

51.81
1022
FELIXC—256 計算機

F O R T R A N

程序语言教材

北京航空學院 計 算 機 室
二〇四教研室

一九七五年十月

目 录

程序語言的发展.....	1
第 一 章.....	4
§1.1 FORTRAN 的例	4
§1.2 FORTRAN 的基本符号.....	11
§1.3 FORTRAN 元素.....	13
第二章 表达式和赋值語句.....	16
§2.1 数据类型.....	16
§2.2 变量和类型說明.....	18
§2.3 数组和維数語句.....	20
§2.4 表达式.....	23
§2.5 赋值語句.....	30
第三章 FORTRAN 的其它語句.....	33
§3.1 轉移語句.....	33
§3.2 停語句、暫停語句和繼續語句.....	44
§3.3 循环語句.....	45
§3.4 輸入輸出語句.....	54
§3.5 格式語句.....	58
第四章 函数和子程序.....	67
§4.1 内部函数和基本外部函数.....	67
§4.2 語句函数.....	72
§4.3 函数段.....	75
§4.4 子程序段	79
§4.5 外部語句的使用.....	83

第五章 涉及存貯分配的語句 85

§ 5.1 公共語句..... 85

§ 5.2 等价語句..... 87

§ 5.3 数据初值語句..... 89

§ 5.4 数据子程序段..... 90

FORTTRAN 語句分类..... 91

程序結構..... 93

控制卡..... 94

編程序的例..... 95

习 題 99

程序語言的發展

在电子計算机产生的初期，人們直接采用机器指令編写程序。(手編程序) 这些指令都是一些数字代碼，指令功能的含义不直觀，对于程序的編写，檢查和修改都比較复杂。为了适应計算机发展的需要，后来产生了符号語言，符号語言的基本思想就是将指令代碼和內存的地址都用直觀的符号来代替。

例如在 112 机中，我們可以用以下的符号代替操作碼：

功 能	操作碼	記 忆 碼	由 来
取数	021	LD	(LoaD 加载)
送数	022	ST	(STore 存入)
加法	010	ADD	(ADDition 加)
減法	011	SUB	(SUBtraction 減)
乘法	012	MU	(MULtiplication 乘)
除法	013	DIV	(DIVision 除)

其中地址也用习惯的符号代替，常数用直接数代替。

例如：

符号程序可以这样写：

LD	X	$X \Rightarrow A$ 寄存器
MU	X	$X^2 \Rightarrow A$ 寄存器
SUB	0.5	$X^2 - 0.5 \Rightarrow A$ 寄存器
ST	Y	$X^2 - 0.5 \Rightarrow Y$

要使得符号程序在机器上直接执行是不可能的。所以要通过一个符号語言的編譯程序把符号語言变为机器語言(用指令所編出的程序)，才能在計算机上执行。

符号語言的編譯程序主要是一些造表和查表的过程。如在上问题中先造变量和常数的分配单元对照表

1000	X
1001	0.5
1002	Y

然后根据内存分配表和操作碼对照表，一条一条查看符号程序，将記憶碼換成机器操作碼，将变量符号和常数換为内存单元。

LD	X	00	021	1000	⇒ 在112机上执行
MU	X	00	012	1000	
SUB	0.5	00	011	1001	
ST	Y	00	022	1002	

将符号語言变为机器語言的工作完全是由編譯程序在机器上自动进行的。所以符号語言在一定程度上簡化了程序設計的工作。

后来符号語言逐漸扩大了功能，形成了現在的汇编語言，汇编語言在很大程度上簡化了程序設計的工作。

在實踐中，人們又設計了另一种程序語言，这种語言是和人們日常使用的語言习惯（主要指英語）和实际的数学问题更加相近的一种語言。所以学习比較容易，語言比較直觀，檢查和修改程序都比較方便。因而这种語言发展很快，目前种类也很多，主要用于科技計算的FORTRAN語言ALGOL語言和商业企业管理用的COBOL語言，用于机械加工方面APT語言，及大型綜合性語言PL/I等。

FORTRAN 語言簡介

这是一种适合于在工程，科学計算中使用的程序語言。此名由Formula TRANslater（公式翻譯）头几个字母併成，目前国际上有标准的FORTRAN的文本（标准規定），但由于每台計算机的能力不同，所以实现标准的FORTRAN語言的規定时，也略有差異。我們現在介紹的是FELIX C-256計算机上的FORTRAN語言，它是以国际标准FORTRAN IV为基础的。

