



高级语言程序设计

GAOJI YUYAN CHENGXU SHEJI

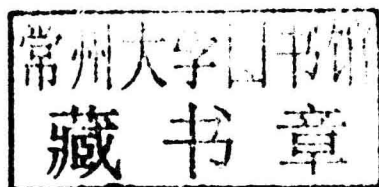
主 编 宋宏光 高心丹

副主编 徐正芳

主 审 任洪娥

高级语言程序设计

主 编 宋宏光 高心丹
副主编 徐正芳
主 审 任洪娥



东北林业大学出版社
• 哈尔滨 •

版权专有 侵权必究

举报电话：0451-82113295

图书在版编目 (CIP) 数据

高级语言程序设计 / 宋宏光, 高心丹主编. — 哈尔滨 :
东北林业大学出版社, 2014. 8

(东北林业大学优秀教材丛书)

ISBN 978-7-5674-0486-1

I. ①高… II. ①宋…②高… 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 (2014) 第 197483 号

责任编辑: 任兴华

封面设计: 乔鑫鑫

出版发行: 东北林业大学出版社

(哈尔滨市香坊区哈平六道街 6 号 邮编: 150040)

印 装: 哈尔滨市石桥印务有限公司

开 本: 787mm×960mm 1/16

印 张: 16

字 数: 287 千字

版 次: 2014 年 8 月第 1 版

印 次: 2014 年 8 月第 1 次印刷

定 价: 35.00 元

前 言

高级语言程序设计在世界范围内都是高等学校的一门基本的计算机教育课程,而 C 语言的功能丰富、表达能力强、应用广、运行效率高等特点不仅使其成为高级语言程序设计教程中首选的一门程序设计语言,还为后续课程开展发挥着重要的基础教学作用。

在以往教学过程中,学生普遍反映 C 语言比较难学,本书编者根据多年的教学实践经验,提出程序设计学习“要遵循整体结构概念、软件工程概念,程序设计过程应先实践、后总结、再归类分析、继续实践推广”的理念。

按照上述理念,本书编者在前人的研究基础上进一步深入发掘与整理,在书稿撰写过程中结合本科教学实践工作,并引入软件工程基本理念,一改以往教材单纯片面地侧重于程序设计语言的细节而忽略了系统整体的结构,教材始终贯彻了“实践—认识—再实践”的理念,将学习程序设计的过程与软件工程实施过程进行了有效的统一。书中的每一个概念、每一个例题、每一个案例都经过了作者的深思熟虑与实际调试。

本书内容先进,概念清晰,讲解详尽且通俗易懂,书中的概念、例题都凝聚了作者多年积累的经验,由浅入深,由简单到复杂,前后贯穿,达到举一反三的目的。读者在学习过程中能轻松入门并在总结中快速提高,避免入门即陷入烦琐的细节问题,导致学习兴趣下降。

本书第 1 章、第 2 章、第 3 章、第 4 章、第 5 章、第 6 章、第 9 章由黑龙江工程学院徐正芳老师编写,第 7 章、第 8 章由东北林业大学高心丹老师编写,第 10 章、第 11 章、第 12 章由东北林业大学宋宏光编写,全书由宋宏光进行最后统稿,由任洪娥审定。在本书编写过程中,东北林业大学信息学院高级语言重点课建设组的老师们给予了大力支持,并提出了宝贵意见,在此表示衷心的感谢!

由于编写水平有限,书中难免有不妥和错误之处,恳请广大读者批评指正(编者邮箱:ghsong_57@hotmail.com.)。

编者

2013 年 12 月

目 录

1 计算机系统概述	(1)
1.1 计算机系统组成	(1)
1.2 如何学习 C 语言程序设计	(4)
习题	(4)
2 基本数据类型	(6)
2.1 C 语言的数据类型	(6)
2.2 计算机中各种进制数据的表示形式	(9)
2.3 整型数据	(10)
2.4 实型数据	(14)
2.5 字符型数据	(17)
2.6 变量赋初值	(20)
2.7 各类数值型数据(整型、实型、字符型)的混合运算	(20)
习题	(22)
3 运算符与表达式	(26)
3.1 C 运算符简介	(26)
3.2 算术运算符和算术表达式	(27)
3.3 赋值运算符和赋值表达式	(28)
3.4 自增、自减运算符	(31)
3.5 逗号运算符和逗号表达式	(32)
3.6 小结	(33)
习题	(34)
4 数据的输入与输出	(39)
4.1 C 语句概述	(39)
4.2 数据输入与输出	(42)
4.3 格式输入/输出	(44)
习题	(50)
5 结构化程序设计	(55)
5.1 简介	(55)
5.2 算法	(55)

