

21世纪高等学校计算机教育实用规划教材

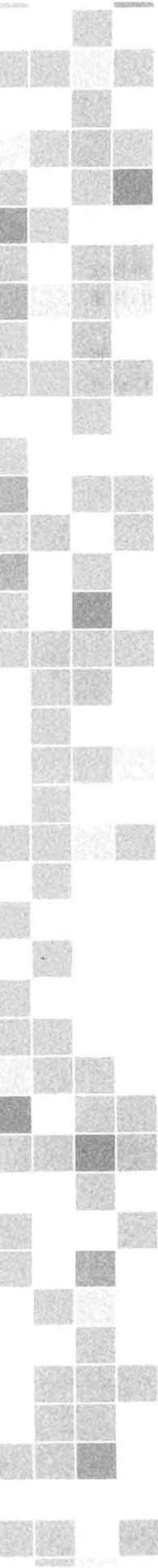
C语言程序设计案例教程

张丽华 主编

梁 田 殷联甫 副主编

清华大学出版社





21世纪高等学校计算机教育实用规划教材

C语言程序设计案例教程

张丽华 主编

梁 田 殷联甫 副主编

清华大学出版社
北京

内 容 简 介

C 语言不仅是目前应用最为广泛的一种高级程序设计语言,也是高等学校程序设计课程的首选入门语言。本书共分为 10 章,主要包括 C 语言概述、程序与算法、数据类型和表达式、程序控制结构、数组、函数、结构体与共用体、文件、位运算、综合实例等。本书采用任务驱动的方式,通过引例引出基本概念和基本方法,并以综合实例为主线,贯穿各主要章节。全书案例丰富、阐述清晰、分析透彻、层次分明,注重培养读者分析问题和解决问题的能力。

本书可作为高等学校理工类专业 C 语言程序设计课程的教学用书,也可作为计算机二级考试的培训或自学教材。为配合教学,本书配有 PPT 教学课件,各章实例源程序,习题参考答案,并有配套的《C 语言程序设计实验指导》供读者参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计案例教程/张丽华主编. —北京:清华大学出版社,2015

21 世纪高等学校计算机教育实用规划教材

ISBN 978-7-302-38798-5

I. ①C… II. ①张… 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 296312 号

责任编辑:黄 芝 薛 阳

封面设计:常雪影

责任校对:梁 毅

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:三河市君旺印务有限公司

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:18.75 字 数:467 千字

版 次:2015 年 2 月第 1 版 印 次:2015 年 2 月第 1 次印刷

印 数:1~2000

定 价:39.00 元

产品编号:061737-01

出版说明

随着我国高等教育规模的扩大以及产业结构调整的进一步完善,社会对高层次应用型人才的需求将更加迫切。各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,合理调整和配置教育资源,在改革和改造传统学科专业的基础上,加强工程型和应用型学科专业建设,积极设置主要面向地方支柱产业、高新技术产业、服务业的工程型和应用型学科专业,积极为地方经济建设输送各类应用型人才。各高校加大了使用信息科学等现代科学技术提升、改造传统学科专业的力度,从而实现传统学科专业向工程型和应用型学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时,不断更新教学内容、改革课程体系,使工程型和应用型学科专业教育与经济建设相适应。计算机课程教学在从传统学科向工程型和应用型学科转变中起着至关重要的作用,工程型和应用型学科专业中的计算机课程设置、内容体系和教学手段及方法等也具有不同于传统学科的鲜明特点。

为了配合高校工程型和应用型学科专业的建设和发展,急需出版一批内容新、体系新、方法新、手段新的高水平计算机课程教材。目前,工程型和应用型学科专业计算机课程教材的建设工作仍滞后于教学改革的实践,如现有的计算机教材中有不少内容陈旧(依然用传统专业计算机教材代替工程型和应用型学科专业教材),重理论、轻实践,不能满足新的教学计划、课程设置的需要;一些课程的教材可供选择的品种太少;一些基础课的教材虽然品种较多,但低水平重复严重;有些教材内容庞杂,书越编越厚;专业课教材、教学辅助教材及教学参考书短缺,等等,都不利于学生能力的提高和素质的培养。为此,在教育部相关教学指导委员会专家的指导和建议下,清华大学出版社组织出版本系列教材,以满足工程型和应用型学科专业计算机课程教学的需要。本系列教材在规划过程中体现了如下一些基本原则和特点。

