

依据教育部考试中心 1998 年
制定考试大纲编写

最新计算机等级考试(二级)用书

FORTRAN程序设计

刘万春 朱玉文 龚圆明 编



依据教育部考试中心 1998 年制定考试大纲编写

最新计算机等级考试(二级)用书

FORTRAN 程序设计

刘万春 朱玉文 龚圆明 编

国防工业出版社

· 北京 ·

图书在版编目(CIP)数据

FORTRAN 程序设计/刘万春等编. —北京:国防工业出版社, 2000. 3

(最新计算机等级考试(二级)用书)

ISBN 7-118-02169-5

I. F … I. 刘 … II. FORTRAN 语言—程序设计
N. IP312

中国版本图书馆 CIP 数据核字(1999)第 51642 号

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京怀柔新华印刷厂印刷

新华书店经销

*

开本 787×1092 1/16 印张 17½ 396 千字

2000 年 3 月第 1 版 2000 年 3 月北京第 1 次印刷

印数:1—3000 册 定价:24.00 元

(本书如有印装错误,我社负责调换)

出版者的话

掌握和使用电脑已是现代人综合能力的重要组成部分,近几年来,参加计算机等级考试的人数逐年递增。

教育部考试中心 1998 年制定的计算机等级考试大纲,对考试内容及要求重新做了说明。根据这一变化,我们出版了最新计算机等级考试(二级)用书(共 6 册):

计算机基础知识(DOS Windows)

QBASIC 程序设计

Pascal 程序设计

C 语言程序设计

FORTRAN 程序设计

FoxBASE 数据库管理系统

同时为帮助读者顺利通过等级考试,请读者去各考试点购买基于 Windows 95/98 环境编写的等级考试软件,有的软件具有电脑自动组卷、阅卷、评分及用户自行扩充试题数目、修改试题难易度等诸多功能,可用作实际考前训练。

本套书由北京理工大学计算机系教师编写,参加编写人员为教授、副教授,均为相应课程主讲教师,等级考试辅导班主讲教师,有多年教学经验。

本套书内容紧扣新大纲、详实、极具针对性,并附有大量习题和参考答案,是参加 1999 年及以后计算机等级考试人员理想用书。

我们真诚地期望,本套书能为您的考试顺利过关助一臂之力。

前 言

FORTRAN 语言是一种适合于数值计算的计算机高级语言,在科学和工程界得到广泛的应用。全国各地的高校也开设 FORTRAN 语言程序设计课程。国家教委考试中心主办的全国计算机等级二级考试大纲也将 FORTRAN 语言程序设计作为其中一种语种。

为了适应计算机高级语言发展的需要,应参加 FORTRAN 语言程序设计考试的广大读者要求编写了这本书。我们根据全国统一考试的要求,在本书中采用先综合各种 FORTRAN 语言教材的优点以及考试大纲的内容,科学划分章节,在每章中集中、通俗地讲解该章的内容,然后按各章中心内容给出例题进行分析,并在每章后收集大量试题、选择题、填空题、编程题供读者练习。

该书共分十一章,第一章程序设计基础;第二章 FORTRAN77 程序设计基础;第三章基本语句;第四章分支选择结构程序设计;第五章循环结构;第六章数组的应用;第七章函数与子程序;第八章数据联系;第九章字符处理;第十章文件;第十一章程序调试与上机指导;第十二章实践练习题答案。全书的内容覆盖了教育部考试中心计算机等级考试(二级——FORTRAN 语言程序设计)的考试要求。

计算机语言程序设计考试与其它学科的考试相似,需要打好基础、练好基本功。根据编者长期从事等级考试辅导班的经验,应试学习突出一个勤学苦练,只有通过实际练习,从练习中掌握 FORTRAN 语言语句应用范围和功能,研究程序设计中各种算法,总结出解决各类问题的方法。由此,希望读者阅读本书应在注重各章对基本问题的讲解,例题分析的基础上,大量作练习,上机实践,祝读者取得好成绩。

