高等学校计算机应用规划教材

C 语言程序设计基础教程

李少芳　张　颖　编著

清华大学出版社

北　京

内 容 简 介

本书共 9 章，分别介绍了 C 语言概述、基本数据类型与运算、结构化程序设计、数组、函数、指针、结构体和共用体、文件、面向对象基础等内容。各章从易到难给出丰富的教学案例，并配有课后习题。书中例题代码均已在 Dev C++开发环境下调试并能正常运行。

本书配有《C 语言程序设计习题与实验指导》辅导教材(ISBN 978-7-302-55790-6)，针对本书各章内容设计了上机实验和配套习题，帮助学生了解自己对内容的掌握程度。此外，辅导教材还提供了多套模拟试卷和一个详细的课程设计报告范例，方便学生自测学习效果，指导学生撰写课程设计报告。

本书是 C 语言程序设计编程入门教科书，既可以作为高等学校计算机及相关专业师生 C 语言课程的教学用书，也可以供学习 C 语言的读者自学使用。

图书在版编目(CIP)数据

C 语言程序设计基础教程 / 李少芳，张颖 编著. －北京：清华大学出版社，2020.6
高等学校计算机应用规划教材
ISBN 978-7-302-55694-7

I. ①C… II. ①李… ②张… III. ①C 语言－程序设计－高等学校－教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2020)第 103154 号

责任编辑：王　定
封面设计：高娟妮
版式设计：孔祥峰
责任校对：马遥遥
责任印制：杨　艳

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：三河市国英印务有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：16.25　　**字　　数**：395 千字
版　　次：2020 年 8 月第 1 版　　**印　　次**：2020 年 8 月第 1 次印刷
定　　价：48.00 元

产品编号：083639-01

本书编委会

编委：(按姓氏拼音排列)

陈庆枝	陈志辉	陈淑清	陈　贞
黄朝辉	黄　海	黄淋云	李海霞
李少芳	刘剑武	罗艳霞	佘玉萍
沈　林	王智明	吴珍发	谢　莹
许朝阳	许荣斌	严　涛	张　颖
郑继绍	郑　鹏	周　超	

前　言

本书是C语言程序设计编程入门教材，适用于理工类学生程序设计能力的培养。学习编程，首先要学习数据类型、控制结构、语法规则等编程入门基础知识，然后学会程序分析，认识算法在编程中的重要性。通过循序渐进地阅读、分析程序，多看参考书和现有程序，从模仿简单程序开始，掌握常用算法程序模块，逐渐看懂并学会复杂编程。

C语言程序设计是一门实践性很强的课程，平时要重视上机操作，切实掌握程序调试技术。本书详细介绍了C语言编程入门知识，使初学者能够在有限的学时内掌握C语言程序设计的基本技能，学会编写规范、可读性好的C语言程序，快速有效地掌握C语言程序设计方法。本书在教学内容和教学案例设计上，对易错、易漏的知识给予强调，并配有例题讲解。

本书共9章，分别介绍了C语言概述、基本数据类型与运算、结构化程序设计、数组、函数、指针、结构体和共用体、文件、面向对象基础等内容。各章从易到难给出丰富的教学案例，并配有课后习题。书中例题代码均已在Dev C++开发环境下调试并能正常运行。

本书配有《C语言程序设计习题与实验指导》辅导教材(ISBN 978-7-302-55790-6)，针对本书各章内容设计了上机实验和配套习题，帮助学生了解自己对内容的掌握程度。此外，辅导教材还提供了多套模拟试卷和一个详细的课程设计报告范例，方便学生自测学习效果，指导学生撰写课程设计报告。

本书是C语言程序设计编程入门教科书，既可以作为高等学校计算机及相关专业师生C语言课程的教学用书，也可以供学习C语言的读者自学使用。

本书由李少芳和张颖编著，具体分工如下：第3~5章由张颖编写，其他章节由李少芳编写，全书由李少芳统稿。本书的成功出版离不开莆田学院和清华大学出版社的大力支持和鼓励。在文稿组织、案例选择以及实验的设计与验证上得到莆田学院信息工程学院“C语言程序设计”课程组和“程序设计基础(C/C++)”课程组各位同事的鼎力帮助，在此一并表示衷心的谢意。

由于编写时间仓促，书中难免有不足之处，欢迎读者批评指正。

本书提供课件、实例源文件、习题参考答案、下载地址如下：

课件　　实例源文件　　习题参考答案

编　者

2020年3月于莆田学院

目　录

第 1 章

C语言概述

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。

(1) 机器语言是计算机最原始的语言，由 0 和 1 的代码构成，CPU 在工作的时候只认识机器语言，即 0 和 1 的代码。

(2) 汇编语言是一种低级语言，它用人类容易记忆的语言和符号来表示一组 0 和 1 的代码，例如 AND 代表加法。

(3) 高级语言是在低级语言的基础上，采用接近于人类自然语言的单词和符号表示一组低级语言程序，使编程变得更加简单、易学，且写出的程序可读性强。

高级语言又分为面向过程的编程语言和面向对象的编程语言。面向过程的编程每实现一次功能都要编写一次代码，代码的重用性较差。而在面向对象的编程中引入了类的概念，实现同样的方法只要编写一次代码，用到时只需要调用该类即可，代码重用性高，这是目前流行的编程方式。C语言是一门面向过程的高级语言，C++是面向对象程序的设计语言，同时也可以进行基于过程的程序设计。

本章介绍 C 语言的发展历史及特点、算法、C 语言程序的基本结构、C 语言程序的编译与运行、C/C++开发环境 Visual C++ 6.0 和 Dev C++的使用方法。

【学习目标】

1. 了解 C 语言的发展历史、程序设计中算法的重要性。
2. 掌握和理解 C 语言程序的特点和基本结构。
3. 熟悉 C/C++开发环境，以及 C 语言程序的编译、连接和运行的过程。

【重点与难点】

通过运行简单的 C 语言程序，掌握 C 语言程序的编译、连接和运行的过程。

1.1 C 语言的发展历史及特点

C 语言是一门面向过程、抽象化的通用程序设计语言，广泛应用于底层开发。本节主要介绍程序与软件的概念、C 语言的发展历史、C 语言的特点。

1.1.1 程序与软件

作为一种能自动计算的机器，计算机通过执行一系列指令来完成给定的计算工作。因此，要让计算机完成某项任务，就必须将完成这项任务的方法和具体步骤编写成计算机可以直接或间接执行的一系列指令，使之在执行这些指令后就可以完成给定的任务。这样的一系列指令的集合称为计算机程序，简称程序(Program)，编写这些指令的工作就是程序设计(Programming)。

程序是为使计算机执行一个或多个操作，或者执行某一任务，按序设计的计算机指令的集合。

程序设计给出解决特定问题程序的方法和过程，是软件构造活动的重要组成部分。程序设计过程包括“需求分析→设计→编码→测试→维护”5 个阶段(见图 1-1)，并生成各种文档资料。程序设计最终需要以某种程序设计语言为工具，编写出该程序的语言。

对 C/C++语言来说，源程序文件的扩展名为.c/cpp，可执行程序的文件的扩展名为.exe。

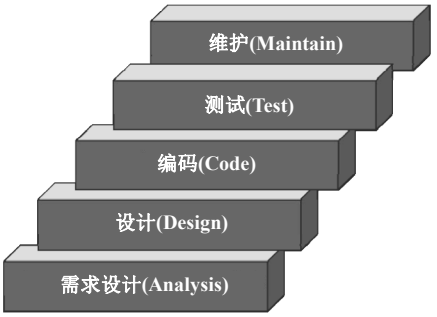

图 1 -1　程序设计过程的 5 个阶段

软件(Software)是一系列按照特定顺序组织的计算机数据和指令的集合。简单地说，软件就是程序加文档。软件并不只包括可以在计算机上运行的电脑程序，与这些电脑程序相关的文档一般也被认为是软件的一部分。根据软件的功能，软件可以分为系统软件和应用软件。

