

王亚勇 编著

FORTRAN

77

程序设计语言

冶金工业出版社

FORTRAN 77程序设计语言

王 亚 勇 编 著

351056

冶金工业出版社

内 容 提 要

本书系统介绍FORTRAN 77程序设计语言,运用通俗的语言解释语法规则并提供程序设计的实例。可供科学研究、工程技术、企业事业管理各部门的程序设计人员学习参考和作为程序设计语言手册使用,也可作为大专院校师生的教学参考书。

FORTRAN 77程序设计语言

王亚勇 编著

冶金工业出版社出版

(北京北河沿大街嵩祝院北巷39号)

新华书店北京发行所发行

冶金工业出版社印刷厂印刷

787×1092 1/16 印张 13 字数 304 千字

1985 年 5 月第一版 1985 年 5 月第一次印刷

印数00,001~35,000册

统一书号: 15062·4226 定价2.40元

前 言

FORTRAN算法语言是目前国际上广泛流行的一种计算机高级语言。由于它具有标准化程度高、程序结构灵活、可移植性强和运算速度快等优点，在计算机软件工程中起着十分重要的作用。几乎所有数字电子计算机上都配备有FORTRAN语言。在科学技术领域，越来越多的人喜欢使用FORTRAN语言编制程序解决科研设计中的课题。这些年来，由于社会发展的需要和FORTRAN语言本身的不断改进发展，在许多非计算领域，例如：经济管理、行政管理、商业服务、情报信息处理……，FORTRAN语言也得到了广泛的应用。

在六十年代和七十年代中期，国际上所通用的FORTRAN算法语言称为FORTRAN IV，基本上是1966年美国国家标准学会所发表的ANSI, X3.9—1966 FORTRAN，也就是1972年的FORTRAN国际标准。目前，在我国大部分电子计算机上所使用的就是FORTRAN IV。

经过十多年的实际运用，FORTRAN语言和各種电子计算机的操作系统有了很大的发展。为了将这些新的发展标准化，以提高FORTRAN程序的可移植性，美国国家标准学会对上述ANSI, X3.9—1966 FORTRAN，即FORTRAN IV进行了修订，于1978年4月3日发表了“美国国家标准程序设计语言FORTRAN(全集)”ANSI, X3.9—1978，并命名为FORTRAN 77。目前，FORTRAN 77正在逐步取代沿用的FORTRAN IV。国外许多新型的电子计算机（包括微处理机）一般都配有FORTRAN 77程序设计语言。某些早期的计算机上，也在逐步采用以FORTRAN 77为基础的FORTRAN语言新版本更新原有的FORTRAN IV旧版本。

近年来，在我国，一些新型的国产电子计算机上已开始配备FORTRAN 77语言。在引进的许多中小型和微型电子计算机上，也都带有FORTRAN 77。由于FORTRAN 77比FORTRAN IV具有更广泛的功能和更高的效率，可以预料，在不久的将来，在国产电子计算机上将会普遍使用FORTRAN 77程序设计语言。为了配合FORTRAN 77程序设计语言在我国的推广应用，特编写本书，供广大程序设计人员和大专院校师生学习参考。

本书所述的语法规则的主要依据是“美国国家标准程序设计语言FORTRAN(全集)”ANSI, X3.9—1978。为了解释语法条文的方便和提供程序设计实例，又以美国数字设备公司(DEC)的PDP—11系列机所用的FORTRAN 77语言手册(1981年9月版)为主要参考书。因此，本书所述的语法规则中带有某些DEC所增加的改进了的FORTRAN 77的内容，对这部分语法规定，本书统一用星号“*”在有关条文的左上角标记。

考虑到FORTRAN 77的语言规定比较严格，语言抽象繁琐，本书在编写过程中，尽量使用通俗的语言加以解释并提供实例说明。书中所举的程序实例一般都经过上机考验，可供学习和运用。但是，有时为了语法解释的需要，有意迎合语法条文规定，或者为了便于读者理解，某些实例可能不够简洁合理，读者完全可以改写成更为有效的形式。

本书所叙述的FORTRAN 77程序设计语言是一个完整的系统。对于具有初步FORTRAN语言知识，特别是了解FORTRAN IV语言并有某些程序设计经验的读者，学习和

使用本书所述语法，应该不会有太大困难。初学者阅读本书时，只要循序渐进，学用结合，相信也能掌握要领。对于使用PDP-11系列电子计算机的读者，则可将本书做为程序设计语言手册使用。

