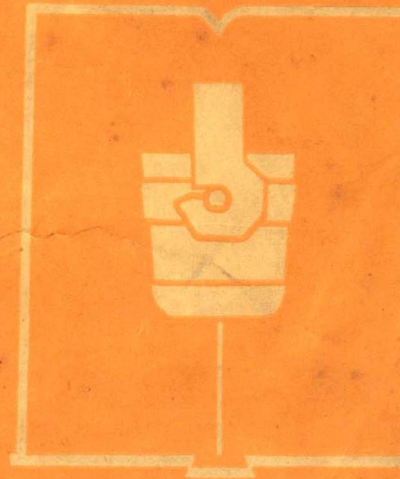


书

FORTRAN语言 程序设计

GAODENG XUEXIAO JIAOXUE YONGSHU



冶金工业出版社

高等学校教学用书

FORTRAN语言 程序设计

北方工业大学 邓德祥 编

冶金工业出版社

前 言

随着新技术革命的不断发展和计算机基础知识已和数学、外语一样,被列为高等理工院校各专业的必修课, FORTRAN语言程序设计则是计算机基础知识中的一个重要组成部分。为了满足教学上的需要, 现把该书奉献给读者。该书是在多年研究与教学实践的基础上编写的, 内容安排上有如下特点:

(1) 以程序设计为中心。首先概括地介绍与本书内容有关的计算机基础知识, 然后着重讲述了三种简单结构(顺序、选择和循环)程序设计、多重结构程序设计以及三种辅程序的设计和文件的使用方法。而FORTRAN语言语法规则是根据各种程序设计的需要, 按照人们认识规律, 以启发方式逐步引入的, 这也正体现了子集扩展法的原则。

(2) 为提高程序的可读性、可靠性, 培养良好的程序设计习惯, 简要地介绍了结构化程序设计方法, 并以它指导程序设计, 使本书所列举的程序有着统一的风格。

(3) 对例题的示范作用给予了足够的重视。为讲述典型的程序结构、语法规则和程序设计的基本方法和技巧, 本书列举了许多形式多样、短小实用、便于讲授的例题。这些例题兼顾了数值计算与非数值计算、科学计算与事务处理, 编程序与读程序、实用性与趣味性等各个方面的需要, 此外, 全部例题均经上机调试通过。

(4) 把DATA语句与变量同步安排, 为读者能够尽早上机、实现理论学习与上机实习的紧密结合创造了条件。实践证明, 一边学习一边实践, 既可提高学习兴趣, 又有利于培养实际工作能力。

(5) 鉴于近期计算机教学与应用主要以微型机为主, 以及为了继承丰富的软件成果, 确定本书的语言文本以FORTRAN66^①为基础。此外, 为了适应科技知识不断更新的需要, 在每章之后讲述了“FORTRAN77的有关扩展与限制”, 同时在附录1中给出了“语句一览表”, 以便于读者学习、比较和查找。

综上所述, 本书在内容上作了精心的安排, 目的是使读者更好地掌握FORTRAN语言的基本语句、程序的基本结构以及程序设计的基本方法和技巧。程序设计能力的培养是本书的出发点和归宿。

本书可作为高等理工院校各专业的教材, 也可供职工大学和有关短训班使用, 以及各类工程技术人员自学或参考。

北京钢铁学院冯克清同志担任了本书的主审工作, 东北工学院翟秉仁、北方工业大学田成方和中南工业大学熊传杰同志认真审阅了初稿, 提出了许多宝贵意见; 华中工学院葛果行, 北京航空学院杨文龙, 北京大学谢柏青也阅读了初稿, 并提出了不少有益的建议, 在此表示诚挚的感谢。

