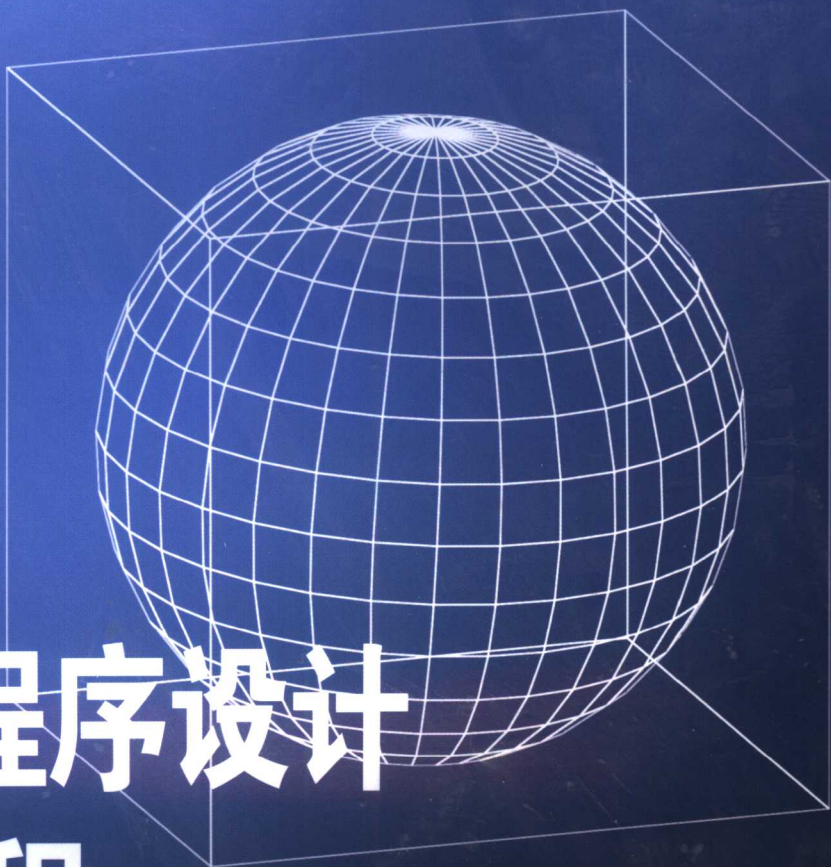




普通高等教育“十一五”国家级规划教材
(高职高专教育)



C语言程序设计 基础教程

崔发周 主 编



高等教育出版社

普通高等教育“十一五”国家级规划教材

(高职高专教育)

C 语言程序设计基础教程

崔发周 主编

高等教育出版社

内容提要

本书是普通高等教育“十一五”国家级规划教材（高职高专教育）。

本书以培养学生结构化程序设计的基本能力为主线，采用由浅入深、循序渐进的教学策略进行讲解，主要内容包括 C 语言程序设计概述、顺序结构程序设计、分支结构程序设计、循环结构程序设计、数组与字符串、函数、编译预处理、指针、结构体与共用体、文件以及位运算等内容，非常适合初学者使用。教材编排时充分考虑了教师组织教学的需要，各章均提供了教学目标、本章小结及丰富的例题、习题，还可以为用户提供各章的源程序代码、教学建议、习题参考答案等教学资料。

本书可作为应用性、技能型人才培养的各类教育相关专业的教学用书，也可供各类培训、计算机从业人员和爱好者参考。

图书在版编目（CIP）数据

C 语言程序设计基础教程 / 崔发周主编. —北京：高等教育出版社，2007.5

ISBN 978-7-04-021229-7

I. C… II. 崔… 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（2007）第 037099 号

策划编辑 冯 英 责任编辑 彭立辉 封面设计 杨立新 责任绘图 朱 静
版式设计 陆瑞红 责任校对 姜国萍 责任印制 张泽业

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京丰源印刷厂

开 本 787×1092 1/16
印 张 19
字 数 460 000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 [http:// www.hep.edu.cn](http://www.hep.edu.cn)
[http:// www.hep.com.cn](http://www.hep.com.cn)
网上订购 [http:// www.landracom.com](http://www.landracom.com)
[http:// www.landracom.com.cn](http://www.landracom.com.cn)
畅想教育 [http:// www.widedu.com](http://www.widedu.com)

版 次 2007 年 5 月第 1 版
印 次 2007 年 5 月第 1 次印刷
定 价 22.00 元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 21229-00

