

▶ 普通高等教育“十二五”规划教材

C语言实训教程

■ 邓树文 周洁 闻彬 主编



COMPUTER
TECHNOLOGY



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

普通高等教育“十二五”规划教材

C 语言实训教程

主 编 邓树文 周 洁 闻 彬

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是与《C 语言程序设计》(ISBN 978-7-121-27419-0) 配套使用的教学参考书。书中介绍了 Visual C++6.0 的开发环境, 详细地设计了覆盖《C 语言程序设计》所有知识点的基础实验, 按知识模块组织了 16 个实验模块, 4 个实训项目, 最后本书列出了全国计算机等级考试二级 C 语言程序设计考试大纲和 3 套全国计算机等级考试二级 C 语言程序设计试题及参考答案。

本书中所有的实验代码在 Visual C++6.0 中测试通过。本书是学习 C 语言和上机实验的参考书, 也可以作为备考全国计算机等级考试二级 C 语言程序设计的参考书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

C 语言实训教程 / 邓树文, 周洁, 闻彬主编. —北京: 电子工业出版社, 2016.2
普通高等教育“十二五”规划教材

ISBN 978-7-121-27418-3

I. ①C… II. ①邓… ②周… ③闻… 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 (2015) 第 246562 号

策划编辑: 袁 玺

责任编辑: 郝黎明

印 刷: 保定市中华美凯印刷有限公司

装 订: 保定市中华美凯印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 16.25 字数: 468 千字

版 次: 2016 年 2 月第 1 版

印 次: 2016 年 2 月第 1 次印刷

定 价: 38.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言

C 语言程序设计是一门实践性很强的课程，需要进行大量的实践。在实践中遇到问题，分析问题，解决问题，才能更好地理解 C 语言，并最终学会利用 C 语言解决实际问题。本书从三个方面组织内容，帮助读者提高 C 语言的理解和实践能力。

首先，书中安排了大量的基础实验模块，基本覆盖了教材中 C 语言所有的知识点，通过实验引导，加强读者对 C 语言知识的理解和运用。

其次，编者安排了 4 个综合实训项目，其目的是为了让学生在现实世界中，思考如何利用计算机程序来解决问题，进一步提高读者的问题分析和解决能力。

最后，书中安排了大量的章节训练题，以及全国计算机等级考试真题，进一步巩固知识点。本书是《C 语言程序设计》配套教材，内容包括以下四大部分。

第 1 部分：介绍 Visual C++ 开发环境和基本的使用方法，重点讲解 C 程序的编辑、编译、调试的基本步骤和方法。

第 2 部分：C 语言上机实验，包括 16 个实验模块，每个实验模块又进一步细分为基础训练、进阶训练和思考训练三个层次。基础训练是帮助读者解决对应的知识点；进阶训练是进一步拓展知识点；思考训练是帮助读者了解所学知识点能处理的实际问题，进一步提高分析问题和解决问题能力。而且，每部分都准备了大量习题，帮助读者复习和训练章节的知识模块。

第 3 部分：本书精心准备了 4 个利用 C 语言为解决实际问题而编写的实训项目。读者可以通过动手实践，上机完成实训项目，进一步发掘计算机编程与实际应用的紧密关系。

第 4 部分：全国计算机等级考试二级 C 语言程序设计考试大纲和 3 套全国计算机等级考试二级 C 语言程序设计试题及参考答案。学习完本书之后，读者可以利用 C 语言程序设计试题考察自己 C 语言学习的程度。

由于时间紧迫，以及作者水平有限，书中难免有错误或者不妥之处，恳请读者批评指正。

编者

目 录

第 1 部分 Visual Studio 6.0 开发环境介绍

1.1 安装与启动	1
1.2 环境介绍	1
1.3 编译、链接和运行程序	3
1.4 调试程序	4

第 2 部分 模块训练

模块 1 开始编写程序	9
模块 2 简单 C 语法	13
模块 3 顺序结构	22
模块 4 选择结构 (1)	28
模块 5 选择结构 (2)	34
模块 6 循环结构 (1)	50
模块 7 循环结构 (2)	56
模块 8 数组 (1)	73
模块 9 数组 (2)	81
模块 10 函数 (1)	87
模块 11 函数 (2)	98
模块 12 指针 (1)	104
模块 13 指针 (2)	113
模块 14 结构体与共用体	125
模块 15 链表	144

