

21世纪高等教育计算机规划教材

COMPUTER

C 语言程序设计

(附微课视频 第3版)

C Programming

■ 朱立华 陈可佳 主编

■ 刘林峰 吴家皋 郭剑 副主编

重点突出，难点分散

微课视频，图表丰富

示例经典，讲解细致



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等教育计算机规划教材

COMPUTER

C语言程序设计

(附微课视频 第3版)

C Programming

■ 朱立华 陈可佳 主编

■ 刘林峰 吴家皋 郭剑 副主编



人民邮电出版社

北京

图书在版编目(CIP)数据

C语言程序设计：附微课视频 / 朱立华, 陈可佳主编. — 3版. — 北京：人民邮电出版社, 2018.8(2019.1重印)
21世纪高等教育计算机规划教材
ISBN 978-7-115-48833-6

I. ①C… II. ①朱… ②陈… 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(2018)第148686号

内 容 提 要

本书是采用C语言进行程序设计的入门教程, 主要面向没有程序设计基础的读者, 详细介绍了C语言的基本概念、语法规则及编程技术。全书共分为12章, 内容包括: 与程序设计有关的计算机基础知识、常量与变量、运算符与表达式、流程控制、函数、数组、指针、结构、文件、多文件工程等, 最后给出了一个学生成绩管理系统综合实例。该例采用结构化程序设计的思想和方法, 对C语言中几乎所有的知识进行了实践, 便于读者从模仿开始学习编写综合性的程序。

本书的配套教材《C语言程序设计习题解析与实验指导(第3版)》, 包含了主教材中思考题的解析、每章的习题解答、补充习题与答案以及10个配套的实验与指导, 建议与本书配合使用。

本书可作为高等学校本专科各专业以C语言为基础的程序设计课程的教材, 也可作为编程爱好者自学C语言的教材和参考书。

-
- ◆ 主 编 朱立华 陈可佳
副 编 刘林峰 吴家皋 郭 剑
责任编辑 刘 尉
责任印制 沈 蓉 彭志环
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
固安县铭成印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 19.5 2018年8月第3版
字数: 508千字 2019年1月河北第2次印刷

定价: 55.00元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

信息科学技术的发展不仅极大地促进了整个科学技术的发展,而且显著地加快了经济信息化和社会信息化的进程。其中,程序设计是信息科学与技术领域创新型人才应掌握的基本技能。一直以来,各大高等院校均开设了程序设计类的相关基础课程。本教材选用C语言作为程序设计的入门教学语言,其主要原因为:一方面,C语言是主流的程序设计语言之一,广泛应用于软件系统的设计与开发以及信息科学与技术领域的研究中;另一方面,C语言也是学习其他高级程序设计语言以及计算机类专业课程如“数据结构”“操作系统”等课程的基础。

南京邮电大学计算机学院程序设计课程组自2005年起负责全校的C语言程序设计课程,经过10多年的课程建设,积累了丰富的教学经验,并形成了一套完整的课程体系。课程组于2009年自主编写了《C语言程序设计》系列教材,并于2014年进行改版。本书是2018年的再次改版,整体延续了第2版的风格,遵循循序渐进、由浅入深的教学理念,从基础知识出发到解决实际问题,让读者潜移默化地掌握C语言的语法与编程技巧。与上一版不同,此版教材增改了部分具有实际意义的程序例题并为每个例题附上了详细的演示讲解视频(读者通过扫描二维码即可观看)。此外,本书在语言和结构上也都有了明显的改进。

本书共有12章,包含了程序设计的一般性概念以及C语言特有的语法知识,主要有:C语言基础知识、数据类型及定义、程序流程控制及结构化程序设计思想、数据的永久存储等。本书的主要特点如下。

(1) 注重基础,内容翔实

本书面向编程零基础的读者,因此在第1章先简单而全面地介绍了计算机软、硬件的基本知识,特别是和程序相关的存储器知识、二进制及进制转换知识等,便于读者更好地理解C程序的开发过程。本书各章知识点内容完整,不仅涵盖了C语言的基本理论知识,还给出了实际应用中的编程技巧,对编程细节给出了提醒。

(2) 由简入繁,逐步深入

本书在内容的组织上遵循:首先提出语法,然后给出简洁易懂的示例,再在后续进阶例题的反复使用中加深理解。本书前后章节的例题具有连贯性,即随着新问题的提出引入新的知识,并逐步完善之前的例题。这种编排方式不仅可以避免初学者出现畏难情绪,而且可以让读者理解各知识点存在的必要性。

(3) 选例经典,讲解细致

本书精选的示例、例题能充分体现相关章节的知识。此外,本书还引入了一些经典且实用的算法,以增加读者的兴趣。如:斐波那契数列、判断质数、数组排序、生成随机数、验证密码等。每个例题的源代码注释详细、运行结果分析透彻、配套讲解全面、思考题有启发性。读者可通过例题索引表快速查找本书的示例,进行阅读。

本书配套的教学资源包括:《C 语言程序设计习题解析与实验指导(第3版)》、教学课件、教材源程序、所有例题的演示讲解视频、习题答案等,均可以在人邮教育社区(<http://www.ryjiaoyu.com>)免费下载。

