新世纪

新世纪应用型高等教育计算机类课程规划教材
信息技术基础系列教材

C语言程序设计

C YUYAN CHENGXU SHEJI

主编 彭正文 徐新爱
副主编 胡 佳 卢 昕 章 逸
主审 涂振宇

大连理工大学出版社

新世纪应用型高等教育计算机类课程规划教材
信息技术基础系列教材

新世纪

C语言程序设计

C YUYAN CHENGXU SHEJI

主编 彭正文 徐新爱
副主编 胡 佳 卢 昕 章 逸
主审 涂振宇

大连理工大学出版社

图书在版编目(CIP)数据

C语言程序设计/彭正文,徐新爱主编. —大连:大连理工大学出版社,2009.8
新世纪应用型高等教育计算机类课程规划教材
ISBN 978-7-5611-5056-6

Ⅰ.C… Ⅱ.①彭…②徐… Ⅲ.C语言—程序设计—高等学校—教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2009)第 150051 号

大连理工大学出版社出版
地址:大连市软件园路 80 号　邮政编码:116023
发行:0411-84708842　邮购:0411-84703636　传真:0411-84701466
E-mail:dutp@dutp.cn　URL:http://www.dutp.cn
大连美跃彩色印刷有限公司印刷　　大连理工大学出版社发行

幅面尺寸:185mm×260mm　印张:14.25　字数:329 千字
印数:1～3100
2009 年 8 月第 1 版　　2009 年 8 月第 1 次印刷

责任编辑:潘弘喆　孙就鑫　　责任校对:王　朝
封面设计:张　莹

ISBN 978-7-5611-5056-6　　定　价:26.00 元

前　言

C语言程序设计是高等院校计算机专业及其他专业一门重要的基础课程，该门课程学习的好坏，直接影响着后续专业课程的学习。如何选好一本教材是C语言学习的关键。

本书的目标是力争成为通俗易懂、实用的教材和参考资料。具体体现在以下几个方面：

1. 为了检查读者的学习效果、巩固知识点，在大多数例题后面都有“注意”、“思考”和“实践”三大环节。

2. 以VC++ 6.0编译环境介绍程序的调试过程，所有的程序都在VC++ 6.0下调试通过。

3. 配备大量实用经典例题，对程序中每行语句都做了详尽的解释。

4. 注重章节学习，提出章节学习目标。

5. 配备大量的习题。习题类型丰富，具有广泛的代表性和实践性。

本书第1章介绍了程序设计语言的基本知识和计算机解决问题的一般过程；第2章通过三个例题介绍了C语言源程序的基本元素；第3章介绍C语言源程序的基本元素的作用与应用；第4章通过大量的例题，分别介绍C语言源程序的三种基本结构，让读者了解程序设计的基本思路和方法；第5章介绍数组的使用；第6章介绍函数的结构和使用；第7章通过多个例题介绍指针和构造类型在程序设计过程中的使用；第8章介绍文件的使用；第9章介绍预处理命令；第10章介绍Windows窗口编程基础，这部分内容属于选学内容。附录一给出了ASCII码对照表，附录二列出了C语言中的保留字，附录三列出了常用的一些C语言库函数，供读者参考。

新世纪

我们还编写了配套的实验指导书，书中包括典型习题解析、上机实验指导和实验题。每个上机实验都经过编者精心设计。读者可以先按照上机示例一步步操作，然后再做实验题，最后写出完整的实验报告，循序渐进地理解和掌握C语言程序设计的思想、方法和技术。

本书在编写过程中，得到了有关部门的大力支持，计算机教研室的同事们为本书的编写做了许多的工作，在此一并表示衷心的感谢。还要感谢大连理工大学出版社的领导和编辑们对我们编写工作的大力帮助！更要感谢广大读者的厚爱！

由于作者水平有限，书中难免会有不足之处，恳请读者批评指正。

所有意见和建议请发往：gzjckfb@163.com
欢迎访问我们的网站：http://www.dutpgz.cn
联系电话：0411-84707492　84706104

编　者

2009年8月

此为试读，需要完整PDF请访问：www.ertongbook.com

目　录

第1章 概 述

导引

本章将介绍计算机程序设计的基本概念和程序设计语言的基础知识，带领读者迈入程序设计的大门。通过介绍C语言的历史及发展来初步了解这门历史悠久、应用广泛的计算机程序设计语言。