由于编者水平有限, 时间仓促, 错漏和不妥之处在所难免, 敬请读者批评指正。

编 者

1986.8

^①FORTRAN66为FORTRAN X3.9-1966, 相当FORTRAN IV。

目 录

前言

1 计算机、语言和程序设计	1
1.1 计算机简介.....	1
1.2 计算机语言概述.....	3
1.3 FORTRAN程序设计概述.....	5
习题.....	8
2 FORTRAN初步知识	9
2.1 FORTRAN字符集与符号名.....	9
2.2 数据(一) 整型常数与实型常数.....	10
2.3 数据(二) 整型变量与实型变量.....	12
2.4 语句.....	14
2.5 FORTRAN源程序与流程图概述.....	16
2.6 结构化程序设计简介.....	21
2.7 FORTRAN77有关扩展与限制.....	22
习题.....	23
3 顺序程序设计	25
3.1 常用的格式输出.....	25
3.2 常用的格式输入.....	28
3.3 算术计算.....	30
3.4 算术赋值语句.....	33
3.5 标准函数、语句函数.....	35
3.6 顺序程序设计.....	37
3.7 格式输入/输出续介.....	39
3.8 FORTRAN77有关扩展与限制.....	45
习题.....	48
4 逻辑型数据	50
4.1 逻辑常数与变量.....	50
4.2 逻辑变量的输入/输出.....	50
4.3 逻辑运算.....	51
4.4 逻辑赋值语句.....	54
4.5 FORTRAN77有关扩展与限制.....	55
习题.....	56
5 选择程序设计	58
5.1 无条件转移语句(GOTO 语句).....	58
5.2 逻辑IF语句(逻辑条件语句).....	59
5.3 计算GOTO 语句.....	65
5.4 其它转移语句.....	67

5.5	FORTRAN77有关扩展与限制	70
	习题	73
6	循环程序设计	75
6.1	循环概述	75
6.2	DO循环	82
6.3	多重循环	88
6.4	FORTRAN77有关扩展与限制	96
	习题	97
7	数组及其应用	99
7.1	数组概述	99
7.2	数组说明	100
7.3	数组元素	101
7.4	数组的输入/输出	103
7.5	数组应用举例	107
7.6	FORTRAN77有关扩展与限制	117
	习题	119
8	辅程序设计	121
8.1	辅程序概述	121
8.2	函数辅程序	122
8.3	子程序辅程序(子程序)	126
8.4	虚拟变元与虚实结合	132
8.5	公用语句与公用结合	142
8.6	等价语句	146
8.7	数据块辅程序	148
8.8	FORTRAN77有关扩展与限制	153
	习题	157
9	双精度型数据和复型数据	160
9.1	双精度型数据	160
9.2	复型数据	162
9.3	算术型数据小结(包含FORTRAN77内容)	164
	习题	166
10	字符型数据及其应用	168
10.1	霍勒列斯(Hollerith)常数	168
10.2	字符的存取与输入/输出	168
10.3	格式数组及其应用	176
10.4	FORTRAN77有关扩展与限制	179
	习题	190
11	文件及其应用	193
11.1	文件的种类和性质	193
11.2	读/写语句与辅助输入/输出语句	195
11.3	FORTRAN77有关扩展与限制	196
11.4	文件的应用	202

习题	218
附录 1 FORTRAN语句一览表	219
附录 2 FORTRAN的标准函数表	223
附录 3 FORTRAN66与FORTRAN77的主要差异	226
附录 4 字符—ASCII代码—EBCDIC代码对照表	228
附录 5 习题11.1参考答案	231
参考文献	234

1 计算机、语言和程序设计

随着计算机的普及和推广,有愈来愈多的科技人员、管理干部和大学生,迫切地希望学会使用计算机作科学计算和事务处理。人步入“计算机世界”,自然要使用一种工具与计算机交流信息,这个工具通称为计算机语言。当前,使用通用计算机,人们不仅要告诉计算机“做什么”,还要告诉计算机“怎样做”,怎样做的步骤要用一系列的语句(或指令)表示。这个能完成一定任务的语句(或指令)序列就叫**程序**。而编写程序的过程就叫**程序设计**。计算机的一切操作是按程序执行的,所以要想使用电子计算机,就必须学会计算机语言程序设计。

FORTRAN语言是国内外广泛应用的计算机语言之一。

这一章先简单地介绍一下计算机组成、计算机语言和FORTRAN语言程序设计,使读者对本课程及它的环境有个概括的了解。

1.1 计算机简介

世界上第一台电子计算机ENIAC是1946年问世的。计算机的发展非常迅速,应用特别广泛,对人类社会的发展起着巨大的推动作用,致使人类社会进入了一个新阶段——“信息社会”。下面简单地介绍一下计算机的特点、组成和工作过程。

1.1.1 计算机的特点

(1) 具有计算能力。速度快,精度高,可靠性强。

计算机的速度已经可以达到每秒几十亿次,一般为每秒几百万次到每秒几千万次。这样的速度是人工计算不能想象的。计算机的计算精度一般为有效数字6位,也可以为十几位,甚至上百位。这也是任何其它计算工具所望尘莫及的。计算机的可靠性很强,它可以连续工作几个月、几年不出错误。