1.1.2 C 语言的发展历史

第一台按冯・诺依曼原理制成的通用电动计算机是 1951 年美国兰德公司的 UNIVAC-1，其采用机器语言进行程序设计，即数据和指令(存储地址码、操作码)都以二进制编码输入。实际程序用八进制和十六进制数，输入后是二进制的。单调的数字极易出错，于是将操作码改作助记的字符，这就是汇编语言，如用 ADD A,B 表示两数相加。机器语言和汇编语言属于低级语言，其编码依赖于所使用的计算机硬件，与特定的机器有关，执行速度快，但编写复杂、费时，容易出差错，而且程序修改维护困难。

高级语言的表示方法比较接近于自然语言，在一定程度上与具体的计算机硬件及其指令系统无关，可阅读性更强，相对来说更易于学习和掌握，也便于维护，但是其代码的执行速度比低级语言慢。1954 年出现第一个完全脱离机器的高级语言 Fortran I，1957 年其被改进为 Fortran II。Fortran 语言主要用于数值计算，是进行大型科学和工程计算的重要工具。1958 年出现算法语言 Algol 58，改进版有 Algol 60、结构化的 Algol W 和 Algol 68。Algol 语言是程序设

计语言的开拓者，为软件自动化和可靠性研究奠定了基础。1960 年出现 Cobol 60 语言。由于其优异的输入/输出功能，报表、分类归并方便快速，所以该语言牢固占领商用事务软件市场。直到今天它在英语国家的商业领域还有一定的地位。早期的软件市场在计算机应用上可以说是由 Fortran、Cobol 和汇编三分天下，在科学计算上有 Fortran，在商用事务处理方面有 Cobol，在工程控制方面有汇编语言。1971 年出现第一个结构化程序设计语言 Pascal，1975 年丹麦学者汉森(B.Hanson)开发了并发 Pascal。Pascal 在结构化程序设计方面是一个示范性语言，在推行结构化程序设计教学上发挥了卓越的作用，但在工程实践上暴露出其在设计上的诸多缺点。

C 语言是一种高效的编译型结构化程序设计语言。C 语言的诞生可以追溯到 Algol 语言(1960 年)→CPL 语言(1963 年，英国剑桥大学)→BCPL 语言(1967 年，剑桥大学马丁・理查德)→B 语言(1970 年，美国贝尔实验室肯・苏姆普逊)→C 语言(1972 年，丹尼斯・瑞奇和布朗・卡尼汉；1983 年，ANSI C；1990 年，ISO C 等)。

C 语言最早是由美国贝尔实验室的丹尼斯・瑞奇(Dennis M. Ritchie)在 B 语言的基础上开发出来，并于 1972 年在一台 Decpdp-11 计算机上首次实现。C 语言的设计初衷是为描述和实现 Unix 操作系统提供一种工作语言，作为计算机专业人员写 UNIX 操作系统的一种工具，在贝尔实验室内部使用。1973 年，肯・苏姆普逊(Ken Thompson)和丹尼斯・瑞奇两人合作，把 UNIX 系统 90%以上的内容用 C 语言改写，即 UNIX 第 5 版。随后几年，贝尔实验室又对 C 语言进行了多次改进，但仍局限在内部使用。直到 1975 年 UNIX 第 6 版公布后，C 语言的突出优点才引起人们的普遍注意。1977 年出现了不依赖具体机器的 C 语言编译文本《可移植 C 语言编译程序》，使 C 语言移植到其他机器时所需做的工作大大简化，同时也推动了 UNIX 系统迅速地在各种机器上实现。1978 年，布朗・卡尼汉(Brian W. Kernighan)和丹尼斯・瑞奇合作出版了名著《C 程序设计语言》(*The C Programming Language*)。此书被翻译成多种语言，成为 C 语言最权威的教材之一。1983 年，美国国家标准化协会(ANSI)根据 C 语言问世以来各种版本，对 C 语言的发展和扩充制订了一套 ANSI 标准，称为 ANSI C。1987 年，ANSI 又公布了新标准 87ANSI C。1990 年，国际标准化组织 ISO 接受 87ANSI C 为 ISO C 的标准(ISO9899 1990)。C 语言的发展历史简图，如图 1-2 所示。

从第一个高级语言 Fortran 问世至今，已经有数百种高级语言出现。高级语言的发展经历了从面向过程程序设计(Procedure Oriented Programming，POP)到面向对象程序设计(Object Oriented Programming，OOP)，从字符方式到可视化的过程。随着软件规模的增大，Fortran、Basic、Pascal、C 语言等面向过程的编程语言已经无法满足运用面向对象方法开发软件的需要。20 世纪 70 年代末 80 年代初，为了解决一些非过程性的问题以及在大规模软件开发中软件的维护与管理问题，有人提出面向对象的程序设计思想。面向对象的程序设计是一种基于结构分析、以数据为中心的程序设计方法。面向对象的程序设计方法更加抽象，但程序更加清晰易懂，更适合大规模程序的编写。C++、Java 等面向对象程序设计语言的出现，使软件开发也逐渐变成了有规模、有产业的商业项目。

可视化程序设计语言，即图形界面的程序设计语言，如 Visual C++、Visual Basic、Delphi 等，是在 Windows 操作系统出现以后发展起来的。C++语言与 C 语言相兼容，运行性能高，又有数据抽象和面向对象的特性，是当前面向对象程序设计的主流语言。目前，常用的 C/C++语

言开发软件有 Turbo C(简称 TC 1987，POP)、Turbo C++(1990 年，OOP)、Borland C++(1991 年)、Win-TC、Dev C++、Visual C++(可视化，简称 VC，20 世纪 90 年代末，OOP)和 Visual C++.NET(C#)。其中，VC 语言既可以使用 Windows 图形用户界面，又可以调用 Windows 的其他资源。Visual C++.NET(C#)为 Windows 和 Web 应用程序提供动态开发环境。

