



高等学校新课程体系计算机基础教育系列教材

# C语言程序设计

李丽娟 牛莉 主编  
欧阳利军 于述春 李盛欣 副主编



中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE



高等学校新课程体系计算机基础教育系列教材

# C 语言程序设计

李丽娟 牛 莉 主 编

欧阳利军 于述春 李盛欣 副主编

中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

---

## 内 容 简 介

本教材共有 11 章, 可分为 3 个部分, 使读者能够由浅入深地学习 C 语言的知识, 同时掌握一定的编程方法。

本书第 1 章和第 2 章为第一部分, 为初学者的入门知识, 使读者对 C 语言编写程序的步骤、方法和程序结构有一个大致的了解; 第 3~5 章为第二部分, 是程序设计的基础部分, 使读者详细了解描述程序算法的流程图及 C 语言的控制结构, 掌握了第一、二部分的内容后, 读者就可以完成简单的程序设计了; 第 6~11 章为第三部分, 讲述了模块化程序设计的概念和实现的方法, 并讲述了图形方式的程序设计方法和对文件的操作方法。

每一个章节里都有详细的程序范例来说明程序语句的概念、作用、含义和使用方法, 通过通俗易懂的语言, 由浅入深, 注重程序语句本身的功能与作用, 强调对 C 语言语句的掌握。

本书适合作为大专院校的教材, 也可作为一般工程技术人员的参考书。

### 图书在版编目 (CIP) 数据

C 语言程序设计/李丽娟, 牛莉主编. —北京: 中国铁道出版社, 2006. 7

(高等学校新课程体系计算机基础教育系列教材)

ISBN 7-113-07276-3

I. C... II. ①李... ②牛... 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 (2006) 第 085168 号

书 名: C 语言程序设计

作 者: 李丽娟 牛 莉等

出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 秦绪好

责任编辑: 苏 茜 谢立和

封面设计: 薛 为

责任校对: 熊严飞

印 刷: 河北省遵化市胶印厂

开 本: 787×1092 1/16 印张: 16.75 字数: 403 千

版 本: 2006 年 7 月第 1 版 2006 年 7 月第 1 次印刷

印 数: 1~5 000 册

书 号: ISBN 7-113-07276-3/TP·1974

定 价: 23.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签, 无标签者不得销售

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。



# 高等学校新课程体系计算机基础教育系列教材

## 编审委员会

顾 问： 桂卫华 中南大学  
 主 任： 邹北骥 中南大学  
 副主任： 蔡 放 陈溪辉 郭观七 黄铜城 蒋加伏 李丽娟  
 刘任任 刘少华 彭小宁 沈 岳 施荣华 谭敏生  
 度 清 羊四青 余绍黔 张新林

委 员：（排名不分先后）

邹北骥	中南大学	施荣华	中南大学
刘卫国	中南大学	骆嘉伟	湖南大学
李丽娟	湖南大学	刘任任	湘潭大学
王 毅	湘潭大学	刘相滨	湖南师范大学
沈 岳	湖南农业大学	张林峰	湖南农业大学
蒋加伏	长沙理工大学	谢中科	长沙理工大学
谢 兵	邵阳学院	黄铜城	邵阳学院
陈溪辉	衡阳师范学院	魏书堤	衡阳师范学院
羊四青	湖南人文科技学院	张新林	湖南科技学院
林 华	湖南科技学院	高为民	湖南工学院
龚德良	湘南学院	石良武	湖南商学院
余绍黔	湖南商学院	郭观七	湖南理工学院
王惠宇	湖南理工学院	彭小宁	怀化学院
胡德斌	怀化学院	蔡 放	长沙学院
杨秀平	长沙学院	谭敏生	南华大学
马淑萍	南华大学	度 清	吉首大学
李春来	吉首大学	丁 超	怀化医专
谢建全	湖南财专	刘少华	湖南女子大学

