

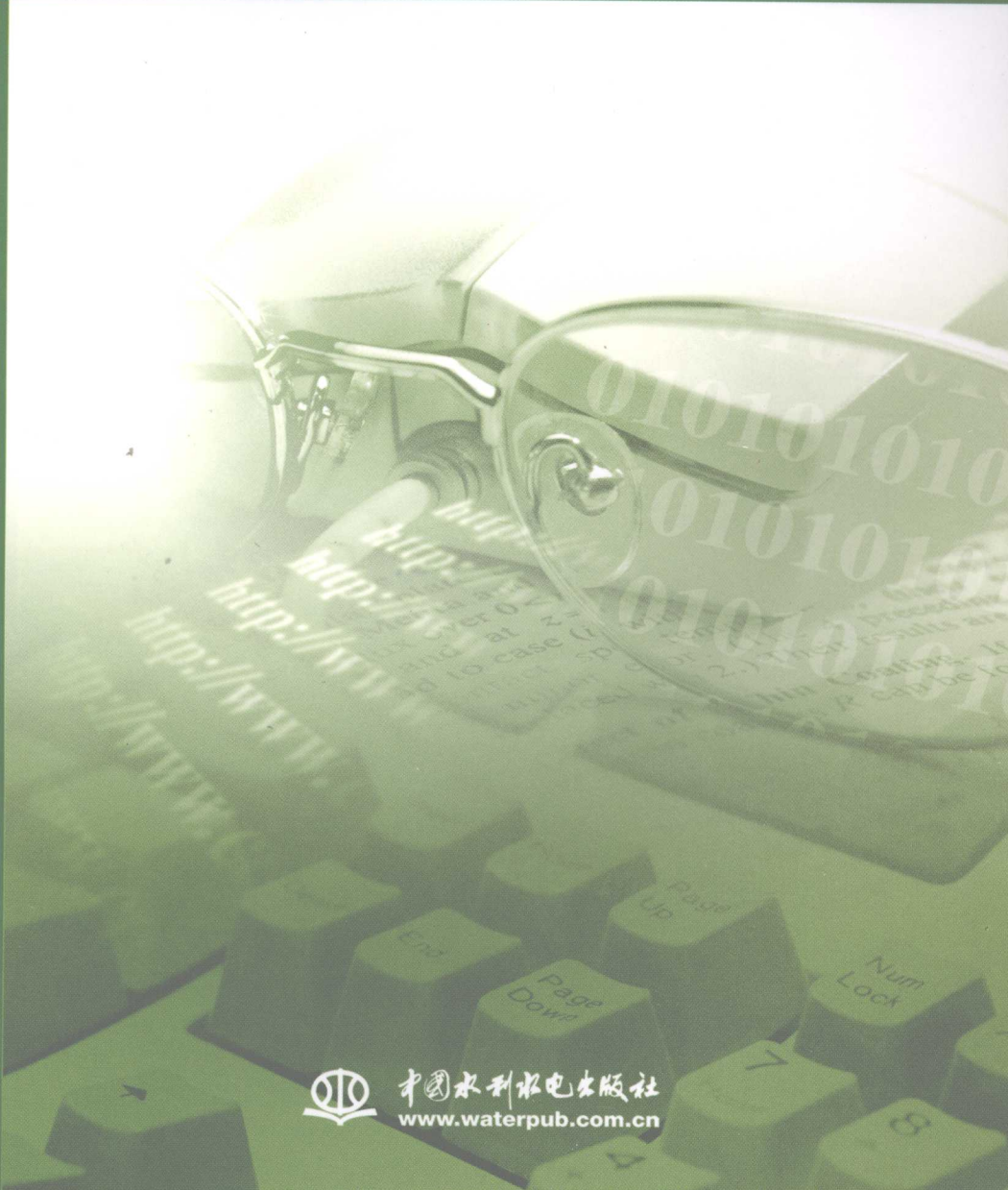
21

世纪 高职高专规划教材

程序设计基础——C语言

武春岭 主 编 高灵霞 周建辉 程书红 何 欢 副主编

21SHIJIGAOZHIGAOZHIGUANGUIHUAJIAOCAI



中国水利水电出版社
www.waterpub.com.cn

程序设计基础——C 语言

武春岭 主 编

高灵霞 周建辉 程书红 何 欢 副主编

中国水利水电出版社

内 容 提 要

本书是借助于“中澳职教项目”的重要职教思想编写的,突出体现了“以学生为中心,以能力为本位”的核心思想。以大量实例为主载,以朴实言简意赅的理论描述为驱动,以画龙点睛的提示突出重点,以灵活多样的技能习题实现过程考核,较好地实现了“寓教于乐、寓学于练”的学习效果。

