

使用算法语言编程序

——FORTRAN 语言浅谈

许德明 编著

科 学 出 版 社

1 9 8 5

内 容 简 介

本书通过解释 FORTRAN 语言,介绍了使用算法语言的基本概念、语法规则及其应用的一般知识。为便于读者自学,书中通过实例对各种基本概念、语法规则和使用方法进行了阐述,并在每章结束前,都对其内容进行归纳小结。

本书共分九章,并附有 FORTRAN 语句一览表、思考题和部分参考答案。可供具有中等文化程度以上的学生、教师,计算机爱好者,以及有关工程技术人员阅读参考。

使用算法语言编程序 ——FORTRAN 语言浅谈

许德明 编著

责任编辑 陈永镛 曾美玉

科 学 出 版 社 出 版

北京朝阳门内大街 137 号

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1985 年 2 月第 一 版 开本:787×1092 1/32

1985 年 2 月第一次印刷 印张:9

印数:0001—40,300 字数:175,000

统一书号:15031·629

本社书号:3865·15—8

定价: 1.70 元

前 言

算法语言,指为了表达解题而设计的、主要面向解题过程的语言。使用算法语言编制计算机程序后,即可利用电子计算机解决各种问题。

FORTRAN 语言是一种用于科技计算的算法语言。在同类算法语言中, FORTRAN 语言目前在国内外应用最广。这种语言具有许多优点,诸如简明直观,结构清晰,描述能力较强,输出格式完备,标准化程度高,计算速度快等等。因此,它受到科技工作者的欢迎,从五十年代以来一直沿用至今。

本书是一本 FORTRAN 语言的入门读物,供初学者使用参考。在书中,作者力图通过实例对各种基本概念、语法规则和使用方法进行阐述,注意总结有关内容,并做到尽量有利于读者自学。为此,书中提供了较多例题,在每章结束前都对本章的主要内容进行了归纳整理,并在书末列出了一批思考题和部分参考答案,以供读者使用。在解题时,算法语言编制的程序可以有多种形式,因而不必拘泥于书中的答案。

本书承蒙北京工业大学陈祖荫同志、中国科学院计算技术研究所王树林、黄琪瑞同志、中国人民解放军 38662 部队肖国伦同志和中国科学院生物物理研究所韩慧婉同志的审阅修改,陈建军同志参加了框图的绘制工作,谨在此表示感谢。

对于书中的不妥之处,敬请读者批评指正。

目 录

前 言	v
第一章 从计算机的使用谈起	1
(一) 问题的分析	1
(二) 框图的设计	2
(三) 程序的编制	9
(四) 调试与计算	12
(五) 例题	13
(六) FORTRAN 语言简介	17
第二章 基本符号和赋值语句	24
(一) 字符集	24
(二) 符号名	25
(三) 常数	26
(四) 变量	29
(五) 算术表达式	32
(六) 函数	38
(七) 算术赋值语句	41
(八) 小结	42
第三章 输入/输出语句	44
(一) 输入	45
(二) 输出	51
(三) 文字输出	58
(四) 格式语句的规则	60
(五) 停语句和暂停语句	60

(六) 其它语句和描述符	62
(七) 例子	67
(八) 小结	70
第四章 转语句和条件语句	74
(一) 无条件转语句	74
(二) 逻辑条件语句和逻辑赋值语句	75
(三) 简单循环	82
(四) 算术条件语句	84
(五) 计算转语句和赋值转语句	88
(六) 例子	93
(七) 小结	100
第五章 循环	103
(一) 循环语句	103
(二) 继续语句	112
(三) 多重循环	115
(四) 例子	119
(五) 小结	129
第六章 数组	133
(一) 一维数组	133
(二) 维数语句	136
(三) 类型说明	138
(四) 下标的形式	139
(五) 二维数组	144
(六) 三维数组	149
(七) 输入/输出中的隐含循环	151
(八) 例子	156
(九) 小结	165
第七章 字符处理和数据初值语句	167