模块 16 文件	162
----------------	-----

第 3 部分 实训项目

实训项目 1 学生成绩管理程序	180
实训项目 2 简易 ATM 存取款系统	186
实训项目 3 家庭财务管理	196
实训项目 4 实用小型通讯录管理系统	213
附录 A 全国计算机等级考试二级 C 语言程序设计考试大纲	219
附录 B 全国计算机等级考试二级 C 语言程序设计试题及参考答案	222
参考文献	253

第 1 部分 Visual Studio 6.0 开发环境介绍

本书以 Visual C++ 6.0 作为 C 源程序的实践开发环境，本章将首先介绍 Visual C++ 6.0 环境的基本操作，包括 Visual C++ 6.0 的安装和启动，C 源程序的编辑、运行与调试。

1.1 安装与启动

如果计算机上未安装 Visual C++ 6.0，则按照安装向导直接安装，具体步骤此处不再详述。安装过程中建议同时安装 MSDN，以便日后自学。

成功安装 Visual C++ 6.0 后，可以在桌面上看到如图 1.1 所示的图标或通过“开始”→“程序”→“Microsoft Visual Studio”找到“Visual C++ 6.0”启动程序。

双击 Visual C++ 6.0 图标或者单击“开始”→“程序”→“Microsoft Visual Studio”→“Visual C++ 6.0”即可启动 Visual C++ 6.0 集成开发环境，正常启动开发环境后，可见如图 1.2 所示的主窗口。



图 1.1 Visual C++ 6.0 图标

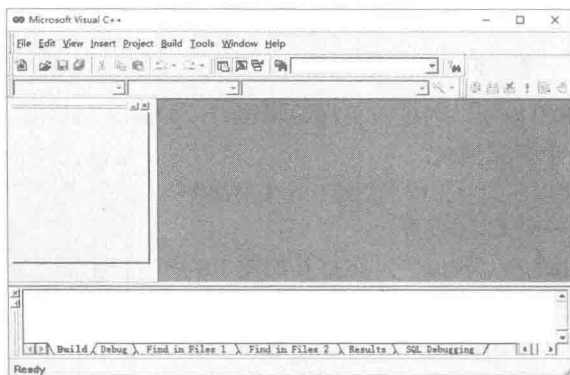


图 1.2 Visual C++ 6.0 主窗口

1.2 环境介绍

编译一个 C 源程序的前提是新建一个程序或者打开现有程序。本节首先介绍如何新建一个程序，如何打开现有程序，并在此基础上对文件进行编译。

1. 新建一个工程

在 Visual C++ 6.0 主窗口中的主菜单栏中单击“File(文件)”菜单项，然后单击“New(新建)”菜单项，也可按 Ctrl+N 组合键进入“NEW(新建)”对话框，如图 1.3 所示。

出现如图 1.4 所示的一个“New(新建)”对话框，单击此对话框上方的“Projects(工程)”标签，在其下方菜单栏选择“Win32 Console Application”项，输入工程名，设定存储路径(Location)然后单击“OK”按钮。

