学习

FORTRAN 语言学习指南

冯博琴　姚全珠　傅长龙　编著

机械工业出版社

指南

FORTRAN语言学习指南

冯博琴　姚全珠　傅长龙　编著

机械工业出版社

本书是FORTRAN语言学习方法的指导书，对初学者及钻研FORTRAN语言的读者均有所帮助。全书共八章，分为三个部分：第1~4章是第一部分，介绍FORTRAN语言的重、难点内容；第二部分是第5章，介绍上机操作的技巧；第三部分是第6~8章，介绍进一步的知识，内容包括编程风格、FORTRAN5.0高级编程指南和学习第二种程序设计语言的方法。在每一章中，作者着重指出最重要的概念、易犯的错误，解释疑难问题和提供FORTRAN应用技巧和体会。

本书可作为大专院校FORTRAN语言教学用的辅助教材，也可作为各类人员学习FORTRAN语言自学用的参考书。

图书在版编目（CIP）数据

FORTRAN语言学习指南/冯博琴，姚全珠，耿庆龙编著.—北京：机械工业出版社，1996
ISBN 7-111-05172-6

Ⅰ.F… Ⅱ.①冯… ②姚… ③耿… Ⅲ.微型计算机-应用 Ⅳ.TP3[illegible]

中国版本图书馆CIP数据核字（95）第1427[illegible]号

出版人：马九荣（北京市百万庄南街1号 邮政编码100037）
责任编辑：杨存彩 版式设计：[illegible] 颖
封面设计：姚 毅 责任印制：卢子祥
[illegible]印刷有限公司印刷·新华书店北京发行所发行
1996年[illegible]月第1版第1次印刷
787mm×1092mm 1/16·15.[illegible]印张·[illegible]千字
0 001—[illegible]册
定价：19.80元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

前　言

从本世纪的 60 年代开始，一场新的信息革命悄然来临，它把人类带进信息化社会，与这个社会相适应的社会技术是信息技术，它的核心是计算机技术，由于这项技术在人类发展史上大大改变了人类创造物质财富和精神财富的方式、方法、过程和结果，也改变了社会结构和人类自身的生活方式、习惯等，因此引起了各部门空前的重视。

21 世纪将是一个信息化社会，这个社会对人才素质和知识结构都会提出更高的要求，对于高等教育的各个学科，计算机的作用已不仅仅是一种工具，而是各学科本身的重要组成部分。加强计算机基础教育不仅是为了提高计算机本身的水平，而且将为提高其他学科的教育水平打好基础。由此可见，计算机基础教育既是文化基础教育、人才的素质教育，又是强有力的技术基础教育，加强这种教育不仅是信息化社会的需要，也是各学科发展的需要。计算机教育水平高低已成为评价学校教学质量的重要指标，学生本人的计算机应用能力反映了个人的素质，影响着他的竞争能力，因此，目前理、工、文、管、农、医等各类学校都在努力使计算机教育上一新台阶。

目前计算机程序设计语言课是计算机基础教育的最基本内容之一，因为大多数用户还是用程序设计语言作为编制程序的手段来利用计算机的。根据我们的教学体会，教学和教材都存在着这样问题：学生大多数是计算机的初学者，对计算机解题过程和程序设计非常陌生，学时又十分紧张；教材编写受格式、篇幅限制，学生学习的困难和问题颇多，不少人是带着许多问题结束这些课程学习的。

作者求学时，读过吴映奎先生编写的《高等数学学习方法指导》，受益匪浅，受此启发，觉得有必要跳出教材，站得稍高些，对程序设计语言的难、点、热点进行深入一点的讨论，因此组织了几位长期从事此项工作的教师编写了《FORTRAN 语言学习指南》和《C 语言学习指南》。

这些书不同于教材，总体结构上包括三部分，即语言的基本内容、上机指导和进一步的知识。所选择的内容基本上是该语言的重点、难点或热点。每一章围绕一个主题，一般都包含知识概要、易犯的错误和释疑，以及有关的应用技巧或体会。

作者希望本书对以下两种人有所帮助，一种是正在学习这门语言的读者，不论他使用何种教材，本书都能为他解难，成为他的辅导老师；另一种是想提高编程能力的读者，因为各章都有这方面的技巧、经验和体会，同时还包含一些新内容，如 FORTRAN5.0 的介绍、编程风格、图形处理和混合编程等。这些在教材中一般“无暇”顾及，但在应用中是热点，学生很感兴趣。