本书编写过程中，编者力图保持语言手册的系统性和语法定义的严密性，同时避免语法手册里语言的抽象和枯燥，使叙述通俗易懂，便于学习。但是，由于编者水平有限，对语法的解释难免带有片面性，甚至错误，所提供的实例也不尽完美准确，请读者指正。

本书编写出版，得到中国土木工程学会计算机应用学会缪兆杰同志和冶金工业部建筑研究总院张洪钧同志的关心和支持，并由缪兆杰同志审阅了全稿，在此特致谢意。

编 者

一九八三年十月

目 录

第一章 绪论	1
第一节 简介	1
一、FORTRAN 77完全语言ANSI, X3.9—1978的主要特点	1
二、数字设备公司(DEC)所做的改进	1
第二节 程序组成	2
一、语句	3
二、注解	3
三、字符	3
第三节 程序行的格式	4
一、每列一字方式	4
二、制表键方式	5
三、语句标号区	5
四、续行区	6
五、语句区	6
六、序号区	6
第四节 程序结构	7
*第五节 INCLUDE语句 ^①	8
第二章 语句成分	10
第一节 符号名	10
第二节 数据类型	11
第三节 常数	12
一、整型常数	12
二、实型常数	13
三、双精度常数	14
四、复型常数	14
*五、八进制和十六进制常数	15
六、逻辑型常数	17
七、字符常数	17
*八、霍尔瑞斯常数(Hollerith)	18
第四节 变量	20
一、数据类型的指定	21
二、隐含的数据类型	21
第五节 数组	21
一、数组说明符	22

① 本书中的语法内容,凡属于数字设备公司(DEC)所增加的改进了的FORTRAN 77,均于标题左上角以星号“*”标注。

二、下标	25
三、数组的存贮	25
四、数组的数据类型	25
五、无下标数组引用	26
六、可调数组	27
第六节 字符子串	27
第七节 表达式	28
一、算术表达式	28
二、字符表达式	32
三、关系表达式	32
四、逻辑表达式	33
第三章 赋值语句	36
第一节 算术赋值语句	36
第二节 逻辑赋值语句	37
第三节 字符赋值语句	38
第四节 标号赋值 (ASSIGN) 语句	39
第四章 控制语句	42
第一节 转移 (GOTO) 语句	42
一、无条件转移语句	42
二、计算转移语句	43
三、赋值转移语句	44
第二节 条件 (IF) 语句	45
一、算术条件语句	45
二、逻辑条件语句	47
三、块条件语句	48
第三节 循环 (DO) 语句	53
一、DO循环控制	54
二、嵌套DO循环	56
三、在DO循环内的控制转移	56
*四、扩展的循环范围	57
五、DO循环的实例	57
第四节 继续 (CONTINUE) 语句	59
第五节 调用 (CALL) 语句	60
第六节 返回 (RETURN) 语句	60
第七节 暂停 (PAUSE) 语句	62
第八节 停止 (STOP) 语句	62
第九节 结束 (END) 语句	63
第五章 说明语句	64
第一节 隐指 (IMPLICIT) 语句	64
第二节 类型说明语句	65
一、数字类型说明语句	65

二、字符类型说明语句	65
第三节 维数 (DIMENSION) 语句	67
第四节 公用 (COMMON) 语句	67
*第五节 虚数组 (VIRTUAL) 语句	70
一、应用虚数组的限制	71
二、在辅程序中虚数组的引用	72
第六节 等价 (EQUIVALENCE) 语句	73
一、数组等价	74
二、字符串等价	75
三、扩展公用块	76
第七节 保存 (SAVE) 语句	78
第八节 外部 (EXTERNAL) 语句	78
第九节 内部 (INTRINSIC) 语句	79
第十节 数据 (DATA) 语句	81
第十一节 参数 (PARAMETER) 语句	83
第十二节 程序 (PROGRAM) 语句	84
第十三节 数据块 (BLOCK DATA) 语句	84
第六章 函数和子程序	86
第一节 函数和子程序参数	86
一、函数和子程序参数的有关规定	87
二、可调数组	90
三、虚构大小 (ASSUMED-SIZE) 的哑数组	91
第二节 语句函数和辅程序	93
一、语句函数	93
二、函数辅程序	95
三、子程序辅程序	97
四、入口语句	99
第三节 内部函数和其它库函数	102
一、内部特定函数调用	102
二、内部同类函数调用	102
三、内部同类函数的应用实例	103
四、字符库函数和词法比较库函数	106
第七章 输入输出语句	109
第一节 输入输出 (I/O) 概述	110
一、记录 (Records)	110
二、文件	111
三、内部文件	112
四、存取模式	112
第二节 输入输出 (I/O) 语句成分	113
一、控制信息表	113
二、输入输出 (I/O) 表列	118