单击“OK”按钮后，会弹出如图 1.5 所示的对话框，选中“An empty project”单选按钮，单击“Finish”按钮。

之后弹出如图 1.6 所示的对话框。注意：该对话框上显示了工程文件的。

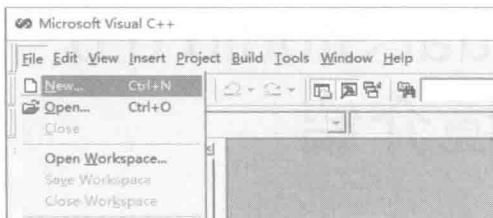


图 1.3 新建工程的操作过程

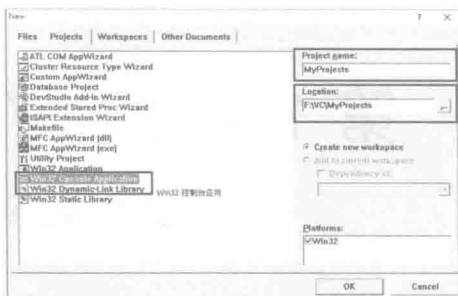


图 1.4 “New”对话框

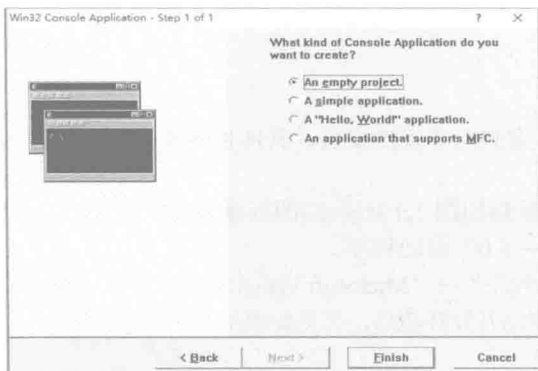


图 1.5 工程类型选择



图 1.6 新工程信息

单击“OK”按钮，则进入工程编辑窗口，如图 1.7 所示。在工程编辑窗口可以看到 MyProjects 工程下只有三个空文件夹。

此后，就可以在当前工程下创建一个 C 源程序。

2. 新建一个 C 源程序

单击“File”→“New”，在打开如图 1.8 所示的新建对话框中，类似新建工程操作，在“Files”标签选择“C++ Source File”项；然后选中“Add to Project”复选框，并选择添加到新建的工程“MyProjects”；在“File”输入框中，输入后缀名是.c 的文件名，单击“OK”按钮。


注意，如果在文件名中不显示扩展名.c，则 VC++ 将为文件附上默认扩展名.cpp，并按照 C++ 语言的语法进行检查。由于 C++ 的语法检查要比 C 语言的语法更为严格，因此，建议还是输入文件扩展名.c。

单击“OK”按钮后，可以进入 MyFile.c 的编辑窗口，如图 1.9 所示在窗口的标题栏和工程窗口中都显示出了当前要编辑的文件名字“MyFile.c”。

此时，程序编辑窗口被激活，可以输入和编辑源程序了。这里编写了一个简单的 C 程序，输出“Hello World!”，如图 1.10 所示。

3. 打开一个现存的程序

与新建的操作类似，可以通过“File”→“Open Workspace”项进入工程窗口，如图 1.11 所示。双击工程文件（扩展名为.dsw），即可打开已有的工程。此外，也可以通过双击工程文件直接打开现有程序。

进入编辑状态后，如果对程序进行了修改且未保存则在标题栏中文件名字后面会出现“*”提示，选择“File”→“Save”、单击  按钮或者使用 Ctrl+S 组合键都可以保存文件。保存之后，标

