



工业和信息化普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

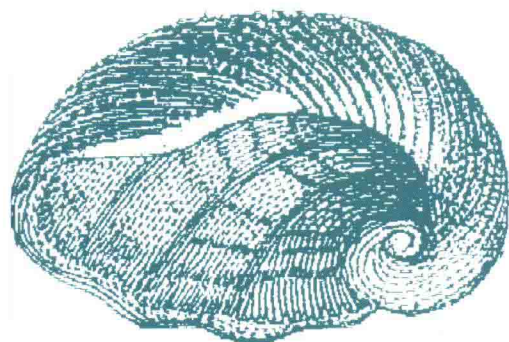
C语言程序设计 实验指导

Experimental Guidance Book For
C Programming Language

宋雅娟 徐志伟 李柯景 主编

郑山红 李万龙 张春飞 副主编

- 模块式编排，丰富知识全面覆盖，按需取用
- 启发式思维，常用算法深入剖析，透彻理解
- 专家式指导，典型案例精确分析，轻松掌握



高校系列



人民邮电出版社
POSTS & TELECOM PRESS



工业和信息化普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

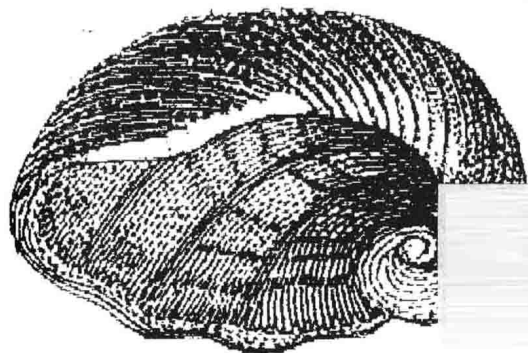
21st Century University Planned Textbooks of Computer Science

C语言程序设计 实验指导

Experimental Guidance Book For
C Programming Language

宋雅娟 徐志伟 李柯景 主编

郑山红 李万龙 张春飞 副主编



高校系列

人民邮电出版社

北京

图书在版编目 (C I P) 数据

C语言程序设计实验指导 / 宋雅娟, 徐志伟, 李柯景
主编. — 北京: 人民邮电出版社, 2012.10
21世纪高等学校计算机规划教材
ISBN 978-7-115-29276-6

I. ①C… II. ①宋… ②徐… ③李… III. ①
C语言—程序设计—高等学校—教学参考资料 IV.
①TP312

中国版本图书馆CIP数据核字(2012)第211064号

内 容 提 要

本书是《C语言程序设计(第2版)》(ISBN 978-7-115-27051-1, 人民邮电出版社出版)一书的配套用书, 其主要内容由22个实验组成, 实验内容的难度和实验量适中, 并给出了相应的指导。全书内容根据普通院校教学的实际编写, 实用性较强。

21世纪高等学校计算机规划教材

C语言程序设计实验指导

-
- ◆ 主 编 宋雅娟 徐志伟 李柯景
 - ◆ 副 主 编 郑山红 李万龙 张春飞
 - 责任编辑 武恩玉
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
 - 邮编 100061 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京艺辉印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
 - 印张: 8.25 2012年10月第1版
 - 字数: 213千字 2012年10月北京第1次印刷

ISBN 978-7-115-29276-6

定价: 20.00元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言

C 语言程序设计是一门专业基础课,也是一门实践性很强的技术课程。开设实验课程的目的在于培养学生的编程能力,促进理论与实践的结合。本书提供大量的编程实例,使学生了解和掌握程序设计的基本算法和思想,掌握基本的编程方法和技巧,与此同时,使学生在程序设计方面得到基本的训练。

C 语言对初学者来说有很大难度,本书提供大量的实例和多种多样的实验题型,并对程序设计题进行了非常详细的编程指导。

本书的使用对象包括普通高等院校的学生和想了解程序设计并进行程序设计实践的读者,实验环境是 Visual Studio 6.0,本书实验一详细介绍了在 Visual C++ 中进行编程及调试的详细过程,后续部分章节中也根据实验内容进行了详细的调试说明。