学习目标

- ◇掌握计算机解决问题的过程
- ◇了解C语言的历史及发展
- ◇了解C语言的特点
- ◇了解C编译环境

1.1 程序设计的基本概念

程序(Program)就是为使电子计算机按特定的逻辑顺序执行多个操作而编排的计算机指令的集合，一般分为系统程序和应用程序两大类。程序设计(Programming)是指设计、编制、调试程序的方法和过程。程序设计通常分为问题分析并建模、算法设计、编写代码和编译调试四个阶段。

1.1.1 计算机和程序

中国古代最早采用的一种计算工具叫筹策，又被叫做算筹，它算得上是人类发明的最古老的计算工具了。这种算筹多用竹子制成，也有用木头、兽骨充当材料的，约二百七十枚一束，放在布袋里可随身携带。今天仍然在使用的珠算盘，是中国古代计算工具领域中的另一项发明，明代的珠算盘已经与现代的珠算盘几乎相同了。在古欧洲及西亚也同样出现过一些类似的计算工具，可见，对于计算工具，人类是在不断的探索中。

1642年，年仅19岁的法国伟大科学家帕斯卡借用算盘的原理，发明了第一部机械式计算器。在他的计算器中有一些连锁的齿轮，一个转过十位的齿轮会使另一个齿轮转过一位，人们可以像拨电话号码盘那样，把数字拨进去，计算结果就会出现在另一个窗口中，但是他的这种计算器只能做加减计算。1694年，莱布尼兹在德国将其改进成可以进行乘除运算的计算器。

以上提到的算盘和各类机械计算器有一个共同的缺点：不能按人们事先设计好的计算步骤进行运算。长期以来，人们一直渴望能发明一种可以按照人类意愿处理一些计算事务的计算工具，来帮助人们处理一些现实世界中繁琐的计算事务。

要让机器听人类的话，按人类的意愿去计算，就要实现人与机器之间的对话，或者

说，要把人类的思想传送给机器，让机器按人的意志自动执行。实现人与机器对话的创始人是两位法国纺织机械师。他们发明了一种能指挥机器工作的“程序”，把编织图案直接“注入”到了提花编织机的针尖上。1725年，法国纺织机械师布乔(B. Bouchon)突发奇想，想出了一个“穿孔纸带”的绝妙主意。布乔首先设法用一排编织针控制所有的经线运动，然后取来一卷纸带，根据图案打出一排排小孔，并把它压在编织针上。启动机器后，正对着小孔的编织针能穿过去钩起经线，其他的针则被纸带挡住不动。这样一来，编织针就自动按照预先设计的图案去挑选经线，布乔的“思想”于是“传递”给了编织机，而编织图案的“程序”也就“储存”在穿孔纸带的小孔之中。

另一位法国机械师杰卡德(J. Jacquard)，大约在1805年完成了“自动提花编织机”的设计制作。杰卡德编织机上“千疮百孔”的穿孔卡片，让机器编织出绚丽多彩的图案，这是程序控制思想的萌芽，早期电脑的存储程序和数据穿孔纸带及穿孔卡片思想大概就是来源于此了。或许，我们现在把“程序设计”俗称为“编程序”，就引申自编织机的“编织花布”的词义。

现在公认的第一台可编程的计算机器是由英国数学家查尔斯·巴贝奇设计的。它采用类似杰卡德编织机穿孔卡片的思想来选择步骤执行算术运算。巴贝奇设计的机械计算机限于当时的技术条件没有生产出来。英格兰诗人拜伦(Byron)勋爵的女儿左斯特·艾达·洛夫莱斯伯爵夫人是一位数学家，她曾对巴贝奇的笔记、手稿进行了整理和修正。她为巴贝奇的计算机设计了许多算法，从某种意义上说，她是世界上第一位计算机程序员。

现代的计算机模型是由美籍匈牙利数学家约翰·冯·诺依曼(如图1-1所示)发明的。由他领导设计的第一台电子计算机于1946年2月14日问世。电子计算机的发明大大促进了科学技术和社会生活的进步。鉴于冯·诺依曼在发明电子计算机中所起到的关键性作用，他被誉为“计算机之父”。