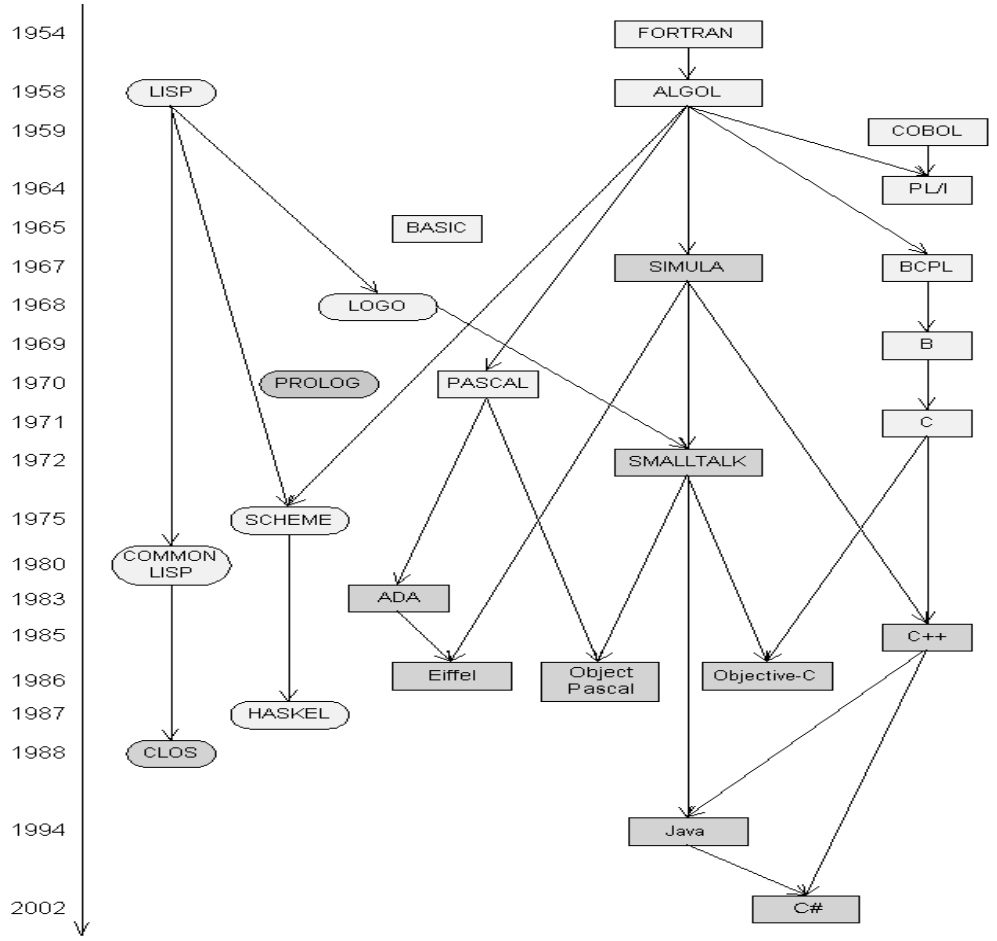

图 1-2　C 语言的发展历史简图

1.1.3　C 语言的特点

1. C 语言的优点

C 语言是国际上流行的计算机高级语言，它被广泛应用于系统软件和应用软件的编写，是公认的最重要的几种编程语言之一，被称作“低级语言中的高级语言，高级语言中的低级语言”。其优点如下。

(1) 语言简洁紧凑，使用方便灵活。关键字是编程语言里事先定义好并赋予了特殊含义的单词，对语言编译器有特殊的意义，用来表示一种数据类型或者表示程序的结构。在 ANSI C 标准中，C 语言共有 32 个关键字，如表 1-1 所示。

C 语言共有 9 种控制语句，如表 1-2 所示。程序书写自由，主要用小写字母来表示。

表 1-1　C 语言的 32 个关键字

序号	关键字	说明	序号	关键字	说明
1	auto	自动变量	17	static	静态变量
2	short	短整型	18	volatile	变量在程序执行中可被隐含地改变
3	int	整型	19	void	空类型
4	long	长整型	20	if	条件语句
5	float	浮点型	21	else	条件语句否定分支(与 if 连用)
6	double	双精度	22	switch	开关语句
7	char	字符型	23	case	开关语句分支
8	struct	结构体	24	for	一种循环语句
9	union	共用体	25	do	循环语句的循环体
10	enum	枚举类型	26	while	循环语句的循环条件
11	typedef	给数据类型取别名	27	goto	无条件跳转语句
12	const	只读变量	28	continue	结束当前循环，开始下一轮循环
13	unsigned	无符号类型	29	break	跳出当前循环
14	signed	有符号类型	30	default	开关语句中的其他分支
15	extern	外部存储变量	31	sizeof	计算数据类型长度
16	register	寄存器变量	32	return	返回

表 1-2　C 语言的 9 个控制语句

序号	控制语句	说明
1	if 语句	条件选择语句
2	switch 语句	开关分支语句
3	while 语句	当型循环语句
4	do while 语句	直到型循环语句
5	for 语句	计数循环语句
6	continue 语句	中止本次循环语句
7	break 语句	终止执行 switch 或循环语句
8	goto 语句	无条件转移语句
9	return 语句	函数返回语句

(2) 数据类型和运算符丰富多样。C 语言内置了丰富的运算符，共有 34 种运算符。

(3) 可移植性强。可移植性是指程序可以从一个环境下不加修改或稍加修改就可移到另一个完全不同的环境下运行。想跨越平台来执行 C 语言，通常只要修改极少部分的程序码，再重新编译即可执行。据统计，不同机器上的 C 编译程序 80%的代码是相同的。

(4) 高效率的编译式语言，生成的目标代码质量好，程序执行效率高。C 语言通过编译器(Compiler)将整个程序码编译成机器码，然后执行，其执行速度远比解释型(Interpreter)快。解释型程序，如 BASIC 程序，边解释边执行，虽然占用的存储器较少，但执行的速度会变慢，效率较低。另外，C 语言允许直接访问物理内存，进行位操作，具有低级语言的许多功能，程序执行效率高。

(5) 介于高级与低级之间的语言。低级语言，如汇编语言与机器语言，适合计算机阅读；高级语言贴近人类语言习惯，如 BASIC，适合人类阅读。C 语言兼具低级与高级语言的优点与特色，是介于高级与低级之间的语言，所以也有人称它为中级语言。

C++与 Java 均是以 C 语言为基础，再加上面向对象程序设计(OOP)技术，使得它们活跃于 Windows 可视化程序设计与网络程序设计中。Flash 的 ActionScript 的语法与 C/C++也非常接近。

2. C 语言的局限性

C 语言并不是完美无缺的，在流行的同时也暴露出了它的局限性。

(1) C 语言类型自检机制较弱，使得程序中的一些错误不能在编译时被发现。

(2) C 语言本身缺乏支持代码重用的机制，使得各个程序的代码很难为其他程序所用。

1.2 算法概述

著名瑞士计算机科学家沃思(Nikiklaus Wirth)曾提出一个公式“程序=数据结构+算法”，并因此成为 1984 年图灵奖得主。目前这个公式已经修改为：

程序=算法+数据结构+程序设计方法+语言工具和环境

其中，数据结构主要是数据的类型和数据的组织形式，是对程序中数据的描述；算法则是对程序中操作的描述，也就是操作步骤。

1.2.1 算法的概念

算法(Algorithm)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。计算机算法是用计算机求解一个具体问题或执行特定任务的一组有序的操作步骤(或指令)，是构成计算机程序的核心部分。通俗点说，算法就是计算机解题的过程。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。

例如，要求计算圆的面积，算法为：

(1) 输入或指定半径值 r。

(2) 使用公式 $S=\pi r^2$。

(3) 输出显示 S 的值。

程序是利用计算机程序设计语言设计出的能够在计算机上运行，并且能够解决实际问题的工具。一个程序应该包括两方面的工作内容：一是对数据进行合理的组织，即在程序中要指定数据的类型和数据的组织形式，即数据结构(Data structure)；二是设计解决问题的算法，即操作步骤。

用计算机求解问题，必须编写求解问题的程序，可以通过以下两种途径获得：一是利用现有的程序；二是自己编程。编程的依据是求解问题的算法。

算法是程序设计的精髓，程序设计的实质是构造解决问题的算法，将其解释为计算机语言。算法是数据结构、算法设计、数学以及与问题有关的知识的综合应用。算法设计是计算机工作者，特别是计算机专业教师和学生必备的基本功。