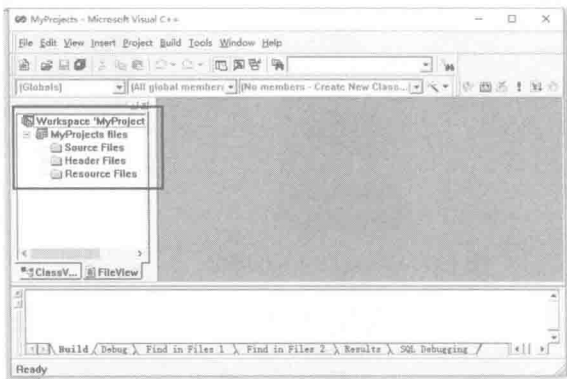


图 1.7 MyProjects 工程编辑窗口

题栏的“*”消失。

如果不想将源程序保存到指定的文件中，可以选择“File”→“Save As”项，重新指定文件保存的位置以及文件名字。

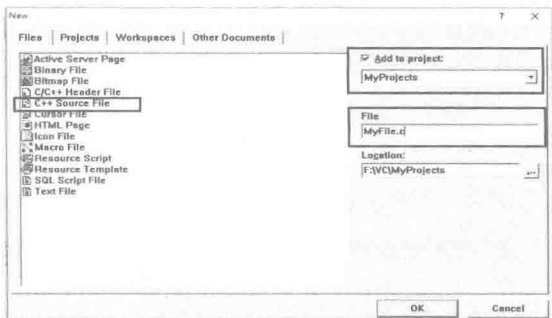


图 1.8 新建 C 源程序

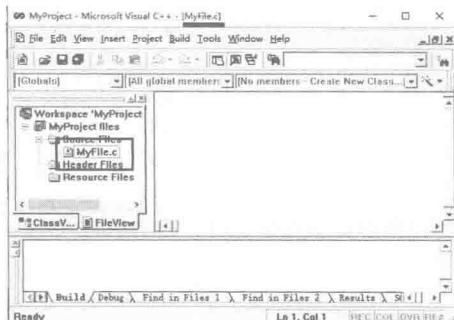


图 1.9 MyFile.c 编辑窗口

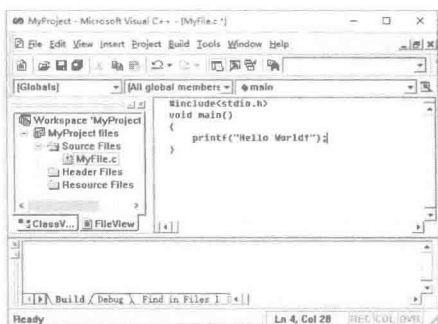


图 1.10 MyFile.c 的编辑




图 1.11 打开工程窗口

1.3 编译、链接和运行程序

1. 程序编译


编译可以检查程序中是否存在语法错误并生成目标文件(.obj)，编译结果显示在输出窗口中，如图 1.12 所示。

选择“Build”→“Compile MyFile.c”？即可对 MyFile.c 进行编译。此外，也可以通过单击  按钮或者直接按 Ctrl+F7 组合键进行编译。

如果程序中存在语法错误，则可以通过双击错误提示在程序文件中定位到错误所在的代码行。例如，把 printf 语句的分号去掉，则编译出现如图 1.13 所示的结果。根据提示框中的提示信息“syntax error; missing ';' before '}'”以及编辑区中的蓝色提示箭头则可以直接定位到错误处进行修改。

值得注意的是，语法错误分为 error 和 warning 两类。error 是一类致命错误，程序中如果有此类错误则无法生成目标程序，更不能执行。warning 则是相对轻微的一类错误，不会影响目标文件及可执行文件的生成，但是有可能影响程序的运行结果。因此，建议最好把所有错误（无论是 error 还是 warning）一一修正。

2. 程序链接

链接将生成可执行文件(.exe)。选择“Build”→“Build”选项可以完成程序的链接。此外，也可以通过单击  按钮或通过快捷键 F7 进行链接。类似的，链接后的结果也会出现在输出窗口中，如图 1.14 所示。如果链接失败，则同样会显示失败的具体原因。

