

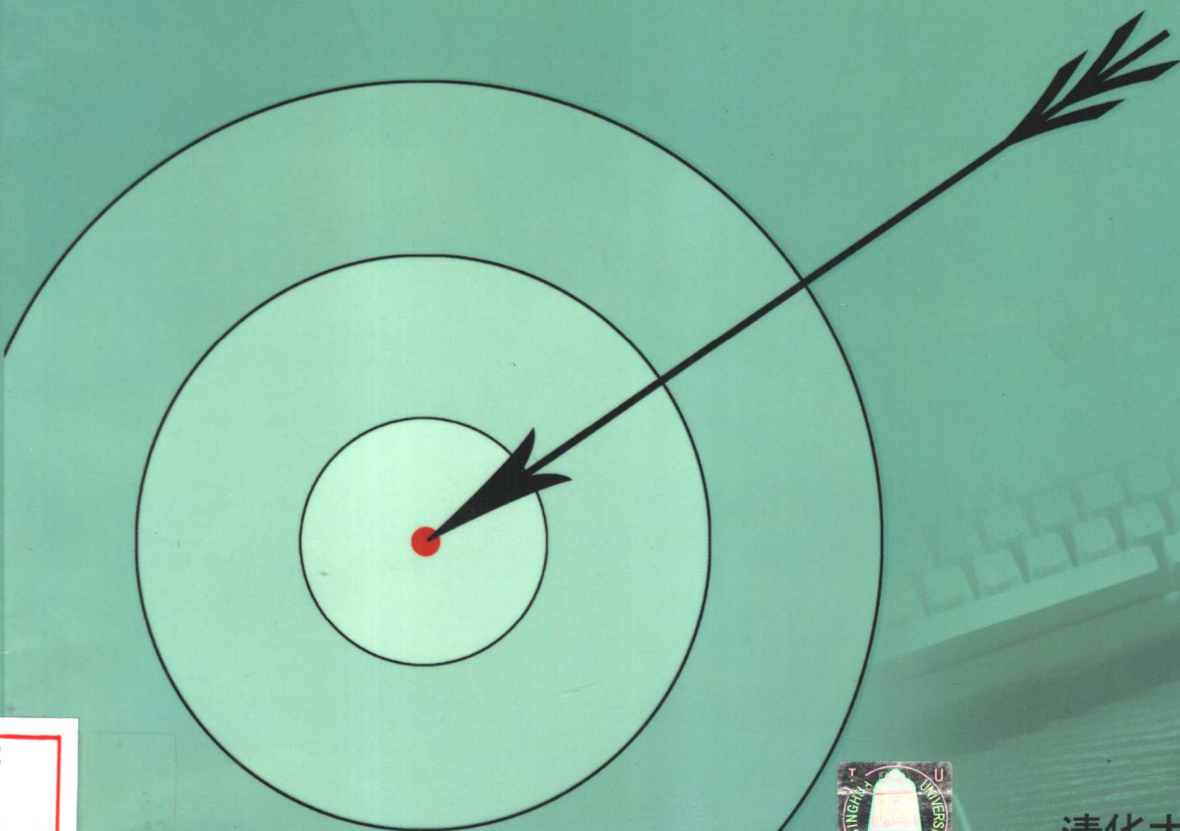


全国计算机等级考试全真训练丛书

全国计算机等级考试(二级 Fortran)

全真训练

黄 明 宋存利 时维国 编著



清华大学出版社



全国计算机等级考试全真训练丛书

全国计算机等级考试（二级 Fortran） 全真训练

黄 明 宋存利 时维国 编著

清 华 大 学 出 版 社

北 京

内 容 简 介

本书是根据 2002 年最新考试大纲,为参加全国计算机等级考试二级 Fortran 语言程序设计的考生编写的一本全真训练教材。

本书紧扣考试大纲,以考试要点为导向,以全真训练为中心,通过对考点的归纳分析和典型习题的强化训练,使考生较快地达到考试临战状态。

本书是针对报考全国等级考试二级 Fortran 语言程序设计的考生编写的,但由于本书包含很多典型习题的分析与训练内容,对现行相关计算机教材有较强的补充作用,所以本书也是一本很好的 Fortran 语言程序设计的辅助教材。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

全国计算机等级考试(二级 Fortran)全真训练./黄明,宋存利,时维国编著. —北京:清华大学出版社, 2003

(全国计算机等级考试全真训练丛书)