1.2.2　算法的特性

算法有以下 5 个重要特征。

(1) 有穷性：任何算法都应该在执行有穷步骤之后结束。

(2) 确定性：算法的每一步骤必须有确切的定义，不能具有二义性。算法中每一步的语义都应该清晰明了，明确指出应该执行什么操作，如何执行操作。

(3) 可行性：算法原则上能够精确地运行，而且人们用笔和纸做有限次运算后即可完成。根据算法编写出来的程序应具有较高的时空效率，执行时间短，不占用过多内存。

(4) 有零个或多个输入：算法可以有零个或多个输入，用来刻画运算对象的初始情况。

(5) 有一个或多个输出：算法必须具有一个或多个执行结果的输出，用来反映对输入数据加工后的结果。没有输出的算法是毫无意义的，是一个无效算法。

1.2.3　算法的表示

描述算法的常见方式有以下几种。

(1) 自然语言表示：易理解和交流，但易产生二义性。

(2) 伪代码表示：伪代码使用介于自然语言和计算机语言之间的文字和符号来描述算法。

(3) 程序流程图：用图形符号和文字说明表示数据处理的过程和步骤。流程图所使用的图形如图 1-3 所示，图形解释如表 1-3 所示。

(4) N-S 流程图：也称方框图，适于结构化程序设计的算法描述工具。

传统的程序流程图由一些特定意义的图形、流程线及简要的文字说明构成，它能明确地表示算法的运行过程，是描述算法的良好工具。

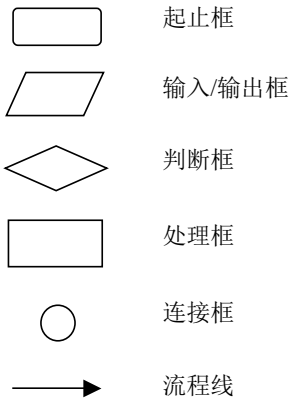

图 1-3　流程图所使用的图形

表 1-3　流程图的图形解释

图形	名称	图形含义
圆角矩形	起止框	表示算法开始或结束的符号
平行四边形	输入/输出框	表示算法过程的信息输入和输出
菱形	判断框	表示算法过程中的选择分支结构，框内注明判断条件，决定程序的流向，通常上顶点表示入口，其余的顶点表示出口
矩形	处理框	表示算法过程中需要处理的内容或程序段，框内用文字简述其功能。只有一个入口和一个出口
圆形	连接框	框内注有字母，当流程图跨页或出现流向线交叉时，用它表示彼此之间的关系，相同符号的连接框表示它们是相互连接的
箭头线	流程线	表示算法流程的方向，以单向箭头表示

举一个生活中的例子：如果下雨，则带伞，否则戴太阳眼镜；不管是否下雨，最后都要出门。图 1-4 给出对应的出门事件的流程图。传统流程图的一个主要不足是流程线的用法缺乏规范。由于流程线可以转移流程的执行方向，如果使用不当或流程控制转移不明晰，容易导致程序的混乱和出错。

1973 年，美国学者纳斯(I.Nassi)和施内德曼(B.Schneiderman)提出 N-S 流程图(也称为方框图)，它没有使用流程线，而是把整个算法写在一个大框图内，这个大框图由若干个小的基本框图构成，算法按照从上到下、从左到右的顺序执行。

需要注意的是，算法一般只是对处理问题思想的一种描述，不是计算机可以直接执行的程序代码。因此，算法本身是独立于计算机的，算法的具体实现则由计算机完成。从这个意义上说，程序设计的本质就是要将算法转化为计算机程序。处理一个问题，可以有不同的算法。设计和选择算法是至关重要的，不仅要保证算法正确，还要考虑算法的质量和效率。

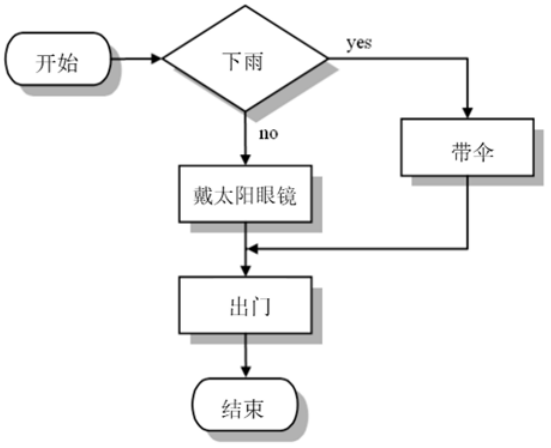

图 1-4　出门事件的流程图

结构化程序设计的观点认为，任何算法功能都可以通过 3 种基本控制结构以及它们的嵌套组合来实现，这 3 种结构分别是顺序结构、选择(分支)结构和循环结构。N-S 流程图是一种适于结构化程序设计的算法描述工具。顺序结构的流程图如图 1-5 所示，双分支选择结构的流程图如图 1-6 所示，当型和直到型循环结构的流程图如图 1-7 和图 1-8 所示。

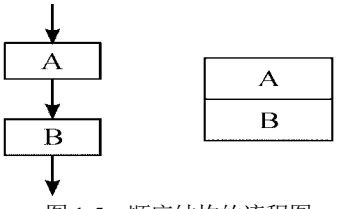

图 1-5　顺序结构的流程图

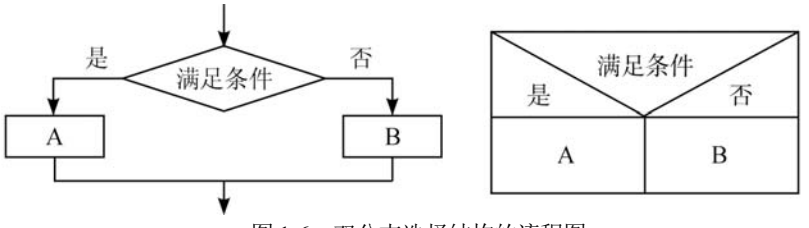

图 1-6　双分支选择结构的流程图

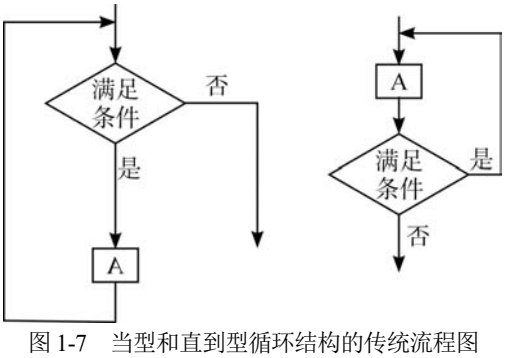

图 1-7　当型和直到型循环结构的传统流程图

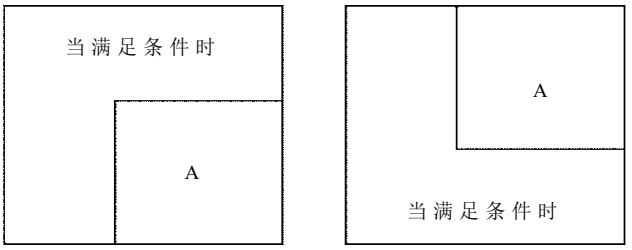

图 1-8　当型和直到型循环结构的 N-S 流程图

1.3　C 语言程序的基本结构

一个 C 语言程序由一个固定名称为 main 的主函数和若干个其他函数(可没有)组成。下面通过几个例题，总结出 C 语言程序的结构特点。在 Dev C++环境下编写的第一个 C 语言如下。

【例 1-1】在屏幕上输出 Hello everyone!。

```
#include <stdio.h>                  /*编译预处理命令文件包含*/
int main()                          /*主函数*/
{
    printf("Hello everyone!\n");    /*在屏幕上输出语句*/
    return 0;                       /*主函数正常运行完毕，则返回 0*/
}
```

将例 1-1 编译、连接、运行后，在屏幕上输出：Hello everyone!

1. 文件包含

文件包含的格式：

```
#include <stdio.h>
```

C 语言是一种“装配式”语言，许多常规的工作如输入、输出、数学函数等，往往事先由程序员做成各种程序模块，存放在各种头文件(.h)中。

文件包含的作用是根据需要把相应的某个头文件的内容在编译时先整体嵌入所编写的程序中。用户也可以将自己设计的程序模块等做成头文件，供其他程序包含。使用文件包含功能的优点是提高程序设计效率和程序可靠性，减少程序员重复劳动量。

Turbo C 提供了 300 多个标准库函数，存放在若干个头文件中。常用的函数如下。

- stdio.h：标准输入/输出函数。
- math.h：数学函数。
- stdlib.h：常用函数。

一个优秀的程序员不应是事无巨细、万事都要从头做起的“工匠”，而应是一个“策划师”+“组装师”。所以，逐步熟悉并掌握常用函数等现有功能模块，是学习 C 程序设计的一个重要内容。

2. 主函数

主函数的一般形式：

```
int main()
{   数据定义(变量说明语句);     /*数据结构*/
    数据处理(执行语句);         /*算法*/
}
```