习题	(60)
6 程序控制	(61)
6.1 关系运算	(61)
6.2 逻辑运算	(62)
6.3 选择程序控制	(65)
6.4 循环程序控制	(73)
6.5 break 和 continue 语句	(83)
习题	(89)
7 函数	(99)
7.1 函数概述	(99)
7.2 函数的一般定义形式	(101)
7.3 函数的参数	(102)
7.4 函数的调用	(104)
7.5 函数的嵌套和递归	(106)
7.6 变量的作用域(有效范围)	(110)
7.7 变量的生存期(存储类别)	(114)
7.8 内部函数和外部函数	(117)
习题	(118)
8 数组	(125)
8.1 概述	(125)
8.2 一维数组	(126)
8.3 二维数组	(134)
8.4 一维字符数组	(139)
8.5 常用字符串函数	(142)
8.6 字符数组应用举例	(144)
习题	(146)
9 编译预处理	(152)
9.1 概述	(152)
9.2 宏定义	(152)
9.3 文件包含	(156)
9.4 条件编译	(157)
习题	(158)
10 指针	(161)
10.1 变量的地址与变量的指针	(161)

10.2	一维数组与指针.....	(168)
10.3	二维数组与指针.....	(173)
10.4	字符串与指针.....	(178)
10.5	函数与指针.....	(181)
10.6	指针数组与指向指针的指针.....	(186)
10.7	指针运算举例.....	(189)
	习题	(190)
11	结构体、联合体和位运算	(197)
11.1	结构体.....	(197)
11.2	联合体.....	(207)
11.3	用 typedef 定义类型	(211)
11.4	位运算.....	(212)
	习题	(213)
12	文件	(220)
12.1	文件概述.....	(220)
12.2	文件类型指针.....	(221)
12.3	文件的操作.....	(222)
	习题	(231)
	参考文献	(234)
	附录	(235)

1 计算机系统概述

计算机系统已发展成为一个庞大的家族,其中的每个成员,尽管在规模、性能、结构和应用等方面存在着很大的差别,但是它们的基本结构是相同的。计算机系统包括硬件系统和软件系统两大部分。硬件系统由中央处理器、内存储器、外存储器和输入/输出设备组成,如图 1-1 所示。

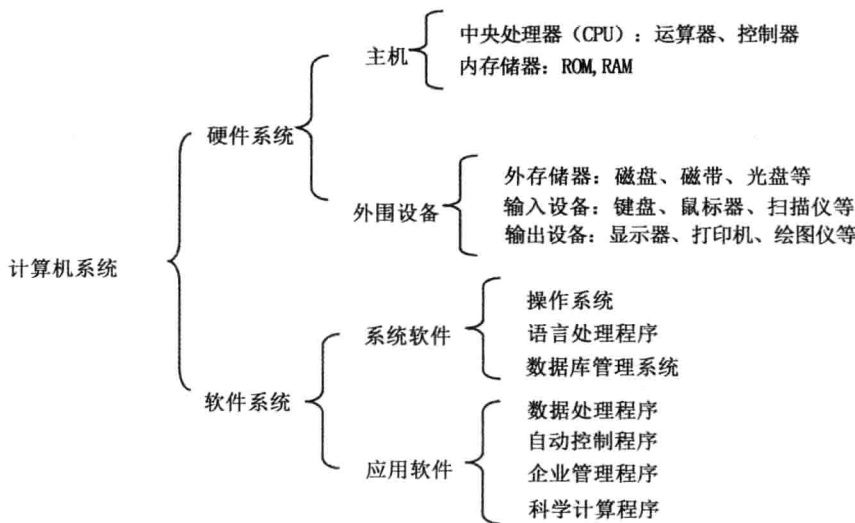


图 1-1 计算机系统组成

计算机通过执行程序而运行。计算机工作时,软、硬件协同工作,两者缺一不可。

1.1 计算机系统组成

1.1.1 硬件系统

硬件系统是构成计算机的物理装置,是指在计算机中看得见、摸得着的有形实体。在计算机的发展史上做出杰出贡献的著名应用数学家冯·诺依曼(von Neumann)与其他专家于 1945 年为改进 ENIAC,提出了一个全新的存储

