前　言

近年来，在计算机的程序设计教学中，不管是计算机专业或者是非计算机专业，都选择了 C 语言作为首选语言。其原因是 C 语言语法简明，语句精练，数据类型和运算符丰富，所实现的功能强大。原作为 UNIX 操作系统的描述语言已演变成适用于开发系统软件和应用软件的通用语言，越来越受到人们的极大关注。

本教材是针对高职高专学生而写的。我们认为，这门课程的教学应以教会学生程序设计的基本思想和技术为目的。程序设计能力的培养主要是对学生理性思维方式的培养并充分考虑计算机对人们思维方式的影响。理性思维就是培养学生解决问题的思维的条理性和严密性，具有把一个复杂问题逐步分解成许多小任务的能力。这些小任务就是 C 语言中的函数概念。计算机本身的特点是容量大、速度快，这种特性也极大地改变着人们的思维方式。某种解决问题的思维方式对于人来说可能是难以实现的，然而对计算机来说却是轻而易举的事情。因此，任何一个程序都是设计者有条理的思维与计算机特点相结合的产物。为此，本教材想突出这个重点，而不对 C 语言的许多细枝末节过费篇幅。

学习程序设计必须循序渐进。我们建议“一看二仿三创作”。“一看”是指努力读懂已出现的程序，真正搞清楚程序运行的方式及所完成的功能；“二仿”是指模仿例题编写相似的程序，逐步训练自己举一反三的能力；“三创作”是针对问题独立地编写出自己的程序。只有通过自己的不断实践才能有所进步和提高，别无其他途径。

全书共分 9 章：第 2、3、4、5、7、9 章由周建中老师执笔，第 1、6 章由邱希春老师执笔，第 8 章和附录由陈莲君老师执笔，全书由邱希春老师统稿。

在本书编写的过程中，始终得到上海建桥学院信息技术系主任汪燮华教授的鼓励和指导，马妮娜和王敏慧老师帮助做了一些书稿的录入工作，在此一并表示感谢。

由于作者水平有限，疏漏和错误之处难以避免，恳请使用本书的老师和同学提出宝贵意见。

作　者

2007.7

修 订 说 明

自本书出版以来，有幸受到不少高校同行的厚爱。在教学过程中，他们也对本书提出许多建议和意见。正值再版之际，根据这些意见对本书大体上作了两个方面的修改：其一是对本书中的疏漏和印刷错误作了修改和订正。其二是对有些章节的习题又作了重新的编排，使之更加适合学生的水平。

由于作者水平有限，错误和疏漏在所难免，敬请老师和同学不吝指正。

目　　录

第 1 章　引论 ……1
1.1　C 语言简史 ……1
1.2　C 语言的特点 ……1
1.3　源程序的组成 ……2
1.3.1　源程序的宏观成分 ……2
1.3.2　源程序的微观成分 ……2
1.4　C 语言程序的开发过程 ……5
1.4.1　启动 Turbo C ……6
1.4.2　编辑源程序 ……6
1.4.3　编译、运行源程序 ……7
习题 1 ……7
第 2 章　算术类型数据 ……8
2.1　整数类型 ……8
2.1.1　变量定义 ……8
2.1.2　常量书写规则 ……11
2.1.3　数据的输入输出 ……13
2.2　实数类型 ……15
2.2.1　变量定义 ……16
2.2.2　常量书写规则 ……16
2.3　符号常量 ……17
习题 2 ……18
第 3 章　基本运算和表达式 ……20
3.1　基本运算 ……20
3.2　算术运算 ……23
3.2.1　四则运算 ……24
3.2.2　模运算 ……26
3.2.3　增 1 和减 1 运算 ……27
3.3　赋值运算 ……28
3.4　关系运算 ……29
3.5　逻辑运算 ……30
3.5.1　逻辑否运算 ……30
3.5.2　逻辑与运算 ……31
3.5.3　逻辑或运算 ……32