(2) 具有记忆能力、计算机能存储程序和数据。

(3) 具有逻辑判断能力。计算机能对两个信息进行比较,根据比较结果决定下一步该执行什么。例如,计算机可以和人博奕,就是一个好棋手也常因一步之差而败在计算机的手下。

概括地说,计算机是一种能高速、自动、准确地进行计算的机器,已被广泛地用于科学计算、数据处理、自动控制、人工智能(如机器人)和计算机辅助设计等方面。

1.1.2 计算机的组成

计算机并不神秘。计算机的基本功能是“计算”,因此它的组成和人手工计算很相似,可分为两大部分,一部分是机器系统(硬件),另一部分是程序系统(软件)。

(1) 硬件

如图1-1所示,计算机硬件分五大部分。

1) 运算器。主要用来作算术运算和逻辑运算,相当于算盘。

2) 存储器。用来存储程序和数据,具有记忆功能,相当于纸。存储器的特点是“取

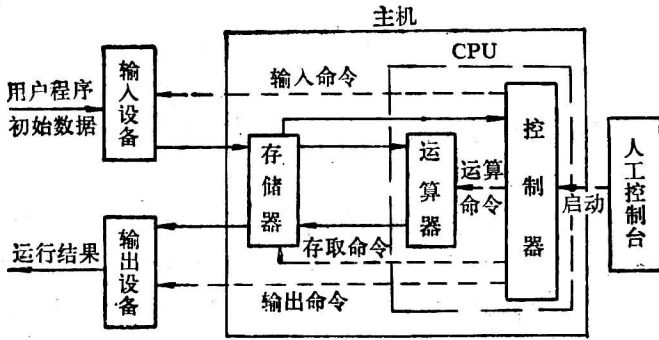


图 1-1 计算机硬件结构示意图

←表示数据传送路线

←←表示控制信号传送路线

之不尽，存之储新”。

3) 控制器。通过指令序列的执行，控制、协调各部分的操作，部分地代替了人脑的工作。

4) 输入设备。用来输入程序和初始数据等。常用设备有键盘、磁盘机、磁带机、卡片输入机和光电机（纸带）等。

5) 输出设备。用来输出程序清单、计算结果等。常用设备有显示器、打印机、磁盘机、磁带机、卡片输出机、穿孔机（纸带）和 $x-y$ 绘图仪等。

运算器和控制器合起来叫中央处理机（CPU）。CPU再加上存储器叫主机。输入和输出设备统称为外部设备。计算机的机器系统又称为裸机，或称为硬件。

一台裸机要正常工作和充分发挥效能，还需要配上功能完善的程序系统（软件），以使计算机能按照用户的指令自动地进行各种操作。

（2）软件

确切地说，软件是程序系统和文档资料（有关程序资料及使用说明等）的总称。这里是把文档资料看作程序不可缺少的附件，关于这一点今后不再单独说明。简单地说，软件是程序的总称，它又可分为

1) 系统程序（系统软件）——常指与硬件配套出售的基本软件，包括管理整个计算机的操作系统，翻译各种语言的编译系统以及检查、诊断计算机自身的服务系统等。

2) 应用程序（应用软件）——指计算机软件公司为便于用户使用计算机而开发的程序，通常以软件包的形式出售，例如数学计算软件包、绘图软件包和企业管理软件包等。

有了硬件和软件就构成了计算机完整的系统，那么这样一台计算机又如何工作的呢？

（3）计算机运行的示意过程

如图1-1所示，计算机运行过程可分为如下几个步骤：

- 1) 启动（开机）。通过人工控制使机器进入系统软件控制之下。
- 2) 输入用户程序和初始数据。使用计算机的人用计算机语言表示出来的解题步骤称为用户程序。
- 3) 运行用户程序，输出计算结果。
- 4) 如果还有程序运行则重复第二、第三步，直至所有程序运行完毕。
- 5) 关机。

1.2 计算机语言概述

1.2.1 机器语言

如前所述，计算机语言就是人与计算机交流信息的工具。这种工具最初级的形式就是二进制代码，即按特定规则由0和1组成的一定位数的数字代码，它能控制计算机进行特定的动作（例如加、减、取数、存数等）。由于这种二进制代码能被计算机识别，所以称为机器指令。这些指令的集合称为机器语言。机器语言是因机而异的，就是不同机型，机器语言不同，因此可移植性差。用机器语言写出来的程序称为机器语言程序，这种程序的优点是计算机能直接运行，效率高。例如，求两个数值之和（56+43），用Z-80指令可编程序如表1-1所示。

