

```
#include<stdio.h>

int main()
{
    printf("hello world!\n");
    return 1;
}
```

王小峰 ©主编

高级语言程序设计 (C语言)



厦门大学出版社 国家一级出版社
XIAMEN UNIVERSITY PRESS 全国百佳图书出版单位



```
#include <stdio.h>
int main(void){
    char* p = NULL;
    char* tmp = "12345678";
    p = (char*)(tmp+4);
    //p[-1] = ?, p[-5] = ?.
    return 0;
}
```

```
#include <stdio.h>
int main(void){
    int a, b;
    printf("input two numbers:");
    scanf("%d%d",&a,&b);
    if(a>b)
        printf("max=%d\n",a);
    else
        printf("max=%d\n",b);
    return 0;
}
```



高级语言程序设计

(C语言)

主 编 王小峰
副主编 方 捷 王玉娟



厦门大学出版社 国家一级出版社
XIAMEN UNIVERSITY PRESS 全国百佳图书出版单位

图书在版编目(CIP)数据

高级语言程序设计:C语言/王小峰主编. —厦门:厦门大学出版社, 2016. 12
ISBN 978-7-5615-6306-9

I. ①高… II. ①王… III. ①C语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2016)第 294744 号

出版人 蒋东明
责任编辑 眭蔚
封面设计 蒋卓群
责任印制 许克华

出版发行 厦门大学出版社
社址 厦门市软件园二期望海路 39 号
邮政编码 361008
总编办 0592-2182177 0592-2181406(传真)
营销中心 0592-2184458 0592-2181365
网 址 <http://www.xmupress.com>
邮 箱 xmupress@126.com
印 刷 三明市华光印务有限公司

开本 787mm×1092mm 1/16
印张 22.5
字数 536 千字
版次 2016 年 12 月第 1 版
印次 2016 年 12 月第 1 次印刷
定价 45.00 元

本书如有印装质量问题请直接寄承印厂调换



厦门大学出版社
微信二维码



厦门大学出版社
微博二维码

前 言

在当今的信息化时代,计算机编程已经成为一种必备能力,人人都有通过编程来提高解决问题的能力与效率的需求。C语言作为一种经典且经久不衰的编程语言(诞生于1971年,2016年编程语言TIOBE指数仍然排名第二),一直是国内外绝大多数高校的指定入门教学编程语言。

本书的编写团队浸淫于计算机编程教学工作十余年,教授过的计算机编程/脚本语言遍及C、C++、Win32 Api/MFC、Java、JavaScript、Php、Python、R等,具有丰富的计算机编程教学与实践经验。经过多个回合的探讨,大家一致觉得有必要在多种编程语言之上总结出一种“形而上学”的计算机编程语言的教学方法,并将该方法实践于C语言的教学过程当中。

本书首先提出“为什么要编程”和“编程与编程平台”这两个议题,并给出了最本质的答案:如果将计算机当作一个交流对象,非编程手段的计算机常规操作只相当于向计算机打手势这种隔靴搔痒的交流;而通过编程则可以与计算机直接对话,能让计算机最大限度地为我们服务。另外,编程是有场景的,这就是编程平台。以Web前端编程语言JavaScript为例,其场景是浏览器窗口的客户区,这就决定了它可以操纵客户端页面中的html和css,但不能访问客户端本地资源(如硬盘等)。综上,我们首先要明白什么是编程,为什么要编程,某种编程语言能干什么,又不能干什么。考虑到我们要将侧重点放在编程语言本身,因此本书采用了似乎稍显过时但最为简单的DOS应用程序为编程平台,以此来展开C语言程序设计的教学工作。

本书在国内外首次提出了“要将计算机编程语言作为一种语言来学习”的观点。可以想见,计算机编程语言既然是一门语言,将其作为语言来学习自然可以收到事半功倍的效果。我们选择大学生最为熟悉的英语语法来做类比教学,通过已知来了解未知,可以极大地提升学习的效率和自信心。

我们将英语和计算机编程语言作如下类比:英语被称为动词的语言,名词(相当于C语言的数据类型)和以动词(相当于C语言中的运算符)为核心的谓语构成简单句(相当于C语言中的表达式语句),为表达更复杂的逻辑,英语语法又扩充