计算机技术的发展和广泛应用,正深刻地改变着现有的社会生产方式和生活方式,成为信息社会的重要支柱。信息化社会对人才的培养提出了更高的要求 and 标准。掌握和了解计算机技术并具有应用计算机的能力是适应信息化社会的基础。

“高等学校新课程体系计算机基础教育系列教材”是根据教育部高等学校非计算机专业计算机基础课程教学指导分委员会提出的《关于进一步加强高等学校计算机基础教学的意见》中新的精神和教学基本要求来进行调整编写的。我们从计算机技术发展的趋势和教学改革对人才培养的需求出发,实现知识传授与能力培养的有效结合,通过对教学内容的基础性、科学性和前瞻性的研究,体现以有效知识为主体,构建支持学生终身学习的计算机知识基础和 能力基础,反映计算机技术的最新发展成果。本系列教材强调理论与实践相结合,既注重基本原理、基本概念的介绍,又注重基本操作、基本能力的培养;根据计算机技术的发展和 应用,加重了新技术、新知识的内容;同时有配套的实验教材,主教程主要侧重于介绍原理、概念和有关理论知识,实验教程有助于学生的上机操作,提高动手能力。

教育是科学,其价值在于求真。教育是艺术,其生命在于创新。大学教育真正要教会学生的或者说最能体现学习意义的应该是学习精神、学习能力和创新能力。学习应该是超越课本知识的一个过程。本系列教材内容广泛新颖、取材丰富实用、阐述深入浅出、结构合理清晰。本系列教材的出版,不仅是编者们的努力的结果,同时也凝结了编委会许多人的心血,中国铁道出版社的编辑们为系列教材的出版任劳任怨、一丝不苟。因此,本系列教材的出版是集体智慧的结晶,是各院校优势互补、突出学校特色的一次有益尝试。在此,我们向所有为本系列教材的出版付出辛勤劳动的老师们及铁道出版社的同仁们表示崇高的敬意和衷心的感谢!系列教材在编写过程中也曾得到教育部计算机基础教学指导委员会许多委员的悉心指导以及许多高校从事计算机公共课教学的老师们的大力支持,编委会在此向他们表示衷心感谢!

本系列教材既可作为高等学校各学科非计算机专业或计算机应用专业的教材,也可作为信息技术的培训教材或作为全国计算机等级考试(NCRE)的参考书。

由于时间仓促,书中粗浅疏漏或叙述欠严密之处在所难免,恳请读者批评指正。我们将每年对系列教材进行一次认真的修订工作,为此我们热切期待着所有授课教师在教学实践中对系列教材提出宝贵意见和建议。

湖南省高教学会计算机教育专业委员会  
高等学校新课程体系计算机基础教育系列教材编委会

2006年6月28日

C 语言程序设计是许多大学计算机基础课程中一门重要的计算机基础必修课程，该课程的目的是通过对 C 程序设计语言的学习，为今后进一步学习计算机知识打下基础，从而进一步培养学生的逻辑思维和抽象思维的能力，学会提出问题、解决问题；掌握综合分析问题的方法，培养一种科学的思考问题和解决问题的能力。

本书作者都是多年来从事 C 语言程序设计课程教学的老师，这些老师还从事相关科学研究，具有较丰富的科研实际经验和程序开发能力，对程序设计有较好的研究，同时在 C 语言程序设计的教学过程中有较好的经验积累。全书共有 11 章，第 1 章简要介绍了 C 语言的发展历史、简单的 C 语言程序结构及 C 语言程序的实现方法，目的是使学生清楚地了解 C 语言程序的编辑、编译、运行及基本程序语句；第 2 章介绍了制定程序算法的流程图和两种重要的程序语句结构：分支结构和循环结构，目的是使学生掌握分析问题的方法，学会用 C 语言程序的结构来思考问题和解决问题；第 3~5 章介绍了构成简单程序的基本语句和基本结构；第 6 章介绍了模块化程序设计的方法，目的是要使学生了解和掌握怎样通过程序功能来划分程序模块；第 7~11 章进一步介绍了通过数组、指针、结构、文件、位运算等全面了解 C 语言程序设计的手段和方法，培养和锻炼开放性思考问题和解决问题的能力。