冯·诺依曼计算机思想的主要贡献在于存储程序原理：程序由指令组成，并和数据一起存放在存储器中，计算机只要知道第一条指令的位置就能自动地按照程序指令的逻辑顺序逐条把指令从存储器中读出来，自动完成由程序所描述的处理工作。存储程序原理的提出是计算机发展史上的一个里程碑，也揭示了计算机与其他计算工具的根本区别。

如今，计算机产业已经发展了60余年，生产出了种类繁多、性能差异很大的各种计算机，但不管怎样，它们都是基于冯·诺依曼模型结构的。

图1-1 冯·诺依曼

此为试读，需要完整PDF请访问：www.ertongbook.com

现代计算机模型可以分为四个子系统：存储器、算术逻辑单元、控制逻辑单元、输入输出单元，如图 1-2 所示。

• 存储器(Main Memory)：用来存储程序和数据的电子器件。计算机中的全部信息，包括输入的原始数据、计算机程序、中间运行结果和最终运行结果都保存在存储器中，它根据控制器指定的位置存入和取出信息。

• 算术逻辑单元(Arithmetic Logic Unit，ALU)：计算机中执行各种算术和逻辑运算操作的部件。运算器的基本操作包括加、减、乘、除四则运算，与、或、非、异或等逻辑运算，以及移位、比较和传送等操作。计算机运行时，运算器执行的操作由控制器决定。运算器处理的数据来自存储器，处理后的结果数据通常送回存储器，或暂时寄存在运算器中。

• 程序控制单元(Program Control Unit)：计算机的核心控制部件，负责对整个计算机进行控制，包括从存储器中取指令、分析指令(即指令译码)、确定指令操作和操作数地址、取操作数、执行指令规定的操作及送运算结果到存储器或 I/O 端口等。它还向微机的其他各部件发出相应的控制信号，使主机和设备能协调工作。

• 输入输出设备(I/O Equipment)：负责计算机与外部的数据交换。其定义相当广泛，如键盘、鼠标等属于输入设备，显示器、打印机等属于输出设备。

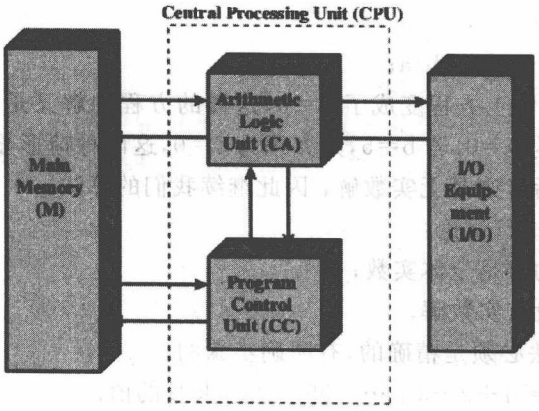

图 1-2 冯·诺依曼计算机模型

1.1.2 算法和数据结构

1. 算法(algorithm)

算法是在有限步骤内求解某一问题所使用的一组定义明确的规则。通俗点说，就是计算机解题的过程和步骤，在这个过程中，无论是形成解题思路还是编写程序，都是在实施某种算法，前者是推理实现的算法，后者是操作实现的算法。

一个算法应该具有以下五个重要的特征：

(1)有穷性：是指解决问题的过程能在一定的时间内结束，不能无限进行下去。因此，在设计算法时，我们必须确定一个条件终止算法执行。

(2)确定性：是指算法包含的每一个步骤都必须是确定的，无二义性。

(3)可行性：是指算法包含的每一个步骤都能够在计算机上有效地被执行，并得到正

确的结果。

(4)有零个或多个输入:任何一个算法被执行时,都离不开原始数据。获取原始数据有两种方法,可以直接设定一些值,也可以通过相应的设备输入数据。

(5)有一个或多个输出:我们利用计算机处理问题的最终目的就是求得某个问题的正确结果。这个结果必须输出让用户可见,才能最终确定是否正确。因此,一个没有输出结果的算法或程序是没有任何实际意义的。

下面通过几个例子来说明算法的特性。

【例 1-1】 编制算法计算下列式子的值。

(1)s=1+2+3+…+100

(2)s=1+2+3+…+100+…

(3)s=1+2+3+…+n(n≥1,且 n∈N)