出了并列句、复合句等语法结构(相当于C语言中的流程控制)。我们可以很快地发现,几乎所有讲述计算机编程语言的书籍,其前几章必然是“数据类型”“运算符”“流程控制”,这就使得读者可以很快掌握住学习主线,知其然也知其所以然。稍作思考,就可以发现其实只要有了数据类型、运算符、流程控制,作为一种程序设计语言则已然具备了其核心功能,后续语法功能可视为进一步的增强与扩展。例如数组是为了批量表示和处理相同类型的一组数据,结构体是为了封装逻辑相关的一组数据(类型可不相同),函数是为了提高代码复用效率,指针是为了跨越名字空间访问数据,文件是为了将数据在外存中长期保存,等等。

综上,本书的特点是以“计算机编程的目的”“编程语言和编程平台的关系”“编程语言的语法功能结构”为三大主线展开教学,以期读者能够在学习许多编程语言都具有的语法结构的同时,能自然而然地进行宏观思考和无障碍的知识整理,这将使得读者具备明晰的学习主线,大幅提高其学习效果与效率。

在本书的编写分工上,深圳大学传播学院网络与新媒体的王小峰老师负责全书的规划与统稿,担任主编并参与编写第6、7、8章;福建师范大学福清分校电子与信息工程学院方捷老师编写第1、2、3、4、5章;中山大学新华学院信息科学系王玉娟老师编写第9、10、11、12章和全部附录。由于时间仓促,作者水平有限,本书难免存在遗漏与错误,作者敬请读者批评与指正,我们将会在后续的工作中不断地调整与改进!

王小峰

2016年10月27日

于深圳大学文山湖

目 录

第 1 章 C 语言程序设计和编程平台概述	1
1.1 计算机语言和程序设计	1
1.1.1 机器语言	1
1.1.2 汇编语言	1
1.1.3 高级语言	2
1.2 C 语言程序简介	2
1.2.1 C 语言简介	2
1.2.2 简单的 C 程序举例	2
1.2.3 C 语言程序的基本特点	4
1.3 C 程序设计和编程平台	5
1.3.1 编程语言和编程平台	5
1.3.2 主流平台上的 C 开发流程概述	6
1.4 DOS 平台下的 C 开发	8
1.4.1 Turbo C 编程环境	8
1.4.2 编辑、编译、链接、运行第一个 C 程序	10
1.4.3 运行 C 程序前的 Directories 选项设置	12
1.4.4 编码规范及编程习惯	13
1.5 Windows 平台下的 C 开发	13
1.5.1 Windows 的窗口本质	14
1.5.2 Windows 的消息驱动机制	14
1.5.3 安装和使用 Visual C++ 6.0 IDE 环境	16
1.5.4 Windows 程序的框架结构	20
1.6 Linux 平台下的 C 开发	24
总结与提高	25
第 2 章 C 语言的基本数据类型及其基本操作	27
2.1 C 语言的数据类型	27
2.1.1 基本数据类型	27
2.1.2 构造类型	27
2.1.3 指针类型	28

2.1.4	空类型	28
2.2	常量和变量	28
2.2.1	常量	28
2.2.2	变量	29
2.3	整型数据	30
2.3.1	整型常量的表示	30
2.3.2	整型常量的类型	31
2.3.3	整型变量	32
2.3.4	整型常量的类型匹配	34
2.4	实型数据	35
2.4.1	实型常量的表示	35
2.4.2	实型变量	35
2.5	字符型数据	37
2.5.1	字符型常量	37
2.5.2	字符串常量	38
2.5.3	字符变量	39
2.5.4	字符数据的存放形式及使用方法	39
2.6	算术运算符与算术表达式	40
2.6.1	基本算术运算符	40
2.6.2	算术表达式及算术运算符的优先级和结合性	41
2.6.3	逻辑表达式	41
2.6.4	自增、自减运算符	42
2.7	赋值运算符与赋值表达式	44
2.7.1	赋值运算符	44
2.7.2	复合赋值运算符	45
2.7.3	赋值表达式	46
2.8	逗号运算符与逗号表达式	46
2.9	位运算符	47
2.9.1	按位与(&)	47
2.9.2	按位或()	48
2.9.3	按位异或(^)	48
2.9.4	按位取反(~)	49
2.9.5	移位运算(>>、<<<)	49
2.10	基本的输入输出函数	49
2.10.1	printf 函数	50
2.10.2	scanf 函数	54
2.10.3	putchar 函数与 getchar 函数	56

