FORTRAN 77

结构化程序设计

孙家启　吴国凤　编著

兵器工业出版社

FORTRAN77结构化程序设计

孙家启　吴国凤　编著

胡正家　主审

兵器工业出版社

（京）新登字049号

内 容 简 介

本书按照国家教委1993年颁布的高级语言程序设计课程教学基本要求，根据国家标准GB3057—82《程序设计语言FORTRAN 77》详细讲述FORTRAN77语言的构成，设计方法与技巧。

全书共九章：第一章概述，第二章FORTRAN77语言基础，第三章有格式输入/输出，第四章程序的控制结构，第五章数组，第六章函数与子程序，第七章数据的公用结合，第八章字符处理，第九章数据文件。书后附有上机指南等。

本书可作为高等院校非计算机专业的教材，也可供使用计算机技术人员阅读、参考。

图书在版编目（CIP）数据

FORTRAN77结构化程序设计／孙家启，吴国凤编著。
—北京：兵器工业出版社，1994.9
ISBN 7-80038-802-6
Ⅰ.F…
Ⅱ.①孙…②吴…
Ⅲ.FORTRAN语言-结构化-程序设计
Ⅳ.TP312F0
中国版本图书馆CIP数据核字（94）第03941号

兵器工业出版社出版发行
（北京市海淀区车道沟10号）
各地新华书店经销
北京百善印刷厂印装
*
开本：787×1092　1/16　印张：16　字数：388 千字
1994年9月第1版　1994年9月第1次印刷
印数：0-3000　定价：13.80元

前　言

FORTRAN 77是国际上FORTRAN新版本。FORTRAN 77语言丰富、功能较强，是当今世界广泛应用于科技与工程计算的一种高级语言。

本书根据国家标准GB3057—82《程序设计语言FORTRAN》为背景，详细介绍FORTRAN77语言及其程序设计方法。为适应高校各专业的教学要求，本书以国家教委1993年颁布的高级语言程序设计课程教学基本要求和安徽高校计算机水平考试大纲为依据，结合算法、数据结构、计算机语言、结构化程序设计等新思想和新方法，介绍各种程序设计方法和技巧，以便使读者能运用这一思想、方法和基础知识解决高等院校学生各自专业中的有关问题。

本书编写过程中力求做到概念清楚、深入浅出、突出难点，同时指出初学者在编程中容易出错之处。书中列举的程序均在机器上运行过，可供读者在学习时参考。

本书第一、六、七、八、九章由孙家启编写；第二、三、四、五章由吴国凤编写。在本书编写过程中，我们得到了机械工业部高校处，机械部高校计算机基础教育研究会许多专家的关心和帮助，特别是合肥工业大学教材科长郑象鸽工程师全力支持该书出版；国家教委全国高等学校工科计算机基础课程指导委员会主任委员，西安交通大学胡正家教授对本书进行了主审，并提出了有益的建议和修改意见，在此我们表示衷心的感谢。

由于水平所限，书中不妥和错误之处在所难免，恳请专家、读者批评、指正。

编者

1994年1月

目　录

第一章 概 论

§1-1 FORTRAN语言简介

FORTRAN语言是世界上第一个被正式推广使用的高级语言。它是1954年被提出来的，1956年开始正式使用，至今已有30多年历史，但仍历久不衰、它始终是数值计算领域所使用的主要语言。

FORTRAN是Formula Translation的缩写，意为“公式翻译”。它是为科学、工程问题或企事业管理中的那些能够用数学公式表达的问题而设计的，其数值计算的功能较强。

FORTRAN语言问世以来，根据需要几经发展，先后推出了不同的版本，其中最流行的是1958年出现的FORTRAN Ⅱ和1962年出现的FORTRAN Ⅳ。1966年美国标准化协会（American National Standard Institute，简称ANSI）公布了两个美国标准文本：

