工业和信息产业科技与教育专著出版资金
cbjj.phei.com.cn

普通高等教育“十二五”规划教材

# C语言程序设计
## （第2版）

王 苗 林皓波 徐建民 编著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

工业和信息产业科技与教育专著出版资金资助出版

普通高等教育“十二五”规划教材

# C语言程序设计

## （第2版）

王　苗　林皓波　徐建民　编著

電子工業出版社

Publishing House of Electronics Industry

北京・BEIJING

## 内 容 简 介

本书是工业和信息产业科技与教育专著出版资金项目的规划教材。本书以传授程序设计的基础知识和基本方法为出发点，通过构建层次合理、内容适当的教学体系，以通俗易懂的语言、由浅入深地介绍C语言的基本语法和结构化的程序设计方法，旨在提高学生的计算思维能力，培养学生运用计算机编程语言解决实际问题的技能。

全书内容共分8章，分别讲述程序设计基础知识、C程序设计初步、结构化程序设计及控制语句、函数及变量的存储类别、数组、指针、其他构造数据类型和文件等知识。

本书结构清晰、重点明确，语言表述准确、精炼，内容编排强调知识的层次性；例题和习题丰富、应用性强，注重编程技能和计算思维能力的培养。在内容安排上，遵循由简入难、层层递进的原则，降低了学生学习的难度。本书既适用于计算机及相关本、专科专业的程序设计课程教学，也可以作为计算机等级考试和各种程序设计培训的教材。



**图书在版编目（CIP）数据**

C语言程序设计 / 王苗，林皓波，徐建民编著. —2版. —北京：电子工业出版社，2015.6

ISBN 978-7-121-25427-7

Ⅰ. ①C… Ⅱ. ①王… ②林… ③徐… Ⅲ. ①C语言—程序设计—高等学校—教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2015）第010898号

策划编辑：冉　哲
责任编辑：郝黎明
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　邮编　100036
开　　本：787×1 092　1/16　印张：15　字数：384千字
版　　次：2002年8月1版
　　2015年6月2版
印　　次：2015年6月第1次印刷
定　　价：34.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

# 前　言

程序设计作为计算机科学的载体，对于提高学生的计算思维能力、培养学生运用计算机语言和相关开发工具解决问题的技能，有着重要的作用。

20 世纪 70 年代，贝尔实验室在 B 语言的基础上设计出 C 语言，多年来 C 语言被迅速传播，目前已成为世界上最流行、使用最广泛的高级程序设计语言之一。C 语言是一种通用的、结构化的程序设计语言。无论是系统软件、应用软件，还是数据处理、数值计算等，都可以很方便地使用 C 语言来开发。

基于思维训练和技能培养的基本思路，在本书的编写过程中主要遵循了以下原则。

（1）知识讲授和技能培养并重的原则。程序设计不仅是一种知识，同时还是一种技能，因此程序设计课程的教学内容在注重知识讲授的同时，还应注重学生技能（如编程能力、编程风格等）方面的培养。

（2）循序渐进的原则。知识的掌握和技能的训练都是一个循序渐进的过程，学生学习程序设计的知识和技能的培养应该分三步走：掌握基本知识，能够读懂实例，模仿编写程序，逐步达到能够独立编写程序的目的。

（3）分解与综合的原则。人们认识复杂事物的一个基本方法是分解。对学生来说，一门新课程相当于一个需要接受的新事物，采用“分解与综合”的方法可以使他们感觉到学习更容易。因此本书内容的安排基本遵循“模块大小适中”的原则，即每一章、一节乃至一个知识点都尽可能保持适中，将难点适当分解，便于学生掌握。

（4）规范化的原则。程序设计是软件开发过程的一个基本阶段，一开始就注意培养学生良好的程序习惯十分重要，因此本书中的每个例题程序都尽可能有良好的程序风格。

本书是工业和信息产业科技与教育专著出版资金项目的规划教材。本书内容共分 8 章，第 1、2 章介绍程序与算法、C 语言的数据类型、输入/输出函数、运算符及表达式等基础知识；第 3 章介绍 C 语言的控制结构及结构化的程序设计方法；第 4 章介绍函数及变量的存储类别；第 5～7 章介绍数组、结构体等构造数据类型及指针的应用；第 8 章介绍 C 语言的文件操作。其中第 1、3、4、6、7 章由王苗编写，第 2、5 章由林皓波编写，第 8 章由徐建民编写。最后由王苗和徐建民共同完成定稿。