第三节 读 (READ) 语句	120
一、顺序读语句	120
二、直接存取读语句	125
*三、索引读语句	126
四、内部读语句	127
第四节 写 (WRITE) 语句	128
一、顺序写语句	129
二、直接存取写语句	131
*三、索引写语句	132
四、内部写语句	133
*第五节 重写 (REWRITE) 语句	134
*第六节 ACCEPT 语句	135
*第七节 TYPE 和 PRINT 语句	135
*第八节 编码 (ENCODE) 和解码 (DECODE) 语句	136
第八章 格式语句	139
第一节 场和编辑描述符	140
一、BN编辑描述符	140
二、BZ编辑描述符	141
三、SP编辑描述符	141
四、SS编辑描述符	141
五、S 编辑描述符	141
六、I 场描述符	141
*七、O场描述符	142
*八、Z场描述符	144
九、F场描述符	145
十、E场描述符	147
十一、D场描述符	148
十二、G场描述符	149
十三、L场描述符	150
十四、A场描述符	151
十五、H场描述符	153
十六、X编辑描述符	155
十七、T编辑描述符	156
十八、TL编辑描述符	157
十九、TR编辑描述符	157
*二十、Q编辑描述符	158
*二十一、美元符号 (\$) 编辑描述符	159
二十二、冒号 (:) 编辑描述符	159
二十三、比例因子 nP	160
二十四、复型数据编辑	162
二十五、重复次数和成组重复次数	163
二十六、隐指的场描述符	163

*第二节 可变格式表达式	164
*第三节 回车控制符	165
第四节 格式说明分隔符	165
第五节 外部场分隔符	166
第六节 实时格式	167
第七节 格式控制与I/O表列的相互作用	169
第八节 格式语句的规则	171
一、一般规则	171
二、输入规则	172
三、输出规则	173
第九章 辅助的输入输出语句	174
第一节 文件打开(OPEN)语句	174
一、存取 (ACCESS)方式	179
二、关联变量 (ASSOCIATEVARIABLE)	180
三、空格处理 (BLANK)	180
四、缓冲器大小 (BLOCKSIZE)	180
五、缓冲器数量 (BUFFERCOUNT)	181
六、回车控制 (CARRIAGECONTROL)	181
七、文件处置 (DISPOSE)	181
八、错误出口 (ERR)	182
九、文件扩展 (EXTENDSIZE)	182
十、文件名 (FILE)	182
十一、文件记录格式 (FORM)	182
十二、文件块数分配 (INITIALSIZE)	183
十三、键场指定 (KEY)	183
十四、最多记录数 (MAXREC)	184
十五、文件名 (NAME)	184
十六、不跨越磁盘块规定 (NOSPANBLOCKS)	184
十七、文件结构 (ORGANIZATION)	184
十八、写保护 (READONLY)	184
十九、记录长度 (RECL)	184
二十、记录大小 (RECORDSIZE)	185
二十一、记录形式 (RECORDTYPE)	186
二十二、分享规定 (SHARED)	187
二十三、文件状态 (STATUS)	187
二十四、文件类型 (TYPE)	187
二十五、逻辑部件 (UNIT)	188
二十六、用户外部函数打开文件 (USEROPEN)	188
第二节 文件关闭(CLOSE)语句	188
第三节 返回文件开头(REWIND)语句	189
第四节 返回记录开头(BACKSPACE)语句	189
*第五节 删除(DELETE)语句	190

一、顺序删除语句	190
二、直接删除语句	190
*第六节 解开(UNLOCK)语句	191
第七节 文件结束(ENDFILE)语句	191
附录一 FORTRAN 77与FORTRAN IV已知的差别	192
附录二 ASCII码表	193
附录三 FORTRAN 77库函数表	195

第一章 绪 论

第一节 简 介

本书所介绍的 FORTRAN 77 程序设计语言包括了美国国家标准程序设计语言 FORTRAN 77子集语言 ANSI, X3.9—1978、FORTRAN 77完全语言和数字设备公司 (DEC) 所提供的经改进的 FORTRAN 77子集标准。