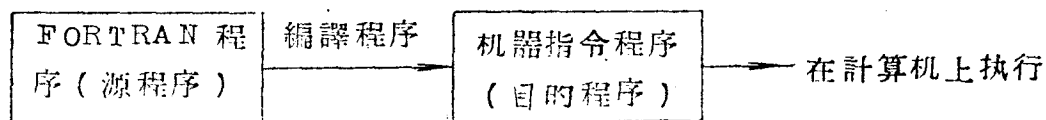
在 112 机程序設計中，我們曾編过下式的程序

$$F = \frac{(0.5 + 0.25 - 0.3) 0.9}{0.8}$$

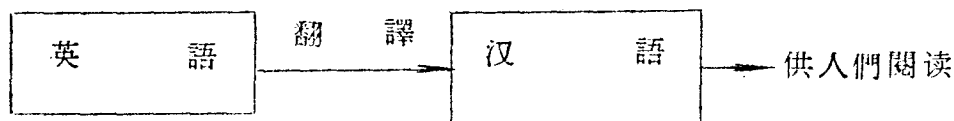
在 FORTRAN 語言中就直接写成下列的式子，送到机器中

$$F = (0.5 + 0.25 - 0.3) * 0.9 / 0.8$$

但是机器並不能执行上式，所以还必须通过一个編譯程序（翻譯程序）把 FORTRAN 的程序，变成机器指令写出的程序，然后在机器上执行。（这个过程和符号語言的翻譯过程类似）。



就象不懂外語的人要读一篇文章必須經過下列步骤一样：



英語由 26 个字母和一些專門的符号有，？！……按一定的詞法和語法規則构成一篇文章才能使翻譯正确的翻譯出来。

同样算法語言的翻譯程序也是按照規定的符号和語法規則将源程序翻譯成目的程序。所以人們在书写程序时就必須按照它所提供的符号和語法規則书写，不能有一点的馬糊。否則編譯程序就不能很好的进行編譯甚至无法进行編譯。这一点是非常重要的。

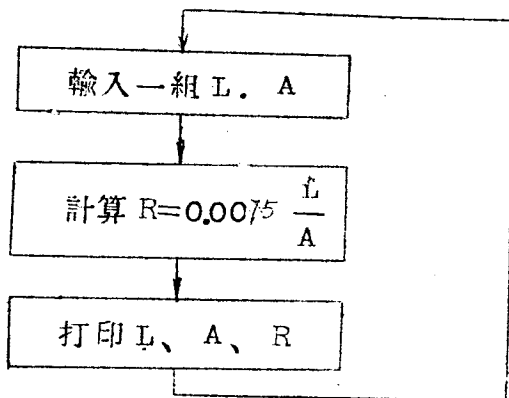
第一章

§1.1 FORTRAN 程序的例

例1. 计算电阻 $R = 0.0075 \frac{L}{A}$ L 导体长度 A 横截面积

$$\text{其中} \begin{cases} L_1 = 300.5 \\ A_1 = 2 \end{cases} \quad \begin{cases} L_2 = 492.3 \\ A_2 = 125 \end{cases}$$

計算框图



1	2	3	4	5	6	7	8	9	10	11	12	13
C	E	X	A	M	P	L	E	1				
C	C	O	M	P	U	T	E	R	E	S	I	S
					R	E	A	L	L	,	A	,
					R	E	A	D	(	1	0	5
					,	2	0	)	L	,	A	
					F	O	R	M	A	T	(	2
					F	7	.	4	)			
					R	=	0	0	0	7	5	*
					L	/	A					
					W	R	I	T	E	(	1	0
					,	3	)	L	,	A	,	R
					F	O	R	M	A	T	(	1
					X	,	F	8	.	4	,	1
					0	X	,	F	8	.	4	,
					1	0	X	,	F	1	0	.
					4	)						
					G	Ø	T	Ø	1			
					E	N	D					
					0		2					
					0		2	5				
					.	E	O	F				

1~10 行是源程序部分 11~13 为数据部份

在源程序部份

1~2 两行是注释部份，这一部份对程序来讲是可有可无的，编译程序不对注释进行翻译，它往往是为了阅读程序时方便而加进的一些注解，在 1·2 行中第一列都是“C”这是注解行的标志，第一行注释内容是“例 1”第 2 行注释内容“计算电阻”。