（1）标准FORTRAN（X3.9—1966），大致相当于FORTRAN Ⅳ。

（2）标准基本FORTRAN（X3.10—1966），大致相当于FORTRAN Ⅱ。

1972年国际标准化组织（International Standard Organization，简称ISO）接受了美国标准，在稍加修改后公布了ISO FORTRAN标准，即《程序设计语言FORTRAN ISO 1539—1972》，它分为三级，即：

（1）完全的（一级）FORTRAN，相当于FORTRAN Ⅳ。

（2）中间的（二级）FORTRAN，介于FORTRAN Ⅱ和FORTRAN Ⅳ之间。

（3）基本的（三级）FORTRAN，相当于FORTRAN Ⅱ。

FORTRAN Ⅳ（即FORTRAN 66）流行了十几年，几乎统治了所有的数值计算领域，许多应用程序和程序库都是用FORTRAN Ⅳ语言编写的。在结构化程序设计方法提出以后，人们开始感到FORTRAN Ⅳ已不能满足要求。FORTRAN Ⅳ不是结构化的语言，没有直接实现三种基本结构的语句，在程序中往往需要用一些GOTO语句以实现特定的算法。不少计算机厂商对FORTRAN Ⅳ进行了扩充。美国标准化协会（ANSI）在1976年对ANSI FORTRAN（X3.9—1966）进行了修订，基本上把各厂商扩充的行之有效的功能都吸收进去，同时又增加了不少新的内容，原预定在1977年通过，为了区别于FORTRAN66，新标准定名为FORTRAN77。实际上到1978年4月才由ANSI正式公布作为新的美国国家标准。即FORTRAN（X3.9—1978）同时宣布撤销FORTRAN(X3.10—1966)。FORTRAN77和FORTRAN66基本上是兼容的，也就是说，用FORTRAN66即FORTRAN Ⅳ编写的程序基本上不用修改（或作很少的修改）即可用FORTRAN77编译程序进行编译并运行，这是考虑到FORTRAN语言的继承性，不致使目前大量流行的FORTRAN Ⅳ程序作废而造成极大浪费。1980年，FORTRAN77被接受为国际标准，即《程序设计语言FORTRAN ISO 1539—1980》，该标准分为全集和子集。

FORTRAN77还不是完全结构化的语言，但由于增加了一些结构化的语句（特别是“块IF”语句，提供了IF—ELSE—END IF形式的判断控制语句），使FORTRAN扩充了字符

处理功能，使FORTRAN不仅可用于数值计算领域，还可以适用于非数值运算领域。

目前，FORTRAN77已在国内外广泛使用，大多数计算机系统都已配置了FORTRAN77编译程序，FORTRANⅣ已开始被淘汰。为了适应结构化程序设计的要求，应该学习和掌握FORTRAN77语言。

我国制订的FORTRAN标准，基本采用了国际标准，于1983年5月公布执行，标准号为GB3057-82。

目前正在酝酿国际新的FORTRAN标准，功能将有更大的扩充，成为能适应现代程序设计思想的现代程序设计语言。人们期待已久的FORTRAN 8X终于以FORTRAN90的形式问世，它的美国国家标准文件号是ANSI X3.198—199X，国防标准文件号是ISO/IEC 1539—1991，通俗的称呼是FORTRAN90。但它得到广泛推广使用恐怕还需要一些时日。因此，近几年内，FORTRAN77将是FORTRAN语言的主流。在学习和掌握了FORTRAN77的基础上进一步学习FORTRAN90新标准不会是很困难的。

目前，大多数大、中型计算机系统配置的是FORTRAN77全集，微型计算机多配置FORTRAN77子集。本书是以FORTRAN77全集为基础进行叙述的。如果读者所用计算机系统上配备的是FORTRAN77子集，则对书中某些程序需作一些必要的修改，才能在该计算机上进行，这点请读者注意。

§1-2 FORTRAN 77字符集

计算机语言不是任何一种外文字母或数字符号都能被接受的。例如，希腊字母α、β、γ……，数学符号π、$\sum$、$\int$，$\frac{dy}{dt}$……，还有罗马数字Ⅰ、Ⅱ、Ⅲ、……，这些字母及符号一般计算机语言是不能接受的，因此每一种计算机语言分别规定了它允许使用的字符。