本书由冯博琴任主编，姚全珠编写第 1、2、3、5 章，傅其龙编写第 3、4、6 章，乔晓彤编写第 7 章。由于作者水平有限，取材和写法未必合适，疏漏和谬误之处也在所难免，希望同行和读者指正。

冯博琴

1995.7. 于西安交通大学

目　录

第1章 概 论

1.1 浅谈计算机硬件结构

在计算机问世之前，人们在解决一些计算问题时也借助于某些计算工具，例如算盘、手摇式计算机、对数计算尺等。这些计算工具是非智能化的，需要人逐步操作。在用算盘计算 7 －32×2 时，首先要将原始数据 74，32，2 和运算步骤记录在纸上，运算时的步骤为：①32×2＝64；②将中间结果（64）记在纸上；③74 －64＝10；④将结果 10 记录在纸上。

从上述运算过程中可以看到，算盘是专门用于运算的工具，纸被用来记录原始数据、运算步骤、中间结果及最终结果。只有在人脑的正确指挥下，按预先制定的运算步骤对原始数据及中间结果进行正确运算，才能得到正确的最终结果。计算机的工作原理是对人运算过程的模仿。计算机硬件上的运算器相当于算盘，存储器的作用与纸相同，控制器相当于人脑。由于计算机中的信息是以电磁信号形式存在的，要想把原始数据及运算步骤存入计算机中，就得借助于输入设备；要想把运算结果输出，就得借助于输出设备。一台计算机的硬件组成及各部分的联系如图 1-1 所示。

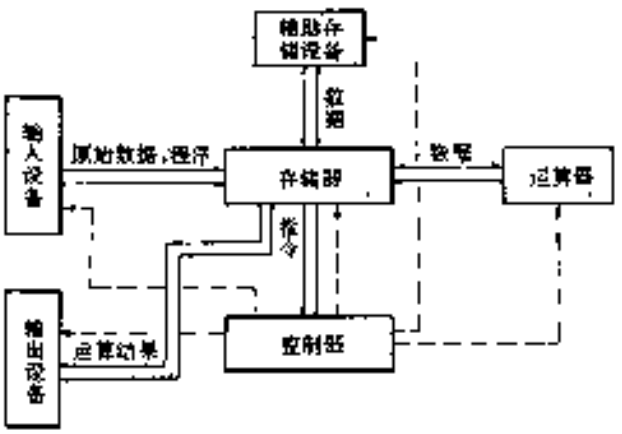

图 1-1 计算机各部分联系示意图

（虚线表示控制命令）

计算机的工作过程如下：用户首先通过输入设备（例如键盘、卡片输入机等）将原始数据、运算步骤（程序）输入到存储器中。在控制器的控制下，把用命令形式描述的运算步骤，逐一取到控制器并对其进行分析（译码），然后发出能实现该运算步骤的一系列控制操作（如把保存数据存储器送至运算器，对其执行规定运算，并把结果送回存储器等）。在控制器的控制下，根据程序描述，可把某些运算结果送至输出设备输出（例如打印机，显器等）。由于全部程序及数据已全部输到存储器，控制器能够从存储器中取一条指令，对其分析，并按其要求控制其他部件协调地执行该命令规定的操作，当此指令执行完后，再取下一条指令……。用面向个程序的执行过程不需人工干预，自动化程度很高，程序和数据不用的时候，往往放在某种辅助存储设备内，例如磁盘或磁带，当要用时可随时调入主存储器内。

1.2 系统软硬件之间的关系

上节中介绍了计算机的基本硬件特征，这是裸机的形象。一般没有软件的计算机，给计我们直接面对这样的机器系统编程将是十分困难的。首先计算机中的程序和数据都是使用二进制编码，例如想要做 7－10 这样的运算，用 PC 机指令助记符形式表示的程序为：

```
MOV AL，7
ADD AL，10
HLT
```

而上述助记符方式的程序只有翻译成机器码，计算机才能识别并执行。

```
MOV AL，7—[1011 0000
          [0000 0111
ADD AL，10 [0000 0100
           [0000 1010
HLT—       1111 0100
```

若我们把想要做的事情都以机器码的方式描述显然十分困难。其次，当我们才开始学习计算机课程的时候，想要把从键盘上输入的数据加上37，结果输出到打印纸指定位置上，则必须学会编制键盘及打印机管理程序。这涉及电学、机械学等方面的许多知识，显然也很困难。再者若多个用户同时使用一台计算机，如何对计算机系统的软、硬件资源（存贮器、CPU、I/O设备、文件系统等）进行有效管理，使得每个用户的程序都能得到安全有效的执行，系统资源得到最充分的利用，这是用户无法考虑的问题。为了使十用户对计算机的使用，向用户提供一个友好的服务界面，计算机工作者提出了扩充机器及分层机器的概念。简单的分层机器结构如图1-2所示。