本书的编写得到南京邮电大学程序设计课程组诸多老师的支持:第1、5章由陈可佳老师编写,第2、3章由吴家皋老师编写,第4章由郭剑老师编写,第6章由朱旻如老师编写,第7章由刘林峰老师编写,第8、9章由周剑老师编写,第10章由汪云云老师编写,第11、12章由朱立华老师编写。朱立华和陈可佳老师负责全书的统稿工作。此外,浙江大学何钦铭教授、南京邮电大学张伟教授也对本书的编写提出了许多宝贵的意见和建议,在此深表谢意。部分学生和读者对上一版书进行了及时的反馈,并对本书的改进提出了中肯的建议,为本次改版提供了更好的思路 and 方向,在此一并表示感谢。

由于编者水平有限,书中难免有疏漏不当之处,恳请读者批评指正,并直接与编者联系,不胜感激。

编者 E-mail 地址为: chenkj@njupt.edu.cn、zhulh@njupt.edu.cn。

编者
2018年5月

本书主要例题索引表

知识模块	对应知识点	例题号	本例主要功能与技巧
具有顺序结构的简单 C 程序	C 程序基本结构	例 2.1	C 程序基本框架, 源程序结构、基本符号
	变量定义及输入/输出	例 2.2	用 scanf、printf 进行输入/输出处理
	字符变量的输入/输出	例 2.3	用 getchar、putchar 进行字符数据的读写
	用 const 限定变量只读	例 2.4	只读变量的定义与使用
	运算符与表达式	例 3.1	算术运算符及其表达式
	强制类型转换	例 3.2	用强制类型转换 (int) 进行取整
	顺序结构程序的实现	例 4.1	用海伦公式求三角形面积
用 if、switch 实现分支结构	用 if...else 实现分支结构	例 4.2	比较得到两数中较大的那个数
	用 if...else 实现分支结构	例 4.3	改进例 4.1 判断三边是否能构成三角形
	嵌套的 if...else 分支结构	例 4.4	进一步改进例 4.3, 判断直角三角形
	用 switch 实现分支结构	例 4.5	用 switch 进行多分支选择的控制
用 while、for、do...while 实现循环结构	while 语句控制当型循环	例 4.6	累加求和的循环控制方法
	do...while 语句控制直到型循环	例 4.7	阶乘 (累乘) 的循环控制方法
	for 语句控制当型循环	例 4.8	各项有规律的数列求和方法
	break 与 continue 在循环中的用法	例 4.9	break 和 continue 语句的区别
	for 语句与 if 语句的嵌套结构	例 4.12	质数判断方法
循环嵌套结构	循环嵌套打印规则图表	例 4.10	九九加法 (乘法) 表的打印
	循环嵌套打印规则图形	例 4.11	循环嵌套打印有规律图形的基本方法
	循环嵌套的使用——穷举法	例 4.13	用穷举法求解百钱百鸡问题
函数的定义、调用、原型声明	函数的原型声明、定义与调用	例 5.2	函数定义、调用、声明的完整示例
	函数的定义	例 5.3	定义判断质数的函数
	函数的定义	例 5.4	定义无参无返回类型的画线函数
	函数的调用	例 5.5	调用判断质数的函数
	函数的调用	例 5.6	调用画线函数
	多个函数之间的定义与调用	例 5.11	根据题目要求合理划分功能定义函数
函数的递归	递归函数的定义与调用	例 5.7	用递归方法求阶乘, 递归的执行过程
	递归函数的定义与调用	例 5.8	用递归进行十进制向任意进制数转换
变量的生命周期与作用域	全局变量、局部变量的作用域	例 5.9	使用全局、局部变量, 统计质数个数
	静态局部变量的特性	例 5.10	巧妙利用静态局部变量求解阶乘问题
一维、二维数组的定义及基本使用	一维数组的定义、元素批量访问和处理	例 6.1	一维数组元素的定义、批量输入、求和、求平均
	一维数组应用于存储数列项	例 6.2	斐波那契数列求解、格式化输出控制
	一维数据应用于数字统计	例 6.3	巧妙利用数组下标, 统计数字出现次数
	二维数组的定义与基本操作	例 6.5	使用二维数组表示矩阵的方法
	二维数组的定义与基本操作, 随机函数 rand、srand 的使用	例 6.6	使用二维数组表示矩阵、求转置, 随机产生数组元素, 以及输出控制
二维数组的定义与应用	例 6.7	使用二维数据模拟发牌游戏	
数组实参与“数组”形参	向函数传递一维数组	例 6.4	定义函数并以数组为参数, 求一维数组元素中的最大值 (最小值)
	向函数传递二维数组	例 6.8	二维数组作实参时对应形式参数的设定