3~9 行是 FORTRAN 程序的语句（也称指令，命令）行，每一语句实现一种功能。

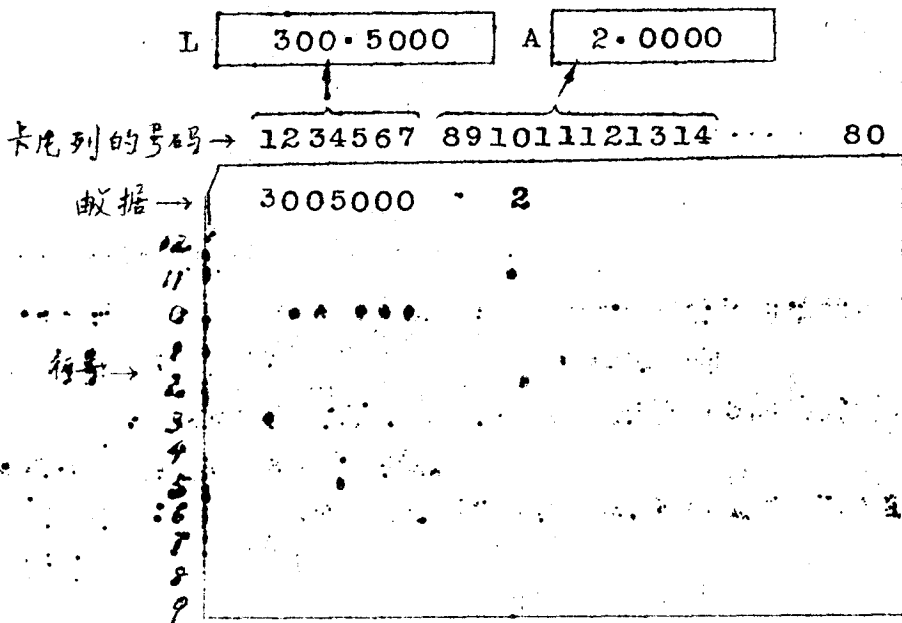
第 3 行，REAL L, A, R 类型语句，它告诉编辑程序 L, A, R 三个变量是实型变量。

第 4 行 1 READ (105, 20) L, A 是输入语句。

它指出从代号为 105 的读卡机上，按照标号为 20 的格式语句所提供的格式从卡片上读出二个数送到 L 和 A 所佔的内存单元中去（这是由编译程序在进行编译时，对变量和数据所分配的内存单元）。

第 5 行 20 FORMAT (2F7.4) 格式语句

按照这个格式语句并连系上句，就是将卡片上的 7 列数为一组读入两组，其中每 7 个数的后 4 位是小数部份执行结果是这样
L=300.5000、A= 2.0000



上面是源程序，近似的列可以读为 01

第 6 行 $R=0.0075 \cdot L/A$ 是赋值语句

即求出 $0.0075 \cdot 300.5/2$ 的值送到变量 R 的单元中

第 7 行 `WRITE(108, 3) L, A, R` 是输出语句

在代号为 108 的宽行打印机上按照语句标号

为 3 的格式语句提供的格式打印出, L, A, R 的值。

第 8 行 `3 FORMAT (1X, F8.4, 10X, F8.4, 10X, F10.4)`

格式语句。

宽行打印机的印刷纸一行能印 132 个字符。打印结果是这样的

300.5000		bb2.0000		bbbb11269
↓	10 个空格	↓	10 个空格	↓
L(格式 F 8.4)	对应 10X	A 值 (格式 F8.4)	(10X)	R 值 (格式 F10.4)

格式语句括号中的第一个 1X 是命令印刷纸前进一行, 即将打印在一新行上。b (blank) 代表空格, 在印刷纸上是空白, 在卡片上对应列无孔。

第 9 行 `GO TO 1` 无条件转移语句。

一般情况下 FORTRAN 的语句是一个接一个按顺序执行, 当遇到控制转移语句时就可以改变执行顺序。GO TO 1 是命令无条件转向标号 1 的语句。这样又执行读语句, 从第二张卡片上又读入一组数 L_2, A_2 , 计算 R_2 打印结果又转向标号 1 的语句直到碰到数据卡片上为 . EOF (完结) 标志已没有数据提供计算, 程序也就停止。