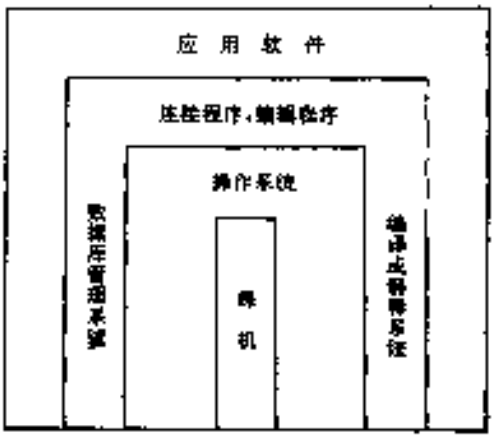

图1-2 简单的分层机器

操作系统是用户和计算机之间的媒介。用户通过命令调用的方式指挥操作系统来使用计算机。操作系统的目的是使用户能够以更灵活、更容易掌握的方式利用计算机的资源。操作系统中含有输入/输出设备管理程序，用户要输入/输出数据时，只要简单地以命令方式调用输入/输出程序就能达到目的。编辑程序在操作系统的支持下，帮助用户方便地输入、修改源程序。而编译/解释程序则是在操作系统的支持下，把用户用比较接近自然语言的高级语言所编写的程序，翻译成机器语言程序。编译/解释系统还提供了许多常用的函数子程序库。这些函数作为标准函数用户可在自己的程序中直接调用。连接程序则是在操作系统的支持下，对翻译成功的机器语言程序进行装配，使之成为能直接在计算机上运行的可执行程序。

可以看出，计算机系统软件是对硬件功能的扩充。面向用户的是功能扩充了的机器系统，称为虚拟计算机。从功能上可以认为软件是利用计算机本身提供的逻辑功能，合理地组织计算机的工作，简化或代替人们在使用计算机过程中的各个环节，给用户提供一个便于掌握和操作的工作环境。今后我们为用户开发的各类应用软件，也要注意软件包装，尽量给用户提供一个安全可靠、操作方便、面面优美、提示清晰、运行效率高的工作环境。综上所述，计算机系统的组成如图1-3。

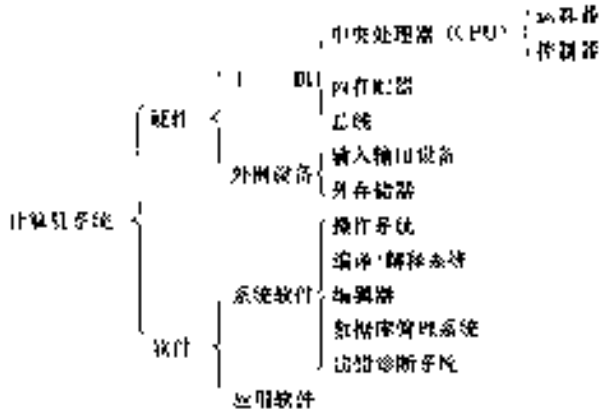

图 1.3 计算机系统组成

1.3 如何迅速学好第一语言

1.3.1 问题的提出

高级语言是一种比较接近自然语言的用以描述解题过程和算法的工具。在以往的教学过程中，同学们普遍反映第一门高级语言的学习中存在以下问题：

1）高级语言很抽象，尤其是不了解计算机的工作原理或知之甚少时，更是如此。计算机执行程序的过程是在机器内部电路中进行的，它不像机械运动那么直观。

2）高级语言规定很多，不象数学那样逻辑性强。数学具有公理、定理，有公理、定理的基础上进行推理演算。由于计算机科学涉及到电子、机械学等多门学科知识，要让普通用户直接面对系统硬件编程既不可能，也没有必要。计算机系统（包括软件、硬件）下作者已经把与硬件有关的管理工作编写成系统程序，用户可直接用高级语言和理想的虚拟计算机"交谈"，因此它的程序必须符合该语言的词法、语法规定。因而计算机语言很"死板"，语句中稍有写错一个字母也不行。

