普通高等教育“十一五”国家级规划教材配套实验教材

21世纪大学计算机系列教材

C/C++ 程序设计实验指导与测试（第3版）

孙淑霞　李思明　刘焕君　编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY　http://www.phei.com.cn

普通高等教育“十一五”国家级规划教材配套实验教材

21世纪大学计算机系列教材

C/C++ 程序设计实验指导与测试（第3版）

孙淑霞　李思明　刘焕君　编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书是普通高等教育“十一五”国家级规划教材《C/C++程序设计教程》（第3版）的配套实验与测试教材。全书共分为5部分，其中包括实验指导、测试、测试解答、模拟考试及答案、Visual C++ 6.0实验环境简介。实验指导由11个实验项目组成；测试部分由11章组成。实验指导和测试部分的内容均与《C/C++程序设计教程》（第3版）的每一章内容相对应，以便进行实验教学和学生课后练习。对于每一道实验题，书中都给出了较为详细的提示和帮助性的指导；测试部分的题型主要是针对理论考试题型设置的。书中还给出了5套C语言笔试和上机考试的模拟测试题及答案。附录中提供了C语言常见的编译错误信息、连接和运行中的错误信息。

本书可作为大专院校非计算机专业本科生、研究生的相关课程的实验教学用书，也可作为计算机专业相关课程的实验教材，还可作为C/C++程序设计自学者的参考书。

图书在版编目（CIP）数据

C/C++程序设计实验指导与测试/孙淑霞，李思明，刘焕君编著. —3版. —北京：电子工业出版社，2009.11
（21世纪大学计算机系列教材）
ISBN 978-7-121-09882-6

Ⅰ.C… Ⅱ.①孙…②李…③刘… Ⅲ.C语言—程序设计—高等学校—教学参考资料 Ⅳ.TP312

中国版本图书馆CIP数据核字（2009）第207160号

责任编辑：胡丽华　　文字编辑：王昌铭
印　　刷：北京市李史山胶印厂
装　　订：
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：787×1092　1/16　印张：18　字数：490千字
印　　次：2010年2月第2次印刷
印　　数：2 000册　定价：29.00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zlts@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

本书是与普通高等教育“十一五”国家级规划教材《C/C++程序设计教程》（第3版）配套使用的实验指导与测试。该书的前两版在多年的使用中，得到不少学校的支持，同时提出了一些宝贵意见。为了更好地满足广大教师和学生的需求，便于教师的教学和学生的学习，我们对其进行了进一步的改进，使该书具有如下特点：

1．在内容的安排上注重理论和实践的结合，使学生在学完每一章后，都可以通过完成相应的实验和测试习题巩固所学的理论知识。考虑到初学者的困难，实验指导根据题目的难易程度给予了不同程度的提示和帮助。测试习题部分的分析也根据习题的难易程度给予详略程度不同的解释，使学生能够真正掌握所学的知识点。

2．针对学生的不同基础，实验和习题中都给出了选做（打“*”的题目）题，尽可能地满足每个学生的需求。学生除了在老师指导下完成基本实验外，还可以根据自己的实际情况选做具有设计性和综合性的实验。

3．每一个实验后面的测试题由学生独立完成，可用于教师随堂测试学生的测试题，也可以作为学生的自测试题。

4．为了使学生从一开始就掌握在Visual C++6.0环境下调试C/C++程序，在实验1中就安排了Visual C++ 6.0的实验内容，以便在后面的学习中，学生可以根据具体情况选择编写程序和调试程序的平台。

