微机应用开发人员也许会有一定的参考和实用价值。

本书第一篇第一章的第一至六节及第二章、第三章和第二篇的第一章由张宁执笔；第二篇的第二至十章由徐启甫执笔；第一篇第一章的第七节及第四章、第五章和第三篇的第二、三、五章由郭文华执笔；第三篇的第一章由杨学军执笔；第三篇的第四章由李修华执笔。

本书承南京大学张福炎教授主审，同时参加审阅的还有南京大学钱士钧、蒋新儿、蔡士杰、戈孝珍、李滨宇等同志，并在认真细致审阅的基础上，提出了许多宝贵的意见。在编写过程中，还得到了中国计算机函授学院钱测胜等同志的支持和帮助，编者在此谨一并向他们表示衷心的感谢。

由于编写时间仓促，加之编者水平有限，书中错误和不妥之处在所难免，敬请读者不吝批评指正。

编 者

一九八八年五月

目 录

第二篇 程序设计语言及其使用

第四章 FORTRAN语言

§ 4.1 基本概念	(1)
4.1.1 字符集和源程序书写格式	(1)
4.1.2 常量、变量和数据类型	(2)
4.1.3 表达式和赋值语句	(3)
4.1.4 数组	(5)
4.1.5 子字符串	(6)
4.1.6 内部函数	(7)
4.1.7 简单的输入输出操作	(11)
4.1.8 程序结构	(14)
§ 4.2 控制语句	(15)
4.2.1 ASSIGN语句和GOTO语句	(15)
4.2.2 算术IF和逻辑IF语句	(16)
4.2.3 块IF语句和块IF	(17)
4.2.4 DO语句和CONTINUE语句	(20)
4.2.5 STOP、PAUSE、END和RETURN 语句	(23)
§ 4.3 函数和子程序	(24)
4.3.1 SUBROUTINE子程序	(24)
4.3.2 FUNCTION函数子程序和语句函数	(25)
4.3.3 程序单位之间的数据通讯——虚实结合	(26)
4.3.4 可调数组和假定大小数组	(28)
4.3.5 SAVE语句和ENTRY语句	(30)
§ 4.4 说明语句	(30)
4.4.1 EXTERNAL和INTRINSIC语句	(30)
4.4.2 PARAMETER和DATA语句	(31)
4.4.3 COMMON语句和BLOCK DATA子程序	(32)
4.4.4 EQUIVALENCE (等价) 语句	(34)
§ 4.5 输入输出	(35)
4.5.1 基本输入输出语句	(36)
4.5.2 FORMAT语句	(39)
4.5.3 内部文件	(42)
4.5.4 例子——直接文件的建立、修改和查询	(42)

第五章 FORTRAN的使用

§ 5.1 APPLE PASCAL下使用FORTRAN	(45)
------------------------------	--------

5.1.1	FORTRAN程序上机操作	(45)
5.1.2	错误信息	(46)
§ 5.2	APPLE II 机CP/M下使用FORTRAN	(53)
5.2.1	FORTRAN程序上机操作	(53)
5.2.2	错误信息	(55)
§ 5.3	IBM—PC DOS下使用FORTRAN	(59)
5.3.1	FORTRAN程序上机操作	(59)
5.3.2	编译程序元命令	(65)
5.3.3	错误信息	(66)

第六章 PASCAL语言

§ 6.1	基本概念	(77)
6.1.1	词汇和标识符	(77)
6.1.2	常量和表达式	(78)
6.1.3	数据的类型	(79)
6.1.4	变量及其类型说明	(81)
6.1.5	标准函数	(81)
6.1.6	标准类型	(82)
6.1.7	复合语句和赋值语句	(84)
6.1.8	输入输出	(85)
6.1.9	程序结构	(87)
§ 6.2	控制语句	(89)
6.2.1	IF语句和GOTO语句	(89)
6.2.2	循环语句	(91)
6.2.3	CASE语句	(93)
§ 6.3	枚举类型和子界类型	(95)
6.3.1	枚举类型	(95)
6.3.2	子界类型	(96)
§ 6.4	函数和过程	(97)
6.4.1	函数说明和过程说明	(97)
6.4.2	函数和过程的调用	(98)
6.4.3	变量作用域	(101)
6.4.4	递归调用	(103)
§ 6.5	数组和记录	(105)
6.5.1	数组类型	(105)
6.5.2	记录类型和WITH语句	(112)
§ 6.6	集合类型	(117)
§ 6.7	指针与动态数据结构	(120)
6.7.1	指针	(120)
6.7.2	链表	(123)