本书对程序设计的 3 种基本结构及数据的组织方式进行了较为详尽的知识介绍和对应的实验内容指导,重点放在了循环结构和数组上。在这些部分中,程序设计部分讲解详细,重点在于指导学生如何根据题目要求进行算法思想分析,进而进行实际的编程。各部分主要分为程序范例分析、程序填空、程序改错和程序设计模块。

为了使学生在掌握基本知识后能培养综合编程的能力,书中后面部分安排了综合实验内容,以一个常见的通讯录为例,从头搭建程序框架,进行菜单程序设计,之后进行各主要函数的内容设计,给出各函数要求,以函数程序填空和自主编写函数的形式进行,综合运用了程序的 3 种基本结构、函数、数组、结构体、文件等内容,涵盖了 C 语言程序设计中涉及的主要内容。

为了使学生进一步运用指针学习动态数据结构,本书最后一个实验安排了扩展内容——链表,在相关知识部分介绍了链表的作用、定义及创建链表、动态插入、删除等基本编程方法,与之前综合实验部分的通讯录程序对应,重新使用链表设计各部分程序,涵盖了链表应用的各环节,重点在于如何使用链表完成一个综合实验。

全书共有 22 个实验,使用本书时,可根据实验大纲选择书中的一个实验或多个实验作为实验课程的一节课内容,最后的链表部分可作为选修或由学生自学选看。

本书的实验一至实验七、实验十八、实验十九由李柯景执笔,实验八至实验十四由徐志伟执笔,实验十五至实验十七、实验二十至实验二十二由宋雅娟执笔,综合实验内容及链表部分的实验程序完全由读者自行设计并在实际环境中验证。

由于编者水平有限,书中错误和不妥之处在所难免,恳请读者批评指正。

编者

2012 年 6 月于长春

目 录

实验一 运行一个简单的 C 程序.....1	实验十三 函数与变量的作用域.....76
实验二 基本输出7	实验十四 编译预处理81
实验三 运算符和表达式的使用13	实验十五 指针变量85
实验四 基本输入17	实验十六 指针与数组89
实验五 顺序结构21	实验十七 指针与函数93
实验六 选择结构24	实验十八 结构体96
实验七 开关语句30	实验十九 共用体与枚举102
实验八 单重循环34	实验二十 文件操作107
实验九 多重循环42	实验二十一 综合实验113
实验十 数组49	实验二十二 链表120
实验十一 数组的应用57	参考文献126
实验十二 函数的定义与调用64	

实验一

运行一个简单的 C 程序

一、知识介绍

在 VC++环境中编写 C 语言程序的步骤。

【注意】在开始编程之前，最好先创建一个文件夹，文件夹的位置可以自己选择，该文件夹用来存放以后创建的 C 文件，以方便个人的查找。这里创建的文件夹路径为 D:\my_c 。

1. 启动 VC++

选择“开始”——“程序”——“Microsoft Visual Studio 6.0”——“Microsoft Visual C++ 6.0”，可启动 VC++，屏幕上将显示如图 1-1 所示的窗口。

2. 新建/打开 C 程序文件

在 VC++环境中可以先建立工程文件，再新建 C 语言的源文件，也可以直接新建一个源文件，此处介绍第 2 种方法。选择“文件”菜单的“新建”菜单项，在对话框的“文件”标签下（见图 1-2）选中“C++ Source File”项，在“文件”框中输入文件名，例如“program_1.c”，在“目录”框中选择输入先前创建的用来存放 C 文件的文件夹“D:\my_c”，单击“确定”按钮，即可打开程序的编辑窗口。