5．第2部分的测试题用于课后学生自测学习；对于其中的*题可供学有余力的学生课后练习、提高编程能力。

为了帮助读者解决上机调试程序中的错误，附录中提供了C语言常见的编译错误信息、连接和运行中的错误信息。

要想学好程序设计课程，需要教师和学生的共同努力。对于学习者来说，需要多动手，多实践，多思考。一分耕耘，一分收获，坚持耕耘定会得到意想不到的收获。

本书第1、4、5部分由孙淑霞编写，第2、3部分由李思明和刘焕君编写。丁照宇、肖阳春、彭舰、陈佩良也参加了本书的策划、出题、解题和实验等工作，孙淑霞统编了全书。由于水平有限，书中难免有错误之处，请读者批评指正。

最后要感谢电子工业出版社在本书的出版过程中给予的大力支持。

该书作为国家级精品课程《C/C++程序设计教程》使用的配套教材，对其进行了配套的资源建设。对于使用本教材的学校，如果需要书中的有关资源，可以从该课程的精品课程网站上 http://202.115.138.30/ec3.0/c57/zcr-1.htm 直接下载，也可以直接与我们联系（邮件地址：ssx@cdut.edu.cn）。

编著者

2009年9月

目　录

第1部分　实 验 指 导

实验1　C/C++语言简单程序的编写和调试

一、实验目的

熟悉 Turbo C 集成环境和 Visual C++ 6.0 编译系统，掌握在两种环境中进行程序调试的一般方法。

二、实验要求

1．熟悉 Turbo C 集成环境的使用方法，掌握在 Turbo C 集成环境下输入、编译、连接、运行和调试 C 程序的基本过程和方法。

2．熟悉 Visual C++ 6.0 编译系统的窗口界面，学会在该系统下编辑、编译、连接、运行和调试 C++程序的基本方法。

3．通过编写简单程序，掌握 C/C++程序的基本组成和结构，以及用 C/C++程序解决实际问题的步骤。

4．在老师指导下完成实验 1.1~1.5，独立完成测试 1.1 和 1.2。

三、实验内容

在课程网站上下载并解压 CSY.RAR 文件，其中 SY1 文件夹中包含本实验中的相关文件：S1-1.C、S1-3.C、S1-4.CPP、S1-5-1.CPP、S1-5-2.CPP、Z1-1.C、Z1-2.C。

1.1 在 Turbo C 集成环境中打开程序 S1-1.C，对其进行编译，连接和运行。

【指导】

（1）启动 Turbo C，进入 Turbo C 集成环境。

（2）按【F3】键打开程序 S1-1.C。

源程序 S1-1.C

```
#include <stdio.h>
void main()
{
  printf ("Hello, C! \n");
}
```

（3）执行“Compile”菜单中的“Compile to OBJ”命令，编译程序 S1-1.C，产生目标程序 S1-1.OBJ。

（4）执行“Compile”菜单中的“Link EXE file”命令，连接目标程序 S1-1.OBJ 和函数库 stdio.h，生成可执行程序 S1-1.EXE。

（5）执行“Run”菜单中的“Run”命令，按组合键 Alt-F5，观察屏幕上程序的运行结果：

Hello, C!

提示：

“Compile”菜单中的“Make EXE file”命令也用于连接程序，与“Link EXE file”命令不同的是当程序重新编辑后，可直接使用“Make EXE file”命令，编译程序会自动完成编译、连接过程；而“Link EXE file”命令只用于连接程序。

1.2 使用块定义、块移动、块复制等操作将 S1-1.C 编辑为 S1-2.C，练习程序的编辑和“另存为”命令。

源程序 S1-2.C

```
#include <stdio.h>
void main()
{
  printf ("Hello, C! \n");
  printf ("Hello, Student!\n");              /* 用块操作完成该行的增加 */
}
```

【指导】

熟练使用块操作可以提高编辑程序的效率，因此不要忽视该题目的练习。

（1）按如下方法增加语句行：

```
printf ("Hello, Student!\n");
```

操作步骤：

① 将光标定位在 printf ("Hello, C! \n");语句行的开始，按组合键 Ctrl-K B（按住 Ctrl 键，然后依次按字母键 K 和 B）定义块首；