表 1-1 计算56+43的Z-80机器语言程序

存储单元地址 (16进制) ^①	存储单元内容 (2进制)	说 明
00 40 00 41	00 11 11 10 00 11 10 00 (56)	取56送A累加器
00 42 00 43	00 00 01 10 00 10 10 11 (43)	取43送B寄存器
00 44	10 00 00 00	把B加到A上
00 45 00 46	00 11 00 10 01 00 10 00	把A内容送到0048单元
00 47 00 48	00 00 00 00 01 10 00 10 (99)	空操作 (存结果的单元)
00 49	01 11 01 10	停机

① 16进制的基数 0 1 2 3 4 5 6 7 8 9 A B C D E F
对应的10进制数 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

作为一个用户最关心是，如何学习和使用计算机语言，通过上述小小的例子，可以看出机器语言难学难用，而且易错难改，十分枯燥无味。为了便于记忆，计算机专家们又创造了汇编语言。

1.2.2 汇编语言

汇编语言是用特定的助记符号代表数字代码，帮助人们记忆，它和机器指令基本上是一一对应的。这个改进不算大，所以人们把机器语言和汇编语言通称为低级语言。上述计算用Z-80汇编语言可编程序如表1-2所示，汇编程序语义和机器语言程序一一对应。

1.2.3 高级语言

第一台电子计算机问世后，又经过十年的努力，到了1956年，在IBM704机上才实现了FORTRAN I。这是由低级语言到高级语言的飞跃。高级语言之所以高级是因为它是用英语和人们熟悉的数学公式来表达的，并且又能为计算机所接受，具有较好的通用性。上述例子的FORTRAN程序，如表1-3所示。

这里存在一个问题，即上面提到电子计算机能直接识别二进制代码，但又如何识别这

表 1-2 计算56 + 43的Z-80汇编程序

汇编语言程序	说 明
START, LD A, 38H	38H表示38是16进制数
LD B, 2BH	2BH表示2B是16进制数
ADD A, B	A为累加器
LD (48H), A	B为寄存器
NOP	空语句
HALT	停语句

表 1-3 计算56 + 43的FORTRAN程序

FORTRAN程序	说 明
N=56	把56赋给N
M=43	把43赋给M
L=M+N	把M+N之和赋给L
WRITE (6, 20) L	输出L的值
20 FORMAT (iX, I4)	输出形式说明
STOP	程序停止运行
END	程序结束

些字符呢？事实上，高级语言程序，通常称为源程序，输入计算机之后，先要被翻译成机器语言程序，通常称为目的程序，然后再执行。这个翻译工作是由一个系统软件——“编译程序”完成的。不同的机器、不同的高级语言有不同的编译程序。FORTRAN语言的编译程序，通常叫做FORTRAN。显然，如果没有FORTRAN编译程序，计算机是不能运行FORTRAN程序的。编译程序的功能，如图1-2所示。

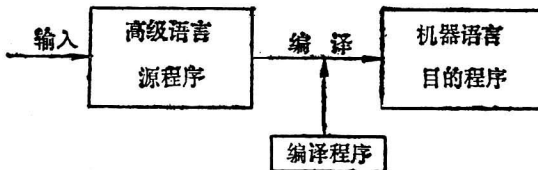


图 1-2 编译程序的功能

目前国内外比较常用的高级语言有十几种，分别具有不同的特点，现略举几种如下：

- BASIC (小型会话式语言)
- FORTRAN 66, ALGOL 60 (适用于科学计算)
- FORTRAN 77 (适用于科学计算, 数据处理)
- COBOL (适用于数据处理)
- PL/1, ALGOL 68 (大型通用语言)
- PASCAL (结构程序设计语言)
- C (系统程序设计语言)
- BCY (汉语拼音的ALGOL-60)

1.2.4 FORTRAN语言的优点

FORTRAN来源于英文“FORmula TRANslation”，意思是公式翻译。美国国家标准协会(ANSI)于1966年公布标准FORTRAN X3·9—1966(简称为FORTRAN 66,在表中记作F66)。于1978年4月又公布FORTRAN 77,这个标准又分为子集FORTRAN,和FORTRAN(在表中记作F77)。为了便于读者比较和查找,在附录1中,给出了FORTRAN语句总表。FORTRAN语言有如下优点:

(1) 易于学习。FORTRAN语言和人们习惯用的英语和数学公式很相似,只要注意和数学的规定作比较,是很容易学习的。

(2) 可移植性好。FORTRAN是标准化程度较高的语言,按FORTRAN标准规定编写的程序,不作修改或作少量修改,即可输入到不同的计算机系统中使用。

(3) 速度快。FORTRAN语言诞生在汇编语言广泛使用时期,为了与汇编语言争生存,所以在提高速度方面作了比较大的努力。FORTRAN语言在高级语言中速度是相当快的。

(4) 应用广。FORTRAN语言是最早在机器上实现的高级语言,目前可在大型、中型、小型和微型机上使用,并有极丰富的软件成果。FORTRAN 77扩展了控制语句和字符型数据,增强了数据处理功能。

1.3 FORTRAN程序设计概述

1.3.1 用计算机解题过程

学习FORTRAN语言进行程序设计是本课程的中心任务。下面通过一个例子来说明在使用计算机解题过程中, FORTRAN语言程序设计的地位,以便使读者抓住学习重点。例题中一些细节,当前不必深究。

A 明确问题

在着手解决问题之前,首先要弄清问题的条件和最终要求。例如,“若A、B两点之间有着不可逾越的障碍,求A、B两点之间的距离”。这样的问题就很明确。

B 建立模型

一个现实的问题,往往比较复杂,影响因素是多方面的,这时要进行试验分析,抓住主要矛盾,丢掉次要的矛盾,把原来的问题化作一个理想的模型。例如,对于上述测量A、B两点间距离这样一个问题,根据测量学,选择一点C,如图1-3所示,于是可测得:

$$a=1.5 \quad b=3.2 \quad \alpha=0.8 \text{ (弧度)}$$

这样,就把实际问题化作“已知三角形两边和夹角,求第三边c”的数学问题了。这一步

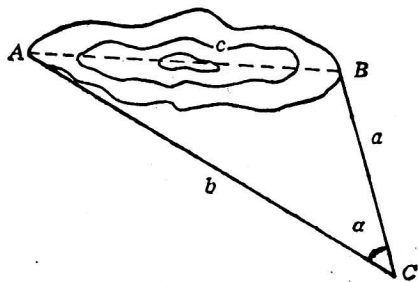


图 1-3

也叫建立数学模型。

C 算法设计

所谓算法设计，就是把数学模型规定的任务，转化为具体的解题步骤，即产生解题的逻辑蓝图。如果说数学模型规定“做什么”，那么算法设计就是解决“怎样做”。例如对于上述问题，由三角学知道，

$$c = \sqrt{a^2 + b^2 - 2ab\cos\alpha}$$

把已知值代入，即可算出 c 。

D 编写程序

上述问题，用FORTRAN语言可编程序如下：

```
C      COMPUTING THE SIDE OF A TRIANGLE
      READ (5,5) A, B, ALPHA
5     FORMAT (3F4.1)
      D=A*A+B*B
      -2.0*A*B*COS (ALPHA)
      C=SQRT(D)
      WRITE(6,10)A,B,C
10    FORMAT(1X,3F10.4)
      STOP
      END
```

E 编译运行

用户程序通过编译程序的处理，使语法是否有错得到检查。如果用户程序不存在语法错误，则将通过编译生成目的程序，并且程序中用到的一切变量将分配到相应的存储单元。运行目的程序时，首先读入初始数据：

```
. . . 1 . 5 . 3 . 2 . 0 . 8 .
```

最后，输出结果如下

```
. . . . . 1 . 5 0 0 0 . . . . . 3 . 2 0 0 0 . . . . . 2 . 4 0 8 7
```

F 结果分析

通过结果分析可以检查是否还隐藏着逻辑性错误。上述解题过程，可以概括地用图1-4来表示。

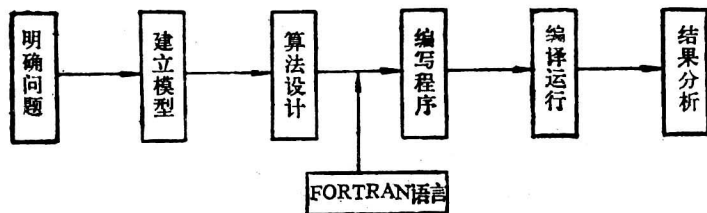


图 1-4 用计算机解题步骤

本课程的重点是讲述FORTRAN语言和程序设计，其它步骤也会涉及到，但不作为重点。例如编译运行，将涉及到计算机原理、操作系统和编译方法等方面的知识，这都是计算机方面的专业课，为了满足上机实习的需要，仅做如下介绍。