知识模块	对应知识点	例题号	本例主要功能与技巧
一维数组中的经典算法	在一维数组中查找某个值	例 6.9	在一维数组中进行顺序搜索, 查找元素
	在一维数组中插入某个值	例 6.10	在有序的一维数组中插入元素保持原序
	在一维数组中删除某个值	例 6.11	从一维数组中删除某元素的方法
	一维数组中元素的排序	例 6.12	用冒泡法对一维数组里的元素排序
指针基本知识	变量的值与变量的地址	例 7.1	显示一个 int 型变量的值及其地址
指针与数组: 一级指针与一维数组、行指针与二维数组; 一级、二级指针、指针数组、函数指针等知识	数组名的地址常量本质	例 7.2	使用数组名(指针常量)访问数组元素
	指针变量的定义、赋值、移动	例 7.3	定义指针变量访问数组元素(移动下标法)
	用指针访问一维数组	例 7.11	十进制数转换为二进制数, 各位值存于数组
	用指针访问二维数组	例 7.4	用指针访问二维数组各元素, 输出地址与值
	用一级指针访问二维数组	例 7.5	用列指针访问二维数组元素并输出
	用行指针访问二维数组	例 7.6	用行指针访问二维数组元素并输出
	用指针数组访问二维数组元素	例 7.7	用指针数组访问二维数组元素并输出
	二级指针的定义及使用	例 7.14	二级指针与一级指针、普通变量的关系
函数指针的定义、赋值与使用	例 7.17	使用函数指针调用函数, 注意函数首部要求	
指针形参及返回值在函数中的使用	传地址调用函数可改变实参变量	表 7.3	swap 交换变量函数的传值与传地址调用的区别, 对实参变量的不同影响
	通过指针参数使函数返回多个值	例 7.8	通过函数求两数的和与差
	返回值与返回指针的区别	例 7.9	利用返回指针求两数中的较小(大)值
	一维数组名形参实质为指针形参	例 7.10	指针作为形参访问一维数组, 查找异常数据
	用指针作为形参接受实参数组	例 7.12	选择法排序的函数实现, 采用指针形参
用行指针作为形参访问二维数组	例 7.13	定义函数计算矩阵的对角线之和并输出矩阵, 用行指针作为函数的形参	
用指针管理动态空间	用一级指针管理动态空间, 申请动态一维数组	例 7.15	用筛选法求一定范围内所有质数, 用指针申请的动态一维数组来存放数据
	用二级指针管理动态空间, 申请动态二维数组, 随机函数的使用	例 7.16	用二级指针管理动态二维数组空间, 调用随机函数产生元素, 输出矩阵, 释放动态空间
单个字符串的存储及处理	字符数组及字符指针处理字符串	例 8.1	用字符指针访问字符数组, 并统计
	字符串与字符数组的区别	例 8.2	用%s 控制字符串的输出
	字符串专用处理函数的使用	例 8.4	几个常用字符串专用处理函数的使用示例
	字符数组及字符指针处理字符串	例 8.5	定义函数判断一个串是否为回文
	字符数组及字符指针处理字符串	例 8.6	统计一个串中单词出现的次数
字符指针逐个访问串中的字符	例 8.7	密码问题, 定义函数判断密码是否正确	
多个字符串的存储及处理	用二维字符数组处理多个字符串	例 8.3	用二维字符数组处理多个字符串, 用 gets/puts 函数进行字符串的输入/输出
	用一维字符指针数组和二维字符数组处理多个字符串	例 8.8	用二维字符数组实现多个字符串的选择法排序
	带参数的 main 函数, 用二级指针作为形参	例 8.9	带参数的 main 函数示例, 注意第 2 个形参的两种表达形式, 本质上是二级指针
	用二维字符数组管理多个字符串, 函数中用行指针变量作为形参	例 8.10	字符串操作的综合实例: 单词本管理, (定义了)新增、删除、查找和显示 4 个功能的函数

续表

知识模块	对应知识点	例题号	本例主要技巧技能
宏定义、文件包含、条件编译等指令	无参宏的定义及使用	例 9.1	展示无参宏的定义及使用
	带参宏的定义及使用	例 9.2	带参宏定义, 理解宏替换
	条件编译指令的使用	例 9.3	条件编译指令应用示例
多文件工程的组织及相关技术	多文件工程的组织, 头文件与源文件的使用	例 9.4	定义一个工程, 包含 5 个不同的文件, 理解头文件及对应的源文件, 及文件包含的使用
	extern 声明外部变量、外部函数	例 9.5	在多文件工程中, 不同文件中外部变量与外部函数的定义、声明及调用
	多文件工程的组织、数组作为参数在函数中的传递与使用	例 9.6	多文件工程程序, 实现一维数组的输入、输出、统计、查找等功能, 正确使用文件包含
结构体类型的定义; 结构体变量、指针、数组的使用	结构体类型的定义、结构嵌套、结构变量的定义、初始化及成员的点运算符访问	例 10.1	学生结构体类型与日期结构体类型的嵌套定义, 对学生结构体变量的初始化, 对其成员的点运算符访问方式以及输入/输出处理
	采用结构体指针访问结构成员	例 10.2	结构体指针的定义, 对成员的两种访问方式
	结构体数组的定义及使用	例 10.3	结构体数组的定义及初始化, 用指针法和下标法分别访问结构数组元素的成员
	结构体作为函数的形参	例 10.4	结构体变量和结构体指针作为形式参数
	排序算法在结构体数组中的应用, 结构体数组作为实参进行传址	例 10.5	定义学生结构体, 并且根据学生的成绩进行排序, 所定义的函数中形参为结构体指针
联合与枚举类型的定义与使用	联合类型的定义, 联合变量的定义及成员的访问	例 10.6	联合类型的定义, 联合变量的定义和对其成员的访问方式, 联合变量空间大小示意
	枚举类型的定义及成员访问	例 10.7	枚举类型数组的定义和对枚举变量的访问
单链表的各种基本操作, 注意头指针的保护及变化	链表结点类型的定义, 链表示例	例 10.8	定义链表结点的类型, 定义相应的记录及指针, 通过赋特定地址值形成链表
	链表的建立、遍历、释放结点空间, 头指针的正确赋值与使用	例 10.9	用尾部插入法建立单链表并遍历输出, 最后释放所有结点空间, 用函数实现各功能, 注意头指针的传入及传出
	从链表中删除某结点, 注意头指针的变化	例 10.10	链表中删除结点 3 步骤: 定位、脱链、释放, 若删除第一个结点则改变头指针
	向链表中插入某结点保持元素值有序	例 10.11	链表中插入结点 3 步骤: 定位、生成、插入, 若插入结点为新的头结点则改变头指针
文本文件及二进制文件的操作过程、读写函数的使用	文本文件处理的过程, 单字符写入控制	例 11.1	定义文件指针后, 文件操作的全过程, 用 fputc 将内容写入到文本文件中
	文本文件单字符读取	例 11.2	用 fgetc 逐字符从文件中读出内容
	文本文件按行追加写入	例 11.3	用 fputs 按追加方式往文本文件中写入内容
	文本文件的格式化读写方式	例 11.4	用 fscanf/fprintf 控制文本文件的格式化读/写
	二进制文件的数据块写入	例 11.5	用 fwrite 向二进制文件中成块写入数据
	二进制文件的数据块读出	例 11.6	用 fread 从二进制文件中成块读出数据
	指针定位函数的使用	例 11.7	用 fseek 重新定位文件指针位置
文本文件的复制	例 11.8	进行文本、二进制文件的读写, 并用 fgetc 和 fputc 函数进行文本文件的复制	