前 言

因材施教是各级各类教育取得良好效果的一个基本教学原则。目前, 高职高专的计算机教材存在 3 个方面的突出问题: 一是以编者为中心, 而不是以学生为中心, 内容偏难、偏多, 脱离读者的实际需要; 二是品种单一, 不适应高等教育大众化的需要; 三是结构化程度低, 不能较好地体现职业能力的培养。高职高专院校学生与普通本科高校学生在学习上存在差异, 这是一个不争的事实。如果职业教育工作者有意或无意地以本科院校作参照, 而不是以高职高专院校的教育实际需要作参照, 不仅不能提高培养工作水平, 还会产生适得其反的效果, 这一点已经被高职教育的实践所证明。因此, 编写一套真正适合高职高专教育层次的计算机教材, 不论对高职高专教师还是学生都非常重要。

高职高专院校计算机类专业的程序设计课程是一门重要的专业基础课, 其课程目标应该定位在培养学生结构化程序设计的基本能力。由于计算机技术的快速发展, 学生在校期间不可能获得计算机应用所需要的全部知识, 更不可能获得终生工作所需要的知识。因此, 对计算机专业来说, 培养可迁移的通用职业能力比掌握具体的专业知识更为重要。程序设计课程就是一门培养学生通用职业能力的基础性课程, 对学生进一步学习软件开发和增强对未来工作岗位的适应能力都具有关键作用。C 语言作为一种通用的、流行的结构化程序设计语言, 非常适合作为程序设计课程的教学语言。正是出于这种考虑, 作者将本书定名为《C 语言程序设计基础教程》。学完本课程之后, 继续学习面向对象的程序设计课程将成为一件非常轻松的事情。编者真诚地建议, 读者一定要抓住程序设计能力这条主线, 以程序设计带动“语言”的学习, 而不是相反。如果读者还不具备足够的程序设计能力, 就盲目地追求新出现的各种程序设计语言, 将得不到好的效果。也许读者在若干年之后才会真正明白这个道理, 但在学习过程中就这样做将会受益无穷。

为了取得满意的教学效果, 编者对本教材各部分的学时分配建议如下:

序 号	授 课 内 容	学 时 分 配		
		总 学 时	讲 课	实 践
1	C 语言程序设计概述	4	2	2
2	顺序结构程序设计	10	4	6
3	分支结构程序设计	8	4	4
4	循环结构程序设计	8	4	4
5	数组与字符串	8	4	4
6	函数	10	4	6
7	编译预处理	4	2	2
8	指针	8	4	4
9	结构体与共用体	8	4	4
10	文件	8	2	6
11	位运算	4	2	2
12	综合实训	16	3	13
	合 计	96	39	57



由于各职业院校在程序设计教学中已经实现理论教学和实践教学的一体化，因此以上课时安排仅供教师参考。

本书由崔发周主编，第 8 章、第 9 章由关艳奎编写，第 10 章由薛桂云编写，第 11 章由邢秀红编写，其余各章由崔发周编写，附录 A 至附录 E 由邢秀红编写，附录 F 由崔羽佳编写。刘晓军为本书绘制了部分插图，全书由崔发周统稿。承德石油高等专科学校的宋汉珍教授在百忙中进行审稿，并为本书的改进提出了宝贵的建议，在此深表感谢。高等教育出版社高等职业教育研究与出版中心计算机分社的编辑对本书的出版进行了精心策划和热心指导，唐山工业职业技术学院的领导对本书的出版给予了大力支持，在此谨表谢意。

编者很清楚地了解高职高专院校计算机教学的问题所在，并尽自己所能努力地改变目前的状况。但不得不承认，面对十分复杂的职业教育现象，作者的认识还十分肤浅，从根本上解决问题还要依靠广大同仁的共同努力。我们恳切欢迎兄弟院校的师生对本教材提出宝贵建议，编者电子信箱：cui999@tom.com。