(一) H型字段描述符	167
(二) A型字段描述符	169
(三) 数据初值语句	176
(四) 文字型数据	181
(五) 小结	184
第八章 过程与辅程序	186
(一) 语句函数	188
(二) 函数辅程序	192
(三) 子程序辅程序	198
(四) 子程序辅程序和函数辅程序的比较	209
(五) 辅程序的优点	209
(六) 可调数组	210
(七) 例子	215
(八) 小结	223
第九章 其它语句	227
(一) 外部语句	227
(二) 等价语句	230
(三) 公用语句	233
(四) 数据块辅程序	238
(五) 小结	239
附录A FORTRAN 语句一览表	241
附录B 思考题与部分参考答案	244

第一章 从计算机的使用谈起

电子数字计算机是一种现代化的计算工具，它能接收外部信息，并按照事先存储在计算机内的程序对信息自动进行处理，然后提供处理结果。计算机通常由存储器、运算器、控制器以及输入、输出设备组成。一般认为，计算机的使用范围可分为科学计算、数据处理和自动控制三个领域。本书以介绍科学计算问题为主，但它的很多原理同样适用于其它两个方面。

使用计算机解题的过程大致可归纳为分析问题，设计框图，编制程序与调试计算四个步骤。下面逐一介绍这些步骤。

(一) 问题的分析

问题分析，通常包括以下三个方面：

建立数学模型，即建立用于解决该问题的数学公式或方程。

确定数值解法，即确定解数学模型的计算方法。数值解法应能保证所需的精确度，还应尽量节省运算时间和存储空间。

规定算法，即规定以机械执行的一系列计算步骤。计算

机根据算法逐步进行运算以求得所需的结果。算法必须具有确定性(含意明确,只能有一种解释),大量性(可以接受不同的初值并计算出对应的结果)和有效性(当初值在允许的范围内变化时,计算总能停止并得到结果)。下面将要介绍的框图和程序都是算法的具体表达形式。

(二) 框图的设计

框图是算法的一种表现形式。所谓框图,就是用预先规定好的各种几何图形、流线和文字说明,去描述计算过程的图,也称为流程图。

框图的优点是直观、清晰、易懂,便于检查、修改和交流,框图的设计有助于进行正确的程序设计。尤其是在处理较为复杂的问题时,更能显出它的优越性。因此,通常都把框图设计作为程序设计的必要步骤。

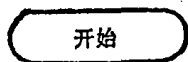
下面,介绍本书所使用的框图及其表示法。

终端框

表示程序的开始或结束,用符号



表示。例如,用



表示一个程序的开始;而用

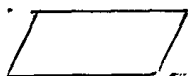
结束

表示一个程序的结束。

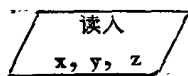
终端框也可在插入信息或离开程序时使用。

输入/输出框

当从光电输入机、卡片输入机等外部设备把数据输入到计算机内或从内存储器输出数据到外部设备或进行打印时，用符号



表示。输入时在框中写上“输入”或“读入”；输出时在框中写上“输出”或“打印”。例如，输入三个数 x, y 和 z ，可表示为



判断框

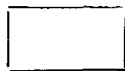
在一个程序中，常常遇到需要检验某些条件是否成立，并根据检验结果而转移到程序的不同部分或执行不同的任务。这可用符号