本书共分为9章,主要包括C语言概述、C语言程序设计基础知识、程序设计初步、循环结构程序设计、模块化程序设计、数组、指针、结构体和共用体、文件。涵盖了C语言程序设计的基础内容,有效地满足了当前高职院校计算机程序设计基础教育的目的。本书每章后均有习题和实训内容,配套辅导书《C语言程序设计基础习题集》(中国水利水电出版社出版)补充了大量习题和典型习题分析,并配有大量综合测试试卷和答案,可有效增加学生技能训练项目,达到巩固提高的目的,同时也为应对全国计算机等级考试二级考试提供练兵场。

本书适合作为大专院校计算机专业的教材,尤其适用于高职高专类教学使用,也适用于各类成人学历教育,以及对C语言有兴趣的各类读者自学使用。

本书配有免费电子教案可以从中国水利水电出版社网站下载,网址为:
<http://www.waterpub.com.cn/softdown/>。

图书在版编目(CIP)数据

程序设计基础: C语言 / 武春岭主编. —北京: 中国水利水电出版社, 2008

21世纪高职高专规划教材

ISBN 978-7-5084-5191-6

I. 程… II. 武… III. C语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2008)第032686号

书 名	程序设计基础——C语言
作 者	武春岭 主 编 高灵霞 周建辉 程书红 何 欢 副主编
出版 发行	中国水利水电出版社(北京市三里河路6号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787mm×1092mm 16开本 13.75印张 331千字
版 次	2008年4月第1版 2008年4月第1次印刷
印 数	0001—4000册
定 价	25.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换
版权所有·侵权必究

前言

C 语言是当今最流行的程序设计语言之一。C 语言提供了丰富的数据结构,可以实现复杂的算法,胜任各种类型的开发工作。虽然随着 C++ 语言的出现, C 语言的使用有所“降温”,但是 C 语言以其独特的优势仍然活跃在嵌入系统的开发领域,具有不可替代的作用,同时,它也是当前学习程序设计思想最好的阶梯。

C 语言提供的以函数为单位的开发思想及结构化控制语句,正好体现了前期软件工程“模块化”和“结构化”的思想。一旦掌握了 C 语言基本的程序设计方法,不管学习者是继续走 C (包括 C++) 开发之路,还是转向其他开发语言的学习,都会具备较好的程序设计基础,能够与时俱进,紧跟软件行业方向,实现自己的“可持续发展”。

本书是借助于“中澳职教项目”的重要职教思想编纂的,突出体现了“以学生为中心,以能力为本位”的核心思想。该书正式出版前,已经在重庆电子科技职业学院、重庆电子工程职业学院试用了 5 年时间。经过实际应用和不断改进,目前已相对完善和成熟,并得到了广大师生的一致好评。

本书共 9 章内容。主要包括 C 语言概述、C 语言程序设计基础知识、程序设计初步、循环结构程序设计、模块化程序设计、数组、指针、结构体和共用体、文件。这些内容是 C 语言最基本的东西,也是全国计算机二级(C 语言)考试大纲的基本考试内容。学习程序设计一定要多思考(思考解题思路)、多看例程代码(通过阅读掌握更多的解题及编程方法)、有意识地多写程序代码(可通过课后题目),多仔细研究程序运行结果。通过这些锻炼,相信你会得到满意的效果!

本书由武春岭任主编并执笔,高灵霞、周建辉、程书红、何欢任副主编。此外,王静、李贺华、周璐璐、陈顺立、武群岭、刘建立、段立文参与了部分章节的编写工作,最后由武春岭统稿、定稿。同时,该书在编写过程中,还得到重庆电子工程职业学院龚小勇副教授、吕红副教授、陈鸿副教授的支持和指导,重庆师范大学罗凌老师、重庆电子职业技术学院徐受蓉教授也给予了很多帮助,在此一并感谢!。