第 10 行 `END` 是 FORTRAN 程序段的结束, 它标志本程序段已结束, 没有要编译的内容了。

程序段是一个独立的程序单位, 它完成一定的功能, 一般能独立进行编写和调试。正象我们在手编程序设计中的主程序和子程序都是程序段。

例 2. 计算

$$\begin{cases} \sigma_1 = A^2 - 2B + 1 \\ \sigma_2 = A^2 + 2B + 1 \end{cases} \quad \begin{cases} A = 9 \\ B = 8 \end{cases}$$

FORTRAN 程序

1	2	3	4	5	6	7	8	9...
1	C	E	X	A	M	P	L	E
2						I	N	T
3						R	E	A
4				1		F	O	R
5						M	A	T
6						C	I	G
7						A	=	A
8	9	9	9	9	9	**	2	+
9						2	*	B
10						+	1	
11						C	I	G
12	9	8				A	=	A
						**	2	+
						2	*	B
						+	1	
						W	R	I
						T	E	
						(	1	0
						8	,	9
						9	9	9
						9	9	9
						)	A	,
						A	,	B
						,	C	I
						C	I	G
						A	1	,
						C	I	G
						A	2	
						F	O	R
						M	A	T
						(	2	I
						1	)	
						S	T	O
						P		
						E	N	D

1~11 源程序部份

12 为数据部份

第一行 注译 例 2

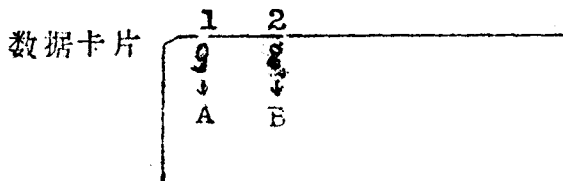
第2行 INTEGER A, B, CIGAMA1, CIGAMA2 类型语句

说明 A, B CIGAMA1, CIGAMA2, 四个变量为整型量

第3行 READ (105, 1) A, B 输入语句

第4行 1 FORMAT (2I1) 格式语句

READ 语句执行的结果是将2个1位的整数分别送到A和B单元中去, (每一位对应卡片上的一列)



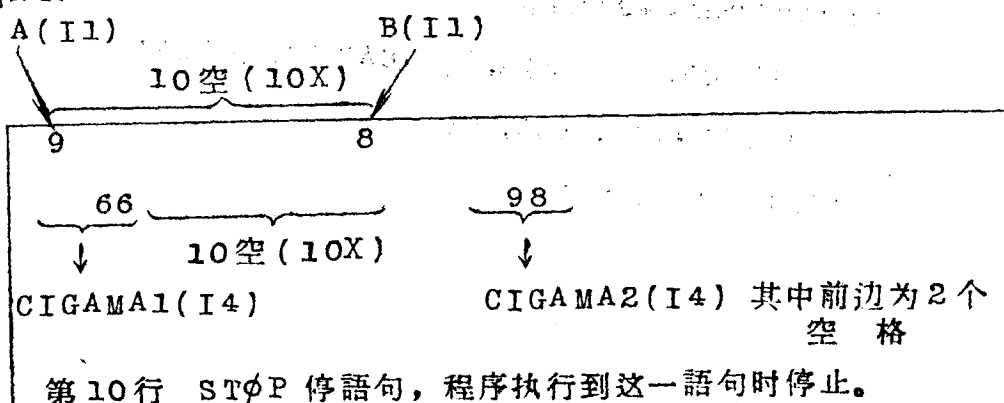
計

(16)

第5, 6行是赋值语句, 求出 CIGAMA1 和 CIGAMA2 的值
 第7行 WRITE(105, 99999)A, B, CIGAMA1, CIGAMA2,
 按照标号 99999 的语句提供的格式在宽行打印机上印出四个变
 量的值: A, B, CIGAMA1, CIGAMA2。