本书每一章的后面都配备了大量的程序范例和习题，帮助读者理解基本概念，通过书中的习题，进一步熟练掌握 C 语言的语法结构和应用，提高程序设计能力。

C 语言本身并不难学，学习用 C 语言进行程序设计是一种综合思维训练。本书可以为那些还没有任何程序设计经验的学生提供一个良好的入门教程。通过系统的学习，正确认识什么是程序，并学会 C 语言程序的编写，掌握思考问题和解决问题的方法，并初步掌握用 C 语言来进行程序设计。

C 语言是一种非常有效的编程语言，它简练的语法结构和灵活的机制是开发系统软件和应用软件的首选语言。同时，C 语言又是 C++ 程序设计语言的基础，学会了 C 语言，也就为将来学习 C++ 程序设计语言打下了一个坚实的基础。为了让读者既能够学会 C 语言的语法规则，又能够解决一些实际问题，本书提供了丰富的程序范例和习题，这些程序范例的难度适中。希望读者通过本书中的程序范例，体会到程序设计的过程，并且可以改写程序，使程序逐步完善。

本书精选了大量的范例程序，每一个程序都是经过了上机验证的，并且对程序的结构、函数的设计、变量的设置进行了有效的说明，借以提高读者程序设计的能力。

与本书配套的《C 语言程序设计实验教程》给出了本书中习题的全部参考答案和学生上机实验的内容。通过完成实验、上机编写程序，达到掌握知识的目的。还可以将 C 语言概念方面的一些问题，编写成程序，然后编译、运行，查看程序的运行结果。从程序的运行结果可以讨论程序的正确与否，从而真正掌握 C 语言程序设计的基本方法和基本技能。

该书的特点是通俗易懂，由浅入深，内容安排紧凑，可操作性强。系统讲述了 C 语言的基本语法结构。

C 语言程序设计课程建议学时数为 88 学时（约 4 学分），其中课堂教学学时数为 40 学时，上机实验学时为 48 学时，各章节的教学学时数可大致划分如下：

章节	内容	课堂教学时数	上机实验时数
1	引言	2	2
2	程序的简单算法制定	2	2
3	基本程序语句	4	6
4	选择结构	2	4
5	循环结构	2	4
6	函数与编译预处理命令	8	8
7	数组	4	4
8	指针	8	8
9	构造数据类型	4	4
10	文件	2	4
11	位运算	2	2

实际教学中可以根据具体情况予以调整，适当减少或增加学时数。

本书第1章、第2章由李丽娟编写，第3章由牛莉编写，第4章、第5章由李盛欣编写，第6章、第7章由马淑萍编写，第8章、第9章由欧阳利军编写，第10章由谢建全编写，第11章由于述春编写。