3）编程序很难，往往给出一个问题之后不知从何下手。

1.3.2 方法的探讨

针对上述问题，根据教学实践，我们认为在高级语言的学习中应遵守以下几条。

1. 纵观全局，有的放矢

在学习具体语言之前，首先应对高级语言的组成结构，具有什么功能，有那几类语句等有关高级语言基本知识有个初步了解。这将大大有助于以后对语言的学习和掌握。由于编译系统的存在，可以认为在学习高级语言时，面对的是一台理想的虚拟计算机。这种理想的抽象机已经超脱了硬件技术的许多限制，而特别侧重于人们的思维和表达的习惯。不与这样的机器交往时，使用的是高级语言。面级语言是面向过程的（近年来发展的某些语言如PROLOG，C++等是面向对象的），在设计程序时，不必考虑机器内部构造和不同机器的特点，只要将解题算法用高级语言语句描述出来，计算机就能执行。

在现实生活中的数据千差万别，但在不同的高级语言中，对要处理的数据都进行了分类，程序中每一个变量、常量或表达式的值，都属于确定的数据类型。每种数据类型都规定有相应的一组基本操作。变量对应于内存中的一个（组）存储单元。变量名实际上是为了便于在该存储单元中存取数据而给该单元起的名字。存储单元如同一个旅馆中的房间。变量的值是

指其name存储于该存贮单元中的数据，它如同旅馆某一房间中的客人，不同类型变量占用存贮单元的多少也不一样，就好比客房有单人间、双人间的差别。在程序设计语言中，使用数据类型的概念带来了程序的简明性，明白地告诉我们变量应该取什么类型的值，指出应该含什么样类型的数据上进行，从而可以防止诸多类似于给整型变量赋布尔值之类的错误，也便于计算机在实际执行程序之前，为变量各内存中分配存贮单元。

在不同的高级语言中，规定能使用的数据类型不完全相同，但一般可以划分为以下几类：

1）**简单类型** 是基本的不可再分的数据项，例如在FORTRAN中有整型、实型、双精度型、复型、逻辑型、字符型；在PASCAL中则有整型、实型、布尔型、字符型、枚举型和子界型。

2）**结构类型** 是由已知类型（例如简单类型）按一定规则构造而成的复合类型数据，包括数组、集合、记录和文件。

3）**指引元类型** 主要用于构造各种形式的动态数据结构，如链表、队、栈、树、图等。

用于描述数据结构的语句是说明型语句，一般放在本程序单元的首部；用于描述算法指挥结构的语句一般属于可执行语句，不同语言的可执行语句数量、语法结构、书写格式都不相同，但最基本最主要的都包含以下三种语句：

① 输入/输出语句

② 赋值语句

③ 控制语句：分支语句
　　循环语句

输入/输出语句是在程序的运行过程中进行人机会话的主要手段，借助于输入设备人可以及时地通过输入语句给变量赋所期望的值，在输出设备的支持下，通过输出语句可以及时输出经计算机运算所得的中间结果及最终结果。

赋值语句是数据处理过程中进行运算的主要手段，它的一般格式为：

变量名 = 表达式　　（在PASCAL中指：=，在FORTRAN和C中不指：）

其作用是首先计算表达式的值，再把计算所得结果转化为与变量一致的类型，并赋给变量。在书写表达式时应注意符合语言规定，在执行该语句时表达式应具有确定的值，这就是说包含在该表达式中的每个变量及函数，在此时应具有确定的值。赋值号左边必须是一个变量名，因为只有变量名才与内存中唯一确定的存贮单元相对应，可以存放计算机求得的值。