总结与提高	57
第 3 章 流程控制	60
3.1 顺序结构.....	60
3.1.1 顺序结构的概念.....	60
3.1.2 顺序结构程序举例.....	61
3.2 选择结构.....	63
3.2.1 选择结构的概念.....	63
3.2.2 关系表达式和逻辑表达式.....	64
3.2.3 if 语句	65
3.2.4 switch 语句	71
3.2.5 条件运算符和条件表达式.....	72
3.3 循环结构.....	73
3.3.1 循环结构的概念.....	73
3.3.2 while 语句	74
3.3.3 do...while 语句	78
3.3.4 for 语句	79
3.3.5 break 语句、continue 语句和 goto 语句	81
总结与提高	83
第 4 章 数 组	87
4.1 一维数组的定义、初始化和引用	87
4.1.1 一维数组的定义方式.....	87
4.1.2 一维数组的初始化.....	89
4.1.3 一维数组元素的引用.....	89
4.1.4 一维数组程序举例.....	90
4.2 二维数组的定义、初始化和引用	94
4.2.1 二维数组的定义.....	94
4.2.2 二维数组的初始化.....	96
4.2.3 二维数组元素的引用.....	96
4.2.4 二维数组程序举例.....	97
4.3 字符数组与字符串	101
4.3.1 字符数组与字符串的关系	101
4.3.2 字符数组的定义	101
4.3.3 字符数组的初始化	102
4.3.4 字符数组的引用	103
4.3.5 字符数组的输入输出	104
4.3.6 字符串处理函数	106

4.3.7	字符串的输入输出	110
4.3.8	程序举例	110
	总结与提高	115
第5章	结构体和共用体	118
5.1	结构体类型与结构体变量	118
5.1.1	结构体类型的定义	118
5.1.2	结构体变量的定义	119
5.1.3	结构体变量的初始化	122
5.1.4	结构体变量的引用	123
5.1.5	结构体变量的输入和输出	124
5.2	结构体数组	126
5.2.1	结构体数组的定义	126
5.2.2	结构体数组的初始化	128
5.2.3	结构体数组的引用	130
5.2.4	结构体数组的输入和输出	130
5.3	结构体指针	132
5.3.1	指向结构体变量的指针	132
5.3.2	指向结构体数组的指针	134
5.4	结构体作为函数参数	135
5.4.1	结构体变量作为函数参数	135
5.4.2	结构体指针变量作为函数参数	137
5.4.3	函数的返回值为结构体类型	138
5.5	链表	140
5.6	共用体	141
5.6.1	共用体的概念	141
5.6.2	共用体类型与共用体变量的定义	141
5.6.3	共用体变量的初始化和引用	143
5.6.4	共用体类型数据的特点	143
5.6.5	共用体的应用	144
5.7	枚举类型	144
5.8	用 typedef 定义类型	146
	总结与提高	147
第6章	函数	149
6.1	概述	149
6.1.1	什么是模块化	149
6.1.2	什么是函数	150
6.2	函数的定义与分类	151

6.2.1	函数的定义	151
6.2.2	函数的参数和返回值	152
6.2.3	函数的分类	155
6.3	函数的调用	156
6.3.1	函数调用的一般形式	156
6.3.2	函数调用的方式	157
6.3.3	对被调用函数的声明和函数原型	157
6.3.4	嵌套调用	159
6.3.5	递归调用	160
6.3.6	程序设计举例	161
6.4	常见的库函数	165
6.4.1	库函数概述	165
6.4.2	字符与字符串函数	165
6.4.3	简单数学函数	166
6.4.4	基本屏幕控制函数*	167
6.5	变量的性质	171
6.5.1	变量的作用域(可见性)概述	172
6.5.2	变量的生命期(存在性)概述	172
6.6	变量的作用域(结合变量的性质)	172
6.6.1	局部变量	172
6.6.2	全局变量	173
6.6.3	全局变量作用域的扩展和限制	175
6.6.4	总结	178
6.7	变量的生命期(结合变量的性质)	178
6.7.1	动态局部变量	178
6.7.2	静态局部变量	179
6.8	内部函数和外部函数	181
6.8.1	外部函数	182
6.8.2	内部函数	182
6.9	多文件程序——项目*	182
6.10	怎样创建项目、自己的库函数*	183
6.10.1	创建并运行项目	183
6.10.2	创建自己的库函数	183
6.11	程序设计举例	185
	总结与提高	187
第7章	指 针	190
7.1	指针的基本概念	190