由于作者水平有限，书中难免会有不足之处。希望各位读者和专家提出宝贵的意见，帮助作者不断改进和完善，我们将不胜感激。

编者

2006年5月

<b>第 1 章 引言</b> .....	1
1.1 C 语言的特点.....	1
1.2 简单的 C 语言程序介绍.....	2
1.3 C 语言程序的结构.....	4
1.4 C 程序设计语言的执行.....	4
1.4.1 源程序翻译.....	4
1.4.2 链接目标程序.....	5
1.4.3 集成开发环境.....	6
小结.....	6
习题一.....	7
<b>第 2 章 程序的简单算法制定</b> .....	8
2.1 结构化程序的算法制定.....	8
2.2 结构化程序的算法描述.....	8
2.2.1 流程图.....	9
2.2.2 N-S 图.....	13
2.2.3 PAD 图.....	14
2.3 算法制定范例.....	15
小结.....	18
习题二.....	18
<b>第 3 章 基本程序语句</b> .....	20
3.1 C 语言的数据类型简介.....	20
3.2 几个基本概念.....	20
3.2.1 标识符.....	20
3.2.2 常量.....	21
3.2.3 变量.....	22
3.3 C 语言的 3 种基本数据类型.....	23
3.3.1 整型数据.....	23
3.3.2 实型数据.....	25
3.3.3 字符型数据.....	25
3.4 基本运算符和表达式.....	26
3.4.1 C 语言运算符简介.....	26
3.4.2 算术运算符和算术表达式.....	27
3.4.3 赋值运算符和赋值表达式.....	28
3.4.4 关系运算符和关系表达式.....	32
3.4.5 逻辑运算符和逻辑表达式.....	32



3.4.6	条件运算符和条件表达式.....	34
3.4.7	逗号运算符和逗号表达式.....	35
3.4.8	强制类型转换运算符.....	36
3.5	C 基本语句.....	36
3.5.1	简单语句.....	36
3.5.2	复合语句.....	38
3.6	数据的输出.....	38
3.6.1	格式化输出函数.....	38
3.6.2	字符输出函数.....	43
3.7	数据的输入.....	44
3.7.1	格式化输入函数 (scanf 函数).....	44
3.7.2	字符输入函数.....	46
3.8	简单程序设计举例.....	46
小结	.....	48
习题三	.....	48
<b>第 4 章</b>	<b>选择结构</b> .....	<b>52</b>
4.1	if 语句.....	52
4.1.1	单分支 if 语句.....	52
4.1.2	双分支 if 语句.....	52
4.1.3	多分支 if 语句.....	53
4.2	switch 语句.....	56
4.2.1	switch 语句简介.....	56
4.2.2	break 语句在 switch 结构中的运用.....	57
4.3	实例解析.....	59
小结	.....	63
习题四	.....	63
<b>第 5 章</b>	<b>循环结构</b> .....	<b>68</b>
5.1	goto 语句构成循环.....	68
5.2	while 循环结构.....	69
5.3	do...while 循环结构.....	70
5.4	for 循环结构.....	71
5.5	break 语句和 continue 语句在循环结构中的运用.....	73
5.5.1	break 语句.....	73
5.5.2	continue 语句.....	74
5.6	实例解析.....	75
小结	.....	78
习题五	.....	79

<b>第 6 章 函数与编译预处理命令</b> .....	84
6.1 库函数 .....	84
6.2 函数的定义和说明 .....	85
6.2.1 函数的定义 .....	85
6.2.2 函数的返回值 .....	87
6.2.3 对被调用函数的说明和函数原型 .....	88
6.3 函数的调用 .....	90
6.3.1 函数调用的一般形式 .....	90
6.3.2 函数调用的方式 .....	91
6.4 函数的参数 .....	91
6.5 函数的嵌套调用和递归调用 .....	94
6.5.1 函数的嵌套调用 .....	94
6.5.2 函数的递归调用 .....	95
6.6 变量的存储类型 .....	102
6.6.1 变量的作用域与生存期 .....	102
6.6.2 变量的存储类型 .....	102
6.6.3 局部变量 .....	103
6.6.4 全局变量 .....	105
6.7 内部函数、外部函数 .....	108
6.7.1 内部函数 .....	108
6.7.2 外部函数 .....	109
6.8 编译预处理命令 .....	109
6.8.1 宏替换 .....	110
6.8.2 文件包含 .....	112
6.8.3 条件编译 .....	113
小结 .....	116
习题六 .....	116
<b>第 7 章 数组</b> .....	125
7.1 一维数组 .....	125
7.1.1 一维数组的定义 .....	125
7.1.2 一维数组元素的引用 .....	126
7.1.3 一维数组的初始化 .....	126
7.2 二维数组 .....	129
7.2.1 二维数组的定义和引用 .....	129
7.2.2 二维数组元素的初始化 .....	130
7.3 多维数组 .....	132
7.3.1 多维数组的定义、使用与存储 .....	132
7.3.2 多维数组的初始化 .....	133