由美国计算机和商务设备制造者协会秘书处和美国国家标准协会1978年4月3日批准的美国国家标准程序设计语言 FORTRAN (全集) 包括了上述 FORTRAN 77子集语言 ANSI, X3.9—1978 和 FORTRAN 77完全语言, 是对 FORTRAN 语言ANSI, X3.9—1966, 即人们通常称之为 FORTRAN IV的修订。在本书附录一中列举了二者之间的主要差别, 供学过FORTRAN IV的读者对照。下面再将本书介绍的 FORTRAN 77语言所具有的主要特点及DEC公司所做的改进分别归纳如下。

一、FORTRAN 77完全语言ANSI, X3.9—1978的主要特点

1. 使用双精度型和复数型数据。
2. 拥有某些内部函数, 例如: 求字符表达式长度的 LEN 函数、求某一个参数的 ASCII值的 ICHAR 函数、求字符子串在字符表达式里的位置的 INDEX 函数等。
3. 使用双精度数据来表示指数。
4. 使用格式编辑描述符 S、SP、SS、T、TL 和 TR。
5. 定义的库函数具有同类名字, 即属名 (Generic) 和用于不同数据类型的特定名字。
6. 运用实型或双精度型变量作为循环语句的控制变量。
7. 运用整型、实型或双精度型数字表达式做循环语句的初值、步长和终值。
8. 使用 OPEN 和 CLOSE 语句来打开和关闭数据文件。
9. 在读 (READ) 语句和写 (WRITE) 语句中, 使用 ERR=S 的说明, 当有读或写错误发生时, 可以将运算和控制转移到 S 语句。
10. 运用直接表列式 (表控格式) 输入输出 (I/O) 语句去执行有格式的 I/O, 而不需要指定相应的格式说明。
11. 在 WRITE、REWRITE 和 PRINT 语句的 I/O 表列中, 可以使用常数和表达式。也就是说, 可以使用上述语句输出常数值和表达式的值。
12. 在数组说明中, 可以对数组的维规定一个“下界”。
13. 在子程序、辅程序和函数辅程序中, 应用入口 (ENTRY) 语句来指定几个入口。
14. 应用参数语句对一个符号名指定一个常数。

二、数字设备公司 (DEC) 所做的改进

1. 增加了 LOGICAL * 1 和 LOGICAL * 2 两种不同长度的逻辑型数据。

2. 增加了如下输入输出 (I/O) 语句:

(1) 对特定设备的I/O:

ACCEPT 从终端输入

TYPE 由终端输出

(2) 无格式直接存取I/O:

READ (u'r)

WRITE (u'r)

FIND (u'r)

(3) 有格式直接存取I/O:

READ (u'r, fmt)

WRITE (u'r, fmt)

(4) 文件控制和属性说明:

DEFINE FILE

(5) 在存储器里进行的有格式数据转换:

ENCODE

DECODE

(6) 索引I/O:

READ (u, f, key)

READ (u, key)

(7) 记录控制和更新:

REWRITE

DELETE

UNLOCK

3. 在任何语句之后使用惊叹号 ! , 可以增加一个注解。

4. 可以使用调试语句。在程序中, 如果语句的第一列标以D字母, 那末, 在程序编译时, 运用特定的开关, 将该句作为调试语句, 否则当作一般的注解行。

5. 在一个计算转移语句 (见第四章第一节) 里, 可以用任何算术表达式作为控制参数。

6. 可以使用虚数组, 在程序可编地址空间之外开辟很大的数据区。

7. 在OPEN、CLOSE、FIND、DELETE、UNLOCK、BACKSPACE、REWIND或ENDFILE语句里, 使用一个ERR=S的开关, 当一个错误发生时, 可以将控制转到标号为S的语句去。

8. 使用INCLUDE语句, 可以将一些FORTRAN语句从一个分开的文件合并到一个FORTRAN程序里去。

9. 增加了INTEGER*4数据类型, 指定一个包括符号位和31个二进制位的整型数。

10. 可以用八进制和十六进制常数来代替任何数字常数。

第二节 程序组成

FORTRAN程序包括语句和注解两大部分。语句是决定计算步骤的, 注解则仅起注

释作用，不影响程序的运行。每个程序必须由一个END语句作为结束。一个可执行程序必须有一个主程序和一个或几个辅程序，也可以没有辅程序。主程序或辅程序通称为程序单位。程序的组成如下面一个简单的例子所示。

程序单

```
C  * * * * *
C  PROGRAM FOR CALCULATION OF SUM
C  OF INTEGER SERIES 1 TO 100
C  * * * * *
      REAL SUM
      SUM=0.
      DO 10 I=1,100
10    SUM=SUM+I
      OUTPUT THE RESULT
      TYPE *, 'SUM= ', SUM
      END
```

