

实用 FORTRAN-77

段银田等 编著

程序设计

郑州工学院计算中心

前 言

本教材是根据段银田同志多年来的教学讲稿编写而成的，郎洪同志起草了第十章和第十一章，李鹏同志参加了本教材的定稿工作。现对本教材作以下几点说明：

- 1、本教材是根据FORTRAN—77标准编写的，必要时也对其它标准作了少量说明。
 - 2、本教材是通过在各种PC机上使用F—77的各种子集经多次实践而形成的。
 - 3、选材力求精练，实用，以适应减少学时的需要。
 - 4、叙述力求通俗，概念力求准确透彻。
 - 5、为使读者能较快地掌握FORTRAN程序设计方法和便于上机实践，特增加了第十一章上机操作的有关内容和附录C—错误信息表。
 - 6、本教材未选择较大的程序示例，也未讲述常用的数值方法。这些内容可在作者的《实用数值方法的程序设计》教材中找到。
 - 7、本教材可以单独使用，也可以和《实用数值方法的程序设计》合为一门课讲授。
- 由于定稿时间仓促，一定有许多不当之处，敬请指正。

编 者

一九八六年八月二十日

目 录

第一章	FORTRAN基本知识	(1)
§1、	F程序的特点及书写规定	(1)
§2、	F语言字符集、输入及程序实现	(3)
§3、	常数和变量	(4)
§4、	标准函数	(6)
§5、	算术运算符与算术表达式	(7)
第二章	F程序初步	(9)
§1、	计算赋值语句	(9)
§2、	参数说明语句	(10)
§3、	输入/输出的初步概念	(10)
§4、	常用输入/输出格式符介绍	(14)
§5、	停语句与暂停语句	(19)
§6、	例题	(19)
第三章	控制语句	(21)
§1、	无条件转移语句(GO TO)	(21)
§2、	算术条件语句	(21)
§3、	关系条件语句(简单逻辑条件语句)	(23)
§4、	计算转移语句(计算GO TO)	(26)
§5、	标号赋值与赋值转语句	(28)
第四章	逻辑条件语句与块IF结构	(33)
§1、	逻辑型量的运算规定	(33)
§2、	逻辑条件语句	(35)
§3、	块IF结构	(37)
第五章	循环	(41)
§1、	循环开始语句与继续语句	(41)
§2、	多重循环	(48)
§3、	关于循环的若干注记	(44)
第六章	数组	(46)
§1、	关于数组的概念	(46)
§2、	数组说明语句	(47)
§3、	数组元素在内存中的存放方式	(47)
§4、	数组的输入/输出和隐式DO循环	(49)

第七章 语句函数	(56)
§1. 语句函数语句	(56)
§2. 例题	(57)
第八章 函数子程序	(60)
§1. 引言	(60)
§2. 函数子程序的基本结构	(61)
§3. 函数子程序说明语句	(61)
§4. EXTERNAL与INTRINSIC 语句	(63)
第九章 例行子程序	(67)
§1. 例行子程序	(67)
§2. 变界数组	(69)
第十章 数据通信语句与数据置初值	(72)
§1. 等价语句 (EQUIVALENCE)	(72)
§2. 公用语句 (COMMON)	(73)
§3. 赋初值语句 (DATA)	(75)
§4. 数据块子程序 (BLOCK DATA)	(76)
第十一章 建立、编辑、编译与执行F程序	(78)
§1. 建立F源程序文件	(78)
§2. 编译、连接与执行F程序	(84)
§3. 几点补充与说明	(87)
附录A. 内部函数表	(91)
附录B. ASCII字符代码	(94)
附录C. 错误信息表	(96)
(98)	
(99)	
(100)	
(101)	
(102)	
(103)	
(104)	
(105)	
(106)	
(107)	
(108)	
(109)	
(110)	
(111)	
(112)	
(113)	
(114)	
(115)	
(116)	
(117)	
(118)	
(119)	
(120)	
(121)	
(122)	
(123)	
(124)	
(125)	
(126)	
(127)	
(128)	
(129)	
(130)	
(131)	
(132)	
(133)	
(134)	
(135)	
(136)	
(137)	
(138)	
(139)	
(140)	
(141)	
(142)	
(143)	
(144)	
(145)	
(146)	
(147)	
(148)	
(149)	
(150)	
(151)	
(152)	
(153)	
(154)	
(155)	
(156)	
(157)	
(158)	
(159)	
(160)	
(161)	
(162)	
(163)	
(164)	
(165)	
(166)	
(167)	
(168)	
(169)	
(170)	
(171)	
(172)	
(173)	
(174)	
(175)	
(176)	
(177)	
(178)	
(179)	
(180)	
(181)	
(182)	
(183)	
(184)	
(185)	
(186)	
(187)	
(188)	
(189)	
(190)	
(191)	
(192)	
(193)	
(194)	
(195)	
(196)	
(197)	
(198)	
(199)	
(200)	