说明：

(1) C 语言是一种函数式语言，C 程序的基本组成是函数。一个函数实际上就是一个功能模块。

(2) 一个 C 程序是由一个固定名称为 main 的主函数和若干个其他函数(可没有)组成。

(3) 一个 C 程序必须有一个也只能有一个 main 主函数。

(4) 主函数在程序中的位置可以任意，但程序执行时总是从主函数开始，在主函数内结束。

(5) 主函数可以调用其他各种函数(包括用户自定义函数)，但其他函数不能调用主函数。

3. 注释

在编写程序时，为了使代码易于阅读，通常会在实现功能的同时为代码加一些注释。注释是对源程序进行注解，对程序的某个功能或者某行代码的解释说明，增加程序的可读性。它只在 C 源文件中有效，对编译和运行不起作用，在编译程序时编译器会忽略这些注释信息。

C 语言中的注释有两种类型，具体如下。

(1) 单行注释格式：//注释内容

说明：符号“//”后面为被注释的内容。

例如：

```
int c=10;       //定义一个整型变量
```

(2) 多行注释格式：/*注释内容*/

说明：以符号“/*”开头，以符号“*/”结尾。

例如：

```
/*   int c=10;
int x=5;              */
```

注释可以出现在一行的最右侧，也可以单独成为一行，如果需要，程序中的任意一行都可以加上注释。

【例 1-2】输入两个整数，求两个整数的和。

```
#include <stdio.h>
int main()
{   int a,b,c;                      /*变量声明，定义整型变量 a、b、c */
    scanf("%d,%d",&a,&b);           /*输入语句，输入两个整数分别赋值给 a 和 b */
```

```
    c=a+b;                          /*计算变量 a 和 b 的和并赋值给变量 c */
    printf("c=%d\n",c);             /*输出结果*/
    return 0;
}
```

输入 1,2，则程序运行结果：c=3

4. 数据类型定义语句

数据类型定义语句格式：

```
变量类型关键字  变量名;
```

例如：

```
int a,b,c;                          //定义 a、b、c 为整型变量
float r,s;                          //定义 r、s 为单精度实型变量
```

注意：

在 C 语言程序中，所有变量都要先定义后使用，否则就会出现编译错误提示。例如:

```
Error: Undefined symbol 'a' in function main
```

5. 赋值语句

赋值语句格式：

```
变量名=常量或表达式;
```

作用：使变量获得具体的运算值。

例如：

```
r =3.0;                     /*把 3.0 赋值给变量 r*/
c=a+b;                      /*将 a 与 b 相加后的和赋值给 c*/
```

变量赋初值也可在数据类型定义时进行，如：

```
 float r =3.0;
```

6. 输出语句

输出语句格式：

```
printf(格式控制字符串,输出列表);
```

作用：普通字符在输出时按原样输出，转义字符则输出它所代表的字符。

例如：

```
printf("Hello,everyone!\n");   /*引号中的字符原样输出*/
```

其中，\n 是转义字符，代表换行符，即表示回车光标移到下一行开头处。

输出格式控制符以%开始，后面跟格式字符，用于以指定的格式输出数据。如例 1-2 中

```
printf("c=%d\n",c);
```

输出结果是 c=3，其中格式控制符%d 表示输出十进制整数，此处代替整型变量 c 的值。

以下是常用格式控制符。

- 字符型：%c 表示单字符；%s 表示字符串。
- 数值型：%d 表示整数(十进制)；%f 表示实数(小数形式，默认为 6 位小数)。

7. 输入语句

输入语句格式：

```
scanf(格式控制字符串,输入项地址列表);
```

作用：以格式控制字符串指定的格式输入数据，并存入地址列表所对应的内存中。

例如：

```
scanf("%d,%d",&a,&b);
```

其中，&是地址运算符，用于获取变量在内存中的地址。

注意：

(1) 若格式控制字符串间用逗号隔开如"%d,%d"，则输入的两个数用逗号隔开，如 1,2。

(2) 若格式控制字符串间用空格隔开，如"%d %d"，或者没有隔开，如"%d%d"，则输入的两个数都用空格或者回车隔开。

例如：

```
1 2
```

或者

```
1
2
```

【例 1-3】给定圆的半径为 3.0，求圆的面积。

```
#define PI 3.14159                          /*编译预处理——宏替换*/
#include <stdio.h>                          /*编译预处理——文件包含*/
int main()                                  /*主函数*/
{   float r,s;                              /*定义变量 r、s 类型为单精度实型*/
    r=3.0;                                  /*变量 r 赋初值*/
    s=PI*r*r;                               /*计算圆面积 s*/
    printf("r=%.2f,s=%.2f\n",r,s);          /*输出结果*/
    return 0;
}
```

程序运行结果：

```
r=3.00,s=28.27
```

8. 宏定义

宏定义格式：

```
#define 标识符 文本
```

其中，标识符就是所谓的符号常量，也称为宏名。例如：

```
#define PI 3.14159
```

其中，PI 为符号常量，即宏名，最好用大写，以区别一般变量。3.14159 为宏体，宏体也可以是一个表达式。

作用：用简单符号代表宏体部分内容(编译时会先自动替换)。

意义：直观，可多次使用，便于修改。

注意：

#define 可出现在程序的任意位置，其作用范围为由此行到程序末尾。

宏定义不是 C 语句，不必在行末加分号，否则会连分号一起置换。

9. 条件选择语句

例 1-3 中的程序有两个不足：

(1) 如果要求多个半径 r 值时的面积 s，每次都必须修改源程序并重新编译处理。

(2) 如果半径 r 为负值，也会有正常的 s 值输出。

为此，可将程序进行如下修改。

(1) 增加键盘输入函数。

键盘输入函数的格式：

```
scanf("%f",&r);   /* &r   变量 r 的存储单元地址*/
```

功能：将键盘输入的值存放到变量 r 所对应的存储单元中。scanf()函数通常与 printf()函数组合使用，实现“人机对话”功能。

(2) 增加 if 条件判断 if (r>=0)。

条件选择语句格式：

```
if (条件表达式)语句或{复合语句};
```

功能：如果条件表达式的值为真，就执行指定语句或复合语句。

扩展形式：

```
if … else 语句;
```

或

```
if  (条件表达式)  语句或复合语句;
else  语句或复合语句;
```

注意：

条件表达式必须用()括起，且不能跟分号。关于 if 语句的详细讲解见后面章节。

例 1-3 修改后的程序为：

```
#define PI 3.14159
#include <stdio.h>
int main()
{   float r;
    printf("请输入半径 r：");
    scanf("%f",&r);
    if(r>=0)
        printf("r=%.2f,s=%.2f\n",r,PI*r*r);
    else
        printf("半径不能为负数！");
    return 0;
}
```

【例 1-4】改编例 1-2 的程序，将求两个整数的和用自定义函数编写，由主函数调用。

```
#include <stdio.h>
int main()
{   int a,b,c;
    int add(int x, int y);              /*函数声明，声明本函数要调用的 add 函数*/
    scanf("%d, %d", &a, &b);            /*输入变量 a 和 b 的值*/
    c=add(a, b);                        /*调用 add 函数，将函数的返回值赋给 c*/
    printf("c=%d", c);                  /*输出 c 的值*/
}
int add(int x,int y)                    /*定义函数值为整型，形式参数 x、y 为整型的 add 函数*/
{   int z;                              /*add 函数中的声明部分，定义本函数中用到的整型变量 z*/
    z=x+y;
    return (z);                         /*返回 z 的值到该函数被调用处*/
}
```

