

1038
农业系统工程讲义

第三分册

算 法 语 言

ALGOL 60 (DJS-21机)

丛 书 本

中国科学院农业现代化研究委员会
全国农业系统工程研究会学习班 编印
湖南省系统工程学会农业系统工程研究会

一九八一年八月

前 言

电子计算机是近三十多年来发展起来的一门新技术。它具有运算速度快、计算精度高、存储量大等特点，因此，它可以解决大量繁杂的计算任务和信息处理任务。目前它已成为国民经济、国防建设、科学研究等部门不可缺少的工具，它的应用越来越广泛，已深入到人类生活的各个方面。

本讲义主要介绍DJS-21机算法语言，介绍利用DJS-21机算法语言编写程序的方法，还通过一些例子说明在编写程序时应注意的一些问题，本内容可作为算法语言的入门，对DJS-21机，掌握了本内容完全可以编写ALGOL算法语言程序，在DJS-21机上进行算题，以解决我们工作中遇到的许多科学计算问题，它又是手头上的一本较详尽的参考使用手册。

本讲义较适用于初次接触电子计算机和算法语言的读者，内容力求做到：深入浅出，条理清楚，例子详尽，上机操作说明具体，各章附有习题，书后附有习题答案，便于工程技术人员掌握和自学。

121机是国内较为普及的一种机器，该机的ALGOL60编译系统是由北大、八三〇厂等单位设计的，后经数学所，中大，川大及用户进行了修改补充，使之逐步完善。本书内容除第一章外主要取材于DJS-21机ALGOL60算法语言（川大等），上机操作说明，部分取材于〈ALGOL121使用手册〉（长沙矿山设计研究院计算站）。为了适用于农业系统工程学习班，作了部分增删。

121机ALGOL60算法语言和编译系统，在我国许多经济、技术部门推广应用计算机方面是很有贡献的，现在从全国农业战线来的同志们，来学习使用这种语言，它将在我国农业上推广应用计算机，影响深远，实值欣慰。

错漏之处，恳望批评指正。

一九八一年七月

目 录

第一章 电子计算机概述.....	(1)
§ 1、电子数字计算机简介.....	(1)
§ 2、数制及其互相转换.....	(2)
§ 3、程序设计自动化和ALGOL 60语言.....	(7)
第二章 基本符号和基本概念.....	(9)
§ 1、基本符号.....	(9)
§ 2、标识符.....	(10)
§ 3、数.....	(11)
§ 4、标准函数.....	(12)
第三章 表达式、说明、语句.....	(14)
§ 1、类型和类型说明.....	(14)
§ 2、简单算术表达式.....	(15)
§ 3、赋值语句.....	(17)
§ 4、复合语句和分程序.....	(19)
§ 5、输入和输出语句.....	(21)
§ 6、条件语句、简单布尔表达式.....	(24)
§ 7、条件表达式.....	(30)
§ 8、转向语句、开关说明、命名表达式、空语句.....	(33)
§ 9、数组说明.....	(38)
§ 10、循环语句.....	(42)
§ 11、循环优化.....	(53)
第四章 程序的结构、量的作用域.....	(58)
§ 1、程序的结构.....	(58)
§ 2、量的作用域.....	(59)
第五章 过程与函数.....	(68)
§ 1、无参过程.....	(68)
§ 2、有参过程.....	(68)
§ 3、函数过程.....	(70)
§ 4、过程的例.....	(80)
§ 5、标准过程与辅助语句.....	(83)
§ 6、有关宽行输出.....	(87)
第六章 语句的描述.....	(93)

第七章 操作使用说明	(100)
§ 1、源程序准备	(100)
§ 2、纸带准备	(102)
§ 3、上机操作	(106)
§ 4、程序调整	(109)
附录 1、ALGOL121 错误性质表	(113)
附录 2、ALGOL121 与 ALGOL60 的区别	(119)
附录 3、输出格式	(121)
附录 4、库过程使用说明	(121)
习题解答	(123)

第一章 电子计算机概述

§ 1 电子数字计算机简介

电子数字计算机是由电子元件构成的一种现代化计算工具，它好象一个自动的数字加工厂，用数字和代码作为原料经过处理、运算得出数字的成品。电子计算机所以能够高速运算，除了它使用了电子元件这一因素外，本质的原因是它具备了计算的全部功能并能事先接受计算任务，接受后能自动执行，因此数字计算机必须具备记忆、运算、控制功能。为了人—机交换信息还必须输入输出设备。