第一章 FORTRAN基本知识

FORTRAN 语言是流行最广的一种高级程序设计语言。它被广泛地应用于科学和工程的计算中。目前各种计算机上常用的是FORTRAN (Formula Translation) ——77版本的某种子集。为叙述方便,今后在我们的叙述中把FORTRAN语言简称为F语言;把用F语言制作的程序称为F程序。

§1 F程序的特点及书写规定

一、F程序的结构

F程序的结构可以用16个字来概括:“程序分块,块内分行,行中分区,区内按格”。一个F程序通常是由一个主程序或一个主程序及若干个子程序组成。

1、F程序块

一个程序块系由唯一的一个END行结束的若干程序行组成。F程序块有以下几种类型:

- (1) 主程序块:以规定的语句开始,可以单独编译,单独运行。
- (2) 例行子程序块:以规定的语句开始,可以单独编译,但不能独立运行。
- (3) 函数子程序块:以规定的语句开始,可单独编译,但不能独立运行。
- (4) 数据块子程序:以规定的语句开始,可单独编译,但不能独立运行。

由上可知,每一个程序块是一个独立的编译单位。

2、F程序区

如图(1-1),F程序每行由80格组成,每格只能书写一个字符。每行从左到右可划分成四个区。

- (1) 标号区:1—5格为标号区,标号可用无符号整数表示,前零与空格均为无效。
- (2) 续行区:第六格为续行标志位,简称续行区。续行标志只能用非零非空格字符表示。通常用数字或·号表示。
- (3) 语句区:7—72格为语句区,书写语句的内容。如果一行写不完可用下一行的语句区作续行,但续行至多不得超过19行。即一个语句至多不能超过 $66 \times 20 = 1320$ 个字符。
- (4) 说明区:73—80格为说明区,该区可供程序员自行说明一些事情。

3、F程序行

从书写的格式要求上来分,F程序行有四种类型:

- (1) 注释行:用于对程序或程序段作注解。注释行的标志是在每行的第一格写上字母C,后跟注释内容。注释行可以出现在程序块的任何地方。用于对程序或程序段的内容进行注解。

- (2) 正文行:由一些实质性的F语句组成。这些语句通常又分成可执行与非执行语句两类。正文应写在7—72格上。

- (3) 继续行:如正文一行写不下,则可以占下一行,第6格上的非零,非空字符应是

56	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80
C TO OBTAIN THE ROOT OF ITERATION METHOD															
C MAIN PROGRAM															
EXTERNAL FUN															
READ(*,*)X															
R=RCOT(FUN,X)															
WRITE(*,*)R															
STOP															
END															
C THE FUNCTION SUBROUTINE															
FUNCTION ROOT(F,X0)															
EPS=1E-5															
ROOT=F(X0)															
IF(ABS(ROOT-X0).LE.															
*EPS)RETURN															
X0=ROOT															
GOTO 10															
END															
C EXTERNAL FUNCTION SUBROUTINE															
FUNCTION FUN(X)															
FUN=.5*(ALOG10(X)+7)															
RETURN															
END															
56	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80

继续行的标志。但注释行不可有继续行。

(4) 结束行：标志一个程序块的结束。只要三个字母END顺序写在结束行的7—72格的任何位置上均标志程序块结束。

二、注 意

- 1、每一个自然行必须以回车符结束。
- 2、END是块结束标志，也可视为一个语句。
- 3、F程序的特点决定了书写程序的严格性，必须按照块、行、区、格的要求书写。

