



"十二五"普通高等教育本科国家级规划教材

C 语言程序设计 教程

(第2版)

□ 龚本灿 主编

□ 周治钰 副主编

高等教育出版社



“十二五”普通高等教育本科国家级规划教材

C 语言程序设计教程

C Yuyan Chengxu Sheji Jiaocheng

(第2版)

龚本灿 主 编

周治钰 副主编

吴西燕 任小燕 付云侠 宋建萍 王少蓉 编

高等教育出版社·北京

内容提要

本书以“学生信息管理系统”案例贯穿全书,并以此为主线将 C 语言各章节的主要知识点有机结合,概念清晰、通俗易懂。全书共 10 章,内容包括: C 语言概述、数据类型、分支结构程序设计、循环结构程序设计、数组、函数、指针、文件、编译预处理和用户定制数据类型。

本书采取案例驱动、设问解答、举例分析的组织方式,从实际应用系统开发的角度出发,由浅入深、循序渐进地培养读者的初步编程能力,帮助读者掌握 C 语言的语法知识,引导读者掌握大规模应用程序的编写方法。本书配有多媒体课件、例题和习题源代码,并配有辅助教材《C 语言程序设计习题集(第 2 版)》。

本书适合作为高等学校各专业的 C 语言程序设计课程教材,也可作为从事计算机相关工作的科技人员、计算机爱好者及各类自学人员的参考书。

图书在版编目(CIP)数据

C 语言程序设计教程 / 龚本灿主编. --2 版. --北京: 高等教育出版社, 2015.3

ISBN 978-7-04-041943-6

I. ①C… II. ①龚… III. ①C 语言-程序设计-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2015) 第 019115 号

策划编辑 刘 茜	责任编辑 刘 茜	封面设计 张申申	版式设计 马敬茹
插图绘制 黄建英	责任校对 王 雨	责任印制 尤 静	

出版发行 高等教育出版社	网 址 http://www.hep.edu.cn
社 址 北京市西城区德外大街 4 号	http://www.hep.com.cn
邮政编码 100120	网上订购 http://www.landraco.com
印 刷 北京宏信印刷厂	http://www.landraco.com.cn
开 本 787mm×1092mm 1/16	版 次 2011 年 12 月第 1 版
印 张 18	2015 年 3 月第 2 版
字 数 400 千字	印 次 2015 年 3 月第 1 次印刷
购书热线 010-58581118	定 价 29.00 元
咨询电话 400-810-0598	

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 41943-00

前 言

C 语言是目前最为流行的高级程序设计语言之一,是许多计算机专业人员和计算机爱好者学习程序设计语言的首选,因此,面向初学者,编写一本兼具可读性和实用性的 C 语言程序设计教材非常重要。本书具有以下特点:

(1) 以“学生信息管理系统”案例为主线来组织教材内容,将 C 语言各章节的主要知识点有机地结合起来,形成一个整体,让读者充分理解各章节知识点之间的联系,做到学有所用、融会贯通。在案例的求解过程中,讲授程序设计的基本方法,侧重对模块化程序设计思想的介绍,并将软件工程相关的思想渗透其中,以提高读者编程解决实际问题的能力。

(2) 选用的案例与学生联系紧密、易于理解,内容的讲解由问题驱动,让读者以当事人身份,从实际应用系统开发的角度来分析问题,由浅入深、循序渐进地引导读者掌握大规模应用程序的编写方法,让读者在饶有兴趣的案例设计中逐步培养编程能力、掌握 C 语言的语法知识。

(3) 除了“学生信息管理系统”案例外,本书还精选了一些知识性、趣味性强的例子,如鸡兔同笼问题、百钱买百鸡问题、汉诺塔问题等,力图通过这些经典的实例来开拓读者思路、引导读者深入理解程序设计的基本思想、培养读者的计算思维能力。

(4) 本书在写作上遵循学生的认知规律,力图避免以往教材的一些问题。以往教材通常是以语言知识为主线来组织教学内容、注重知识模块的完整性,这些教材的优点是内容系统、语法详尽,但忽略了对程序设计思想的训练,对语法的讲授,不是为应用而讲语法,而是为语法而讲语法,使学生普遍感觉 C 语言语法规则多、枯燥难懂,全部内容学完后,很多学生只能看懂小型程序,不能充分理解各章节知识点之间的联系,不能综合运用所学知识进行一个实际项目的开发。