本书作者都是长期从事大学计算机相关课程教学工作的一线教师，具有多年的 C 语言程序设计和相关专业课程的教学经验。但由于作者水平、精力所限，再加上时间紧迫，书中难免有不足甚至错误之处，敬请读者批评指正。

作者

# 目　录

# 第 1 章　程序设计基础知识

## 1.1　计算思维与自动化计算

### 1.1.1　科学思维及其分类

思维是社会人所特有的反映形式，它的产生和发展都同社会实践和语言紧密地联系在一起。思维是人所特有的认识能力，是人的意识掌握客观事物的高级形式。

思维由生命进化而产生，生命在生存过程中进化出意识、思维。思维是人类高级的心理活动形式。思维由思维原料、思维主体和思维工具组成：客观世界提供思维的原料，人脑是思维的主体，认识的反应形式则是思维的工具，这三者结合才能产生思维活动。思维在社会实践的基础上，对感性材料进行分析和综合，通过概念、判断、推理的形式，形成合乎逻辑的理论体系，反映客观事物的本质属性和运动规律。思维过程是一个从具体到抽象，再从抽象到具体的过程，其目的是在思维中再现客观事物的本质，达到对客观事物的具体认识。思维规律由外部世界的规律决定，是外部世界规律在人的思维过程中的反映。

思维有多种分类方法。按照思维的凭借物和解决问题的方式不同，可以把思维分为直观动作思维、具体形象思维和抽象逻辑思维。按照解决问题时的思维方向不同，可以把思维分为收敛思维与发散思维、横向思维与纵向思维等。按照思维的形成和应用领域，可以把思维分为科学思维和日常思维。

科学思维通常是指形成并运用于科学认识活动、对感性认识材料进行加工处理的方式与途径的理论体系；是真理在被认识的统一过程中，对各种科学的思维方法的有机整合，是人类实践活动的产物。科学思维比日常思维更具严谨性和科学性。

在科学认识活动中，科学思维有 3 个基本原则：在逻辑上要求严密的逻辑性，达到归纳和演绎的统一；在方法上要求辩证地分析和综合两种思维方法；在体系上，实现逻辑与历史的一致，达到理论与实践的具体的历史的统一。

从人类认识世界、改造世界的思维方式出发，科学思维又可分为理论思维、实验思维和计算思维三种。

（1）理论思维，又称逻辑思维，是指通过抽象建立描述事务本质的感念，再应用科学的方法探寻概念之间联系的一种思维方法。科学思维既包括以归纳推理为主要内容的归纳逻辑，也包括以演绎推理为主要内容的演绎逻辑，它是一个从具体到抽象，再从抽象到具体的过程，其目的是在思维中再现客观事物的本质，达到对客观事物的具体认识。例如，在数学中，定义是理论思维的灵魂，公理化方法则是理论思维方法。

（2）实验思维，又称实证思维，是通过观察与实验获取规律的一种思维方法，以观察和归纳自然规律为特征。与理论思维不同，实验思维往往需要借助特定的设备获取数据来进行分析。例如，物理、化学、生物等学科中常常用到实验思维的方法。

（3）计算思维，又称构造思维，是从具体的算法设计规范入手，通过算法过程的构造与实施

解决问题的一种方法。计算思维是生活在信息时代的人们应具备的一种基础思维方式，它以设计和构造为特征，以计算机学科为代表，是思维过程的计算模拟方法论。计算机不仅为不同专业领域提供了解决专业问题的有效方法和手段，而且提供了一种独特的处理问题的思维方式。

### 1.1.2 计算思维的概念与特征

2006 年 3 月，美国卡内基·梅隆大学的周以真教授，在美国计算机权威期刊 Communications of the ACM 上给出了计算思维的定义：计算思维是运用计算机科学的基础概念进行问题求解、系统设计及人类行为理解等涵盖计算机科学之广度的一系列思维活动。