§ 2 F语言字符集，输入及程序实现

一、字符集：采用ASCII字符集

- 1、数码：十进制0—9
- 2、英文字母：A—Z
- 3、运算符：+，-，*，/，
- 4、标点符：.，，，(，)
- 5、货币号：\$ (¥)
- 6、空格符：□
- 7、专用字符：其它ASCII字符中的可印刷的字符，如：#，，，!，[，]，<，>，，，?，"，=。

二、程序的输入及实现

1、字符代码：

每一个字符均有一个相应的八进制数的代码，称之为ASCII码。即美国信息交换标准码，编写词ASCII的原词组是：

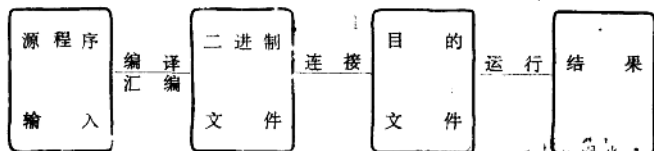
American Standard Code for Information Interchange

2、程序与数据的输入

(1) 程序应严格按照程序清单的顺序逐字符地输入计算机。在各种微型计算机上，程序一般是通过键盘输入计算机的。

(2) 数据输入：数据一般用固定格式与自由格式两种方式输入。数据的形式及输入方式在以后的章节中将详细介绍。

3、程序的实现过程



从图中可以看出，上机操作分为：输入、编译、连接和运行四个步骤。具体作法将在本书的第十一章给予详尽的介绍并提供示例。

§3 常数和变量

常数和变量是F程序的基本运算对象。常数也称为常量，它是在程序运行过程中不变的量。相反，在程序运行中可以随时改变数值的量称为变量。

F程序中使用的常数与变量共有七种类型，本节的目的介绍各种常用常数的书写方法及各种常用变量的类型说明方式。

一、名字

在程序设计语言中，一些运算对象，如变量，函数，子程序等均要有一个名字，以便识别与称呼。

F语言中的名字可用以字母打头的字母数字串组成。串长可达1320个字符。但是作为名字，F编译系统只识别前6个字符，F语言名字中允许插入空格，但认为是无意义的。此外，F语言中名字也不受保留字的影响，但为了避免错误最好不用象系统函数名，和语句专用字这样的名字作为用户定义的名字。

例如：A, B1, C2, AA, B5G, XVS均可作为名字使用。

DO, SIN, IF, GO TO等均不宜作为名字使用。

1B4, 13.4X, B-12, \$100等均不能作为名字使用。

二、常数和变量

1、整常数和整变量

(1) 一个整常数在存贮介质上可以占两个字节或四个字节，请参照说明书，按需要选用。这两种情况分别可以表达的整数范围是：

- 32767—32767

- 2147483647—2147483647

整数可以带性质符号，如：

1, 100, -1357, +1001, 999999

注意：特别强调，一个整数的结尾不允许有小数点出现。

(2) 整变量及其说明方法

取值为整常数的变量叫整变量。

F语言允许用两种方式之一把一个名字说明成整型变量：

①、隐式说明：亦称I—N规则。凡是以I, J, K, L, M, N六个字母开头的变量名，均为整型变量名。如：

I, J, K, LL, MP, NIQR, NAME98等。

②、显式说明：即用特定的语句来说明。

语句格式：INTEGER 若干变量名

例如：INTEGER AMAX, BMIN, IJL

注意：显式说明优于隐式说明。

2、实常数与实变量

(1) 实常数：F语言中，实常数有两种书写形式，即带小数点的形式与指数形式。

①带小数点形式。如：

0.0, 1.0, -1., 8000., -721.0, +3.14159, -0.00001, 456.789等。

②指数形式：即10的整数次幂的形式。F语言规定，必须用字母E表示底数10来书写一个指数形式的数。如：

1.000E-2, 1.E-2, 1E-2, 100.E-4, 0.0001E+2

均表示百分之一。

又如：3.14159E+0, +314.159E-2均表示 π 精确到五位小数的近似值。

注意：

a. 书写程序时，尾数中的小数点可放在任何位置。但在计算机内部与输出结果时，均为“规格化”的形式。即：

$$\pm 0.\underbrace{\text{XXXXXXXX}}_{\uparrow \text{尾数}} \text{E} \pm \underbrace{\text{XX}}_{\uparrow \text{阶码}}$$