② 将光标定位在该行的末尾，按组合键 Ctrl-K K，定义块尾，这时该行被着色；

③ 将光标定位在下一行的行首，按组合键 Ctrl-K C，将定义的块复制到该行；

④ 按组合键 Ctrl-K H，取消块定义；

⑤ 将 printf ("Hello, C! \n"); 中的"Hello, C! \n"修改为："Hello, Student!\n"。

（2）执行 File 菜单中的“Write to”命令（相当于“另存为”命令），在“New Name”框中输入文件名 S1-2.C，按回车键，将当前文件保存为 S1-2.C。

1.3 练习程序 S1-3.C 的调试，直到 S1-3.C 调试正确。

源程序 S1-3.C

```
#include <stdio.h>
void mian()
{
  int a, b, sum;

  a=40; b=50;
  sum=a+b;

  printf ("Hello, C! \n")
  printf ("Sum is %d \n", sum);
}
```

【指导】

（1）执行“Compile”菜单中的“Compile to OBJ”命令，编译 S1-3.C，这时 Turbo C 集成环境的 Message 窗口中出现如下信息：

```
Compiling E:\C\S1-3.C:
Error E:\C\S1-3.C 9: Statement missing ; in function main
Warning E:\C\S1-3.C 10: 'sum' is assigned a value while is never used in function main
```

其中：

第 1 条信息表示当前正在编译的程序是 E: 盘文件夹 C 中的 S1-3.C；

第 2 条信息指出在 S1-3.C 的第 9 行有语法错误，错误是：在 main 函数的第 9 行少了分号“;”。注意，这种错误通常是因为上一行的末尾少了分号“;”，而不是当前行少分号；

第 3 条信息指出在 S1-3.C 的第 10 行有警告错误，错误是：在 main 函数中，sum 被赋予了一个未使用过的值。这个错误是由于上一个错误造成的，如果在上一行的末尾加上分号，该错误就不会再出现。

（2）执行“Compile”菜单中的“Compile to OBJ”命令，重新编译 S1-3.C，没有错时生成 S1-3.EXE。

这时不再有错。可见，上面的警告错误并没有被修改就自动消失了，这是因为改正了第 2 条信息的错误。因此，修改 C 程序编译的错误时，每修改一处，就要编译一次程序，后面的一些错误可能是前面某处错误引起的。

（3）执行“Compile”菜单中的“Make EXE file”命令，连接 S1-3.OBJ 生成 S1-3.EXE。

这时 Message 窗口中显示如下错误信息：

```
Linking E:\C\S1-3.EXE:
Linking Error: Undefined symbol '_main' in module COS
```

其中：

第 1 条信息表示当前正在连接的程序是 E: 盘文件夹 C 中的 S1-3.EXE；

第 2 条连接错误信息是：没有定义 main 函数，造成该错误的原因是函数名 main 写错了。

🕮 提示：

（1）在调试程序的过程中，如果出现编译错误，要由上至下一个一个去修改，每改一处，就要重新编译一次，不要想着一次把所有错误都修改后再编译。因为，有时一个错误会引起下面程序段中与之有关的行也出现错误，改正了这一个错误，其他与之有关行的错误也就随之消失了。

（2）有些错误会出现在连接阶段，例如，在实验 1-3 中，把 main 误写成了 mian，编译程序把 mian 当成是用户自定义函数进行编译，没有语法错误，也就没有报错。但由于 C 程序必须要有一个且只能有一个 main()函数，连接程序没有发现 main()函数，因此在连接阶段报错。

与之类似的错误有：将 printf 误写为 print，连接程序也会因为找不到相应的库，在连接阶段给出连接错误的提示信息：

```
Undefined symber '_print'
```

说明 print 是未定义符号。

（3）当调试程序中出现了编译、连接或运行错误，可以查看附录 C 中提供的常见错误信息。要注意培养自己独立分析问题和解决问题的能力，积累查错的经验，逐渐提高调试程序的能力；千万不要被错误所吓倒，相信自己一定会在调试程序的过程中成长起来。