(5) 本书配有多媒体课件、例题和习题源代码,并配有辅助教材《C 语言程序设计习题集(第 2 版)》,可供读者同步练习。习题集的每一章分两节,第一节为课后习题解答,第二节为等考模拟试题,每道题都附有答案,对稍难一点的试题还进行了解析和说明。试题的安排由易到难,课后习题部分相对简单,是学生需要熟练掌握的内容,等考模拟试题部分相对较难,能够满足“全国计算机等级考试二级 C 语言”的需要。

本书由龚本灿任主编并统稿,周治钰任副主编,参编人员有周治钰、吴西燕、任小燕、付云侠、宋建萍、王少蓉。参与本书讨论和校对工作的有赵昕、郭德明、杨华甫、

II 前言

杨景华、高蓉、冯家林、袁伟、丰京丹、叶华、石勇涛。在本书的编写过程中，董方敏教授和周学君副教授对书中内容提出了许多宝贵的意见和建议。在此对他们的支持和帮助表示衷心的感谢。

我们力求精益求精，但由于编者水平有限，书中难免有疏漏之处，恳请广大读者来信指正。作者 E-mail 地址为 gonbc@sina.com、gonbc@tom.com。

编者

2014年11月

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任；构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人进行严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话 (010) 58581897 58582371 58581879

反盗版举报传真 (010) 82086060

反盗版举报邮箱 dd@hep.com.cn

通信地址 北京市西城区德外大街4号 高等教育出版社法务部

邮政编码 100120

目 录

第1章 C语言概述	1
1.1 计算机程序设计语言的发展	1
1.2 C语言的发展及特点	2
1.2.1 C语言的发展	2
1.2.2 C语言的特点	3
1.3 简单的C语言程序	4
1.4 C语言程序的开发步骤及上机过程	10
1.4.1 C语言程序的开发步骤	10
1.4.2 C语言程序的上机过程	11
1.5 算法及其描述	15
1.5.1 算法的概念	15
1.5.2 算法的描述	16
1.6 学生信息管理系统案例简介	18
本章小结	20
习题	20
第2章 数据类型	24
2.1 数据类型的分类	24
2.2 整型数据	25
2.2.1 整型变量	25
2.2.2 整型常量	26
2.3 实型数据	27
2.3.1 实型变量	27
2.3.2 实型常量	27
2.4 字符型数据	28
2.4.1 字符型变量	28
2.4.2 字符型常量	29
2.5 数据类型转换	30
2.6 格式化输入/输出函数	32

2.6.1 格式化输入函数	32
2.6.2 格式化输出函数	33
本章小结	34
习题	34
第3章 分支结构程序设计	37
3.1 关系运算符及其表达式	37
3.2 逻辑运算符及其表达式	38
3.3 if语句	40
3.3.1 单分支条件语句	40
3.3.2 双分支条件语句	43
3.3.3 多分支条件语句	45
3.3.4 if语句的嵌套	48
3.4 switch语句	49
3.5 运算符及其表达式	52
3.5.1 算术运算符及其表达式	53
3.5.2 赋值运算符及其表达式	54
3.5.3 条件运算符及其表达式	55
3.5.4 逗号运算符及其表达式	56
3.5.5 运算符的优先级和结合方向	56
3.6 案例分析与实现	57
本章小结	58
习题	58
第4章 循环结构程序设计	63
4.1 for语句	63
4.2 while语句	68
4.3 do-while语句	71
4.4 break语句和continue语句	73
4.4.1 break语句	73
4.4.2 continue语句	75