3.6 条件运算……33
3.7 位运算……34
3.7.1 移位运算……34
3.7.2 位逻辑运算……35
3.8 逗号运算……37
3.9 计算存储区大小……37
3.10 强制类型转换……38
3.11 表达式……38
习题 3……40
第 4 章 语句……42
4.1 基本语句……42
4.1.1 表达式语句……42
4.1.2 空语句……43
4.1.3 复合语句……43
4.2 选择控制语句……44
4.2.1 if 语句……44
4.2.2 switch 语句……47
4.3 循环控制语句……49
4.3.1 for 语句……49
4.3.2 while 语句……51
4.3.3 do-while 语句……52
4.3.4 嵌套的循环控制语句……52
4.4 转移语句……53
4.4.1 break 语句……53
4.4.2 continue 语句……55
4.4.3 return 语句……56
4.4.4 goto 语句……56
4.5 语句的综合应用……57
习题 4……62
第 5 章 数组类型……63
5.1 数组变量定义……63
5.2 数组元素的引用……64
5.3 数组的典型处理……65
5.3.1 顺序处理数组中满足指定性质的数据……65
5.3.2 顺序处理数组中满足指定性质的数据对……67
5.3.3 在数组中插入或删除一个数据……69
5.3.4 在数组中查找一个指定的数据……70
5.3.5 对数组中的数据进行排序……72
5.3.6 合并两个有序数组中的数据到一个数组……75

5.4 字符数组和字符串处理……76
5.4.1 字符串常量……76
5.4.2 字符数组和字符串的输入输出……76
5.4.3 字符串的典型处理……78
5.5 二维数组及处理……80
习题 5……83
第 6 章 函数……85
6.1 函数定义……86
6.1.1 函数定义的格式……86
6.1.2 函数定义之例……88
6.2 函数调用……91
6.2.1 函数调用的格式……91
6.2.2 函数调用中的一些问题……91
6.3 函数调用时的数据传送机制……93
6.4 函数的原型说明……94
6.5 递归……95
6.6 变量的存储类、作用域及初始化……97
6.6.1 变量的存储类与作用域……97
6.6.2 变量的初始化……101
6.7 预处理程序……103
6.7.1 无参#define……103
6.7.2 有参数的宏定义……105
6.7.3 包含文件……105
6.8 算法之例……106
6.8.1 排序……106
6.8.2 搜索……108
习题 6……109
第 7 章 指针类型……110
7.1 指针的基本概念……110
7.1.1 变量的存储区和变量的地址……110
7.1.2 计算变量的地址……111
7.1.3 指针变量……111
7.2 与指针类型相关的基本运算……112
7.2.1 取变量地址运算……112
7.2.2 赋值运算……113
7.2.3 间接访问运算……114
7.2.4 指针加、减整型量……114
7.2.5 指针减指针……116
7.2.6 指针变量的增 1、减 1 运算……117

7.2.7 指针的关系运算……118
7.2.8 数组成员选择……119
7.2.9 强制类型转换……120
7.2.10 指针的逻辑运算……120
7.3 二级指针……120
7.4 指针数组……121
7.5 指针与数组……123
7.5.1 引用数组中下标连续的一批成员……123
7.5.2 指向数组的指针……124
7.6 指针与函数……125
7.6.1 向函数传递一个数组……126
7.6.2 函数通过指针类型的参数向外传递计算结果……127
7.6.3 返回值为指针类型的函数……129
7.6.4 指向函数的指针……130
7.7 命令行参数……132
习题 7……134
第 8 章 自定义数据类型——结构与联合……137
8.1 概述……137
8.2 结构类型的定义……137
8.3 结构变量的定义和使用……139
8.3.1 结构变量的定义……139
8.3.2 结构变量的使用……140
8.4 结构与数组……144
8.4.1 结构数组的定义……144
8.4.2 结构数组的初始化……146
8.5 结构与指针……147
8.5.1 结构成员可以是指针类型的变量……147
8.5.2 指向结构变量的指针……148
8.6 结构与函数……151
8.6.1 基本数据类型作为函数形参……151
8.6.2 结构变量和结构指针作为函数的形参……152
8.6.3 结构变量作函数的返回值……155
8.7 链表……157
8.7.1 链表概述……158
8.7.2 单链表结点类型的定义……158
8.7.3 单链表的建立……159
8.7.4 单链表的遍历……164
8.7.5 单链表的插入和删除……165
8.7.6 单链表的综合应用……169