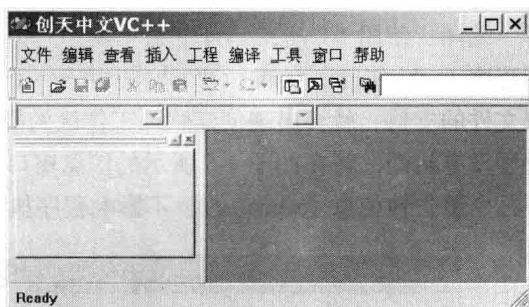


图 1-1 VC++窗口

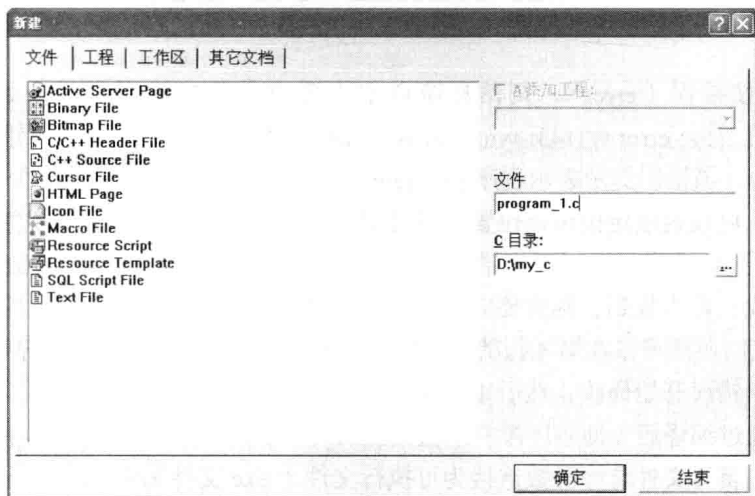


图 1-2 新建文件

如果程序已经输入并保存在了磁盘上,可选择“文件”菜单的“打开”菜单项,并在查找范围中找到正确的文件夹,调入指定的程序文件。

3. 编辑并保存源程序

在编辑窗口中输入源程序代码,如图 1-3 所示,然后通过“文件”菜单保存源文件。

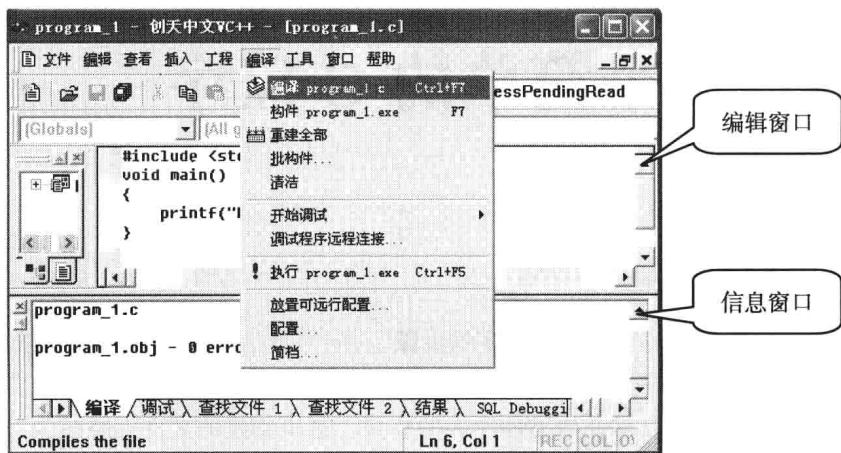


图 1-3 编译 C 程序

4. 编译、连接、执行程序

编译:使用 VC++ “编译”菜单中的“编译”菜单项,将 C 语言源程序(.c 文件)编译为目标程序(.obj 文件),如图 1-3 所示。由于 VC 环境中的程序需要在工作区中运行,若此时没有工程文件的支持,就会出现创建默认工作区的提示,如图 1-4 所示,单击“是”按钮即可。如果源程序没有错误,将在如图 1-5 所示的信息窗口中显示这样的内容:0 error(s) 0 warning(s)。有时出现几个警告性信息(warning),不影响程序执行。