本书编写过程中得到国防工业出版社的大力支持,在此作者表示诚挚的感谢。在试题搜集方面,我们的学生刘俐、邢鹏、杨聪、郑红霞、夏军、徐岸、徐一华、李斌、顾伟、李小刚等也做出了努力,在此一并感谢。

由于作者水平有限,书中难免存在一些缺点和错误,诚恳希望广大读者批评指正。

编 者

1999 年 6 月

内 容 提 要

本书是根据教育部考试中心制定的“全国计算机等级考试大纲”中关于 FORTRAN 语言程序设计的要求编写。

本书系统地总结了 FORTRAN 语言程序设计的方法和技巧,对各部分内容作了合理的安排,按考试大纲的要求力争做到由浅入深,通俗易懂。每章从分析语言基本概念入手,分析语句,解释典型例题,归纳出解决问题的思路和最终求解方法。并且在各章后面附有大量实践练习和答案供读者练习和自我测试。

本书是为参加全国计算机二级等级考试的读者而编写的考前辅导教材,也可作为 FORTRAN 语言程序设计知识的自学教材和培训教材。

目 录

第一章 程序设计基础

1.1 什么是程序设计	1
1.1.1 目标问题的分析与算法	1
1.1.2 数据结构与描述工具	2
1.1.3 编程	7
1.1.4 程序调试、测试和运行	8
1.2 程序设计方法	8
1.2.1 结构化程序设计	8
1.2.2 模块化程序设计	8
1.2.3 逐步细化的设计方法	9
1.3 程序设计语言	9
1.4 习题分析	11
1.5 实践练习	12

第二章 FORTRAN77 程序设计基础

2.1 FORTRAN77 概述	13
2.1.1 FORTRAN77 程序构成	13
2.1.2 FORTRAN77 源程序的编译、连接和运行	16
2.1.3 FORTRAN77 程序在 DOS 下的执行过程	17
2.2 数据类型与运算表达式	18
2.2.1 常量	18
2.2.2 变量	20
2.2.3 运算表达式	21
2.3 常用内部函数	24
2.4 习题分析	25
2.5 实践练习	27

第三章 基本语句

3.1 执行语句和非执行语句	32
----------------------	----

3.2	赋值语句	32
3.3	PROGRAM、END、STOP、PAUSE 和 GOTO 语句	34
3.4	表控输入、输出语句	35
3.4.1	表控输出语句	35
3.4.2	表控输入语句	36
3.5	输入、输出格式语句	38
3.5.1	输出格式语句	38
3.5.2	FORTRAN77 常用格式编辑符	38
3.5.3	输入格式语句	42
3.5.4	输入、输出语句与格式语句的补充说明	44
3.6	PARAMETER 参数说明语句	46
3.7	习题分析	47
3.8	实践练习	49
3.8.1	选择题	49
3.8.2	填空题	52

第四章 分支选择结构程序设计

4.1	逻辑 IF 语句	56
4.2	块 IF 结构	57
4.3	用 ELSE IF 语句实现分支选择结构	59
4.4	块 IF 的嵌套	61
4.5	习题分析	62
4.6	实践练习	67

第五章 循环结构

5.1	当型循环	74
5.2	直到型循环	75
5.3	当型循环与直到型循环的联系与区别	77
5.4	DO 循环	78
5.4.1	DO 循环的一般形式	78
5.4.2	DO 循环的执行过程	79
5.4.3	DO 循环结构的规定	80
5.5	循环的嵌套	83
5.5.1	多重循环	83
5.5.2	循环嵌套的规定	84
5.5.3	循环程序举例	85
5.6	习题分析	90

5.7 实践练习	94
----------------	----

第六章 数组的应用

6.1 数组的定义	102
6.1.1 用类型说明语句定义数组	103
6.1.2 用 DIMENSION 语句定义数组	104
6.2 数组元素的引用	106
6.3 数组的存储结构	107
6.4 数组的初始化与输入输出	108
6.4.1 数组的初始化	108
6.4.2 利用 DO 循环对数组进行输入输出	111
6.4.3 用数组名实现整个数组输入输出	113
6.4.4 输入输出语句中使用隐含 DO 循环	114
6.5 习题分析	117
6.5.1 选择与填空	117
6.5.2 编程题	121
6.6 实践练习	124