1.3.2 编译运行

FORTRAN语言程序设计是实践性很强的一门课。要学好它，不仅要学习理论（语法规则以及程序设计方法），还要学会编译运行程序，（简称上机操作），使理论与实践结合起来。上机操作步骤因计算机系统不同而稍有不同，现在在IBM PC/XT硬盘上建立了FORTRAN系统进行操作为例，简单地介绍一下上机操作步骤，如图1-5所示。

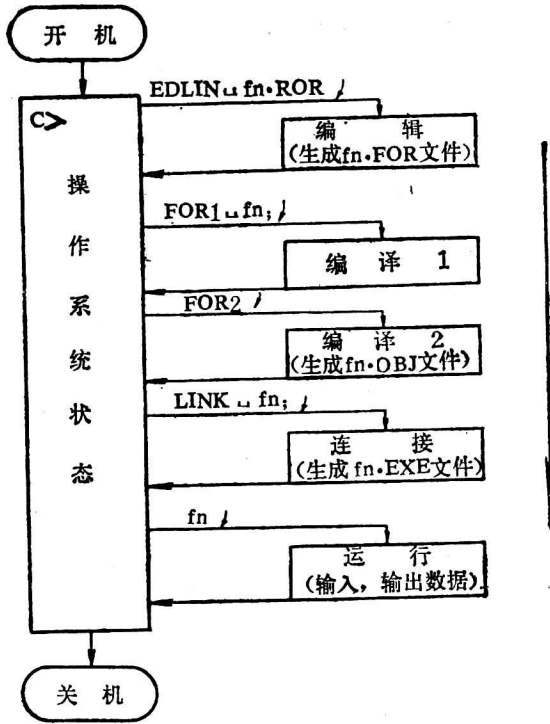


图 1-5 IBM PC上机操作步骤

编译运行的全部过程都是在操作系统支撑下进行的。操作系统处理的对象都叫文件，如源程序文件、目标文件和数据文件等。编译运行操作分四个阶段，即编辑，编译，连接和运行。

A 编辑

在编辑程序的支撑下，可以编写新文件或修改旧文件，通过键盘和显示器作对话式的操作。

EDLIN fn.FOR 是操作系统命令，它表示调用行编辑程序，编辑文件fn.FOR，fn为任选的文件名。若这个文件名是新的，就可以编写新文件，即把一个程序用插入子命令写入。若这个文件名是旧的，就可以把原有旧文件调出，并用列清单子命令可以把它显示出来。另外还可以用删除，置换、移动等子命令处理文件。最后存盘结束，返回操作系统状态。编辑结束可以生成fn.FOR文件。

输入源程序的方法是多种多样的，除行编辑之外，还有全屏幕式编辑（如，C-10机）。此外，还可以用卡片、纸带、软盘等输入源程序文件。

B 编译

编译是把源程序翻译成浮动目的程序。在编译过程中，如果发现程序有语法错误，则

会打印或显示“出错信息”，把出错的位置和性质提示给用户，并作适当处理，如跳过、终止编译等。

通常，调用FORTRAN 66编译程序只需作一次编译，而IBM FORTRAN是FORTRAN 77的子集FORTRAN，它的编译程序比FORTRAN 66的要长，为解决存储容量的不足，所以一次编译被分成FOR1和FOR2两次编译。注意，这两步要连续进行，否则易于出错。如果编译通过，则生成fn·OBJ文件。

C 连接

连接是把浮动目的程序，以及所调用的标准函数等连接生成一个目的程序，即机器语言程序，以备运行。在连接过程中，如果发现程序有错，则会发出“出错信息”（显示或打印），把出错位置和性质提示给用户，并作适当处理，如终止连接等。应该指出，不少计算机是把连接作为编译的一部分，无需单独操作。如果连接通过，则生成fn·EXE文件。