8.8 联合 ……172
8.8.1 联合的定义和引用 ……173
8.8.2 联合举例 ……174
8.9 用 typedef 重定义类型名 ……174
8.9.1 概念 ……174
8.9.2 典型用法 ……175
8.9.3 typedef 与#define 的区别 ……176
习题 8 ……176
第 9 章 文件和文件处理 ……178
9.1 文件 ……178
9.1.1 打开文件 ……178
9.1.2 关闭文件 ……181
9.2 文本文件 ……182
9.3 二进制文件 ……187
习题 9 ……192
附录 A ASCII 码表 ……195
附录 B Turbo C 2.0 常用库函数及其头文件 ……196
参考文献 ……200

第1章 引 论

1.1 C语言简史

C语言诞生于20世纪70年代初期，它的前身是英国剑桥大学的Martin Richards在20世纪60年代开发的BCPL语言。该语言是Martin Richards为描述和实现UNIX操作系统而为自己设计的工作语言。1970年，美国贝尔实验室的Ken Thompon继承和发展了BCPL语言，提出了B语言，并用B语言在当时最新型的小型机PDP-7上实现了第一个UNIX操作系统。1972年，美国贝尔实验室的Dennis M. Ritchie和Brian W. Kerninghan对B语言作了进一步的完善和发展，提出了一种新型的程序设计语言——C语言。1973年，K. Thompson和Dennis M. Ritchie合作用C语言成功地改写了UNIX操作系统。自从C语言问世以来，表现出极强的生命力，从最初为记述UNIX操作系统而开发的语言，到现在已经发展成为广泛应用的系统描述语言和通用的程序设计语言。随着微型计算机软硬件的发展，C语言已经成为微机上用于开发各种软件尤其是系统软件的主要工具之一。

1.2 C语言的特点

1. C语言的基本特点

C语言是介于汇编语言和高级语言之间的一种程序设计语言。C语言与硬件系统比较接近，它有直接访问硬件的功能，并具有汇编语言的大部分功能；C语言又具有高级语言面向用户、容易记忆、便于编程和阅读的优点。所以C语言既是系统描述语言，又是通用的程序设计语言。

函数是C语言的基本单位。C语言程序是由若干个函数组成的，其中必包含一个main()函数，即主函数。main()函数由用户自己定义，程序由main()函数开始执行。其余的函数可由用户自己定义，也可利用C语言函数库中任意一个函数。由于这个特点，C语言便于实现程序的模块化设计。

C语言运算符特别丰富，数据结构类型广泛，同时具有结构化控制语句，是一种结构化程序设计语言，即具有顺序、分支、循环三种基本结构。

C语言的语言简洁、紧凑，使用方便、灵活，语法限制不太严格，程序设计自由度大。

C语言生成的目标代码质量高，程序执行效率高。而且用C语言写的程序移植性好。

2. C语言的书写风格

每个语句占用一个书写行。

每个函数都可按语句的层次关系形成缩进形式。

一个语句也可以占用几个书写行。但最好不要把一个关键词、标识符、常量、运算符和

字符串拆分为两行。当确实需要这样做时，则在上一行末尾加上< \ 回车>，在下一行接着书写其他字符也可解决。

为了增加程序的可读性，可在程序任何需要的地方加上注解。注解总是用/* 和 */括起来，其中的内容是给阅读源程序的人看的，而不是让计算机执行的，编译系统在把源程序翻译成目标程序时，总是忽略这些注解的。