第8行 99999 FORMAT (1X, I1, 10X, I1

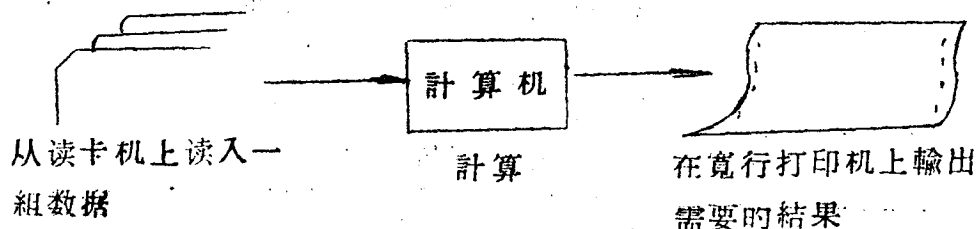
第9行是第8行的继续行, 继续行的标志是在本行第6列上不是空
 格不是 0 是其他 FORTRAN 的符号, 此处是 1, /1X 是指示下边的
 输出值在下一行上印刷



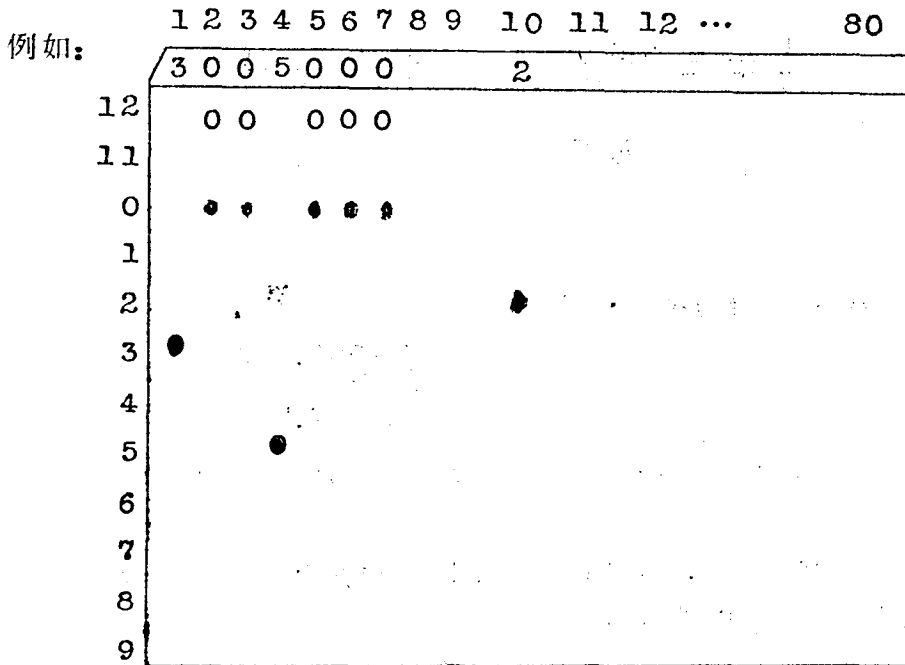
输入输出与格式语句

为了使初学 FORTRAN 语言各语句时就能编出一个较完整的程序,
 我们将输入输出和格式语句先作一感性介绍。

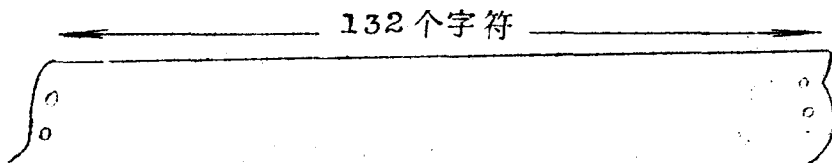
例 1, 例 2 的计算过程是这样:



一张卡片有 12 行 80 列, 每一列上有 12 个位置在不同的位置上穿
 孔, 表示不同的符号或数据 (见附录表) (这一列无孔时代表空白) 一
 张卡片最多能容纳 80 个字符。 (附录表在本册 12 页, 基本符号编
 码表)



本机寬行打印机的印刷紙規格:



我們称一張卡片, 或印刷紙的一行为一个記錄

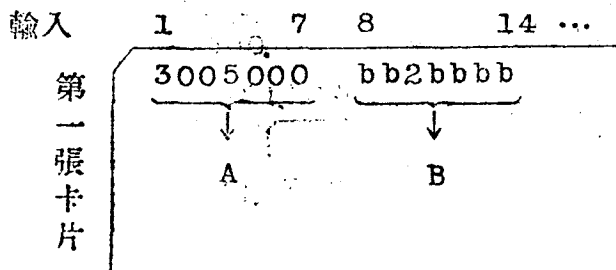
105 为卡片读入机 108 为寬行打印机的設備号。

例:

```

READ (105, 1) A, B, I, C
1  FORMAT(2F7.4/1I1, 10X, F52)
   WRITE (108, 2) A, B, I, C
2  FORMAT(1X, F8.4, 10X, F8.4/1X, 1I1, 17X, F8.4)

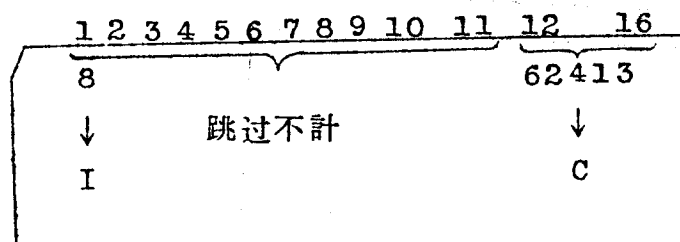
```



計

①6

第二張卡片



READ 語句執行結果:

A=300.5000

B=002.0000

I=8

C=624.13

對照格式語句，變量和卡片上的數據，察看 READ 語句執行的結果：

1. 我們稱 2F7.4, 1I1, 10X, F5.2 為區域描述符更一般的可寫為 rFW.d, rIW, nX。

FW.d 為實形描述符

rIW 為整型描述符

FW.d 是從卡片上順次取 W 個字符 (W 中不含小數點的話，後 d 位作為小數部分送到相對應的變量單元中，r 是這個形式的取法作 r 次，送到相應 r 個變量中。

IW 是在外部卡片上取連續的 W 個字符，作為整數送到相應變量中去。

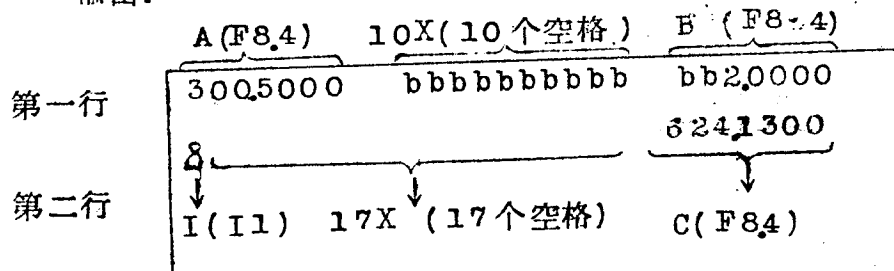
nX 是跳過卡片上的 n 個字符不輸入

2. 每一個描述符都有，或者／把他們分開

3. 遇到／之後換了一張卡片，(即從下一記錄上讀數據)

4. 格式語句中的區域描述符卡片上的數據和輸入變量之間是從左向右一一對應，(nX 是跳過卡片 n 個字符不與輸入變量對應)。

輸出：



对照輸出变量格式語句和打印結果，可以看出：

1. $FW.d$ 輸出一个实型数，共佔 W 个字符位置，后 d 位是小数
 IW 輸出一个整型数佔 W 个字符的位置

两种情况前边位数不足时补空

2. nX 是在打印紙上空 n 个符号位，这样便于观察結果数据。

3. 描述符之間用，（逗号）隔开

4. 遇到 $/IX$ 时将要輸出的变量值，打印在一新的行上。

5. 輸出变量和区域描述符（重复因子 r 算 r 个）之間是从左向右一一对应的， nX 不对应任何变量。

6. 無論对輸入輸出的格式語句永远从一个新記录（一張卡片一个打印行）开始。

§12 FORTRAN 的基本符号

26 个大写拉丁字母 A B C D E F G H I J K L M N O P Q R S T U
V W X Y Z （O 写为 ϕ 为区别于数字 0）。

10 个阿拉伯数字 0 1 2 3 4 5 6 7 8 9

13 个专用符号 = + - * / ()

， · & \ (引号) \$ (百分符)