也就是说，小数点定在第一个有效数字之前。

b. 实数应带小数点，书写时应与整数严格区别。如2除以3的运算在F语言中能产生以下两种结果：

$$2/3 = 0, \text{ 而 } 2./3. = 0.666\dots$$

c. 实数范围：设X为实数，则某些系统给出：

$$3.0E-39 \leq |X| \leq 1.7E+38$$

(2) 实变量：取值为实常数的变量称为实变量。F语言亦允许用两种方式把一个名字说明为实变量。

①、隐式说明，亦称I—N规则，即：凡不是以I—N字母打头的名字，系统自行识别为实变量。如：

CL, BBF, XIE, YABC, ALPHA, BEDA等

②、显式说明：即用语句形式说明

语句格式：REAL 若干变量名

例如：REAL KMAX, KMIN, B12, IABC

注意：

a. 显式说明优于隐式说明，即一经显式说明之后就不再遵从I—N规则了。

b. 一个单精度常数在内存中占4个字节，精度可达十进制6位。

3、双精度实数与双精度变量

为了提高科学计算的精确度，F语言提供了实数与变量的双倍精度类型。精度可达十进制15位。

(1) 双精度常数

双精度常数在内存中占8个字节，一个双精度常数必须写成指数形式，但需要把单精度指数形式中的字母E换成字母D。如：

$$0.2D+3, -987654321D+0, 5.D-2, 0.3141592653589798D+1$$

如果把最后一个数写成：3.141592653589798则输入后编译程序会自动进行截断，只能得单

精度数: 3.14159.

(2) 双精度变量

双精度变量必须显式说明。

语句格式: DOUBLE PRECISION 若干变量名

例如: DOUBLE PRECISION TEST, DPO. 在PC机上可以用下列语句代替:

REAL*8 TEST, DPO

注意: ①、双精度数的范围是: $4.91D-307 \leq |X| \leq 1.67D+308$

②、对双精度数进行运算时, 其速度会成倍地降低, 这是应当注意的。

4、逻辑常数与逻辑变量

(1) 逻辑常数: 逻辑常数只有两个:

.TRUE. (真) .FALSE. (假)

(2) 逻辑型变量: 逻辑型变量必须显式说明。

语句格式: LOGICAL 若干变量名

例如: LOGICAL A, L1, LABC

注意: 逻辑常数可以占2个或4个字节, 若需要占两个字节, 则可写成 LOGICAL*2 A, L1, LABC

5、文字常数和文字变量

(1) 文字常数: 即字符串常数。

F-77规定, 字符串常数应放入单引号中。

如: 'ABCDE', 'BASIC', 'QRST123'

每一个文字常数的最大长度不得超过127个字符, 各种文本有所不同。通常每个字符占一个字节。

(2) 文字变量: 文字变量必须显式说明。

语句格式: CHARACTER 若干变量名 (数组名)

例如: CHARACTER*5 RST*10, AB, MP 这个语句说明 RST 最多可以有10个字符, AB, MP 各5个字符。

6、复常数和复变量 (略去)

7、双精度复常数与双精度复变量 (F-77无此类型)

注意: 在许多微机, 如8080, Z-80等的F语言中无复型数。

§ 4 标准函数

所谓标准函数系由F语言函数库提供的函数, 这些函数可以作为初等量写入表达式中。不同的F语言版本提供的标准函数多少不一, 使用时可参照具体语言版本的说明书。

数值计算中常用的标准函数如下

名称	意义	自变量个数	自变量类型	函数类型
ABS(a)	a	1	R	R
AMAX0{a _i }	求最大	≥2	I	R

AMAX1{a,}	求最大	≥ 2	R	R
AMIN0{a,}	求最小	≥ 2	I	R
AMIN1{a,}	求最小	≥ 2	R	R
ALOG10(a)	常用对数	1	R	R
ALOG(a)	自然对数	1	R	R
EXP(a)	指数(e^a)	1	R	R
SQRT(a)	平方根	1	R	R
SIN(a)	正弦	1	R	R
COS(a)	余弦	1	R	R
FLOAT(a)	整化实	1	I	R
IFIX(a)	取整	1	R	I