为了让人们更易于理解，周教授进一步将计算思维定义为：通过约简、嵌入、转化和仿真等方法，把一个看起来困难的问题重新阐释成一个我们知道问题怎样解决的方法；是一种递归思维，是一种并行处理，是一种把代码译成数据又能把数据译成代码，是一种多维分析推广的类型检查方法；是一种采用抽象和分解来控制庞杂的任务或进行巨大复杂系统设计的方法，是基于关注分离的方法（SoC 方法）；是一种选择合适的方式去陈述一个问题，或对一个问题的相关方面建模使其易于处理的思维方法；是按照预防、保护及通过冗余、容错、纠错的方式，并从最坏情况进行系统恢复的一种思维方法；是利用启发式推理寻求解答，即在不确定情况下的规划、学习和调度的思维方法；是利用海量数据来加快计算，在时间和空间之间、在处理能力和存储容量之间进行折中的思维方法。

计算思维吸取解决问题所采用的数学思维方法，结合现实世界中复杂系统设计的工程思维方法，是涉及复杂性、智能、心理、人类行为的理解等的一般科学思维方法。计算思维建立在计算过程中个人的能力和限制之上，计算方法和模型使得我们可以去处理一些原本无法由个人独立解决的问题。

计算思维具有以下特性。

（1）概念化，不是程序化。计算机科学不是计算机编程。像计算机科学家那样去思维意味着远远不止能为计算机编程，还要求能够在抽象的多个层次上思维。

（2）基础性，不是机械的技能。基础的技能是每个人为了在现代社会中发挥职能所必须掌握的。生搬硬套机械的技能意味着机械地重复。具有讽刺意味的是，只有当计算机科学解决了人工智能的宏伟挑战——使计算机像人类一样思考之后，思维才会变成机械的生搬硬套。

（3）计算思维是人的思维，不是计算机的思维。计算思维是人类求解问题的一条途径，但绝非试图使人类像计算机那样来思考。计算机枯燥且沉闷，人类聪颖且富有想象力。配置了计算设备，我们就能用自己的智慧去解决那些计算时代之前不敢尝试的问题，就能建造那些其功能仅仅受制于我们想象力的系统。

（4）数学和工程思维的互补与融合。计算机科学在本质上源自数学思维，因为像所有的科学一样，它的形式化解析基础建立于数学之上。计算机科学又从本质上源自工程思维，因为我们建造的是能够与实际世界互动的系统。基本计算设备的限制迫使计算机科学家必须计算性地思考，不能只是数学性地思考。构建虚拟世界的自由使我们能够超越物理世界去建造各种系统。

（5）计算思维是思想，不是人造品。不只是我们生产的软硬件人造品将以物理形式到处呈现，并时时刻刻触及我们的生活，更重要的是还将有我们用以接近和求解问题、管理日常生活、与他人交流和互动的计算性的概念。

（6）计算思维面向所有人和所有地方。计算思维日渐渗透到其他学科中，并渗透到我们的

生活中。例如，在统计学中，采用机器学习理论中的统计学习方法，用于统计各类问题的规模。在生物学中，利用计算机科学在海量序列数据中搜索寻找模式规律的本领，如用合适的数据结构与算法，描述蛋白质的结构等。还有，计算博弈理论正改变着经济学家的思考方式，纳米计算改变着化学家的思考方式，量子计算改变着物理学家的思考方式。

### 1.1.3 计算思维的举例——自动化计算

学习程序设计是理解计算机工作特点的最好途径，程序设计课程的内容最能够体现语言级的问题求解方法，是计算思维能力培养的重要内容。以下举例可以体现采用计算思维解决特定问题的方式。