§ 6.8 文卷.....	(126)
6.8.1 文卷类型.....	(126)
6.8.2 非文本文卷的操作.....	(126)
6.8.3 文本文卷的操作.....	(128)
附录: PASCAL 语法图.....	(130)

第七章 PASCAL的使用

§ 7.1 APPLE PASCAL下使用PASCAL.....	(133)
7.1.1 PASCAL程序上机操作.....	(133)
7.1.2 出错处理.....	(136)
§ 7.2 IBM—PC DOS下使用PASCAL.....	(137)
7.2.1 PASCAL程序上机操作.....	(137)
7.2.2 编译程序元命令.....	(137)
7.2.3 错误信息.....	(142)

第八章 COBOL语言

§ 8.1 基本概念.....	(154)
8.1.1 字符集、用户定义名和直接量.....	(154)
8.1.2 层次式数据描述.....	(156)
8.1.3 程序结构及程序书写格式.....	(157)
§ 8.2 标识部分和环境部分.....	(158)
8.2.1 标识部分.....	(158)
8.2.2 环境部分.....	(159)
§ 8.3 数据部分及其子句.....	(159)
8.3.1 文件描述子句.....	(160)
8.3.2 数据描述子句.....	(161)
8.3.3 工作存贮节和连接节.....	(167)
§ 8.4 过程部分及MOVE语句.....	(169)
§ 8.5 非文件方式少量数据的输入输出.....	(174)
§ 8.6 算术语句(加、减、乘、除和计算语句).....	(174)
§ 8.7 控制语句.....	(176)
8.7.1 GO语句和IF语句.....	(176)
8.7.2 PERFORM语句和EXIT语句.....	(181)
8.7.3 CALL语句和STOP语句.....	(185)
§ 8.8 输入输出文件.....	(189)
8.8.1 磁盘顺序文件的输入输出.....	(190)
8.8.2 磁盘索引文件的输入输出.....	(192)
8.8.3 报表打印.....	(197)
§ 8.9 排序.....	(198)
§ 8.10 表处理.....	(199)

8.10.1 表的建立和引用.....	(200)
8.10.2 指标值的初置和修改——SET语句.....	(203)
8.10.3 用SEARCH语句查表.....	(205)
8.10.4 用PERFORM语句表查.....	(207)
附录 ANSI COBOL x3.23——1974的语句格式表.....	(208)

第九章 COBOL的使用

§ 9.1 在CP/M下使用COBOL—80.....	(225)
9.1.1 COBOL—80程序上机操作.....	(225)
9.1.2 错误信息.....	(227)
§ 9.2 在IBM PC DOS下使用COBOL.....	(235)
9.2.1 COBOL程序上机操作.....	(235)
9.2.2 错误信息.....	(237)
§ 9.3 COBOL程序中易犯的语法错误举例.....	(238)

第十章 IBM PC编译 BASIC

§ 10.1 编译BASIC与解释BASIC.....	(242)
10.1.1 编译BASIC的编译元命令.....	(242)
10.1.2 语言上的差别.....	(243)
10.1.3 综合性差别.....	(247)
§ 10.2 IBM PC DOS下使用编译BASIC.....	(247)
10.2.1 建立源程序.....	(247)
10.2.2 编译.....	(248)
10.2.3 连接运行.....	(250)
10.2.4 错误信息.....	(251)

第三篇 实用软件

第一章 中文文字处理软件CWORDSTAR

§ 1.1 启动CWORDSTAR.....	(254)
§ 1.2 基本命令——建立帮助级.....	(251)
§ 1.3 打开文件命令.....	(255)
1.3.1 打开文书文件.....	(255)
1.3.1.1 进入文字编辑.....	(255)
1.3.1.2 利用控制键进行增、删、插、改.....	(255)
1.3.1.3 Q命令.....	(257)
1.3.1.4 屏幕命令.....	(260)
1.3.1.5 K命令.....	(261)
1.3.1.6 排版.....	(263)