ISBN 7-302-06742-2

I. 全… II. ①黄… ②宋… ③时… III. ①电子计算机-水平考试-习题 ②Fortran 语言-程序设计-水平考试-习题 IV. TP3-44

中国版本图书馆 CIP 数据核字(2003)第 046114 号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

责任编辑: 刘利民

封面设计: 秦 铭

版式设计: 张红英

印 刷 者: 北京昌平环球印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印 张: 25.25 字 数: 576 千字

版 次: 2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

书 号: ISBN 7-302-06742-2/TP·5025

印 数: 1~5000

定 价: 29.80 元

丛书编写委员会

(排名不分先后)

主编：徐孝凯 王岳斌

策划：欧振旭 刘利民

编委：

徐孝凯	王岳斌	黄 明	吴艳辉	袁 慧	丁卓平
何光明	朱 嵬	严太山	李 毅	赵红梅	曾孝文
刘世峰	杨克昌	陶 睿	严权锋	刘胜钢	谭用秋
周细义	邵 静	刘生平	宋勇刚	成正祥	赵东霞

“全国计算机等级考试全真训练丛书”序

全国计算机等级考试是在计算机技术的飞速发展引发了新的工业浪潮和世界性的技术革命的大背景下产生的，它以普及和提高计算机应用水平为目的，旨在提高全民科学文化素质。自从计算机等级考试推出以来，已有上百万人参加了考试，它已成为全国范围内普及最广、参加人数最多的计算机考试，并有力地推动了计算机应用技术的发展。

全国计算机等级考试根据计算机应用水平的不同分为四个等级，分别为一、二、三、四级。人们可以根据自己的实际水平参加不同级别的考试。

为了帮助广大参加考试的人员顺利地通过计算机等级考试，并全面提高自己的计算机应用水平，清华大学出版社组织了一些高校计算机系的教师 and 该类考试辅导班的教师编写了“全国计算机等级考试全真训练丛书”。该丛书以教育部考试中心 2002 年对计算机等级考试所做的调整为依据，以此次调整后的新大纲为指导，精心策划而成，完全可以适应新形势下的计算机等级考试的要求。

丛书内容

□ 一级考试

《全国计算机等级考试一级全真训练》

《全国计算机等级考试一级 B（Windows 环境）全真训练》

□ 二级考试

《全国计算机等级考试（二级 C）全真训练》

《全国计算机等级考试（二级 QBasic）全真训练》

《全国计算机等级考试（二级 FoxBASE+）全真训练》

《全国计算机等级考试（二级 Fortran）全真训练》

《全国计算机等级考试（二级 Visual Basic）全真训练》

《全国计算机等级考试（二级 Visual FoxPro）全真训练》

□ 三级考试

《全国计算机等级考试（三级 PC 技术）全真训练》

《全国计算机等级考试（三级信息管理技术）全真训练》

《全国计算机等级考试（三级网络技术）全真训练》

《全国计算机等级考试（三级数据库技术）全真训练》

□ 四级考试

《全国计算机等级考试（四级）全真训练——计算机系统组成及工作原理》

《全国计算机等级考试（四级）全真训练——数据结构与算法》

- 《全国计算机等级考试（四级）全真训练——离散数学》
- 《全国计算机等级考试（四级）全真训练——操作系统》
- 《全国计算机等级考试（四级）全真训练——数据库》
- 《全国计算机等级考试（四级）全真训练——软件工程》
- 《全国计算机等级考试（四级）全真训练——计算机体系结构》
- 《全国计算机等级考试（四级）全真训练——计算机网络与通信》

丛书特色

- 充分考虑到考生在考前训练的需要，将典型考题和考试大纲进行剖析，将指定的考试内容进行精缩，用言简意赅的语言对每一个考试知识点进行总结。
- 在每一章中，以例题的形式对考试内容进行详细的讲解和分析，后面还附有大量的习题和习题参考答案，以帮助考生巩固所学的知识。
- 每一级别的考试都准备了几套模拟试题。
- 一级考试、二级考试（二级 Fortran 除外）和三级考试的书都配有上机模拟考试系统盘，其考试界面、题型和考试环境与真实考场基本相同。

读者对象

本系列丛书适用于参加全国计算机等级考试的各个级别的考生，也适用于高等院校的师生和计算机爱好者。

编委会

2003 年 8 月

前 言

全国计算机等级考试是面向社会的计算机应用能力水平考试，该项考试促进了计算机知识的普及和计算机技术的推广，在用人单位考核工作人员应用计算机水平等方面适应了社会的需要，在社会上享有良好的声誉。