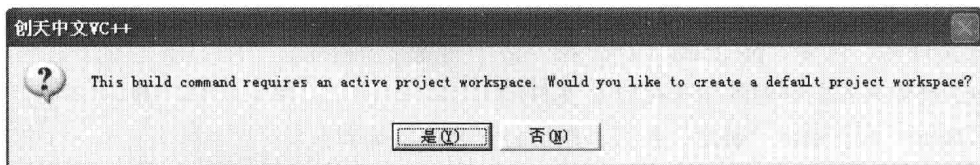


图 1-4 创建默认工作区提示

假如有致命性错误(error),则信息窗口中出现致命语法错误提示,如图 1-5 所示:
d:\my_c\program_1.c(5): error C2143: syntax error : missing ';' before "{" (该出错信息中括号里的 5 表示出错的行号,该错误提示表示程序 program_1.c 的第 5 行 "{" 之前缺少分号";")。根据出错提示的行号,可以直接定位出错位置,只需双击信息窗口中的某行出错信息,程序窗口中会用蓝色图标“■”自动指示出对应出错位置,此时可根据信息窗口的提示修改程序。找错时可先在指示的行上找,若未找到,则需要向上一行继续找错。例如上面的错误,虽然信息窗口提示第 5 行出错,但我们还是习惯在第 4 行的末尾加上分号。有时一个语法错误会导致多个错误提示,建议大家从第一个错误开始修改,然后重新编译,直到没有错误提示为止。

连接:程序通过编译后(即程序没有语法错误),使用 VC++ “编译”菜单中的“构件”菜单项,将编译生成的目标文件与库函数连接为可执行文件(.exe 文件)。

执行:程序通过连接后(即执行连接后错误为 0 个),使用 VC++ “编译”菜单中的“执行”

菜单项，将执行程序。如果程序运行时需要输入或输出数据，VC++将自动弹出如图 1-6 所示的数据输入输出窗口。按任意键将关闭该窗口。

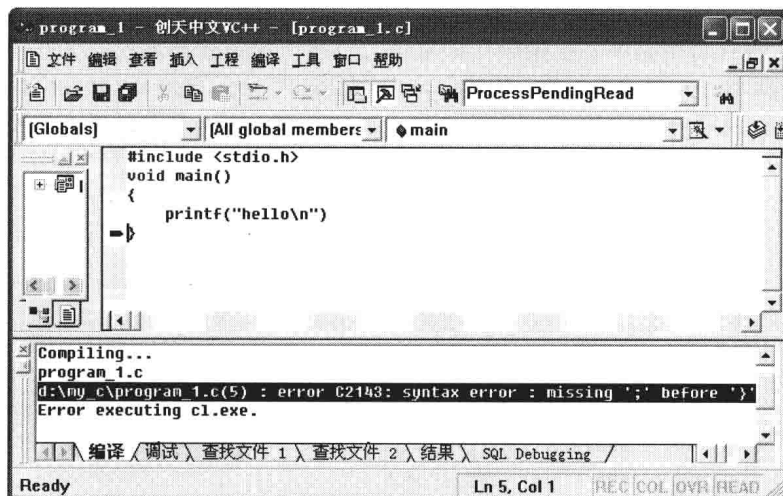


图 1-5 程序中有语法错误的情况

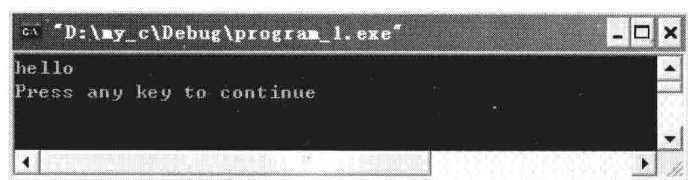


图 1-6 数据输入输出窗口

5. 关闭程序工作区

在创建一个 C 工程时，VC++系统自动产生相应的工作区，以完成程序的运行和调试。若想执行第 2 个 C 程序，必须关闭前一个程序的工作区，然后新建一个 C 工程，产生第 2 个程序的工作区。否则运行的将一直是前一个程序。

“文件”菜单提供关闭程序工作区功能，如图 1-7 所示。



图 1-7 关闭程序工作区

6. 程序调试

(1) 使程序执行到中途暂停，以便观察阶段性结果。

方法一：使程序执行到光标所在的那一行暂停。

① 在需暂停的行上单击鼠标，定位光标。

② 如图 1-8 所示，分别单击菜单“编译”——“开始调试”——“Run to Cursor”，或按“Ctrl+F10”组合键，程序将执行到光标所在行暂停。如果把光标移动到后面的某个位置，再按“Ctrl+F10”组合键，程序将从当前的暂停点继续执行到新的光标位置第 2 次暂停。