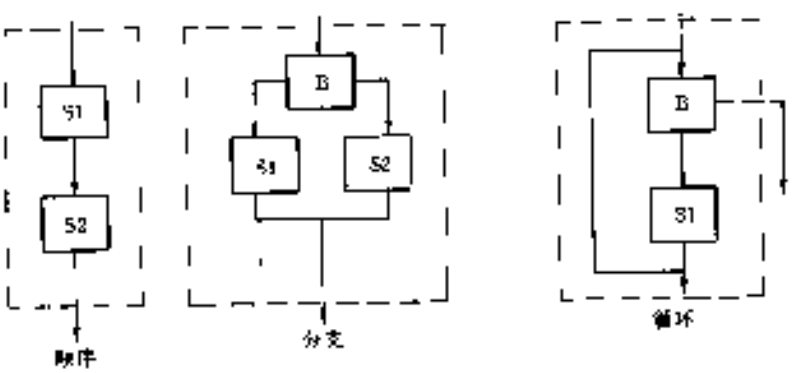

图1-4　程序的三种控制结构

控制语句主要用于改变程序执行的走向，在结构化程序设计中所提倡的三种程序基本结构是顺序结构、分支结构和循环结构，见图1.4。

其特点是每种结构中严格地只有一个入口和一个出口。用输入输出语句和赋值语句就可以实现只需顺序处理的算法，但现实生活中绝大部分情况并非如此。例如要根据一个同学的考试成绩来决定他是升级还是留级；又如统计一个班某门课不及格的人数时，对全班每一个人都要判断看他的成绩是否低于60分，若低于60分时，不及格的人数就要加1。为了实现类似于这样的算法，高级语言中设计了分支型和循环型控制语句。

为了完成程序结构，设置了过程和函数语句。借助于文件操作语句，程序员可以方便地将程序和数据保存在磁盘上或从磁盘读入内存。

2. 循序渐进

高级语言的语句比较多，编程技巧性也很强。但基本语句并不多，常用算法和技巧也屈指可数，在学习过程中应注意归纳和逐步掌握。比如在首次上机练习时，可先编写一个程序，不管这个程序有多么简单，这个程序可能只含有一个输出语句；再在原来的程序中增加一条赋值语句，并通过输出语句输出赋值结果；再在上述程序中增加一条输入语句，在赋值语句对输入变量的值进行处理后，由输出语句输出；接着改变输入输出语句的格式，重新执行该程序，看会有什么变化……。这样就掌握了学习的主动性，学起语言来会感觉轻松。

一个复杂的算法往往可以分解成一些基本的算法，对于课程中的一些常用的基本算法应注意归纳和总结。比如求累加和的数学模型为S=S+X；连乘积为P=P*T；求多项式的和为A=AX+T等等。这样在遇到问题后，就知道该问题属于哪种类型，该用什么样的算法去解决。例如要计算表达式（1+1/2+1/3+…+1/10）的值，若能认识到它是一个累加和的过程，这里X取1/I，I从1开始，到达10结束，按I+1→I的规律在变化，问题就迎刃而解了。

3. "粗"、"细"结合。

如前所述高级语言中的规定和需要记忆的东西比较多，而且不同的语言版本中有些规定不一样，比如整数的范围、实数的范围、它们在机器中存储时所占的字节数、输出时的有效位数、默认的宽度等。对这些有的不影响高级语言的初步学习，有的可通过上机逐步掌握的内容不必死记，可粗一些。但对基本语句、基本函数等的书写格式、作用、执行过程等，一定要记牢，才会灵活运用。

4. 利用框图读程序和写程序

框图是描述算法的有力工具，它比较直观，整体性强，在框图上对算法进行修改，比在程序上要容易得多。在编写一个程序或在读别人的较复杂程序时，可首先画出框图，再对照框图读程序或编写程序。

5. 多上机练习

高级语言程序设计是一门实践性很强的课程，只有通过多读程序、多编程序，多上机练习，才能提高编程和调试程序的能力。程序设计具有艺术性的特点，只有经过一定的实践才能掌握其要领。不能只满足于上课听懂了，更要注意作业是否会做了，上机调试是否得到了正确的结果。

6. 注意掌握结构化程序设计的思想，养成良好的程序设计风格

用一种基本结构设计出的程序，各个模块之间接口关系简单，可以相对独立地设计各个子模块，静态地分析控制关系，并验证它们的正确性。如果某一子模块需要修改，只要接口不变，就不会影响到其它子模块乃至整个程序，因此，具有这样结构的程序是好结构程序。尽管好结构程序的效率不一定最好，但它结构清晰，易于理解和验证，便于维护和移值。且随着计算机硬件的发展，运算速度越来越快，因而好结构程序受到人们的推崇，在学习程序设计时，一开始就应养成良好的程序设计风格，有关好的程序设计风格及其实现方法请参见 6.2 节。

第 2 章　基本知识解疑

2.1　FORTRAN 语言的一般知识

2.1.1　概述

FORTRAN 语言是用户与计算机之间"会话"的一种中介语言，它比较接近自然语言，用户易学易懂，出错易于修改。

例 2.1　用 FORTRAN 语言编写计算半径为 2 的圆面积程序。

```
C       EXAMPLE 2.1
        R=2.0
        S=3.14*R*R
        WRITE (*,*) S
        END
```

第一条语句的作用是使表示半径的变量 R 的值为 2；第二条语句的作用是按公式 S=πR² 计算半径为 R 的圆面积，并将结果存入变量 S 之中；第三条语句的作用是将变量 S 中的值显示在屏幕上。