3. 程序运行

选择“Build”→“Execute MyFile.exe”选项可以启动程序的执行。此外，也可以通过单击  按

钮或通过 Ctrl+F5 组合键启动程序的运行。开始运行后,将弹出一个 DOS 窗口,显示程序的运行结果,如图 1.15 所示。

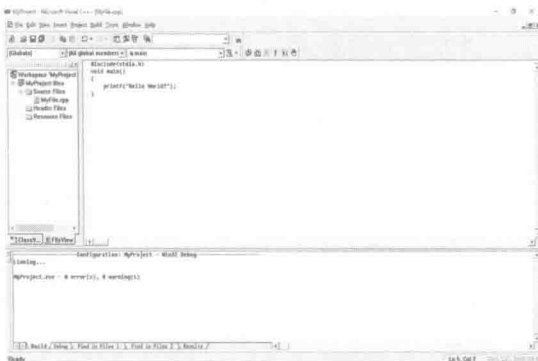


图 1.12 程序编译结果

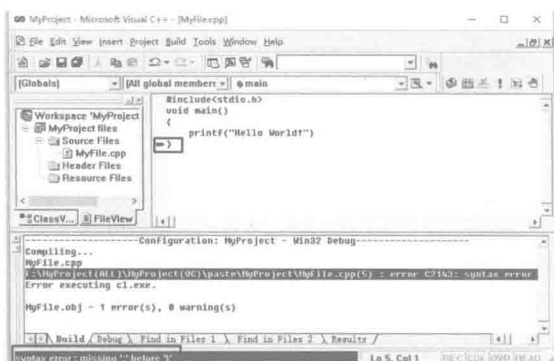


图 1.13 语法错误

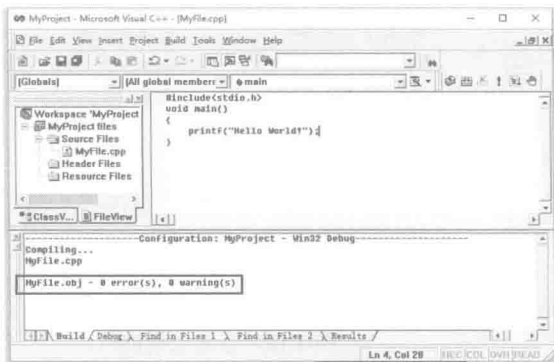


图 1.14 程序链接结果

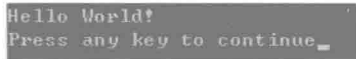



图 1.15 程序运行结果


1.4 调试程序

调试是为了发现程序中的错误,包括语法错误和逻辑错误。其中,语法错误能够在编译的过程中发现并修改;而逻辑错误往往无法直观地被发现,即程序通常能够被成功地编译和链接甚至执行,然而其执行结果却与预期结果不一致。逻辑错误的调试是比较困难的,因此,一般的程序开发环境都会提供完整的程序调试工具。本节将主要介绍如何使用 Visual C++ 6.0 中的调试工具发现程序中的逻辑错误。

1. 设置断点

当需要调试程序时,可以首先大致判断程序中可能从哪条语句开始出现问题,并将光标移动到该语句行,单击  按钮,则在语句行左侧出现一个红点,称为断点,如图 1.16 所示。此后,程序调试运行过程中遇到断点时,将自动暂停执行,进入调试状态。

2. 调试界面

设置好断点之后,单击  按钮,以调试的方式启动程序的运行。当运行到断点所在的语句时,将直接进入调试状态,如图 1.17 所示。代码行左侧的黄色小箭头表示了程序的当前执行位置。调试工具条中的按钮是常用的调试工具按钮,表 1.1 中列举了常用按钮的功能。屏幕下方左右两个窗口则分别是自动变量框和观察框,可以看到变量的当前值,作为判断程序是否出错的参考。其中,观察框支持对各种变量值的观察,在“Name”列输入变量的名字,则在“Value”列会显示出该变量的当前值。

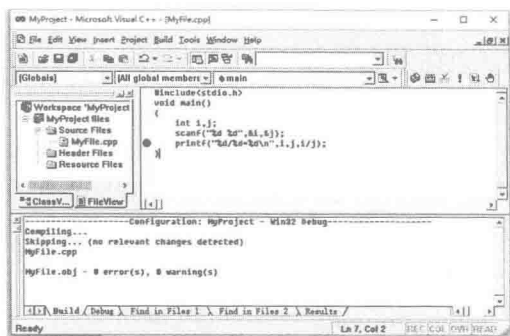


图 1.16 设置断点

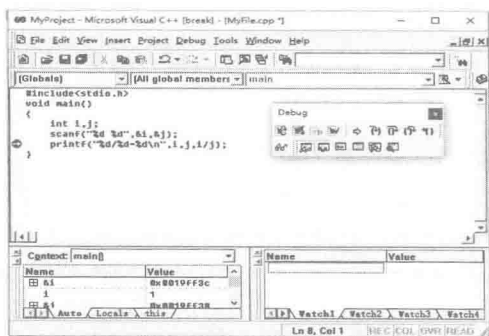








图 1.17 调试界面

表 1.1 调试工具常用按钮说明

按钮	功能	按钮	功能
	重新开始调试		停止/退出调试
	跟踪到函数中		逐个函数调试
	从跟踪的函数跳出		运行到光标处

3. 常见错误

C 语言的最大特点是：功能强、使用方便灵活。C 编译的程序对语法检查并不像其他高级语言那么严格，这就给编程人员留下“灵活的余地”，但还是由于这个灵活给程序的调试带来了许多不便，尤其对初学 C 语言的人来说，经常会出一些连自己都不知道错在哪里的错误。在本节编者整理了部分常见错误，及它们的解决办法。

(1) 书写标识符时，忽略了大小写字母的区别。

```
main()
{ int a=5;
  printf("%d",A);
}
```

提示：编译程序把 a 和 A 认为是两个不同的变量名，而显示出错信息。C 语言认为大写字母和小写字母是两个不同的字符。习惯上，符号常量名用大写表示，变量名用小写表示，以增加可读性。