下面我们以DJS—21型电子计算机（简称21机）为背景，介绍电子数字计算的主要结构和工作原理。

DJS—21机是我国自行设计和制造的一种中小型晶体管通用数字计算机，主要供科学计算使用，也可做为控制机用，采用二进制数，字长42位，平均运算速度为3万次/秒。内存容量为16384个单元。主要部件情况如下：

1. 存储器 它分内存和外存，用来存储程序、原始数据和计算结果。DJS—21机的内存存储器是由磁芯体构成，它好比是一座大楼，由许多称为单元的房舍所组成，单元的总数称为计算机的容量。磁芯体单元是由许多个能够表示二进制数字的小元件串联而成的，小元件的个数称为单元的字长。另外为了安排存取信息的需要，每个磁芯体的单元还人为的编上一个号码依次为0~16383（十六进制数编号为0000~3555）。单元字长42位，另有两个奇偶校验位，共44位。

外存存储器（它不直接和运算器发生联系）由磁鼓和磁带机，可以配到四台鼓，每台容量为16384个单元。磁带机两台，其容量视带长而定，如果考虑磁带换用，其容量便是任意的了。外存做为内存的补充，它可与内存成批交换代码。

存储器的每个单元都有这样的特点，代码读出后，该单元还保留原来的代码，当对某单元送入新的代码时，原来的内容便告消失。所以在存储器这个特点可概括为“取之不尽，一拼就走”、初学使用者必须对此给予充分注意。

2. 运算器 运算器的作用相当于算盘，在计算机中它是用来对数或指令进行算术运算和逻辑运算的。如：加、减、乘、除等，还能进行其他操作，在计算时先要从内存中取出代码到运算器中然后才能运算，计算结果仍保留于运算器中，也可以再送入内存。

21机的运算器是由I、II、III寄存器和一个全加器组成。I寄存器是结果寄存器，存放器则运算第一个数和计算结果。II寄存器是乘商寄存器，存放器则运算第二个数，有时也存放部分运算结果。III寄存器是兼有指令寄存器与变址寄存器功能。全加器能做加减运算，在传送代码时全加器是一个中间通道。

3. 控制器 它是一个指挥机构，它指挥全机按照人们预先制定的步骤进行各种操作。它由指令计数器、脉冲分配器、译码器、中央控制器、中断控制线路组成，其功能是对程序中每条指令内容进行分析判断，决定执行那种操作，通过控制器发出一系列控制信号，控制全机自动协调的工作。

4. 控制台 它是供使用者和机器维护人员操纵用的, 控制台有许多开关和电钮, 我们按动开关和电钮可操作全机工作。控制台绿灯向使用者显示机器内部工作状态。

5. 输入、输出设备 输入设备有光电输入机两台, 其速度为1000个符号/秒, 它将穿孔纸带上所穿制的代表程序或数据的信息送入机器的内存储器。还有电传打字机, 输入速度慢, 每秒7个字符以下, 它也是一种输出设备, 是程序自动化的一种辅助设备。

输出设备有窄行快速打印机, 每行80个字符, 每秒打印5—10行。

121机整机布局大体如下:

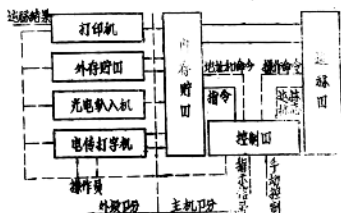


图1-1 121机主要部件

存储器、运算器和控制器构成主机部分, 为了使主机与人及外部发生关系, 需要配置各种设备, 这些部分统称外设部分。121机还可以配置绘图机, 快速穿孔机等。大型计算机还要配置更多的外设。主机和外设, 通常称为“硬件”。硬件好比没有受过“训练”的头脑和四肢。一个计算机厂家不仅要生产硬件, 更重要是使自己的产品“受教育”, 为它配置“编译程序”, 使它们能读解和执行不是指令而且“语言”写出的程序。还要配上管理自己的程序(“操

作系统”)和方便用户的程序(“服务程序”)以及其它程序, 这些程序统称为“软件”。软件好像是知识, 它反映了计算机的文化水平。这是一个无形的财富, 它凝聚了人们使用计算机的大量经验、技巧和智慧。硬件只有在软件的指挥调动下, 才能获得生命力, 成为处理能力强的计算机系统。