上述程序称为FORTRAN 语言源程序，它是用户向计算机描述自己所要做工作的一种手段，但此程序只有经过FORTRAN编译程序编译成机器码之后，计算机才能执行。用户在用FORTRAN 语言描述自己的算法时，必须符合FORTRAN编译系统的词法及语法规则，FORTRAN编译器才能看懂。如上例中乘号必须用"*"；R=2 不能写作 2=R 等。在FORTRAN语言的学习中，对每一条语句必须记住它的一般格式、书写及用法。

2.1.2　数据为什么要分类

在日常生活及科学计算中，把数据分为实数与虚数、实数又分为整数与小数等。高级语言是描述算法的工具，必然涉及不同类型的数据。根据不同数据的特点和使用要求在计算机中采取了不同的存储方式。在 FORTRAN 中将常量分为 6 类：①整型常量；②实型常量；③双精度型常量；④复型常量；⑤逻辑型常量；⑥字符型常量。在微型计算机中对于字长为 16 位的计算机，一般用 2 个字节（16 位二进制数）来存放一个整数。字长为 32 位时，一般用 4 个字节（32 位二进制数）来存放一个整数。对于字长为 16 位的机子，其整数存放格式如图 2-1。

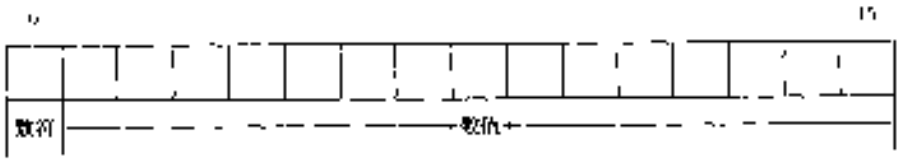

图 2-1　整数存放格式

当数值以补码方式表示时，其范围为－32768～32767。当数值在其表示范围内时，能够

被准确地转化为二进制方式，可按机内格式存放，没有转换误差。

在计算机内存中，实数的一种存放格式如图 2-2 所示。

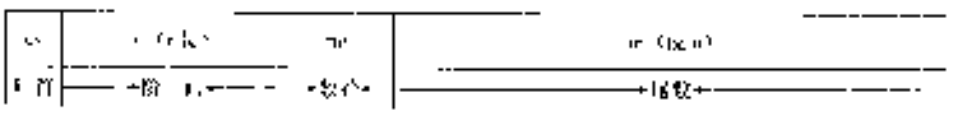

图 2-2 实数存放格式

若尾数用 n 位小数表示，则实数的表示范围根据阶码位数 r 和尾数位数 n 决定，其浮点数表示范围为：

$$-2^{2^r-1}\times(1-2^{-n}) \le X \le +2^{2^r-1}\times(1-2^{-n})$$

在 PC 机中，一般以 4 个字节（32 位二进制数）来存储一个实数，指数部分占 1 个字节（阶符占 1 位，阶码占 7 位），尾数部分占 3 个字节（数符占一位，尾数占 23 位），故实数的机内取值范围为

$$2^{-2^7}\times(1-2^{-24}) < |X| < 2^{2^7-1}\times(1-2^{-24})$$

即

$$2^{-128}\times(1-2^{-24}) < |X| < 2^{127}\times(1-2^{-24})$$

这个取值范围大体在 $10^{-38}\sim10^{38}$ 之间。

双精度数则用两倍于实型的存储空间来存放。例如若实型数据用 4 个字节存放，则双精度数用 8 个字节存放，其存放格式为指数部分仍然占 1 个字节（阶符 1 位，阶码 7 位），尾数部分占 7 个字节（数符一位，尾数 55 位）。双精度数的表示范围与实数相同，而其精度（有效数字位数）也要比数据多 1 倍以上。

复数在计算机中存放时，是以两个实数的序列分别存放其实部和虚部。逻辑型变量用一个字节来存储，用 -1 表示真，用 0 表示假，其格式见图 2-3。字符型变量也用一个字节来存放字符的 ASCII 或其它形式（EBCDIC）编码，其格式见图 2-4。