编 者

2006 年 12 月

目 录

第1章 C语言程序设计概述.....1

1.1 程序设计在计算机应用中的地位和作用.....2

1.2 程序设计必备的基础知识——二进制基础.....3

1.2.1 进位计数制.....4

1.2.2 二进制.....4

1.2.3 十六进制和八进制.....5

1.2.4 计算机中的编码.....5

1.2.5 数据的存储形式.....7

1.3 程序设计语言.....7

1.3.1 概述.....7

1.3.2 程序设计语言的发展与分类.....8

1.4 C语言的特点.....9

1.5 Turbo C 2.0 集成开发环境.....11

1.6 一个简单的C语言程序.....18

本章小结.....19

习题.....20

第2章 顺序结构程序设计.....21

2.1 算法.....21

2.1.1 算法的特征.....21

2.1.2 算法的直观描述.....22

2.2 数据.....24

2.2.1 基本数据类型.....24

2.2.2 常量和变量.....25

2.2.3 数据的运算.....28

2.3 基本语句.....33

2.4 基本输入/输出函数.....34

2.4.1 字符数据的输入/输出.....35

2.4.2 格式输入与格式输出.....36

2.5 顺序结构程序设计.....41

2.6 典型顺序结构程序举例.....42

本章小结.....44

习题.....45

第3章 分支结构程序设计.....47

3.1 if 语句.....48

3.1.1 提出问题.....48

3.1.2 if 语句的基本结构.....48

3.1.3 关系表达式和逻辑表达式.....49

3.1.4 条件运算和条件表达式.....54

3.1.5 if 的嵌套.....56

3.2 switch 语句.....62

3.2.1 提出问题.....62

3.2.2 switch 的一般形式.....62

3.3 典型程序举例.....66

本章小结.....68

习题.....68

第4章 循环结构程序设计.....70

4.1 提出问题.....71

4.2 while 语句.....72

4.3 do-while 语句.....75

4.4 for 语句.....79

4.5 循环的嵌套.....83

4.6 break 语句和 continue 语句.....86

4.6.1 break 语句.....87

4.6.2 continue 语句.....89



4.7 while、do-while 和 for 三种循环的比较	91	6.2.1 有参函数定义的一般形式	137
4.8 典型程序举例	93	6.2.2 无参函数定义的一般形式	138
本章小结	98	6.2.3 空函数定义的一般形式	138
习题	98	6.3 函数参数和函数返回值	139
第5章 数组与字符串	100	6.3.1 形式参数和实际参数	139
5.1 提出问题	101	6.3.2 函数的返回值	141
5.2 一维数组	102	6.4 函数的调用	142
5.2.1 一维数组的定义	102	6.4.1 函数调用的一般形式	142
5.2.2 一维数组元素的引用	102	6.4.2 调用过程	143
5.2.3 一维数组的初始化	104	6.4.3 对被调函数的声明	144
5.2.4 一维数组的输入/输出	105	6.5 数组作为函数的参数	148
5.2.5 一维数组应用举例	105	6.6 函数的嵌套调用	154
5.3 二维数组	112	6.7 函数的递归调用	156
5.3.1 二维数组的定义	112	6.8 外部变量与内部变量	159
5.3.2 二维数组的引用	113	6.8.1 外部变量的定义	160
5.3.3 二维数组的存储	114	6.8.2 外部变量的作用范围	160
5.3.4 二维数组的输入/输出	114	6.8.3 外部变量的使用说明	162
5.3.5 二维数组的初始化	115	6.9 静态变量与自动变量	164
5.3.6 二维数组应用举例	115	6.9.1 静态存储方式与动态存储方式	165
5.4 字符串处理	118	6.9.2 静态变量的声明	165
5.4.1 字符数组的定义和引用	119	6.9.3 静态变量的使用说明	165
5.4.2 字符数组的初始化	120	6.10 变量的存储类别	167
5.4.3 字符数组的输入/输出	120	6.10.1 auto 类型	168
5.4.4 字符串处理的标准函数	123	6.10.2 register 类型	168
5.4.5 字符数组应用举例	128	6.10.3 static 类型	168
本章小结	131	6.10.4 extern 类型	169
习题	131	6.11 内部函数与外部函数	171
第6章 函数	134	6.11.1 内部函数	171
6.1 概述	135	6.11.2 外部函数	172
6.1.1 自定义函数在 C 语言程序设计中的作用	135	6.12 多文件程序的编译与连接	174
6.1.2 C 语言程序各函数之间的关系	135	6.12.1 利用项目文件实现	174
6.1.3 C 函数的分类	136	6.12.2 利用文件包含命令实现	175
6.2 函数定义的一般形式	137	本章小结	176
		习题	176
		第7章 编译预处理	178
		7.1 宏定义	179