1.3 源程序的组成

1.3.1 源程序的宏观成分

从宏观上看，一个 C 语言源程序主要是由若干个并列的函数定义组成的。每个函数定义构造了一个函数，即一个程序模块，一个程序模块用来完成整个计算任务的一部分。构成一个源程序的所有程序模块之间，以一定的方式相互联系，以完成程序设计者规定的整个计算任务。当计算任务比较简单时，整个计算任务可以用一个程序模块来完成，本书前半部分的所有例子中涉及的计算任务都相对简单，因此完成这些计算任务的源程序都仅有一个函数定义（一个程序模块）。

C 语言除了构造程序模块的函数定义外，还可以定义供各个函数共同使用的变量。变量是计算机内存中数据的存储区，用来存储计算过程中的各种数据。例如，计算所需的原始数据、计算过程中产生的中间结果或最终的计算结果等。

此外，一个 C 语言源程序还可以包含若干个适当的编译预处理命令，用来指示 C 语言的编译系统，在对源程序实际进行编译之前，完成某些适当的处理。例如，在源程序的指定位置上插入预先准备好的某个文件中的所有文字，或用一个字符序列来替代源程序中的所有另一个字符序列，如用 3.141 592 6 来替代源程序中的所有字符序列 PI，等等。

【例 1-1】 用来计算两个正整数的最大公因子的 C 语言源程序。

一条编译预处理命令

```
#include <stdio.h>
int    main (void)
{   int    a, b, c;
    printf ("Input a and b:");      scanf("%d%d", &a, &b);
    if (a<1 || b<1) return 1;
    c = a%b;
    while (c!=0) {    a=b;    b=c;    c=a%b;    }
    printf ("Result is  %d\n", b) ;
    return 0;
}
```

一个函数定义

1.3.2 源程序的微观成分

从微观上看，一个 C 语言源程序是由字符的序列构成的。由于“换行字符”是可以使用的一种字符，因此一个 C 语言源程序也可以被看成是由若干行组成的，而每一行由若干个字

符构成。

事实上，一个 C 语言源程序是由一系列取自“基本字符集”中的字符构成的。由若干个字符按一定的规则，构成诸如标识符、常量、运算符和分隔符之类的基本词法单位（基本词法单位是源程序中具有完整意义的最小语法单位），若干个基本词法单位又按一定的规则，构成诸如表达式、语句和函数等更大的语法单位。如图 1-1 所示。

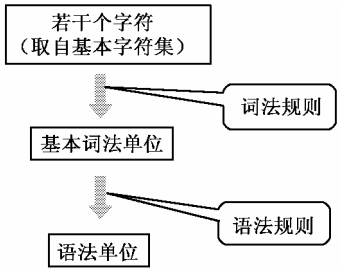

图 1-1　C 语言源程序的微观构成

1．基本字符

C 语言的基本字符集至少包含下列字符。

① 大写英文字母：

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

② 小写英文字母：

a b c d e f g h i j k l m n o p q r s t u v w x y z

③ 十进制数字字符：

0 1 2 3 4 5 6 7 8 9

④ 数字、字母外的下列 29 个印刷字符：

! " # % & ' ( ) * + , – . / : ; < = > ? [\ ] ^ _ { | } ~

⑤ 其他 3 个字符：

空格符、制表符和换行符。

2．基本词法单位

主要有标识符、常量、运算符、分隔符这 4 类基本词法单位。

（1）标识符

标识符用来命名各种程序元素，如语句的种类、变量的名称和函数的名称等。一个标识符是同时满足下面两个词法规则的字符序列：

① 以英文字母（不论大小写）或下划线字符（_）为序列的第一个字符；

② 第一个字符之后有一个长度为 n（$n \geqslant 0$）的、仅由英文字母（不论大小写）或下划线字符或十进制数字字符构成的字符序列。

标识符的词法规则如图 1-2 所示。

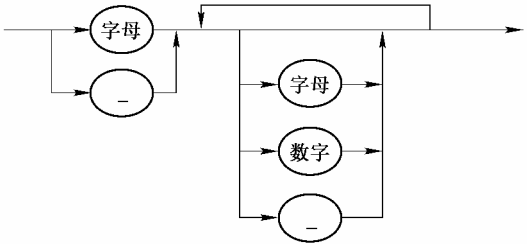

图 1-2　标识符的词法规则

例如，Flag，Flag_10 和_File_Count 是三个正确的标识符；Flag 和 flag 是两个正确的、互不相同的标识符；Cash.num 和@Dec 是两个错误的标识符。