(2) 忽略了变量的类型，进行了不合法的运算。

```
main()
{ float a,b;
  printf("%d",a%b);
}
```

提示：%是求余运算，得到 a/b 的整余数。整型变量 a 和 b 可以进行求余运算，而实型变量则不允许进行“求余”运算。

(3) 将字符常量与字符串常量混淆。

```
char c;
c="a";
```

提示：在这里就混淆了字符常量与字符串常量，字符常量是由一对单引号括起来的单个字符，字符串常量是一对双引号括起来的字符序列。C 语言规定以“\”作字符串结束标志，它是由系统自动加上的，所以字符串"a"实际上包含'a'和'\两个字符，而把它赋给一个字符变量是不行的。

(4) 忽略了“=”与“==”的区别。

C 语言中，“=”是赋值运算符，“==”是关系运算符。例如：

```
if (a==3)a=b;
```

提示：前者是进行比较，a 是否和 3 相等，后者表示如果 a 和 3 相等，把 b 值赋给 a。由于习惯问题，初学者往往会犯这样的错误。

(5) 忘记加分号。

分号是 C 语句中不可缺少的一部分，语句末尾必须有分号。

```
a=1
b=2
```

提示：编译时，编译程序在“a=1”后面没发现分号，就把下一行“b=2”也作为上一行语句的一部分，这就就会出现语法错误。改错时，有时在被指出有错的一行中未发现错误，就需要看一下上一行是否漏掉了分号。

(6) 多加分号。

对于一个复合语句，如：

```
{ z=x+y;
  t=z/100;
  printf("%f",t);
};
```

提示：复合语句的花括号后不应再加分号，否则将会画蛇添足。

(7) 输入变量时忘记加地址运算符“&”。

```
int a,b;
scanf("%d%d",a,b);
```

提示：这是不合法的。scanf 函数的作用是：按照 a、b 在内存的地址将 a、b 的值存进去。“&a”指 a 在内存中的地址。

(8) 输入数据的方式与要求不符。例如：

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

输入时，不能用逗号作两个数据间的分隔符，如下面输入不合法：

```
3, 4
```

输入数据时，在两个数据之间以一个或多个空格间隔，也可用 Enter 键、Tab 键。

又如：

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

提示：C 规定，如果在“格式控制”字符串中除了格式说明以外还有其他字符，则在输入数据时应输入与这些字符相同的字符。

(9) 输入字符的格式与要求不一致。

在用“%c”格式输入字符时，“空格字符”和“转义字符”都作为有效字符输入。例如：

```
scanf("%c%c%c",&c1,&c2,&c3);
```

如输入“a b c”。

提示：字符“a”送给 c1，字符“ ”送给 c2，字符“b”送给 c3，因为%c 只要求读入一个字符，后面不需要用空格作为两个字符的间隔。

(10) 输入输出的数据类型与所用格式说明符不一致。

例如，a 已定义为整型，b 定义为实型：

```
a=3;b=4.5;
printf("%f%d\n",a,b);
```

提示：编译时不给出出错信息，但运行结果将与原意不符。这种错误尤其需要注意。

(11) 输入数据时，企图规定精度。例如：

```
scanf("%7.2f",&a);
```

提示：输入数据时不能规定精度。

(12) switch 语句中漏写 break 语句。

例如：根据考试成绩的等级打印出百分制数段。

```
switch(grade)
{ case 'A':printf("85~100\n");
  case 'B':printf("70~84\n");
  case 'C':printf("60~69\n");
  case 'D':printf("<60\n");
```

```
default:printf("error\n");
```

提示：由于漏写了 `break` 语句，`case` 只起标号的作用，而不起判断作用。因此，当 `grade` 值为 `A` 时，`printf` 函数在执行完第一个语句后接着执行第二、三、四、五个 `printf` 函数语句。正确写法应在每个分支后再加上 “`break;`”。

(13) 忽视了 `while` 和 `do-while` 语句在细节上的区别。

例如：

```
main()
{ int a=0, I;
  scanf("%d", &I);
  while(I<=10)
  { a=a+I;
    I++;
  }
  printf("%d", a);
}
```

又如：

```
main()
{ int a=0, I;
  scanf("%d", &I);
  do
  { a=a+I;
    I++;
  } while(I<=10);
  printf("%d", a);
}
```