1.4　在 Visual C++ 6.0 中练习单文件程序 S1-4.CPP 的编译、连接和运行，使程序得到正确结果。

【指导】

（1）在 Visual C++ 6.0 中打开 S1-4.CPP

启动 Visual C++ 6.0，打开 Visual C++ 6.0 窗口。执行“File| Open”命令，或单击“标准”工具栏上的“打开”按钮，弹出“打开”对话框。在该对话框中的“查找范围”框中指定待打开文件被存放的位置，在文件列表框中找到要打开的文件 S1-4.CPP，双击该文件，或单击该文件再单击“打开”按钮。

（2）编译

选择下面方法之一进行编译：

① 执行“Build | Compile S1-4.cpp”命令；

② 单击“编译微型条”工具栏中的“编译”按钮；

③ 按<Ctrl>+<F7>键编译程序 S1-4.cpp。

如果没有建立工程工作区，这时将弹出如图 1.1 所示的对话框。询问是否要建立一个默认的工程工作区？Visual C++必须有工程才能编译，因此这里必须回答“是”。

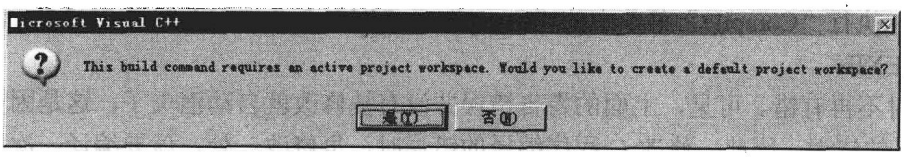

图 1.1　信息提示对话框

单击“是”按钮，开始编译该文件。编译过程中，如果有错，则在主窗口下方的编译输出窗口中显示错误信息。错误信息将指出错误的性质、出错位置（比如行数）以及错误原因；如果没有错，编译输出窗口中就会显示生成的.OBJ 文件名。例如：

S1-4.obj - 0 error(s), 0 warning(s)

这时，在保存 S1-4.C 文件的文件夹中会生成与 S1-4.C 同名的.dsw 和.dsp 文件。

修改编译错误时，在编译输出窗口中双击某一错误信息行时，在该错误信息行前就会出现一个提示箭头。用户可以根据错误的性质修改程序，修改后再重新编译，直到没有任何错误为止。

（3）连接

执行“Build |Build S1-4.exe”命令或按<F7>键对被编译后的目标文件进行连接。如果连接过程中发现错误，则会发出连接错误的信息，修改程序直到连接没有错误为止。编译连接成功时，在编译输出窗口中就会显示如下信息：

S1-4.exe—0 error(s)　0warning(s)

表明这时生成了可执行文件 S1-4.exe。

（4）执行

执行“Build | Execute S1-4.exe”命令，或按<Ctrl>+<F5>键，或单击“编译微型条”工具栏中的“执行”按钮，S1-4.exe 文件被执行，并在另一个输出结果的 DOS 窗口中显示结果：

area is 200

Press any key to continue

按任意键后，关闭输出结果窗口，回到源程序窗口。

提示：

Visual C++ 6.0 的调试功能很强大，表 1.1 是最基本的调试命令及其图标、快捷键和说明。在调试程序中适当的使用这些命令可以提高调试程序的效率。

表 1.1　VC++ 6.0 基本调试命令

命　令	图　标	快 捷 键	说　明
Go		F5	运行程序
Insert/Remove Breakpoint		F9	插入或删除断点
Run to Cursor		Ctrl+F10	运行到光标所在行
Step Into		F11	进入函数内部单步执行
Step Out		Shift+F11	跳出当前函数
Step Over		F10	执行下一条语句，不进入函数
Stop Debugging		Shift+F5	停止调试程序