1 表示 True　　0 表示 False

图 2-3 逻辑型数存放方式

字符 'A' 的 ASCII 存放格式　　字符 'A' 的 EBCDIC 编码存放格式

图 2-4 字符型数据存放方式

不同类型的数据其用途不同，在其上所能进行的操作也不相同，在计算机中不同类型数据的精度、表示范围、运算结果也可能不同。例如在 FORTRAN 中 1.0/3.0=0.3333333 而 1/3=0，整数 123 能被准确地用二进制数 1111011 表示，而实数 0.1 则不能被准确地转化为有限位二进制数。整数的取值范围小，实数的取值范围大；整数运算速度快，用浮点格式表示的实数运算操作复杂、速度慢。在某些工程计算中，为了满足较高的精度要求，必须采用

双精度型数据，在数学中求方程的复根，电工学中对交流电路的计算，自动控制系统中传递函数的计算，都要用到复数运算。为了满足数据处理的不同需要，FORTRAN 对数据进行了分类。FORTRAN 中的变量被用来存放常数，是对存放常数的存储单元所起的名字，因此变量也相应地区分为整型变量、实型变量、双精度型变量、复型变量、逻辑型变量、字符型变量。

2.1.3 程序中使用变量的好处

FORTRAN 源程序中所使用的变量名，通常是以字母开头的字母数字序列，其前 6 个字符有效可用。变量的类型有 3 种说明方法：

1）隐含约定（又称 I—N 规则）：若程序中变量名以 I、J、K、L、M、N 六个字母之一开头，则默认为整型变量，以其它字母开头则默认为实型变量。

2）类型说明：若变量类型不准备遵循隐含约定，则可用类型说明语句对变量类型进行说明。

3）隐含说明：若想将某一字母开头的全部变量指定为所需类型，则可用隐含说明语句对变量类型进行说明。

在用上述 3 种变量类型说明方法对变量进行说明时，若发生矛盾，则以类型说明最优先，隐含说明次之，I—N 规则说明级别最低。

例 2.2 阅读下述程序并写出运行结果。

```
C       EXAMPLE 2.2
        IMPLICIT DOUBLE PRECISION (R)
        INTEGER R1
        R1=3.2
        R1=4
        WRITE(*,*) R1
        END
```

R1 既不遵从 I—N 规则，也不服从 IMPLICIT 说明，而是按 INTEGER 语句被说明为整型变量。当调用 FORTRAN 语言或译程序编译此程序时，计算机就会在内存中为我们（或四个）字做为一个整体，并将其命名为 R1。当执行语句 R1=3.2 时，计算机就在该单元中存入机（0000000300000011）$_2$，则变量 R1 的当前值即为 3。当执行语句 R1=4 时，则此存储单元的值变为（0000000400000100）$_2$，执行 WRITE（*，*）R1 会输出当前值 4。

以上说明了变量的命名规则，变量的类型说明方法，变量名的含义及变量的当前值概念，在程序中正确地使用变量具有以下好处。

1. 便于程序修改，防止数据不一致

例 2.3 要求一个半径为 0.213m 的圆周长、圆面积及球体的体积，不使用变量 R 表示半径时，用 FORTRAN 语言编一程序。

```
C       EXAMPLE 2.3
        L=2*3.14*0.213
        S=3.14*0.213*0.213
        V=4.0/3.0*3.14*0.213*0.213*0.213
        WRITE(*,*)L,S,V
```

```
      END
```

当把半径改为 0.214 时，上述程序必须修改 5 处之多，修改工作量大，稍不注意就会忘记修改某处，导致错误。可引用变量来表示。

例 2.4　若引用变量 R 来表示半径，编一程序。

```
C     EXAMPLE 2.4
      R=0.213
      L=2*3.14*R
      S=3.14*R*R
      V=4.0/3.0*3.14*R*R*R
      WRITE(*,*)L,S,V
      END
```

此时当把半径改为 0.214 时，只须修改第一条赋值语句即可。

在同一程序中，即使对同一个量前后用了不同的变量名，要想使它们一致时，只需加个等价语句问题就解决了。

2. 增强程序的可读性

好程序的设计风格强调变量名应尽量接近其实际含义，以增加程序的可读性。

例 2.5　把例 2.4 程序改写。

```
C     EXAMPLE 2.5
      PI=3.14
      RADIUS=0.213
      CIRCUMFERENCE=2*PI*RADIUS
      AREA=PI*RADIUS*RADIUS
      VOLUM=4.0/3.0*PI*RADIUS**3
      WRITE(*,*)CIRCUMFERENCE,AREA,VOLUM
      END
```