为了帮助考生巩固所学知识，顺利通过考试，作者编写了本书。本书有如下特点：

1. 本书不是以系统讲解理论知识为重点，而是紧扣考试大纲，以常见考试要点为指导，重点突出对考生的解题训练。

2. 本书中作者根据多年教学和考试辅导经验，对重要的、容易出现的考点做了详细的归纳和分析。

3. 对每一类考点，都精心选择了典型的例题，并尽可能从不同的角度进行分析，以使读者触类旁通。

4. 每章都配有习题，这些习题有的是从以往的考题中筛选出来的，有的是在原来考题的基础上结合实际应用改编而来，这些习题涵盖了不同的知识点，具有很强的代表性。

5. 本书内容由浅入深，循序渐进；语言凝炼，通俗易懂；结构清晰，层次分明。

本书适用于准备参加全国计算机等级考试二级 Fortran 语言程序设计的考生，也可作为大专院校和培训班的教学参考书。

本书由大连铁道学院黄明、宋存利和时维国等共同编写。

读者在使用本书过程中如有疑难问题，可与作者 E-mail 联系，作者的电子邮箱是：
dlhm@263.net。

编 者
2003 年 8 月

目 录

第 1 章 Fortran 语言基础知识	1
1.1 考试要点	1
1.1.1 Fortran 语言介绍	1
1.1.2 Fortran 的数据类型及其运算	3
1.1.3 常用的 Fortran 内部函数	9
1.2 例题详析	10
1.2.1 选择题	10
1.2.2 填空题	16
1.3 习题	18
1.3.1 选择题	18
1.3.2 填空题	23
1.3.3 习题参考答案	23
第 2 章 Fortran 语言的程序结构	24
2.1 考试要点	24
2.1.1 顺序结构程序设计	24
2.1.2 Fortran 语言的选择结构	34
2.1.3 Fortran 语言的循环结构	39
2.2 例题详析	46
2.2.1 选择题	46
2.2.2 填空题	60
2.3 习题	77
2.3.1 选择题	77
2.3.2 填空题	98
2.3.3 习题参考答案	113
第 3 章 数组、函数及子程序	116
3.1 考试要点	116
3.1.1 数组	116
3.1.2 函数	125
3.1.3 子程序	128

3.1.4	参数传递.....	130
3.2	例题详析	134
3.2.1	选择题.....	134
3.2.2	填空题.....	155
3.3	习题	173
3.3.1	选择题.....	173
3.3.2	填空题.....	188
3.3.3	习题参考答案.....	204
第 4 章	Fortran 数据结构与算法.....	208
4.1	考试要点	208
4.1.1	Fortran 数据结构.....	208
4.1.2	字符型数据.....	212
4.1.3	算法.....	217
4.2	例题详析	229
4.2.1	选择题.....	229
4.2.2	填空题.....	234
4.3	习题	242
4.3.1	选择题.....	242
4.3.2	填空题.....	245
4.3.3	习题参考答案.....	249
第 5 章	数据联系和文件	250
5.1	考试要点	250
5.1.1	数据联系.....	250
5.1.2	文件.....	259
5.2	例 题 详 析	264
5.2.1	选择题.....	264
5.2.2	填空题.....	276
5.3	习题	286
5.3.1	选择题.....	286
5.3.2	填空题.....	296
5.3.3	习题参考答案.....	301
附录 A	模拟试卷	303
附录 B	历年试题	346
附录 C	全国计算机等级考试说明	388

第 1 章 Fortran 语言基础知识

大纲要求:

1. Fortran 程序的构成 (主程序和子程序);
2. Fortran 源程序的书写格式;
3. 数据类型及其相关运算。

1.1 考试要点

Fortran 是“Formula Translation”的缩写, 意思为“公式翻译”。它是为解决科学研究、工程问题或企事业管理中那些能够用数学公式表达的问题而设计的, 主要用于数值计算, 是被广泛使用的计算机高级语言之一。