1.3.1.7 点命令.....	(263)
1.3.2 打开非文书文件.....	(264)
§ 1.4 文件命令.....	(264)
1.4.1 打印文件.....	(264)
1.4.2 重命名文件.....	(266)
1.4.3 拷贝文件.....	(266)
1.4.4 删除文件.....	(267)
§ 1.5 字处理选择——合并打印.....	(267)
§ 1.6 系统命令.....	(270)
§ 1.7 应用举例.....	(271)
§ 1.8 命令汇集.....	(272)

第二章 汉字表格软件SUPERCALC3

§ 2.1 SuperCaLc 3 的使用.....	(275)
§ 2.2 SuperCaLc 3 基本命令.....	(276)
2.2.1 单元区域的表示.....	(277)
2.2.2 表格的装入、保存和退出命令.....	(277)
2.2.3 插入、删除、复制及数据移动命令.....	(278)
2.2.4 格式命令.....	(280)
2.2.5 全局命令.....	(281)
2.2.6 标题锁定命令.....	(281)
2.2.7 窗口命令.....	(282)
2.2.8 数据保护和解保护命令.....	(283)
2.2.9 排序命令.....	(283)
2.2.10 输出命令.....	(284)
§ 2.3 绘图命令.....	(285)
§ 2.4 数据管理命令.....	(288)
§ 2.5 公式及函数.....	(290)
2.5.1 数值函数.....	(291)
2.5.2 逻辑函数.....	(291)
2.5.3 日历函数.....	(291)
2.5.4 经济函数.....	(292)
2.5.5 特殊函数.....	(294)

第三章 汉字表格软件OFFICE

§ 3.1 概述.....	(296)
§ 3.2 数据参数及运算符.....	(297)
§ 3.3 命令.....	(299)
§ 3.4 OFFICE函数.....	(306)

§ 3.5 运算条件的设定	(312)
---------------	---------

第四章 汉字dBASE III 关系数据库管理系统

§ 4.1 汉字dBASE III 的初步知识	(314)
4.1.1 文件及其类型	(314)
4.1.2 常量和变量	(316)
4.1.3 表达式和函数	(317)
4.1.4 命令 (或语句)	(321)
4.1.5 全屏幕工作方式	(322)
4.1.6 运行汉字dBASE III	(322)
§ 4.2 dBASE III 的基本命令	(323)
4.2.1 dBASE III 自学习命令	(323)
4.2.2 文件操作命令	(327)
4.2.3 计算并输出表达式值的命令	(328)
4.2.4 关于数据库文件结构的命令	(328)
4.2.5 关于数据库文件内容的命令	(332)
4.2.6 关于多个数据库文件的命令	(348)
§ 4.3 应用程序设计	(359)
4.3.1 内存变量的使用	(359)
4.3.2 应用程序的建立和运行	(363)
4.3.3 应用程序的结构与控制	(364)
4.3.4 屏幕显示格式设计	(375)
4.3.5 打印报表格式设计	(382)
4.3.6 系统控制参数	(386)
4.3.7 一个应用实例	(394)
§ 4.4 编译dBASE III 简介	(403)
4.4.1 编译程序CLIPPER简介	(403)
4.4.1.1 概述	(403)
4.4.1.2 系统环境	(404)
4.4.1.3 系统组成	(404)
4.4.2 编译与解释dBASE III 语言的异同	(404)
4.4.2.1 编译dBASE III 语言增强的功能	(404)
4.4.2.2 与解释dBASE III 语言的差别	(409)
4.4.3 应用程序的编译和连接	(411)
4.4.3.1 程序的编译	(412)
4.4.3.2 程序的连接	(412)
4.4.4 应用程序的调试	(414)
4.4.5 几个实用程序	(417)

