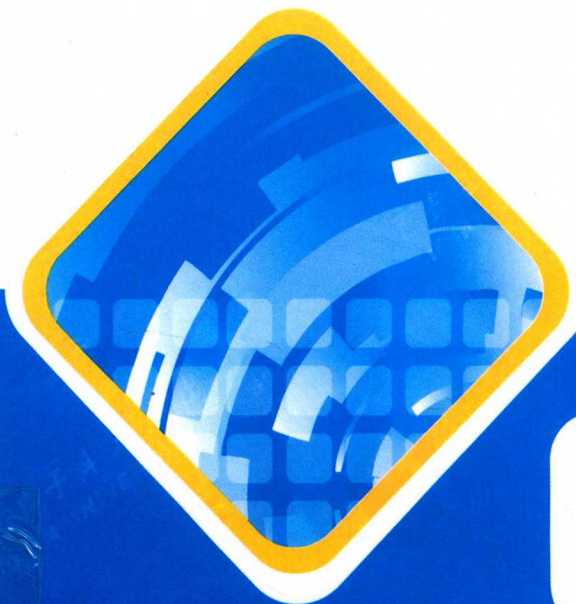
高职高专计算机专业“十二五”规划教材

C语言程序设计



主编　李圣良　虞 芬

Computer program

西安电子科技大学出版社
http://www.xduph.com

高职高专计算机专业“十二五”规划教材

C 语言程序设计

主编　李圣良　虞　芬
参编　代　飞　胡志锋　艾　迪
裴南平　王城华

西安电子科技大学出版社

内 容 简 介

本书是高职高专学生学习计算机编程的入门教材，着重讲述了计算机程序设计的基础知识、基本算法和应用编程思想，其目的在于使学生学习 C 语言程序设计之后，能结合社会生产实际进行应用程序的开发。

本书是作者多年来在讲授“C 语言程序设计”课程的基础上，总结多年的教学经验，对授课讲义进行整理而成的。全书共分 10 个单元，主要内容包括编写 C 程序的基础知识、顺序和选择结构程序设计、循环结构程序设计、数组、函数、指针、结构体和共用体、文件、编译预处理、位运算；另配有实验指导部分。本书整体结构编排合理，组织形式新颖，例题丰富，符合学生的认知规律和学习特点。通过本书的学习，学生可掌握程序设计的基本思想和常见简单问题的算法，并可编写程序加以实现。

本书层次分明、结构紧凑，叙述深入浅出、通俗易懂，适合作为高职高专相关专业教材，也可作为等级考试和其他计算机编程人员的参考书。

图书在版编目(CIP)数据

C 语言程序设计/李圣良，虞芬主编.
—西安：西安电子科技大学出版社，2015.2
高职高专计算机专业“十二五”规划教材
ISBN 978-7-5606-3661-0

Ⅰ. ① C… Ⅱ. ① 李… ② 虞… Ⅲ. ① C 语言—程序设计—高等职业教育—教材
Ⅳ. ① TP312

中国版本图书馆 CIP 数据核字(2015)第 028853 号

策　　划　邵汉平
责任编辑　邵汉平
出版发行　西安电子科技大学出版社（西安市太白南路 2 号）
电　　话　(029)88242885　88201467　邮　　编　710071
网　　址　www.xduph.com　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　陕西华沐印刷科技有限责任公司
版　　次　2015 年 2 月第 1 版　2015 年 2 月第 1 次印刷
开　　本　787 毫米×1092 毫米　1/16　印张 23
字　　数　548 千字
印　　数　1～3000 册
定　　价　40.00 元

ISBN 978 – 7 – 5606 – 3661 – 0 / TP
XDUP 3953001-1

前　言

最近 20 多年来，计算机技术飞速发展，出现了很多高级程序设计语言，其中 C 语言家族最具影响力。C++、Java 和 C#都属于 C 语言家族，C 语言是它们的基础。因此，国内高校的很多专业都将 C 语言作为第一门程序设计语言课程开设。作者于 2009 年编写了《C 语言程序设计》讲义，并作为校本开发教材在学校连续使用了 6 年。该讲义充分考虑了高职高专学生的实际情况，力求具备起点低、概念准确、讲解通俗、深入浅出、注重实践、强化应用的特点，多年来一直反映良好。本书即为对讲义进行优化整理而成的。