能设计算法求解的只有 1 式和 3 式,2 式很显然不满足算法的有限性。

【例 1-2】 写一个算法让计算机来解方程:ax+b=0 ,其中参数 a、b 由键盘任意输入,由计算机输出结果。

对于该问题,我们不能简单地让计算机输出"x=-b/a"的结果,因为当 a=0 时,这种输出是错误的。所以,我们需要分情况讨论:

(1)输入 a,b;

(2)若 a≠0,则输出 x=-b/a;

如果 a=0 呢?实际上方程变成了 b=0,这样的方程的解又是什么呢?还要讨论参数 b,若 b=0,则方程为 0=0;若 b=5,则方程为 5=0,这两种情形显然是不一样的,前者的解是任意实数,而后者则是无实数解,因此继续我们的算法:

(3)若 a=0,对 b 进行讨论:

①若 b=0,方程的解是全体实数;

②若 b≠0,方程没有实数解。

可见一个好的算法必须是精确的,有明确步骤的。

【例 1-3】 已知 S=1+2+3+…+99+100,求 S 的值。

高斯的解法:$S=(1+100)+(2+99)+(3+98)+\cdots+(50+51)$

$=101+101+101+\cdots+101$

$=50\times101$

$=5050$

普通的解法:$S_1=1$,

$S_2=S_1+2=3$,

$S_3=S_2+3=6$,

…

$S_{99}=S_{98}+99=4950$,

$S=S_{100}=S_{99}+100=5050$

众所周知,高斯的解法比较优秀。但如果用计算机来求解上面的问题,高斯的解法中两数求和时每一次的操作数都不相同,还需要进一步加工。而普通解法只是完成一个

简单的累加过程，这种重复的事让计算机做起来很简单，所以其算法的复杂度较低。

2. 描述算法的工具

为了表示一个计算机算法，我们可以使用不同的工具。描述算法的工具有：自然语言、流程图、N-S 图、伪代码、PAD 图、计算机语言和 UML 图等。这一节主要介绍流程图和 N-S 图。

(1)流程图

流程图是一种利用图框来表示各种操作的传统算法描述工具。美国国家标准化协会 ANSI 规定了一些常用的流程图符号(见图 1-3)。

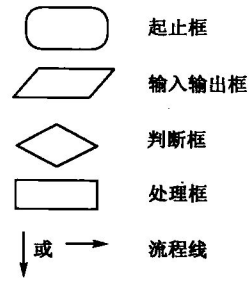

图 1-3 流程图基本符号

结构化程序设计的算法由三种基本结构构成，它们是顺序结构、选择结构和循环结构。因此，只要能够用流程图描述这三种结构，具体的算法对应的流程图也就可以很容易画出。如图 1-4、图 1-5、图 1-6、图 1-7 所示是常用的程序结构对应的流程图。

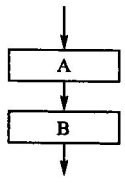

图 1-4 顺序结构流程图

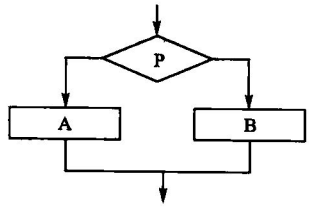

图 1-5 选择结构流程图

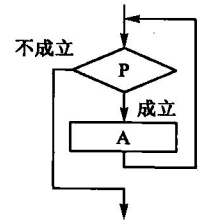

图 1-6 循环结构流程图 1

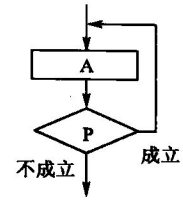

图 1-7 循环结构流程图 2

流程图描述算法的优点是简明直观、易于理解。

【例 1-4】 输入两个整数 a,b,输出较大的数。

计算过程：

①输入两个整数，分别为 a 和 b。

②比较。如果 a>b，则输出 a，否则输出 b。

该算法的流程图描述如图 1-8 所示。

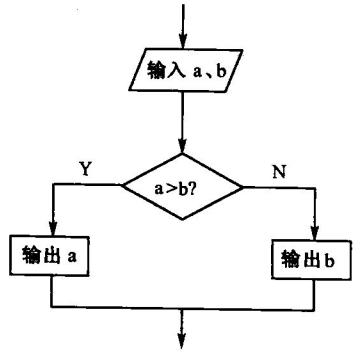

图 1-8 例 1-4 的流程图

(2)N-S图

为了避免流程图在描述程序时的随意跳转，1973年由美国人Nassi和Shneiderman提出了用方框图代替流程图，即N-S图。它采用图形的方法描述处理过程，全部算法写在一个大的矩形框中，框内包含若干个基本处理框，没有指向箭头，所以N-S图又称为盒图。利用N-S图描述三种基本流程结构的形式如图1-9、图1-10、图1-11和图1-12所示。