FORTRAN77允许使用下列三种字符。

1．英文字母（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

IBM FORTRAN及SIEMENS 7570-C FORTRAN等编译系统允许使用小写字母，且视大、小写为等效。

2．数字（10个）

0 1 2 3 4 5 6 7 8 9

3．专用字符（13个）

␣ 空格（空白），书中用␣表示空格；

= 赋值号，用于赋值、循环等语句中；

+ 加（正）号；

− 减（负）号；

* 星号，作乘法、乘幂运算符；

/ 斜线，作除法运算或字段分隔符；

(左括号；

)　　右括号；

,　　逗号，作变量、数据、字段分隔符；

.　　圆点，作小数点用；

$　　货币符（美元号）；

'　　撇号，作字符常数定界符及撇号编辑用；

:　　冒号，作数组、字符子串上、下界分隔符和冒号编辑用。

另外，不同的计算机系统还会规定个别其他符号，请注意参阅有关资料。

在FORTRAN 77 源程序中，除一些特殊规定的场合外，一般只使用这49个字符，而且不区分字母的大小写，空格符也是无意义的，但能改善源程序的可读性。

当空格和字母大小写均有意义时，一般用于：字符型常数、H与撇号编辑描述符中或注解行中的空格与字母。而且在这些场合中使用的字符可以超出基本字符集的范围，当然以处理系统能接受为限。

此外，数字串作数值解释时应理解为十进制数。

§1-3　FORTRAN77源程序的书写格式

FORTRAN77的程序是模块结构，它是由1至多个程序单位组成。程序单位是由语句序列和任选的注解行序列组成。在FORTRAN77中，程序单位分主程序单位、函数子程序单位、子例程子程序单位、数据块子程序单位四种。一个可执行的FORTRAN77程序，有且仅个主程序单位，但可以有0到多个子程序单位。

FORTRAN77语言对源程序的书写格式有严格的要求。FORTRAN77语言规定一行只能写1个语句，每行有80列，每列写一个字符。一行中各列从左至右依次连续编号为1到80列。这80列又分成四个区，分别书写不同的内容，见表1-1。

表1-1　FORTRAN程序纸

1 5	6	7 10 15 20 25 30 35 40 45 50 55 60 65 70 72	73 75 80
C		EXAMPLE	
		READ(*,*)A,B	
		C=SIN(A)+LOG10(ABS(B))	
		D=EXP(A)+TAN(30.0*3.14159/180.0)	
		E=C/D	
		WRITE(*,10) E	
10		FORMAT(1X,7.3)	
		END	

1．标号区

1～5列属于标号区。语句标号主要用来区分不同的语句，标号可作为语句引用的标志，或作为控制语句转向的目标。因此，格式说明语句和要被控制语句转移到达的语句必须标有标号。如表1-1中第7行的FORMAT语句，它就是一个格式说明语句，它前面必须标有标号。

说明：（1）语句标号取值范围是1～99999的无符号整数。0不能作为语句标号。

（2）不满5位的标号在1～5列的任何地方出现，作用都相同。

（3）同一程序单位内不得有重复的语句标号。

（4）语句标号的前后或中间的空格无意义。

例如：100␣␣，␣100␣，␣10␣0，1␣0␣0作用完全相同。

2．续行区

仅指第6列。FORTRAN规定当一个语句在一行内写不完，或者为了整齐起见可在本行内任意处中断而在下一行接着写，此时，必须在下一行续行区（第6列）上写一个非零或非空格的任一字符。这时则表明该行是上一行的继续行。

说明：（1）一个语句最多可以有19个继续行，即不得多于1320个字符。

（2）续行不是一个独立的语句，它的标号区应该是空的。

3．语句区

指第7列至第72列。每一个语句可以从一行的第7列及其以后任一列开始书写。一般不要求一定从第7列开始写语句，有时为了避免出错，常常有意从第8列或第9列开始写语句，这样与续行标志位之间有一定间隔，使续行标志比较醒目。