第五章 Auto CAD绘图软件包

§ 5.1	概述	(419)
§ 5.2	Auto CAD的配置	(419)
§ 5.3	Auto CAD功能菜单	(423)
§ 5.4	基本概念和术语的含义及有关参数的输入	(425)
§ 5.5	Auto CAD的基本命令	(427)
5.5.1	HELP帮助命令	(427)
5.5.2	FILES磁盘文件目标显示	(428)
5.5.3	SAVE 保存命令	(429)
5.5.4	SHELL 执行操作系统命令	(429)
5.5.5	END 结束图型编辑命令	(430)
5.5.6	QUIT 退出编辑命令	(430)
5.5.7	MENU 菜单命令	(430)
5.5.8	绘图命令	(431)
§ 5.6	Auto CAD的绘图命令	(431)
5.6.1	LINE 直线命令	(431)
5.6.2	POINT 点命令	(432)
5.6.3	CIRCLE 园命令	(433)
5.6.4	ELLIPSE椭圆命令	(434)
5.6.5	ARC 弧命令	(435)
5.6.6	TRACE 线条命令	(438)
5.6.7	PLINE 多义线命令	(439)
5.6.8	SOLID 实体命令	(443)
5.6.9	TEXT和DTEXT文字注解命令	(444)
5.6.10	STYLE文本字型命令	(447)
5.6.11	DIM 半自动标注尺寸命令	(448)
5.6.12	HATCH 阴影线命令	(454)
§ 5.7	AUTO CAD图形编辑命令	(456)
5.7.1	编辑状态控制	(456)
5.7.1.1	STATUS 绘图状态显示命令	(456)
5.7.1.2	LIMITS 图幅命令	(457)
5.7.1.3	UNITS 单位命令	(457)
5.7.1.4	ATTDISP 属性显示开关	(458)
5.7.1.5	AXIS 坐标轴标尺开关	(458)
5.7.1.6	BLIPMODE命令	(459)
5.7.1.7	DRAWMODE动态提示开关	(459)
5.7.1.8	REGENAUTO自动重新生成开关	(460)
5.7.1.9	FILL 填充开关	(460)
5.7.1.10	GRID 网格命令	(460)

5.7.1.11	ORTHO正交方式开关.....	(461)
5.7.1.12	AUTOCAD的捕捉机制.....	(462)
5.7.2	屏幕显示控制.....	(463)
5.7.3	基本编辑命令.....	(465)
5.7.4	计算命令.....	(469)
5.7.5	其它编辑命令.....	(470)
5.7.6	图层、线型及颜色.....	(478)
5.7.6.1	LAYER 图层管理命令.....	(478)
5.7.6.2	LINETYPE线型命令	(480)
5.7.6.3	LTSCALE线型比例命令	(483)
§ 5.8	块.....	(483)
5.8.1	概述.....	(483)
5.8.2	BLOCK块的定义命令.....	(483)
5.8.3	INSERT块的插入命令.....	(484)
5.8.4	WBLOCK块存盘命令.....	(485)
§ 5.9	三维功能.....	(486)
§ 5.10	屏幕菜单命令.....	(490)
§ 5.11	图型输入设备的使用.....	(492)
§ 5.12	AutoLISP的使用.....	(492)
附一	AutoCAD命令表.....	(493)
附二	AutoLISP函数表.....	(499)

第二篇 程序设计语言及其使用

第四章 FORTRAN语言

FORTRAN语言是多年来一直广泛使用的语言，它是FORMula TRANslation的缩写，故又称公式翻译语言。用它编写的程序简明直观、结构清晰、标准化程度高，能方便地描述复杂的数学模型且计算速度很快。因此，最适宜用于科学计算。

自1954年提出FORTRAN语言以来，人们对它进行了多次修改，形成了二个重要标准：

ANSI x3.9-1966 FORTRAN (称FORTRAN66,大致相当于FORTRANIV)

ANSI x3.9-1978 FORTRAN (称FORTRAN77)

FORTRAN77在FORTRAN66上作了一些改进和扩展，在某些细微之处也存在一些矛盾，熟悉用66标准的人用77标准时须留心。一般说来，用FORTRAN66 (或FORTRANIV) 编写的程序可以不作修改 (或稍加修改) 便可在FORTRAN77编译程序下编译。

目前，很多计算机系统上都配备了FORTRAN77。例如：APPLE II FORTRAN是FORTRAN77的子集。IBM PC FORTRAN也是FORTRAN77的子集，但也具有一些全集水平的特性。

本章浓缩了FORTRAN77的基本特征，旨在使读者能迅速了解FORTRAN的基本概貌。

§4.1 基本概念

4.1.1 字符集和源程序书写格式

一. 字符集

FORTRAN77中允许使用的字符有：英文字母A~Z，数字0~9以及十三个特殊符号。另外，各具体的计算系统上的版本还可能扩充一些字符。特殊符号及其使用如下：

- (空格) 只有字符常数中的空格才有意义，在语句的其它地方出现时无效。
- ' (撇号) 用作字符常数的起止符。
- \$ (货币号) 在CALL语句中用作自变量，以代表某一语句号。
- (和) 括号；在表达式中，括号内的先计算。在很多场合中都须用它。
- + 算术加，正号；打印时作纵向走纸控制符。
- 减号；负号。
- , 在参数语句、等价语句、复常数和格式控制中用来分隔两个数据。
- / 除号；数据初值语句中数据列表和有名公用区语句中区名的起止号；FORMAT语句中作记录结束符。
- = 等号；赋值号。
- * 乘号；第一列上写上“*”表示此行为注解行；在子例行程序语句的虚变量中用*代表可返回点；连续两个(**)作为乘幂运算符；在输入输出语句中用作隐含设备号；也可用于参数的长度选择。
- . 小数点；逻辑运算符和逻辑常数起止符。
- :