D 运行

运行就是让计算机执行经过上述操作生成的fn·EXE程序。这时，只要送入文件名fn，即可调入运行，并且可以反复调入运行。在运行过程中，如果发现有错误，也会打印或显示“出错信息”，把出错的位置和性质提示给用户，并作适当处理，如溢出、终止运行。如果运行正常，当执行到输入语句时，则等待键入数据，或从指定数据文件中自动取入。当执行到输出语句时，则显示或打印数据，或送入指定数据文件，直至停语句运行完毕，即fn·EXE程序运行结束后会自动返回操作系统。即重新置于操作系统控制之下。

注意事项：

(1) 要作好上机前的准备工作，包括：

- 1) 编好程序，作好静态检查，选好初始数据，并预示结果。
- 2) 把程序和数据写在程序纸上，至少要抄写工整。

(2) 在编译运行过程中，无论哪一步有错误信息出现，都要全力找出引起错误的原因，并及时返回到编辑状态，对源程序作适当修改，切莫带着错误进入下一步，否则是徒劳无益的。

(3) 要作好上机后的结果分析工作。上机实习之后，如果没有运行结果，则要找出原因，供以后借鉴。如果有运行结果也不要认为完事大吉，还要分析结果，检查是否还隐含逻辑性错误。例如改变输入数据，是否会出现不合理的结果？要力争达到：“程序能做它该做的事，而不做它不该做的事。”

关于编译运行更详细、更具体的叙述，读者可查看所用机器的上机操作说明和FORTRAN编译程序。

习 题

- 1.1 什么叫程序？什么叫程序设计？为什么要学习程序设计？
- 1.2 计算机有哪些特点？有哪几个主要方面的应用？
- 1.3 简要说明计算机由哪几个部分组成？存储器的特点是什么？
- 1.4 FORTRAN编译程序的作用是什么？
- 1.5 简要说明FORTRAN语言在用计算机解题过程中的地位。

2 FORTRAN初步知识

通过前一章的学习,对计算机的基本组成,以及FORTRAN语言和程序设计在计算机应用中的地位有了概括地了解。这一章介绍FORTRAN语言的初步知识,包括字符集、数据、语句和FORTRAN程序。

2.1 FORTRAN字符集与符号名

2.1.1 字符集

和自然语言不同,计算机语言允许使用字符的个数是有限的。FORTRAN 66规定:书写FORTRAN语言程序只允许使用如下47个字符,这些字符可分为三大类。

(1) 字母(共26个)

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

(2) 数字(共10个)

0 1 2 3 4 5 6 7 8 9

(3) 特殊字符(共11个),如表2-1所示。

上述字符有的含义是明显的,如,+,-,(,)等,和数学上含义一致;而有的含义就不大明显,如,*,/等,它们在不同场合有着不同作用,今后将分别给予说明。

表 2-1 特殊字符表

名称	字符	名称	字符	名称	字符
加号	+	左括号	(等号	=
减号	-	右括号)	空格	
星号	*	小数点	.	币符	\$
斜线	/	逗号	,		

几点说明:

(1) 空格 为了程序易于阅读,在书写程序时,可在适当地方留一个或几个空格。在书写时,常用□表示空格;而在打印或显示时,空格就是空白,没有任何标记。

(2) 币符 FORTRAN没有统一规定它的书写形式。\$是美国规定的。我国规定用¥。由于没有统一规定,所以通常很少使用。

(3) 在FORTRAN程序中,一般地不允许使用这47个字符以外的字符。例如使用 α 、 β 、 γ ,或 π 、 θ 、 ω 等都是不允许的。

2.1.2 符号名(标识符)

在日常生活中,人们为区别各种东西,要给每个东西取个名字。同样,程序中常用到各种变量,函数等,也要分别取个名字。在FORTRAN语言中,它们都是以符号名表示的。因此,“符号名”是最常用的基本概念。

A 符号名的形式

符号名是字母开头的，六个以内的字母数字序列。例如

X, Z1, ROOT, A1234D均为合法的符号名。

3X, Y(1), T-2, π 均为非法的符号名。

B 几点说明

(1) 标准规定，符号名最大长度不超过6个字符，如

N123456 AVERAGE

按标准规定都是非法的。

(2) 符号名的选择，原则上说，只要符合符号名形式的规定，程序员有权自由选择。习惯是：常用从简，专用从意，习惯相近，使用方便。这样，可以提高程序的可读性。

例如在数学中使用 $a, b, \alpha, \pi, I_{max}$ ，而在程序中可使用 A, B, ALPHA, PI, IMAX。

2.2 数据 (一) 整型常数与实型常数

我们常说，编写程序就是为了处理数据。反过来，数据就是程序处理的对象。当然，对于不同的程序，处理的对象也不同，有的是数值，有的是文字，还有的是文件。分类处理和存储数据，目的是为了提提高效率（即，节约存储单元，提高处理速度）和减少错误。