授课内容和学时分配建议

章	本章主要内容	32 学时	48 学时	56 学时
第 1 章 初识计算机、程序与 C 语言	计算机及其基本结构	√	√	√
	计算机程序与计算机语言	◎	√	√
	C 语言简介★	◎	√	√
	进制转换知识	√	√	√
	本章建议学时数	1	2	2
第 2 章 初识 C 源程序及其 数据类型	C 源程序及其符号★	√	√	√
	C 语言中的数据类型	√	√	√
	常量★	√	√	√
	变量★	√	√	√
	基本数据类型在计算机内部的表示	◎	◎	√
本章建议学时数	3	3	4	
第 3 章 运算符与表达式	什么是运算符与表达式	√	√	√
	运算符的优先级与结合性★	√	√	√
	常用运算符★	√	√	√
	运算过程中的数据类型转换	√	√	√
	位运算符	◎	◎	√
本章建议学时数	3	3	4	
第 4 章 程序流程控制	语句与程序流程	√	√	√
	顺序结构★	√	√	√
	选择结构★	√	√	√
	循环结构★	√	√	√
	break 与 continue	◎	√	√
	应用举例——二维文本图形打印、质数判断、百钱百鸡	√	√	√
本章建议学时数	5	6	6	
第 5 章 函数的基本知识	模块化程序设计与函数	√	√	√
	函数的定义★	√	√	√
	函数的调用★	√	√	√
	递归函数	◎	√	√
	变量的作用域与存储类型★	√	√	√
	应用举例——定义函数求面积与体积	◎	◎	√
本章建议学时数	4	6	8	
第 6 章 数组	一维数组★	√	√	√
	二维数组★	√	√	√
	数组常用算法介绍	√	√	√
	本章建议学时数	4	6	6

续表

章	本章主要内容	32 学时	48 学时	56 学时
第 7 章 指针	指针变量★	√	√	√
	指针与数组★	√	√	√
	指针与函数★	√	√	√
	应用举例	◎	√	√
	指针进阶	◎	◎	√
	本章建议学时数	4	7	8
第 8 章 字符串	字符串的定义与初始化★	√	√	√
	字符串的常用操作★	√	√	√
	应用举例	◎	√	√
	带参的 main 函数	◎	◎	√
	综合应用实例——单词本管理	◎	◎	√
	本章建议学时数	2	5	6
第 9 章 编译预处理与多文件工程程序	编译预处理★	√	√	√
	多文件工程程序★	√	√	√
	应用举例——多文件结构处理数组问题	√	√	√
	本章建议学时数	2	2	2
第 10 章 结构、联合、枚举	结构体★	√	√	√
	联合	◎	◎	√
	枚举	◎	◎	√
	链表	◎	√	√
	本章建议学时数	2	2	4
第 11 章 文件	文件与文件指针★	√	√	√
	文件的打开和关闭★	√	√	√
	文件读写★	√	√	√
	位置指针的定位	◎	◎	√
	应用举例——文件的综合操作	◎	√	√
	本章建议学时数	2	4	4
第 12 章 学生成绩管理系统的设计与实现	数据类型的定义	◎	√	√
	为结构体类型定制的基本操作	◎	√	√
	用二进制文件实现数据的永久保存	◎	√	√
	用两级菜单四层函数实现系统	◎	√	√
	本章建议学时数	0	2	2

说明：(1) 表中所列的学时数为课堂讲授所需要的理论学时数，并非课程总学时数；

(2) √表示课堂讲授内容，◎表示可自学、选学内容；

(3) ★表示必须掌握的重点内容。

目 录

第 1 章 初识计算机、程序与 C 语言

1.1 计算机及其基本结构	1
1.1.1 电子计算机概述	2
1.1.2 存储器的基本知识	4
1.2 计算机程序与计算机语言	5
1.2.1 计算机程序与程序设计	6
1.2.2 程序设计语言简介	6
1.3 C 语言简介	8
1.3.1 C 语言的起源与发展	8
1.3.2 C 语言的优势	9
1.3.3 C 程序的开发过程	10
*1.4 进制转换知识	12
1.4.1 二进制、八进制、十六进制	12
1.4.2 进制间的相互转换	13
1.5 本章小结	15
习题 1	15