程序的通用电子计算机方案。这个方案规定了新机器由 5 个部分组成:运算器、逻辑控制装置、存储器、输入和输出,并描述了这 5 个部分的职能和相互关系。这个方案与 ENIAC 相比,有两个重大改进:一是采用二进制;二是提出了“存储程序”的设计思想,即用记忆数据的同一装置存储执行运算的命令,使程序的执行可自动地从一条指令进入到下一条指令。这个概念被誉为计算机史上的一个里程碑。计算机的存储程序和程序控制原理被称为冯·诺依曼原理,按照上述原理设计制造的计算机称为冯·诺依曼机,如图 1-2 所示。

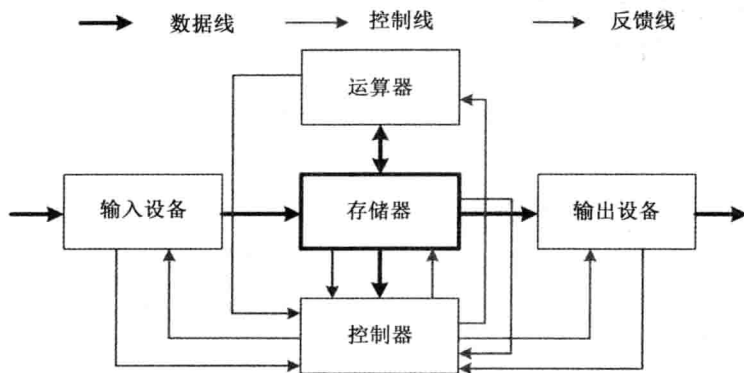


图 1-2 计算机逻辑组成

概括起来,冯·诺依曼结构有 3 条重要的设计思想:

- (1) 计算机应由运算器、控制器、存储器、输入设备和输出设备 5 大部分组成,每个部分具有一定的功能;
- (2) 以二进制的形式表示数据和指令,二进制是计算机的基本语言;
- (3) 程序预先存入存储器中,使计算机在工作中能自动地从存储器中读取程序指令并加以执行。

1.1.2 计算机的基本工作原理

1.1.2.1 计算机的指令系统

指令是能被计算机识别并执行的二进制代码,它规定了计算机能完成的某一种操作。一条指令通常由如下两个部分组成。

(1) 操作码:它是指明该指令要完成的操作,如存数、取数等。操作码的位数决定了一个机器指令的条数。当使用定长度操作码格式时,若操作码位数为 n ,则指令条数可有 2^n 条。

(2) 操作数:它指操作对象的内容或者所在的单元格地址。操作数在大多数情况下是地址码,地址码有 0~3 位。从地址代码得到的仅是数据所在的

地址,可以是源操作数的存放地址,也可以是操作结果的存放地址。

1.1.2.2 计算机的工作原理

计算机的工作过程实际上是快速地执行指令的过程。当计算机在工作时,有两种信息在流动,一种是数据流,另一种是控制流。

数据流是指原始数据、中间结果、结果数据、源程序等。控制流是由控制器对指令进行分析、解释后向各部件发出的控制命令,用于指挥各部件协调地工作。

下面以指令的执行过程来认识计算机的基本工作原理。计算机的指令执行过程分为如下几个步骤:

(1)取指令:从内存存储器中读取出指令送到指令寄存器。

(2)分析指令:对指令寄存器中存放的指令进行分析,由译码器对操作码进行译码,将指令的操作码转换成相应的控制电信号,并由地址码确定操作数的地址。

(3)执行指令:它是由操作控制线路发出的完成该操作所需要的一系列控制信息,以完成该指令所需要的操作。

(4)为执行下一条指令做准备:形成下一条指令的地址,指令计数器指向存放下一条指令的地址,最后控制单元将执行结果写入内存。

上述完成一条指令的执行过程叫作一个“机器周期”。计算机在运行时,CPU 从内存读取一条指令到 CPU 内执行,指令执行完,再从内存读取下一条指令到 CPU 执行。CPU 不断地取指令、分析指令、执行指令,再取下一条指令,这就是程序的执行过程。总之,计算机的工作就是执行程序,即自动连续地执行一系列指令,而程序开发人员的工作就是编制程序,使计算机不断地工作。