上面这个程序只有主程序，开头四行的第一列以C字母当头，第九行以惊叹号！当头，因此都为注解行，其余为语句。

一、语句

语句分两大类：一类是可执行，另一类是不可执行语句。可执行语句规定程序的运行；不可执行语句描述数据排列和特征，提供编辑和数据变换的信息。上面那个程序中，第一个语句：REAL SUM是不可执行语句，其余的语句是可执行语句。

语句写成行，一行为一串字符，每行最多不超过80个字符，如果超过，可以续行，续行用一个续行号，写在第六列表示（见本章第三节）。

在程序中可以用一个标号来标志一个语句，使其它语句可以引用它，或将控制转给它，或得到它所包含的信息。语句标号是一个整数，放在一个语句初始行的第1~5列。任何语句都可以有标号，但只有可执行语句和格式语句可以被引用。

二、注解

注解不影响程序的运行，仅仅是一种辅助工具，可用于描述程序的作用，标明程序节、过程及阅读源程序方便，或者根据程序编写者的其它目的所加的任何文字信息。用一个字母C或星号*放在一个源程序行的第一列表示该行为注解行。另外，可以在一个程序行的除了第六列之外的任何一列置一个惊叹号！，惊叹号之后的内容即作为注解。（如果放在某行的第六列，则该行被认为是续行。）

在注解行里可以使用任何可以打印的字符。

注解行如第二节程序组成中的例子所示。

三、字符

FORTRAN 77使用下列字符，这些字符均可以用来组成一个源程序：

1. 所有大、小写的英文字母（A到Z，a到z）。

2. 数字 0 到 9

3. 某些特定的符号, 如:

空格		逗号	,
制表键	TAB	句号	.
等号	=	单引号	'
加号	+	双引号	"
减号	-	美元符号	\$
乘号	*	惊叹号	!
除号	/	冒号	:
左括号	(	左尖括号	<
右括号	)	右尖括号	>

4. 其它可打印的 ASCII 符号。但是, 这种字符只能作为语句里的字符常数。对这些字符, FORTRAN 77 编译程序要区分它们的大小写, 而对上述三种字符, 则不必区分大小写。

ASCII 字符是美国标准信息交换码 (见附录二)。

第三节 程序行的格式

上节叙述了程序行所用的字符形式, 并规定了每行最多有 80 个字符。本节将叙述程序行的格式, 也就是每个程序行的组成部分。

每个程序行的 80 个字符分放在第 1~80 列。这 80 列可以分为四部分, 或四个区:

1. 第 1~5 列, 语句标号区;
2. 第 6 列, 续行标志区;
3. 第 7~72 列, 语句区;
4. 第 73~80 列, 序号区。

可以用两种方式来书写并输入一个程序行:

1. 每列一字方式。这种方式是逐字打进字符和必要的空格, 形成一行程序。
2. 每列一字并使用制表键 TAB 从一个区跳到另一个区。

应用第一种方式可以把源程序记在穿孔卡片上或者由终端键盘打入源程序。第二种方式仅适用于由终端键盘打入源程序。

一、每列一字方式

假定有以下四行程序要书写并输入计算机:

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
C          F I R S T   L I N E
1 0          I = J + 5 * K + L * M
          P = 4 . 2 5 6 * * I - 3 . 5
          1   * X / Y 2
          R E S T = 1 . + 2 .
```

每列一字方式就是逐字书写并输入各行的字符和空格。

二、制表键方式

这种方式只用于由终端键盘输入源程序。所谓终端通常是一个带打字键盘的屏幕显示器，可用于向计算机打入命令，输入源程序和数据。使用终端键盘输入源程序时，常用到所谓制表键TAB，这个键一般在键盘左边。每打一次制表键，屏幕显示器上的光标会向右移动八格位置，而打其它字符或空格键，光标只向右移动一格。下面我们按顺序在键盘上打入上述的四行程序（制表键表示为(TAB)）：