4．注释区

指第73列至第80列。此区是供程序员作标记或注释用的，不能写程序。注释区一般是空的，必要时，写上注释内容。该区内容不送入计算机，编译程序对这部分内容也不处理。

除上述四个区外，当任意行的第1列写有字母C或字符＊时，则表示该行是注释行。注释行的作用是对源程序作某些说明、注解，以便增强源程序的可读性。注释内容一般是文字，若一行写不下，可以在下一行继续注释，注释的续行仍在第1列用字符C或＊作为标志。注释可随程序送入机器，机器不编译注释行，只在列程序清单时原样打印出来。例如表1-1中的第一行的内容为注释行。

每个程序单位以END语句作为结束标志。现对END语句说明如下：

结束语句一般形式为：END

功能：标志一个程序单位的结束。主程序单位中的END语句，表示程序执行到END语句时，就结束主程序单位的运行，这时本语句和STOP语句起相同的作用。在函数子程序或子例行子程序单位中，若执行到END语句，则其作用相当于执行RETURN语句。

END语句是一个可执行语句。END只能写在起始行的第7到第72列上。

FORTRAN77程序的每个程序单位的最后一行必须是END语句，它表示一个程序单位的结束。但一个程序单位中不允许有2个或2个以上的END语句。

§1-4　计算机算法

一、定义

在计算机学科中，我们把解决某一类特定问题的方法称为“算法”。所谓算法就是解决问题的一步一步的过程，也就是解决问题的逻辑步骤。数学解题步骤是数学上的各种算法。计算方法就是指用计算机解决各种数学问题的算法，现代计算机已远远地突破了数值计算的范围，它包括大量的非数值计算，如检索信息，表格处理，判断和决策，逻辑演绎等。因此，算法不仅仅是计算方法，它还应确定解决问题的各个逻辑步骤的过程、规则和方法。它和程序的结构是不可分割的。算法的研究是计算机科学的核心，而算法概念本身也是计算机

程序设计中最基本的概念之一。

正确的算法必须满足下列三个条件：

（1）每个逻辑块（模块）必须由可以实现的语句来完成。

（2）每个模块间的关系应该是唯一的（即块与块的关系一定要确定）。

（3）算法要能终止（不能造成死循环）。

下列过程就不是一个正确的算法。计算过程：

第1步：令N等于0

第2步：N加1

第3步：转向第2步

如果你用计算机执行此过程，从理论上讲，计算机将永远执行下去（死循环）。

而下列过程就是一个正确的算法，因为它能结束。进行一百万计数的算法。

第1步：设N等于0

第2步：N加1

第3步：如果N小于1000000，则转向第2步，否则停止。

这里我们不要把“计算方法”（Computational method）和“算法”（Algorithm）这两个词混淆。前者指的是求数值解的近似方法。后者是指解决问题的一步一步过程。

二、算法简例

例1 求$1\times2\times3\times\cdots\times20$

此例如果用原始的方法：$1\times2\Rightarrow2$，$2\times3\Rightarrow6$，$6\times4\Rightarrow24$，…，一步一步去写，算法虽然是正确的，但太繁琐，它要写19个步骤才能完成，因而这种方法不可取，在实际使用中，一般对这种例题，采用一种通用的方法表示。

设置两个变量：一个变量代表被乘数，一个变量代表乘数。然后再把每次乘积的结果放在被乘数变量中，并用用一种循环方式来表示算法、设T为被乘数，I为乘数，算法如下：

（1）$1\Rightarrow T$

（2）$2\Rightarrow I$

（3）$T\times I\Rightarrow T$

（4）$I+1\Rightarrow I$

（5）若$I\leqslant20$，返回（3）。否则，结束。

在这个算法中（3）到（5）组成一个循环，在实现算法中反复多次执行（3），（4），（5）步骤，直到I值大于20为止，算法结束，最后一次变量T的值就是所求的结果。可以看出这种方法表示的算法具有通用性、灵活性。