1.1.3 计算机软件系统

软件系统是指使用计算机所运行的全部程序的总称。软件是计算机的灵魂,是发挥计算机功能的关键。有了软件,人们可以不必过多地去了解机器本身的结构与原理,可以方便灵活地使用计算机,从而使计算机有效地为人类工作、服务。

随着计算机应用的不断发展,计算机软件在不断积累和完善的过程中,形成了极为宝贵的软件资源。它在用户和计算机之间架起了桥梁,给用户的操作带来极大的方便。

在计算机的应用过程中,软件开发是个艰苦的脑力劳动过程,软件生产的自动化水平还很低,所以许多国家投入大量人力从事软件开发工作。正是有

了内容丰富、种类繁多的软件,使用户面对的不仅是一部实实在在的计算机,而且还包含许多软件的抽象的逻辑计算机(称为虚拟机),这样人们可以采用更加灵活、方便、有效的手段使用计算机。从这个意义上说,软件是用户与计算机的接口。

在计算机系统中,硬件和软件之间并没有一条明确的分界线。一般来说,任何一个由软件完成的操作也可以直接由硬件来实现,而任何一个由硬件执行的指令也能够用软件来完成。硬件和软件有一定的等价性,如图像的解压,以前低档微机是用硬件解压,现在高档微机则用软件来实现。

软件和硬件之间的界线是经常变化的。要从价格、速度、可靠性等多种因素综合考虑,来确定哪些功能用硬件实现合适,哪些功能由软件实现合适。

1.2 如何学习 C 语言程序设计

- 学习语言的语法结构。
- 学习语法是为了更好地应用。
- 在实践中学习语法。
- 课上教学分模块进行,课下要求能够举一反三,最终能够搭建完整功能的程序。

习题

1.1 自学并简答

- (1)设法查找有关计算机组成、工作原理及计算机发展方面的书籍。
- (2)查找程序设计语言及发展的书籍,了解计算机程序设计方面的计算机术语。
- (3)了解 C 语言的主要特点和优点。
- (4)设法查找 ANSI C 标准或者中国国家标准 GB/T 12272-94《程序设计语言 C》,浏览这些标准的目录,了解在定义一个程序语言时需要说明哪些内容。
- (5)了解 C 语言的一些常用编译系统或是集成开发环境,熟悉它们的使用方法和基本操作。

1.2 思维训练——写出下面题目的计算过程

(1) 有两个瓶子 A 和 B, 分别盛放醋和酱油, 写出符合情理的计算过程将它们互换。

(2) 有三个数, 写出找出其中最大数的计算过程。

(3) 写出求一元二次方程 $ax^2 + bx + c = 0$ 的根的计算方法。

(4) 写出判断一个年份是闰年的方法。

(5) 写出用以求 100 以内能被 3 和能被 7 整除的所有数之和的计算过程。

2 基本数据类型

2.1 C 语言的数据类型

程序、算法处理的对象是数据。数据通常是以某种特定的形式存在(如整数、实数、字符)的,那么数据如何放在计算机中处理,又是以什么样的途径放入计算机去处理表示呢? C 语言采取的途径是使用数据类型与变量来实现将数据交给计算机去处理。

2.1.1 数据类型

(1)数据类型是某一类数据的共同特征。

(2)数据类型隐含地说明了该类型数据在计算机内应用时所需要的存储空间的大小。

(3)数据类型隐含地说明了该类型数据在计算机内能够进行的运算操作。

C 语言的数据结构是以数据类型的形式体现。也就是说,C 语言中数据是有类型的,数据的类型简称数据类型,如整型数据、实型数据、整型数组类型、字符数组类型分别代表我们常说的整数、实数、数列、字符串。C 语言的数据类型见图 2-1。