关于最后两个函数示例如下:

设 $I=40$, 则 $\text{FLOAT}(I) \approx 40.0$

1 设 $A=9.8$, 则 $\text{IFIX}(A)=9$

仅就整、实两种数型的相互转换与截断来说, F-77就提供了十多种内部函数(标准函数), 可参考附录二, 内部函数表。

注意:

- ①、使用函数时应注意自变量与函数的类型要求, 按说明书的规定提供。
- ②、整型类的函数系统均以 I—N 字母开头。

§ 5 算术运算符与算术表达式

一、算术运算符

+	加或正符号
-	减或负符号
*	乘法运算符号
/	除法运算符号
**	乘方、幂运算符号

二、算术表达式

用算术运算符把运算对象连接起来的式子称为算术表达式。也叫数值表达式。在书写表达式时应注意:

(1) 运算的优先级;

(), **, *, +

/, - (高——→低)

(2) 除**符之外, 其它运算符不可连续书写, 如 $A + (-B)$ 不能写成 $A + - B$ 。

(3) 为了正确描述数学表达式, 应正确使用括号。如数学式子 $\frac{ab}{cd}$ 应写为:

$(A*B)/(C*D)$ 不应写为 $A*B/C*D$ 。

(4) 正确使用数据类型。

例如：数学式子： $(a+b)^2$ 应写为 $(A+B)**2$ ，不应写为： $(A+B)**(3/2)$ ，这是因为

$$(A+B)**(3/2) = (A+B)**1 = A+B$$

(5) 表达式中，不同类型量的混合运算问题：

①、四则运算 (+, -, *, /)

$$\begin{pmatrix} I \\ R \end{pmatrix} +, -, *, / \begin{pmatrix} I \\ R \end{pmatrix} \Rightarrow \begin{pmatrix} I \\ R \end{pmatrix}$$

同类型运算对象运算的结果亦为同一类型。

②、F-77 允许不同类型的量作混合运算，运算中低型量自行转化为高型的量，结果取高型量。如： $0.5*2/4 = 1.0/4 = 0.25$ ，但不能写成 $0.5*(2/4)$ ，因为 $0.5*(2/4) = 0.5*0 = 0$ 。

注意：混合运算的速度较低，所以应尽量避免书写混合运算的式子。

③、关于乘方运算的规定

指 底	I	R
I	I	R*
R	R	R

*：某些 F 语言版本不允许这种选择。

(6) 其它注意事项请参考具体版本的使用说明书。

第二章 F 程序初步

从事F语言的程序设计必须从逐个熟悉F语言的语句入手。如前所述，F语言的语句可分成两大类：

- 可执行语句：运行这些语句时，计算机可产生一系列的动作，完成语句规定的功能。如：赋值语句，输入/输出语句，转移语句等。
- 非执行语句：这类语句只是向编译系统提供程序中的有关信息，运行时并不产生动作。如：类型说明语句，格式语句，维数说明语句等。

在今后几章中，我们将陆续介绍这两类语句，本章则着重介绍：计算赋值语句，输入/输出语句以及格式语句的一些初步知识。在完成本章的学习之后，读者将可以制作一些最简单的F语言程序。输入/输出语句及格式语句是学习F语言的难点之一。请读者一开始就重视它们的使用。