一个软件或硬件人员须对计算机有较仔细的了解, 对我们工作中使用算法语言的科技人员, 也应了解一些有关软、硬知识, 以发挥计算机的潜力, 然而这不是使用计算机的先决条件。

§ 2数制及其互相转换

一、电子计算机通用二进制数, 它同人们日常所采用的十进制数是有差别的, 十进制数是‘逢十进一’计数制, 每个数都是用0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 正负号, 小数点组合起来的。

例如: $2768_{(10)} = 2 \times 10^3 + 7 \times 10^2 + 6 \times 10^1 + 8 \times 10^0$

记号(10)一表示该数是十进制形式。

又如: $25.683_{(10)} = 2 \times 10^1 + 5 \times 10^0 + 6 \times 10^{-1} + 8 \times 10^{-2} + 3 \times 10^{-3}$

二进制数就是数的每一个数位上都是由0, 1两个数字中之一构成的, 外加正负号和小数点, 其进位规则是“逢二进一”。

例如: $11101_{(2)} = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$$= 16 + 8 + 4 + 0 + 1 = 29_{(10)}$$

$$\text{又如: } 110.011_{(2)} = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} \\ = 6\frac{3}{4} = 6.375_{(10)}$$

$$100000000000_{(2)} = 1 \times 2^{12} = 4096_{(10)}$$

二进制四则运算简单, 只要记住“逢二进一”和向上位借 1 算 2 就行了。

例如:

$$\begin{array}{r} 101.1 \\ + 111.01 \\ \hline 1100.11 \\ \\ 11.001 \\ \times 1.01 \\ \hline 11.001 \\ 1100.1 \\ \hline 11.11101 \end{array} \qquad \begin{array}{r} 101.1 \\ - 100.001 \\ \hline 1.01 \\ \\ 10.1 \\ - 11.001 \\ \hline 1.01 \\ - 10.1 \\ \hline 101 \\ - 101 \\ \hline \end{array}$$

为什么电子计算机一般都是采用二进制呢? 这是因为二进制记数法有许多优点, 主要是:

1. 易于在物理装置上实现。二进制数每位只有 0 和 1 两种数字, 这用物理状态表示非常容易。例如电位的高低, 信号的有无, 半导体管子的导通和截止, 磁芯的两种不同饱和状态, 灯的亮与不亮……等等都可以用来表示二进制的每一位数。

2. 可以节省材料。十进制每位可以是 0~9, 共有十种不同状态, 因此为了要表示 00~99 的所有数字必须有 $2 \times 10 = 20$ 个状态的装置, 而以这 20 种物理状态表示二进制数, 则可以表示十位。这时数的范围是 0~1023。就这个意义讲二进制比十进制节省设备。

3. 运算简单。因为二进制构造简单, 其运算规则也十分简单, 例如十进制中乘法规则(九九表)就没有了, 二进制中只有四种情况:

$$0 \times 0 = 0$$

$$1 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 1 = 1$$

这样, 机器实现四则运算的线路便大为简化。

二、二进制数与十进制数换算

当然, 我计算问题时总是使用十进制数, 而机器内部是二进制运算, 我们使用电子计算机时, 应该了解二进制与十进制间的换算规则。

例如: 24.25 是用十进制表示的, 用二进制表示它可改写为

$$24.25 = 16 + 8 + \frac{1}{4}$$

$$= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

因而 24.25 的二进制表示是: 11000.01 , 一般来说, 对一个数 $A_{(10)}$ 可以写成

$$A_{(10)} = a_k \cdot 2^k + a_{k-1} \cdot 2^{k-1} + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0 + a_{-1} \cdot 2^{-1} + \dots + a_{-2} \cdot 2^{-2} + \dots + a_{-n} \cdot 2^{-n}$$

因而可用代码 $a_k a_{k-1} \dots a_1 a_0 a_{-1} a_{-2} \dots a_{-n}$ 表示数 $A_{(2)}$, 其中 a_i ($i=k, k-1, \dots, -n$) 为 1 或 0。