II 目录

4.5 循环语句的嵌套	76	6.2 函数的应用举例	125
4.6 C 语言程序的调试方法	78	6.3 函数的递归调用	129
4.7 案例分析与实现	81	6.4 数组作为函数参数	133
本章小结	82	6.4.1 数组元素作为函数实参	133
习题	82	6.4.2 数组名作为函数实参	134
第 5 章 数组	88	6.5 变量的作用域	136
5.1 一维数组	88	6.5.1 局部变量	137
5.1.1 一维数组的定义	88	6.5.2 全局变量	138
5.1.2 一维数组的引用	89	6.6 变量的生存期	141
5.1.3 一维数组的初始化	89	6.7 外部函数与内部函数	144
5.1.4 一维数组的应用举例	90	6.8 案例分析与实现	145
5.2 二维数组	91	6.8.1 案例中的自定义函数简介	145
5.2.1 二维数组的定义	91	6.8.2 案例中函数之间的调用关系	145
5.2.2 二维数组的引用	92	6.8.3 案例中部分功能模块的实现	146
5.2.3 二维数组的初始化	92	6.8.4 多文件的编译与连接	149
5.2.4 二维数组的应用举例	93	本章小结	152
5.3 字符数组	97	习题	153
5.3.1 字符数组的定义	97	第 7 章 指针	159
5.3.2 字符数组的引用	97	7.1 指针概述	159
5.3.3 字符数组的初始化	97	7.1.1 指针的概念	159
5.3.4 字符串输入/输出函数	99	7.1.2 指针变量的定义和引用	160
5.3.5 字符串处理函数	102	7.1.3 指针变量作为函数参数	163
5.4 排序算法	103	7.2 指向一维数组的指针	165
5.5 案例分析与实现	108	7.2.1 指向一维数组的指针变量的定义	166
5.5.1 结构体类型	108	7.2.2 指针变量的算术运算	167
5.5.2 学生结构体数组的定义和引用	112	7.2.3 指针变量的应用举例	168
5.5.3 案例中部分功能模块的实现	114	7.2.4 指向字符串的指针	170
本章小结	117	7.3 指向结构体的指针	173
习题	117	7.3.1 指向结构体变量的指针	173
第 6 章 函数	121	7.3.2 指向结构体数组的指针	175
6.1 函数的定义、调用和声明	121	7.4 返回指针的函数	176
6.1.1 函数的定义	121	7.5 动态内存分配	178
6.1.2 函数的调用	122	7.6 指针数组	181
6.1.3 函数的声明	123	7.7 指向二维数组的指针	184
6.1.4 函数的调用过程	124	7.7.1 二维数组的地址	184

7.7.2 指向二维数组的指针·····	185	9.1 宏定义·····	229
7.8 指向函数的指针·····	187	9.1.1 无参数的宏定义·····	229
7.9 指向指针的指针·····	190	9.1.2 有参数的宏定义·····	231
7.10 案例分析与实现·····	191	9.2 文件包含与案例分析·····	233
7.10.1 链表简介·····	191	9.2.1 文件包含·····	233
7.10.2 学生信息链表中结点的定义·····	192	9.2.2 案例分析·····	234
7.10.3 学生信息链表的操作·····	192	9.3 条件编译·····	235
本章小结·····	200	本章小结·····	238
习题·····	200	习题·····	238
第8章 文件 ·····	206	第10章 用户定制数据类型 ·····	242
8.1 文件的基本知识·····	206	10.1 共用体类型·····	242
8.1.1 文件的概念·····	206	10.1.1 共用体类型的定义·····	242
8.1.2 文件的分类·····	206	10.1.2 共用体变量的定义和引用·····	243
8.1.3 缓冲文件系统·····	207	10.2 枚举类型·····	245
8.2 文件的打开与关闭·····	208	10.2.1 枚举类型的定义·····	245
8.2.1 文件的打开·····	208	10.2.2 枚举变量的定义和引用·····	245
8.2.2 文件的关闭·····	210	10.3 自定义类型名·····	247
8.3 文件的读写操作·····	210	10.4 位段与位运算·····	250
8.3.1 字符读写函数·····	211	10.4.1 位段·····	250
8.3.2 字符串读写函数·····	213	10.4.2 位运算·····	252
8.3.3 格式化文件读写函数·····	215	本章小结·····	255
8.3.4 数据块读写函数·····	216	习题·····	256
8.4 文件的随机读写·····	218	附录A ASCII 码表 ·····	258
8.5 案例分析与实现·····	221	附录B C 运算符的优先级与结合性 ·····	260
本章小结·····	224	附录C C 语言常用库函数 ·····	262
习题·····	225	附录D 学生信息管理系统源代码 ·····	266
第9章 编译预处理 ·····	229	参考文献 ·····	276

第1章 C语言概述

C语言是目前国际上比较流行的高级程序设计语言之一，因其语言简洁、使用方便且功能强大，受到广大编程人员的普遍青睐。本章主要介绍C语言的基本知识，通过几个简单的C语言程序实例，使读者掌握C语言程序的基本结构和书写规则、明确C语言程序的上机操作步骤。另外，本章还将简要介绍“学生信息管理系统”案例的基本功能。