§1. 计算赋值语句

一、计算赋值语句的格式与功能

1、格式：变量 = 表达式

$$V = e$$

2、功能：把右边表达式的值赋给左边的变量。

二、赋值过程中，数据类型的转化规则

我们再次强调，在F语言中数据与变量均特别强调类型。这里的转化规则是：

- 若V, e类型一致，则照赋。
- 若V, e类型不一致，则系统自行把e转化为V的类型后再赋给V。列表如下：

V \ e	I	R
	← 照 赋	← 化 赋
I	← 照 赋	← 化 赋
R	← 化 赋	← 照 赋

例如：I = 8 I取值为8

J = 3.5 J取值为3

A = 3 + 4 取值为7.0

P2 = A * X ** 2. + B * X + C 照赋

总之，类型一律向左看齐。

例：已知三角形的三边分别为：A = 2, B = 3, C = 4, 求三角形的面积。

解：可大致写成如下程序：

A = 2.0

B = 3.0

C = 4.0

F = (A + B + C) / 2.0

AREA = SQRT(F * (F - A) * (F - B) * (F - C))

END

这是一个逻辑上完整的程序，由于无输出语句，所以计算的结果我们尚不能见到。

§ 2. 参数说明语句

F语言中，通常把常数以规定的形式直接写入程序中。其类型也就由书写方式决定。但是，在F-77以及某些非标准的F语言版本中，允许为常数取一个名字，以便于程序的制作，这个名字连同其数值就是通过参数说明语句来给定的。

一、语句的格式与功能

1、格式：

PARAMETER 参数名 = 常数表达式，...

PARAMETER P = e, ...

2、功能：

以左边名字代表右边表达式的值。程序中可以用这个名字代表相应的常数。

例如：

PARAMETER PI = 3.1416, E = 2.71828

二、注意

- 1、左边的名字不是变量，程序中不能对其再行赋值，换句话说，它是一成不变的。
- 2、它不是执行语句。不同于计算赋值语句。
- 3、F-77对参数语句有所扩充，请参考所用版本的说明书。

例：用参数语句改写求三角形面积的程序。

解：程序如下：

REAL L

PARAMETER A = 2., B = 3., C = 4.

L = (A + B + C) / 2.

AREA = SQRT(L * (L - A) * (L - B) * (L - C))

END

这个例子说明，使用参数说明语句，可以省去一些赋值语句。但是，要改变边长，则必须改写参数赋值语句。

§ 3. 输入/输出的初步概念

我们把计算机（主机）与外界交换信息的过程叫做输入/输出。显然，这是人们关心的

主要问题之一。因为不具备输入/输出能力的计算机是没有用的，事实上也不存在。计算机与外界交换信息是通过输入/输出语句并借助于各种输入/输出设备实现的。既然是交换信息，那么在输入/输出语句中就应当向计算机提供以下的信息：

①数据传递的方向，即指明是输入还是输出。

②数据的传递对象（到达部位的名称）

③数据的传递格式。

④数据的传递内容。

应当注意的是，输入/输出的参照系是计算机的主机。换句话说，从外设进入主机的信息叫输入；从主机送往外设的信息叫输出。输入/输出简称I/O。

F语言可以通过输入/输出语句提供广泛的输入/输出格式，以适应用户可能提出的各种需要。正因为种类繁多，也就给学习这种语句造成了一定的困难。本节只着重介绍最常用的输入/输出形式。有了这些，对于一般的科学计算来说就足够了。更多的内容待需要时可查阅有关的使用说明书或参考其他有关F语言的教科书。

一、F语言常用的输入/输出种类

1、自由格式输入，固定格式输出

数据按系统规定的格式进行转换，再输入/输出。

2、带格式的输入/输出

数据按用户指定的格式及类型进行转换，然后再输入/输出。

3、二进制形式的输入/输出

由于数据在主存中与大容量存储器中均是以二进制形式存放的，所以可直接以二进制形式传送，这样可以提高传送效率。主要用于主机与磁盘或磁带存储器之间的信息传送。

4、辅助输入/输出

配合完成文件传送的辅助功能

5、会话型输入/输出

这是一种非标准的规定。主要用于终端与主机之间的简短通信。某些F语言版本有这种功能。

在以上五种形式中，最主要的是前两种形式，本节分别给予介绍。

二、带格式输入/输出语句的格式与功能

1、语句格式

READ (设备号, 格式说明) 变量表

WRITE (设备号, 格式说明) 变量表

标号 FORMAT (一串格式说明符)

如果用符号表达以上格式，则可以写成下列形式：

READ (u, f) V1, V2, ...