当A为整数时, $a_i \equiv 0 \quad (i=1, 2, \dots, n)$

当A为小数时, $a_i \equiv 0 \quad (i=0, 1, 2, \dots, k)$

我们分三种情况来讨论, 将十进制数化为二进制数的方法

1. 整数情形: 假若要把195写成

$$\begin{aligned} 195 &= a_k \cdot 2^k + a_{k-1} \cdot 2^{k-1} + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0 \\ &= 2(a_k \cdot 2^{k-1} + a_{k-1} \cdot 2^{k-2} + \dots + a_1) + a_0 \\ &= 2(2(a_k \cdot 2^{k-2} + a_{k-1} \cdot 2^{k-3} + \dots + a_2) + a_1) + a_0 \\ &= 2(\dots 2(a_k \cdot 2 - a_{k-1}) + \dots + a_2) + a_1 + a_0 \end{aligned}$$

上式两端除以2商为: $a_k \cdot 2^{k-1} + a_{k-1} \cdot 2^{k-2} + \dots + a_1$ 余数为 a_0 , 再除以2商为: $a_k \cdot 2^{k-2} + \dots + a_2$, 余数为 a_1

逐次继续下去便可得 $a_0, a_1, a_2, \dots, a_k$. 即:

2	195	余 数
2	97	1
2	48	1
2	24	0
2	12	0
2	6	0
2	3	0
2	1	1
	0	1

所以 $195_{(10)} = 11000011_{(2)}$

总结整数十换二的方法是: 将十进制数逐次除2, 取余数直至商为零。将各次余数按先下后上的顺序排列, 便是二进制整数。

2. 小数情形: 假定要化0.5625为二进制小数, 设化得结果为 $0.a_1a_2\dots a_n$, 于是

$$0.5625_{(10)} = a_{-1} \cdot 2^{-1} + \dots + a_{-n} \cdot 2^{-n}$$

上式两端乘以2, 右端得:

$$a_{-1} + a_{-2} \cdot 2^{-1} + \dots + a_{-n} \cdot 2^{-n+1}$$

此式继续上面的过程, 最后可得到:

$$0.5625_{(10)} = 0.1001_{(2)}$$

总结起来, 十进小数化为二进制小数规则是: 将十进小数乘2取整数部分, 再乘再取……取出的整数部分按先左后右的顺序排在小数点之后便得二进制小数。

3. 十进制化为二进制: 对于兼有整数和小数两部分的十进制数, 化为二进制数时, 将两部分分开, 分别化为二进制数后再合起来, 便得之。

4. 二进制化为十进制: 这只需将二进制数写成幂的形式, 按十进制数算出便得。例如:

$$\begin{aligned} 1001101.101_{(2)} &= 2^6 + 2^3 + 2^2 + 2^0 + 2^{-1} + 2^{-3} \\ &= 64 + 8 + 4 + 1 + 0.5 + 0.125 \\ &= 77.625_{(10)} \end{aligned}$$

三、十六进制: 二进制数固然有许多优点也有其缺点, 不直观, 书写一个数很长。为了克服这个书写不便利十六进制数进行书写 (有的机器用8进制书写), 因为 $16=2^4$, 所以二

进制数与十六进制间转换十分容易,四位二进制可以表示一个十六进制。十六进制基数是16,每位有0, 1, 2, 3, 4, 5, 6, 7, 8, 9, $\overline{0}$, $\overline{1}$, $\overline{2}$, $\overline{3}$, $\overline{4}$, $\overline{5}$ 十六种可能性,这里 $\overline{0}$ — $\overline{5}$ 分别表示10—15在计算机打印时印成A, B, C, D, E, F。

十六进制的加法规则是“逢十六进一”。

十六进制与十进制转换规则可完全仿照前面所讲的 $10 \rightarrow 2$, $2 \rightarrow 10$ 的规则,只要将2换成16就可以了。例如将十进整数化为16进制整数的方法是“用除法,取余数”。

例如将563化成16进制数,其算式是

$$\begin{array}{r}
 16 \overline{) 593} \quad \text{余数} \\
 \underline{16 \overline{) 37}} \quad 1 \\
 \underline{16 \overline{) 2}} \quad 5 \\
 0 \quad 2
 \end{array}$$

所以 $593_{(10)} = 251_{(16)}$

将十进小数化为十六进制小数的方法也是“用乘法,取整数”。兹不一一介绍。

这里我们主要讲一下十六进制与二进制的关系,因为在DJS-21机上要用到这一点,请熟记下面的表:

十进制	二进制	十六进制
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	$\overline{0}$
11	1011	$\overline{1}$
12	1100	$\overline{2}$
13	1101	$\overline{3}$
14	1110	$\overline{4}$
15	1111	$\overline{5}$