如把例 2.4 的程序改写为例 2.5 的程序，则稍懂英文的人一看便知此程序的作用，当使用汉化 FORTRAN 版本时，变量名也可直接用汉字。

3. 增强程序的通用性

程序是用计算机语言描述的对数据进行处理的一套算法。原始数据往往在程序首部作为初值被赋给变量。如例 2.5 中的 PI 和 RADIUS。当这些变量赋予不同初值时，就表达了不同的实际问题。当 RADIUS 被赋值于 3.5 时，上述程序就变成了求半径是 3.5m 的圆周长、面积及球体积。

若用键盘输入语句来给变量赋初值，就可在程序的运行过程中，随心所欲地给变量赋以不同的值，根本不用修改源程序，使得程序的通用性大大增强。

4. 实现不同模块的数据传递

FORTRAN 语言允许进行模块化程序设计，整个程序可由一个主模块和许多子模块组成，当模块之间需要传递参数时，可借助于变量的虚实结合来实现。具体内容可参见 2.5 节。

以上是从程序员的角度介绍了程序中使用变量的好处。从系统角度看，在程序中允许使用变量，简化了程序员对计算机的使用。程序员只须将表达式的值存入某个变量，而不必考

应该指出仅有于内存中哪个存储单元、如何存放等问题。

2.1.4 标准函数的概念及使用

在工程实践及科学计算中经常要用到一些专门运算，例如求 sinX、cosX、lnX、e^X 等。这时可以通过将函数展开成 X 的幂级数，而根据一定的精度要求取幂级数的前 n 项来求其值，例如 sinX 可展开为

$$X-\frac{X^3}{3!}-\frac{X^5}{5!}-\frac{X^7}{7!}+\cdots+(-1)^{n-1}\frac{X^{2n-1}}{(2n-1)!}+\cdots$$

对于给定的 X 和精度要求 ε，只要保证 $\left|\frac{X^{2n-1}}{(2n-1)!}\right|<\varepsilon$ 就能满足精度要求。

据此不难编写出求 sinX 的程序。由于这些运算经常遇到，每次都让用户自己来编写相应程序会给用户带来不必要的麻烦。软件制造商们已将这些常用算法编成了一个个子程序，放在子程序库中，当用户要使用时，只需按规定格式直接调用即可。当对源程序进行编译连接时，编译连接软件们动将这些子程序连接入源程序中，生成可执行的机器语言。例如用户要求表达式 $\sin0.6+e^{2.7}$ 的值，在程序中只需写出如下语句：

```
A=SIN(0.6)+EXP(2.7)
```

则当计算机执行此语句时，就会调用标准函数 sinX 和 e^X，算出表达式 $\sin0.6+e^{2.7}$ 的值，并将其放入变量 A 中。

不同语言的适用范围不同，所提供的标准函数的侧重点也有所不同。FORTRAN 主要用于数值计算，因此用于数值计算的函数比较多，而 FOXBASE 适用于数据处理，因而数据处理的函数功能很强。

种编译系统所提供的标准函数的多少是不一样的、提供的标准函数越多，用户使用起来就会越方便。在用一种语言来编写一个实用程序时，应首先搞清楚系统提供了哪些标准函数，各个函数的功能和用法如何。

在编写程序时，应尽量使用系统所提供的标准函数，使用标准函数具有以下优点：

1）编写出的程序简洁易懂，效率高。如上例中若由用户自己编写程序求 sin(0.6)和 $e^{2.7}$ 则程序势必很长，而且普通用户编写的程序肯定不如掌握了大量程序设计技巧的专门软件编制者写出的程序效率高。使用标准函数写出的程序简明易懂，容易阅读。

2）编写出的程序通用性好、易于移植，因为大多数应用的标准函数，无论哪种语言、哪个编译版本都会提供，因而很容易移植，程序的通用性好。

在使用标准函数时应注意如下事项：

1）标准函数的书写格式要正确，例如 sin(0.6)中 0.6 必须用括号括起来。

2）注意自变量的取值及其范围。例如在 FORTRAN 中三角函数的自变量应以弧度为单位，LOG（X）中要求 X>0。

3）有些非标准函数可用标准函数来构造，例如系统只提供了正切 TAN（X），则 ctgX 可用三角函数关系 $\mathrm{ctg}X=\frac{1}{\mathrm{tg}X}$ 而求得。

2.1.5 表达式的用法

用运算符（算术、逻辑、字符等）将相应的量连接起来，所构成的符合 FORTRAN 规定的式子叫 FORTRAN 表达式。用算术运算符连接算术量所组成的式子叫算术表达式，用逻辑运算符连接逻辑量所构成的式子叫逻辑表达式，关系表达式是逻辑表达式的特例。有使用