7.1.1	预备知识	190
7.1.2	指针	192
7.1.3	指针其名	192
7.1.4	变量的指针与指针变量	192
7.2	指针变量的定义和赋值	194
7.2.1	指针变量的定义	194
7.2.2	指针变量的赋值	195
7.2.3	void 指针	196
7.3	指针变量的使用	197
7.3.1	与指针相关的运算符	197
7.3.2	变量的存取方式	198
7.3.3	停下来思考一下	198
7.3.4	指针变量作为函数参数	199
7.4	指针与数组	200
7.4.1	数组和数组元素的指针	200
7.4.2	指向数组和数组元素的指针变量	201
7.4.3	数组元素的引用	203
7.4.4	数组名作为函数参数	204
7.4.5	字符串的指针和指向字符串的指针变量	206
7.4.6	指针数组	208
7.4.7	指针与二维数组	210
7.5	指向指针的指针	213
7.5.1	指向指针的指针	213
7.5.2	定义指向指针变量的指针变量	213
7.5.3	指向指针的指针变量的应用	214
7.6	指针与结构	216
7.6.1	指向结构变量的指针/指针变量	217
7.6.2	指向结构体数组的指针/指针变量	217
7.6.3	指向结构体的指针作为函数参数	218
7.7	指针与函数	219
7.7.1	返回指针类型的函数	219
7.7.2	函数的指针和指向函数的指针变量	220
	总结与提高	221
第 8 章	指针的应用——链表	226
8.1	链表概述	226
8.2	简单静态链表	227
8.3	动态链表和动态内存分配函数	228

8.3.1 动态链表	228
8.3.2 动态内存分配函数	229
8.3.3 利用指针和动态内存分配函数实现不定长数组	229
8.4 建立动态链表	230
8.5 对链表的插入与删除操作	233
8.5.1 对链表的插入操作	233
8.5.2 对链表的删除操作	235
8.6 链表综合应用	236
总结与提高	241
第 9 章 编译预处理	243
9.1 宏定义	243
9.1.1 不带参数的宏定义	243
9.1.2 带参数的宏定义	245
9.2 文件包含	248
9.3 条件编译	249
9.3.1 格式 1	250
9.3.2 格式 2	250
9.3.3 格式 3	251
总结与提高	251
第 10 章 文 件	253
10.1 C 文件概述	253
10.1.1 二进制文件和文本文件	253
10.1.2 二进制文件和文本文件的比较	254
10.2 文件的打开与关闭	255
10.2.1 文件的打开(fopen 函数)	255
10.2.2 文件的关闭 fclose 函数)	258
10.3 文件的读写	258
10.3.1 fscanf 函数和 fprintf 函数	258
10.3.2 fread 函数和 fwrite 函数	260
10.3.3 fgetc 函数和 fputc 函数	261
10.3.4 其他读写函数	264
10.4 文件的定位	266
10.4.1 rewind 函数	266
10.4.2 fseek 函数	267
10.4.3 ftell 函数	268
10.5 文件的状态	269
10.5.1 feof 函数	269

10.5.2	ferror 函数	269
10.5.3	clearerr 函数	269
10.6	文件综合应用:个人小金库的管理	270
10.6.1	顺序文件和随机文件.....	270
10.6.2	需求及功能分析.....	270
10.6.3	源程序.....	271
	总结与提高.....	274
第 11 章	位运算	277
11.1	位运算的类型.....	277
11.1.1	按位与	277
11.1.2	按位或	278
11.1.3	按位异或	279
11.1.4	取反	279
11.1.5	左移	279
11.1.6	右移	279
11.2	位运算举例.....	280
11.3	位段.....	281
	总结与提高.....	283
第 12 章	综合实例	285
12.1	贪食蛇游戏.....	285
12.1.1	程序说明.....	285
12.1.2	源程序代码.....	286
12.2	学生成绩管理系统.....	291
12.2.1	程序说明.....	291
12.2.2	源程序代码.....	292
12.3	学籍管理系统.....	308
12.3.1	函数简介.....	308
12.3.2	编程算法思路.....	309
12.3.3	设计小结.....	310
12.3.4	源程序代码.....	310
	总结与提高.....	328
附录 1	C 语言语法之“大局观”	329
附录 2	ASCII 码表及其中控制字符的含义	331
附录 3	C 语言中的关键字	332
附录 4	C 语言运算符的优先级与结合性	333
附录 5	常用库函数	334
附录 6	常见错误分析及处理方法	340