【学习目标】

- 掌握C语言程序的基本结构和书写规则
- 掌握C语言程序的上机操作步骤
- 熟悉Visual C++ 6.0集成开发环境
- 了解计算机程序设计语言的发展
- 了解C语言的特点
- 了解算法及其描述

1.1 计算机程序设计语言的发展

一个完整的计算机系统包括硬件系统和软件系统两大部分，硬件是计算机进行运算的物质基础，而软件则是计算机的灵魂，用来管理和指挥计算机的运行。离开了软件，计算机就什么事情都干不了。所有软件的程序代码都是用某种计算机语言编写的，因此，要想设计软件首先必须掌握一门编程语言。计算机程序设计语言的发展经历了从机器语言、汇编语言到高级语言的发展历程，下面简要介绍各种程序设计语言的特点。

1. 机器语言

计算机可以直接处理的是由“0”和“1”组成的二进制数，在计算机出现的初期，编程人员只能用一连串由“0”和“1”编码组成的指令序列来操作计算机。例如，某种型号的计算机用10000000表示加法指令，用10010000表示减法指令。称这种直接使用二进制编码的程序语言为机器语言。

机器语言程序能在计算机上直接运行，具有占用内存少、执行速度快的优点，但机器语言不直观、难记、难认、不易查错，编程非常麻烦，只能被少数专业人员掌握。另外，不同型号计算机的指令系统通常不同，使得在一台计算机上编写的程序，在另一台计算机上可能无法

运行，需要重新编写，因此，用机器语言编写的程序可移植性差。

2. 汇编语言

为了减轻使用机器语言编程的痛苦，人们对机器语言进行了改进，用一些简洁的助记符来代替二进制的机器指令。例如，用“ADD”表示加法，用“SUB”表示减法，等等。称这种用符号表示的程序设计语言为汇编语言。使用汇编语言编写的程序在计算机上不能直接运行，需要通过一个专门的程序将其翻译成机器语言，这个翻译程序叫做汇编程序。

汇编语言比机器语言更容易理解，能充分发挥计算机硬件的功能和特长，程序精炼而高效，但汇编语言程序同样依赖于计算机硬件、可移植性不好，且助记符多、难以记忆。

3. 高级语言

高级语言以接近人们习惯的自然语言和数学语言为表示形式，其语法结构类似于普通英文，编程非常方便，一般人经过学习之后都可以进行编程。另外，高级语言程序不依赖于计算机硬件，能在所有机器上通用，可移植性好，但高级语言程序的运行效率要低于汇编语言和机器语言。高级语言的种类很多，常用的有C、C++、BASIC、FORTRAN、Pascal等。

用高级语言编写的程序称为源程序。同汇编语言一样，高级语言源程序也需要经过翻译转换成机器语言后才能执行，每一种高级语言都有它对应的翻译程序，翻译的方式有两种：编译方式和解释方式。

编译方式是通过一个专门的程序（即编译程序）一次性地将整个源程序翻译成机器语言程序，以后每次运行时直接执行已经翻译好的机器语言程序。例如，C、FORTRAN、Pascal都采用编译方式。

解释方式是每次运行时，将源程序翻译一句执行一句，解释方式不产生目标程序，每次运行都需要进行翻译，这样在程序的执行速度方面比编译方式要慢。例如，BASIC采用解释方式。

1.2 C语言的发展及特点

1.2.1 C语言的发展

C语言的出现与UNIX操作系统密切相关，UNIX操作系统具有稳定、可靠、安全等优点，是军队、电信、银行等大型机构最受信任的操作系统。早期的UNIX操作系统是用汇编语言编写的，汇编语言程序对计算机硬件的依赖性强、可读性和可移植性差，而一般的高级语言程序虽具有良好的可读性和可移植性，但不能对硬件直接进行操作，因此，需要设计一种集高级语言和低级语言功能于一体的语言，这样C语言应运而生。UNIX的研制者Ken Thomson和Dennis M. Ritchie当初开发C语言的主要目的就是为了更好地描述UNIX操作系统。C语言的发展经历了以下过程：

1960 年出现了 ALGOL 60 语言,它是一种面向问题的高级语言,不能操作硬件。

1967 年,英国剑桥大学在 ALGOL 60 的基础上推出了比较接近硬件的 BCPL (Basic Combined Programming Language),但软件规模比较大。