表示。需判断的条件写在框中，根据条件选择的路线由从判断框引出的流线表示。

加工(处理)框

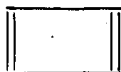
程序中的计算和存储器中数据的传送，用符号



表示。所执行的任务写在框中。

调用子程序框

当在主程序（或一个子程序）中需要调用另一个子程序时，用符号



表示。调用的子程序名和实在参数写在框中。

循环语句框

当需要反复运行某一段程序时，用符号



表示。循环终端语句的标号、循环参数写在框中。

流线

框图中各个框之间的连接线称为流线，用符号



表示。箭头表示前进的方向。

连接符

为缩短流线长度，避免引起混乱，用符号



或



表示各部分流线的汇合点。在符号中，可用数字或字母表示各部分流线间的连接关系。

设计框图的规则：

① 每个程序框图应有开始框和结束框；

② 框图中每个框应有流线连接，如果一页纸写不下，必须利用连接符；

③ 框图中不允许流线穿过任何一个框；

④ 框里面可写上句子、文字或符号；

⑤ 可用双引号（“...”）标记有关信息。

下面用几个例子说明设计框图的方法。

【例1】 输入两个数 X 和 Y ，计算它们的和 (SUM) 与差 (DIF)；并打印输出四个数： X 、 Y 、 SUM 和 DIF 。程序框图可用图 1.1 表示。

【例2】 确定一元二次方程

$$ax^2 + bx + c = 0 \quad (a \neq 0)$$

的根的性质。

根据初等代数，可知一元二次方程根的表达式为：

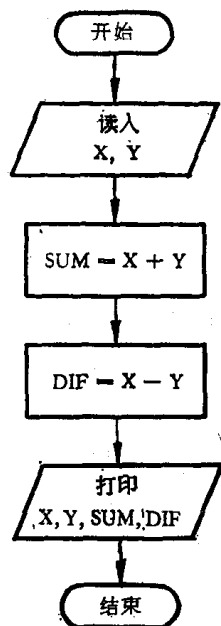


图 1.1

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

根的性质取决于 $\Delta = b^2 - 4ac$ 的值:

- ① 当 $\Delta < 0$ 时, 有两个复数根;
- ② 当 $\Delta = 0$ 时, 有两个相等的实数根;
- ③ 当 $\Delta > 0$ 时, 有两个不相等的实数根。