```
C(TAB) FIRST LINE  
10(TAB) I = J + 5 * K + L * M  
(TAB) P = 4 . 2 5 6 * * I - 3 . 5  
(TAB) 1 * X / Y 2  
(TAB) REST = 1 . + 2 .
```

一般的规则是：在每一个程序行开头，打入第一个TAB键后，便可以进入续行区或语句区。如果程序行是续行，则在该行的续行标志区打入任何数字（1~9），数字之后，即可打入语句。如果不是续行，则在第一个TAB键之后，便可以直接进入语句区，打入语句内容。这样做，不会将语句的字符与续行标志混淆，因为没有一个是FORTRAN 77的语句是以数字开头的。习惯上，在续行标志与后面的语句内容之间要加上一个或几个空格，以便阅读源程序。

利用TAB键来编排语句，主要目的是使程序语句整齐美观，它在语句里所形成的空格将被编译忽略，但是打印源程序时将被打印出来（即被空出来）。当一个源程序编排时，如果连续打入1、2、3……个TAB键，随后紧接的字符将被打印在第9、17、25……列上。

三、语句标号区

在语句标号区里书写语句标号。一个语句标号由1~5个十进制整数组成，空格和第一个数字前面的0被忽略，不能使用全部为0的语句标号。

被别的语句所引用的任何语句必须要有标号。在同一个程序单位里不允许对一个以上语句定义同样的标号。但是，因为主程序及有关的辅程序被看作不同的程序单位，因此可以在主程序与辅程序里分别使用同样的标号来定义语句。

在语句标号区的第一列还可以使用注解标志符C及*和调试语句标志符号D，当然也可以在语句标号区里使用惊叹号！来标识一行注解。

一个续行的语句标号区必须是空白的，也就是说续行不能带语句标号，有关续行标识符的用法见本节四。

1. 注解标志符

将C或*符号放在某程序行的第一列，表示该行为注解句；用惊叹号！放在某程序行除第6列之外的任何一列，表明该行在！之后的内容为注解。因此，如果惊叹号！被放在第一列，其作用和C、*一样，表示该行是注解行。全部是空格的一行表示一个空格注解。

注解标志符的具体用法可参见本章第二节程序组成的例子。

*2. 调试语句标志符

FORTRAN 77提供一种调试语句,它可以是任何合法的FORTRAN语句,但是,这种语句只有在编译时加上特定的开关才被编译和执行,否则仅作为注解句。

编写程序时,将调试语句放在程序中所需要的地方,程序运行时可以作某些为调试所需要的运算,输出某些提示信息或运算的中间结果,使你了解程序运行的情况及计算结果理想与否等,一旦程序通过后不再需要这些调试步骤和中间结果时,就将这些调试语句变为注解句,不再影响程序的运行。

将D字符放在某程序行的第一列就使该语句成为一个调试语句。一个调试语句可以带有标号,它的标号必须写在第2~5列,跟在D字符之后;如果调试语句有续行,所有续行都必须在第一列写上字符D,第六列写上续行标志。

使用如下编译命令使调试语句作为语句被编译:

```
> F77 PROGRAM/DE=PROGRAM
```

其中,PROGRAM表示源程序的名字。如果不加DE开关,编译命令成为:

```
> F77 PROGRAM=PROGRAM
```

则调试语句不被编译,作为注解句,不影响程序的运行。

四、续行区

续行区在一个程序语句的第六列,在续行区使用续行标志符来表示该行是一个续行。如果该程序行是以每列一字方式写入的,续行标志符可以是除空格和0以外的任何字符(包括字母或数字);如果该程序行是以制表键方式写入的,续行标志符只能用除0以外的任何数字(1~9),紧跟在第一个(TAB)之后。

一个语句如果太长,可以从任何地方被分开而带有续行(但是,END语句不能有续行)。一个语句最多只能带有19个续行,*如果续行多于19个,则必须使用如下编译开关:

```
> F77 PROGRAM/CO:n=PROGRAM
```

其中,n可以从0到99。

注解不可以带续行,实际上每个注解行前面都得带有注解标志符。注解行可以放在语句的某一行和它的续行之间,也可以放在不同的续行之间。

五、语句区

语句区包含源程序的主要内容,由一个程序行第7~72列里的字符组成,超过72列的内容在编译时被忽略,而没有任何提示信息打印出来。因此,如果源程序行太长,将会带来各种无法预料的错误,必须引起注意。在某些型号的计算机终端上,当编写源程序的程序行书写超过某一列时会有警告音响提示程序员,以避免上述错误。

六、序号区

为了编排和查找源程序句子的方便,通常可以将这些句子编号。这些编号可以放在一个程序行的第73~80列里,它只标识源程序句子的顺序。序号区里也可以放其它识别信息。编译程序时忽略这些序号和信息,实际上,序号区里的内容对源程序的运行不起任何作用。

在序号区里不能使用TAB键。

本节详细叙述了程序行的格式,编写一个FORTRAN程序时应特别细心,严格按照这