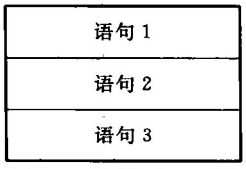

图1-9 顺序结构N-S图

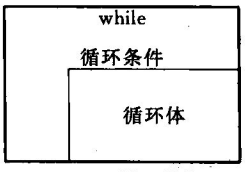

图1-10 while循环结构N-S图

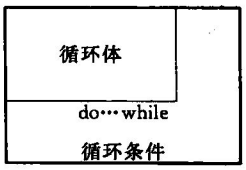

图1-11 do-while循环结构N-S图

N-S图描述算法的优点是形象直观、可读性强，限制了随意的控制转移。例1-4的N-S图描述如图1-13所示，请读者与图1-8比较。

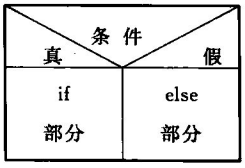

图1-12 选择结构N-S图

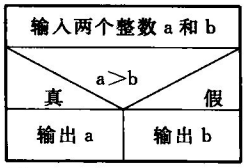

图1-13 例1-4的N-S图

3. 数据结构(Data Structure)

在计算机发展的初期，人们使用计算机的目的主要是处理数值计算问题。当我们使用计算机来解决一个具体问题时，一般需要经过下列几个步骤：首先要从该具体问题抽象出一个适当的数学模型，然后设计或选择一个解此数学模型的算法，最后编出程序进行调试、测试，直至得到最终的解答。例如，求解梁架结构中应力的数学模型的线性方程组，该方程组可以使用迭代算法来求解。

由于最初所涉及的运算对象是简单的整型、实型或布尔型数据，所以程序设计者的主要精力集中于程序设计的技巧上，而无须重视数据结构。随着计算机应用领域的扩大和软、硬件的发展，非数值计算问题变得越来越重要。据统计，当今处理非数值计算性问题占用了90%以上的机器时间。这类问题涉及的数据结构更为复杂，数据元素之间的相互关系一般无法用数学方程式加以描述。因此，解决这类问题的关键不再是数学分析和计算方法，而是要设计出合适的数据结构，才能有效地解决问题。

【例1-5】 建立学生信息检索系统。

当我们需要查找某个学生的有关情况的时候，或者想查询某个专业或年级的学生的有关情况的时候，只要我们建立了相关的数据结构，按照某种算法编写了相关程序，就可以实现计算机自动检索。可以在学生信息检索系统中建立一张按学号顺序排列的学生信息表(如表1-1所示)和分别按姓名、专业、年级顺序排列的索引表。由这些表构成的文件便是学生信息检索系统的一种数学模型。

表 1-1 学生信息表

学号	姓名	系别	专业	年级
200801001	张力	计算机科学系	软件工程	08级
200801002	李平	计算机科学系	软件工程	08级
200801003	王志平	计算机科学系	软件工程	08级
…	…	…	…	…

数据结构是由数据元素依据某种逻辑联系组织起来的，对数据元素间逻辑关系的描述称为数据的逻辑结构。数据必须在计算机内存储，数据的存储结构是数据结构的实现形式，是其在计算机内的存储表示。此外，讨论一个数据结构必须同时讨论在该类数据上执行的运算才有意义。通常情况下，精心选择的数据结构可以带来具有更高运行效率或者存储效率的算法。

数据是对客观事物的符号表示，数据元素是数据的基本单位，在计算机程序中通常作为一个整体考虑。一个数据元素由若干个数据项组成，数据项是数据不可分割的最小单位。

数据元素相互之间的关系称为结构，基本结构有四种：集合结构、线性结构、树形结构、图状结构(网状结构)。

数据结构在计算机中的表示(映象)称为数据的物理(存储)结构，它包括数据元素的表示和关系的表示。数据元素之间的关系有两种不同的表示方法：顺序映象和非顺序映象，并由此得到两种不同的存储结构：顺序存储结构和链式存储结构。顺序存储方法是把逻辑上相邻的结点存储在物理位置相邻的存储单元里，结点间的逻辑关系由存储单元的邻接关系来体现，由此得到的存储表示称为顺序存储结构。顺序存储结构是一种最基本的存储表示方法，通常借助于程序设计语言中的数组来实现。链式存储方法不要求逻辑上相邻的结点在物理位置上亦相邻，结点间的逻辑关系是由附加的指针字段表示的，由此得到的存储表示称为链式存储结构。链式存储结构通常借助于程序设计语言中的指针类型来实现。