**1．求最大值问题**

首先看求 4 个整数中的最大值。假设这 4 个整数是 $a$、$b$、$c$、$d$，采用直观思维的方法：

（1）先求 $a$、$b$ 的最大值，假设用 x 表示；

（2）再求 $c$、$d$ 的最大值，假设用 $y$ 表示；

（3）比较 $x$、$y$ 的大小，大的一个即为所求最大值。

当数据量比较小时，这种方法还可以便捷地解决问题。但若数据量比较大，如求 10 个数、100 个数、100000 个数的最大值，这样的方法显然就不适用了。

仍然求 $a$、$b$、$c$、$d$ 中的最大值，假设有一个变量 MAX，用来存放最大值。用以下思路进行求解。

（1）令 MAX 的值为 $a$ 的值。

（2）比较 $b$ 和 MAX 的大小，若 $b$ >MAX，令 MAX 的值为 $b$ 的值。

（3）比较 $c$ 和 MAX 的大小，若 $c$ >MAX，令 MAX 的值为 $c$ 的值。

（4）比较 $d$ 和 MAX 的大小，若 $d$ >MAX，令 MAX 的值为 $d$ 的值。

这样，MAX 中即可获得 4 个数中的最大值。

这种方法下，当数据更多时，求最大值的方法是不变的，只是多了若干次比较。

另外，若采用第一种方法，求 $n$ 个数中的最大值需要另外 $n$−1 个变量来表示中间数据。而第二种方法，无论数据量多大，在比较过程中，只需用到一个变量 MAX。显然第二种方法更好，更适用一般情况。

**2．排序问题**

排序（Sorting）是计算机程序设计中经常使用的一种重要操作，其功能是将一个数据元素集合或序列重新排列成一个按数据元素某个数据项值进行有序排列的序列。在日常生活中，排序的例子很常见，如电话号码簿、词典、仓库清单等，都被整理得井井有条，整理的过程就是排序。

排序的方法很多，下面介绍一种思路比较简单的排序方法。

简单选择排序，其过程为：第一趟，从 $n$ 个记录中找出关键码最小的记录与第一个记录交换；第二趟，从第二个记录开始的 $n$−1 个记录中再选出关键码最小的记录与第二个记录交换；以此类推，第 $i$ 趟，则从第 $i$ 个记录开始的 $n$-$i$+1 个记录中选出关键码最小的记录与第 $i$ 个记录交换，直到整个序列按关键码有序排列。

例如，有以下序列：25　36　30　36　10　56　12，请按简单选择排序方法进行排序。排序过程如图 1.1 所示。

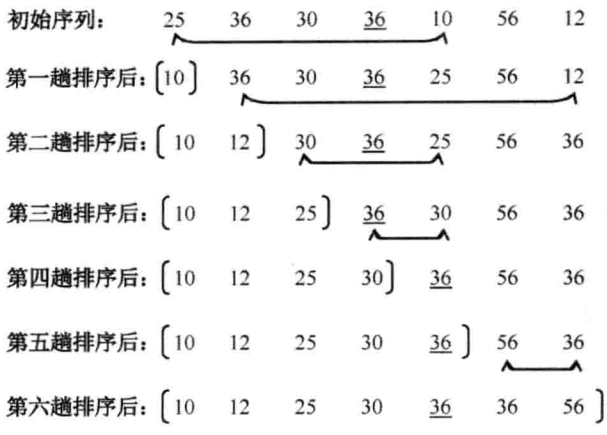



图 1.1　简单选择排序示例

## 1.2　计算机内的数据表示

随着计算机应用领域的日益广泛，计算机除了用于进行复杂的科学计算，还能进行文字的处理、图像的识别、声音的加工等。数字、汉字、图像和声音的表象千差万别，但对于计算机而言，它们都被称为数据或信息。在计算机科学中，数据是指所有能被计算机识别、存储和处理的符号的总称。

### 1.2.1　数制及其转换

**1．进位计数制**

进位计数制是常用的计数方法。进位计数制是采用有限个数码来表示数据，数据中各个数字所处的位置决定它的权值，每个数字所表示的数值就等于该数字本身乘以它的位置所代表的权值。

各数位只允许选用有限个数码，每一数位所能表示的最大值等于可选用的最大数码乘以其权值，超过这个值就要向高位进位。允许选用的数码的个数就是计数制的基数。

一般，基数为 r 的 r 进制数：

$$(N)_r=K_nK_{n-1}\cdots K_1K_0 \quad K_{-1}K_{-2}\cdots K_{-m}$$

的数值可表示为：

$$(N)_r=K_n\times r^n+K_{n-1}\times r^{n-1}+\cdots+K_1\times r^1+K_0\times r^0+K_{-1}\times r^{-1}+K_{-2}\times r^{-2}+\cdots+K_{-m}\times r^{-m}$$