1.5　在 Visual C++ 6.0 中练习多文件程序 S1-5-1.CPP 和 S1-5-2.CPP 的编译、连接和运行，使程序得到正确结果。

【指导】

（1）创建工程(project)

在 Visual C++ 6.0 主窗口中，执行“File | New”命令，出现“New”对话框，单击“Projects”标签，然后按以下步骤执行：

① 在列出的工程中选择“Win32 Console Application”，这时，在对话框的“Platforms(平台)”框中出现“Win32”；

② 在右边的“Projects Name”文本框中输入要建立的工程名(如 ps1_5)；

③ 在“Location”框中选择工程所在的路径，单击“OK”按钮，如图 1.2 所示。

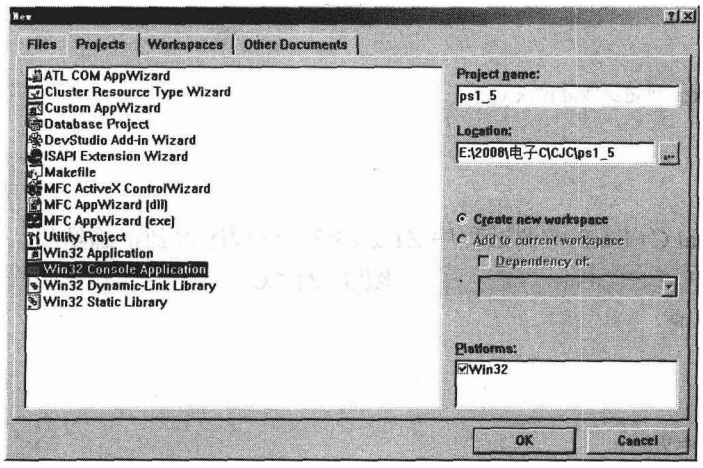

图 1.2　“New”对话框

④ 屏幕上出现“Win32 Console Application-Step 1 of 1”对话框，选择“An empty project”单选按钮，单击“Finish”按钮，弹出“New Project Information”对话框，单击“OK”按钮，返回系统主窗口。

这时，空工程 ps1_5 创建结束。

（2）向工程中添加文件

在系统主窗口中，执行“Projects | Add File into Project | Files”菜单命令，出现“Insert Files into Project”对话框，在该对话框中，在“查找范围”框内选择要添加到当前工程中的文件所在的目录（文件夹），在列表框内选定要添加的所有文件，如图 1.3 所示。

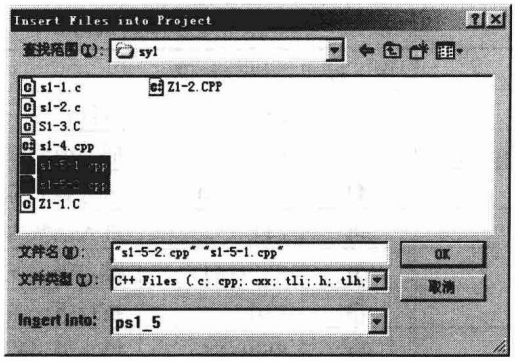

图 1.3 “Insert Files into Project”对话框

单击“OK”按钮。

（3）编译连接和运行工程文件

执行“Build | ps1_5.exe”命令，系统将对 ps1_5 中的各个文件逐个进行编译，然后连接。如果没有错，则生成一个可执行文件，并执行该文件，运行结果显示在 MS-DOS 窗口中。

四、测试

1.1 在 TC 2.0 中调试 Z1-1.C，使程序得到正确结果。

源程序 Z1-1.C

```
#include <stdio.h>
void main()
{
  int x,y,z;
   printf ("x=%d,y=%d,z=%d\n",x,y,z);
   x=5,y=8;
   z=x+y;
}
```