(1) 面向工程型与应用型学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映基本理论和原理的综合应用,强调实践和应用环节。

(2) 反映教学需要,促进教学发展。教材规划以新的工程型和应用型专业目录为依据。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材建设仍然把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现工程型和应用型专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材要配套,同一门课程可以有多种具有不同内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材,教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机教育实用规划教材编委会

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前 言

C 语言既是目前应用最为广泛的一种高级程序设计语言,也是一种非常优秀的程序设计入门语言。读者一旦掌握了 C 语言,再学习其他语言就轻而易举了。本书的主要特点如下。

(1) 主要章节的内容采用任务驱动的方式,通过引例引出基本概念和基本方法,内容叙述自然,顺理成章。

(2) 将指针及其相关内容分布于各章节,而不单独设置一章。从指针的概念引入到指针变量的应用,讲解由浅入深、层层推进,便于读者理解和掌握。

(3) 以综合实例为主线,贯穿各主要章节。帮助读者在应用中加深对 C 语言基本语法和程序设计方法的理解,培养分析、解决实际问题的基本思路和方法。

本书共分为 10 章内容,第 1 章介绍 C 程序的结构及其特点;第 2 章介绍算法及其结构化程序设计;第 3 章是数据类型和表达式,介绍 C 语言常用的数据类型及运算符;第 4 章是程序控制结构,即顺序结构、选择结构和循环结构及其应用;第 5 章是数组及其应用,主要介绍一维数组、二维数组和字符数组的定义、初始化和应用,重点对指针变量访问数组进行了详细解说;第 6 章是函数及其应用,介绍函数的定义和调用、变量的作用域与生存期、编译预处理命令等,重点介绍了指针作为函数参数的使用;第 7 章介绍结构体与共用体的定义和引用,并举例说明了单链表的操作;第 8 章是文件的应用,介绍文件的打开、关闭与读写操作;第 9 章介绍位运算操作;第 10 章是综合实例,系统地介绍了利用 C 语言进行完整的应用程序设计及其实现的过程。

《C 语言程序设计案例教程》具有通俗易懂、分析透彻、开拓思路的特点,有利于读者自学。教材中所有实例的源程序代码均在 Visual C++6.0 集成环境中调试通过。本教材推荐使用时间为一学期(64 学时或 80 学时,含实验 32 学时)。通过循序渐进地系统学习,帮助读者较好地掌握 C 语言程序设计方法和技巧,为后续的学习打下扎实的基础。

程序设计课程是一门实践性较强的课程,实践环节特别重要。为了更好地培养学生的编程能力,本书写作组的老师们编写了《C 语言程序设计学习指导》作为配套辅助教材,为课堂教学、实验教学和读者自学提供全方位的支持。

本书的作者均是多年从事 C 语言程序设计教学、具有丰富实践教学经验的高校教师。本书是他们在多年教学基础上的经验总结,希望能对广大读者有所帮助。

本书由张丽华任主编,梁田、殷联甫任副主编,参与编写的人员还有刘小军、宋柱芹和张彬,在此表示感谢。

对支持本书出版的清华大学出版社表示感谢!