7.4 字符串与字符数组.....	133
7.4.1 字符串与字符数组.....	133
7.4.2 字符数组的初始化.....	134
7.4.3 字符数组的输入/输出.....	135
7.4.4 字符串处理函数.....	139
7.5 字符串数组.....	140
7.6 数组作为函数参数.....	141
7.6.1 数组元素作为函数参数.....	141
7.6.2 地址量作为函数参数.....	141
小结.....	146
习题七.....	146
<b>第 8 章 指针.....</b>	<b>154</b>
8.1 指针简介.....	154
8.2 指针变量定义及引用.....	155
8.2.1 指针变量的定义.....	155
8.2.2 指针变量的引用.....	155
8.3 指针运算.....	158
8.3.1 赋值运算.....	158
8.3.2 加减算术运算.....	159
8.3.3 关系运算.....	159
8.4 指针与数组.....	160
8.4.1 指向数组的指针变量的定义.....	160
8.4.2 指向数组的指针变量的引用.....	161
8.4.3 指向数组的指针变量的加减.....	162
8.4.4 指针与字符串.....	164
8.5 多维数组的指针.....	167
8.6 指向指针的指针.....	170
8.7 指针与函数.....	170
8.7.1 函数指针变量.....	170
8.7.2 指针型函数.....	171
8.7.3 指针数组与函数.....	173
8.7.4 main 函数的参数.....	176
小结.....	177
习题八.....	178
<b>第 9 章 构造数据类型.....</b>	<b>187</b>
9.1 结构体.....	187
9.1.1 结构类型定义和结构变量说明.....	187
9.1.2 结构体类型变量的引用.....	189
9.1.3 结构体数组.....	191

9.1.4 结构体与指针 .....	193
9.1.5 结构体与链表 .....	195
9.1.6 C 语言的动态存储分配函数 .....	202
9.2 共用体 .....	204
9.3 枚举 .....	206
9.4 自定义数据类型 .....	207
小结 .....	207
习题九 .....	208
<b>第 10 章 文件</b> .....	<b>212</b>
10.1 文件、流和文件系统的概念 .....	212
10.2 文件类型指针 .....	213
10.3 缓冲文件系统 .....	213
10.3.1 文件的打开、关闭和文件结束测试 .....	214
10.3.2 文件的读/写 .....	216
10.3.3 文件的定位 .....	227
10.3.4 出错的处理 .....	229
10.4 非缓冲文件系统 .....	230
10.4.1 文件的打开、建立与关闭 .....	230
10.4.2 文件的读/写 .....	232
小结 .....	234
习题十 .....	234
<b>第 11 章 位运算</b> .....	<b>235</b>
11.1 位运算概述 .....	235
11.1.1 计算机中数的表示 .....	235
11.1.2 硬件接口寄存器位的读取与设置 .....	236
11.2 位运算使用方法 .....	237
11.2.1 按位与运算 .....	237
11.2.2 按位或运算 .....	238
11.2.3 按位异或运算 .....	239
11.2.4 按位取反运算 .....	239
11.2.5 左移运算 .....	240
11.2.6 右移运算 .....	242
11.3 位运算应用举例 .....	243
11.3.1 各种位运算的运算顺序 .....	243
11.3.2 综合举例 .....	245
11.4 位域 .....	248
小结 .....	251
习题十一 .....	252
<b>参考文献</b> .....	<b>255</b>