有了上表, 一个二进制数, 很快就可写成十六进制数, 如二进制数 1100110111 可分为三组,

反之，知道一这十六进制数，很容易写出二进制数。如：

四、数的定点、浮点表示和规格化数

10×32.158 ; $10^3 \times 0.32158$; $10^4 \times 0.032158$ 等等,这几个数写法其值完全相等。在计算机中数一般按后面两种形式表示。一般任何数M可写为

其中 $0 \leq N < 1$, p 为整数。称 N (如 0.32158) 为数的尾数, 而 p (如 10^3 中 3) 叫做阶码, 阶码有 + 或 -, 叫阶符, 数也有 + 或 -, 叫做数符。

阶P是可变的计算机叫浮点计算机、这样的数称为浮点数，每个数除给出尾数外，还要给出数的阶。

$$M = \pm 2^{\pm P} \times S \quad (0 \leq S < 1)$$

其中阶P是可变的，就是P可浮动。S叫做尾数。在浮点数中，尾数S的第一位的数不为零叫做规格数。如果S的第一位是零称非规格化数。

$2^4 \times 0.0101010 \dots$ 叫做非规格化数。

121机实数输出都是十进制规格化数。

DJS-21机的内存贮器为42位，因此用42位表示一个数。阶码阶码占7位，数符占1位，尾数点34位，具体划分如下：

当 $P \geq 0$ 阶符用0表示, $P < 0$ 阶符用1表示。当 $S < 0$ 数符用1表示, $S \geq 0$ 数符用0表示。

在121机中，尾数是用原码表示的，阶码是用补码表示的。当 $P < 0$ 时，我们规定把

$$2^7 - |P|$$

当做P的补码。例如：

$P = -101101$ 它的补码为1010011

因为

$$\begin{array}{r} 10000000 \\ - 101101 \\ \hline 1010011 \end{array}$$

仔细观察一下补码形式，便可发现，负数的反码就是其反码加1，即先用7位二进制数写出 $|P|$ （不足7位，前面补0），再求 $|P|$ 的反码（即把7位中的0改为1，1改为0），然后加1而得。

如 $P = -64 = -2^6 = (-1000000)_2$

$P_{反} = 0111111$

$P_{补} = 1000000$

前面我们规定了阶码为负数时阶符为1，因此负阶码的表示范围是1000000~1111111即为十进制 $-64 \sim -1$ 。每正阶码的符号为0因此正阶码的表示范围是0~63。

由此可见，121机能表示数的阶码范围是 $-64 \leq P \leq 63$ 。数的范围是：

$$2^{-64} \leq |x| \leq 2^{63}$$

相当于

$$10^{-19} \leq |x| \leq 10^{19}$$

§ 3 程序设计自动化和ALGOL 60语言

每种计算机都有自己特有的指令系统。小型机有十几条指令，大型机可有几百条指令。DJS-21机有59条指令。为了编制程序就必须知道“机器语言”，即掌握其指令系统，编制程序时，无论是内存分配、指令执行顺序、输入输出组织等等，都由程序一条一条精心安排。这是一种及其烦琐、容易出错的工作。检查和排除错误，也要花费很大人力。而且这种程序难以交流和互用。在实践中人们想，把编程和检查错误的繁重劳动大部分交给计算机自己去作。这就是程序设计自动化。为此人们就制定了一套与具体计算机无关的“算法语言”。算题人员只需用这种语言写出一篇类似于文章的文字一称“源程序”。每种计算机的生产要由专业程序员用“机器语言”编写一套“编译程序”，它能接受和阅读源程序，检查其中的错误，最后把它翻译成机器语言程序一称“目标程序”。

使用电子计算机和算法语言解题的全过程示于图1—2中，它包括提出问题、分析问题、建立数学模型、选择算法、制定框图、编写调试程序、准备原始数据，实际上机计算及评价计算结果。

在动手编写程序之前，要下功夫调查、分析、思考自己提出的实际问题，整理出数学公式描述，使之尽可能反映问题的量与量之间的关系。然后选择恰当的计算方法，这是解题过程中最体现人的主观能动性的阶段，极为重要的阶段。

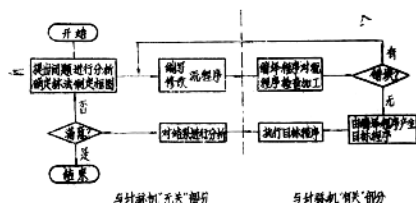


图4-2 用算法语言解题过程

用程序语言把计算过程描述出来就是程序设计。程序设计好了，要仔细审查，直到查不到错误以后才进行穿孔。

上机算题先调试，调试通过后才正式算题，计算结果是否正确要细心分析。结果正确与否与数学模型、计算方法、程序及机器运行是否正确等都有关系。解题的全过程实际上是一个循环往复，逐渐趋于正确认识的过程。