其中，数位 $K_i$（$-m\leqslant i\leqslant n$）的数值可选择 $r$ 个数码中的某一个，其权值是 $r^i$。

1）十进制计数法

日常生活中用的最多的是十进制计数法。例如，十进制数 $(1286.75)_{10}=1\times10^3+2\times10^2+8\times10^1+6\times10^0+7\times10^{-1}+5\times10^{-2}$。其各位的权值分别为 $10^3$、$10^2$、$10^1$、$10^0$、$10^{-1}$、$10^{-2}$。

十进制计数法“逢十进一”，即基数为 10；十进制的数码为 0、1、2、3、4、5、6、7、8、9。

例如，个位数的权值是 $1(10^0)$，那么个位数所能表示的最大数值为 $9\times10^0=9$，超过 9 就要向十位进位；十位数的权值是 $10(10^1)$，那么十位数所能表示的最大数值为 $9\times10^1=90$，超过 90 就要向百位进位。

2）二进制计数法

在计算机内部，数据是用二进制计数法表示的。计算机内用电子器件的两种不同状态来表

示数字信息，如用高电平来表示 1，用低电平表示 0，因此对于计算机的物理实现而言，用二进制表示数据更方便。例如，二进制数$(1001101.101)_2=1\times2^6+0\times2^5+0\times2^4+1\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$。

显然，其各位的权值分别为$2^6$、$2^5$、$2^4$、$2^3$、$2^2$、$2^1$、$2^0$、$2^{-1}$、$2^{-2}$、$2^{-3}$。

二进制计数法“逢二进一”，即基数为 2；二进制的数码为 0、1。也就是说，二进制数的每一位或者为 0，或者为 1；超过 1 就要向高位进位。

3）八进制计数法与十六进制计数法

采用二进制计数法表示数据的一个缺点是所需位数很多，容易出错，因此为了便于阅读和书写，常采用八进制数与十六进制数来代替二进制数。

例如，八进制数

$$(621)_8=6\times8^2+2\times8^1+6\times8^0$$

其各位的权值分别为$8^2$、$8^1$、$8^0$。

八进制计数法“逢八进一”，即基数为 8，八进制计数法的数码为 0、1、2、3、4、5、6、7。

这 8 个数码相当于十进制数的 0～7，接下来下一位数字就向高位进位，即$(10)_8$，相当于十进制数 8，那么$(11)_8$相当于十进制数 9。

又如，十六进制数

$$(8A1F)_{16}=8\times16^3+10\times16^2+1\times16^1+15\times16^0$$

其各位的权值分别为$16^3$、$16^2$、$16^1$、$16^0$。

十六进制计数法“逢十六进一”，即基数为 16。十六进制计数法的数码为 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。

十六进制中有 16 个数码，但从 10 开始到 15 就没有阿拉伯数字可用了，因此规定用字母 A(a)、B(b)、C(c)、D(d)、E(e)、F(f)来表示 10、11、12、13、14、15。

**2．数制转换**

1）任意进制转换为十进制

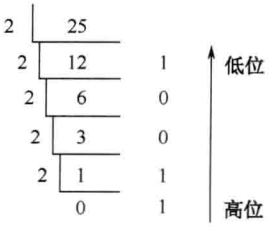
由 r 制数转换为十进制数可按照如下公式进行多项式展开求和即可。

$$K_nK_{n-1}\cdots K_1K_0\quad K_{-1}K_{-2}\cdots K_{-m}=$$
$$K_n\times r^n+K_{n-1}\times r^{n-1}+\cdots+K_1\times r^1+K_0\times r^0+K_{-1}\times \mathrm{r}^{-1}+K_{-2}\times \mathrm{r}^{-2}+\cdots+K_{-\mathrm{m}}\times r^{-\mathrm{m}}$$

2）十进制转换为任意进制

可以采用除基取余法将十进制整数转换为 r 进制整数：将十进制整数除以 r，得到商和余数，余数对应为 r 进制数低位的值；继续让商再除以 r，得到商和余数，重复此操作，直至商为 0，如此得到的一系列余数就是所求的 r 进制数的各位数字，先得到的是低位，后得到的是高位。