7.1.1 无参宏定义	179	9.1.1 结构体类型定义	219
7.1.2 带参宏定义	182	9.1.2 结构体变量定义	221
7.2 文件包含	185	9.2 结构体变量的引用与初始化	222
7.3 条件编译	186	9.2.1 结构体变量的引用规则	222
本章小结	188	9.2.2 结构体变量的初始化	223
习题	188	9.3 结构体数组	223
第8章 指针	190	9.4 指向结构体类型数据的指针	224
8.1 指针与指针变量的概念	191	9.4.1 指向结构体变量的指针	224
8.2 指针变量的定义与引用	193	9.4.2 指向结构数组的指针	226
8.2.1 指针变量的定义与相关运算	193	9.4.3 指向结构数据的指针作函数的 参数	226
8.2.2 指针变量作函数参数	196	9.5 用 typedef 定义已有类型的别名	227
8.3 数组的指针和指向数组的 指针变量	198	*9.6 链表处理——结构指针的应用	230
8.3.1 概述	198	9.6.1 概述	230
8.3.2 通过指针引用数组元素	198	9.6.2 创建一个新链表	232
8.3.3 指向数组的指针作函数的参数	200	9.6.3 对链表的插入操作	233
8.3.4 二维数组的指针及其指针变量	201	9.6.4 对链表的删除操作	234
*8.3.5 动态数组的实现	203	9.7 共用体和枚举类型	235
8.4 字符串的指针和指向字符串的 指针变量	207	9.7.1 共用体	235
8.4.1 字符串的表示与引用	207	9.7.2 枚举类型	237
8.4.2 字符串指针作函数的参数	209	本章小结	238
8.5 返回指针值的函数	210	习题	238
8.6 指针数组与主函数 main() 的 形参	211	第10章 文件	240
8.6.1 指针数组	211	10.1 文件的概念	240
*8.6.2 主函数 main() 的形参	213	10.2 文件指针	241
8.6.3 指向指针的指针变量	214	10.2.1 文件缓冲区	242
8.7 函数的指针和指向函数的指针	215	10.2.2 文件指针的定义	242
本章小结	216	10.3 文件打开和关闭函数	242
习题	216	10.3.1 文件的打开函数 fopen()	243
第9章 结构体与共用体	218	10.3.2 文件的关闭函数 fclose()	244
9.1 结构体类型与结构体变量的 定义	219	10.4 常用文件操作函数	245
		10.4.1 字符读/写函数 fgetc() 和 fputc()	245
		10.4.2 字符串读/写函数 fgets() 和 fputs()	248
		10.4.3 数据块读/写函数 fread() 和 fwrite()	250



10.4.4 格式化读/写函数 fscanf()和 fprintf()	252	本章小结	267
10.4.5 文件的随机读/写	253	习题	268
10.5 文件检测函数	255		
本章小结	256		
习题	257		
第 II 章 位运算	258	第 12 章 综合实训	269
11.1 位运算符和位运算	258	12.1 基本要求	269
11.1.1 按位与 (&)	259	12.2 程序示例	270
11.1.2 按位或 ()	261	本章小结	273
11.1.3 按位取反 (~)	261		
11.1.4 按位异或 (^)	262	附录	274
11.1.5 左移位 (<<)	262	附录 A C 语言中的关键字	274
11.1.6 右移位 (>>)	263	附录 B 常用字符与 ASCII 码对照表	274
11.1.7 位复合赋值运算符	263	附录 C C 语言运算符的优先级及 其结合性	275
11.2 位运算应用	263	附录 D 输入/输出函数的格式控 制符	276
11.3 位段	265	附录 E 常用库函数	277
11.3.1 位段结构类型的定义	265	附录 F Turbo C 常见错误信息	284
11.3.2 位段变量的定义和引用	266		
11.3.3 位段的说明	267	参考文献	295

(注: 带“*”号的章节为选学内容)