1.2 在 Visual C++ 6.0 中调试程序 Z1-2.CPP，使程序得到正确结果。

源程序 Z1-2.C

```
#include <stdio.h>
void main()
{
    int a=2,b=3;
    float x=8.2,y=9.2,z;

    z=x/2+y/b;
    printf ("z=%d\n",z);
}
```

实验 2 C 语言程序设计基础

一、实验目的

掌握 C 语言的运算符和表达式的正确使用以及 C 语言几种基本数据类型的定义和初始化，掌握基本输入/输出函数的使用方法。

二、实验要求

1．通过编程进一步理解和掌握运算符的确切含义和功能。

2．理解和掌握运算符与运算对象的关系。例如单目运算符只对一个运算对象进行操作，双目运算符需要两个运算对象。

3．理解和掌握运算符的优先级和结合方向。

4．通过编写程序，掌握 C 语言的几种基本数据类型，如整型 int、字符型 char、实型 float、双精度型 double，以及由这些基本类型构成的常量和变量的使用方法。

5．掌握基本输入/输出函数的使用方法。其中，包括 printf()函数、scanf()函数、getchar()函数和 putchar()函数。

6．在老师指导下完成实验 2.1~2.7，独立完成测试 2.1~2.3。

三、实验内容

在课程网站上下载并解压 CSY.RAR 文件，其中 SY2 文件夹中包含本实验中的相关文件：S2-1-1.C、S2-1-2.C、S2-1-3.C、S2-3.C、S2-4.C、S2-5.C、S2-6.C、S2-7.C、Z2-1.C、Z2-2.C。

2.1　依次运行程序 S2-1-1.C、S2-1-2.C、S2-1-3.C，比较 3 个程序，认识如何为用户编写一个好用的程序。

源程序 S2-1-1.C

```
#include <stdio.h>
void main()
{
  int x=1,y=1,sum;
  sum=x+y;
  printf ("sum=%d\n",sum);
}
```

源程序 S2-1-2.C

```
#include <stdio.h>
void main()
{
  int x,y,sum;
  scanf("%d%d",&x,&y);
  sum=x+y;
  printf ("sum=%d\n",sum);
}
```

源程序 S2-1-3.C

```
#include <stdio.h>
void main()
{
  int x,y,sum;
  printf("\nEnter x y: ");
  scanf("%d%d",&x,&y);
  sum=x+y;
```

```
    printf ("sum=%d\n",sum);
}
```

【指导】

（1）程序 S2-1-1.C 运行结果为：

```
sum=2
```

在 S2-1-1.C 中，变量 x, y 的值由变量初始化得到。该程序只能求 x=1 与 y=1 的和，如果要求其他数据的和，必须修改程序中的初始化，然后重新编译、连接并运行程序。这种编程方法不灵活，只供初学编程使用。

（2）程序 S2-1-2.C 运行实例：

```
1 1↙
sum=2
```

在 S2-1-2.C 中，变量 x, y 的值是在程序运行过程中通过 scanf()函数输入的，这种方法不需要对程序做任何修改，就可以计算其他数据之和。但在输入数据前，没有任何提示，初学者很难知道程序在等待输入，因此需要进行进一步修改。

（3）程序 S2-1-3.C 运行实例：

```
Enter x y: 1 1↙
sum=2
```

在 S2-1-3.C 中，变量 x,y 的值也是在程序运行过程中通过 scanf()函数输入的，与 S2-1-2.C 不同的是，在输入数据前，增加了屏幕提示信息：

```
Enter x y:
```

用户可按屏幕提示信息进行数据的输入。这种方法比前两种更易于理解，更加灵活和人性化。同学以后可以模仿 S2-1-3.C 的编程方法。

🕮 提示：

通过运行 S2-1-1.C、S2-1-2.C、S2-1-3.C，应该学会用良好的编程风格编写程序。

（1）编写程序时要考虑程序的通用性，需要变化的量尽量不要通过赋值的方式给定（例如，S2-1-1.C 中的 x 和 y），而是通过输入的方式使变量得到当前所需的值（例如，S2-1-2.C 中对 x 和 y 的输入）。