按照功能的不同，我们把标识符分成如下 3 类。

① 关键字：C 语言的编译系统已经给予固定意义的标识符。某些关键字是数据类型的名称，另一些关键字指出语句的种类，或指出程序元素的其他性质。

表 1-1 所示为 C 语言的全部关键字，在本书的其他章节中将介绍它们的意义和用法。

表 1-1 C 语言的关键字

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

其中：

- int：整数类型的类型名。
- double：双精度实数类型的类型名。
- for：执行重复计算的循环语句。
- if：执行分支计算的选择语句。

② 标准标识符：C 语言的程序设计环境中，已经被给予指定意义的标识符。如：

- printf：格式化输出库函数的函数名。
- scanf：格式化输入库函数的函数名。

INT_MAX：整数类型的最大数据（在 TURBO C 程序设计环境中，代表整数值 32767）。

③ 用户定义的标识符：除了关键字和标准标识符之外的其他标识符。在不混淆的情况下，把“用户定义的标识符”简单地说成“标识符”。

通常，程序设计者使用标识符来命名程序中的变量、函数或其他程序元素。

变量是计算机的内存中用来存储数据的区域，由连续的 *n* 个字节构成。一个变量占用的存储区的大小（即字节数 *n*），一方面由数据的类型确定，另一方面也由不同的 C 语言编译系统确定。例如，在程序设计环境 TURBO C 中，一个 int（标准整数）类型变量的存储区大小为 2，一个 double（双精度实数）类型变量的存储区大小为 8，等等。在程序的计算过程中，我们可以通过某种方法，把一个数据存储到指定的变量中，以后可以任意多次地取出该变量中的这个数据，进行适当的计算处理。只要不把一个新的数据存储到这个变量中去，该变量中的数据总是不会发生任何变化。一旦向这个变量中存储了一个新的数据，该变量中原先存储的旧数据将会丢失，新的数据将被保存在这个变量中。

（2）常量

常量是计算过程中，其值固定不变的数据。在程序中使用常量时，我们只需要根据常量的类型，按照相应的规则直接书写所需的常量。例如：

- 128：int（标准整数）类型的常量，如用来表示一本书的页数；
- 22.5：double（双精度实数）类型的常量，如用来表示应保持的最高温度；
- "Addr"：字符串常量，这是内容固定不变的一段文字。

在本书的第 2 章中，我们将介绍不同类型的常量的书写规则。

（3）运算符

使用不同的运算符来代表不同的基本运算。基本运算是 C 语言提供的对数据进行的最简单的处理，通常，一个基本运算对确定个数的数据（常量和/或变量）进行处理，并按照一定的规则产生一个确定的计算结果。如：+ 代表基本运算“加法”；% 代表基本运算“取余”，即，计算左边的整数除以右边的整数所得的余数；sizeof 代表基本运算“计算存储区大小”。

在本书的其他章节中，我们将介绍其他的基本运算。

（4）分隔符

分隔符的主要作用是分隔两个相邻的常量和（或）标识符的符号。可以有如下 4 种形式不同的分隔符。

① 空白：一个或一个以上连续的空格字符。

② 换行符：由确定键（如键盘中的 Enter 键或 Return 键）所表示的字符。

③ 制表符：由 Tab 键所表示的字符。

④ 注解：注解的一般形式是“/*　字符序列　*/”。

这里，“/*”是注解的开始记号，“*/”是结束记号，中间是字符的一个序列。注解不能嵌套，即在注解中不能再出现另一个注解。因此，在该字符序列中不能再出现注解的开始和结束记号。