第七章 函数与子程序

7.1 内部函数	137
7.2 语句函数	137
7.2.1 语句函数的定义	138
7.2.2 语句函数的引用	139
7.3 函数子程序	140
7.3.1 函数子程序的定义	140
7.3.2 函数子程序的调用	141
7.4 子例行程序	142
7.4.1 子例行程序的定义	142
7.4.2 子例行程序的引用	143
7.5 形参与实参的结合	144
7.5.1 变量、数组作为形参	144
7.5.2 可调数组	146
7.5.3 子程序名作为形参	147
7.6 习题分析	148
7.6.1 选择与填空	148
7.6.2 编程题	153
7.7 实践练习	154

第八章 数据联系

8.1 公用语句(COMMON 语句)	169
8.1.1 无名公用区	169
8.1.2 有名公用区	172
8.2 数据块子程序	174
8.3 等价语句	175
8.4 习题分析	177
8.5 实践练习	179

第九章 字符处理

9.1 字符串、字符型变量与字符型数组	183
9.1.1 字符串	183
9.1.2 字符型变量	184
9.1.3 字符型数组	184
9.1.4 字符子串	184
9.2 字符型数据的赋值与运算	185
9.2.1 字符型数据的赋值	185
9.2.2 字符表达式	186
9.2.3 字符关系表达式	187
9.2.4 常用字符串处理内部函数	188
9.3 字符型数据的输入输出	189
9.3.1 自由格式输入输出字符串	189
9.3.2 格式输入输出	189
9.4 习题分析	191
9.5 实践练习	194

第十章 文 件

10.1 文件与记录的概念	201
10.1.1 文件的组成	201
10.1.2 文件与文件存取方式	202
10.2 文件的操作	202
10.2.1 打开文件	202
10.2.2 文件的读写	204
10.2.3 文件定位	206
10.2.4 关闭文件	206

10.3	有格式顺序文件的存取	207
10.4	有格式直接文件的存取	209
10.5	无格式文件的存取	211
10.6	习题分析	211
10.7	实践练习	214

第十一章 程序调试与上机指导

11.1	FORTRAN 考试系统介绍	218
11.1.1	系统环境	218
11.1.2	上机考试时间	219
11.1.3	上机考试题型及分值	219
11.1.4	上机考试登录	219
11.1.5	试题内容查阅工具	222
11.1.6	考生目录和文件的恢复	224
11.1.7	文件名的说明	224
11.2	上机考试	225
11.2.1	DOS 常用命令操作举例	225
11.2.2	程序修改、调试、运行	227
11.2.3	编程、调试、运行	230
11.3	实践练习	232

第十二章 实践练习题答案

12.1	实践练习 1.5 答案	239
12.2	实践练习 2.5 答案	239
12.3	实践练习 3.8 答案	239
3.8.1	选择题	239
3.8.2	填空题	240
12.4	实践练习 4.6 答案	240
12.5	实践练习 5.7 答案	241
12.6	实践练习 6.6 答案	242
12.7	实践练习 7.7 答案	243
12.8	实践练习 8.5 答案	244
12.9	实践练习 9.5 答案	244
12.10	实践练习 10.7 答案	245
附录 A	ASCII 码表	247
附录 B	二级考试大纲与 FORTRAN 笔试样卷	248

第一章 程序设计基础

1.1 什么是程序设计

什么是程序设计？不同的人有不同的回答。一般认为“程序设计就是编制一个解决某一问题的计算机程序的过程”。这实质上是不对的，至少只是说到了程序设计过程中的某一阶段，表达不全面。通常程序设计包括多方面的内容，涉及算法、数据结构、设计方法和设计工具。要实现对目标问题的程序设计工作，可归结为下面的几个基本实现步骤：