图 1-8 执行到光标所在行暂停

方法二：在需暂停的行上设置断点。

① 在需设置断点的行上单击鼠标，定位光标。

② 单击图 1-9 所示的“编译微型条”中最右面的按钮或按 F9 键，被设置了断点的行前面会有一个红色圆点标志。该操作是一个开关，单击是设置，双击是取消设置。如果有多个断点想全部取消，可执行“编辑”菜单中的“断点”菜单项，屏幕上会显示“Breakpoints”窗口，窗口下方列出了所有断点，单击“Remove All”按钮，将取消所有断点。

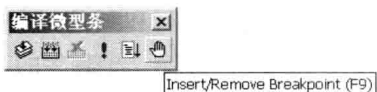


图 1-9 设置断点

断点通常用于调试较长的程序，在使用“Run to Cursor”(运行程序到光标处暂停)或“Ctrl+F10”功能时，经常要把光标定位到不同的地方。而对于长度为上百行的程序，要寻找某位置并不太方便。

如果一个程序设置了多个断点，按一次组合键“Ctrl+F5”会暂停在第 1 个断点，再按一次“Ctrl+F5”会继续执行到第 2 个断点暂停，依次执行下去。

需要提醒的是：不管是通过光标位置还是断点设置，其所在的程序行必须是程序执行的必经之路，亦即不应该是分支结构中的语句，因为该语句在程序执行中受到条件判断的限制，有可能因条件的不满足而不被执行。这时程序将一直执行到结束或下一个断点为止。

(2) 设置需观察的结果变量。

采用上述两种方法，使程序执行到指定位置时暂停，目的是查看有关的中间结果。在图 1-10 左下角窗格中“Locals”标签下，系统自动显示了当前函数内局部变量的值，其中 data1 和 data2 的值分别是 3 和 5，而变量 result 的值是不正确的，因为此时程序还未执行赋值语句，result 还未被赋值。图 1-10 中程序左侧的箭头指示当前程序暂停的位置。如果还想观察其他变量的值，可在图

1-10 中右下角的 Watch 窗口中的“Name”框中填入相应变量名。

(3) 单步执行。

单步执行单击“调试”中“Step Over”按钮或按 F10 键,如图 1-11 所示。如果遇到自定义函数调用,想进入函数内部进行单步执行,可单击“Step Into”按钮或按 F11 键。当想结束函数的单步执行,可单击“Step Out”按钮或按“Shift+F11”组合键。对不包括函数调用的语句来说,F11 键与 F10 键作用相同。

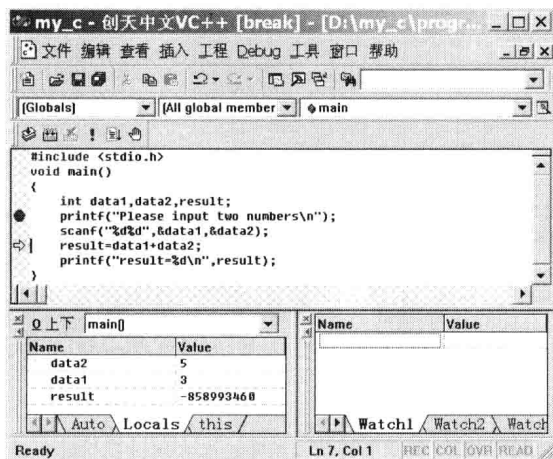


图 1-10 观察结果变量

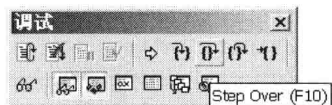


图 1-11 单步调试

(4) 停止调试。

使用“Debug”菜单的“Stop Debugging”菜单项,或按“Shift+F5”组合键可以结束调试,从而回到正常的运行状态。

二、实验目的

1. 通过运行简单的 C 程序,了解 C 语言程序的结构。
2. 掌握运行一个 C 语言程序需要经过的步骤。

三、实验内容

在 VC++ 环境中运行以下程序,并查看结果。

(1) 输入下面的程序,注意区分大小写。

```
#include<stdio.h>
void main()
{
    printf("This is a C program.\n");
}
```