对于 C 语言编译系统而言，一个注解相当于一个空格字符，因此在程序中加入注解，并不影响程序执行的效果。实际上，使用注解对程序适当位置上的程序元素进行简要的说明。例如，说明在某处定义的变量的用途，或说明某个语句的作用，等等。适当地使用注解对程序元素进行必要的说明可以提高程序的可读性，使人们对程序的理解更为容易。

1.4　C 语言程序的开发过程

用 C 语言开发一个程序的过程如下。

（1）编制源程序

即根据问题的实际情况，规划数据的存储形式，设计解决问题的算法，用 C 语言编写程序。这个程序称为 C 语言源程序。

（2）编辑源程序

用计算机系统提供的编辑程序或用 C 语言集成系统自身提供的编辑系统，将源程序输入计算机，并通过修改、编辑后存入文件中。该源程序以文本文件的形式存储在计算机中。源文件的名称由用户自己定义，扩展名一般定为“.C”（在 Turbo 系统中）或“.CPP”（在 Borland C++中）。

（3）编译源程序

用 C 编译程序对源程序进行编译。首先处理预处理部分，进行宏替换并把头文件合并到源程序中，再对该源程序进行编译，生成目标程序，扩展名为“.OBJ”。编译时还对源程序的语法和程序的逻辑结构等进行检查，当发现错误时，在显示器上列出错误的类型和位置，此

时要对源文件再次进行编辑。

（4）链接

将目标程序和库函数连接成可执行的程序，扩展名为“.EXE”。

（5）运行

上面生成的 EXE 文件就可以运行了。

下面以 Turbo C 为例，简单地介绍 C 语言程序编辑、运行的过程。详细过程请查看 Turbo C 使用和参考手册。

1.4.1 启动 Turbo C

在 DOS 提示符下，输入“TC”，回车后立即出现 Turbo C 的主屏幕菜单，如图 1-3 所示。该主屏幕分成 4 个部分：主菜单、编辑窗口、消息窗口和功能键提示行。

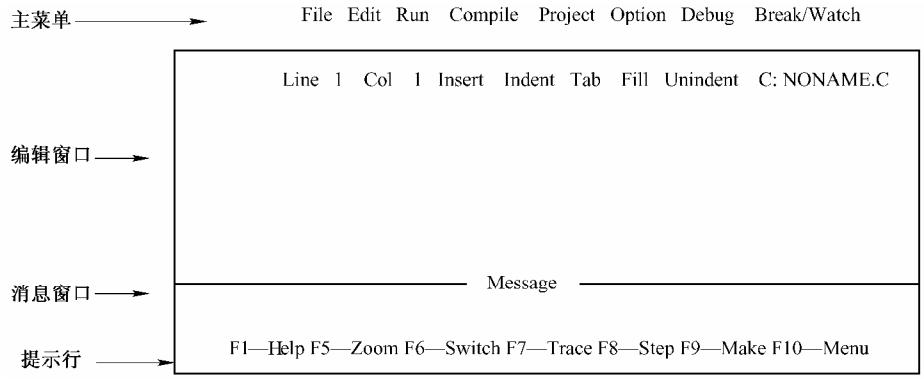

图 1-3 Turbo C 主屏幕

按 F10 键激活主菜单。用光标移动键可以在主菜单中移动，并下拉出一个菜单，选中某一菜单选项后，按 Enter 键。按 Esc 键退出主菜单。

1.4.2 编辑源程序

建立新文件：在主菜单下，选择 File →Load，屏幕出现一个对话框，输入新文件名。输入新文件名后，回到编辑窗口。

修改老文件：在主菜单下，选择 File →Open，屏幕出现一个对话框，输入旧文件名。输入旧文件名后，在编辑窗口显示老文件内容，等待修改。

编辑文件时，常用的编辑键如下。

- Up / Down，Left / Right，PgUp / PgDn：翻滚正文
- Ctrl+Y：删除一行。
- Ctrl+T：删除一个单词。
- Ctrl+KB：设置块开始。
- Ctrl+KK：设置块结尾。
- Ctrl+KV：块移动。
- Ctrl+KC：块拷贝。