例如，将$(25)_{10}$转换为二进制整数。




因此，$(25)_{10}=(11001)_2$。

可采用乘基取整法将十进制小数转换为 $r$ 进制小数：将十进制小数乘以 $r$，去掉乘积的整数部分，再将余下的纯小数乘以 $r$，重复此操作，直至乘积等于 0 或达到所需的精度为止，如此得到的一系列整数就是 $r$ 进制小数的各位数字，先得到的是高位，后得到的是低位。

例如，将$(0.625)_{10}$转换为二进制小数。

0.625×2=1.25　　1　高位

0.25×2=0.5　　0

0.5×2=1　　1　低位

因此，$(0.625)_{10}=(0.101)_2$。

由于整数和小数的转换方法截然不同，将十进制数转换为 $r$ 进制数时，整数部分和小数部分要分开来进行转换。

$r$ 进制小数能精确地转换为十进制小数，但十进制小数往往不能精确地转换为 $r$ 进制小数。例如，$(0.1)_{10}=(0.0001100110011)_2$。

3）二进制与八进制、十六进制之间的转换

对于一个二进制数，只要依次（整数部分由低位到高位，小数部分由高位到低位）将其每 3 位或 4 位分成一组，就可以直接转换为八进制数或十六进制数。

例如，二进制数$(1100101111010011.01101)_2$

若按 3 位一组划分，就可以直接写出其对应的八进制数：

1　100　101　111　010　011．011　010

1　4　5　7　2　3．3　2

即$(1100101111010011.01101)_2=(145723.32)_8$。

若按 4 位一组划分，也可以直接写出其对应的十六进制数：

1100　1011　1101　0011．0110　1000

C　B　D　3．6　8

即$(1100101111010011.01101)_2=(CBD3.68)_{16}$。

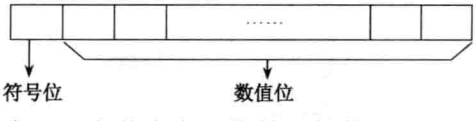
反之，将八进制数中的每一位用 3 位二进制数表示，将十六进制数中的每一位用 4 位二进制数表示，也能转换为对应的二进制数。

## 1.2.2　原码、反码及补码

计算机中，数据的最小单位是“位”，一般将 8 位二进制位定义为 1 字节，计算机中的存储量就是以字节来计算的。

在计算机内，不仅数值是用二进制数表示的，符号也是用二进制数表示的。一般规定：用 0 表示“＋”，用 1 表示“－”；符号位放在数值位之前。一个数连同其符号在机器中的二进制数表示形式称为机器数，它所代表的数值称为机器数的真值。

机器数的一般格式如下。




为简便起见，下面讨论中用 1 字节来表示带符号的数据。

在计算机中，带符号的数的表示方法有 3 种：原码、反码和补码。

**1．原码**

原码表示法是符号位用 0 表示正数，用 1 表示负数，数值位表示数值本身。例如，$[+27]_{原}=$

00011011，$[-27]_{原}$=10011011。

在原码表示法中，0 的表示方法不唯一，即$[+0]_{原}$=00000000，$[-0]_{原}$=10000000。

**2．反码**

反码表示法中正数与负数的表示方法不同，正数的反码与原码同形。例如：

$[+27]_{反}$=00011011。

负数的反码为：符号位仍为 1，数值位是对原码取反。例如：

$[-27]_{反}$=11100100。

在反码表示法中，0 的表示也不唯一，即$[+0]_{反}$=00000000，$[-0]_{反}$=11111111。

**3．补码**

用原码表示数据，简单明了，但用原码进行加减运算却很不方便。例如，当两个数的符号相同时，可以数值位相加，结果符号不变。但当两个数的符号不同时，加法运算实际上要转换为减法进行；为了进行减法，应先判断两数的绝对值，让绝对值大的数减去绝对值小的数，运算结果的符号与绝对值大的数的符号相同。计算机实现上述过程是相当烦琐的。