- ①对目标问题的分析。
- ②选择解决问题的算法。
- ③数据结构与算法描述工具。
- ④编程。
- ⑤程序调试、测试和运行。

1.1.1 目标问题的分析与算法

1. 目标问题分析

对目标问题的分析是程序设计的基础。只有对问题进行充分地分析、理解后，才能寻找出正确的算法，有把握编制出高水平的程序，从而求得出正确的结果。问题的分析很复杂，程序员面对的是各种各样的问题，当然不同的问题就需要不同的解法。如何针对问题着手分析，一般来说应从下面的几个方面着手。

(1)从问题的性质方面：

人面对的问题是各种各样，而对于不同性质的问题，使用的方法、工具一般是不同的。例如：首先程序员应分析你面对的问题属于数值型数据的计算还是非数值型数据处理的问题。对于数值型数据的计算问题要考虑计算结果的精度，从而定义输入数据和中间结果的数据类型，以求获得一个合理的精度要求。对非数值型数据处理，则考虑输出结果与输入的关系，合理定义输入数据的数据类型，求得数据类型的统一。

(2)输入/输出：

程序设计中分析问题常常通过从输出的要求回溯到输入或从输入数据分析一步一步到输出。在这个分析过程中输入/输出的数据应从下面几个方面考虑。

- ①数据的类型，即数据设为整型、实型、字符型、双精度型等。
- ②输入/输出时数据定义格式。
- ③由哪些设备完成数据输入/输出。

(3)数学模型:

通过分析问题的性质,输入/输出数据类型后,一般就要考虑数学模型的设计。寻找一个适合于该问题的算法。

2. 算法的设计

算法一般说来是在有限步骤内解决一个目标问题,有明确意义描述的步骤的集合。即指对解题过程的准确而完整的描述。例如:要求一个圆的周长和面积就应知道求它们的公式:周长 $=2\pi R$,面积 $=\pi R^2$ 。写出对这个问题的算法如下。

①从键盘输入的半径 R , $\pi=3.1415926$

②计算周长 $Z=2 * \pi * R$

③计算面积 $S=\pi * R * R$

④显示 Z 和 S

从上面例子中看到算法不同于一般公式,算法应具有以下一些基本特征。

(1)可行性:

描述对某一问题的算法,每一步必须都能实现。例如,求算术平方根中出现在实数范围内对负数求平方根这样的算法描述,在除法运算中出现分母为零的情况等,在算法设计中就要避免,不允许出现。

(2)唯一性:

算法是针对某一问题提出的解决本问题的操作步骤。每一步必须确定,不允许出现模棱两可的解释,也不允许多义性,而是要给出针对此问题的唯一执行步骤。

(3)有穷性:

算法不允许无限制的进行计算,必须在一定的时间内完成。对于像数学中的无穷级数,在算法描述中只能按实际要求取有限项。

(4)明确数据源:

一个算法是否有效,要根据提供的数据进行实际检测,获得输出结果是否满足设计要求。这就要求数据的充分性,大量的实际数据和特殊数据才能验证算法的正确性和有效性。

综上所述,算法是一个针对某一问题的一组严谨的运算顺序的规则,它的每一规则都有明确的定义,并且都将在有限次的操作中终止。又因为算法是针对某一问题建立的,所以算法按其操作的数据对象常又分为:数值型求解算法和非数值型数据处理算法。

1.1.2 数据结构与描述工具

1. 设计数据结构

数据结构是计算机科学的重要基础,近年来已发展成一门专门的学科,其内容涉及到计算机的硬件和软件,它是应用程序设计中必不可少的一项数据描述。

(1)数据类型

在低级语言时,机器语言和汇编语言是不区分数据类型的,只有高级语言才区分数据的数据类型。这种数据类型的区分,使得程序员能按实际问题的要求进行数据组织,有效利用资源和实现算法。数据类型分原始与组合两大类型。