例2 求二个正整数M和N的最大公因子，

算法如下：

（1）输入M，N的值。

（2）用求余至数MOD，求M除以N的余数，然后把余数赋给变量R，即MOD（M，N）$\Rightarrow$R。

（3）判断R是否为0。

（4）如果$R\neq0$，则把$N\Rightarrow M$，$R\Rightarrow N$，返回（2）。

（5）若$R=0$，则输出最大公因子，结束。

此种算法表示的是求二个正整数M和N最大因子的欧几里德算法。

三、算法的特征

一般，计算机的算法应具有以下五个特征：

1．有穷性

一个算法必须在有限次执行后完成。

上面的实例表明，当执行到第三轮R＝0时，求得28和16的最大公因子为4时算法结束。显然它是满足有穷性的条件的。这是因为在第（1）行输入正整数M和N以后，接着按第(2)行执行，在第（2）行后，R的值一定小于N。当R≠0时，下一轮再执行第（2）行时，N的值将减少。而正整数的递减序列必然导致最后的终止。当然，选择较大的M和N值可能增加每一步骤的执行次数。

2．确定性

一个算法中的每一个步骤必须有明确的定义。计算机和自然语言大不相同，一切都要在程序中予以安排，不能有语意不明的地方。

这就是说，要执行的动作应该是非常明确的。它没有语义上解释的困难。诸如执行“X加6或者加7赋给变量Y”这类的步骤是含糊不清的。因为它没有明确X应该加6或7中的那一个数，在算法中类似这类的含糊不清的步骤是不允许的。在上面的实例中，第（1）行到第（5）行所表明的各个步骤其含义都是十分明确的。

3．输入

算法总是要施加到运算对象上，输入刻划了运算对象的初始情况，输入是算法的起点，所以一个算法应有0个或多个输入。

在上面的实例中应有二个输入数据，它们是正整数M和N。

4．输出

一个算法要有一个或多个输出。在例2中，它有一个输出，就是最后一轮在第（3）行中判别R＝0后输出的结果N。输出是与输入有特定关系的量。

5．可行性

一个算法的可行性是指所有待实现的运算必须是相当基本的，至少在原则上人们可以用纸和笔做有限次即可完成的。

在例2中，只有两个输入的正整数和输出这二个正整数的最大公因子的操作。另外在第（2）行到第（4）行所示的一些步骤中用到了二个整数的除法运算，判别一个整数是否为零以及一个变量赋值给另一个变量的运算。这些运算都是一些基本的可行的运算。

算法中实质上反映的是解决问题的思路。许多问题，只要仔细分析对象数据，就容易找到处理方法，对于定义明确的任务往往是这样。

一个算法可以用自然语言来表示，也可以用流程图来表示（见下节），还可以用介于自然语言和高级语言之间的一种称为伪代码（Pseudo code）来表示。用计算机语言（包括低级语言和高级语言）写的程序也是算法的表示形式。

§1-5　程序流程图

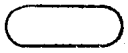
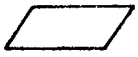
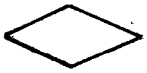
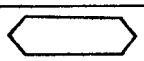
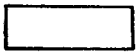
程序流程图由一些特定的图形符号组成，图形框内可以填写表示完成某种处理的字句。

程序流程图习惯上称为框图。通常，在正式编写源程序之前，先列出问题求解的步骤，再根据逻辑关系画出表示程序执行顺序的流程图。若求解一个复杂问题，画出其程序流程图，要经历构思、画粗框和细化等过程。一般细化到能清楚地表达程序的各处理步骤和它们之间的逻辑关系为好，以便按照流程图可以顺利地写出程序。

用程序流程图来形象地描述解决问题的程序执行顺序，思路清楚、简洁方便、使人一目了然。它不仅可以指导编写程序，而且在调试程序时，对照流程图很容易查错。程序流程图也是程序说明书的一部分，以便帮助别人理解程序员编写程序的思路和结构。为此，建议读者应当养成先画流程图、弄清程序执行的流向再编写程序的良好习惯。