从五十年代以来，世界上制订了几百种算法语言，有的适用于科学计算，有的适用于行政管理 and 经济业务等等，如FORTRAN语言、ALGOL语言、COBOL语言、BASIC语言、PL/I语言……关于算法语言本身及其编译技术已形成了系统的理论，成为计算机科学的一个重要部分，并且有了很大的研究设计专业队伍。一般用户只需熟习运用某语言来解决自己的实际问题。

ALGOL 60是国际上通用的一种算法语言，ALGOL是英语ALGOrithmic Language的缩写，即算法语言的意思，它是1960年在法国巴黎召开的国际会上被接受为国际语言的，故称为ALGOL 60。

本书主要介绍DJS-21机ALGOL语言，它与ALGOL 60基本相同，比较容易掌握。在国产机器上大都配备了与ALGOL 60接近的语言，因此掌握了一种机器的算法语言之后，也就很容易掌握另一种机器上的算法语言。目前国内121机，108乙机，X-2机，709机，TQ16机等所配的ALGOL 60语言，基本大同小异。

习 题

1. 将下列十进制数化为二进制和十六进制数：
(1)4095, (2)256, (3)0.1, (4)1/4
2. 区别下列名词：源程序，ALGOL程序，编译程序，目标程序。
3. 试说明计算机解题的全过程。

第二章 基本符号和基本概念

§ 1. 基本符号

任何语言的字和词都是由一些基本符号组成的。例如英语的基本符号就是26个英文字母和十个标点符号。ALGOL₁₂₁算法语言（以下简称本语言）也是由一些基本符号组成的。这些基本符号共有91个，可将它们分为四类，即：字母、数字、逻辑值、定义符（介限符）。现分述如下：

一、字母（26个英文字母）

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,

字母没有独立的意义，可用它们组成标识符和行。字母可以大写，也可以小写，如果使用了小写字母，便作对应的大写字母来处理。

二、数字

0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

数字是用以组成数，标识符和行的。

三、逻辑值

TRUE（真），FALSE（假）。

逻辑值有确定的意义，它表示某个条件成立或不成立，当条件成立时取值为TRUE，否则取值为FALSE。

四、定义符（介限符）

定义符可分为五类，即运算符、分隔符、括号、分类符、说明符。

1. 运算符

运算符又可分为四类，即：算术运算符、关系运算符、逻辑运算符、顺序运算符。

（1）算术运算符

+, -, ×, /, //, ××

（加），（减），（乘），（除），（整除），（乘方）

（2）关系运算符

'LS', 'LQ', '=', 'GR', 'GQ', 'NQ',

（小于），（小于等于），（等于），（大于），（大于等于），（不等）

（3）逻辑运算符

'OR', 'AND', 'NOT', 'IMP', 'EQV'

（逻辑加），（逻辑乘），（逻辑非），（蕴涵），（等价）

（4）顺序运算符

'GOTO', 'IF', 'THEN', 'ELSE', 'FOR', 'DO'

(转向), (如果), (则), (否则), (对于), (做)

2. 分隔符

, , . , 10 , ; , : , := , :: = , 'STEP',
(逗号), (小数点), (小拾), (冒号), (分号), (赋值号), (双点赋值号), (空格), (步长),
'UNTIL', 'WHILE'
(直到), (当)

3. 括号

(,) , [,] , ' , ' , 'BEGIN', 'END'
(圆括号), (方括号), (定义符号括号), (行括号), (语句括号)

4. 分类符

'LABEL', 'VALUE'
(标号), (值)

5. 说明符

'REAL', 'INTEG', 'BOOLE', 'ARRAY', 'PROC', 'SWITCH',
(实型), (整型), (布尔型), (数组), (过程), (开关),
'LIBR', 'DRUM', 'TAPE'
(库), (鼓), (带)

用本语言写的ALGOL₁₂₁程序就是这些基本符号的有序序列。为使写出的程序清楚易读, 在程序中, 还可以随意地使用空格、返回和换行, 它们对ALGOL₁₂₁程序没有影响。这些基本符号的作用将在以后各章中陆续说明。