原始数据类型:数值类型、逻辑类型、字符类型、指针类型。

组合数据类型:由原始类型按某种方式组合构成。例如,数组类型、记录类型、文件类型等。

数据类型定义了一个运算的集合,不同的数据类型的数据参与不同运算方式。例如数值型才可以参加算术运算;字符型只能进行字符串的连接,求字符串长度等运算;而逻辑型数据只能进行逻辑运算(与、或、非等)。

数据类型还规定了该类型的数据定义域,例如,逻辑型数据的取值只能是“真”和“假”;字符型数据只能允许出现计算机中提供的字符集中元素组成;数值型数据在FORTRAN 77中,整型的取值范围一般微机系统中为 $-32768 \sim 32767$ 。

(2) 数据结构

数据结构是研究如何合理地组织处理数据,通常涉及三个方面:

数据逻辑结构

数据物理结构

数据结构上实施的运算

① 数据的逻辑结构。

在数据结构的研究中,常把数据称为一个结点(或原子)。各个结点之间的逻辑结构是数据元素代表的客体之间的联系。数据结构分线性结构和非线性结构两种。

线性结构的特点 仅有一个开始结点和一个终止结点,并且所有结点只有一个父结点和一个子结点,开始结点只有一个子结点,终止结点只有一个父结点。例如计算机中的队列、堆栈、表格等。

非线性结构的特点 有多于一个开始结点或终止结点,至少有一个结点有多于一个父结点或子结点。如树型、图等。

② 数据的物理结构。

数据的物理结构就是数据在计算机中的存储结构,数据的存储实现大致有以下四种方式。

顺序存储 将逻辑上相邻结点存储在物理上相邻的存储单元中。如图1.1所示,逻辑上线性结构为: $K_1 \rightarrow K_2 \rightarrow K_3 \rightarrow K_4 \rightarrow K_5 \rightarrow K_6$,顺序存放的存储单元为4000~4005。顺序存储特点是:没有连接信息域,只有信息本身,因此存储空间利用率高,但是插入、删除数据不方便。

地址	4000	K ₁
	4001	K ₂
	4002	K ₃
	4003	K ₄
	4004	K ₅
	4005	K ₆

图 1.1 顺序存储

链接方式 链接方式中每个结点由两部分组成:一部分存放信息;另一部分存放指针。指针用来指向该节点的父结点或子结点。链接方式有线性链接和非线性链接两种,如图1.2(a)与(b)所示。