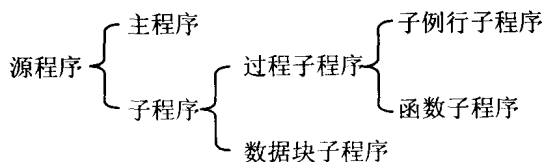
1.1.1 Fortran 语言介绍

1. Fortran 语言的发展概况

Fortran 语言是世界上第一个被正式推广使用的高级语言, 1954 年被提出来, 1956 年 1 月开始正式使用。1980 年, 国际标准化组织 ISO 正式接受 Fortran 77 为国际标准, Fortran 77 即《程序设计语言 Fortran ISO 1539—1980》。该标准分为子集和全集。1990 年, 国际标准化组织 ISO 公布了 Fortran 新的国际标准, 即 Fortran 90, 1991 年 5 月正式执行, ISO 编号为 ISO/IEC 1539—1991。新标准中增加了许多新的特性和功能, 如由我国提出的多字节字符集数据类型及相应的内部函数, 为非英语国家在使用 Fortran 方面提供了极大的方便。之后不久, 又出现了 Fortran 95。

2. Fortran 程序的结构

Fortran 源程序由一个主程序或者再加上一个或多个子程序组成, 其结构关系如下:



一个 Fortran 程序是由一个或多个程序单位组成的。主程序和每一个子程序分别是一个独立的程序单位。子程序有函数子程序、子例行子程序、数据块子程序 3 种，各个程序单位的顺序没有规定，一般把主程序放在最前面以便于阅读，而程序的执行总是从主程序的第一条可执行语句开始。一个程序单位又称为一个模块。

一个程序单位包括若干行，在 Fortran 语言中，行分为语句行和非语句行。语句行：由一条 Fortran 语句组成，它又分为执行语句和非执行语句。其中执行语句使计算机在程序运行时产生某些操作（如打印语句、赋值语句等）；非执行语句是将有关信息通知编译系统，以便程序在进行编译时作出相应的处理（如有说明语句、数据语句等）。非语句行（即注释行）：它不是 Fortran 语句，不被翻译成机器目标指令，不产生任何机器操作，而是为了方便人们阅读程序而加入的注释。它是程序的一部分，在打印程序清单时被原样打印。可用英文、汉语拼音或汉字作注释。注释行中即使有错误（如英文单词错误）也不影响程序运行。

Fortran 程序中，一行只能写一条语句而不能写多条语句，若一条语句太长而在一行写不下时，可以写在续行上（续行是在本行下面有“续行标志”的行）。

Fortran 程序中的语句可以有标号也可以没有标号，根据需要而定。标号只起标识的作用，它标志着被标号的语句将被其他语句引用。它不影响语句的执行顺序。Fortran 程序的执行顺序一般按照语句的物理顺序执行，即书写在前面的语句先执行，后面的语句后执行。在同一个程序单位中不允许有两个或两个以上具有相同标号的语句。

同一个程序单位中各类语句的位置是有规定的。如 PROGRAM 语句是主程序块的第一条语句，FUNCTION 语句是函数子程序的第一个语句，END 语句是程序块的最后一条语句。

Fortran 源程序要按照一定的格式书写。哪些内容应写在哪一行中的哪一列（或哪几列上）都有严格的规定。

3. Fortran 源程序的书写格式

Fortran 源程序必须严格地按照一定的格式书写。源程序需按行书写，每行有 80 列，共分为 4 个区：标号区、续行区、语句区和注释区。

（1）标号区（1~5 列）

标号可以是 1~99999 之间的任意无符号整数，也可以没有标号。标号区的空格不起作用，标号只是一种标记，标号数值的大小与它出现的顺序无关，不影响程序的执行顺序。

标号区内不得出现标号以外的内容，但注释行例外。第一列出现“C”或“*”字符，该行将按注释行处理。一行中的第一列只能出现数字、空格、“C”或“*”字符，否则，编译时按出错处理。

（2）续行区（第 6 列）

如果在一行中的第 6 列写一个非零或非空格的字符，则该行作为其上一行的续行，编译时把一行与它的多个续行连接在一起组成一个语句。Fortran 允许一个语句有 19 个续行，即一个语句最多可以写 20 行。注释行不能有续行。为了避免引起错觉，一般续行标志采用语句中通常不出现的字符（例如“\$”、“&”等）。

(3) 语句区 (7~72 列)

Fortran 语句只能写在每行的第 7~72 列之间, 如果不在这个范围, 该语句将不会得到准确编译。在一行中只能写一个语句, 除了引号内的字符串中的空格是有效的以外, 一般语句内的空格在编译时被忽略 (例如 $\text{SIN}(X)$ 与 $\text{SIN}(X)$ 是等效的)。