C 语言程序设计和编程平台概述

“学编程要从娃娃抓起”，这句话突显了如今社会对计算机专业人才的迫切需求。大家可能会有这样一个疑问：“我早就通过鼠标、键盘等设备来操纵计算机完成工作了，为什么还学编程？”可以这样简单地回答这个问题：鼠标、键盘的圈圈点点就好比打手势，这样的确能完成一些简单工作（操作），但显然不能直接用语言来与计算机交流（编程）。这就形象地道出了“通过计算机语言编程，可以比操作计算机完成更复杂的任务”这样一个道理。

1.1 计算机语言和程序设计

你对我说了一番话，让我去解决某问题。这番话就是你编写的一个让我去工作的“清单”，而这个清单显然采用我们共识的语言——汉语来组织、编写。

用某种计算机语言（例如 C 语言）写出的让计算机执行任务的清单就是所谓的“程序”，该过程就是基于某种计算机语言的程序设计（编程）。

自计算机问世以来，计算机语言的发展大致经历了以下几个阶段。

1.1.1 机器语言

机器语言是最底层的计算机语言。每一条机器指令都是二进制数符 0 和 1 组合起来的编码。程序中的每一条指令包括操作码和地址码，告诉计算机做何种操作以及被操作的对象存放在什么位置。用机器语言编写的程序，计算机硬件可以直接识别。

机器语言程序由二进制数符 0 和 1 组成，用它编写的程序能被对计算机硬件直接识别，执行效率高；但二进制数序列“难记、难写、难调试”，编写的程序可移植性差。

1.1.2 汇编语言

为了克服机器语言的缺点，人们用有意义的符号来表示机器指令，这种用助记符将机器语言“包装”后的语言称为汇编语言。

汇编语言程序对程序员（人）来说可读性好，容易调试；但必须由专门的翻译程序（汇编程序）翻译成机器语言后，计算机才能识别并执行。如图 1-1 所示。

汇编语言与机器语言一一对应，移植性同样不好。

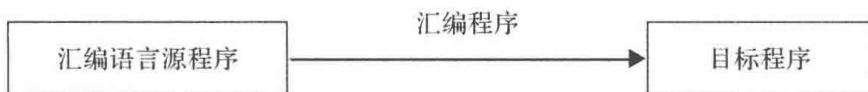


图 1-1 汇编程序的作用

1.1.3 高级语言

高级语言是一种更接近于自然语言的计算机语言,包括 Fortran、Basic、Pascal、Cobol 及 C 语言等。高级语言有各自的语法,独立于具体机器,移植性好。为了使用高级语言编写的程序能够在不同的计算机系统上运行,首先必须将程序翻译成运行程序的计算机特有的机器语言。在高级语言和机器语言之间执行这种翻译任务的程序叫作编译器(compiler)。当在计算机上安装 Turbo C 或其他高级语言运行环境时,实际上安装的内容中很重要的一部分就是该语言的编译器。

表 1-1 三类语言的对比

类别	开发效率	执行效率	可移植性
机器语言	低	高	不可移植
汇编语言	较低	较低	不可移植
高级语言	高	低	可移植

1.2 C 语言程序简介

1.2.1 C 语言简介

C 语言是为描述和实现 UNIX 操作系统而设计的,现在已成为一种成熟的通用编程语言,并被广泛应用于多种机型(如个人计算机、工作站和大型机)和操作系统(如 DOS、Windows、Linux 和 UNIX)上。C 语言既可以处理数据库、网络、图形、图像等,又适合在工业控制、自动检测等方面应用。

1983 年美国国家标准化协会(ANSI)根据 C 语言自问世以来的各种版本,对 C 语言的发展和扩充制定了新的标准,称为 ANSI C。1987 年 ANSI 又公布了新标准,即 87ANSI C。

1.2.2 简单的 C 程序举例

例 1-1 如下 C 程序实现从键盘上输入三个数,求出这三个数的最大数。

```

#include <stdio.h>
/***** Cprogl. C By Wang Xiaofeng *****/
/* This program is designed to seek the maximum from three numbers */

```

```

**** College of communication, Shenzhen University ****
void main()
{
    int num1,num2,num3,max2,max3;    /* 定义存放三个数及较大值的变量 */
    printf("Please input three numbers:"); /* 输入变量值的提示信息 */
    scanf("%d %d %d",&num1,&num2,&num3); /* 输入三个整数 */
    max2=seek_max(num1,num2);        /* 求前两个数的较大数 */
    max3=seek_max(max2,num3);/* 再求前两个数的较大数与第三个数的较大数 */
    printf("\nThe Maximum among %d,%d and %d is %d.",num1,num2,num3,max3);
                                    /* 输出三个数及其最大值 */
}

int seek_max(int x,int y)    /* 求两个数的较大数的函数定义 */
{
    int z;                    /* 定义中间变量 z */
    if(x>y) z=x;              /* 如果 x>y,那么把 x 赋给 z */
    else z=y;                 /* 否则,把 y 赋给 z */
    return z;                 /* 函数返回 x,y 的较大值 */
}