由于编者水平有限,不足与疏漏之处在所难免,敬请读者及同仁不吝赐教。

编 者

2014 年 9 月

目 录

第 1 章 C 语言概述	1
1.1 C 语言程序结构	1
1.1.1 简单的 C 语言程序	1
1.1.2 C 语言程序结构	4
1.1.3 C 语言程序的书写风格	5
1.2 C 语言的发展	5
1.2.1 C 语言的发展	6
1.2.2 C 语言的特点	7
1.3 C 语言程序的运行过程	8
习题	9
第 2 章 程序与算法	10
2.1 程序与算法概述	10
2.1.1 算法的概念	10
2.1.2 算法的组成要素	12
2.1.3 算法的特性	12
2.2 算法的描述	13
2.2.1 自然语言表示法	13
2.2.2 传统流程图表示法	13
2.2.3 N-S 结构化流程图表示法	14
2.2.4 伪代码表示法	14
2.3 结构化程序设计	15
2.3.1 结构化程序设计的特点与方法	15
2.3.2 结构化程序的基本结构	15
2.3.3 结构化程序设计的过程	17
2.3.4 程序设计的原则	18
习题	19
第 3 章 数据类型和表达式	20
3.1 基本字符和标识符	20

3.1.1 基本字符	20
3.1.2 关键字	20
3.1.3 标识符	21
3.2 常量与变量	22
3.2.1 常量	22
3.2.2 变量	26
3.3 数据类型	29
3.3.1 概述	29
3.3.2 基本数据类型	29
3.3.3 指针类型	31
3.4 运算符与表达式	35
3.4.1 算术运算符与算术表达式	35
3.4.2 赋值运算符与赋值表达式	37
3.4.3 关系运算符与关系表达式	38
3.4.4 逻辑运算符与逻辑表达式	39
3.4.5 条件运算符与条件表达式	41
3.4.6 逗号运算符与逗号表达式	41
3.4.7 其他运算符与表达式	42
3.5 数据类型转换	44
3.5.1 赋值类型转换	44
3.5.2 自动类型转换	44
3.5.3 强制类型转换	46
3.6 常用的输入输出函数	46
3.6.1 标准输入输出函数	46
3.6.2 字符输入输出函数	51
3.7 语句类型	53
3.7.1 说明性语句	53
3.7.2 表达式语句	53
3.7.3 控制语句	54
3.7.4 复合语句	54
3.7.5 空语句	54
3.7.6 函数调用语句	54
3.8 应用实例	55
习题	56
第 4 章 程序控制结构	58
4.1 顺序结构	58
4.2 选择结构	60
4.2.1 if 语句	60

4.2.2	switch 语句	66
4.2.3	选择结构的嵌套	71
4.3	循环结构	74
4.3.1	while 语句	75
4.3.2	do-while 语句	78
4.3.3	for 语句	81
4.3.4	循环的嵌套	84
4.4	break 和 continue 语句	87
4.4.1	break 语句	87
4.4.2	continue 语句	88
4.5	程序举例	89
4.6	综合	93
	习题	99
第 5 章	数组	100
5.1	数组概述	100
5.2	一维数组	102
5.2.1	一维数组的定义	102
5.2.2	一维数组的引用	103
5.2.3	一维数组的初始化	104
5.2.4	一维数组程序举例	105
5.3	二维数组	106
5.3.1	二维数组的定义	107
5.3.2	二维数组的引用	108
5.3.3	二维数组的初始化	109
5.3.4	二维数组程序举例	111
5.4	字符数组	112
5.4.1	字符数组、字符串和字符串结束标志	113
5.4.2	字符数组的定义和初始化	113
5.4.3	字符数组的输入输出	115
5.4.4	常用字符串处理函数	118
5.4.5	字符数组程序举例	120
5.5	数组与指针	122
5.5.1	使用指针处理数组元素	122
5.5.2	使用指针处理字符串	131
5.5.3	数组与指针程序举例	133
5.6	数组综合应用举例	134
5.6.1	查找	134
5.6.2	排序	137

5.6.3 求极值	141
5.6.4 其他	143
习题	146

第 6 章 函数 147

6.1 概述	147
6.2 函数的定义	150
6.2.1 有返回值函数的定义	150
6.2.2 无返回值函数的定义	151
6.3 函数的调用	151
6.3.1 函数调用的形式	152
6.3.2 函数的参数	152
6.3.3 被调函数的原型声明	154
6.4 函数的嵌套调用和递归调用	155
6.4.1 函数的嵌套调用	156
6.4.2 函数的递归调用	157
6.5 变量的作用域与生存期	159
6.5.1 局部变量和全局变量	159
6.5.2 变量的生存期	162
6.5.3 存储类别小结	168
6.6 指针与函数	169
6.6.1 指针变量作为函数参数	169
6.6.2 数组作为函数参数	172
6.7 返回指针的函数和函数指针	176
6.7.1 返回指针的函数	177
6.7.2 函数指针	178
6.8 main 函数的参数	181
6.9 编译预处理	182
6.9.1 宏定义	183
6.9.2 文件包含	185
6.9.3 条件编译	186
6.10 函数应用实例	187
习题	191