编译并运行程序。

(2) 关闭工作区,新建一个程序,然后输入并运行一个需要在运行时输入数据的程序。

运行程序后使用 Visual C++ 的程序调试功能,单步执行程序,并通过 Watch 窗口观察变量 a, b, c, x, y 值的变化。

```
#include<stdio.h>
void main()
{
```

```
int a,b,c;
int max(int x,int y);
printf("input a and b:");
scanf("%d,%d",&a,&b);
c=max(a,b);
printf("\nmax=%d",c);
}
int max(int x,int y)
{
    int z;
    if(x>y)
        z=x;
    else
        z=y;
    return(z);
}
```

(3) 输入下面源程序, 并查看结果。

```
#include "stdio.h"
main()
{
    int i,j,result;
    printf("\n");
    for (i=1;i<10;i++)
    {
        for(j=1;j<i+1;j++)
        {
            result=i*j;
            printf("%d*%d=%-3d",i,j,result);    /*-3d 表示左对齐, 占 3 位*/
        }
        printf("\n\n"); /*每一行后换行*/
    }
}
```

四、思考题

1. 利用 VC++6.0 运行一个 C 程序, 共分为几个步骤? 各有什么功能?
2. 使用程序调试发现的错误与编译时发现的错误有什么区别?

实验二

基本输出

一、知识介绍

1. C 语言的数据类型

C 语言的数据类型分为：基本类型、构造类型、指针类型和空类型。
基本类型的分类及特点见表 2-1。

表 2-1 基本类型的分类及特点

	类型说明符	字 节	数 值 范 围
字符型	char	1	0~255
基本整型	int	2	-32768~32767
短整型	short int	2	-32768~32767
长整型	long int	4	-214783648~214783647
无符号型	unsigned	2	0~65535
无符号长整型	unsigned long	4	0~4294967295
单精度实型	float	4	-3.402823E38~3.402823E38
双精度实型	double	8	-1.79769313486232E308~1.79769313486232E308

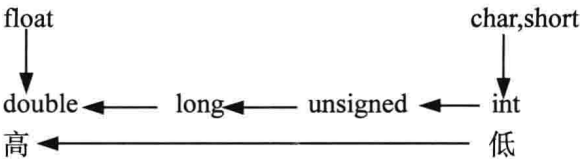


Visual C++ 6.0 中，基本整型与长整型所占字节数及表示范围对应相同。

数据类型转换

(1) 自动转换：在不同类型数据的混合运算中，由系统自动实现转换，由少字节类型向多字节类型转换。不同类型的量相互赋值时，也由系统自动进行转换，把赋值号右边的类型转换为左边的类型。

类型自动转换规则如下：



向左箭头表示当运算对象为不同类型时转换的方向，向下箭头表示必定的转换。

(2) 强制转换: 由强制转换运算符完成转换。

一般形式: (类型说明符)(表达式)

功能：把表达式的运算结果强制转换成类型说明符所表示的类型。

例如: (float) a 把 a 转换为实型

(int) (x+y) 把 x+y 的结果转换为整型

2. 数据格式输出——printf()函数

格式化输出函数 printf()功能：按控制字符串规定的格式，向输出设备输出各输出项。

其基本格式为：`printf("控制字符串", 输出项列表);`

控制字符串由格式说明和普通字符两部分组成。

普通字符原样显示。

格式说明

一般格式为：%[修饰符]格式字符

格式字符规定了输出项的输出格式，格式字符的含义见表 2-2。

表 2-2

格式字符的含义

格 式 字 符	含 义
d	以十进制形式输出带符号整数（正数不输出符号）
o	以八进制形式输出无符号整数（不输出前缀 0）
x, X	以十六进制形式输出无符号整数（不输出前缀 0x）
u	以十进制形式输出无符号整数
f	以小数形式输出单、双精度实数
e, E	以指数形式输出单、双精度实数
g, G	以%f 或%e 中输出宽度较短的输出单、双精度实数
c	输出单个字符
s	输出字符串