当运行程序时输入：100,478↙(↙代表 Enter 键)。程序输出如下：

```
  c=578
```

10. 函数及函数的调用

C 语言是一种函数式语言，其程序基本组成是函数。C 语言包括库函数和用户自定义函数，

库函数是由 C 语言编译系统提供的，可以直接使用，比如 printf、scanf 函数等；而用户自定义函数是依照问题需要由用户自己编写的。

例 1-4 是由主函数 main 和用户自定义的函数 add 组成，main 函数是主调函数，add 函数是被调函数。add 函数的作用是求两个变量的和，并返回求和结果，return 语句将 z 的值返回主调函数 main 中调用 add 函数的位置并赋值给 c。程序第 4 行是对被调用函数 add 的声明，为了使编译系统能够正确识别和调用 add 函数，必须在调用 add 函数之前对 add 函数进行声明。有关函数声明将在第 5 章进一步介绍。

通过以上几个例题可以总结出以下几点。

(1) C 语言是一种函数式语言，其程序基本组成是函数。

(2) 每个 C 程序必须有一个也只能有一个主函数 main。

(3) 不管主函数在程序中的位置如何，程序执行总是从主函数开始。

(4) 所有变量必须先定义后使用。

(5) 每个语句必须用分号(；)结束(注意是“每个语句”而不是“每行语句”)。

(6) 编译预处理命令不是语句(行末不能用分号结束)。

(7) C 语言本身没有输入/输出语句，其功能须通过调用相关函数来实现。

(8) 使用系统提供的标准库函数或其他文件提供的现成函数时，必须使用“文件包含”。

【例 1-5】已知宏定义 #define SQ(x) x*x，执行语句 printf("%d",10/SQ(3));后的输出结果是_____。

A. 1　　B. 3　　C. 9　　D. 10

答案：C

分析：宏替换后的结果是 printf("%d",10/3*3)。

【例 1-6】已知宏定义

```
#define N 3
#define Y(n) ((N+1)*n)
```

执行语句 z=2*(N+Y(5+1));后，变量 z 的值是_____。

A. 42　　B. 48　　C. 52　　D. 出错

答案：B

分析：语句 z=2*(N+Y(5+1))引用了两个宏定义。C 语言是区分字母大小的，第二个宏定义中的 N 直接用 3 替换，用 5+1 替换 n，则有 z=2*(3+(3+1)*5+1)；结果是 48。注意对于带参数的宏也是直接的文本替换，此例中 n 用 5+1 去替换，结果是(N+1)*5+1，而不是(N+1)*(5+1)。

1.4 C 语言程序的编译与运行

源程序也称源代码，是指未编译的、按照一定的程序设计语言规范书写的文本文件，是一系列人类可读的计算机语言指令，可以用汇编语言或高级语言编写。计算机源代码的最终目的

是将人类可读的文本翻译成为计算机可以执行的二进制指令，这种过程叫编译，通过编译器完成。不同的程序设计语言的源程序的扩展名是不同的，例如，用 C 语言编写的源程序，其文件扩展名为.c；用 Java 语言编写的源程序，其文件扩展名为.java。

编译程序是将源程序译成能被 CPU 直接识别的目标程序或可执行指令的程序。例如，用汇编语言书写的源程序要经过汇编程序译成用机器语言表示的目标程序，用高级语言书写的源程序要经过编译程序译成用机器语言表示的目标程序。

目标程序是经编译程序翻译生成的程序，文件扩展名为.obj。

可执行程序是经连接程序处理过的程序，文件扩展名为.exe。

需要指出的是，源代码的修改不能改变已经生成的目标代码。如果需要目标代码做出相应的修改，必须重新编译。

源程序、目标程序及可执行程序的关系如图 1-9 所示。

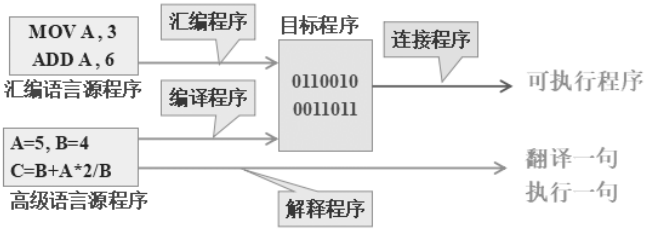

图 1-9　源程序目标程序及可执行程序的关系

C 语言是一种通过编译程序处理的高级程序设计语言，其处理流程(如图 1-10 所示)具体如下。

源程序(.c)———目标程序（.obj）———可执行程序(.exe)
　　编译　　连接

图 1-10　C 语言的处理流程

(1) 编辑 C 语言程序：当确定了解决问题的方案后，可以用 C 语言系统提供的编辑功能编写一个 C 语言源程序，源程序的扩展名为.c。

(2) 编译 C 语言程序生成目标程序：由于计算机只能识别和执行由 0 和 1 组成的二进制文件，而不能识别和执行用高级语言编写的源程序，所以必须先用 C 语言系统的编译程序(即编译器)对其进行编译，以生成以二进制代码形式表示的目标程序，目标程序的扩展名为.obj。

(3) 连接生成可执行程序文件：将目标程序与系统的函数库以及其他目标程序进行连接装配，才能形成可执行程序文件，可执行文件的扩展名为.exe。

(4) 运行可执行程序文件：将可执行程序文件(扩展名为.exe)调入内存并运行，得到程序的结果。

【例 1-7】求 3.5、4.6 和 7.9 这 3 个数的平均值。

```
#include <stdio.h>
int main()
{   float x,y,z,aver;
    x=3.5;
    y=4.6;
```

```
    z=7.9;
    aver=(x+y+z)/3;
    printf("aver=%.1f",aver);
    return 0;
}
```

程序运行结果：

```
aver=5.3
```

1.5 C/C++开发环境

C 语言编译器可以分为 C 和 C++两大类，其中 C++是 C 的超集，也支持 C 语言编程。事实上，编译器的选择不是最重要的，它们都可以完成基本的 C 语言编译。但因为编译器的编译结果存在一定差别，特别是在一些复杂语法的语句编译上，为顺应考试需求，本节主要介绍 Visual C++ 6.0 和 Dev C++的使用方法。

1.5.1 Visual C++开发环境

Visual C++(简称 VC++)是 Windows 环境下最强大、最流行的程序设计语言之一。Visual C++集成开发环境包括程序自动生成向导 AppWizard、类向导 ClassWizard、各种资源编辑器以及功能强大的调试器等可视化和自动化编程辅助工具。

在 Visual C++ 6.0 软件中，调试、连接和运行 Visual C++应用程序项目的步骤如下。

(1) 双击运行 VC++ 6.0 软件，打开如图 1-11 所示 VC++ 6.0 主窗口。

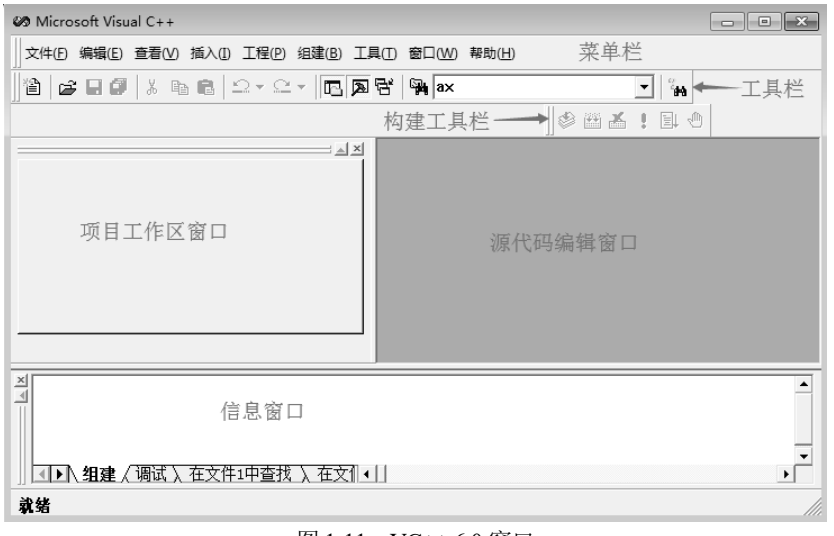

图 1-11 VC++ 6.0 窗口

(2) 创建源程序文件，选择“文件”菜单中的“新建”命令，打开“新建”对话框，单击

“文件”选项卡下的 C++ Source File，然后填写文件名，文件扩展名为.c(C 源文件)或.cpp(C++源文件)，“位置”选项选择已建好的文件夹，如图 1-12 所示。