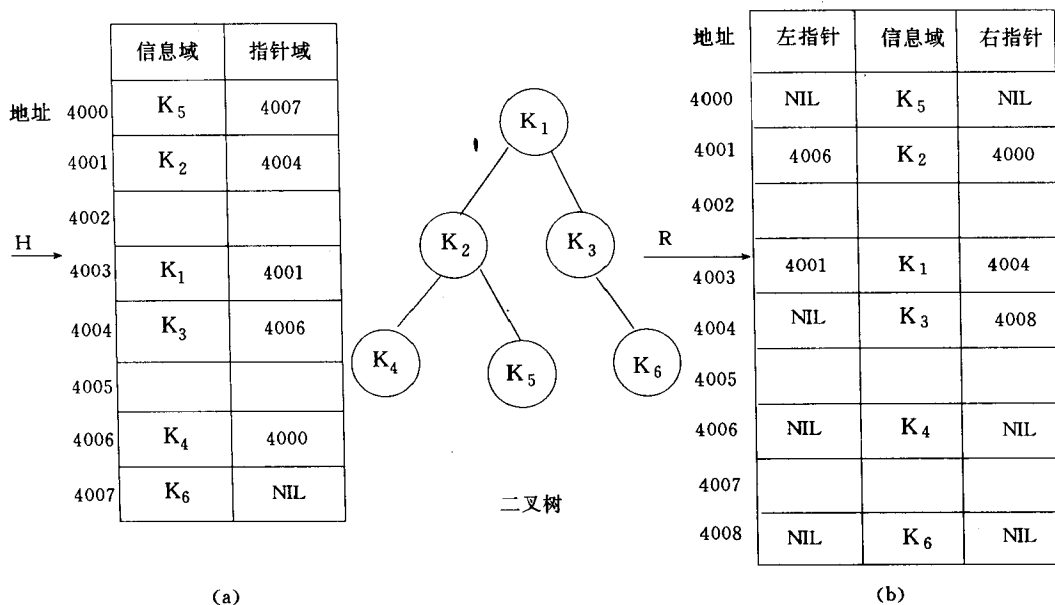


图1.2 链接方式

(a)线性链接;(b)二叉树(非线性链接)。

在线性链接的存储方式中,开始结点用一个指针“H”指向它,最后一个结点的指针为空,用“NIL”表示。

在非线性链接的二叉树物理存储方式中,每一个结点有两个指针域,左指针指向左边的子结点,右指针指向它的右子结点。二叉树的根结点有一个专门指针“R”指向它。

索引方式 通过结点在线性序列中的位置来确定结点的实际存储位置。

散列方式 通过结点的值来确定结点的存储位置。

后两种数据存储的物理结构一般用于系统软件的开发中。

③在数据结构上实施的运算。

常用于数据的运算有:

插入 往数据结构中增加新的结点。

查询 查找满足条件的结点。

更新 改变结点的部分或全部值。

排序 按要求对结点进行排列。

删除 删除指定结点。

2. 算法描述工具

程序设计的过程中,可以使用不同的算法描述工具确定解决问题的详细步骤。常用的描述工具有以下几种。

(1)自然语言

自然语言指人类在日常生活、学习、工作中通用的语言,这种语言不需要设计者作专门的学习和训练。但是使用自然语言作系统描述时要求用语简练,尽量减少语言修饰。例

如用自然语言描述打印两数之和及求平均值的算法：

- ①从键盘读入两个数 A1 和 A2
- ②计算两数之和 $S=A1+A2$
- ③求两数的平均值 $V=S/2$
- ④打印和 S 与平均值

(2) 算法描述工具

自然语言描述算法容易理解，一般适合于较小的程序设计，对于较大的程序设计工程使用自然语言进行算法描述；会令人感到不直观，容易产生多义性，也增加了理解的难度。使用算法描述工具是改善程序设计环境、提高设计效率和质量的一条重要途径。常用的算法描述工具有流程图、NS 图、PAD 图和 PDL 语言等。其中流程图和 NS 图使用最广泛。

① 流程图

流程图是用图形来描述问题的处理过程的工具。流程图普遍用于复杂计算和工程设计工具，它能直观、灵活地表现条件、活动和转移。图 1.3 为国家标准规定的一些常用图符。例如用图 1.3 的常用图符求解一元二次方程 $ax^2+bx+c=0$ 的算法描述如图 1.4 所示。