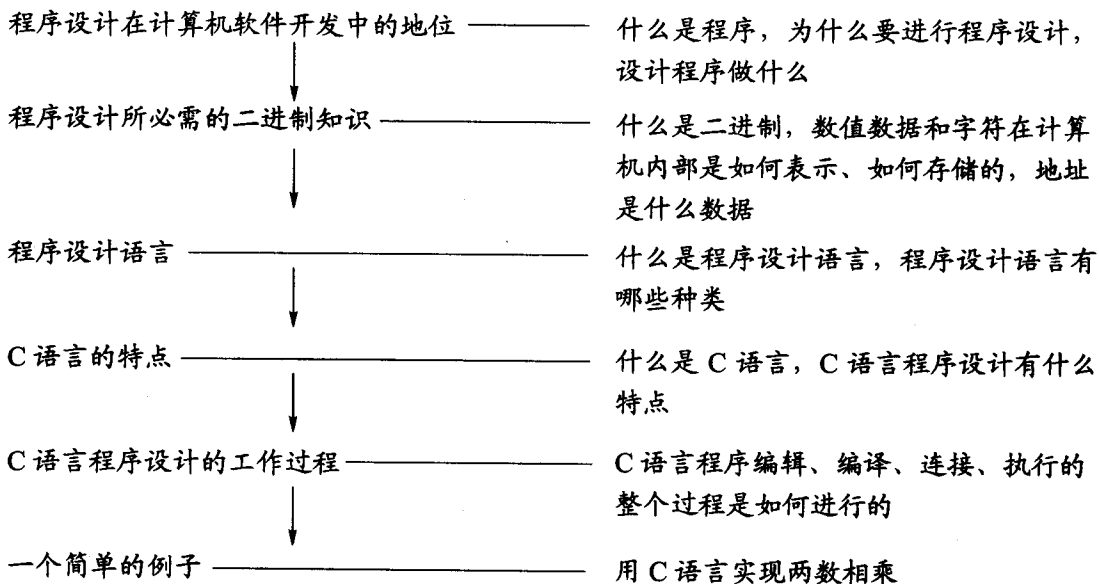
C 语言程序设计概述

教学目标:

- 知道程序设在计算机软件开发中的地位和作用。
- 知道程序设计和程序设计语言的概念,熟悉程序设计语言的分类。
- 知道 C 语言程序设计所必需的二进制知识,会进行不同数制之间的转换。
 - 能完成 C 语言程序编辑、编译、连接、执行的整个过程。
 - 知道 C 语言的特点。



本章结构:



21 世纪的人类社会是一个信息社会,计算机已经成为人们从事各种职业活动不可缺少的工具。一位优秀的程序员应该具备 3 项基本素质:一是对信息技术的现状和发展前景有充分的了解,具有乐此不疲的职业情感;二是掌握必备的职业知识,具有较强的程序设计能力;三是具



有强烈的创新精神，能够跟上信息技术快速发展的步伐。本章将首先介绍一下程序设计的入门知识。

1.1 程序设计在计算机应用中的地位和作用

计算机系统由硬件系统和软件系统两部分组成。软件系统的主体是各种计算机程序，是计算机的“神经系统”；离开计算机软件，计算机将无法正常工作。目前所用的计算机，无论性能高低、规模大小，都是根据开发者和应用者设计的一套程序来工作的。对于用户来说，应用计算机的实质就是编制出解决自身面临问题的程序，例如：完成一个复杂的科学计算，对图书馆的书目数据进行分类和检索，控制导弹准确命中目标，等等。因此，程序设计工作在 IT 领域中占有极其重要的地位。程序在 IT 应用领域中的地位如图 1-1 所示。



图 1-1 软件的价值

下面介绍一下程序设计与软件开发的关系。

简单地说，程序设计是软件开发中的一个核心环节，软件开发是围绕着程序设计来进行的。软件开发的最终结果是生成一组相互关联的程序及必备的文档。为了编制结构合理、功能强大的计算机程序，软件开发工作需要划分为需求分析、系统设计、程序设计和软件测试几个主要阶段。首先，要清楚地了解本次软件开发所面临的技术经济环境和详细的要求。例如，要知道此项目是为了解决用户的什么问题，用户的人员素质和技术素质状况，处理结果的准确程度对用户有什么影响等。为了清楚地知道这些需求，需要经常与客户、项目经理进行交流。其次，要对软件系统的整体结构进行设计。如果没有总体设计就着手设计程序，那么各个程序模块之间就很难实现正确的联系，也就无法实现系统的整体功能。有了需求调查和系统设计的基础后，就可以开始设计程序。一般来说，程序的正确性和合理性往往不是一次就能达到的，需要有一个测试和修改的过程，与写一篇文章的过程有些类似。

本课程的目标是掌握程序设计的基本方法，通过这种训练，可以培养出读者的基本编程能力、良好的工作习惯和必需的创新精神。具备了这些基本素质后，就可以进一步深入地学习开发软件系统的方法，并在未来的职业生涯中较好地适应工作环境的变化。

相关链接

一个实际的软件开发过程包括以下 6 个阶段：

1. 计划

对所要解决的问题进行总体定义,包括了解用户的要求及现实环境,从技术、经济和社会因素等3个方面研究并论证本软件项目的可行性,编写可行性研究报告,探讨解决问题的方案,并对可供使用的资源(如计算机硬件、系统软件、人力等)成本、可取得的效益和开发进度做出预测。

2. 分析

软件需求分析就是回答做什么的问题。它是一个对用户的需求进行去粗取精、去伪存真、正确理解,然后把它用软件工程开发语言(即需求规格说明书)表达出来的过程。本阶段的基本任务是和用户一起确定要解决的问题,建立软件的逻辑模型,编写需求规格说明书文档,并最终得到用户的认可。