二. 源程序书写格式

一个源程序由若干程序行组成，每一行有80列 (80个字符)，分为四个区：标号区、续行区、语句区和注释区。程序行又分起始行、续行、注释行和结束行。

- 标号区：第1列至第5列为标号区。标号最多为5位数字，数字之间的空格不起作用。
- 续行区：第6列上若为任一非零和非空字符，则该行为程序行续行。续行的标号区须为空白，第7列至第72列为续行正文。一个语句最多可有19个续行。
- 语句区：第7列至第72列为语句区。但可从第7列以后任一列开始。语句较长时可续行。
- 注释区：第73列至第80列为注释区。编译时，对这个区的内容不予翻译。
- 起始行：一个语句的第一行，该行第6列须为空白或零，标号只能在起始行上。
- 续行：指语句的续行。注释行无续行。
- 注释行：第1列上的字符为“C”或“*”时，第2列至第72列便为注释正文，当注释内容较长时，可连续使用注释行。注释行可出现在程序中任一行上。
- 结束行：END语句行。

•41.2 常量、变量和数据类型

一. 常量

- 整型数：〔±〕整数 整数范围随微机字长不同各异。
- 实型数：〔±〕〔整数〕. 整数 如：87.6，-1.76
 〔±〕整数，〔.〔整数₂〕〕E〔±〕整数， 如：87.6E-3
- 复型数：(A, B) 其中，A为实部，B为虚部，两者均为实数。
- 双精度数：格式同实数，但用D代替E，占二个存贮单元。
- 逻辑型数：真(·TRUE·)或假(·FALSE·)
- 字符型数：用撇号(′)括起来的字符串，撇号作串的字符时应双写。如：'CHINA'。

以下是错误的常数：

整常数：15.0；5,000；415E4

实常数：-E7；·E3；e；π；407

字符常数：'abcd'；"abcd"；'IT'S'（应为'IT'S'）

逻辑常数：TRUE；·FALSE

二. 变量

- 变量名：由至多6个字符组成，须由字母打头，这6个字符只能是字母A~Z和数字0~9。变量中的空格无效。同一程序单位中的变量须唯一。

合法：A1；X12345；ABCDEF

非法：A(8)；A/B；3B；M·L；A\$；-p

- 变量的类型：可用类型说明语句显式地说明（如：REAL A, B），也可由变量名的第一个字母隐含地说明为整型或实型。

三. 数据类型的说明

1. 显式说明（用类型说明语句）

- 整型：INTEGER list
- 实型：REAL list
- 双精度型：DOUBLE PRECISION list
- 复数型：COMPLEX list
- 逻辑型：LOGICAL list
- 字符型变量：CHARACTER〔*长度〔, 〕〕变量名₁，变量名₂，…

或 CHARACTER 变量名₁ [* 长度₁], 变量名₂ [* 长度₂], ...

说明:

< 1 > .List是用逗号分隔的变量名、数组名、语句函数名或外部函数名的列表。列表中还可含有数组维数说明。

例: REAL A,B,C (100), D (10,100)

< 2 > 字符变量定义中, 长度为变量的串长度, 省略长度说明时隐含为 1。长度可为大于零的整型表达式。

例: CHARACTER A,B,C

CHARACTER * 9, N1,N2,N3

CHARACTER * 9, M1,M2 * 6,M3,M4 * 4

其中, 字符变量A, B和C的长度均为 1; N1,N2和N3的长度均为 9, M1和M3 的长度均为 9, M2的长度为 6, M4的长度为 4。

2. 隐含说明 (I—N规则)

对那些没用类型说明符显式说明的变量, 将根据变量的第一个字母来隐含说明为整型或实型。凡以I、J、K、L、M、N、中之一开头的变量隐含为整型, 以其它字母打头的隐含为实型。