提示：可以看到，当输入 `I` 的值小于或等于 10 时，两者得到的结果相同。而当 `I>10` 时，两者结果就不同了。因为 `while` 循环是先判断后执行，而 `do-while` 循环是先执行后判断。对于大于 10 的数，`while` 循环一次也不执行循环体，而 `do-while` 语句则要执行一次循环体。

(14) 定义数组时误用变量。例如：

```
int n;
scanf("%d", &n);
int a[n];
```

提示：数组名后用方括号括起来的是常量表达式，可以包括常量和符号常量。即 C 不允许对数组的大小作动态定义。

(15) 在定义数组时，将定义的“元素个数”误认为是可使用的最大下标值。例如：

```
main()
{ static int a[10]={1,2,3,4,5,6,7,8,9,10};
  printf("%d", a[10]);
}
```

提示：C 语言规定，定义时用 `a[10]`，表示 `a` 数组有 10 个元素。其下标值由 0 开始，所以数组元素 `a[10]` 是不存在的。

(16) 初始化数组时，未使用静态存储。例如：

```
int a[3]={0,1,2};
```

提示：C 语言规定只有静态存储 (`static`) 数组和外部存储 (`extern`) 数组才能初始化。应改为：

```
static int a[3]={0,1,2};
```

(17) 在不应加地址运算符 `&` 的位置加了地址运算符。例如：

```
scanf("%s", &str);
```

提示：C 语言编译系统对数组名的处理是数组名代表该数组的起始地址，`scanf` 函数中输入项是字符数组名，不必再加地址符 “`&`”。改为：

```
scanf("%s", str);
```

第 2 部分 模块训练

- 模块 1 开始编写程序
- 模块 2 简单 C 语法
- 模块 3 顺序结构
- 模块 4 选择结构 (1)
- 模块 5 选择结构 (2)
- 模块 6 循环结构 (1)
- 模块 7 循环结构 (2)
- 模块 8 数组 (1)
- 模块 9 数组 (2)
- 模块 10 函数 (1)
- 模块 11 函数 (2)
- 模块 12 指针 (1)
- 模块 13 指针 (2)
- 模块 14 结构体与共用体
- 模块 15 链表
- 模块 16 文件

模块 1

开始编写程序

实验目的

- (1) 熟悉 VC++ 实验环境，了解 VC++ 文件管理、编辑、编译、链接和调试的基本技巧。
- (2) 熟悉一般 C 语言程序结构，对头文件、主函数、函数体语法有基本认识。
- (3) 能解决上机过程中遇到的常见问题。
- (4) 掌握算法的基本概念，掌握程序设计的一般流程。
- (5) 了解程序中，输入与输出的目的和意义。

实验准备

- (1) 完成 VC++6.0 环境的安装和设置。
- (2) 完成了 C 语言程序基本结构的学习。
- (3) 完成算法内容的学习。

实验内容

1. 基础训练

【训练 1】解析“hello world”。

源代码如下：

```
#include<stdio.h>
void main()
{
    printf(" hello world ");
}
```

程序的运行结果如图 2.1 所示。

提示：

- (1) #include 为关键字，打开相应头文件。
- (2) stdio.h 为标准输入输出头文件。
- (3) main() 为程序入口，即程序从此开始逐条执行。
- (4) printf() 为输出函数，输出双引号里面的内容。

【训练 2】输出一个好玩的界面！

源代码如下：

```
#include<stdio.h>
void main()
{
    printf("    /\ \    /\ \    /\ \    /\ \    /\ \    /\ \    \n");
    printf("    \/ \    \/ \    \/ \    \/ \    \/ \    \/ \    \n");
}
```

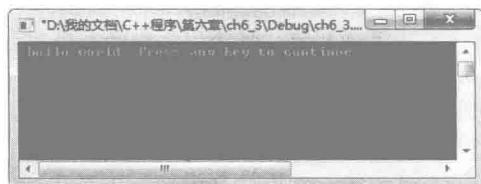


图 2.1 【训练 1】运行结果

```

printf("    || /\ \ || /\ \ || /\ \ || /\ \ || \n");
printf("    ||_\/\|_||_\/\|_||_\/\|_||_\/\|_|| \n");
printf("    ||____||____||____||____||____|| \n");
printf("    ||____||____||____||____||____|| \n");
printf("|@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@| \n");
printf(" |          HAPPY BIRTHDAY!          | \n");
printf("|@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@| \n");
}