§ 2. 标识符

在数学中常常使用字母或符号来表示各种量。例如半径R, 高H, 面积S, 变量X、Y、Z等等。代表这些量的符号在本语言中我们称为标识符。也就是说, 标识符是给程序中所用的量起的名字。在满足以下条件的前提下, 我们可以随心所欲地给程序中的量起名字。

1. 以字母开头的任何字母和数字的组合;

例如: ALPHA, B17A, C1D2T4, X, YZ, 等等, 都是标识符。

2. 不允许有字母、数字以外的其它符号;

例如: A—B, f(x), α , β , 5T, x·y等等, 都不是本语言中的标识符。

3. 逻辑值、显示量、标准函数及标准过程名有固定的意义, 不可作为其它标识符使用;

4. 标识符不超过7个有效字符, 如超过七个时则只取前七个并打出出错信息, 例如TEMPERATURE与TEMPERA是一样的, 前者要打出标识符过长信息。但标识符间可随意地使用空白、空格, 如: ALPHA和ALPH A是同一个标识符。

例如按照通常的习惯, 我们选用S (面积), H (高), ALPHA (α), L (长度), SIGMA (Σ) 等等作标识符。

标识符可以用来标记程序中用到的简单变量、数组、标号、开关和过程。

在通常情况下, 不同的量要用不同的标识符来标记, 但在不同的作用域内却可以用同样的标识符来标记不同的量, 且在不同的作用域内所代表的意义是不同的。

§ 3. 数

本语言中的数，就是通常所熟悉的十进制数，只是书写方式与通常习惯略有差异，例如：

十进制数	本语言的数
23	23
-18	-18
1.7321	1.7321
3×10^4	3 ₁₀ 4
1.25×10^{-2}	1.25 ₁₀ -2
-84×10^{-3}	-84 ₁₀ -3

语言中的数一般形式可表为：

$$\underbrace{\langle \text{数符} \rangle \langle \text{无符号整数} \rangle}_{\text{整数部分}} . \underbrace{\langle \text{无符号整数} \rangle}_{\text{小数部分}} {}_{10} \underbrace{\langle \text{数符} \rangle \langle \text{无符号整数} \rangle}_{\text{指数部分}}$$

其中：〈数符〉或者为正（+）号或者为负（-）号，当其为正时，“+”号可以不写：

〈无符号整数〉是仅由数字组成的序列，如：012，318，4120等等；分隔符“.”和“₁₀”是两个基本符号。

任何一个数都可由这样三个部分组成，也可以只有其中的一个或两个部分，但顺序不能变更。

数可分为两类：

一、整型数

如果一个数只有整数部分，则它就是一个整型数。例如：214，-327，+4587，0；130等等。

二、实型数

除整型数以外的数称为实型数。

实型数可以由如上三个部分组成的数，也可以是由其中一部分或两部分组成的数。

例如：+1.25，12.0，3.4₁₀5，4₁₀5，₁₀-2，-5.23，-0.2₁₀3，-0.025₁₀-3，123.45₁₀-2等等。

特别地，实型数可以以小数点开始，亦可以以小数点结束。

例如：25.，.156，2.₁₀-2，.35₁₀6等等。

实型数的指数部分是一个比例因子，表示10的整数次方，符号₁₀（小拾）与后面的整数组成一个完整的部分，它和正整数10要严格地区别开来，₁₀（小拾）是本语言的一个基本符号，而正整数10是由数字“1”和“0”组成的一个数。

由于指数部分的作用，同一个数可以表示成多种不同的形式。

例如：0.25，25₁₀-2，0.0025₁₀2，2.5₁₀-1等等都表示同一个数；因此，某些实数可以写为更简单的形式，如。

123000.0可写为: 1.23₁₀6

0.0000005可写为: 5₁₀-7

但必须注意, 5和5.0则是两个不同类型的数, “5”是一个整型数, 而5.0则是一个实型数; 同样, 100和1.02也分别表示两个不同类型的数, 前者为整型数, 后者是实型数。

整型数之间进行的运算称为定点运算, 实型数之间进行的运算称为浮点运算。一般说来, 计算机作定点运算比作浮点运算速度更快, 并且整型数在机内的表示是不带误差的, 而实型数作浮点运算则往往会有一些误差, 如 2.0×3.0 的实际结果可能是5.9999999, 也可能是6.0000001而不一定刚好为6 (但这个误差是很小的, 只当所计算的数很小时, 才变成是有意义的)。由于实型数的这个弱点, 当需要用完全精确表示的数时, 用整型数是方便的。但实型数所能表示的范围比整型数要大得多, 所以, 两者各有利弊, 使用者可以根据自己的需要随意选用。