图 1-12　“新建”对话框

(3) 单击“确定”按钮，然后在窗口中输入相应的 C 或 C++源程序代码，并保存。

(4) 单击编译工具条上的编译按钮，在如图 1-13 所示对话框中单击“是”按钮，生成工作区文件，在如图 1-14 所示的调试信息窗口中出现 L1.obj-0 error(s),0 Warning(s)，表示编译正确，生成 L1.obj 目标文件。

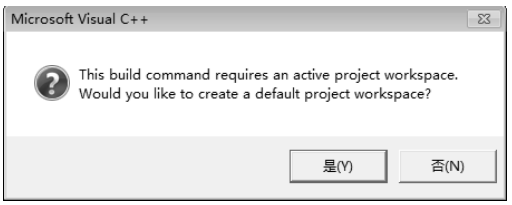

图 1-13　生成工作区提示

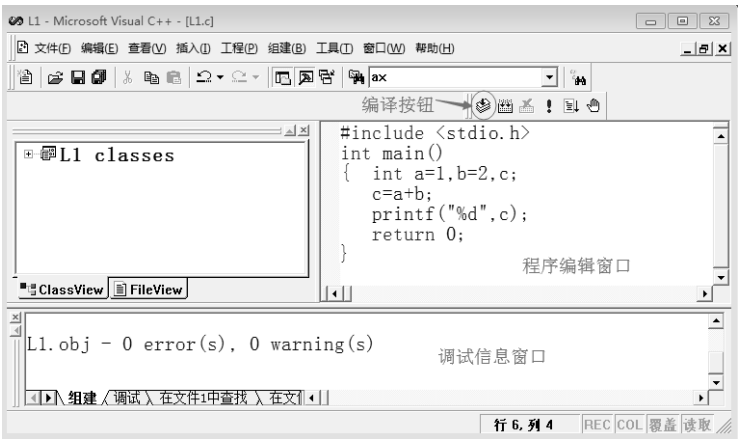

图 1-14　编译源程序

若信息窗口显示有错误 error(s)(见图 1-15)，则需要对程序进行修改。双击错误信息，光标会回到编辑窗口中错误程序所在行或附近行，修改好后再重新编译；若显示的是警告 warning(s)，

不影响生成目标文件，但也建议先修改再编译。

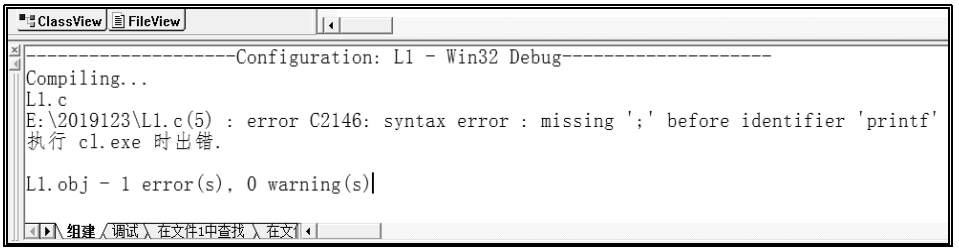

图 1-15　信息窗口显示有错误

(5) 单击编译条上的连接按钮，当信息窗口出现如图 1-16 所示的情况，表示连接成功，产生可执行文件 L1.exe。

(6) 单击编译条上的运行按钮，自动弹出运行窗口，显示运行结果或等待用户输入数据，如图 1-17 所示，然后按任意键继续返回编辑窗口。

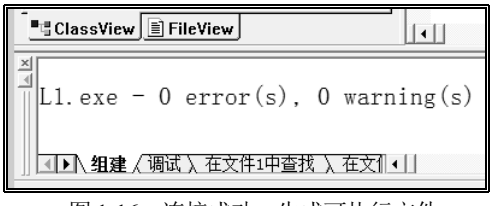

图 1-16　连接成功，生成可执行文件

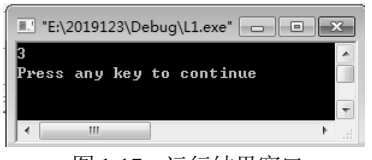

图 1-17　运行结果窗口

(7) 关闭工作空间：单击“文件”菜单下的“关闭工作空间”命令，然后再返回第 2 步新建其他工作区。

1.5.2　Dev C++开发环境

本书选择 Dev C++开发环境，书中所有例题均在此环境下调试通过。Dev C++软件的使用方法如下。

(1) 软件的安装与设置。第一次安装使用 Dev C++软件，通常会提示语言选项，默认为英语，可以选择中文，如图 1-18 所示。初始安装后，默认的字号很小，可以选择“工具”菜单下的“编辑器选项”命令对字体字号进行设置，如图 1-19 所示。

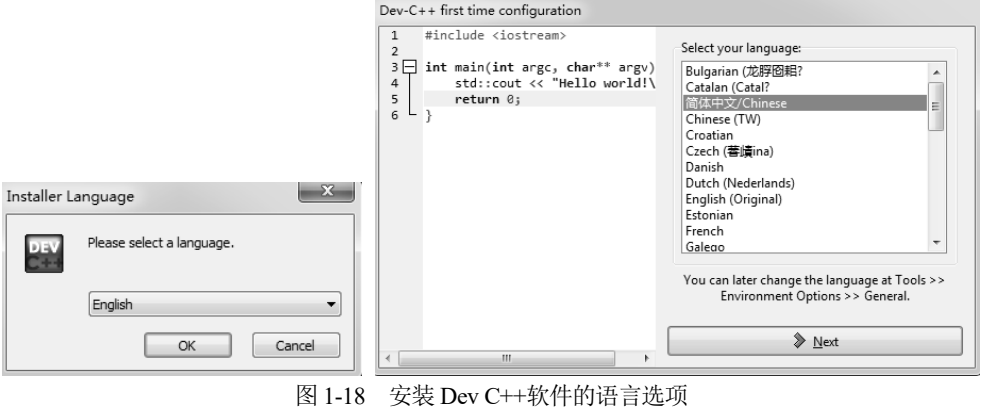

图 1-18　安装 Dev C++软件的语言选项

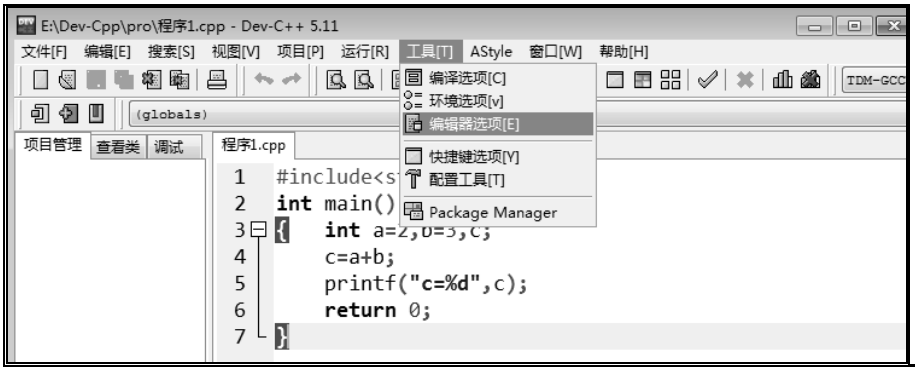

图 1-19　Dev C++软件编辑选项设置

然后在弹出的“编辑器属性”窗口中选择“字体”下拉菜单修改字体，在“大小”下拉列表框中修改字号大小，如图 1-20 所示。

图 1-20　Dev C++软件编辑选项的字体设置

(2) 源程序文件的创建。选择“文件”菜单下的“新建”命令，然后选择“源代码”可创建源文件。

(3) 源程序文件的编辑与保存。新建源程序后，在编辑窗口编辑源程序，然后选择“文件”菜单下的“保存”命令进行保存，可以保存为.c 或.cpp 源程序，如图 1-21 所示。