如: REAL RVE,R1,GOM,MASS,LIMIT

INTEGER SUM,P,BATE,LEVEL,KAPPA

等价于: REAL MASS,LIMIT

INTEGER SUM,P,BATE

因为: RVE、R1、GOM隐含说明为实数, LEVEL和KAPPA隐含说明为整型。

3. 改变隐含类型——IMPLICIT

可用IMPLICIT语句将某一些字母打头的变量指定为任一类型。如:

IMPLICIT INTEGER (A,B,C), REAL (I—M), LOGICAL (D—F)

将以A、B和C打头的变量指定为整型; 将以I、J、K、L和M打头的变量均指定为实型; 将以D、E和F打头的变量均指定为逻辑型。

注意: 类型说明语句说明的类型和IMPLICIT语句说明的类型冲突时, 则以类型说明语句为准。

例: IMPLICIT REAL (F—J)

INTEGER I1,J1

:

X = F1 + I1 + I2 + J1 + J2

这里, I1、J1为整型, I2、J2、F1均为实型。

注意: 这二个语句都只在本程序单位内有效; 并且, IMPLICIT语句应在INTEGER等所有其它说明语句之前出现。

4.1.3 表达式和赋值语句

一. 表达式

1. 运算符及其优先级 (A→B称作A优先于B)

算术运算符: * (乘方) → *, / (除) → +, -

关系运算符: 小于 (·LT·), 小于等于 (·LE·), 等于 (·EQ·) 不等 (·NE

·), 大于(·GT·), 大于等于(·GE·)

逻辑运算符, 非(·NOT·)→与(·AND·)→或(·OR·)→逻辑等(·EQV·)

), 逻辑不等(·NEQV·)

且有: ()→算术运算符→关系运算符→逻辑运算符。

注: 括号具有最高优先级, 同级的运算按从左到右的次序进行。

2. 算术表达式

它可以是变量、常数、函数引用、或由它们与圆括号和算术运算符组成的式子。

3. 字符表达式

它可以是单个的字符常数、字符变量, 或二者与连接符(//)组成的式子。

如: (设A1和B1均为字符变量):

'ABC', 'ABC' //'XYZ', A1, A1 //'UVW', B1(1:7) //'123'

4. 关系表达式

(1) 算术关系表达式

它是由算术表达式和关系运算符组成的式子。如:

1.3E6.GT.1.2*6*SQRT(A), (3*SIN(X))/3.1.LT.6

(2) 字符关系表达式

它是由字符表达式和关系运算符组成的式子。如:

'BOY'.GT.'GIRL', A1.LT.'UVW'

关系表达式根据比较的结果, 将得到一逻辑值(·TRUE·或·FALSE·)——真或假。

算术关系表达式按数值的大小来比。字符关系表达式的比较原则为: 按字符的ASCII码大小来比, 被比较的两个串按从左到右的次序——对应比较, 如: 第一次均用左边第一个字符比; 比较时, 若所有字符对相等, 则串长者大; 一旦遇到不等字符对, 字符的ASCII码大者, 其对应的串大。如:

'ABCD'.GT.'ABC'(真), 'ABCD'.LT.'ABDE'(真)

5. 逻辑表达式

它可以是单个的逻辑常数、逻辑变量、关系表达式, 或由三者与括号和逻辑运算符组成的式子。设A为逻辑变量, 则以下都是合法的式子:

.TRUE., A.OR..TRUE., .NOT.A,

逻辑表达式的计算原则为: 先求算术表达式之值, 再求关系表达式之值, 最后按逻辑运算符的优先级求出逻辑表达式之值。

二. 赋值语句

这里介绍的是算术赋值语句, 其格式为: $V = e$

其中, e为算术表达式, V为数字型变量。

执行规则是: 计算出e的值, 若值与V的类型相同便直接赋给V, 若值的类型与V的类型不同, 则先转换成与V相同的类型, 再赋给V。

类型转换规则为: 若V为整型, 则将e的值截去小数部分化为整数, 若V的复数型, 则先将e的值变成实数作为复数的实部, 虚部为0.0; 若V为双精度型或实型, 则e的值分别转换为双精度型或实型。

例: 设A为整型变量, B为实型, C为复型, D为双精度型, 则以下赋值语句执行后, 结果显示花括号内所示:

A = 456.789	{ 456 }	A = 0.456D2	{ 45 }
B = 999	{ 999.0 }	B = 0.123456789123D2	{ 12.3456789 }
C = 3	{ (3.0, 0.0) }	C = 1.6392D + 2	{ (163.92, 0.0) }
D = 444	{ 0.444D3 }	D = (1.2, 3.4)	{ 1.2D0 }

另外，还有逻辑赋值语句和字符赋值语句，格式也为： $V = e$

若 e 为字符表达式，则 V 须为字符型变量。

若 e 为逻辑表达式，则 V 须为逻辑型变量。

4.1.4. 数组

数组是一些类型相同的数据的序列。在FORTRAN中，**DIMENSION**语句专门用来定义一个数组。另外，在类型说明语句中或**COMMON**语句中也可定义一个数组，后一种语句将在后面予以介绍。

一. 数组的定义

格式1：**DIMENSION** 数组说明符1，数组说明符2，…

2：类型说明符 数组说明符1，数组说明符2，…

其中，数组说明符为：

数组名 ([下界1:] 上界1, …, [下界n:] 上界n)

说明：FORTRAN77的数组最多可到7维，即 $n \leq 7$ ；下界小于上界，省略下界时隐含为1；在同一**DIMENSION**语句中可同时说明多个数组，这些数组的类型可显式的或隐含地说明，格式2可同时说明几个具有同一类型的数组。

例：**REAL M, M1**

INTEGER T, T1

DIMENSION T(50), M(20), T1(5, 20), M1(10, 10, 10)

等价于：**INTEGER T(50), T1(5, 20)**

REAL M(20), M1(10, 10, 10)

此例说明：M为实型数组，有20个元素；T为整型数组有50个元素；M1为实型三维数组，共有 $10 \times 10 \times 10 = 1000$ 个元素；T1为整型二维数组，有 $5 \times 20 = 100$ 个元素

在数组说明符中的上界或下界可为：整常数、整符号常数或整型表达式。

如：**PARAMETER (L1 = 4, L2 = 5)** (参数语句)

DIMENSION A(L1, L2), B(L1, L2)

等价于：**DIMENSION A(4, 5), B(4, 5)**

这里的L1和L2为符号常数符，当用新的参数(PARAMETER)语句赋于L1或L2一个新的值时，则数组A和B便获得新的定义。可见参数语句对常数的修改十分方便。参见4.4.2一节。

二. 数组元素的引用

引用一个数组元素，只需指定数组的名字和该元素在各维中的序号即可，各维的序号是从下界开始，按增量1递增直到上界。亦即，引用一个元素，应使用以下格式的数组元素名(或称下标变量)：

数组名 (下标表达式1, …, 下标表达式n)

说明：括号中下标的个数应与该数组定义的维数相同，下标表达式为任意合法的整型算术表达式，其值应在相应上下界范围内。

例：设有：INTEGER A(2, 3)

将整数序列1、2、3、4、5、6在数组A的存储区中依次存放，各元素的引用关系为：

A(1,1)=1; A(2,1)=2; A(1,2)=3; A(2,2)=4; A(1,3)=5; A(2,3)=6 企图用以下的下标变量来引用A的元素是非法的：

A(1,0), A(1,4), A(0,1), A(3,1), A(3,3)

从以上数组元素的引用关系可以看出，数组在内存中是“按列”存放的。形式如下：

INTEGER	A(2, 3)	INTEGER	B(2, 2, 2)
1.	A(1, 1)	1.	B(1, 1, 1)
2.	A(2, 1)	2.	B(2, 1, 1)
3.	A(1, 2)	3.	B(1, 2, 1)
4.	A(2, 2)	4.	B(2, 2, 1)
5.	A(1, 3)	5.	B(1, 1, 2)
6.	A(2, 3)	:	

通常，人们总是习惯于按行来提取数组元素：A(1,1)、A(1,2)、A(1,3)、A(2,1) A(2,3)，结果取到的可能不是期望的元素。为了避免混乱，可使用DO语句或隐含DO循环来控制数组元素的下标变化，按某一循环规律输入数组元素，尔后在使用各数组元素时也采用同一循环规律。