第 2 章 初识 C 源程序及其数据类型

2.1 C 源程序及其符号	17
2.1.1 C 源程序的组成	17
2.1.2 C 源程序中的 6 种基本符号	19
2.2 C 语言中的数据类型	20
2.2.1 C 语言数据类型的种类	21
2.2.2 基本数据类型及其修饰符	21
2.3 常量	22
2.3.1 整型常量	22
2.3.2 实型常量	22
2.3.3 字符常量	23
2.3.4 字符串常量	24
2.3.5 符号常量	24
2.4 变量	24
2.4.1 变量的定义及初始化	24

2.4.2 变量的输入和输出	25
2.4.3 用 const 修饰符限定变量	31
*2.5 基本数据类型在计算机内部的表示	32
2.5.1 整型数据在内存中的存储形式	32
2.5.2 字符型数据在内存中的存储形式	33
2.5.3 实型数据在内存中的存储形式	33
2.6 本章小结	33
习题 2	33

第 3 章 运算符与表达式

3.1 什么是运算符与表达式	36
3.2 运算符的优先级与结合性	37
3.3 常用运算符	38
3.3.1 算术运算符	38
3.3.2 关系运算符	40
3.3.3 逻辑运算符	41
3.3.4 条件运算符	42
3.3.5 赋值及复合赋值运算符	43
3.3.6 逗号运算符	44
3.3.7 自增、自减运算符	44
3.4 运算过程中的数据类型转换	45
3.4.1 表达式中的自动类型转换	45
3.4.2 赋值中的自动类型转换	46
3.4.3 强制类型转换	46
*3.5 位运算符	48
3.6 本章小结	51
习题 3	51

第 4 章 程序流程控制

4.1 语句与程序流程	53
4.1.1 语句的分类	53
4.1.2 程序流程及其表示	54
4.2 顺序结构	55
4.3 选择结构	56
4.3.1 if 语句	56

4.3.2 switch 语句	60	6.3 数组常用算法介绍	112
4.4 循环结构	61	6.3.1 数组元素查找	112
4.4.1 while 语句	61	6.3.2 插入数组元素	113
4.4.2 do···while 语句	62	6.3.3 删除数组元素	115
4.4.3 for 语句	63	6.3.4 数组排序	117
4.5 break 与 continue	64	6.4 本章小结	118
4.6 应用举例	65	习题 6	119
4.6.1 二维文本图形打印	65	第 7 章 指针	124
4.6.2 质数判断	67	7.1 指针变量	124
4.6.3 百钱百鸡	68	7.1.1 变量地址和变量的值	125
4.7 本章小结	69	7.1.2 指针变量的定义和访问	125
习题 4	70	7.1.3 指针变量的运算	128
第 5 章 函数的基本知识	73	7.2 指针与数组	130
5.1 模块化程序设计与函数	73	7.2.1 指针与一维数组	130
5.2 函数的定义	75	7.2.2 指针与二维数组	132
5.2.1 函数声明	76	*7.2.3 指针数组	135
5.2.2 函数定义	77	7.3 指针与函数	136
5.3 函数的调用	78	7.3.1 传值与传地址	136
5.3.1 函数调用的基本形式	78	7.3.2 指针作形参返回多个值	138
5.3.2 函数调用的完整过程	80	*7.3.3 返回指针的函数	139
*5.4 递归函数	82	7.4 应用举例	140
5.5 变量的作用域与存储类型	86	7.4.1 批量数据的筛查	140
5.5.1 变量的作用域	86	7.4.2 进制转换	141
5.5.2 变量的存储类型	88	7.4.3 选择法排序	142
5.6 应用举例——定义函数求解面积与 体积	91	7.4.4 矩阵运算	144
5.7 本章小结	93	*7.5 指针进阶	145
习题 5	93	7.5.1 const 与指针的结合	145
第 6 章 数组	98	7.5.2 二级指针	146
6.1 一维数组	98	7.5.3 指针与动态空间	147
6.1.1 一维数组的定义	99	7.5.4 指向函数的指针	151
6.1.2 一维数组的访问	100	7.6 本章小结	153
6.1.3 一维数组的初始化	100	习题 7	153
6.1.4 一维数组的应用	101	第 8 章 字符串	158
6.2 二维数组	104	8.1 字符串的定义与初始化	158
6.2.1 二维数组的定义	105	8.1.1 用字符数组处理字符串	158
6.2.2 二维数组的初始化	105	8.1.2 用字符指针处理字符串	160
6.2.3 二维数组的访问	107	8.2 字符串的常用操作	161
6.2.4 二维数组的应用	108	8.2.1 字符串的输入和输出	161
		8.2.2 字符串处理的常用函数	164