由于作者水平有限,书中不足之处和错误在所难免,恳请广大读者不吝指正,我们将在再版时及时改进。主编的 E-mail: wuch50@126.com。

编者

2008 年 2 月

目 录

前言

第 1 章 C 语言概述	1
1.1 C 语言出现的历史背景	1
1.2 认识 C 程序	2
1.3 C 语言程序的特点	4
1.4 C 程序的开发过程	5
1.5 Turbo C 2.0 编辑环境应用实例	5
实训	7
技能训练	9
第 2 章 程序设计基础知识	11
2.1 概述	11
2.1.1 引言	11
2.1.2 C 语言的数据类型	12
2.2 标识符、常量和变量	12
2.2.1 标识符的概念	12
2.2.2 常量的概念	13
2.2.3 变量的概念	13
2.3 基本数据类型及其修饰符	15
2.3.1 整型数据类型	15
2.3.2 实型数据类型	15
2.3.3 字符型数据类型	16
2.3.4 字符串的概念	18
2.3.5 基本类型修饰符	19
2.4 基本数据的输入与输出	20
2.4.1 输出在 C 语言中的实现	21
2.4.2 输入在 C 语言中的实现	21
2.4.3 字符数据的专用输入输出函数	22
2.5 运算符和表达式	23
2.5.1 算术运算符与算术表达式	24
2.5.2 赋值运算符与赋值表达式	25
2.5.3 关系运算与逻辑运算	27
2.6 数据类型转换	30
2.6.1 自动类型转换	30

20	2.6.2 强制类型转换	31
20	2.7 两种特殊的运算符和表达式	31
60	2.7.1 逗号运算符与逗号表达式	32
05	2.7.2 条件运算符和条件表达式	32
85	2.8 综合应用示例	33
65	实训	34
65	技能训练	35
第3章 程序设计初步		38
85	3.1 三种基本程序设计结构	38
95	3.1.1 结构化程序设计	38
08	3.1.2 C 语言的语句	40
08	3.2 顺序程序设计示例	41
58	3.3 选择结构 if 语句	43
88	3.3.1 if 语句格式	43
48	3.3.2 if 语句的嵌套	44
28	3.3.3 if 语句的应用示例	44
28	3.4 多路选择结构 switch 语句	47
68	3.4.1 switch 语句的格式	47
09	3.4.2 switch 语句应用示例	47
19	3.5 综合应用示例	49
89	实训	51
49	技能测试	54
第4章 循环结构程序设计		57
001	4.1 循环程序结构	57
801	4.2 goto 语句	57
801	4.2.1 goto 语句格式	57
801	4.2.2 goto 语句的应用	58
201	4.3 while 语句	58
801	4.3.1 while 语句格式	58
801	4.3.2 while 语句的应用	59
011	4.4 do-while 语句	60
511	4.4.1 do-while 语句格式	60
511	4.4.2 do-while 语句的应用	61
511	4.5 for 语句	61
411	4.5.1 for 语句格式	61
411	4.5.2 for 语句的应用	61
811	4.6 break 和 continue 语句	65

16	4.6.1 break 语句	65
16	4.6.2 continue 语句	65
58	4.7 综合应用示例	66
58	实训	70
68	技能测试	73
74	第5章 模块化程序设计——函数	76
76	5.1 概述	76
78	5.2 函数的定义与声明	78
78	5.2.1 函数的定义	78
78	5.2.2 函数的声明	79
80	5.3 函数的调用	80
80	5.3.1 函数的一般调用方式	80
82	5.3.2 函数的嵌套调用与递归调用	82
83	5.3.3 函数参数	83
84	5.4 变量类型	84
85	5.4.1 局部变量	85
85	5.4.2 全局变量	85
86	5.4.3 变量的存储方式	86
90	5.5 编译预处理	90
91	5.5.1 宏定义	91
93	5.5.2 文件包含	93
94	5.5.3 条件编译	94
96	实训	96
100	技能训练	100
103	第6章 数组	103
103	6.1 一维数组	103
103	6.1.1 一维数组的定义	103
105	6.1.2 一维数组的应用	105
108	6.2 二维数组	108
108	6.2.1 二维数组的定义	108
110	6.2.2 二维数组的应用	110
112	6.3 数组作为函数参数	112
112	6.3.1 数组元素作为函数参数	112
112	6.3.2 数组名作为函数参数	112
114	6.4 字符数组	114
114	6.4.1 字符数组的定义	114
118	6.4.2 字符数组的应用	118

实训	119
技能训练	123
第7章 指针	127
7.1 指针的概念	127
7.1.1 变量存储的相关概念	127
7.1.2 变量的访问方式	128
7.2 指向变量的指针变量	129
7.2.1 指针变量的定义及初始化	129
7.2.2 指针变量的引用	130
7.2.3 指针变量作函数参数	134
7.3 指向一维数组的指针变量	137
7.3.1 一维数组指针的概念	137
7.3.2 一维数组的指针表示方法	138
7.3.3 一维数组元素的引用方法	138
7.3.4 通过指针引用数组元素	139
7.4 指向字符串的指针变量	141
实训	144
技能训练	147
第8章 结构体与共用体	152
8.1 概述	152
8.2 结构体类型及其变量的定义	153
8.2.1 结构体类型的定义	153
8.2.2 结构体变量的定义	153
8.2.3 结构体指针的定义	155
8.2.4 访问结构体成员的运算符	156
8.2.5 结构体变量的初始化	156
8.3 结构体数组	158
8.3.1 结构体数组的定义	159
8.3.2 结构体数组的初始化	159
8.4 共用体类型	162
8.4.1 共用体的概念、定义及其变量说明	162
8.4.2 共用体变量的赋值与应用	163
实训	166
技能测试	170
第9章 文件	173
9.1 文件概述	173
9.1.1 文件的概念	173