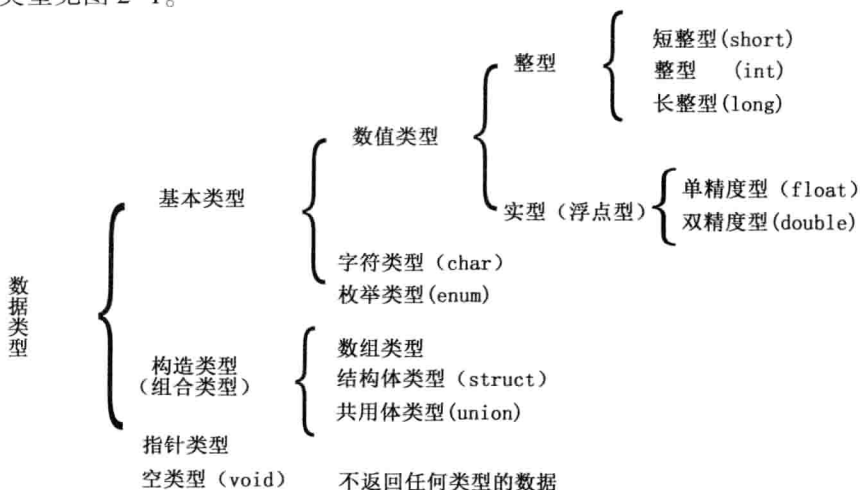


图 2-1 C 语言的数据类型

2.1.2 变量

程序设计语言中的变量是数据类型的具体实现,是数据类型的实例。变量的基本特点如下。

- (1) 变量一定是某一种数据类型的变量,拥有该数据类型的全部特征。
- (2) 变量是该数据类型的数据在计算机内的具体表现。
- (3) 变量本质上是计算机内存空间的一种符号表示,程序设计语言借助于这种符号完成对内存空间的使用。这个符号也称为变量名。
- (4) 在 C 程序设计中变量必须遵循“先定义,后使用”的原则。
- (5) 变量所对应的内存空间中的数据,在程序运行过程中可以随时变化调整,故而称为变量。
- (6) 变量所对应的内存空间的获得,由系统自动决定空间的位置以及分配的时机,见图 2-2。



图 2-2 C 语言的变量

2.1.3 C 语言的标识符

2.1.3.1 C 语言字符集

字符是 C 语言最基本的元素,C 语言字符集由字母、数字、空白、标点和特殊字符组成(在字符串常量和注释中还可以使用汉字等其他图形符号)。由字符集中的字符可以构成 C 语言进一步的语法成分(如标识符、关键词、运算符等)。

- (1) 字母:A-Z,a-z;
- (2) 数字:0-9。

在 C 语言中约定,标识符符号的组合仅能由字母、数字和下划线 3 种要素构成,其中标识符的第一个符号必须是字母或下划线。

需要注意的是,C 语言对有些符号的组合,已经事先约定为表示特殊意义的用途,我们把这些标识符称为保留字或者关键字,这些标识符不能再继续拿来重新定义使用。

2.1.3.2 标识符(名字)

用来标识变量名、符号常量名、函数名、数组名、类型名等实体(程序对象)的有效字符序列。标识符由系统事先约定或者用户自定义(取名字)。C

语言标识符定义规则:

(1) 标识符只能由字母、数字和下划线三种字符组成,且第一个字符必须为字母或下划线。

例如:

合法的标识符: `user _user name x_1 str1`

不合法的标识符: `u ser u&ser n * ame`

(2) 大小写敏感。C 程序员习惯变量名小写,常量名大写,但不绝对,如用 windows 编程,应当使用匈牙利表示法(大小写混用,每个单词词首第一个大写,其余小写,如 WinMain)。

例如: `sum` 不同于 `Sum`; `BOOK` 不同于 `book`。

(3) ANSI C 没有限制标识符长度,但各个编译系统都有自己的规定和限制(TC 32 个字符,MSVC 8 个字符)。

例如: `student_name`, `student_number` 如果取 8 个,这两个标识符是相同的。

(4) 标识符不能与“关键词”同名,也不与系统预先定义的“标准标识符”同名。

(5) 建议:标识符命名应当有一定的意义,做到见名知义。

(6) 关键词(保留字):C 语言规定的具有特定意义的字符串。

(7) 运算符:运算符将常量、变量、函数连接起来组成表达式,表示各种运算。运算符可以由一个或多个字符组成。

(8) 分隔符:逗号,空格。起分隔、间隔作用。

(9) 注释符:“`/ *`”和“`* /`”构成一组注释符。编译系统将`/ * ... * /`之间的所有内容看作为注释,编译时编译系统忽略注释。

① 注释在程序中的作用是提示、解释作用。

注释与软件的文档同等重要,要养成使用注释的良好习惯,这对软件的维护相当重要。记住:程序是要给别人看的,自己也许还会看自己几年前编制的程序(相当于别人看你的程序),清晰的注释有助于他人理解程序段的作用和算法的思路。