8.3 应用举例	168	11.2.2 文件关闭操作	233
*8.4 带参的 main 函数	173	11.3 文件读写	233
8.5 综合应用实例——单词本管理	175	11.3.1 字符读写	234
8.6 本章小结	180	11.3.2 字符串读写	236
习题 8	180	11.3.3 格式化读写	238
		11.3.4 块数据读写	241
第 9 章 编译预处理与多文件		*11.4 位置指针的定位	243
工程程序	184	11.5 应用举例——文件的综合操作	245
9.1 编译预处理	184	11.6 本章小结	251
9.1.1 文件包含	184	习题 11	252
9.1.2 宏定义	185	*第 12 章 学生成绩管理系统的设计与实现	254
9.1.3 条件编译	187	12.1 数据类型的定义	255
9.2 多文件工程程序	188	12.2 为结构体类型定制的基本操作	256
9.2.1 多文件工程程序的组织结构	189	12.3 用二进制文件实现数据的永久保存	262
9.2.2 外部变量与外部函数	191	12.4 用两级菜单提示用户选择	264
9.2.3 静态全局变量与静态函数	192	12.5 主控模块的设计与实现	265
9.3 应用举例——多文件结构处理数组问题	193	12.6 本章小结	274
9.4 本章小结	198	习题 12	274
习题 9	198	附录 A 常用字符与 ASCII 码对照表	275
第 10 章 结构、联合、枚举	201	附录 B C 语言的关键字	277
10.1 结构体	201	附录 C Visual C++ 下各数据类型所占字节数及取值范围	278
10.1.1 结构体类型的定义	202	附录 D C 语言运算符的优先级与结合性	279
10.1.2 结构体变量	203	附录 E 常用的 ANSI C 标准库函数	281
10.1.3 结构体指针	206	附录 F C 程序设计常见错误及解决方案	285
10.1.4 结构体数组	207	附录 G 命名规则	292
10.1.5 向函数传递结构体	209	附录 H C 语言的发展简史	294
10.1.6 结构体应用——学生成绩排名	210	附录 I 函数 printf 的格式转换说明符	295
*10.2 联合	212	附录 J 函数 printf 的格式修饰符	296
*10.3 枚举	215	参考文献	297
*10.4 链表	217		
10.4.1 链表的概念	217		
10.4.2 链表的基本操作	219		
10.5 本章小结	227		
习题 10	227		
第 11 章 文件	230		
11.1 文件与文件指针	230		
11.2 文件的打开和关闭	231		
11.2.1 文件打开操作	232		

第 1 章

初识计算机、程序与 C 语言

计算机擅长接受指令，但不擅长了解你的思想。

Computers are good at following instructions, but not at reading your mind.

——唐纳德·克努特 (Donald Knuth)，计算机科学家

学习目标：

- 了解现代计算机的冯·诺依曼体系结构
- 了解存储器的概念及其容量的计算方法
- 了解程序以及程序设计的基本概念
- 了解 C 语言的发展史，并掌握 C 程序的开发过程
- 了解进制的概念，并掌握进制间的转换方法

继移动互联网之后，人工智能技术已经席卷全球，引爆了下一场信息革命、工业革命、医疗革命、金融革命等。事实上，这一切的发生都借助于一个工具，那就是计算机。所有新思想、新技术的诞生都是由程序而来的。那么，计算机为何如此“神奇”？又如何用计算机实现人类无穷的创造力呢？

一个完整的计算机系统由硬件系统与软件系统组成。其中，软件系统由一个个程序 (Programs) 构成，这些程序或者是系统软件，或者是应用软件，都是用特定的程序设计语言 (Programming Language) 开发的。程序设计语言的发展经历了从机器语言到汇编语言再到各种高级程序设计语言的过程。其中，C 语言作为一种主流的高级程序设计语言，不仅是计算机软件设计与开发的主流语言之一，也是认识和深入掌握其他程序设计语言的基础。因此，本书将主要介绍 C 语言的知识以及如何使用 C 语言进行程序设计。事实上，无论用何种语言进行程序设计，遵循程序设计的基本过程、掌握程序设计的基本方法都是至关重要的。

本章将带领大家初步认识计算机，并了解什么是程序，如何设计程序，以及程序是如何改变世界的。

1.1 计算机及其基本结构

本节要点：

- 冯·诺依曼“程序存储”的思想
- 存储器容量的计算方法

计算机是在什么样的背景下诞生的，经历了哪些发展历程？现代计算机的体系结构是如何组

成的? 程序存放在计算机的哪个部分, 是如何存放的? 本节将一一解答上述问题。

1.1.1 电子计算机概述

计算机, 俗称“电脑”, 本质上是一种电子设备, 因此常被称为“电子计算机”。历史上有很多人对计算机的诞生和发展起到至关重要的作用。英国数学家查尔斯·巴贝奇 (Charles Babbage) 于 1834 年发明了分析机 (也就是现代电子计算机的前身), 并设想现代计算机所具有的大多数其他特性, 被称为计算机原型机之父。同为英国数学家, 阿兰·图灵 (Alan Turing) (如图 1.1 所示) 提出了“图灵机”和“图灵测试”等重要概念, 是计算机逻辑的奠基者, 被称为计算机科学之父、人工智能之父。为纪念他在计算机领域的卓越贡献, 美国计算机协会于 1966 年设立“图灵奖”, 此奖项被誉为计算机科学界的诺贝尔奖。1939 年, 美国的约翰·阿塔纳索夫 (John Atanasoff) 和他的学生克利福特·贝瑞 (Clifford Berry) 造出了第一台真实的电子计算机阿塔纳索夫-贝瑞计算机 (Atanasoff-Berry Computer, 简称为 ABC)。不过, 这台机器不可以编程, 仅设计用于求解线性方程组。