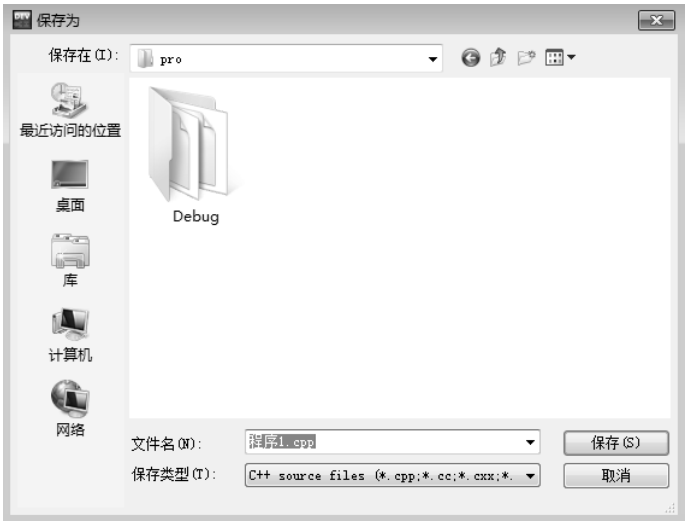

图 1-21　源程序文件的保存

(4) 源程序文件的编译运行。保存后可通过“运行”菜单下的“编译”和“运行”命令进行编译和运行，或者直接选择“编译运行”命令；也可以单击编译运行工具条上的快捷按钮“编

译(F9)”“运行(F10)”或“编译运行(F11)”程序，如图 1-22 所示。

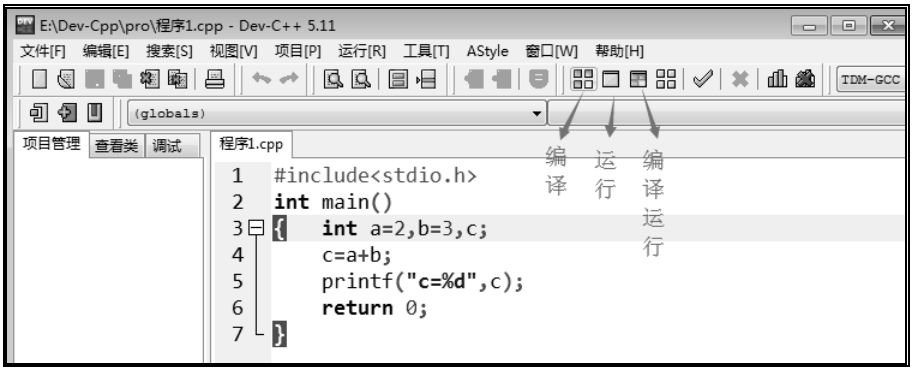

图 1-22　Dev C++程序的编译运行工具条

若程序有错误，编译器里显示错误信息，可通过错误提示修改程序，如图 1-23 所示。

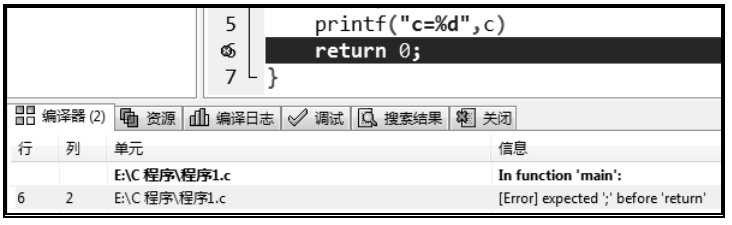

图 1-23　编译器显示错误信息

编译运行成功后弹出运行结果窗口，显示运行结果，如图 1-24 所示。

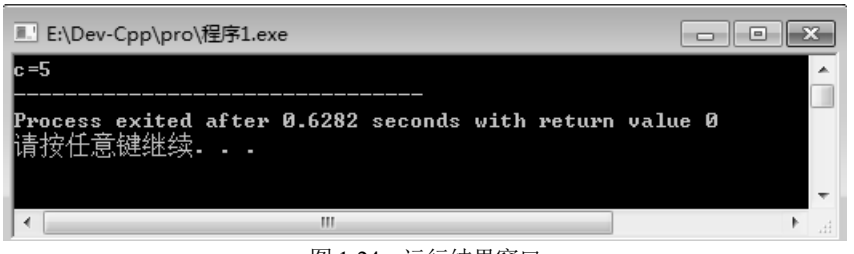

图 1-24　运行结果窗口

1.6　习题

1.6.1　选择题

1. C 语言源程序扩展名为.C，需经过(　　)之后，生成.exe 文件，才能运行。
   A. 编辑、编译　　B. 编辑、连接
   C. 编译、连接　　D. 编辑、改错
2. C 语言可执行程序的开始执行点是(　　)。
   A. 程序中第一条可执行语言　　B. 程序中第一个函数
   C. 程序中的 main 函数　　D. 包含文件中的第一个函数

3. 下列叙述正确的是(　　)。
   A. 一个 C 程序只能包含一个 main 函数
   B. C 源程序可以直接运行
   C. C 语言是一种面向对象的程序设计语言
   D. C 程序文件的扩展名为.CPP
4. 下列叙述正确的是(　　)。
   A. C 语言是一种结构化的程序设计语言
   B. C 程序中不允许出现注释行
   C. C 程序一条语句只能写在一行上
   D. C 程序不允许使用循环结构
5. 下列叙述正确的是(　　)。
   A. C 程序一行内可以有多条语句　　B. C 程序中语句的结束符号为逗号
   C. C 程序中的注释行只能占用一行　　D. C 程序中的变量可以先使用后定义
6. C 语言程序总是从 main 函数开始执行，main 函数要写在(　　)。
   A. 程序文件的开始　　B. 程序文件的最后
   C. 它所调用的函数的前面　　D. 程序文件的任何位置
7. 下列叙述正确的是(　　)。
   A. C 程序中的语句块由＜＞括起来　　B. C 程序中每一行必须有一个分号
   C. C 程序中的函数体由( )括起来　　D. C 程序的复合语句由{ }括起来
8. 下面关于 C 语言的叙述，正确的是(　　)。
   A. 每行只能写一条语句　　B. 程序中必须包含有输入语句
   C. main 函数必须位于文件的开头　　D. 每条语句最后必须有一个分号
9. 下面关于 C 语言的叙述中，错误的是(　　)。
   A. 若一条语句较长，也可分在下一行上
   B. 构成 C 语言源程序的基本单位是表达式
   C. C 语言源程序中大、小写字母是有区别的
   D. 一个 C 语言源程序可由一个或多个函数组成
10. 结构化程序设计的 3 种基本结构是(　　)。
   A. 函数结构、分支结构、判断结构　　B. 函数结构、嵌套结构、平行结构
   C. 顺序结构、分支结构、循环结构　　D. 分支结构、循环结构、嵌套结构

1.6.2　填空题

1. C 程序都是从_____函数开始执行。
2. C 程序的语句都是用_____结束。
3. 用来在屏幕上显示信息的库函数是_____，用来从键盘读取数据的库函数是_____。

4. C 程序中_____用来提高程序的可读性。
5. C 语言共有_____个关键字，_____种控制语句，_____种运算符。

1.6.3 编程题

1. 在屏幕上输出如图 1-25 所示的图案。

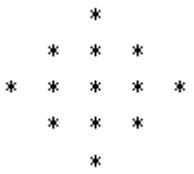

图 1-25 星号图案

2. 编程实现：输入两个整数，输出其中较大数。

1.6.4 简答题

1. 简述 C 语言的发展历史。
2. 低级语言和高级语言的主要区别是什么?
3. 举例说明 C 程序的基本结构。