```

程序的运行结果如图 2.2 所示。

提示：\n 为转义字符，功能为输出换行，关于转义字符后文会有介绍。

2. 进阶训练

【进阶 1】怎么输入字符和数字？

源代码如下：

```

#include<stdio.h>
void main()
{
    int a;
    scanf("%d",&a);
    printf("the input number is %d\n",a);
}

```

程序的运行结果如图 2.3 所示。

提示：

- (1) C 语言中所有变量需要先定义后使用。
- (2) int a 为定义一个 int 形式变量 a。
- (3) scanf() 为输入函数。

【进阶 2】C 程序来帮你做简单数学题。

源代码如下：

```

#include<stdio.h>
void main()
{
    int a,b,sum;
    scanf("%d%d",&a,&b);
    sum=a+b;
    printf("the sum is %d\n",sum);
}

```

程序的运行结果如图 2.4 所示。

3. 思考训练

【思考 1】选择交通工具。

武汉到北京，有飞机、高铁、汽车；假设飞机的票价是 600 元，高铁是 400 元，长途汽车是 300，你手中的钱，将影响你对交通工具的选择。尝试利用计算机程序，快速帮你完成交通工具的选择。

第一步，告诉计算机你手中多少钱。

第二步，让计算机判断你手中的钱可以购买哪种票。

第三步，输出选择的交通工具。

源代码如下：

```

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

```



图 2.2 【训练 2】运行结果



图 2.3 【进阶 1】运行结果

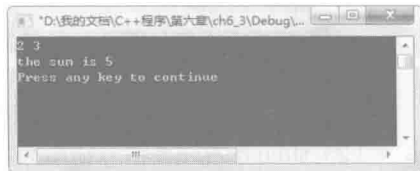


图 2.4 【进阶 2】运行结果


```

int a;
scanf("%d",&a);
if(a>=600)
{
    printf("选择飞机\n");
}
else if(a>=400)
{
    printf("选择高铁\n");
}
else if(a>=300)
{
    printf("选择长途汽车\n");
}
else
{
    printf("无法选择\n");
}
}

```

程序的运行结果如图 2.5 所示。

【思考 2】 怎么求和最方便？

$1+2+3+4+\dots+100=?$ 需要计算机怎么做，可以快速得到正确结果。

分析，要把 $1+2+3+4+\dots+100$ 全部输入，显然不合理，选择用一个循环，每次让被加数加一，再加入到总和里面，当被加数大于 100，停止循环。输出总和。

源代码如下：

```

#include<stdio.h>
void main()
{
    int a=1,sum=0;
    while(a<=100)
    {
        sum=sum+a;
        a=a+1;
    }
    printf("%d\n",sum);
}

```

程序的运行结果如图 2.6 所示。

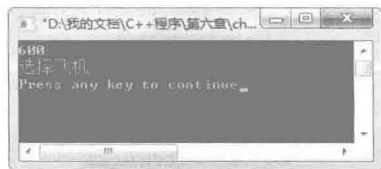


图 2.5 【思考】运行结果

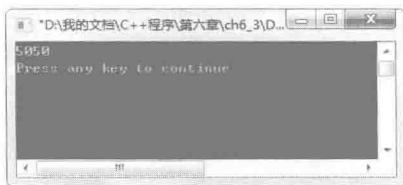


图 2.6 【思考 2】运行结果

章节要点

- (1) 理解和熟练完成 C 程序上机。
- (2) 算法是什么？有什么特点？算法的描述方式有哪些？
- (3) 结构化程序设计的方法是什么？其主要原则有哪些？

课后习题

1. C 语言程序的基本单位是 ()。
 - A. 程序行
 - B. 语句
 - C. 函数
 - D. 字符
2. 以下说法中正确的是 ()。
 - A. C 语言程序总是从第一个函数开始执行
 - B. 在 C 语言程序中，要调用的函数必须在 main() 函数中定义
 - C. C 语言程序总是从 main() 函数开始执行
 - D. C 语言程序中的 main() 函数必须放在程序的开始部分