9.1.2	文件类型指针	174
9.2	文件的打开与关闭	174
9.2.1	文件打开函数 fopen	174
9.2.2	文件的关闭函数 fclose	176
9.3	文件的读写	176
9.3.1	写字符函数 fputc	176
9.3.2	读字符函数 fgetc	178
9.3.3	读字符串函数 fgets	178
9.3.4	写字符串函数 fputs	179
9.3.5	数据块读写函数 fread 和 fwrite	180
9.3.6	格式化读写函数 fscanf 和 fprintf	181
9.3.7	文件的随机读写	182
9.4	文件检测函数	184
9.4.1	文件结束检测函数 feof	184
9.4.2	读写文件出错检测函数 ferror	185
9.4.3	清除错误标志函数 clearerr	185
实训		186
技能训练		187
附录 I	常用字符与 ASCII 编码对照表	189
附录 II	C 语言中的关键字	190
附录 III	运算符和结合性	191
附录 IV	C 常用库函数	192
附录 V	Turbo C (V2.0) 使用指南	198
附录 VI	Turbo C (V2.0) 编译错误信息	203
参考文献		209

第 1 章 C 语言概述

能力目标

- 了解 C 语言的发展过程
- 初步了解 C 程序的大致框架构成
- 认识到 C 程序的基本构成单位是函数
- 了解 C 语言和 C 程序的特点
- 掌握 C 程序的开发过程

主要内容

本章首先介绍 C 语言出现的历史背景，让读者对 C 语言的发展过程有大致地了解，然后通过 C 程序实例，给读者展示 C 程序的一般样式和结构，以此来建立读者对 C 程序的整体印象，并由此顺理成章地总结出 C 程序的特点。最后给出 C 程序的开发过程和运行环境，在实训中，又以 VC++ 6.0 环境为实例，对 C 程序的开发提供了有力支持。

1.1 C 语言出现的历史背景

计算机产生后，它最初接受的是由 0 和 1 序列组成的指令码，这种指令码序列称为机器语言。用机器语言编写的程序，计算机能直接理解并执行，且执行效率高，但是由于机器语言不容易被人理解和记忆，带来了许多不便，所以不易推广。后来又产生了用助记符描述的指令系统，它相对机器语言要容易理解和记忆，这就是汇编语言。汇编语言与机器语言一样对机器的依赖很强，这也束缚了其发展和应用，能否创造一种既接近硬件又不依赖机器类型，同时使用灵活、功能强大的高级语言呢？C 语言就承担了历史重任，慢慢发展成长起来了。

C 语言是一种过程化的程序设计语言。它的前身是 Martin Richards 于 20 世纪 60 年代开发的 BCPL 语言，这是一种计算机软件人员在开发系统软件时作为记述语言使用的程序语言。1970 年美国贝尔实验室的 Ken Thompson 和 Dennis Ritchie 完成了 UNIX 的初版，与此同时，他们还改写了由 Martin Richards 开发的 BCPL 语言，形成了一种称为 B 的语言，此后，B 语言又进一步被改进和完善，形成了称之为 C 的语言，如图 1-1 所示。

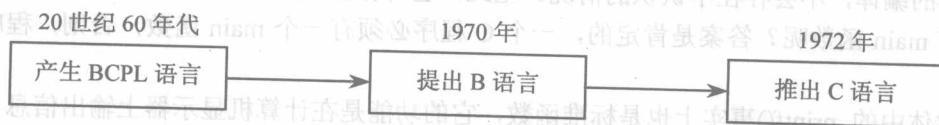


图 1-1

C 语言形成后，1973 年 Dennis Ritchie 把 UNIX 系统中的 90% 又用 C 语言进行了改写。随

着 UNIX 的移植和推广，C 语言也得到移植和推广。C 语言同时具备低级语言和高级语言的特征，所以有人说它是中级语言。由于 C 语言本身强大的功能，自面世以来倍受广大程序员的青睐，并流行至今。

提醒：以前的操作系统等系统软件主要是用汇编语言编写的。由于汇编语言依赖于计算机硬件，程序的可读性和可移植性都比较差。要想提高可读性和可移植性，最好采用高级语言。但一般的高级语言难以实现汇编语言的某些功能（汇编语言可以直接对硬件进行操作，如对内存地址的操作等）。因此，人们希望找到一种既具有高级语言特征，又具有低级语言特征的语言，于是 C 语言就随之产生了。

1.2 认识 C 程序

虽然我们还没有真正学习 C 程序设计，但是只要你留心下面的几个 C 程序，就会对 C 语言乃至 C 程序的特点有所了解，甚至能学会简单的屏幕显示程序，仔细品味下面这几个小程序。

【例 1-1】在屏幕上显示“Hello World!”的信息。

程序如下：

```
#include <stdio.h>
void main()          /*void 是函数的返回值类型*/
{ printf("Hello World!");
}
```

运行结果：

Hello World!