(4) 注释区 (73~80 列) 它不是语句的一部分, 在编译时对注释区 (73~80 列) 并不作处理。程序员一般利用这 8 列为程序编顺序号以便查找, 一般做法是用前三位记录页数, 后五位记录行数, 例如 00100010 表示程序的第 1 页第 10 行。但是在打印程序清单时按照原样打印, 注释区中即使有错误 (如英文单词错误) 也不影响程序的正常运行。

1.1.2 Fortran 的数据类型及其运算

1. 数据类型

(1) Fortran 77 提供的 6 种数据类型

4 种数值型数据:

① 整型 (INTEGER)

整型数也称整数, 它是一个正的或负的或零的整数, 由正负号和一串数字构成 (例如 -10、5 等)。

整数可包括数符 (正负号), 也可不包括数符 (此时数为正), 例如: +100 与 100 等价。程序中常数的各数字之间的空格不起作用, 如 123 和 1 2 3 等价。微机中一般用两个字节 (16 位) 来储存一个整数, 中型计算机中一般以 4 个字节 (32 位) 来存储一个整数, 都是用第一位来存放符号, 其余为数本身。

② 实型 (REAL)

实型数也称实数, 它有两种表示法: 小数形式和指数形式。

小数形式: 即日常习惯使用的小数, 如 +12.3、-68.56、3.75、0.0 等。它由一个或多个数字和一个小数点组成 (必须包含一个小数点而且只能有一个小数点)。小数点前或小数点后可以不出现数字, 但不能在小数点前后都不出现数字。如 +0.8、0.0、60.、.36、.0 都是合法的, 而 18、36、. (只有一个小数点而无数字) 是非法的。

指数形式: 在数学中常用指数形式表示一个大数或小数, 在 Fortran 中把诸如 0.91×10^{-30} 表示为 0.91E-30。可以看到, 用指数形式表示的实数由两部分组成, 即数字部分和指数部分。如上例中 0.91 为数字部分, E-31 为指数部分。数字部分可以是不带小数点的整数, 也可以是实数, 但指数只能是整数。

在指数表示法中, 由于计算机设备无法表示上下角标, 所以用 E 表示以 10 为底的指数。例如, 12.3×10^8 的 Fortran 表示为: 12.3E8 或 12.3E+8。

在计算机内存单元中, 实数的存储按二进制形式存放, 并且一律采用指数形式。存储形式如图 1-1 所示, 它由三部分组成: 数符、指数部分和数字部分, 数字部分最前面有一个隐含的小数点。

数符	指数部分	数字部分
+	0000101	111000000000000000000000

图 1-1

由于存储指数部分和数字部分的位（bit）长有限，因此计算机存储一个实数的有效数字和数的范围都是有限的。当一个实数的有效数字长度超过计算机所能存储的实数的有效数字长度时，它的最末几位数字就会被去掉，这样就造成实数的存储误差，这就是 $1/3.0 \times 3 - 1.0 \neq 0.0$ 和造成溢出的原因。

③ 双精度型（DOUBLE PRECISION）

双精度型数与实数的指数形式十分相似，书写格式惟一的区别是指数部分用 D 取代 E。一般地，如果一个计算机用 4 个字节表示一个实数，那么用 8 个字节表示一个双精度数。在同一台计算机系统中，它同实数的区别在于双精度型数表示的数的精度更高，表示数的范围比实型数更大。同时它也存在误差和溢出问题。

④ 复型（COMPLEX）

复型数据也称复数，它是由实部和虚部构成。在程序中的书写形式为由圆括号括起来的一对用逗号隔开的实数或整数组成。逗号左边的数是复数的实部，右边的是复数的虚部。例如，（4.0,8.2）与（4,8.2）都表示复数 $4.0+8.2i$ 。

在程序中必须对复型变量加以说明。在复型数据的实部和虚部都不是常量而是表达式时，必须用 CMPLX 函数将实部和虚部组成复型数据再赋给复型变量。例如， $C=CMPLX(3.0+A,6.0+B)$ 。如果所用的计算机系统用 4 个字节存放一个实数，则用 8 个字节存放一个复型数。