# 第 1 章 引 言

C 语言是在 20 世纪 70 年代初问世的。1978 年由美国电话电报公司 (AT&T) 贝尔实验室正式发表了 C 语言。同时由 B.W.Kernighan 和 D.M.Ritchie 合著了著名的《THE C PROGRAMMING LANGUAGE》一书, 通常简称为“K&R”, 也有人称之为 K&R 标准。但是, 在 K&R 中并没有定义一个完整的标准 C 语言, 后来由美国国家标准协会 (American National Standards Institute) 在此基础上制定了一个 C 语言标准, 于 1983 年发表, 通常称之为 ANSI C。

## 1.1 C 语言的特点

C 语言简洁、紧凑, 使用方便、灵活, 程序书写自由, 共有 9 种控制语句。ANSI C 规定 C 语言共有 32 个关键字, 关键字就是已被编程语言本身使用的标识符, 不能再用做变量名、函数名等其他用途, 如表 1-1 所示。

表 1-1 C 语言的关键字

auto	const	double	float	int	short	struct	unsign
break	continue	else	for	long	signed	switch	void
case	default	enum	goto	register	static	typedef	volatile
char	do	extern	if	return	sizeof	union	while

不同的实现方式对 C 语言的关键字有不同的扩充, Turbo C 扩充了 11 个关键字, 如表 1-2 所示。

表 1-2 Turbo C 扩充的关键字

_es	_ss	_ds	_es	_ss	cdecl
far	huge	interrupt	near	pascal	

一般而言, C 语言的关键字不允许用户将其用做其他用途, 在 C 语言中, 所有关键字都是小写的。

C 语言还具有如下特点:

(1) 运算符丰富。共有 34 种。C 语言把括号、赋值、逗号等都作为运算符处理。从而使 C 语言的运算类型极为丰富, 可以实现其他高级语言难以实现的运算。

(2) 数据结构类型丰富。

(3) 具有结构化的控制语句。

(4) 语法限制不太严格, 程序设计自由度大。

(5) C 语言允许直接访问物理地址, 能进行位 (bit) 操作, 能实现汇编语言的大部分功能, 可以直接对硬件进行操作。因此有人把它称为中级语言。

(6) 生成目标代码质量高, 程序执行效率高。

(7) 与汇编语言相比, 用 C 语言编写的程序可移植性好。

但是，C 语言对程序员要求也高，程序员用 C 语言编写程序会感到限制少、灵活性大，功能强，但较其他高级语言在学习上要困难一些。

## 1.2 简单的 C 语言程序介绍

为了说明 C 语言源程序结构的特点，先看以下几个程序。这几个程序由简到难，体现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍，但可从这些例子中了解到一个 C 语言源程序的基本组成部分和书写格式。

**【例 1.1】** C 语言源程序结构示例。

**【源程序】**

```
#include<stdio.h>
main()
{
    printf("Hello,world! /n");
}
```

**【程序说明】**

(1) **include** 是文件包含命令，扩展名为 **.h** 的文件称为头文件，在这里包含的文件是 **stdio.h**，它表示在程序中要用到这个文件中的函数。“**#**”是一个标志。

(2) **main** 是主函数的函数名，表示这是一个主函数。

(3) **printf** 是函数调用语句，**printf** 函数的功能是把要输出的内容送到显示器去显示。

**printf** 函数是一个由系统定义的标准函数，在 **stdio.h** 中，可在程序中直接调用。

(4) **main** 函数中的内容必须放在一对花括号“**{}**”中。

**【例 1.2】** **main** 函数示例。

**【源程序】**

```
#include<stdio.h>
#include<math.h>
main()
{
    double x,s;
    printf("input number: ");
    scanf("%lf", &x);
    s=cos(x);
    printf("cos(%lf)is %lf/n", x,s);
}
```