最常用的流程图的图形符号，见表1-2所示。

表1-2 常用流程图图形符号

图形符号	名称	意义
	端点框	表示流程的起始、结束、暂停等
	输入或输出	表示输入或输出。提供处理所需的信息(输入)，或提供处理后的信息(输出)
	判断框	表示判断。经判断后，在几个可供选择的路径中选择其中一个路径
	准备	对一个程序作初始准备
	过程调用(既定处理)	表示调用一个已命名的过程。该过程在别处已详细定义
	处理框	表示一般的处理操作。例如，赋值等。
	流向线	表示流程的去向。连接图形符号
	连接点	表示流程图中流向线的连接点

流程图图形符号的使用说明：

(1)流程的一般方向是从左到右、自上而下。在流向线的末端可以加上箭头指示流程方向。

(2)2根或2根以上的流线可以汇集成一条流线。

(3)图形符号的大小、比例要适当。

(4)图形符号内的文字说明，要求按从左到右、自上而下的方式书写，力求简洁明了。

(5)连接符号的圈内标上字符。字符相同者，表示该流向线是相接的，否则表明不是同一根流向线。

(6)判断框可以有2个或2个以上的可供选择的路径，各路径应加以标识。

可以用流程图表示上一节中例1和例2的算法，见图1-1和图1-2所示。

流程图能形象地表示一个算法，清楚地表示出各个步骤间的先后次序。但画这种流程图比较麻烦，而且占篇幅。尤其是处理一个复杂的问题时尤为突出。但目前这种流程图已被广泛使用，因此学习计算机课程的读者仍应能看懂和使用它。

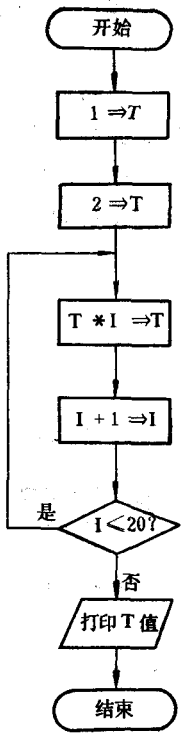


图 1-1

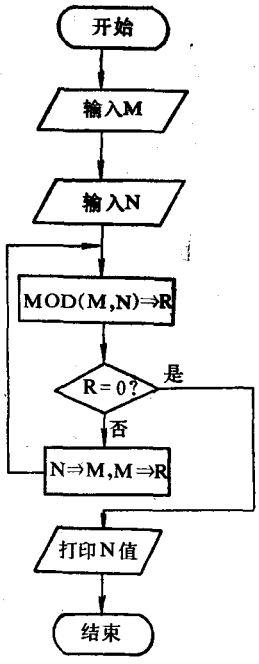


图 1-2

§1-6 FORTRAN77 程序设计步骤和上机过程

一、FORTRAN77程序设计步骤

1．确定问题的需求

人们在用计算机解决实际问题之前，首先要确定该问题究竟要计算机做些什么，即明确待编制的程序要包含哪些功能，以及希望程序得出什么结果等。这一步骤的基本任务是正确理解问题的要求。

2．分析设计

明确了程序该做什么后，接着就是确定程序怎么做。分析设计通常包含选择或建立数学模型，确定算法等工作。算法描述的是计算机解题的基本思想和基本步骤。

为了解决一个复杂问题，人的智力往往不可能一下子就触及到问题的细节方法。在分析了问题的需求之后，首先设计出一个抽象算法。在抽象数据上实施一系列抽象的操作。这些抽象的数据和操作只反映问题的本质属性。然后，作为下一步，再考虑这些抽象数据和操作的具体实现。这里采用的方法是反复地将大问题分解成相对独立的一些子问题，直至子问题足够简单为止。因子问题分别只涉及局部的环境和条件，便于各个击破。采用分而治之方法圆满地解决复杂的问题。习惯上称这种方法为“自顶向下”的程序设计方法。它的宗旨是