程序分析：如果在 C 语言的编译器下运行该程序，将会在计算机屏幕上显示“Hello World!”这样的信息。通过观察，我们发现 C 程序由：

```
void main()
```

```
{
    .....
}
```

这样的框架构成，该框架称为主函数或 main 函数。其中，void 是“空类型”的标识符，是 main 函数的返回值类型，此处是为说明主函数没有返回值的意思，具体意义和用法后面再阐述，对 main 函数来说，void 通常可以省略。main 为函数名，圆括号里一般有参数（main 函数一般没有参数），花括号内为函数体。函数体由 C 语句（程序指令）或 C 函数组成，关于 C 语句后面会逐步学习。main 函数是 C 语言本身函数库已定义好的标准函数，C 编译器能对它进行正确的编译，不会存在不认识的情况。至此，也许你会心生疑惑——是否所有的 C 程序都必须有 main 函数呢？答案是肯定的，一个 C 程序必须有一个 main 函数，否则，程序将无法运行。

函数体中的 printf() 事实上也是标准函数，它的功能是在计算机显示器上输出信息，类似的还有键盘输入函数 scanf()，读者可以先将这两个函数记牢，今后编程一般都要用到。printf 函数的具体内容包含在 C 语言的函数库头文件 stdio.h 中。C 语言的创造者为了方便用户，把一些常用的功能用函数的形式做好，用户在开发应用程序时，若用得上该功能函数，可通过包

含头文件的形式调用,这将大大提高开发效率。所有的标准功能函数都存在于相应的头文件中。C语言中,有关输入输出的标准函数都包含在头文件 `stdio.h` 中,使用这些功能函数时,一般要在程序开头加上 `#include <stdio.h>` 或 `#include "stdio.h"`,不过,应用 `printf()` 和 `scanf()` 函数时,可以省略 `#include <stdio.h>`。详细情况,读者可以参阅附录。

聪明的读者也许对例 1-1 程序的输出结果有所感悟,原来用 C 语言编写显示信息程序这么简单呀!不就是在 `printf` 函数的圆括号内将要输出的信息用双引号括起吗?!! 的确,就这么简单。如果让你现在编写一个在屏幕上显示 “I like C very much!” 的小程序,应该不难吧!

【例 1-2】已知三个数求其平均值。

程序如下:

```
float average(int x,int y,int z) /*求三个数平均值的自定义函数 average*/
{ float aver; /*存储平均值的实型变量 aver*/
  aver=(x+y+z)/3; /*求平均值,并将值存储到变量 aver 中*/
  return(aver); /*返回函数值,即平均值*/
}
#include "stdio.h" /*包含输出头文件,本程序中可省略*/
main() /*主函数*/
{ int a,b,c; /*定义整型变量 a、b、c */
  float ave; /*定义实型变量 ave,用来存储函数 average 的值*/
  a=3;b=4;c=5; /*变量赋初值*/
  ave=average(a,b,c); /*调用函数 average(),并将结果返回给变量 ave */
  printf("average=%f",ave); /*以实型格式%f 输出变量 ave 的值*/
}
```

运行结果:

average=4.000000

程序分析:

(1) 程序由函数组成,它们可以是程序员自定义函数,也可以是标准库函数(如 `printf()` 库函数),但程序的执行总是从 `main` 函数开始的。

(2) 计算机要处理的数据在编程时要将它们存储在变量中,变量相当于容器,没有“容器”,数据无法存储和处理,什么类型的数据要定义什么类型的变量。

例如:本程序中用到的变量 `a`、`b`、`c` 被定义为整型,用 `int` 来表示整型;变量 `ave` 被定义成实型,用 `float` 来表示实型,像这些“标识”是固定的,今后会反复用到,因此,不管初学者对此是否理解,都希望先将它们记牢,随着今后的学习自然就迎刃而解了。