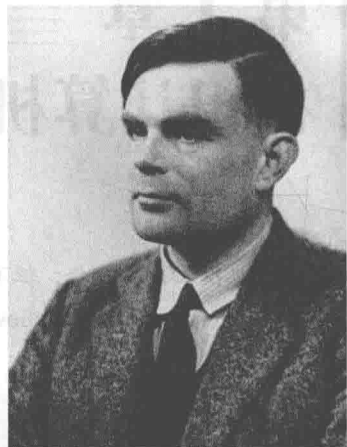


图 1.1 阿兰·图灵

1946 年 2 月 14 日, 由美国军方定制的“电子数字积分计算机” (Electronic Numerical And Calculator, ENIAC) 在美国宾夕法尼亚大学问世了。ENIAC (如图 1.2 所示) 是世界上第一台通用计算机 (即为“图灵完备”的), 也是继 ABC 后的第二台电子计算机。它是美国军方为了满足计算弹道的需要而研制的。这台计算机使用了 17 840 支电子管, 占地面积约 170 平方米, 重约 30 吨, 功耗约为 150 千瓦, 其运算速度为每秒 5000 次加法, 造价约为 48 万美元。

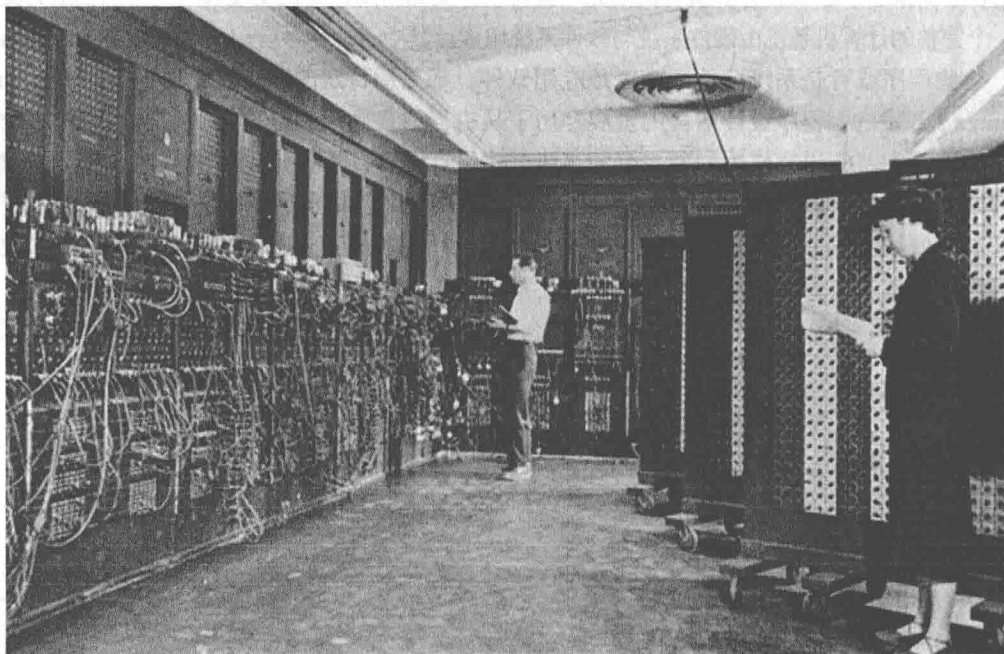


图 1.2 ENIAC 计算机

ENIAC 计算机可以编程解决各种计算问题。不过，它在研制初期却存在一个最大的弱点——没有真正的存储器，所有指令存储在计算机的电路中，解题之前必须先想好所需的全部指令，并手工把相应的电路连通。这种准备工作要花几小时甚至几天的时间，而计算本身只需几分钟。因此，计算的高速与程序设计的低速存在很大的矛盾。幸运的是，当时任弹道研究所顾问、正参与美国第一颗原子弹研制工作的美籍匈牙利数学家约翰·冯·诺依曼（John Von Neumann）（如图 1.3 所示）带着原子弹研制过程中遇到的大量计算问题，在 ENIAC 研制过程中期加入了研制小组。1945 年，冯·诺依曼和他的研制小组发表了一个全新的“存储程序通用电子计算机方案”——EDVAC（Electronic Discrete Variable Automatic Computer）。在此过程中他对计算机的许多关键性问题的解决做出了重要贡献，从而保证了计算机的顺利问世。时至今日，电子计算机的体系结构依然是基于冯·诺依曼的思想。因此，他被称为“现代计算机之父”。



图 1.3 冯·诺依曼

冯·诺依曼的主要贡献是提出了**程序存储（Stored Program）**的思想：把运算程序存放在机器的存储器中，程序设计员只需要在存储器中寻找运算指令，机器就会自行计算，这样，就不必每个问题都重新编程，从而大大加快了运算进程。这一思想标志着自动运算的实现，也标志着电子计算机的成熟，成为了电子计算机设计的基本原则，并被一直沿用至今。冯·诺依曼的另一个重大贡献是建议在电子计算机中采用**二进制（Binary）**：根据电子元件双稳工作的特点，二进制的采用将大大简化机器的逻辑线路。