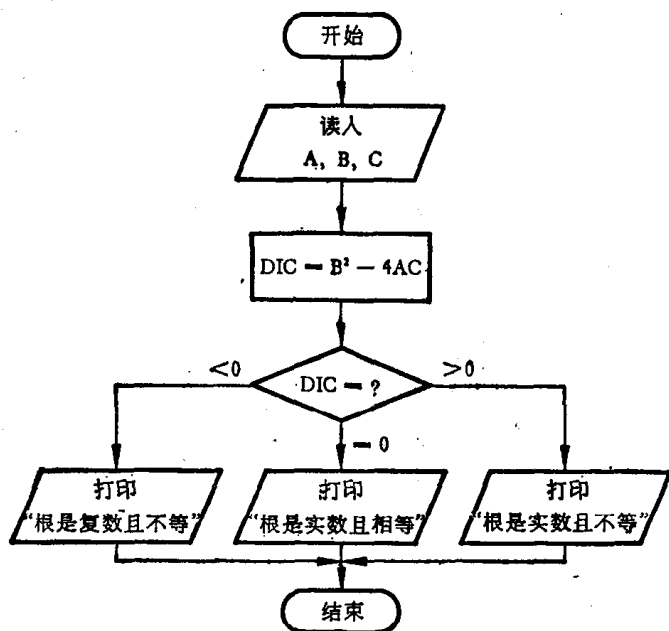


图 1.2

图 1.2 是对应的框图, 它表示先输入 a , b 和 c , 然后检查 $b^2 - 4ac$ 的值, 并输出方程根的性质。

【例 3】 如果要求【例 2】中的程序重复执行 100 次,即逐个判断一百个方程的根的性质,则可把图 1.2 改为图 1.3 (a) 或图 1.3(b)。

在框图 1.3(a)中,计数器 I 控制被处理方程的个数,在开

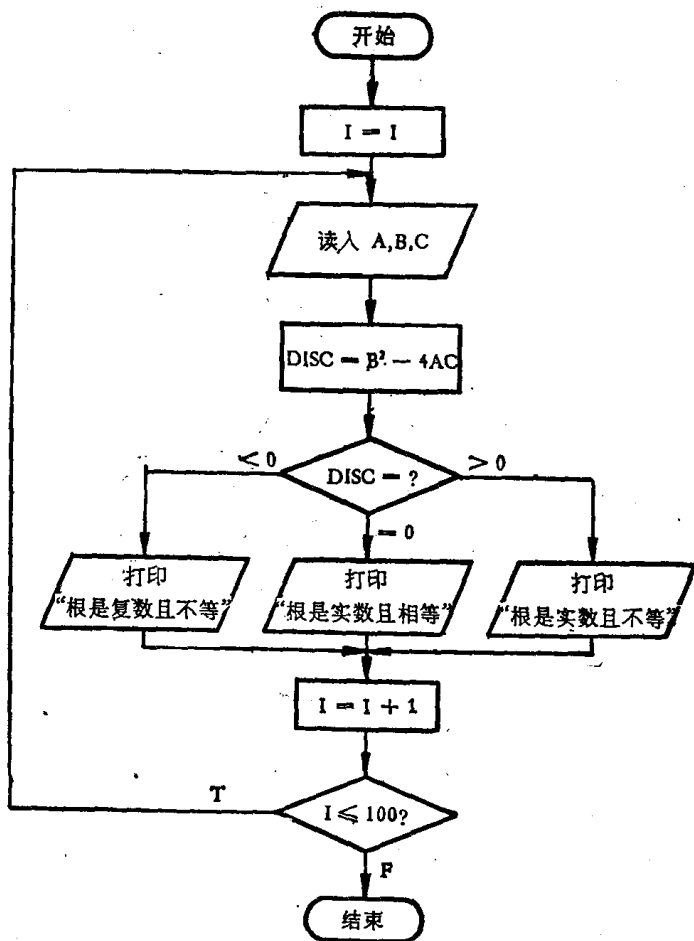


图 1.3(a)

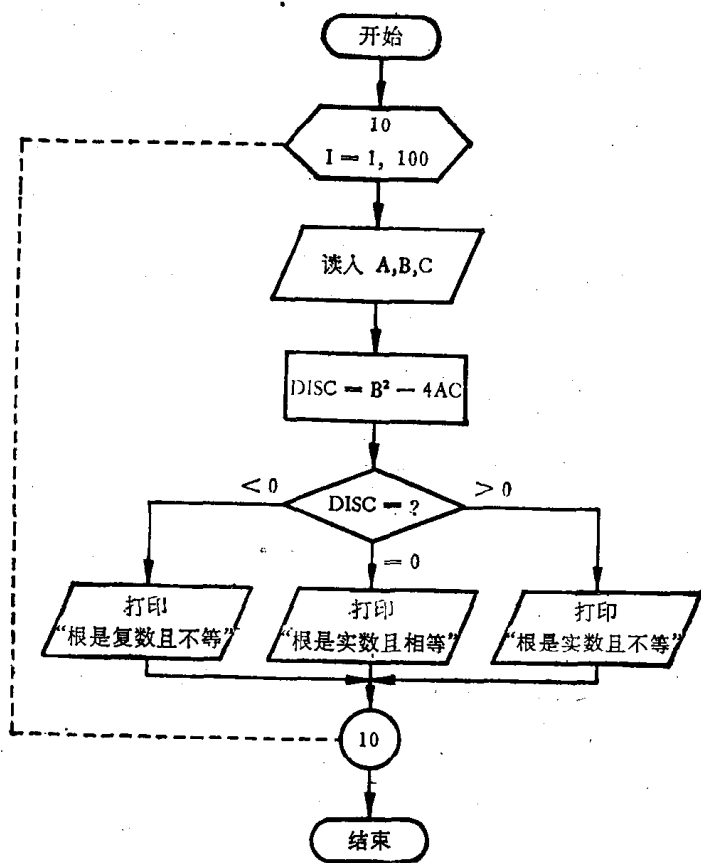


图 1.3(b)

始时值为 1($I = 1$)。每处理完一个方程，计数器 I 增加 1 ($I = I + 1$)，并检验其值是否小于或等于方程个数 100 ($I \leq 100$)。这段程序重复一百次，直到 $I > 100$ 才停止。图中的“T”与“F”分别表示“真”(True)与“假”(False)。