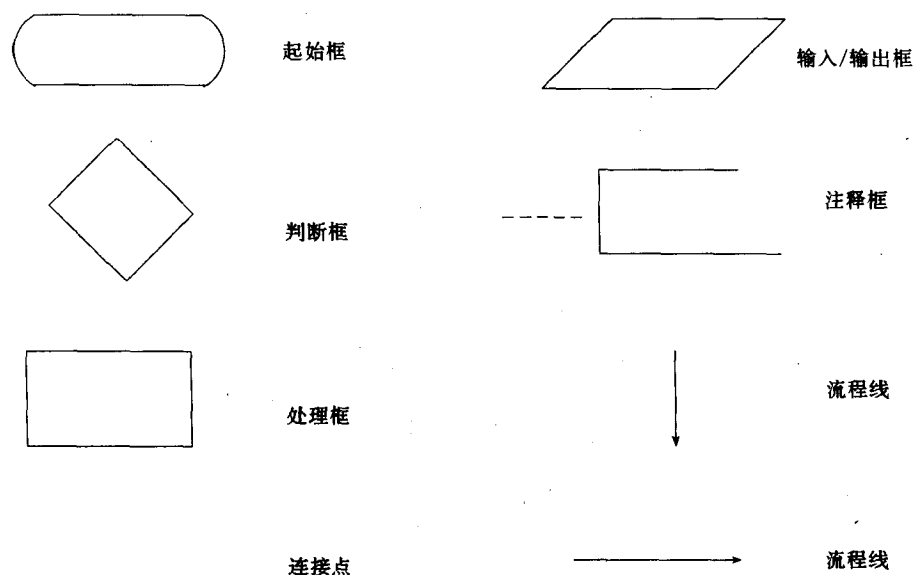


图1.3 常用图符

流程图把控制和执行顺序表达十分清晰，看起来也直观易懂。这就使得习惯用流程图的人对它偏爱，不愿接受其它新的工具。

然而不得不指出流程图存在严重的缺陷，由于流程线方向任意，如使用不当，会使得程序设计成为非结构化方式，影响程序设计质量。

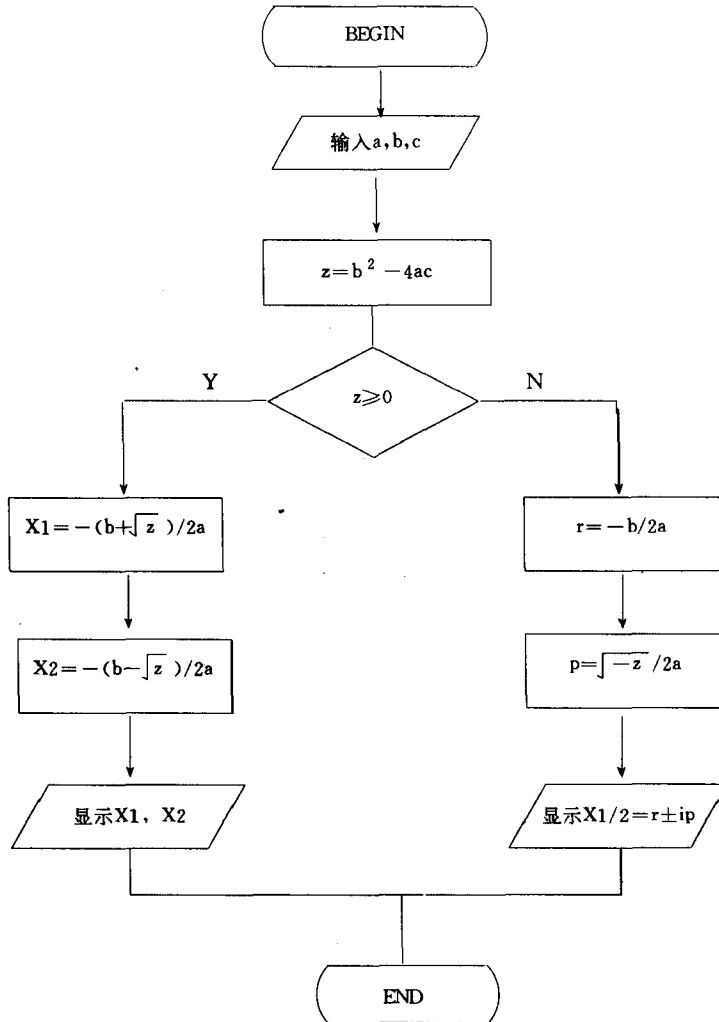


图1.4 一元二次方程流程图

②NS 图

NS 图是 1973 年美国 I. Nassi 和 B. Shneiderman 提出的一种新的符合结构化程序设计的流程图。NS 图有五种基本结构如图 1.5 中(a)所示。用 NS 的基本结构图描述的一元二次方程的算法,如图 1.5(b)所示,该算法描述写在一个框内,去掉了容易引起麻烦的流程线。

从 NS 流程图中看到,NS 图强迫设计者按结构化的要求构造算法。它有助于培养良好的按结构化原则进行程序设计的习惯。NS 图的缺点是由外框开始逐步向内画,可能图的整个布局由于考虑不周,使内部矩形功能域太小,无法向内层扩展,在设计时用户应尽量给出余量。