两种非数值型数据：

① 逻辑型（LOGICAL）

在 Fortran 中规定逻辑型量的值只有“真”和“假”两个，分别用.TRUE.和.FALSE.表示。在程序中必须对逻辑型变量加以说明。

② 字符型（CHARACTER）

字符型数也称字符串，它是由撇号括起来的字符序列组成。例如：'Fortran'、'STUDENT'、'学生'等。字符串中的字符可以是计算机系统中允许使用的任何字符。空格也是一个有效字符，字符串两侧的撇号是字符串的分界符号，不是字符串的一部分。字符串中如出现撇号，要用两个连续的撇号表示。例如：'THAT'S RIGHT!'。当字符以 ASCII 码存储时，两个字符数据比较的规律为：数字 0 小 9 大；数字比字母小；大写字母比小写字母小；字母 A 小 Z 大，a 小 z 大；空格字符最小；其他专有字符比数字小。

（2）常量

常量是指在程序运行期间其值始终保持不变的一些量，如-3.3、5.6、2.0、.FALSE.等都是常量。在 Fortran 语言中，允许使用的常量有 6 种类型：整型常量、实型常量、双精度型常量、复型常量、逻辑型常量、字符型常量。其中数值型常量包括：整型常量、实型常量、双精度型常量和复型常量；非数值型常量包括：逻辑型常量和字符型常量。

(3) 变量

变量是指在程序运行期间其值可以变化的量。计算机系统为每一个变量在内存中开辟一个存储单元，以用来存储变量的数值。

在 Fortran 77 中，允许使用的变量有 6 种类型：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。其中数值型变量包括：整型变量、实型变量、双精度型变量、复型变量；非数值型变量包括：逻辑型变量、字符型变量。

① 变量的命名规则

Fortran 中变量的命名规则是：第一个字符必须是字母，即变量名必须用字母开头，在第一个字母后面可以跟 1~5 个字母或数字。例如，A、X1、CHINA、G81 都是合法的，而 3X、X*Y、U.S.A 则是不合法的。

当变量名超过 6 个字符时，有些计算机系统按前 6 个字符有效，有些计算机系统则按出错进行处理，而有些计算机系统则允许使用多于 6 个字符的变量名。为使程序具有通用性，建议读者命名的变量名不超过 6 个字符。

变量名应尽量做到“见名知义”。例如，I 代表电流，R 代表电阻，U 代表电压，MAX 代表最大值，MIN 代表最小值。在变量名中，大写与小写是等价的，可互相代替。例如：ABC、Abc、abc 等价。

变量名的字符之间插入的空格不起作用。例如，ABC 与 A B C 代表同一个变量。

Fortran 没有规定保留字，可以用语句定义符或函数名作变量名。例如，SIN 是正弦函数，但可以有 SIN=3.5，此时 SIN 就是变量名。由于容易引起混淆，建议不要使用 Fortran 中已有特定含义的关键字作为变量名。

在同一个程序单位中，变量名、函数名或语句定义符不能同名。例如，下面的做法是错误的：

```
SIN=3.5
```

```
SUM=SIN*SIN (2.0)
```

② 变量类型的 3 种说明方法

隐含规则（又称 I-N 规则）：Fortran 中规定，在程序中的变量名，凡以字母 I、J、K、L、M、N 这 6 个字母开头的，即认为该变量为整型变量，以其他字母开头的变量为实型变量。例如，变量 IMAX、NUMBER、I、K1、JOB、MONT 等均为整型变量。而变量 SUM、ABC、YOUNG、A、AMONNT、B1 等均为实型变量。

用类型说明语句确定变量的类型：如果想改变“I-N 规则”对变量类型的约束，可以用类型说明语句专门指定某些变量的类型。例如，INTEGER A, TOTAL, SUM，这样变量 A、TOTAL、SUM 按“I-N 规则”原为实型变量，而用类型说明语句 INTEGER 语句说明就将变量 A、TOTAL、SUM 规定为整型变量了。同样也可将某些变量指定为实型、双精度类型等等。

Fortran 中规定了 6 个类型说明语句：INTEGER（整型说明语句）、REAL（实型说明语句）、DOUBLE PRECISION（双精度型说明语句）、COMPLEX（复型说明语句）、LOGICAL（逻辑型说明语句）、CHARACTER（字符型说明语句）。

IMPLICIT 语句（隐含说明语句）指定变量的类型：可以用 **IMPLICIT** 语句将以某一字母开头的全部变量指定为所需的类型。其一般格式为：**IMPLICIT** 类型说明符（字母表）[，类型说明符（字母表），...]，例如：

IMPLICIT INTEGER (A, B, T-V)

IMPLICIT REAL (I, K)