(3) 程序中以分号结束的代码行称为语句,不过输入输出函数习惯上不称为语句。

(4) 求平均功能的自定义函数:

```
float average(int x,int y,int z)
{ .....
}
```

其中, `float` 是函数值的类型定义, `average` 是函数名,花括号 `{}` 内是函数体,这是定义函数的固定格式。大家暂时可以这样理解函数:函数是实现一定功能的程序单元,函数体是具体实现该功能的程序代码。今后,我们会对函数深入讲解,初学者可慢慢领悟。括号中的 `x`、`y`、`z` 是函数的形式参数,相当于数学中函数的参数,一旦给定了确定的参数,就能得到一个确定

的函数值。在该程序中，确定的参数是由 main 函数中实际参数 a、b、c 提供的。当程序从主函数自上而下执行到语句 `ave=average(a,b,c);` 处时，将调用自定义函数 `average`，同时，用实参 a、b、c 的值代入形参 x、y、z。

(5) 程序中的解释说明部分用 `/* */` 括起来，在程序代码段中用注释的目的是为了便于理解程序思路，注释信息在程序中不会执行，也不会影响程序的执行。千万注意 `/*` 和 `*/` 是成对出现的，若在注释时遗漏了 `*/`，会出现执行错误，思考一下为什么。

(6) 要想在计算机的屏幕上看到程序的运行结果，可在程序中调用标准输出函数 `printf()`，由于实数类型的输出输入格式是 `%f`，因此必须用 `%f` 控制 `ave` 实型变量的输出，`%f` 要求输出的数的小数点后有 6 位小数。

技巧：例 1-2 是对三个固定数（对 3、4、5 求平均值）求平均值，假如我们要求任意三个数的平均值，怎么办呢？要不要靠每次修改程序来实现呢？事实上，我们可以利用键盘输入函数 `scanf()` 来实现！只要修改一下 main 函数就可以了。

程序如下：

```
main()
{ int a,b,c;
  float ave;
  printf("input a,b,c:");
  scanf("%d%d%d",&a,&b,&c);    /*键盘输入函数*/
  ave=average(a,b,c);
  printf("average=%f",ave);
}
```

程序分析：其中，`scanf` 函数双引号内的 `"%d%d%d"` 是分别说明变量 a、b、c 按整型格式输入，`&` 表示地址符号，运用输入函数输入变量值时，必须在变量前加地址符号，这样才能保证输入的值正确存入相应的变量所在的内存单元。有兴趣的读者可以在计算机上运行一下，看效果如何。

1.3 C 语言程序的特点

通过对 C 程序的认识，总结出 C 程序的一些特点如下：

(1) C 程序是由函数构成的，一个 C 源程序至少包含一个 main 函数，也可以包含一个 main 函数和若干个其他函数。

(2) 一个 C 程序总是从 main 函数开始执行的，而不论 main 函数在程序中的位置。

(3) C 程序书写格式自由，一行内可写几个语句。

(4) C 程序中，每个语句和数据定义的最后必须有一个分号。

(5) C 语言本身没有输入输出语句，输入输出是由函数完成的。

(6) 可以用 `/*` 和 `*/` 对 C 程序中的任何部分作注释。

提醒：C 程序的基本单位是函数，一个源程序由若干函数组成，但至少包括一个 main 函数，且 main 函数的位置不限。

1.4 C 程序的开发过程

开发一个 C 程序，一般要经历编辑、编译、连接和运行四个步骤。假设待处理的 C 程序名为 f，则过程如图 1-2 所示。

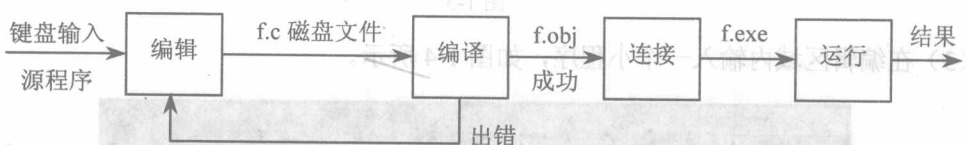


图 1-2

1. 源文件的编辑

用户通过编辑器，将自己开发的 C 语言程序输入计算机的过程称为 C 程序源文件的编辑。编辑生成的文件以文本形式存储，扩展名为.c，也称为 C 的源程序。源程序文件以 ASCII 码形式存储，计算机不能直接执行。

2. 编译

计算机把 C 的源程序翻译成计算机可以识别的二进制形式的目标代码文件，这个过程称为编译，由 C 的编译程序完成。

C 的编译器程序在编译的同时，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，将会列出错误的位置和种类，此时需要重新编辑修改源程序。如果编译成功则生成目标文件，文件名同源程序文件名，扩展名为.obj。

编译生成的目标文件不包含程序运行所需要的库函数等“资源”时，计算机仍然不能直接执行。

3. 连接

连接程序将目标程序和其他目标程序模块，以及系统提供的 C 库函数等进行连接生成可执行文件的过程，称为“连接”。连接生成的可执行文件的文件名同源程序文件名，扩展名为.exe。