3. 设计

软件设计可以分为概要设计和详细设计两个阶段。概要设计就是结构设计,其主要目标是给出软件的模块结构,并用软件结构图表示出来。详细设计的首要任务是设计模块的程序流程、算法和数据结构,次要任务就是设计数据库,常用的方法仍然是结构化程序设计方法。

4. 编码

软件编码是指把软件设计转换成计算机可以接受的程序,即写出以某种程序设计语言表示的源程序清单。

5. 测试

软件测试的目的是以较小的代价发现尽可能多的错误。常用的测试方法是白盒法和黑盒法。白盒法的测试对象是源程序,依据的是程序内部的逻辑结构来发现软件的编程错误、结构错误和数据错误。黑盒法依据的是软件的功能或软件行为描述,发现软件的接口上、功能上和结构上错误。

6. 维护

维护是指完成软件的研制(分析、设计、编码和测试)工作并交付使用以后,根据软件运行的情况,对软件进行适当修改,以适应新的要求,并纠正运行时发现的错误。

在实际开发过程中,并不是从第一步一直进行到最后一步。在任何阶段,都可能有一步或几步的回溯。例如,在测试过程中发现问题后可能要修改设计;用户也可能会提出一些新的需求等。

1.2 程序设计必备的基础知识——二进制基础

从本质上来说,计算机就是一种复杂的数字电路系统。要了解计算机的工作原理,顺利地



编写程序，就要首先搞清数字电路的工作逻辑——二进制。

1.2.1 进位计数制

在日常生活中，人们使用的是“十进位计数制”。在这种计数制度中，有个位、十位、百位、千位、万位等，每个数位上的“1”所表示的数量称为该位的权。当本位的数量达到权的10倍时，就会在高一个数位上增加一个“1”来表示，即“逢十进一”。

事实上，人们在不同的场合，还在使用六十进制、十二进制等各种数制。在计算机科学与技术中，要用到二进制、八进制和十六进制等几种数制。

1.2.2 二进制

二进制的计数规则是逢二进一，只使用0和1这两个数码。例如，十进制的15用二进制表示就是1111。很明显，二进制数每个数位的权都是2的倍数。按照习惯，各个数位的权值自右向左依次升高。

基本训练

1. 分析二进制的计数规律，由低到高计算各位的权值，并填写下表。

二进制数	十进制数	“1”所在数位的权		二进制数位权
		十进制	指数形式	
1	1	1	2^0	1
10	2	2	2^1	2
100				
1000				
10000				
100000				
1000000				
10000000				
100000000				
1000000000				
10000000000				

2. 二进制与十进制的转换。

(1) 将下列二进制数转换为十进制数。

$$(1011)_2 = (\quad)_{10} \quad (1001)_2 = (\quad)_{10}$$

$$(11100011)_2 = (\quad)_{10} \quad (100100100)_2 = (\quad)_{10}$$

(2) 将下列十进制数转换为二进制数。

$$(16)_{10} = (\quad)_2 \quad (15)_{10} = (\quad)_2 \quad (32)_{10} = (\quad)_2$$

$$(33)_{10} = (\quad)_2 \quad (96)_{10} = (\quad)_2$$

3. 厨师表演抻面功夫：先将一个面团拉成长条，然后多次对折，每对折一次称为“一扣”，优秀的面点师傅可以抻出 12 扣。试分析一下，如何用二进制记录下每次对折后的根数，12 扣可以达到多少根？

1.2.3 十六进制和八进制

二进制在计算机中便于实现，但人们读/写起来很不方便；十进制便于人们读/写，但二进制与十进制之间的转换不是十分直观。为了实现既便于读/写、又容易转换的目的，在计算机应用和 C 语言程序设计中，经常要用到八进制和十六进制。

八进制的计数规则是“逢八进一”，采用 0、1、2、3、4、5、6、7 共 8 个数字。例如，十进制的 8 用八进制表示就是 10。采用“三位一段”的方法，可以方便地将二进制数转换为八进制数。例如：

$$(1\ 110\ 111)_2 = (167)_8$$

十六进制的计数规则是“逢十六进一”，采用 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 共 16 个数字。例如，十进制中的 15 在十六进制中用 F 表示。采用“四位一段”的方法，可以方便地将二进制数转换为十六进制数。例如：

$$(1110\ 1010\ 1001)_2 = (EA9)_{16}$$

如果要将一个八进制数用二进制表示，只需按顺序将八进制的每个数位用 3 位二进制数表示，即可形成相应的二进制数；类似地，如果要将一个十六进制数用二进制表示，只需按顺序将十六进制的每个数位用 4 位二进制数表示即可。