第 7 章 结构体与共用体 195

7.1 结构体类型与结构体变量定义	195
7.1.1 引例	195
7.1.2 结构体类型的定义	196
7.1.3 结构体变量的定义	197

7.2	结构体变量初始化与引用	199
7.2.1	结构体变量的初始化	199
7.2.2	结构体变量的引用	199
7.3	结构体数组	201
7.3.1	结构体数组的定义	202
7.3.2	结构体数组的初始化	202
7.3.3	结构体数组的使用	203
7.4	结构体指针变量	204
7.4.1	指向结构体变量的指针	204
7.4.2	指向结构体数组的指针	206
7.5	结构体与函数	207
7.5.1	结构体变量的成员作函数参数	207
7.5.2	结构体变量作函数参数	208
7.5.3	指向结构体变量的指针作函数参数	209
7.6	链表	210
7.6.1	静态链表	211
7.6.2	动态内存函数	212
7.6.3	链表的基本操作	213
7.7	共用体	220
7.7.1	共用体的概念	220
7.7.2	共用体变量的引用	221
7.7.3	共用体类型数据的说明	221
7.8	枚举类型	223
7.9	用户自定义类型	226
	习题	227
第8章	文件	228
8.1	概述	228
8.1.1	文件分类	230
8.1.2	缓冲文件系统与非缓冲文件系统	231
8.1.3	文件指针	231
8.2	文件的打开与关闭	232
8.2.1	文件的打开	232
8.2.2	文件的关闭	233
8.3	文件的读写	234
8.3.1	格式化读写函数	234
8.3.2	字符读写函数	236
8.3.3	字符串读写函数	239
8.3.4	数据块读写函数	240

8.4 文件的其他操作	242
8.4.1 文件的定位	242
8.4.2 文件的检测	245
8.5 文件应用实例	245
习题	247
第 9 章 位运算	249
9.1 计算机中的数据表示	249
9.2 位运算符	250
9.3 位段	254
9.4 位运算程序举例	255
习题	258
第 10 章 综合实例	259
10.1 应用程序的开发流程	259
10.2 综合应用程序实例	260
附录 A 运算符的优先级与结合性	274
附录 B C 语言中的关键字	276
附录 C 常用字符的 ASCII 码	277
附录 D C 语言常用库函数	279
参考文献	285

第 1 章

C 语言概述

主要知识点：

- ◆ C 语言程序结构
- ◆ C 语言的发展
- ◆ C 语言程序的运行过程

从 1946 年世界上第一台电子计算机诞生到现在,计算机的发展经历了 4 个阶段。迄今为止,在计算机中仍然采用计算机之父“冯·诺依曼”的思想——存储程序的原理,即计算机要想做任何事情,都要按照一定的“程序”去做,也就是先编写好程序,然后输入到计算机中,计算机才能按照“程序”的要求去完成。因此要想做一个能够驾驭计算机的高手,那么就一定要掌握用计算机语言编程的本领,而 C 语言是目前国际上广泛流行的计算机语言,它也是步入程序设计殿堂的敲门砖,也是学好计算机相关专业后续课程的专业基础课,目前很多计算机游戏、杀毒病毒、工具软件、控制软件等都是用它编写的。