本书主要适用于工科各专业，书中的部分内容和实例可根据各专业的实际情况进行取舍。建议学时数为 70 学时，其中理论教学 50 学时，课内上机 20 学时。另外，建议计算机相关专业再安排一周的课程设计(实训)。

本书由九江职业技术学院李圣良、虞芬任主编，其中虞芬编写第 3 单元及附录部分，代飞编写第 2、8 单元，胡志锋编写第 4、5 单元，艾迪编写第 6、7 单元，王城华编写第 1、9、10 单元，裴南平编写实验指导部分，李圣良负责本书的统稿工作。

本书内容翔实，层次分明，结构紧凑，叙述深入浅出、通俗易懂。每个单元和每节后均附有大量习题，以利于学生对所学知识的巩固和编程能力的提高。书末配有实验指导，可供学生上机实训使用。本书适合作为高职高专及各类大专院校的教材，也可作为等级考试和其他计算机编程人员的参考书。

由于作者水平有限，书中不妥之处在所难免，恳请读者批评指正。

作　者

2014 年 12 月

目　录

第 1 单元　编写 C 程序的基础知识

单元描述

C 语言具有很多突出的优点，是非常实用且应用广泛的程序设计语言。本单元介绍编写 C 程序的基础知识：算法基础，C 语言程序的结构，简单 C 程序的编写与调试运行，数据类型及常用算术、赋值、逗号等运算符。

学习目标

通过本单元学习，你将能够

- 阅读简单的 C 语言程序，了解 C 程序的结构
- 了解 C 语言的基本数据类型，掌握数据的表示方法
- 掌握算术、赋值、逗号等运算符及其构成的表达式
- 设计简单的 C 语言程序
- 熟练调试和运行 C 语言程序

1.1　C 程序的编写、调试和运行

1.1.1　C 程序的结构

1. C 语言程序的结构分析

在使用 C 语言编写程序时，必须按其规定的格式进行编写。下面从几个简单的 C 语言程序入手，分析 C 语言程序的结构特点和书写格式。

【例 1-1】　简单的 C 程序。

```
# include <stdio.h>
main()                              /* 定义主函数，函数首部 */
{                                   /* 主函数的函数体开始 */
    printf ("Hello,world!\n");      /* 语句，输出“Hello,world!” */
}                                   /* 主函数的函数体结束 */
```

程序运行结果如图 1-1 所示。

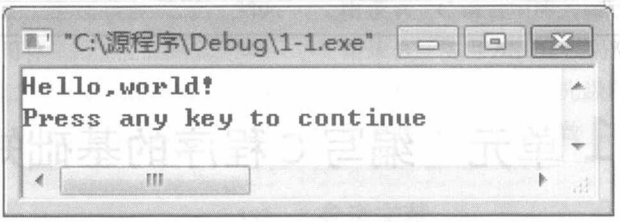


图 1-1　例 1-1 程序运行结果

程序分析如下：

(1) 这是一个完整的 C 程序，包含一个 main 函数(也称为主函数)。

(2) “main()”为主函数的函数首部，随后的“{……}”为主函数的函数体。

(3) 该程序中只有一条语句“printf ("Hello,world!\n"); ”，该语句能够在屏幕上输出“Hello,world!”，语句以分号“;”为结束符。

(4) “/*……*/”是对程序起到说明作用的注释部分，程序执行时并不执行注释部分。

(5) “# include <stdio.h>”是编译预处理命令。printf 是系统提供的标准库函数之一，使用这些函数时，应该在程序开始处使用编译预处理命令，将相应的头文件包含进来。常用的系统标准库函数及对应的头文件见附录 D。

【例 1-2】　函数体略复杂一些的 C 程序。

```
# include <stdio.h>
main( )
{    float r,s ;                          /* 声明部分：定义实数类型变量 r、s */
     r=1.5 ;                              /* 执行部分：圆的半径值赋值为 1.5 */
     s=3.14159*r*r ;                      /* 执行部分：求圆的面积值 s */
     printf("area is : %f\n",s);          /* 执行部分：输出圆的面积值 */
}
```