基本训练

1. 将下列二进制数转换为八进制和十六进制数。

$$(10010111)_2 = (\quad)_8 = (\quad)_{16} \quad (1001001001)_2 = (\quad)_8 = (\quad)_{16}$$

$$(11111111)_2 = (\quad)_8 = (\quad)_{16} \quad (10101010)_2 = (\quad)_8 = (\quad)_{16}$$

2. 将下列八进制数转换为二进制数。

$$(111)_8 = (\quad)_2 \quad (222)_8 = (\quad)_2 \quad (777)_8 = (\quad)_2$$

$$(100)_8 = (\quad)_2 \quad (107)_8 = (\quad)_2$$

3. 将下列十六进制数转换为二进制数。

$$(9A)_{16} = (\quad)_2 \quad (F1)_{16} = (\quad)_2 \quad (AB)_{16} = (\quad)_2$$

$$(FC)_{16} = (\quad)_2 \quad (DD)_{16} = (\quad)_2 \quad (E0)_{16} = (\quad)_2$$

1.2.4 计算机中的编码

1. 数值的编码

在计算机中，要完成数据的运算，除了采用二进制外，还必须解决两个问题：一是将数的符号（正或负）用二进制数字（0 或 1）表示；二是通过合理的编码使数据能够在计算机中方便地进行运算。



第一个问题的解决方法比较简单，一般都是规定最高位作为符号位，用 0 表示正数，用 1 表示负数。在 C 语言程序设计中处理数值数据主要有 16 位和 32 位两种，实际可用的数值位分别为 15 位和 31 位。

相关链接

计算机在同一时间中能够处理的二进制数的位数叫字长。这一指标是由计算机中的 CPU 所决定的，在计算机出厂后就不再改变。通常，称字长为 n 位数据的计算机为 n 位计算机，简称 n 位机。例如，字长为 32 位的 CPU 在同一时间内处理字长为 32 位的二进制数据，使用该 CPU 的计算机就称为 32 位机。早期的 Apple 机 CPU 只有 8 位，现在已出现了 64 位的 CPU。字长越大，计算机的性能就越好，所能表示的有效位数就越多，精度就越高，从而能支持功能更强的指令，使计算机的处理能力更强。

第二个问题可通过使用补码来解决。采用补码后，计算机中的所有数值运算都可以简化为加法和移位。具体的原理，可以参考有关计算机原理的书籍。

下面以 16 位数为例，介绍一下用补码表示的方法。

除符号位外，其他二进制位为数值的绝对值，这种编码称为原码。例如：

+32 的原码表示为：00000000 00100000

-32 的原码表示为：10000000 00100000

+0 的原码表示为：00000000 00000000

-0 的原码表示为：10000000 00000000

对于负数，除符号位以外，其他二进制位在原码基础上取反；而对于正数，采用与原码相同的编码，这种编码称为反码。例如：

-8 的反码表示为：11111111 11110111

-0 的反码表示为：11111111 11111111

在计算机中实际使用的是补码。对于负数，在反码基础上加 1 就形成补码；而正数的补码与原码相同。例如：

-8 的补码表示为：11111111 11111000

-0 的补码表示为：00000000 00000000

+0 的补码表示为：00000000 00000000

可以看出，0 的代码是唯一的；负数补码的补码就是它的原码。

结论：

设有两个整数 a 和 b 对 2^{16} 求余数（模）所得的结果相同（即两数的二进制表示中低 16 位完全相同），那么计算机就认为这两个数是同一个数，计算机的这个特征称为取模原则或模 2^{16} 原则，即任何一个数加上或减去 65 536 所得的结果相同。

2. 字符编码

在数据处理过程中，必然会遇到文字信息的处理问题。与数值数据一样，每个字符在计算机中也要用一个二进制数来表示。目前，计算机中字符编码的国际通用标准是 ASCII 码，即美国信息交换标准编码。与地址编码类似，字符编码（也是一种按照字符表中的排列顺序所进行

的编码。例如，26 个大写英文字母的编码分别为 65（二进制为 01000001）~90（二进制为 01011010）。基本的 ASCII 字符集为 128 个字符，编码占 7 个二进制位，存储时占用一个字节。

C 语言中的字符型数据是以 ASCII 码的形式存储和处理的。需要指出的是，在 DOS 操作系统下，C 语言是不能处理汉字的，但可以处理英文字母形式的汉语拼音。

相关链接

在中文操作系统下，C 语言程序可以处理汉字信息。汉字编码的标准最初为 GB2312—1980，即中华人民共和国信息交换用汉字编码字符集—基本集。该字符集共收录汉字 6 763 个，包含了绝大多数常用的汉字，但有一些仍在使用的冷僻汉字没有被收录。为了解决实际应用中出现的困难，国家相继颁布了不同的汉字编码标准，从 GB2132 到 GBK，从 GBK 到 GB18030，收录的汉字从 6 763 个增加到现在的 27 000 多个。