```

针对例 1-1,做如下几点说明:

1. 预处理命令 #include

例 1-1 程序的第一行是一个预处理命令,用来把 Turbo C 预先提供的与标准输入输出相关的头文件 stdio.h 包含到程序中。

Turbo C 提供了多个头文件,大多数函数可在头文件 math.h 中找到,而当要进行图形处理时,则必须包含头文件 graphics.h。在计算机加载安装 Turbo C 之后,可以在 TC\INCLUDE 目录中查找到 Turbo C 提供的所有头文件。

2. C 程序中的注释

本源程序中包含了多处注释,在 /* 和 */ 之间的内容是注释部分,不参与程序的编译和执行,只是起到说明作用,增强了程序的可读性。

最前面的三行注释是对程序作者、功能、作者所属单位进行说明,而函数 main() 中的注释则是对各语句功能的说明。

3. C 程序由函数组成

一个 C 程序可以包含一到多个的函数,每一个能单独运行的 C 程序都必须有且只能有一个 main 函数作为程序的主控函数,称为主函数。main 函数是 C 语言编译系统使用的专用名字。程序从 main 函数的第一条可执行语句开始执行。

本例中包含了两个函数,即 main 和 seek_max。函数由函数头和函数体两部分组成,例子

中的两个函数的函数头分别是 `main()` 和 `int seek_max(int x,int y)`, 函数体是函数头后面花括号 `{}` 中的内容。

程序中还调用了头文件 `stdio.h` 中的输入函数 `scanf()` 和输出函数 `printf()`。

4. 标识符与保留字

在 C 程序中用标识符来表示函数、类型、变量、符号常量及语句标号的名称。C 程序中的标识符以英文字母 (`a~z` 或 `A~Z`) 或下划线开头, 由英文字母、数字 (`0~9`) 和下划线构成, 区分大小写字母。不同的编译程序对标识符有不同的规定, Turbo C 2.0 规定标识符的字符个数不超过 32, 建议初学者在标识符命名时不宜过长, 要注意简洁、清晰, 让人见名知意。

本例中定义了存放三个数的变量, 分别是 `num1`、`num2` 和 `num3`, 存放前两个数较大值及三个数最大值的变量分别是 `max2` 和 `max3`。

C 语言规定了一些具有特定含义的标识符, 即关键字, 一共有 32 个, 请参见附录 3。例 1-1 中的 `void`、`int`、`if`、`else`、`return` 都是关键字。

5. 输入与输出

在 C 程序中, 可以通过调用编译系统提供的函数 `scanf()` 从键盘输入一些数据, 而用函数 `printf()` 向显示屏幕输出数据。

如本例中的语句 `scanf("%d %d %d",&num1,&num2,&num3)` 表示从键盘上输入三个整数(用 `%d` 表示整数格式符) 给变量 `num1`、`num2` 和 `num3`。这里的“&”是用来取变量地址的运算符。

程序中的 `printf("\nThe Maximum among %d,%d and %d is %d.", num1, num2, num3, max3)`; 是用来按格式要求输出最后结果的。“\n”在 C 语言中表示回车换行符。

执行例 1-1 的程序, 首先将出现如下的提示信息:

Please input three numbers:

当在提示信息后面输入用空格隔开的三个数 23、45、33 并按下回车时, 程序执行结果如下:

Please input three numbers: 23 45 33↵

The Maximum among 23,45 and 33 is 45.

其中加下划线部分表示程序执行过程中用户输入的内容, ↵表示回车(全书都以↵表示回车)。

1.2.3 C 语言程序的基本特点

C 语言程序的特点主要有以下几个方面:

(1) C 语言是一种结构化的程序设计语言, 包括顺序结构、选择结构和循环结构。

(2) C 语言中提供的数据类型有: 有符号基本整型、有符号短整型、有符号长整型、无符号基本整型、无符号短整型、无符号长整型、双精度实型、单精度实型、字符型、枚举类型、指针类型、文件等, 以及由上述类型构造的类型, 如数组、结构体、共用体等。

(3) C 语言具有与汇编语言的接口。在具有汇编编译器的情况下, 可以在 C 程序中调用汇