“分解”，它意味着从抽象到具体的逐步展开。

与“自顶向下”方法相反，“自底向上”的程序设计方法是先解决问题的各个不同部分，然后再将这些部分结合起来构成完整的算法或程序。这称为“综合”方法。

3．画程序流程图

流程图是算法的图形描述。由于流程图往往比程序更加直观清晰，容易被人看懂，所以它不仅可以作为编制程序的依据，而且也是相互交流的重要工具。

4．编程序

根据流程图描述的算法，用计算机可以接受的程序语言恰当地描述出来，这项工作就叫做编程序。

值得指出的是，程序最终是在计算机上运行的，但因交流和程序维护等需要，人们也要反复阅读程序。为此，提倡良好的程序设计风格。编写的程序要层次清楚，适当地插入注释以说明程序语句的功能和程序中使用的变量的意义。

5．调试程序

为了发现并纠正程序中可能包含的错误，必须要经过调试。只有上机通过的程序才是可交付使用的程序。一般说来，上述工作步步做得认真仔细，那么调试阶段暴露的程序错误和缺陷相对少些。程序可能有的错误大致可分为三类：

（1）语法性错误。对于语法性错误比较容易解决，源程序在编译过程中有错误信息提示，指出错误的类别和错误发生的位置，能帮助我们纠正语法性错误。

（2）逻辑性错误。对于逻辑性错误，我们应该精心组织测试用的输入数据，执行程序之后，看其是否能获得预期的结果。如果有错，采用各种程序测试技术确定错误位置和原因，以纠正错误。

（3）程序功能性错误。对于程序功能性错误，有时是由逻辑性错误引起的，随着逻辑性错误得到纠正，功能性错误也得到了纠正。一般来说，程序功能性问题是未能正确理解问题的需求所致，为纠正功能性问题可能要修正程序的结构，因此第一步弄清问题的需求是非常重要的。

当调试成功后，建议读者做第六步整理资料的工作。

6．整理资料

整理资料就是要写出程序说明书。它包括：程序名称、任务的具体要求、给定的原始数据范围、算法、程序流程图、程序清单、测试结果及修正记录、具体操作说明、程序运行的环境（对软件和硬件的配置要求）等内容，以便作为资料交流或保存。

二、FORTRAN77程序的上机过程

（1）编辑。通过编辑程序将编写好的FORTRAN77程序送入计算机，并建立一个存放该源程序的文件。

（2）编译。让编译程序对FORTRAN77的源程序进行编译，产生目标程序文件。如程序有语法性错误，编译程序会指出错误的性质和位置。纠正错误后须再编译，直至经过编译产生正确的目标程序为止。

（3）连接。将所有程序单位的目标程序及标准程序连接成一个可执行的文件。

（4）运行。执行经连接后产生的可执行文件。

目前在一些机型上对编译、连接和运行这三个操作，不再分别使用三个命令，而用一个

命令来完成上述三步的工作。详细过程，请查阅具体机器的操作手册。

§1-7 学习FORTRAN77语言目的与方法

FORTRAN77语言是当今世界上科技计算中最广泛使用的一种程序设计语言。它以其独特的块状结构和很强的科学计算能力，为我们提供了良好的使用计算机的工具。

FORTRAN77语言十分丰富，逻辑严格，功能较强，适用于机械学、电工学、控制论、水利学、结构力学、水资源分析、预测规划等后继课程的课程设计与毕业设计。本课程要求能够准确理解FORTRAN77语言的词法与句法的基本概念，掌握结构程序设计的方法与技巧，能读懂适当规模的程序，并能利用FORTRAN77语言编写的程序解决一些实际问题，增强分析问题和解决问题的能力，为将来应用计算机奠定牢固的基础。