1.2.5 数据的存储形式

二进制数据的最基本单位为“位”（bit）。在计算机中存储数据时，以 8 个二进制位作为一个存储单位，称为字节（byte）。每个字节的存储空间称为一个存储单元。

在 C 语言中，将 32 位二进制数称为一个长字（long），将 16 位二进制数称为一个短字（short）。一个 short 型数据占用 2 个存储单元，而一个 long 型数据则要占用 4 个存储单元。关于 C 语言的数据类型，将在第 2 章进行详细介绍。

就像人们为教学楼内的每一个房间编号一样，在计算机中，为每个存储单元给出一个二进制编号，称为“地址”。地址实际上也是一个二进制数，用来表示另外一个二进制数的存放位置。这种地址数据在 C 语言程序设计中非常重要，若不会使用这种数据，也就无法学会用 C 语言设计程序。图 1-2 所示为数据的存储示意图。



图 1-2 数据的存储

相关链接

$$1\text{KB} = 2^{10}\text{B} = 1\,024\text{B}$$

$$1\text{MB} = 2^{20}\text{B} = 1\,048\,576\text{B}$$

$$1\text{GB} = 2^{30}\text{B} = 1\,073\,741\,824\text{B}$$

1.3 程序设计语言

1.3.1 概述

程序是人们为解决某一问题而编制的指令集合。计算机在本质上属于机器，完全按照人的



指令进行工作。程序设计语言也称计算机语言，用于编制计算机程序，是人与计算机进行交流的工具。程序设计语言有许多种，每一种语言都规定了各自的语法和语义。本书将以目前较为流行的高级语言——C语言为例介绍程序设计的方法。

通常，要利用程序设计语言描述两个方面的问题：一是需要处理的是哪些数据，这些数据需占用多少存储单元，数据之间存在什么样的联系；二是如何对这些数据进行处理，处理后得到什么结果，怎样将这些结果呈现给用户。第一类问题属于数据结构问题，第二类问题属于算法问题。C语言程序的数据结构是通过数据类型来描述的，读者在以后的学习过程中会逐步接触到各种数据类型，要能够准确地理解和运用这些内容。算法是利用计算机解决一个问题的专门方法，是程序的灵魂，最能体现人的创造性。解决问题的算法最终要用计算机语言来描述，因此，读者应该在准确掌握C语言的基础上，充分学习和研究各种基本算法，只有这样才能学好本课程。对于数据结构和算法的相关内容，可参考其他相关书籍。

1.3.2 程序设计语言的发展与分类

1. 机器语言

在计算机发展的早期进行程序设计时，只能用机器自身规定的二进制指令代码来书写，然后通过穿孔纸带将指令代码转换为数字电信号（光线可以从纸孔中穿过或被阻挡），使计算机完成处理工作。这种以机器指令代码构成的语言称为机器语言，用机器语言编制的程序称为机器语言程序。这种工作方式较为直接，工作步骤较少，容易被人理解。但是，直接用二进制代码编写程序不仅枯燥无味，容易出错，而且各个机器之间的指令代码不能通用。目前，这种编程方式已极少使用。

2. 汇编语言

为了提高编程效率，减少代码错误，人们为每一条二进制的机器指令规定了一个助记符号，例如将表示加法的一串二进制代码记为“ADD”，将表示减法的一串代码记为“SUB”。这样，人们就可以用助记符号来书写程序，描述算法，当程序编制完成以后，再对照指令代码表逐一转换为机器指令。这种以助记符方式编写的程序就是汇编语言程序。目前，人们已经可以利用计算机自身的功能，自动地将汇编语言程序转换为机器语言程序。但是，为了完成这一转换任务，还需要编制一种专用的程序，即汇编程序，这种程序可由IT服务商提供。汇编语言在工业控制领域仍在应用。

3. 高级语言

汇编语言实质上仍然是用机器指令来编写程序，不仅工作量大，而且不同的计算机指令代码也不相同，程序的通用性很差。这对不熟悉机器内部指令的普通用户来说，无疑将造成很大的困难。随着计算机技术的发展和广泛应用，人们发明了可以脱离机器指令来编写程序的高级语言。高级语言只针对需要处理的问题，以接近日常语言（目前仅限英语）的形式来描述算法