即指定以字母 **A** 和 **B** 开头的全部变量和由 **T** 到 **V** 开头的全部变量为整型。以字母 **I** 和 **K** 开头的全部变量为实型。

说明：

以上 3 种方式中，以类型说明语句最优先，**IMPLICIT** 语句次之，“**I-N** 规则”的级别最低。

例如，**IMPLICIT INTEGER (A, B)**

REAL ABC

若无此二语句，则变量 **ABC** 为实型，由于 **IMPLICIT** 语句指定以字母 **A, B** 开头的所有变量为整型，但语句 **REAL** 又指定 **ABC** 为实型，由于类型说明语句比 **IMPLICIT** 语句优先级别高，因此变量 **ABC** 最终在程序中应为实型。而其他以字母 **A, B** 开头的变量为整型。

类型说明语句和 **IMPLICIT** 语句是非执行语句，它的作用在于通知编译系统按规定的类型分配内存单元和确定数据的存放方式。

类型说明只在本程序单位内有效。假如主程序中用 **INTEGER** 语句指定 **X** 的类型为整型，子程序中的变量 **X** 则不受其约束，若无类型说明，则按“**I-N** 规则”应为实型。**IMPLICIT** 语句和类型说明语句应该出现在本程序单位中的所有执行语句的前面，其中，**IMPLICIT** 语句又应该在所有的类型说明语句的前面。系统为每一程序中的每一变量都开辟一个存储单元，用来存储变量的值，在每一瞬时，一个变量只能有一个确定的值。

2. 表达式

Fortran 表达式是一个或多个运算的组合，是由 **Fortran** 运算符和括号把 **Fortran** 操作数按一定规则连接起来的式子。**Fortran 77** 有 4 种表达式：算术表达式、关系表达式、逻辑表达式、字符表达式。

（1）算术表达式

算术表达式的一般形式：

一元运算：<算术运算符><算术量>。其中，算术运算符只能是“+”和“-”。例如，+**X**、-**Y**、+（**A**+（**B**））等。

二元运算：<算术量><算术运算符><算术量>。其中，算术量可以是一个数值常量、数值变量、数值函数或者算术表达式。算术表达式的值仍为算术量。

算术表达式有五种运算符：“+”、“-”、“*”、“/”和“**”分别表示加、减、乘、除和乘方。

算术表达式的运算优先次序为：

括号 () 函数 乘方 乘、除 加、减
 (高) —————> (低)

乘号用“*”表示，不能写成“×”，也不能用“.”代替乘号；除号不能用“÷”号，两个运算符不能紧邻，如 $A*-B$ 是不合法的，应写成 $A*(-B)$ ；只有“*”可以连写，代表乘方，例如， $B**3$ 表示 B^3 。

一个算术量进行一元运算“-”或“+”后，其结果的类型与原类型相同。例如， $-I$ 与 I 类型相同； $+ABC$ 与 ABC 类型相同。

相同类型的算术量之间进行算术运算，其结果保持原类型。例如， $1/2=0$ ， $1.0/2.0=0.5$ ，算术表达式 $1/2\text{SIN}(X)$ 转换为 Fortran 表达式时不能写成 $1/2*\text{SIN}(X)$ ，因为不管 X 取何值，此表达式的值总为 0，不能与算术表达式 $1/(2\text{SIN}(X))$ 的值等价，因此，应写成 $0.5*\text{SIN}(X)$ 或 $1.0/2.0*\text{SIN}(X)$ 。

不同类型的算术量之间进行算术运算，则编译系统自动将它们转换成同一类型然后进行算术运算，转换规律是将低级类型转换为高级类型，由低级到高级为：整型（1级）→实型（2级）→双精度型（3级）→复型（4级）。并且转换是从左到右，直到遇到不同类型的算术量才进行转换。例如， $1/2*2.0=0$ ，按算术表达式同级运算从左向右的运算顺序进行计算，所以此表达式首先计算 $1/2$ ，而 1 和 2 都为整型，因此其值为整型，所以 $1/2$ 得 0；再计算 $0*2.0$ ，此时是两个不同类型的数据进行运算，所以应该将低级类型的整数 0 转换为实型，即为 $0.0*2.0$ ，因此结果为实型的 0.0。

乘号不能省略。例如，对数学式 $3\text{COS}(X)$ 必须写成 $3*\text{COS}(X)$ ，而不能写成 $3\text{COS}(X)$ 。

Fortran 中无大、中、小括号之分，一律用小括号。例如， $\{[(X+4)^2+(A+B)^2]+c\}*6$ 相应的 Fortran 表达式应为： $((X+4)**2+(A+B)**2)+C)*6$ 。