（空白我們用 b 代表）

如果在 FORTRAN 的程序中，出現了基本符号以外的符号。

編譯程序就不能識別，在控制台上給出錯誤信息。

在上列中，我們將 σ 写为 CIGAMA 就是这个道理。

基 本 符 号 编 码 表

符 号	卡片穿孔位置	16 进 码	符 号	卡片穿孔位置	16 进 码
0	0	F0	0	11-6	D 6
1	1	F1	P	11-7	D 7
2	2	F2	Q	11-8	D 8
3	3	F3	R	11-9	D 9
4	4	F4	S	0—2	E 2
5	5	F5	T	0—3	E 3
6	6	F6	U	0—4	E 4
7	7	F7	V	0—5	E 5
8	8	F8	W	0—6	E 6
9	9	F9	X	0—7	E 7
A	12-1	C1	Y	0—8	E 8
B	12-2	C2	Z	0—9	E 9
C	12-3	C3	空白	不 穿 孔	4 0
D	12-4	C4	=	8—6	7 E
E	12-5	C5	+	12-8-6	4 E
F	12-6	C6	-	11	6 0
G	12-7	C7	*	11-8-4	5 C
H	12-8	C8	/	0—1	6 1
I	12-9	C9	(	12-8-5	4 D
J	11-1	D1	)	11-8-5	5 D
K	11-2	D2	,	0-8-3	6 B
L	11-3	D3	.	12-8-3	4 B
M	11-4	D4	\$	11-8-3	5 B
N	11-5	D5			

§1.3 FORTRAN 的元素

FORTRAN 的程序紙共有 80 列

1 2 3 4 5	6	7 ...	72	73 ... 80
-----------	---	-------	----	-----------

在例 1 中，我們將 1~10 行称为源程序行，其中第一行为注譯行第 10 行为結束行。

第 3~9 行为語句行。在程序紙的 1~5 列是語句的标号区，若有标号的話就写在这里。例 1 中的，1，20，3，例 2 中的 1，9 9 9 9 9 就是語句标号，无标号的語句标号区为空白。

語句标号：

語句标号只有当这一語句被引用时才有意义，例如格式語句永远是被輸入輸出語句引用的，所以必須有标号。

語句标号是 1~9 9 9 9 9 中的任一个数

注意：

1. 在同一程序段中，不同的語句使用同一个标号这是錯誤的，（否則不能区别）而且語句标号只能放在这一語句的开始行上，不能放在继续行上，例如：例 2 中同一格式語句佔据了 2 行，标号 99999 放在第 8 行上不能放在第 9 行上。

2. 标号中数字前面的空白和数字中間或尾部的空白不改变数字值

b b 1 4 8

这都是同一个标号

1 4 8 b b

1 b 4 b 8

0 1 4 b 8

b 0 1 4 8

标識符（变量名称

有些量在程序执行过程中是变化的例如：数学中的 $Y = X + 3$ Y X 是变量，3 是常量。为了在程序中引用变量，我們就給变量起个名字或者叫标識符，在例 1，例 2 中的，A，B，L，R，CIGAMA1，CIGAMA2 都是标識符。一般在可能的情况下，我們多

用原来在数学上和实际问题中习惯的名字，这样看起来直观。

标识符的组成：

由 26 个拉丁字母和 10 个数字组成，最多可以有 8 个字母数字。第一个符号必须是字母

A, B, A12, ACB5, D12K, IJ3LMNT3

都是正确的标识符。

下面的符号不能作为标识符

3 A, A · B, CN&

如果字母数字的个数多于 8 个时，多余字符忽略不计

如： Z 1 2 3 4 5 YW 被认为是同一个标识符。

Z 1 2 3 4 5 YWV

语句名称

除了赋值语句 $V = e$ (V 变量, e 是表达式) 之外其他语句都有一个语句名称 (指令名称) 是由大写的英文字母构成。

如输入语句 READ, 输出语句 WRITE,

无条件转移语句 GOTO

类型语句 REAL, INTEGER 等

语句分为可执行语句和不可执行语句。可执行语句 (READ, WRITE, GOTO, 等) 是由编译程序编译, 产生目的指令。不可执行语句不产生目的指令, 只是给编译程序提供信息。如 REAL, INTEGER 等, 只是告诉编译程序它后边的变量的数据类型。

§1.4 FORTRAN 程序的书写方式

关于书写方式我们曾在前几节作过一些零散的介绍, 在此我们作一个小结:

一个 FORTRAN 的程序段写在 80 列的程序纸上一行中字符从左到右横着写每个字符佔一列 73 ~ 80 列作为编号或标识用, 不用来写程序。

注释行

第一个列必须是字母 C 第 2 列不能是 / (斜杠有其他含义) 注释