WRITE (u, f) V1, V2, ...

f FORMAT (FORMAT SPECIFIERS STRING)

其中，u_i是为F程序规定的各种输入/输出设备的代码。也叫通道号。它们通常取零、正

用星号*号。F-77规定*号为系统设备号(常指键盘和显示器)

前缀\$为格式语句的语句标号,通常为正整数,也可以是用ASSIGN语句赋值的变量名。

注意:

①不同的F语言编译系统为输入/输出设备规定了自己的设备代码。特别在微型计算机上随机器的不同有一定的差别。在使用计算机时应查阅说明书,或请教有关工作人员。

②F语言也提供了一种语句,允许用户通过它为输入/输出设备规定一个代码。我们将在第九章中给予介绍。

③F-77以及各种MS-FORTRAN常以*号代表系统的输入/输出设备号。它代表键盘或显示器。

在输入表中,可以有简单变量,数组变量和数组名字以及随后将介绍的隐式DO循环。在输出表中,除了以上内容之外还允许有表达式。F-77规定变量表中可以有文字变量。

2. 语句功能

按FORMAT语句中格式符的规定,把外设上的数据输入,顺序赋给表中的各变量,或按格式规定把表中各项的值输出在外部设备介质上。

例题:

```
READ (11, 500) A, B, C, I
500  FORMAT (F4.1, F5.2, F6.3, I4)
WRITE (10, 501) A, B, C, I
501  FORMAT (1X, 3F6.3, I4)
END
```

注意:

①、数据类型必须与格式符类型一致。

②、场宽:数据在外设上所允许占的最大宽度。

③、例题中11, 10分别为输入/输出设备代码。若把设备代码11, 10均改为*号,则数据应从键盘输入,从显示器输出。

④、F-77允许把格式字符串直接写入输入/输出语句中,如:

```
READ (*, '(15, F10.3)') JMP, ABC
READ '(15, F8.3, F9.4)', M, R, P
WRITE (*, '(1X, 15, F10.3)') JMP, ABC
PRINT '(1X, 15, F8.3, F9.4)', M, R, P
```

说明:

a、输入/输出语句中的格式字符串应加单引号。

b、可以不给出设备代码,自动在系统指定的设备上输入/输出,如第二句与第四句,但串后必须有逗号。

c、不指出设备码时,必须用PRINT语句输出,然而有的F-77子集无PRINT语句。

d、输出时,第一个格式字通常应写上一个nx供纵向走纸用。这点以后再详细说明。

三、自由格式输入、固定格式输出语句

1、格式

READ (u) 变量表

WRITE (u) 变量表

其中：u为设备代号

2、功能：当输入时，外部数据不受格式限制，比较自由。只要数据之间以逗号（或回车）分开即可，边读入，边赋给表中的变量。当输出时，系统自行对整型量和逻辑型量提供8个字符的场宽，为实型量提供16个字符的场宽，为双精度量提供32个字符的场宽。

例1：试把数据1.1, 2.2, 3.3, 4.4, 5和6输入计算机。

解：可由以下语句实现。

READ (11) A, B, C, D, I, J

当执行此语句时，可键入

1.1, 2.2, 3.3, 4.4, 5, 6

或打入：

1.1, 2.2, 3.3, 4.4

5, 6

完成以上操作之后，1.1, 2.2, 3.3, 4.4分别赋给A, B, C, D。5, 6分别赋给I, J。

例2：把机内变量I=7, R=7.8的值输出。

解：可由以下语句实现。

I=7

R=7.8

WRITE (10) R, I

当执行以上语句时，在同一行上输出：

△△△△0.780000E+△1△△△△△△7

场宽=16

场宽=8

〔注〕

注 意：

①这种方式比较方便，当对输入/输出无特别要求时采用较多。

②各种F版本对自由格式（亦称表控格式）的规定大同小异，共同之处是用户不提供输入/输出格式。

③F-77表控输入/输出给出了以下语句格式：

READ, 变量表

READ (u, *) 变量表

说明：这里的*号表示自由格式输入。

WRITE (u, *) 变量表

PRINT *, 变量表

〔注〕：本书中用“△”号表示空格。