（2）从键盘输入数据时，最好先给出提示信息，提示要输入数据（例 S2-1-3.C 中在输入 x 和 y 前用 printf()函数输出屏幕提示信息）。

2.2 编写程序 S2-2.C。输入一个学生某学期的 M（M=5）门课程的成绩（整型数），计算并输出该学生的平均成绩（保留两位小数）。

【指导】

程序实现步骤：

（1）给 5 个已定义的整型变量，输入 5 门课程成绩。

```
printf("Enter x1,x2,x3,x4,x5: ");
scanf("%d,%d,%d,%d,%d",&x1,&x2,&x3,&x4,&x5);
```

（2）计算平均成绩，并赋给一个实型变量。

```
average=(x1+x2+x3+x4+x5)/5.0;
```

（3）输出平均成绩。

🕮 提示：

计算平均成绩时不要做整除运算，因为整除运算只能得到整数部分，舍弃小数部分。通过该程序进一步认识数据类型，以及不同类型数据的混合运算。

2.3 运行程序 S2-3.C，观察输出的结果，并对输出结果作出合理的解释。

源程序 S2-3.C

```
#include<stdio.h>
void main()
{
   float a1,a2;
   double b1,b2;

   a1=3141.59; a2=0.000001;
   b1=3141.59; b2=0.000001;
   printf("%f, %lf\n", a1+a2, b1+b2);
}
```

【指导】

程序运行结果：

3141.590089, 3141.590001

程序中的 a1 和 b1、a2 和 b2 的值分别相同，但程序中 a1+a2 和 b1+b2 的输出结果却不相同。这是因为实型数在计算机中是非精确表示的，float 型数据和 double 型数据的精度不同所致。因此编写程序时要根据精度的要求定义变量的类型。

2.4 运行程序 S2-4.C，理解输出结果，并对输出结果给予合理的解释。

源程序 S2-4.C

```
#include<stdio.h>
void main()
{
   char ch;
   int k;

   ch='a'; k=10;
   printf("%d, %x, %o, %c", ch, ch, ch, ch, k);
   printf ("k=%%d\n",k);
   ch='9';
   printf("%c,%d",ch,ch-'0');
}
```

【指导】

程序 S2-4.C 运行结果如下：

97, 61, 141, ak=%d

9, 9

在 C 语言中，字符数据既可以用字符形式输出，也可以用整数形式输出。字符 a 的 ASCII 按十进制形式输出为 97，按十六进制形式输出为 61，按八进制形式输出则为 141。

程序中第 1 个 printf()函数调用中有 4 个格式描述符，输出表列中有 5 项，因此输出项 k 为多余的，不予输出；第 2 个 printf()函数调用中的格式说明中包含了两个连续的%字符，根据 C 语言的规定，%%不再作为格式描述字符使用，而是处理成字符“%”的原样输出，因此，该 printf()中的输出项 k 也没有对应的格式描述符，不予输出。

程序中第 3 个 printf()函数调用中，按%c 输出变量 ch 的值 9（字符 9），按%d 输出 ch-'0'，即'9'-'0'，结果为 9（数字 9）。

提示：

一个数字字符（0~9）减去数字字符 0 的结果为该数字字符对应的整数，即把数字字符转换为整型数。例如：'9'-'0'=9。

通过该程序，学习者可以进一步认识格式描述符以及 printf 函数中参数的正确使用。

2.5 运行程序 S2-5.C，并对输出结果给予合理的解释。

源程序 S2-5.C

```
#include <stdio.h>
void main()
{
   float x;
   double y;

   x=213.82631; y=213.82631;
   printf ("%-4.2f, -6.2e\n", x,y);
}
```