逻辑上可以把数据结构分成线性结构和非线性结构。线性结构的顺序存储结构是一种随机存取的存储结构，线性结构的链式存储结构是一种顺序存取的存储结构。线性结构采用链式存储表示时所有结点之间的存储单元地址可连续可不连续。逻辑结构与数据元素本身的形式、内容、相对位置、所含结点个数都无关。

算法的设计取决于数据(逻辑)结构，而算法的实现依赖于采用的存储结构。数据的运算是在数据的逻辑结构上定义的操作算法，如检索、插入、删除、更新、排序等。

1.2 程序设计语言

计算机程序设计语言的发展经历了从机器语言、汇编语言到高级语言的过程。如果要自行编制程序需要使用某种程序设计语言。“语言”是交流思想的工具，人们要和计算机打交道，要指挥计算机，就需要找到一种双方都能理解的“语言”。

我们知道，计算机内存储器中存储的所有信息都是二进制形式的代码，指令也是以二进制代码的形式存储的。对于某种型号的CPU，它能够执行的所有指令的集合就是它的指令系统，也称为机器语言。机器语言全部由0和1组成，难学、难记、难读，只有少数专业人员可以使用。

1. 机器语言

电子计算机所使用的是由“0”和“1”组成的二进制数，二进制是计算机语言的基础。计算机发明之初，人们只能用计算机的语言去命令计算机干这干那，一句话，就是写出一串串由“0”和“1”组成的指令序列交由计算机执行，这种语言，就是机器语言。由于计算机的指令系统往往各不相同，所以，在一台计算机上执行的程序，要想在另一台计算机上执行，必须另编程序，这造成了重复工作。但由于机器语言使用的是针对特定型号计算机的语言，故而其运算效率是所有语言中最高的。机器语言是第一代计算机语言。

【例1-6】 完成100和256相加功能的8086CPU的代码序列如下：

```
10111000 01100100 00000000
00000101 00000000 00000001
10100011 00000000 00100000
```

对应的十六进制形式表达为：B8 64 00 05 00 01 A3 00 20

2. 汇编语言

为了减轻使用机器语言编程的负担，人们进行了一种有益的改进：用一些简洁的英文字母、符号串来替代特定指令的二进制串，比如，用“ADD”代表加法，“MOV”代表数据传递等等。这样一来，人们很容易读懂并理解程序在干什么，纠错及维护都变得方便了。这种程序设计语言就称为汇编语言，即第二代计算机语言。然而计算机是不认识这些符号的，这就需要一种程序，专门负责将这些符号翻译成由二进制数表示的机器语言，这种翻译程序被称为汇编程序。

汇编语言同样十分依赖机器硬件，移植性不好，但效率仍十分高，针对计算机特定硬件而编制的汇编语言程序，能准确发挥计算机硬件的功能和特长，程序精炼而质量高，所以至今仍是一种常用且强有力的软件开发工具。

【例1-7】 实现100与256相加的MASM汇编语言程序段表达如下：

```
mov ax,100          ;取得一个数据100(对应机器代码:B8 64 00)
add  ax,256         ;实现100+256(对应机器代码:05 00 01)
mov [2000h],ax      ;保存和(对应机器代码:A3 00 20)
```

3. 高级语言

人们逐渐意识到应该设计一种这样的语言：这种语言接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用。经过努力，1954年，第一个完全脱离机器硬件的高级语言——FORTRAN问世了。50多年来，共有几百种高级语言出现，有重要意义的有几十种，影响较大、使用较普遍的有FORTRAN、ALGOL、COBOL、BASIC、LISP、SNOBOL、PL/1、Pascal、C、PROLOG、Ada、C++、VC、VB、Delphi、JAVA等。

【例1-8】 实现100与256相加的C语言程序段表达如下：

此为试读，需要完整PDF请访问：www.ertongbook.com