**【程序说明】**

(1) 程序包含了两个头文件：**stdio.h** 和 **math.h**。

(2) 在 **main** 函数中定义了两个双精度实数型变量：**x** 和 **s**。

(3) **printf("input number:");**用于显示提示信息。

(4) **scanf("%lf",&x);**用于从键盘获得一个实数 **x**。

(5)  $s=\cos(x)$ ;求  $x$  的余弦, 并把它赋给变量  $s$ 。

(6) `printf("cos(%lf)is %lf\n",x,s)`;显示程序运算结果。双引号中有两个格式字符`%lf`, 分别对应着两个输出变量:  $x$  和  $s$ 。

程序的功能是从键盘输入一个数  $x$ , 求  $x$  的余弦值, 然后输出结果。`include` 称为文件包含命令, 其意义是把尖括号`<>`内指定的文件包含到本程序中, 成为本程序的一部分。被包含的文件可以是由系统提供的, 其扩展名为`.h`。因此也称为头文件或首部文件。C 语言的头文件中包括了各种标准库函数的函数原型。因此, 凡是在程序中调用一个库函数时, 都必须包含该函数的原型所在的头文件。在本例中, 使用了 3 个库函数: 输入函数 `scanf`、余弦函数 `cos` 和输出函数 `printf`。`cos` 函数是数学函数, 其头文件为 `math.h`, 因此在程序的主函数前用 `include` 命令包含了 `math.h`。`scanf` 和 `printf` 是标准输入/输出函数, 其头文件为 `stdio.h`, 在主函数前也用 `include` 命令包含了 `stdio.h` 文件。

需要说明的是, 有些编译环境对 `scanf` 和 `printf` 这两个函数可以省去对其头文件的包含命令。所以在例 1.1 和例 1.2 中可以省略文件包含命令`#include<stdio.h>`。

但是, 在任何情况下都写上`#include<stdio.h>`是一个良好的习惯, 否则, 会由于编译系统的不同而发生语法错误。

**【例 1.3】** 头文件示例。

**【源程序】**

```
#include<stdio.h>
int add(int x, int y);
main()
{ int a, b, c;
  printf("please input value of a and b:\n");
  scanf("%d %d", &a, &b);
  c=add(a,b);
  printf("max=%d",c);
}
int add(int x, int y)
{
  return(x+y);
}
```

**【程序说明】**

在例 1.3 中的主函数体中又分为两部分, 一部分为说明部分, 另一部分为执行部分。说明是指变量的类型说明。例 1.1 中未使用任何变量, 因此无说明部分。C 语言规定, 源程序中所有用到的变量都必须先说明, 后使用, 否则将会出错。这一点是编译型高级程序设计语言的一个特点, 说明部分是 C 语言源程序结构中一个很重要的组成部分。

运行本程序时, 首先在屏幕上给出提示串: `please input value of a and b:`, 这是由执行部分的第一行完成的。用户在提示下从键盘上键入两个数, 如 `5.9.1` (或 `5 9.1`), 接着在屏幕上会显示出计算结果: `max=14`。

## 1.3 C 语言程序的结构

一个 C 语言程序可由下面几部分组成：

- ① 文件包含部分
- ② 预处理部分
- ③ 变量说明部分
- ④ 函数原型声明部分
- ⑤ 主函数部分
- ⑥ 函数定义部分

关于程序结构的说明：

(1) 并不是所有的 C 语言程序都必须包含上面的 6 个部分，一个最简单的 C 语言程序可以只包含两个部分：文件包含部分、主函数部分。

(2) C 语言程序文件的后缀为.c，并且每一个 C 语言源程序都必须有且仅有一个主函数，主函数的组成形式如下：

```
main()  
{  
    变量说明部分  
    程序语句部分  
}
```

每一个 C 语言程序的语句都以分号结束。在以后的章节里，会详细地讲述 C 语言程序中各部分的作用和使用方法。

## 1.4 C 程序设计语言的执行

用程序语言编写的程序称为源程序 (Source Program)，实际上计算机本身并不能直接理解这样的语言，必须将程序语言翻译成机器语言，计算机才能理解程序。将源程序翻译成机器语言的过程称为编译，编译的结果是得到源程序的目标代码 (Object Program)；最后还要将目标代码与系统提供的函数和自定义的过程 (或函数) 链接起来，就可得到计算机可执行的程序。计算机可执行的程序称为可执行程序或执行文件。