FORTRAN 66规定：FORTRAN数据按它的性质分为六个“类型”、三种“形式”，如表2-2所示。

算术型又叫数值型。LOGICAL型和Hollerith型又叫非数值型。数组是同类型变量

表 2-2 数据类型与形式

类	型	形 式		
		常 数	变 量	数 组
算 术 型	INTEGER (整) 型	✓	✓	✓
	REAL (实) 型	✓	✓	✓
	DOUBLE PRECISION (双精度) 型	✓	✓	✓
	COMPLEX (复) 型	✓	✓	✓
	LOGICAL (逻辑) 型	✓	✓	✓
	Hollerith型	✓	×	×

注：✓表示有，×表示没有。

的有序集合。例如数学中的向量、矩阵都是数组。数组的每个元素叫数组元素。本节讲述整型常数和实型常数，下一节讲述整型变量和实型变量。其它的数据留以后再讲，具体安排请参看目录。

FORTRAN语言的整常数与实常数都是十进制数。它们与在数学中的表示差不多，如表2-3所示。

2.2.1 整常数与实常数的形式

常数本身既定义了值，又确定了类型，具体规定如表2-4所示。

表 2-3

FORTRAN语言中的常数		数学中的常数
整	型	21
实	型	3.1416 21.0 -12.1E-8 1.0E4
		21 3.1416 21 -12.1×10 ⁻⁸ 10 ⁴

表 2-4 整常数与实常数的形式

类 型	形 式	存储单元	特 点
整型	$[\pm] a$	1	数字序列, 精确值
实型	小数形式 (基本实常数) $[\pm] a.b$	1	带小数点, 近似值
	指数形式 $[\pm] a.bE[\pm] n$ $[\pm] aE[\pm] n$	1	带E, 近似值

注: 1. a 、 b 、 n 是数字序列 2. “+”可以省略。 3. $[\pm]$ 表示可以是“+”或“-”。

例如: 0, -327, 1000, 12000 为整常数。

0.0, -327.0, 1.0E3, 1.2E4 均为实常数。

说明:

(1) 在 FORTRAN 语言中, 小数点前后的 a 或 b 为 0 时, 0 可以省略。但小数点不可省略。 a 、 b 同时为 0, 不能同时省略。为和其它高级语言一致, 也为了便于阅读, 建议不要把 0 省略。

(2) 常数之前如果不带符号称为无符号常数。

(3) $E[\pm]n$ 叫实数的指数部分, 它是实常数的一部分, 它本身不能构成实常数。例如 10^7 可以写作 $1.0E7$ (或 $1E7$), 但若写作 $E7$, 则是错误的, 此时编译将认为它是一个符号名。

(4) 一个实常数按上述规定可以有多种书写形式, 如 12.3 , 可以表示为 $0.0123E3$ 、 $0.123E2$ 、 $1.23E1$ 、 $12.3E0$ 、 12.3 、 $123E-1$ 、 $123.0E-1$ 等形式, 但是无论使用哪种规定的形式, 该数输入计算机后, 都自动转化为与该数等价的规格化浮点数—— 0.1111011×2^{110} (即 $0.123E+02$) 送入存储器 (有时精度会有小的差异)。

2.2.2 数值范围与精确度

从数学理论上说, 整数与实数的取值范围是 $(-\infty, \infty)$, 它的精确度可以到任意位, 但在计算机中, 数是存放在存储器的存储单元中, 这就限制了数值的范围与精度。FORTRAN 标准规定: 一个整常数和实常数各占一个存储单元。由于机型不同, 存储单元的位 (Bit) 数不同, 所以表示数的范围和精度也不同, 如表 2-5 所示, 这是需要程序员注意的。

说明:

(1) 上溢、下溢和零。在 IBM PC 机上实数的绝对值大于 10^{38} 叫上溢, 这时, 程序运行将会中断。实数的绝对值小于 10^{-38} 叫下溢, 这时计算机能自动地进行零处理。

(2) 存储单元、字节和位。若把存储器比作一个大楼, 则存储单元可算是一个房