连接生成的可执行文件，计算机可以直接执行。

4. 运行

在 DOS 环境下直接键入 C 程序的可执行文件名，或者在 C 的集成环境下选择 Run 命令，或者在 Windows 的资源管理器内双击该可执行文件名，都可以获得运行结果。如果运行结果有误，需要重新编辑源程序，再进行编译、连接、运行，直到得到满意的运行结果。

1.5 Turbo C 2.0 编辑环境应用实例

1. 程序的编辑方法

(1) 打开 File 菜单（按快捷键 Alt+F 或按 F10 功能键，然后通过移动光标键来选择 File 菜单）。

(2) 通过光标键选择 New 选项，如图 1-3 所示。

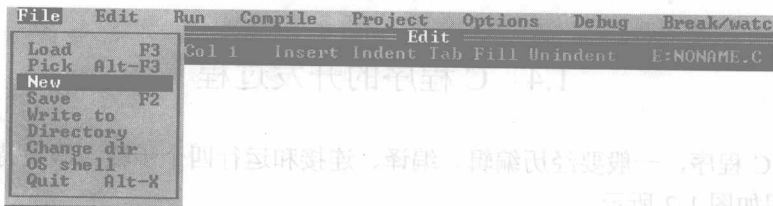


图 1-3

(3) 在编辑区域内输入一个小程序，如图 1-4 所示。

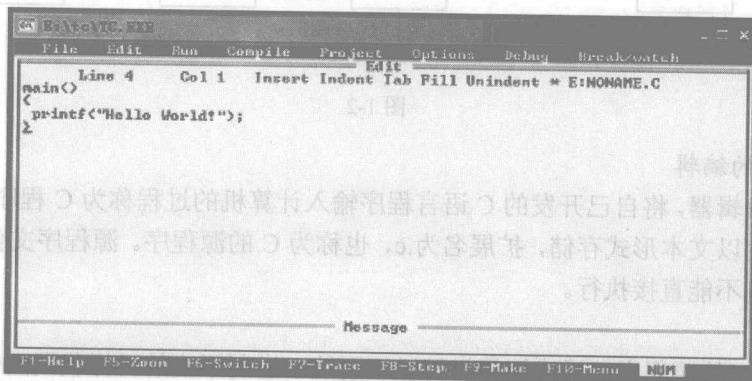


图 1-4

2. 保存程序

(1) 打开 File 菜单，选择 Write to 选项，弹出如图 1-5 所示 New Name 对话框。

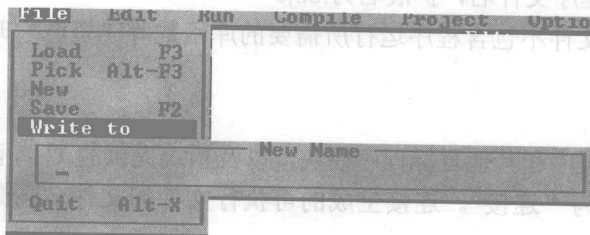


图 1-5

(2) 在图 1-5 所示的 New Name 对话框中，输入自己程序的保存路径及文件名，假如程序名取为 P1，将其保存在 D 盘的 MYFILE 目录下，如图 1-6 所示。回车后，即将所输入的程序保存到指定的路径下。

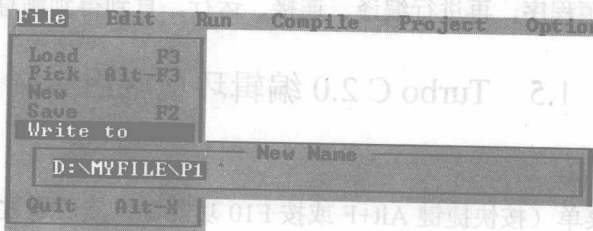


图 1-6

3. 运行程序

打开 Run 菜单, 选择 Run 选项, 或者直接用快捷键 Ctrl+F9 执行, 实现程序的编译连接过程。

若程序无错, 则会显示编译成功画面, 否则显示出错信息, 可排错后再执行。

4. 显示运行结果

程序正确编译后, 并不能出现运行结果, 若要查看运行结果, 可打开 Run 菜单, 选择 User screen 选项 (如图 1-7 所示), 或者直接用快捷键 Alt+F5。

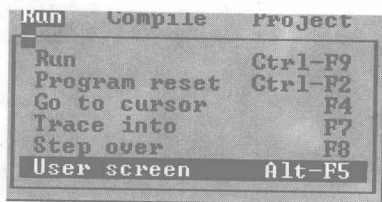


图 1-7

运行结果:

Hello World!