(三) 程序的编制

编制程序是使用计算机的关键，只有把框图转化为机器所能识别的程序，计算机才能进行计算。

要想编制程序，首先必须懂得计算机语言，即程序设计语言。计算机的语言种类繁多，其范围从机器语言到面向过程或面向问题的语言。按其从繁到简的等级，大致可分为机器语言、汇编语言和高级语言等几种。

计算机是以机器语言工作的，因此，其它形式的语言都必须翻译成机器语言。现在就上述三类语言逐一介绍如下：

1. 机器语言

机器语言犹如“方言”，取决于机器的类型，并具有专用性。为某种机器设计的机器语言只能在这种机器上使用。

用机器语言编写的程序称为手编程序。编写手编程序时，要按照机器的“指令”把要解决的问题逐步写出，然后交给计算机运行。

用机器语言编写的程序，不需要经过翻译就可由计算机直接执行。因此，用机器语言编写的程序执行快，所用的存储单元少。早期的计算机，都是使用机器语言的。但是，由于机器语言一般用二进制数字表示，因此，书写繁琐，容易出错，编写程序效率低，而且程序不容易阅读和交流。

为了摆脱对机器指令的依赖，使人们从手编程序的繁重

而机械的劳动中解放出来,减少编写程序的时间,同时也使程序便于交流、修改、阅读和检查。人们又设计出各种使用更为方便的语言。

2. 汇编语言

汇编语言,是一种面向机器的程序设计语言,也是一种用符号表示的低级程序设计语言。它通常也是为特定的计算机专门设计的,很接近机器语言。但是,在汇编语言中可以使用代码表示各种运算和数据,因此可以不必再使用繁琐的二进制数字。用它编写的程序需要翻译成机器指令,但这种翻译工作是由计算机中的翻译程序(称为汇编程序或汇编语言加工程序)做的。汇编语言指令和翻译后的机器指令之间基本是一一对应的。

例如: 计算 $A = B + C$

用汇编语言可写为:

```
LDA  B  (从内存中取 B)
ADD  C  (加上 C; 即 B + C)
STA  A  (存结果到 A)
```

用汇编语言编写的程序,称为源程序。由源程序翻译成的机器语言程序,称为目的程序或目标程序。

汇编程序的优点是:

- ① 容易编制(因为可利用易记忆的各种符号);
- ② 容易修改(因为翻译时计算机能自动调整地址);
- ③ 容易检查错误(因为汇编程序具有查错功能)。

3. 高级程序设计语言

汇编程序虽然比手编程序进了一步,但仍需依赖具体的机器,在不同型号的机器上很难互换,通用性差。随着计算机的不断发展,从五十年代中期起便产生了各种独立于机器、面向过程或面向问题的语言,称为高级语言。

面向问题的语言,是一种独立于机器的语言。使用这种语言解题时,不仅摆脱了计算机的机器指令,而且不必关心问题的解法和计算过程的描述,只需提出问题和输入、输出数据的形式,就能得到所需的结果,如表报语言等。

面向过程的语言,也是一种独立于机器的语言。使用这种语言解题时,也不必了解计算机的机器指令,而主要考虑解题的算术、逻辑运算过程的描述。由于这种语言对解题过程的描述比较接近人的思维和书写方式,因而简单易学,便于推广使用。FORTRAN 语言就是一种面向科技计算过程的高级程序设计语言。

用高级语言编写的程序也称为源程序。源程序送入机器后,必须经过编译程序翻译成机器语言,即转化为目的程序。编译程序就象一个翻译员,它检查程序员编写的程序语法是否正确,并将其翻译成机器指令。

高级语言的优点是:

① 通用性强,用这种语言编写的源程序,可在配有相应编译系统的任何计算机上运行;

② 直观,接近人们陈述问题的习惯,容易理解、学习、修