Fortran 77 规定，对多次乘方运算按“先右后左”原则处理。例如， $4**3**2$ 应先计算 $3**2$ 得 9，再计算 $4**9$ 的运算，它相当于 $4**(3**2)$ 。

由于实型数在存储时，有效数字有可能被丢掉而产生误差，因此，在写表达式时，应尽量使每次运算结果都在有效位数范围之内，否则就会出现误差。例如， $1.0/3.0*3.0$ 结果就不是 1.0，而是 0.9999999。因此如果有两个理论上应相等的实数 A 和 B ，如果判断“ $A-B=0?$ ”，可能得到的结果不等于 0，即 $A \neq B$ 。所以在判断两个实数是否相等或不等时，最好改为判断“ $|A-B| \leq \varepsilon$ ”，其中 ε 为一个很小的数，如 10^{-6} 。如果小于或等于则认为 A 和 B 相等，否则为不等。

(2) 关系表达式

关系表达式的一般形式为：<算术量><关系运算符><算术量>。其中，算术量可以是一个数值常量、数值变量、数值函数、算术表达式。关系表达式的值为逻辑量，即为.TRUE. 或.FALSE.。

常用的关系运算符有：

.LT. 表示小于 .GE. 表示大于等于 .LE. 表示小于等于

.EQ. 表示等于 .GT. 表示大于 .NE. 表示不等于

例如，

数学上的关系表达式

对应的 Fortran 中的关系表达式

$X \leq 3.0$

X.LE.3.0

$(X-Y) \neq (A+B)$

(X-Y).NE.(A+B)

$X+Y+Z=0$

X+Y+Z.EQ.0

在一个关系表达式中可以有算术运算符和关系运算符，它们运算的次序是先进行算术运算再进行关系运算。

关系运算符的字母的两侧各有一个句点，不能省略，如“2.GT.3”不能写成“2GT3”。

两个字符串进行比较时，如果不等长，则系统会自动给较短的字符串的尾部补以空格，然后将两个字符串中的字符自左向右进行比较，如果所有字符都相等，则认为两个字符串相等，如果不相等，则以最左面的第一对不同字符的比较结果为准。例如，'ABCDE'<'ABCED'，前3个字符相同，第4个字符'D'<'E'，故 ABCDE'<'ABCED'为真。

关系表达式的值是一个逻辑量，当关系成立时为“真”，当关系不成立时为“假”。

关系表达式可以将不同类型的两个数值量进行比较，系统会将它们由低级的类型向较高的类型转换为统一类型后再进行比较。例如，A.GT.3，先将整数3转换为实数3.0，再与实型量A比较。

由于实数的存储和运算存在误差，在用“.EQ.”“.NE.”时要特别注意。对于A.EQ.B，为避免可能产生以上问题，可以改用ABS(A-B).LT.1E-6表示。对于C.NE.D，为避免可能产生以上问题，可以改用ABS(C-D).GT.1E-6表示。关系表达式中一个算术量的两侧不能同时出现关系运算符；逻辑型数据不能进行关系比较。例如，A.LT.B.LT.C是不对的。

(3) 逻辑表达式

逻辑表达式的一般形式为：<逻辑量><逻辑运算符><逻辑量>

常用的逻辑运算符有：

.NOT. 逻辑非 .EQV. 逻辑等 .AND. 逻辑与

.NEQV. 逻辑不等 .OR. 逻辑或

逻辑运算符的运算规则如表 1-1 所示。

表 1-1 逻辑运算符的含义

逻辑运算符	逻辑含义	逻辑运算举例	例子含义
.AND.	逻辑与	A.AND.B	A, B 都为真，则 A.AND.B 为真
.OR.	逻辑或	A.OR.B	A 或 B 之一为真，则 A.OR.B 为真
.NOT.	逻辑非	.NOT.A	A 为真，.NOT.A 则为假
.EQV.	逻辑等	A.EQV.B	A 和 B 值为同一逻辑量时，A.EQV.B 为真
.NEQV.	逻辑不等	A.NEQV.B	A 和 B 值为不同一逻辑量时，A.NEQV.B 为真

逻辑表达式中不仅可以有逻辑运算符，还可有关系运算符和算术运算符，最终运算结