若要返回程序编辑状态, 按任意键即可。

提醒: ①对同一个程序来讲, 若修改后再保存, 则用 File 菜单的 Save 选项即可, 不要再写 Write to 选项; ②若想再编辑调试其他程序, 千万不要接着第一个程序往下输入, 可重新选择 File 菜单的 New 选项, 从而开始新程序的编辑。

实 训

一、实训目的

1. 掌握 VC++ 环境下 C 程序的编译方法。
2. 加深对 C 程序的理解。

二、实训内容

1. 内容: 运用 VC++ 6.0 编译器运行 “Hello World” C 程序。
2. 基本要求: 掌握 VC++ 6.0 的基本使用方法。

三、实训过程

1. 实训分析

首先保证电脑上装有 VC++ 6.0 软件, 然后单击 Windows 操作系统左下角的 “开始” → “所有程序” → Microsoft Visual Studio 6.0 → Microsoft Visual C++ 6.0 命令, 启动 VC++ 软件。

2. 实训步骤

(1) 启动 VC++ 后, 执行 “文件” → “新建” 命令, 弹出 “新建” 对话框, 如图 1-8 所示, 单击 “文件” 选项卡, 在对话框中选择 C++ Source File 选项, 在右边的 “目录” 栏中选

择文件存放的目录，本例存放在 d:\Cfile 目录中（目录 Cfile 是事先创建在 d 盘上的）。最后在“文件”栏中输入 C_1.c，其中，.c 表示文件是一个 C 语言源程序（注意，如果不输入文件扩展名，则 VC++ 默认为 .cpp 的 C++ 文件）。

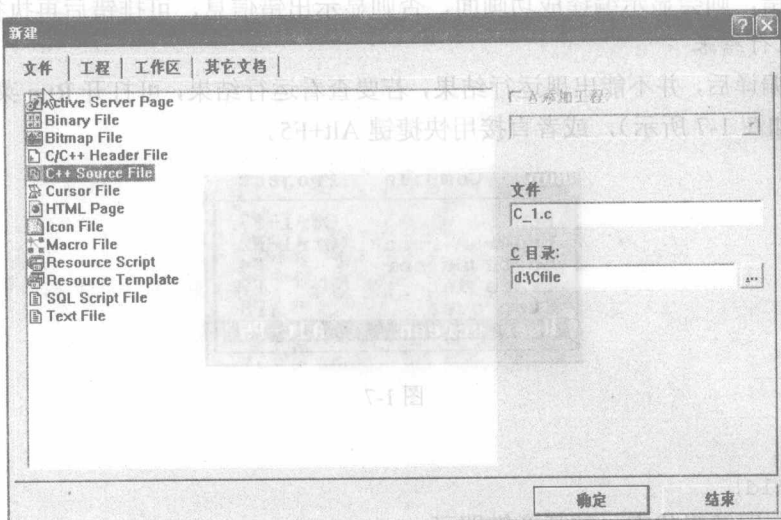


图 1-8

(2) 输入完毕后，单击“确定”按钮，进入代码编辑框，在编辑区域输入程序代码，如图 1-9 所示。

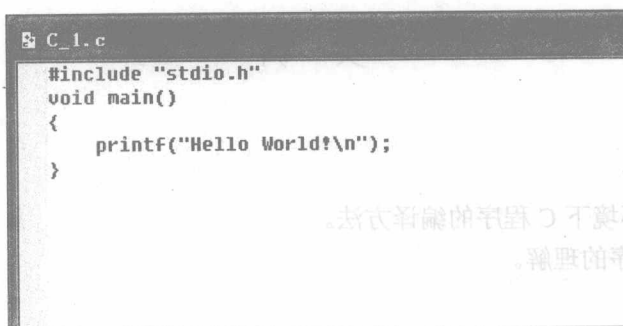


图 1-9

(3) 要执行程序，可单击菜单“编译”→“构建”命令，弹出如图 1-10 所示的消息框，询问是否要建立工作区，单击“是”按钮，弹出如图 1-11 的消息框，又询问是否保存该文件，也单击“是”按钮。

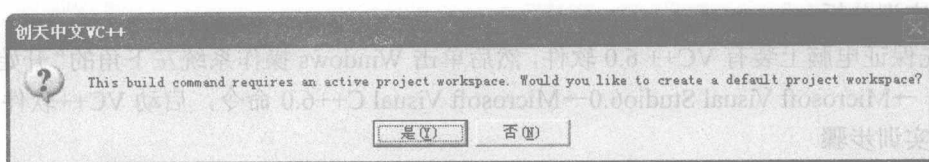


图 1-10