一般来说，基于冯·诺依曼体系结构的计算机（简称为冯·诺依曼机）具有以下功能：（1）能够把程序和数据送至计算机中；（2）必须具有长期记忆程序、数据、中间结果及最终运算结果的能力；（3）能够完成各种算术、逻辑运算以及处理和传送数据的能力；（4）能够控制程序走向，并根据指令控制机器的各部件协调操作；（5）能够按照要求将处理结果输出给用户。

因此，冯·诺依曼机应具备五大基本组成部件，包括：

- （1）**输入设备**：输入数据和程序；
- （2）**存储器**：存放程序指令和数据；
- （3）**运算器**：完成加工处理数据；
- （4）**控制器**：控制程序执行；
- （5）**输出设备**：输出处理结果。

其中，运算器和控制器合称为**中央处理器（Central Processing Unit, CPU）**，是计算机最核心的组成部分。冯·诺依曼机的五大部件之间通过控制总线、地址总线、数据总线这三大总线相联结，有数据流、指令流、控制流通过总线联系各部件（如图 1.4 所示）。

图 1.4 表明，程序（指令）和数据均通过输入设备输入到计算机，并存于存储器中；运算时，指令由存储器送入控制器，由控制器产生控制流来控制数据流的流向以及各部件的工作；数据在控制流的作用下从存储器读入运算器进行运算，运算的中间及最后结果又存回存储器；存储器中的运算结果经输出设备输出。关于计算机的体系结构和详细的工作原理，请参阅“微机原理”或“计算机组成与结构”等计算机专业课程的相关知识。

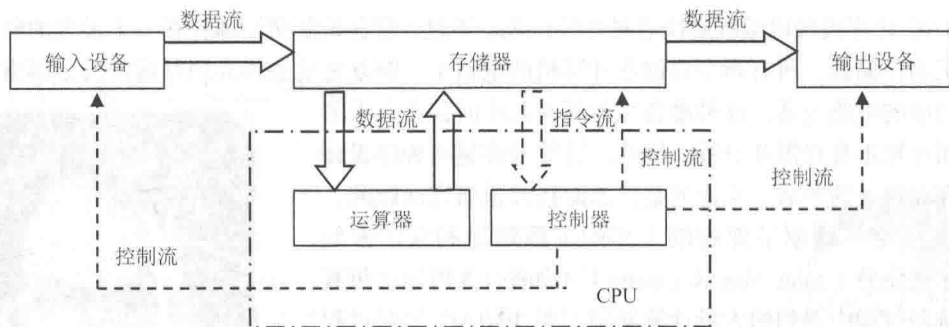


图 1.4 冯·诺依曼电子计算机结构图

ENIAC 的问世具有划时代的意义，表明电子计算机时代的到来。在以后的 60 多年里，计算机技术以惊人的速度发展，没有任何一门技术的性能价格比能在数十年内增长 6 个数量级。按照电子计算机逻辑元件的组成材料，电子计算机的发展可划分为 4 个阶段，如表 1.1 所示。

表 1.1 电子计算机的发展历程

时代	名称	起止年	硬件	软件及应用领域	特点
第 1 代	电子管时代	1946—1958	逻辑元件采用真空电子管；主存储器采用汞延迟线、磁鼓、磁芯；外存储器采用磁带	采用机器语言、汇编语言编程；应用领域以军事和科学计算为主	体积大、功耗高、可靠性差、速度慢（每秒几千至几万次）、价格昂贵
第 2 代	晶体管时代	1958—1964	逻辑元件采用晶体管；主存储器采用磁芯	操作系统，采用高级语言及其编译程序编程；应用领域有科学计算、事务处理和工业控制	体积缩小、能耗降低、可靠性提高、速度提高（每秒几十万次）
第 3 代	集成电路时代	1964—1970	逻辑元件采用中、小规模集成电路；主存储器采用磁芯	分时操作系统以及结构化、模块化程序设计方法；应用领域进入文字处理和图形图像处理	速度更快（每秒几十万到几百万次）、可靠性更高、价格下降，通用化、标准化
第 4 代	大规模集成电路时代	1970 至今	逻辑元件采用大规模和超大规模集成电路	数据库和网络管理系统以及面向对象语言；应用领域除了科学计算、工业控制等，逐步走向家庭	集成度高、体积小，速度极快（每秒百万至数亿次）、微处理器诞生（1971）

随着人工智能、大数据挖掘、分布式数据库以及嵌入式系统等领域的不断发展，对计算机各方面性能要求也在不断增加，势必会带动新的技术和工艺的不断发 展。这些新型计算机将广泛地应用于军事、科研、经济、文化等各个领域。未来，我们也期待出现突破冯·诺依曼体系的具有更高性能的计算机系统。

1.1.2 存储器的基本知识

由于程序及其处理的数据都存放在计算机的存储器中，本小节将简要介绍存储器的基本知识，便于读者理解程序在执行过程中如何存放不同类型的数据，以及数据值是如何变化的。

存储器 (Memory) 是计算机中存储信息的部件。对于计算机来说，有了存储器才有记忆功能，才能保证其正常工作。存储器有不同的种类，根据其在计算机中的用途，可分为主存储器（即内存储器，简称内存）和辅助存储器（即外存储器，简称外存）。大家熟悉的硬盘、光盘等都属于外存。通常，我们把要永久保存的、大量的数据存储在外存上，而把暂时存储的、少量的数据和程序放在内存上。本书只介绍与程序密切相关的内存的基本知识。