由于计算机字长的限制, 数的表示范围作如下规定:

1. 输入数据:

整型数绝对值不超过 $2^{34}-1=17179869183$ 。

实型数绝对值不超过 $2^{63}(1-2^{-34}) \approx 9.2 \times 10^{18}$ 。

2. 输出数据:

限制同1, 但最多只能有8位有效数字。

如果数小于机器所能表示的范围则认为是0, 如果大于机器所能表示的范围则发生溢出。

§ 4. 标准函数

在数学中经常使用一些初等函数, 如三角函数 $\sin x$, $\cos x$, 反三角函数 $\arcsin x$, 指数函数 e^x , 对数函数 $\ln x$ 等等, 在本语言中将这函数作为标准函数。引入标准函数的目的是为了在程序中省写它们的说明, 以简化ALGOL程序。

在写源程序时, 凡需用以下标准函数可以不加说明而直接引用, 这些标准函数是:

1. ABS(E)求E的绝对值, 函数值的类型同E。

其中E为算术表达式 (以后介绍), 以下皆同。

2. SIGN(E)

$$\text{符号函数SIGN}(E) = \begin{cases} -1, & \text{当 } E < 0 \\ 0 & \text{当 } E = 0 \\ 1 & \text{当 } E > 0 \end{cases}$$

是整函数。

3. GN2(E)求E的平方根, 类型为实。

4. GN3(E)求E的立方根, 类型为实。

5. SIN(E)求E的正弦值, 类型为实。

6. COS(E)求E的余弦值, 类型为实。

7. TAN(E)求E的正切值, 类型为实。

8. ARCSIN(E)求E的反正弦主值, 类型为实。
9. ARCTAN(E)求E的反正切主值, 类型为实。
10. LN(E)求E的自然对数, 类型为实。
11. EXP(E)求E的以e为底的指数, 类型为实。
12. TOINTEG(E)对E四舍五入取整, 结果为整。
13. ENTIER(E)求不大于E的最大整数, 类型为整。
14. TOREAL(E)把E化为实型数, 类型为实。
15. READI从数据区读入一个整数, 类型为整。
16. READR从数据区读入一个实数, 类型为实。
17. READB从数据区读入一个逻辑值, 类型为布尔。
18. SUM(A)求数组A的全部元素的代数和, 值的类型同A。
19. MULI(X, Y)求X与Y之内积, 值的类型同X、Y, 且X、Y是介偶表相同的同类型数组。

20. FACT(E)求E的阶乘, 类型为整。当 $E > 13$ 时, 认为参数错。
21. MAXA(A)求数组A全部元素之最大值, 类型同A。
22. MINA(A)求数组A全部元素之最小值, 值的类型同A。
23. TRACE(A)求方阵A的迹, 类型为实。
24. RAND(E)产生伪随机数, 类型为整。

使用标准函数必须注意:

1. 不可丢掉圆括号, 比如ABS(E)表示求E的绝对值, 而ABSE则是一个标识符;
2. 标准函数5, 6, 7中的E是弧度表示。

习 题

1. 下列字符串中哪些是本语言的标识符?

APril, fl, A(2), A·B, C123, SEMI—K, π , LN, 3CT, x', β , MAN'S, y(x), A1B2C3.

2. 以下字符串中哪些是本语言的数?

① 222 ② e^4 ③ $\sqrt{3}$ ④ 21,345 ⑤ $42_{10} - 3$ ⑥ 62×10 ⑦ 2.0 ⑧ $-1.2100, 9$ ⑨ ∞ ⑩ 3344 ⑪ 3.116 $\times 2^3$ ⑫ $2_{10}3.4$.

3. 以下数中哪些是实型数, 哪些是整型数?

1, 1, +4, -231, 4759, .785, -.021, +4.32, 10_3 , -10_2 , -3.14159, -4, $467_{10} - 3$, 4.0, 0.0.

4. 把以下写为形如 $a_{10}b$ 的浮点数, 其中 a 位于区间 $0.1 \leq a < 1$, b 为正的或负的整数。

0.00125348, 12.3674, 7.862, 3.14159, -1, 12000000, 0.00000512.