修饰符规定了输出项的对齐、宽度、小数位数等格式，修饰符含义见表 2-3。

表 2-3

修饰符的含义

修 饰 符	含 义
l	表示输出的是 long 型数据，可加在 d、f、u 前面
h	表示输出的是 short 型数据，可加在 d、f、u 前面
m	表示输出数据的最小宽度
.n	对实数，表示输出 n 位小数；对字符串，表示截取 n 个字符
0	表示左边补 0
+	转换后的数据右对齐
-	转换后的数据左对齐

3. 字符数据非格式输出函数——putchar()函数

putchar()函数的格式: putchar(ch);

该函数的功能是向显示器终端输出一个字符。其中，ch 可以是一个字符变量或常量，也可以

是一个转义字符。
常用的转义字符及其含义见表 2-4。

表 2-4 常用转义字符及其含义

转 义 字 符	含 义
\n	回车换行
\t	横向跳到下一制表位置
\v	竖向跳格
\b	退格
\r	回车
\f	走纸换页
\\	反斜线符 “\”
\'	单引号符
\a	鸣铃
\ddd	一至三位八进制数的值所代表的 ASCII 字符
\xhh	一至二位十六进制数的值所代表的 ASCII 字符

二、实验目的

- 1. 掌握 C 语言中的基本数据类型。
- 2. 掌握格式化输出函数 printf()的功能及方法。
- 3. 掌握字符数据非格式输出函数 putchar()的功能及用法。
- 4. 掌握常用的转义字符的含义及用法。

三、实验内容

1. 程序范例。

(1) 范例 1。

```
#include<stdio.h>
void main()
{
    printf("您好! \n");
    printf("欢迎您使用 VC++6.0。 \n");
    printf("Welcome! \n");
}
```

程序分析：

① printf()是函数语句。在 C 语言中使用库函数时，要使用预编译命令#include，将相关的头文件(*.h)包含到用户源文件中。输入输出函数的使用形式为：

```
#include "头文件"或#include <头文件>
```

- ② 该程序共由 3 条由 printf()构成的函数语句所组成。
- ③ 该程序的功能为向屏幕输出 3 行内容，每输出一行用转义字符 “\n” 实现换行。
- ④ 该程序运行结果为：
您好!

欢迎您使用 VC++6.0。

Welcome!

(2) 范例 2。

```
#include<stdio.h>
void main()
{
    int c=100;        //定义整型变量 c, 并赋初值 100
    char i;           //定义字符型变量 i
    putchar('a');
    putchar('\n');
    putchar('\100');
    i=putchar(c);
    putchar(i);
}
```

程序分析:

① 整型变量和字符型变量之间可以相互转换。如 `putchar(c);`, 由于整型变量 `c` 的初值为 100, 所以 `putchar(100);` 输出 ASCII 码为 100 的字符。字母 `d` 的 ASCII 码为 100。

② `putchar()` 函数参数可以为普通常量、转义字符, 也可以是变量。其中 `'a'` 为普通常量, `putchar('a');` 代表输出字符 `a`。 `'\n'` 和 `'\100'` 为转义字符, `'\n'` 表示换行, `'\100'` 表示将八进制数 100 转换成二进制数 1000000 所对应的字符 `@`。

③ 该程序运行结果为:

```
a
@dd
```

(3) 范例 3。

```
#include<stdio.h>
void main()
{
    float a=123.456;        //定义单精度类型变量 a, 初值为 123.456
    double b=8765.4567;     //定义双精度类型变量 b, 初值为 8765.4567
    printf("第 1 行 a=%f\n",a);
    printf("第 2 行 a=%14.3f\n",a);
    printf("第 3 行 a=%6.4f\n",a);
    printf("第 4 行 b=%lf\n",b);
    printf("第 5 行 b=%14.3lf\n",b);
    printf("第 6 行 b=%8.4lf\n",b);
    printf("第 7 行 b=%.4f\n",b);
}
```

程序分析:

① 浮点数输出时, 默认靠右对齐, `%m.nf` 中 `m` 表示显示数据的总宽度, 当宽度不足时, 按原来数据宽度进行显示, `n` 代表小数点后保留 `n` 位, 不足时后面补 0, 超过时遵循四舍五入的原则。