↘ Ctrl+KY：块删除。

源程序存盘：在主菜单下，选择 File→Save，或在系统中任何时候，按 F2 键。

1.4.3　编译、运行源程序

在系统中任何时候，按 Ctrl+F9 键。首先编译源程序，若有错，则在消息窗口显示出错信息。根据出错信息来编辑修改源程序，窗口之间转换用 F6 键。编译正确无误后，即产生 EXE 文件，并随即运行产生的执行文件。

如果有输入数据，程序运行后将停在用户屏幕，等待输入数据。当输入数据完毕，程序继续执行，正常结束后又返回到编辑窗口。如果有输出数据，可按 Alt+F5 键查看输出结果。

习题 1

1．上机运行例 1-1，熟悉所用系统的上机方法和步骤。

2．参照本章例题，编写一个简单的 C 程序，输出以下信息：

```
*****************************
Welcome  to  study  C  program!
*****************************
```

第 2 章　算术类型数据

计算机程序中，数据被用来记录各种有关的信息，程序中的数据总是以变量和常量这两种形式出现。常量是程序运行过程中值不发生变化的数据；变量实际上是数据的存储区，在程序运行过程中，可以根据需要把数据存储到变量中，也可以从变量中取出数据进行适当的处理，因此变量的值是可以变化的。

按照数据的性质，数据被分成各种类型，不同类型数据的书写格式、在计算机内部的存储格式，以及可以对它们进行的处理是各不相同的。

算术类型是一种基本的数据类型，包括整数和实数两种不同的类型。

一个计算机程序中，可以按照需要使用多个变量和常量。所有变量在使用前都必须进行变量定义，而常量则可按照相关类型的常量书写规则直接书写。

2.1　整数类型

整数类型的数据可以用来表示一定值范围内的整数。本节介绍整数类型变量的定义、整数类型常量的书写规则，以及通过库函数输入、输出整数数据的方法。

2.1.1　变量定义

1．整数变量

整数变量定义的一般形式为：

```
整数类型名　变量名表;
```

事实上，完整的整数类型名包括一个符号限定符、一个长度说明符和关键字 int。

符号限定符：unsigned（不带符号的）和 signed（带符号的）。

长度说明符：long（长的）和 short（短的）。

一个完整的整数类型名中，不出现符号限定符时，默认为是带符号（signed）类型的；不出现长度说明符，默认为是标准长度的。一个通用的整数类型名，是从完整的整数类型名中省略了关键字 int（如果可以省略的话）形成的，使用通用整数类型名显得比较简洁。

长度说明符规定了各种整数类型数据的相对大小，总是有：

```
sizeof (short) <= sizeof (int) <= sizeof (long)
```

上面的 sizeof 是 C 语言的基本运算：计算存储区大小（字节数）。sizeof(short)的计算结果是任何一个 short 类型的变量存储区的字节数，sizeof(int)的计算结果是任何一个 int 类型的变量存储区的字节数，等等。

符号限定符规定了整数变量存储区的使用方法：设 T 是某种整数类型，T 类型的变量 v 拥有 k = sizeof（T）个字节的存储区，即 8k 个二进制位（bit）。若 v 是带符号的 T 类型变量，

则 8k 个二进制位中的最高有效位被用来表示数据的符号（用 1 表示负号，0 表示正号）；若 v 是不带符号的 T 类型变量，则全部 8k 个二进制位都被用来表示数据的值。

各种完整的整数类型名和通用的整数类型名及其长度如表 2-1 所示。

表 2-1　整数类型及其长度

完整的整数类型名	通用的整数类型名	说　　明	存储区大小（字节数）
short int	short	短整数	2
int	int	（标准）整数	2
long int	long	长整数	4
unsigned short int	unsigned short	无符号短整数	2
unsigned int	unsigned	无符号（标准）整数	2
unsigned long int	unsigned long	无符号长整数	4

下面是不同类型的整数变量的定义：

```
int  a,b;            /* （标准）整数类型变量，变量 a 和 b 是带符号的 */
unsigned  c;         /* 无符号的（标准）整数类型变量 */
unsigned  long d;    /* 无符号的长整数类型变量 */
```