1970 年,美国贝尔实验室的 Ken Thomson 对 BCPL 进行简化,设计出了既简单又接近硬件的 B 语言,但 B 语言缺乏丰富的数据类型。

1971 年,美国贝尔实验室的 Dennis M. Ritchie 在 B 语言的基础上增加了丰富的数据类型和强有力的数据结构,最终形成了 C 语言。

1973 年, Ken Thomson 和 Dennis M. Ritchie 把 UNIX 操作系统 90% 以上的程序代码用 C 语言进行了改写。

1978 年, Brian W. Kernighan 和 Dennis M. Ritchie 合作出版了 The C Programming Language 一书,全面介绍了 C 语言。

由于 C 语言小巧灵活,并且功能强大,很快在全世界流行,不久后,便出现了多种 C 语言版本,不同的 C 语言编译器之间存在着细微的差异,这些差异影响了 C 语言的可移植性。为了解决这一问题,美国国家标准化协会 (ANSI) 于 1983 年成立了一个委员会来制定 C 语言的标准,并于 1989 年正式通过了 C 语言标准,即 ANSI 标准 X3.159-1989 (称为 ANSI C)。1990 年国际标准化组织 (ISO) 通过此项标准,并将其作为 ISO/IEC 9899:1990 国际标准,简称 C89。其后, C 语言国际标准化工作组 (WG14) 对 C89 进行了修订,于 1999 年完成并通过了 C 语言的最新正式标准,即 ISO/IEC 9899:1999 国际标准,简称 C99。

1.2.2 C 语言的特点

C 语言之所以发展迅速,成为应用广泛、最受欢迎的高级语言之一,是因为它具备以下主要特点。

1. 功能强大

C 语言具有许多其他高级语言不具备的功能,它能够像汇编语言一样对位、字节和地址进行访问,从而实现对硬件的直接操作。C 语言兼有高级语言和低级语言的双重特点,既可以用来编写应用软件,又可以用来编写系统软件。许多著名的系统软件,如 UNIX、dBase III 等都是用 C 语言编写的。

2. 语言简洁、紧凑,使用方便、灵活

C 语言一共只有 32 个关键字、9 种控制语句,语言简洁、紧凑。C 语言程序书写格式自由,可以添加空行,一行可以写多条语句,一条语句也可以占用多行,使用方便、灵活。

3. 程序执行效率高、可移植性好

C 语言程序的执行效率高于其他高级语言,比汇编语言程序的执行效率低 10%~20%。C 语言程序的可移植性好于汇编程序,可以不做修改或少量修改后在其他操作系统或机型上运行。

4. 编程风格结构化

在 C 语言中,函数是程序的基本组成单位,函数之间可以相互调用,具有良好的结构化编程风格。对大型程序可以采用模块化设计,将整个软件按功能分成若干个模块,每个模块编写相应的函数,这样使得程序层次清晰,便于编程、调试和维护。

1.3 简单的 C 语言程序

下面通过几个例题来介绍 C 语言程序的结构和基本的语法规则。

【例 1-1】 从键盘输入任意两个整数,输出两数之和。

程序代码如下:

/*文件名: p1-1.c

该程序的功能是求两数之和*/

#include <stdio.h> //注明输入/输出函数使用的头文件

void main() //定义主函数

{

int x,y,sum; //定义变量

scanf("%d%d",&x,&y); //调用库函数从键盘输入数据

sum=x+y; //计算两数之和

printf("sum=%d\n",sum); //在屏幕上显示结果

}

程序运行结果如下:

15 12✓

sum=27

程序运行时,从键盘输入任意两个整数,两数之间用空格分隔,✓表示回车。

【程序说明】

1. 注释

在程序的第 1 行和第 2 行,/* ... */之间的部分称为注释,可以是单行,也允许跨越多行。另外,Visual C++ 6.0 还支持另一种注释方式,即//,在程序第 3 行的末尾,//之后的内容为注释,只能占一行,称单行注释。

注释可有可无,可以出现在程序的任何地方,在需要的时候用来对程序进行说明,以增强程序的可读性。注释不会影响程序的功能和正确性,编译器在编译程序时完全忽略注释,不对注释内容进行语法检查。