【指导】

程序 S2-5.C 运行结果如下：

213.83, 2.1e+02

使用 f 格式描述符输出浮点数时的一般形式是%m.nf 或%-m.nf，其中，m 指定输出数据所占的总列数；n 指定小数点后的位数，“-”是使输出数据左对齐，当输出数据宽度大于 m 时，数据的整数部分将按照实际位数输出，在本例中由于输出数据宽度大于 m，因此%后面的“-”对输出格式无影响。使用 e 格式描述符输出浮点数时的一般形式是%m.ne 或%-m.ne，其中，m、n、“-”的含义与 f 格式相同。在不同的计算机中，对指数部分应占的宽度规定不同，而数值部分均按标准化指数形式输出（即小数点前必须有而且仅有一位非零数字）。该题中 y 的实际宽度为 9，而格式说明中所给定的域宽为 6，所以只能按标准化指数形式输出 y 的整数部分，截去小数部分并四舍五入。

通过该程序可以进一步认识 f 格式描述符和 e 格式描述符。

2.6 程序 S2-6.C 的功能是从键盘上输入 x=25，y=36.7，c=C，然后将输入的内容输出到屏幕上。调试程序 S2-6.C，修改有错误的语句行，并输出正确的结果。

源程序 S2-6.C

```
#include<stdio.h>
void main()
 {
   int x;
   float y;
   char c;

   scanf("x=%d,y=%d,c=%c",x,y,c);
   printf("\nx=%d,y=%d,c=%c",x,y,c);
}
```

【指导】

程序 S2-6.C 在编译时有警告错误，如果不改正这些警告错误，程序将得不到正确的运行结果。

调试程序 S2-6.C 时要注意如下问题：

（1）scanf()函数的输入表列要求是变量地址的表列，否则在编译时会给出警告错误：

```
Possible use of 'x' before definition in function main
Possible use of 'y' before definition in function main
Possible use of 'c' before definition in function main
```

（2）scanf()函数的格式描述符要与输入表列中变量的类型一致。如果类型不一致，将得不到正确的输入值。将 float 类型变量用%d 格式输入，得到的是 0；int 型变量用%f 格式输入，运行时会给出错误信息：

```
scanf: floating point formats not linked
Abnormal program termination
```

（3）scanf()函数的格式描述符中有按原样输入的字符“x=,y=,c=”。输入数据时一定要按原样输入这些字符。例如，运行该程序时输入：

```
x=25，y=36.7，c=C
```

是正确的输入格式。如果输入：

```
25，36.7，C
```

则得不到正确结果。

（4）printf()函数的格式描述符要与输出表列中变量的类型一致。否则得不到正确的输出结果。将 float 型变量用%d 格式输出，得到的是 0；int 型变量用%f 格式输出，运行时会给出错误信息：

```
printf: floating point formats not linked
Abnormal program termination
```

2.7　完善程序 S2-7.C，使程序运行后输出 a, b, c, d 的值，以及表达式++a||++b&&++c 和 d=w>x && y>z 的值，验证其结果。

源程序 S2-7.C

```
#include<stdio.h>
void main()
{
    int a,b,c,d,w=1,x=2,y=3,z=4;

    a=b=c=1;
    ++a||++b&&++c;
    d=w>x && y>z;

}
```

【指导】

程序 S2-7.C 运行结果如下：

```
a=2, b=1, c=1, d=0
```

（1）要观察程序 S2-7.C 运行结果，首先要完善程序的输出，即在程序末尾添加：

```
printf("++a||++b&&++c=%d\n", ++a||++b&&++c);
print("d=w>x && y>z=%d", d=w>x && y>z);
printf("a=%d, b=%d, c=%d, d=%d", a, b, c, d);
```

（2）对于逻辑表达式的计算：

① ++a||++b&&++c，由于++a 为真，所以后面的运算不再进行。

② w>x && y>z，由于 w>x 为假，所以整个表达式的结果为假，即为 0。