说明：通过上面的变量定义，得到名为 a、b、c、d 的 4 个整数变量，可以在程序执行期间用来存储整数数据。变量 a、b 是带符号的（标准）整数类型变量，分别拥有 2 个字节的存储区；变量 c 是无符号的（标准）整数类型变量，拥有 2 个字节的存储区；变量 d 是无符号的长整数类型变量，其存储区的大小为 4 个字节。

设在变量 a、b、c、d 中存储了如图 2-1 所示的数据。

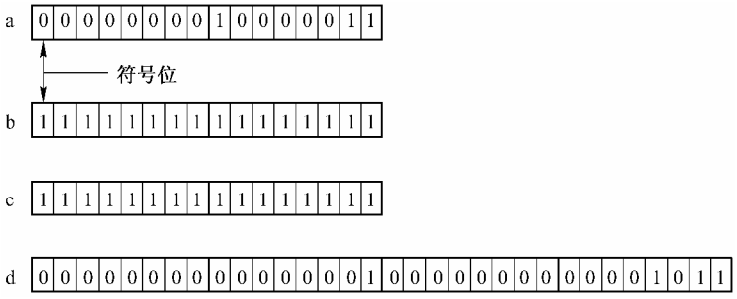

图 2-1　变量存储数据

变量 a 中存储的数据是+131，变量 b 中存储的数据是–1，变量 c 中存储的数据是 65 535，变量 d 中存储的数据是 65 547。

这 6 种不同类型整数的区别在于存储区大小的不同，以及存储区用法（如是否有符号位）的不同，因此不同类型的整数数据的值范围是不同的，如表 2-2 所示。可以根据问题的背景和需要，定义适当的整数类型变量。

变量定义时，可以指定变量的初值，这称为变量的初始化。比如：

```
int  a=0, b, c=1;
```

表 2-2 整数类型及其取值范围

通用类型名	存储区大小（字节）	值 范 围
short	2	–32768～32767
int	2	–32768～32767
long	4	–2147483648～2147483647
unsigned short	2	0～65535
unsigned	2	0～65535
unsigned long	4	0～4294967295

说明：通过上面的变量定义，得到名为 a、b、c 的三个整数变量。变量 a 的定义中带有初始化部分（初值 0），当变量 a 获得存储区时，初值 0 立即被送到变量 a 的存储区内；同样当变量 c 得到存储区时，初值 1 立即被送到变量 c 的存储区内。这样，当变量 a 和 c 在第一次被使用时，它们就已经有了确定的值。当然，为某个变量设置什么样的初值是由问题的背景确定的。变量 b 的定义中无初始化部分，因此，当变量 b 第一次被使用时，我们不能确定该变量的存储区的内容，即该变量的值不确定。

变量初始化部分的一般形式如下：

= 初始化表达式

这里的初始化表达式是可确定其值的表达式，通常是一个常量（例如 12），也可以是对一系列的数据（变量或常量）进行计算的结果（例如，假定 N 代表 12，则 2*N–1 是一个常量表达式，其值为 23）。

2．字符变量

除了以上 6 种不同类型的整数外，字符类型实际上也属于 C 语言中的整数类型。字符类型的变量可能与其他整数类型的变量存储区的大小不同，如表 2-3 所示。但在其他所有方面与整数类型的性质完全相同。

表 2-3 字符类型及其长度

完整的字符类型名	通用的字符类型名	说 明	存储区大小（字节数）
char	char	字符	1
unsigned char	unsigned char	无符号字符	1

字符类型的变量通常拥有 1 字节的存储区，不能使用长度说明符改变字符类型变量存储区的大小，但可以使用符号限定符规定字符变量存储区的使用方法。因此，不同的字符类型数据的值范围是不同的，如表 2-4 所示。可以根据问题的背景和需要，定义适当的字符类型变量。

表 2-4 字符类型及其取值范围

通用类型名	存储区大小（字节）	值 范 围
char	1	–128～127
unsigned char	1	0～255