2. 包含头文件

#include <stdio.h>是文件包含命令,其作用是将头文件 stdio.h 的内容读到该命令的位置处。

为了用户编程方便, C 语言编译系统提供了许多完成各种功能的库函数供用户调用 (见附录 C)。当调用库函数时, 需要在程序的开头注明该库函数所在的头文件, 以便编译系统找到该库函数的程序代码。在例 1-1 中, `scanf()` 为数据输入库函数, `printf()` 为数据输出库函数, 程序中只要用到这两个库函数, 就需要包含 `stdio.h` 头文件。

3. 定义主函数

`main()` 为主函数。构成 C 语言程序的基本单位是函数, 一个 C 语言程序可由一个或多个函数组成, 有且只有一个主函数, C 语言程序总是从 `main()` 函数开始执行, `main()` 函数执行完了, 程序也就结束了。`main()` 函数可以根据需要来调用其他函数。

一个函数由函数首部 and 函数体两部分组成。`void main()` 为函数首部, 在首部后面用一对 `{ }` 括起来的部分为函数体。此函数体内共有 4 条语句, 每条语句必须以分号 (;) 结尾, 分号是语句不可缺少的组成部分。

4. 定义变量

`int x,y,sum;` 语句中, `int` 表示要处理的数据的类型为整型, 该语句定义了 3 个整型变量 `x`、`y` 和 `sum`, `x` 和 `y` 用来存放从键盘输入的两个整数, `sum` 用来存放两数之和。

在 C 语言中, 变量必须先定义后使用, 给变量命名时需要注意以下几点:

(1) 变量名只能由字母、数字和下划线 3 种字符组成, 且第一个字符必须为字母或下划线。例如, 以下变量名是非法的: `4ab` (以数字开头)、`a$b` (含非法字符 \$)。

(2) 变量名一般采用小写字母, 并且区分大小写。例如, `sum`、`Sum` 和 `SUM` 是 3 个不同的变量名。

(3) 变量名不能为关键字。在 C 语言中, 关键字是一些预先规定的、具有特殊含义的单词。例如, `int`、`float` 等。

(4) 给变量命名时应尽量做到“见名知义”。例如, 使用 `sum`、`name`、`age` 分别表示和、姓名、年龄。

变量名、函数名等统称为标识符, 所有标识符的使用都必须满足上述规则。

5. 调用格式输入函数

`scanf("%d%d",&x,&y);` 语句中, `scanf()` 是格式输入函数, 该语句的功能是从键盘输入两个整数给变量 `x` 和 `y`, `scanf()` 函数的调用格式如下:

`scanf("格式控制字符串", 变量地址列表);`

`scanf()` 函数的格式说明如图 1-1 所示, 其中, `%d%d` 为格式控制字符串, 用来指定输入格式。`%d` 是格式说明符, 格式说明符的种类很多, 具体采用哪一个由变量的类型决定, 整型变量必须采用 `%d`。一个变量对应一个格式说明符, 因此, 变量 `x` 和 `y` 对应 `%d%d`。

变量地址列表给出了用来接收键盘数据的各个变量, 每个变量前必须加上 `&` (取地址运算符), 否则, 会出现错误。

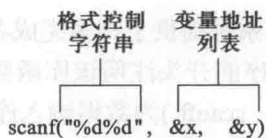


图 1-1 格式输入函数的格式说明

输入数据时，两个数据之间默认的分隔符为：空格、回车或跳格键（Tab）。除了格式说明符外，在格式控制字符串中还可以使用普通字符，输入数据时，普通字符必须原样输入。例如：

```
scanf("%d,%d",&x,&y);
```

在格式控制字符串中包含有逗号，则输入数据时，两个数据之间必须以逗号分隔，因此，输入格式为：15,12✓

6. 两数相加

语句 `sum=x+y;` 将变量 `x` 和 `y` 的值相加，并将结果赋给变量 `sum`。



变量 `x`、`y` 和 `sum` 在计算机中存放在哪儿呢？

答案是：计算机的内存。每个变量在内存中都分配有相应的存储单元，在程序中通过变量名可以对存储单元的内容进行读写操作。在例 1-1 中，定义了 3 个变量，分别占用 3 个存储单元。如果 `x` 和 `y` 的值分别为 15 和 12，则求和过程如图 1-2 所示。