### 1.4.1 源程序翻译

适合 C 语言的编译器不止一种。一般来说，C 语言源程序的翻译过程如图 1-1 所示。它由 3 个部分组成：

- (1) 词法分析器 (lexical)
- (2) 语法分析器 (Parser)
- (3) 代码生成器 (Code Generator)

词法分析器主要是对源程序进行词法分析，它是按单个字符的方式阅读源程序，并且识



别出哪些符号的组合可以代表单一的单元，并根据它们是否是数字值、单词(标识符)、运算符等，将这些单词分类。词法分析器将词法分析结果保存在一个称为记号(Token)的结构单元里，并将这个记号交给语法分析器，词法分析会忽略源程序中的所有注释。

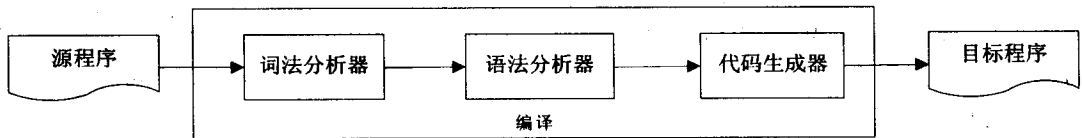


图 1-1 程序语言的翻译过程

语法分析器直接对记号进行分析，并识别每个成分所扮演的角色。这些语法规则也就是程序设计语言的语法规则。

代码生成器是将经过语法分析、没有语法错误的程序指令转换成机器语言指令。

例如，假设编写了一个名为 `mytest` 的程序，源程序的全名为 `mytest.c`，用 Microsoft C 编译器，在命令方式下，可以采用下面这样的方式对 `mytest.c` 进行编译：

```
cl -c mytest.c
```

如果源程序没有错误，就会生成一个名为 `mytest.obj` 的目标代码程序。其他程序语言也会有类似的命令将源程序翻译成目标代码，具体的命令与程序语言的编译器有关。

#### 1.4.2 链接目标程序

通过翻译产生的目标代码尽管是机器语言的形式，但却不是计算机可以执行的方式。这是因为为了支持软件的模块化，允许程序语言在不同的时期开发出具有独立功能的软件模块作为一个单元，一个可执行的程序中有可能包含一个或多个这样的程序单元，这样可以降低程序开发的低水平重复所带来的低效率。因此，目标程序只是一些松散的机器语言，要获得可执行的程序，还需将它们链接起来。

程序的链接工作是由链接器(linker)来完成的。链接器的任务就是将目标程序链接成可执行的程序(或称载入模块)，这种可执行的程序是一种可存储在磁盘存储器上的文件。

例如，假设已对源程序 `mytest.c` 进行编译后生成了目标代码程序 `mytest.obj`，于是，可以利用链接器生成可执行代码，在命令方式下，将用下面的方式来链接程序：

```
link /out:mytest.exe mytest.obj
```

如果不发生错误，就会生成一个名为 `mytest.exe` 的载入模块，也就是可执行的代码程序。最后，可以通过操作系统将这个加载模块加载入内存，执行这个程序。

上面对程序进行编译，链接都只针对了一个源程序文件，实际上，可以将多个源程序文件通过编译，链接成一个可执行文件。

例如，假设有 3 个源程序：`file1.c`、`file2.c` 和 `file3.c`，每一个源程序都包含有不同的函数或过程，在命令方式下可先用编译器对 3 个源程序进行编译：

```
cl -c file1.c
cl -c file2.c
cl -c file3.c
```

分别得到 3 个目标程序：`file1.obj`、`file2.obj` 和 `file3.obj`。接下来可用链接器将 3 个目标程序进行链接。