② `float` 类型数据应以 `%f` 类型输出, 小数点后默认保留 6 位。 `double` 类型应以 `%lf` 类型输出。

③ 程序中 `printf()` 函数双引号内的普通字符要原样显示。

(4) 范例 4。

```
#include <stdio.h>
void main()
{
```

```

int x=010,y=10,z=0x10;
char c1='M',c2='\x4d',c3='\115',c4=77,c;
printf("x=%o,y=%d,z=%x\n",x,y,z);
printf("x=%d,y=%d,z=%d\n",x,y,z);
printf("c1=%c,c2=%x,c3=%o,c4=%d\n",c1,c2,c3,c4);
printf("c1=%c,c2=%c,c3=%c,c4=%c\n",c1,c2,c3,c4);
printf("c1=%d,c2=%d,c3=%d,c4=%d\n",c1,c2,c3,c4);
c=c1+32;
printf("c=%c,c=%d\n",c,c);
}

```

程序分析:

① 以 0 开头的常量为八进制数,以 0x 开头的常量为十六进制数。故以 %o 和 %d 分别显示 x 的值以及以 %x 和 %d 分别输出 z 的值的时候显示的结果不一样。

② $c=c1+32$ 代表将 c1 的值加上 32 再赋给 c,所以 c 的值就为字母 M 的 ASCII 码值 $77+32=109$ 。因此分别以 %c 和 %d 输出 c 的值时,显示结果分别为 $c=m, c=109$ 。

2. 读程序写运行结果,并上机验证。

(1) 程序 1。

```

#include<stdio.h>
void main()
{
char c1='a',c2='b',c3='c',c4='\101',c5='\116'; /* 定义字符型变量 c1, c2, c3, c4, c5*/
printf("a%cb%c\tc%c\tabc\n",c1,c2,c3);
printf("\t\t\b%c%c\n",c4,c5);
printf("这是加号\"+"\\n");
printf("这是减号\"-\"\\n");
printf("这是乘号\"*\"\\n");
printf("这是除号\"/\"\\n");
printf("这是求余符号\"%%\"\\n");
}

```

运行结果解析:

① \t, \b, \n, \, \"均为转义字符。

② “%%”表示输出%。

(2) 程序 2。

```

#include<stdio.h>
void main()
{
int a,b;
unsigned c,d;
a=1;
b=-1;
c=a;
d=b;
printf("%d,%d\n",a,b);
printf("%u,%u\n",a,b);
printf("%u,%u\n",c,d);
}

```

运行结果解析: 由于负数在机器中按照补码形式进行存放,故 b 的值当以 %u 格式进行输出时,结果显示为 4294967295。

(3) 程序 3。

```
#include<stdio.h>
void main()
{
    char c1,c2;
    c1='a';
    c2='b';
    printf("%c %c",c1,c2);
}
```

思考题:

- ① 将该程序的最后一条语句改为 `printf("%d, %d\n", c1, c2);`, 程序的结果是什么?
- ② 将该程序中 `char c1, c2;` 改为 `int c1, c2;`, 然后再运行程序, 观察结果是否相同。

3. 程序填空。

(1) 以每行两个值的形式输出对应的变量值, 两个值之间用逗号分隔。

```
#include<stdio.h>
void main()
{
    int a=10,b=20;
    char c='A',d='a';
    float e=15.6,f=3.14;
    printf(_____,a,b);
    printf(_____,c,d);
    printf("%f,%f\n",_____,_____);
}
```

(2) 打印数值对应的字符。

```
#include<stdio.h>
void main()
{
    int a=1,_____,c=3,_____,e=5;
    printf("这是一个笑脸%c\n",_____);
    printf("这又是一个笑脸%c\n",b);
    printf(_____,c);
    printf("这是方片%c\n",d);
    printf("这是草花%c\n",_____);
}
```

四、思考题

1. 如何在屏幕上每行分别输出 0~9 的数字字符对应的 ASCII 值?
2. 如何实现在屏幕上输出以下内容:

```
*****
*                  *
*          小小电话簿          *
*                  *
*****
```