因此，在计算机中采用补码表示法来表示数据。采用补码表示数据，能够简化设计与运算，可以将减法运算转化为加法运算。

下面以时钟为例说明补码的原理，假设当前时刻是 3 点，若问两小时前是几点，那么可以将时针向前拨两小时，即 3－2＝1，得到 1 点；还可以将时针向后拨 10 小时，即 3＋10＝13＝12＋1，从时钟上看，还是 1 点。从而，3 减去 1，与 3 加上 10 能够达到同样的效果。这是因为时钟的刻度是以 12 为周期的，超过 12 就再从头开始计数，则称 2 和 10 是互补的，即减去 2 就相当于加上 10。这就是减法变加法的原理。

在计算机中，以定长的存储单元来表示数据，因此所能表示数据的范围是有限的，超过它的表示范围后，高位就会溢出。

正数的补码与原码、反码同形。例如：

$[+18]_{原}$=$[+18]_{反}$=$[+18]_{补}$= 00010010。

负数的补码：符号位为 1，数值位等于原码的数值位取反，再加 1，或者说$[x]_{补}$=$[x]_{反}$+1。

例如，

$[-18]_{原}$= 10010010，$[-18]_{反}$= 11101101，$[-18]_{补}$= 11101110。

采用补码表示数据，使得正、负的关系转换成了一种纯数值的关系。补码形式的数据进行运算时，符号位和数值位一样，均参与运算，不必考虑它特别的含义（正、负）。

例如，计算－36＋58 的值。

$[-36]_{补}$= 11011100，$[+58]_{补}$= 00111010。

```
          11011100
        + 00111010
        ----------
溢出 ←  1 00010110
```

显然，$(11011100)_2+(00111010)_2=(00010110)_2=(22)_{10}$，从而验证了－36＋58＝22。

可见，采用补码表示的数据进行运算的规则是很简单的。而且，在补码表示法中，0 的表示是唯一的，若用 1 字节存放数据，则规定

$[0]_{补}$=00000000，$[-128]_{补}$=10000000。

### 1.2.3 定点数及浮点数

一个数据可能含有整数部分，又含有小数部分，这就涉及小数点的表示问题。对小数点的表示方法有两种：定点数和浮点数。

**1．定点数**

若对于所有的数据都限定小数点的位置固定不变，就是定点表示法。

1）定点小数

约定小数点的位置在符号位之后，数值部分之前，则定点小数的表示形式如下。


计算机中并不存放小数点，只是隐含约定小数的位置。

假设用 1 字节存放数据，

则符号位占 1 位，数值位占 7 位，那么，

01000101 表示二进制小数 0.1000101，10110001 表示二进制小数-0.0110001。

若计算机的字长为 $n$，其中一位是符号位，$n$-1 位是数值位，那么，当数值部分的最后一位 $x_{n-1}$ 是 1，其余位是 0 时，数 $x$ 的绝对值最小为 $|x|_{\min}=2^{-(n-1)}$；当数值部分全是 1 时，数 $x$ 的绝对值最大为 $|x|_{\max}=1-2^{-(n-1)}$。

因此，计算机所能表示的定点小数的绝对值范围是 $2^{-(n-1)} \sim 1-2^{-(n-1)}$。

2）定点整数

约定小数点的位置在数值部分之后，即数据是纯整数。其表示形式如下：


字长为 $n$ 的定点整数，考虑其绝对值，最小的数的数值部分全为 0，最大的数的数值部分全为 1，因此其所能表示的数据的绝对值范围是 $0 \sim 2^{n-1}-1$。

在实际处理数据时，一个二进制数 $N$ 可能既有整数部分又有小数部分，那么就需要将其乘上一个倍数以转换成规定的定点整数或定点小数的形式：

$$N=X\times 2^{i}$$

其中，$X$ 为规定的定点整数或定点小数，$i$ 的值一旦选取就固定不变，这是定点数的特点。$i$ 的取值很重要：对于定点小数，$i$ 若选取大了，数据可能溢出；$i$ 若选取小了，可能会有损数据的精度。

**2．浮点数**

为了协调数据的表示范围与精度的要求，提出数据的浮点表示方法，即小数点的位置不再固定不变，而是根据需要浮动。浮点数的格式如下：