② 在软件开发过程中,还可以将注释用于程序的调试,即暂时屏蔽一些语句。

例如,在调式程序时暂时不需要运行某段语句,而又不希望立即从程序中删除它们,可以使用注释符将这段程序框起来,暂时屏蔽这段程序,以后可以方便地恢复。

2.2 计算机中各种进制数据的表示形式

2.2.1 各种进制的数

2.2.1.1 十进制数据

- (1) 计数基数 $0 \sim 9$;
- (2) 采用位置计数法,不同位置有不同位权。

例如:325。

$$325 = 3 * 10^2 + 2 * 10^1 + 5 * 10^0 \quad N_{10} = \sum_{i=-m}^{n-1} K_i * 10^i$$

2.2.1.2 二进制数据

- (1) 计数基数 $0 \sim 1$;
- (2) 采用位置计数法,不同位置有不同位权。

例如:110。

$$110 = 1 * 2^2 + 1 * 2^1 + 0 * 2^0 \quad N_2 = \sum_{i=-m}^{n-1} K_i * 2^i$$

2.2.1.3 八进制数据

- (1) 计数基数 $0 \sim 7$;
- (2) 采用位置计数法,不同位置有不同位权。

例如:0760。

$$760 = 7 * 8^2 + 6 * 8^1 + 0 * 8^0 \quad N_8 = \sum_{i=-m}^{n-1} K_i * 8^i$$

2.2.1.4 十六进制数据

- (1) 计数基数 $0 \sim 9 \quad A \sim F$;
- (2) 采用位置计数法,不同位置有不同位权。

例如:0xFE6。

$$FE6 = 15 * 16^2 + 14 * 16^1 + 6 * 16^0 \quad N_{16} = \sum_{i=-m}^{n-1} K_i * 16^i$$

2.2.2 各种进制之间数的转换

2.2.2.1 其他进制转换为十进制

按位权展开求和即可。

例如:二进制转换为十进制

$$1101 = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0 = 8 + 4 + 0 + 1 = 13$$

2.2.2.2 十进制转换为非十进制

设 $N(x)$ 为非十进制数, 则有 (位置计数法) $N(x) = \sum_{i=-m}^{n-1} K_i * x^i$;

(1) 其中: 对于整数部分 $N(x) = \sum_{i=0}^{n-1} K_i * x^i = K_{n-1} * X^{n-1} + K_{n-2} * X^{n-2} + \dots + K_2 * X^2 + K_1 * X^1 + K_0 * X^0$, 其中 K_i 就是非十进制整数部分各个位权上的值。

等式两边同时除上计数基数 X ; 则有 $N(x)/X = K_{n-1} * X^{n-1} + K_{n-2} * X^{n-2} + \dots + K_2 * X^2 + K_1 * X^1 + K_0$ 则余数 K_0 可得, 以此类推采用除基数取余数的方法, 可以将各个位权上的系数求得。

(2) 其中: 对于小数部分 $N(x) = \sum_{i=-m}^{-1} K_i * x^i = K_{-1} * X^{-1} + K_{-2} * X^{-2} + \dots + K_{-m} * X^{-m}$

等式两边同时乘以基数 X 则有 $N(x) * X = K_{-1} + K_{-2} * X^{-1} + \dots + K_{-m} * X^{-(m-1)}$, 则 K_{-1} 可得, 同理对余下 $k_{-2} \sim K_{-m}$ 同样可得。

2.3 整型数据

2.3.1 带符号的基本整型数据

(1) 带符号的基本整型数据 (int)。

(2) 带符号的基本整型数据使用 2~4 个字节内存存储空间。

(3) 整型数据的定义 `int x`。

(4) 整型数据在计算机内存中的表示:

①在计算机内存中数据的符号和数值是一同存储的。我们约定用 0 表示正数; 用 1 表示负数;

②在计算机内任何数据都是以二进制 0,1 形式存储的。

③假定计算机系统对带符号的数用 2 个字节表示。若有如下情况, `int x`, `y`, `x=+5`, `y=-5`, 则 `x`, `y` 所代表的内存单元存放该数据形式如下:

0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

问题来了, 我们发现将这两块内存单元数据相加得到这样一个结果:

1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---