那么 C 语言能干什么? 用 C 语言又如何编写程序呢? 下面通过实例来了解。

1.1 C 语言程序结构

在了解 C 语言程序之前,先通过几个实例对 C 语言程序结构有一个初步认识。本节主要介绍 C 语言程序的基本结构。

1.1.1 简单的 C 语言程序

“程序”是计算机完成确定任务的具体步骤。

【实例 1-1】 在屏幕上输出“I am a student!”。

```
/* 实例 1-1 */           /* 注释语句 */
#include <stdio.h>         /* 预处理命令 */
void main()               /* 主函数 */
{
    printf("I am a student !\n"); /* 输出字符串 "I am a student !" */
}
```

分析说明: C 程序最大的特点就是所有的程序都是用函数来装配的。通常情况下,函数的命名是没有限制的,但是 main 函数是一个特殊的函数,称为主函数,是所有程序运行的入口。每个 C 语言程序中有且只有一个 main 函数。

以上程序只用到输出库函数 printf(),因此在程序开头一定要有 #include <stdio.h>

预处理命令,此命令的功能是将“stdio.h”包含在当前 C 源程序中,因为库函数 printf() 属于“stdio.h”文件的一部分,换句话说就是建立了头文件“stdio.h”与当前 C 程序的链接。

程序运行结果如图 1-1 所示。

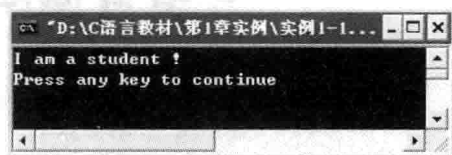


图 1-1 在屏幕上输出“I am a student !”

【实例 1-2】 求两个整数之和。

```
/* 实例 1-2 */           /* 注释语句 */
#include <stdio.h>
void main()
{
    int a,b,sum;           /* 定义整型变量 a,b,sum */
    a = 110;
    b = 120;               /* 对变量 a,b 赋值 */
    sum = a + b;           /* 计算 sum 的值 */
    printf("sum = %d\n",sum); /* 将结果 sum 以整型的方式输出 */
}
```

分析说明: 程序中如“/* 定义整型变量 a,b,sum */”是对程序的注释,包含在/* 与 */之间的内容将被编译器忽略。程序的注释一般在某行的最末尾或另起一行。注释的作用是使程序更易于理解。

在 C 语言中,变量必须先声明后使用。声明通常放在函数的起始处,在任何可执行语句之前。声明用于说明变量的属性。如“int a,b,sum;”说明了 a,b,sum 三个变量都为整型变量。

程序中的语句:

```
a = 110;
b = 120;
sum = a + b;
```

它们分别为变量 a,b,sum 赋值。各条语句以分号“;”结束。

程序运行结果如图 1-2 所示。

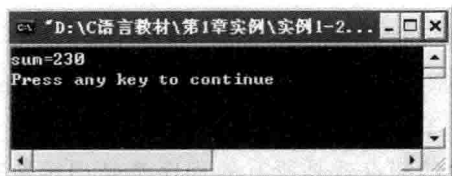


图 1-2 求两个整数之和

在实例 1-2 中,变量 a、b 的值是通过赋值语句得到的,如果要通过键盘输入 a、b 的值,该如何编写程序呢? 请看实例 1-3。

【实例 1-3】 从键盘输入两个整数的值,并求它们的和。

```
/* 实例 1-3 */           /* 注释语句 */
#include <stdio.h>
void main()
{
    int a,b,sum;           /* 定义整型变量 a,b,sum */
    printf("Enter first integer:"); /* 提示信息 */
    scanf("%d",&a);        /* 从键盘输入 a 的值 */
    printf("Enter second integer:"); /* 提示信息 */
    scanf("%d",&b);        /* 从键盘输入 b 的值 */
    sum = a + b;           /* 计算 sum 的值 */
    printf("sum = %d\n",sum); /* 将结果 sum 以整型的方式输出 */
}
```

分析说明: 程序中 scanf() 函数的功能是执行格式化输入。程序语句:

```
scanf("%d",&a);
```

作用为从键盘输入 a 的值,&a 为变量 a 的地址,即从键盘输入的内容存放到变量 a 所在的地址中。%d 是对输入参数的格式化说明,即要求输入的 a 的值为整型。有关输入输出的格式说明,请参考第 3 章中的相关内容。