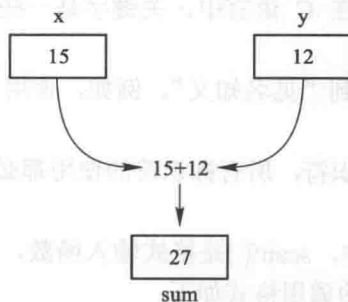


图 1-2 两数相加示意图

计算机在执行 `sum=x+y;` 语句时，会把 `x` 和 `y` 的值读到 CPU 中，执行加法运算，再把相加的结果写入 `sum` 中。



`a=a+1;` 语句将执行什么操作呢？

答案是：把变量 `a` 的值读到 CPU 中，加 1 后，再把结果写入 `a` 中。

7. 调用格式输出函数

`printf("sum=%d\n",sum);`语句中, `printf()`是格式输出函数, 该语句的功能是在屏幕上显示计算结果。`printf()`函数的调用格式如下:

`printf("格式控制字符串", 输出列表);`

`printf()`函数的使用格式与 `scanf()`函数相似。格式控制字符串用来指定输出格式, 由普通字符、格式说明符和控制字符组成, `printf()`函数的格式说明如图 1-3 所示。输出时普通字符原样显示, 格式说明符输出时被变量的值所代替, 控制字符 `\n` 的作用是“换行”。因此, 输出结果为: `sum=27`。

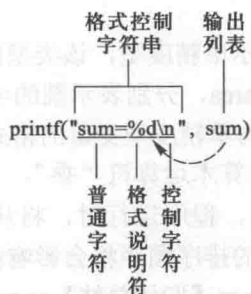


图 1-3 格式输出函数的格式说明



如果想让输出结果显示为: `15+12=27`, 应如何设置 `printf()`函数的参数?

答案是: `printf("%d+%d=%d\n",x,y,sum);`

输出时, 每个格式说明符 `%d` 依次被变量的值所代替, 普通字符 (+和=) 原样输出。

【归纳总结】

从以上程序可知, 编写 C 语言程序的一般过程如下:

- (1) 定义变量。
- (2) 设置变量的值, 可以通过键盘输入或直接赋值。
- (3) 加工处理, 得到结果。
- (4) 输出结果。

【例 1-2】从键盘输入圆的半径, 求圆的面积。

程序代码如下:

`/*文件名: p1-2.c*/`

`#include <stdio.h>`

`//注明输入/输出函数使用的头文件`

`void main()`

`//定义主函数`

`{`


```

float r,area;           //定义单精度型变量
scanf("%f",&r);         //调用库函数从键盘输入半径
area=3.14*r*r;          //计算圆的面积
printf("area=%f\n",area); //在屏幕上显示结果
}

```

程序运行结果如下:

2.1 ✓

area=13.847399

【程序说明】

(1) `float r,area;`语句中, `float` 表示单精度型, 该类型的变量可以存储带有小数位的数据。该语句定义了两个单精度型变量 `r` 和 `area`, 分别表示圆的半径和面积。

(2) `scanf("%f",&r);`语句中, `%f` 为单精度型变量的格式说明符。

(3) `area=3.14*r*r;`语句中, `*`表示算术运算符“乘”。

(4) 上述程序采用的是顺序结构, 程序运行时, 将从第一条语句开始, 逐条向下执行, 直至程序结束, 如果改变程序中语句的排序顺序将会影响程序的执行结果。

===== 【归纳总结】 =====

为了使程序美观、整齐、可读性好, 编写 C 语言程序时应尽量遵循以下规则, 养成良好的编程习惯。

(1) `{` 和 `}` 对齐, 且一般单独占一行。

(2) 使用 Tab 键缩进, 书写格式呈锯齿形, 具有良好的层次结构。

(3) 一条语句占一行。虽然 C 语言程序允许一行写多条语句, 也允许一条语句占用多行, 但为了便于阅读, 最好一条语句占一行。

(4) 有比较详细的注释。

=====

【例 1-3】 从键盘输入任意一个数, 输出该数的平方根, 输出宽度占 7 位, 并保留两位小数。

程序代码如下:

```

#include <stdio.h>       //注明输入/输出函数使用的头文件
#include <math.h>        //注明 sqrt()函数使用的头文件
void main( )            //定义主函数
{
    double a,b;          //定义变量
    scanf("%lf",&a);      //调用库函数从键盘输入数据
    b=sqrt(a);            //计算平方根
    printf("平方根为: %7.2lf\n",b); //在屏幕上显示结果
}

```