参见4.5.1一节中关于数组的输入输出与隐含DO循环的说明，参见4.2.4一节中关于“数组与循环”的片断。

三. 可调数组和假定大小数组

在数组说明符中，如果上界、下界为整常数、整符号常数或由整常数构成的表达式，则此数组为常界数组。若上界、下界为整型变量、或含整型变量的表达式，则此数组为可调数组。可调数组的大小(元素个数)是不定的，其大小由调用程序中对应的某实整变量来确定。

在数组说明符中，若最后一维的上界是*号，则此数组为假定大小数组，此类数组的大小由调用程序中对应的实数组的大小来确定。

可调数组和假定大小数组只能出现在子程序中，在调用子程序时，通过虚实结合办法，使这两种虚数组具有确定的元素个数。

如：DIMENSION DA(-3:I, 2:J, -1:K)

REAL DB(50, *), DC(-5:I, J, *)

其中，DA为三维可调，DB为假定大小的数组。DC为假定大小而又可调的数组。

在4.3.3一节中详细叙述了这两种数组的虚实结合及其应用。

4.1.5 子字符串

一个字符变量中可存放一个字符串，一个字符数组元素也可放置一个字符串。字符串中的一个连续部分称为子字符串。使用时，可根据要求存取整个字符串或其中的一个子字符串(简称子串)。使用子字符串变量来标识并获得一个子串。

子字符串变量的格式：

V(e₁:e₂)

其中，V为字符变量名或字符数组元素名。e₁为子串的下界，e₂为子串的上界，二者均为整型算术表达式。且有：

(1) . $1 \leq e_1 \leq e_2 \leq N$, N 为字符变量或字符数组元素的长度。

(2) . 缺省 e_1 时, 隐含为 1。如: $A(: 6)$ 与 $A(1 : 6)$ 是等价的。

(3) . 缺省 e_2 时, 隐含为 N 。如: 若字符变量 X 的长度为 20, 则 $X(12 :)$ 与 $X(12 : 20)$ 是等价的。

例: 设: 字符变量 A 的值为 'ABCDEF', 字符数组 $B(3)$ 的三个元素分别为: 'GHIJKL', 'MNOPQR', 'STVWXY'

则有: $A(2 : 4)$ 表示 'BCD'

$A(2 :)$ 表示 'BCDEF'

$A(:)$ 表示 'ABCDEF'

$B(3) (5 : 5)$ 表示 'X'

$B(2) (3 : 4)$ 表示 'OP'

(4) . 利用赋值语句 (或输入语句) 给予字符串赋值。如: 执行以下语句:

```
CHARACTER CH1*18, CH2( 20 ) * 6
```

```
CH1( : 7 ) = 'FORMULA'
```

```
CH1( 8 : 12 ) = 'TRANS'
```

```
CH1( 13 : ) = 'LATION'
```

```
CH2( 1 ) ( : 4 ) = 'ABCD'
```

```
CH2( 1 ) ( 5 : ) = 'EF'
```

则, $CH1$ 中的字符串为: FORMULATranslation

$CH2$ 的第一个元素中的字符串为: ABCDEF

(5) 子字符串变量也可出现在字符表达式中作连接运算。如: 执行以下语句:

```
CHARACTER A*13, B*15
```

```
A = 'I HAVE A BOOK'
```

```
B = A( : 9 ) // 'PENCIL'
```

则 B 中的字符串为: 'I HAVE A PENCIL'

注: 字符赋值时, 等号右边的字符表达式的值的长度 (字符个数) 按左对齐的原则存入等号左边的字符变量, 多余部分被截掉, 不足则在字符串右边补以空格。取出时, 右边补的空格被忽略。如:

```
CHARACTER *10, X, Y, Z
```

```
X = 'ABCDEF'
```

```
Y = '1 2 3 4 5 6 7 8'
```

```
Z = X//Y
```

执行后, X 中的字符为 'ABCDEFUUUU', Y 中为 '12345678UU', Z 中为: 'ABCDEF1234'。

4.1.6 内部函数

FORTRAN77 有三类函数: 内部函数、语句函数和外部函数 (FUNCTION 函数子程序)。前者由编译系统提供, 后两者由用户自己定义。所有的函数有相同的引用形式:

函数名 (参数 1, ... 参数 n)

内部函数是一些预先用机器语言编好的标准子程序, 如, $\cos(X)$ 、 $\log(X)$ 、 $\text{ABS}(X)$ 、 $\text{INT}(X)$ 、 $\text{SQRT}(X)$ 等等。它们本质上属于编译程序的一部分。用户引用内部