程序运行结果如图 1-3 所示。

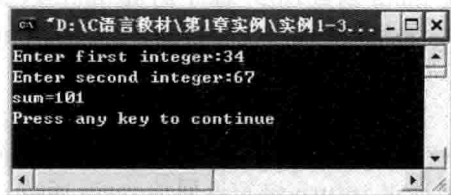


图 1-3 从键盘输入两个整数并求它们的和

【实例 1-4】 求两个数中的最大数。

```
/* 实例 1-4 */
#include <stdio.h>
void main()           /* 主函数 */
{
    int x,y,t;          /* 说明部分,定义整型变量 x,y,t */
    int max(int,int);   /* 函数声明 */
    printf("Please input x,y: \n"); /* 在屏幕上显示字符串 "Please input x,y: " */
                                /* 然后换行 */
    scanf("%d,%d",&x,&y); /* 输入 x,y 的值 */
    t = max(x,y);        /* 函数调用语句,调用后将返回值赋给变量 t */
    printf("x = %d,y = %d,max = %d\n",x,y,t); /* 输出 x,y,t 的值 */
}
int max(int a,int b)   /* 函数 max */
{
    if(a>b)             /* 条件语句 */
        return a;       /* 条件成立,返回值 a */
}
```

```

else
    return b;                /* 条件不成立,返回值 b */
}

```

程序运行结果如图 1-4 所示。

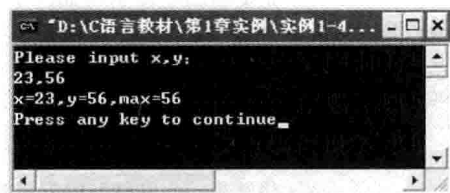


图 1-4 求两个数中的最大数

分析说明: 实例 1-4 是由两个函数组成,一个是主函数 `void main()`,一个是 `int max(int a,int b)` 函数。`void` 是函数的类型,表示此函数执行后不会返回函数值。`int max()` 函数是求 `a,b` 两个数中最大者,函数返回值是整型的。

这里使用的函数如 `printf()`、`scanf()` 等都是函数库中提供的函数。除此之外,还可以自己动手来编写一些函数。如本例中 `max` 函数的作用就是返回两数中的较大值。`main` 也是函数,不过身份有些特殊,首先它是程序的入口;其次 `main` 函数调用 `max` 函数完成任务。

虽然程序的第一行不是 `main` 函数,但是程序仍从 `main` 函数开始执行,因为 `main` 函数为程序的入口。进入 `main` 函数后,按顺序执行程序,运行至语句:

```
t = max(x, y);
```

调用函数 `max`,并将 `x,y` 的值传递给 `a,b`,若从键盘输入 12,30,则 `a,b` 的值为 12,30。

`max` 函数中的计算结果通过 `return` 语句返回给 `main` 函数。

程序语句:

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

作用为从键盘输入 `x` 和 `y` 的值,`&x` 为 `x` 的地址,即从键盘输入的内容存放到 `x` 和 `y` 所在的地址中。输入两个数时用逗号分隔。

1.1.2 C 语言程序结构

以上几个程序都是 C 语言的“源程序”。可以看出 C 语言程序的结构特点如下。

(1) C 语言程序主要由函数构成,C 语言程序中有主函数 `main()`、开发系统提供的库函数(如 `printf()`、`scanf()` 等)以及程序员自己设计的自定义函数(如 `max()` 等)等三种类型的函数。

函数由两部分组成:函数首部和函数体。实例 1-3 中的 `void main()`、`int max(int a,int b)` 都是函数首部,它包括函数的返回值类型、函数名、函数参数的类型、参数名等内容。

例如,`max` 函数的首部为:

<code>int</code>	<code>max</code>	<code>(</code>	<code>int</code>	<code>a ,</code>	<code>int</code>	<code>b)</code>
↑	↑		↑	↑	↑	↑
返回值类型	函数名		参数的类型	函数参数	参数的类型	函数参数