要想学好这门课，难度较大。因此，要求学生必须认真对待听课、阅读教材与做作业、上机实习、复习考试等四个重要教学环节，力求做到：

（1）通过与数学上名词对比的方法，来理解FORTRAN77的常数、变量、内部函数、数组元素及表达式等词的词法规定；

（2）正确理解各语句的含义及其在程序设计中的作用。在此基础上要记住各语句使用的英语单词及句法规定，严格、准确、仔细地做好每一次作业，做到理解、记忆与实践相结合；

（3）广泛阅读例题，勤于思考，看懂每个程序的设计方法与执行步骤。运用别人的经验来指导自己编程，做到读程序例题与编程相结合；

（4）编制程序是一种创造性的工作，方法宜由粗到细、由简到繁。通常，应根据算法先考虑提供数据、计算赋值、输出结果三部分语句，然后再做修饰、优化工作。要仔细审查所编制的程序能否执行预定的计算任务，做到编写与检查、修改程序相结合；

（5）要重视上机实习、珍惜上机机时。通过上机操作，进一步锻炼自己查错的能力，学到修改程序的本领。

从编写程序、上机调试到算出正确结果是一个艰苦的学习过程。如果能经常注意改进学习方法，不断总结编写程序和审查程序的经验，增强学习的自觉性，提高学习FORTRAN语言的兴趣，那么每个人的程序设计思想就能较快地建立起来，FORTRAN语言就一定能学好、用活。

习　题

1. 如何区别起始行与继续行？继续行标志是什么？继续行可否有标号？最多可续多少行？
2. 注释行的标志是什么？注释行能否有继续行？
3. END语句的作用是什么？
4. 标号的取值范围是什么？标号的作用是什么？
5. 什么是算法？试从日常生活中找三个例子，描述它们的算法。
6. 程序流程图（框图）在程序设计中有什么作用？
7. 用程序流程图表示求解以下问题的算法。

（1）依次将十个数输入，要求将其中最大的数打印出来。

（2）有三个数a、b、c，要求按大小顺序把它们打印出来。

（3）求两个数M和N的最大公约数。

8. 试用流程图来描述FORTRAN77解题的一般过程。

第二章　FORTRAN77语言基础

§2-1　数 据 类 型

计算机处理的数据对象是一个广义的概念。比如，230、34、56是数据；'he⌴fei'这一半字符也是数据。前者是数值数据，后者是非数值数据，虽然为了表示这些数据，它们在内存中必须以不同方式存放，为处理这些数据，计算机对它们施加的运算也不相同。为此，FORTRAN语言建立了数据类型的概念，对描述的数据进行分类。

FORTRAN77提供了6种数据类型：

（1）整型（INTEGER）；

（2）实型（REAL）；

（3）双精度型（DOUBLE PRECISION）

（4）复型（COMPLEX）

（5）逻辑型（LOGICAL）

（6）字符型（CHARACTER）

前面4种类型又称为算术型数据。FORTRAN77只能处理这六种数据类型。

FORTRAN77的数据类型具有以下特性：

（1）不同的数据类型有不同的外部表示形式和内部存储形式；

（2）常数的类型由其外部表示形式来确定；

（3）可以通过类型说明语句或FORTRAN77语言所规定的规则来说明数据的类型。

不同类型的数据有各自的数学意义、内部表示形式、书写格式和取值范围。

FORTRAN77规定：一个整型、实型、逻辑型数据占用1个数值存储单元，一个双精度型、复型数据占用相邻两个数值存储单元，而字符型数据中每个字符占用1个字节。一个数值存储单元通常指连续的4个字节。在许多FORTRAN77的编译程序中，一个整型数据只占2个字节。

§2-2　常　　数

FORTRAN77中的常数是指在程序运行过程当中保持不变的数据。FORTRAN77规定的常数分为：

常数
- 整型常数，或叫整常数
- 实型常数，或叫实常数
- 双精度型常数
- 复型常数，或叫复常数

（以上四项）数值型数据

- 逻辑型常数，或叫逻辑常数
- 字符型常数，或叫字符常数

FORTRAN77中的常数应该遵循一定的书写规则，其书写格式不但表示了它的值，而且