程序运行结果如图 1-2 所示。

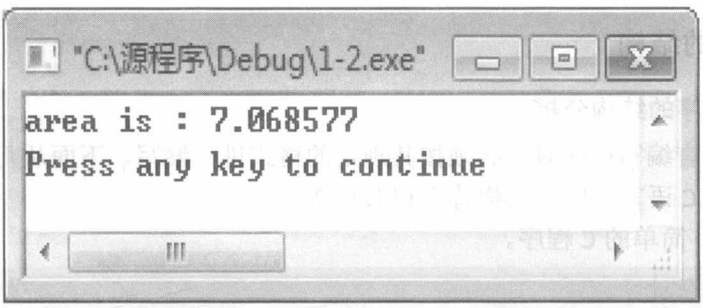


图 1-2　例 1-2 程序运行结果

程序分析如下：

(1) 函数体由声明部分(也称为定义部分)和执行部分组成。

(2) 声明部分是函数体中使用的变量定义及某些函数的声明。

(3) 执行部分由语句构成，用来完成函数所要实现的功能。

【例 1-3】　由多个函数构成的 C 程序。

```
# include <stdio.h>
void hello()          /* 定义函数 hello */
{    printf("*** Hello! How are you? ***\n");
}
void star()           /* 定义函数 star */
{    printf("**************************\n");
}
main()                /* 定义主函数 */
{    star();          /* 调用函数 star */
     hello();         /* 调用函数 hello */
     star();          /* 调用函数 star */
}
```

程序运行结果如图 1-3 所示。

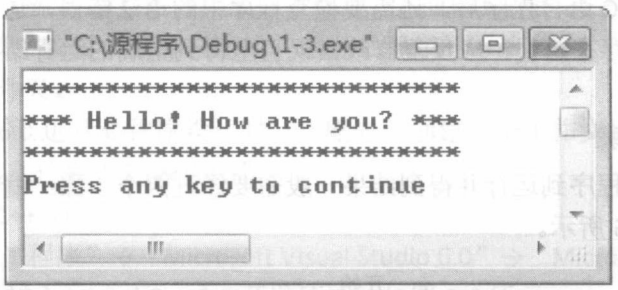


图 1-3　例 1-3 程序运行结果

程序分析如下：

(1) 该程序由主函数 main 及函数名为 hello、star 的函数组成。

(2) 在主函数 main 中使用到了 hello、star 函数，我们称之为调用。关于函数的具体知识将在第 5 单元进行详细介绍。

2. C 语言程序的结构特点

从上面的例子可以总结出 C 程序结构的主要特点：

(1) 函数是组成 C 语言程序的基本单元。编写 C 程序其实就是定义一个个函数。函数构成如图 1-4 所示。

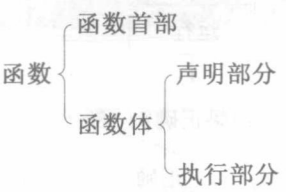


图 1-4　函数构成

(2) 一个 C 程序中有且仅有一个主函数即 main 函数，可以没有或有多个其他函数。主

函数的位置没有特别要求，只要不嵌套定义在其他函数内部。

(3) 一个 C 程序总是从 main 函数开始执行，main 函数执行结束，整个程序即结束。

3. C 语言程序的书写格式

从上面的例子也可以总结出 C 程序的书写格式：

(1) C 程序使用分号“;”作为语句的终止符，复合语句除外。

(2) C 程序书写格式自由，一条语句可以分多行写，一行上也可以写多条语句。

(3) C 语言严格区分大小写。C 程序习惯上用小写字母，但符号常量、宏定义等习惯用大写字母，所有的关键字必须采用小写字母。

(4) 在程序中使用注释对程序进行说明和解释，可增强程序的可读性。注释部分的格式如下：

```
/* 注释内容 */
```

(5) 适当采用缩进格式很有必要。采用缩进格式，可使程序更加清晰、易读。

1.1.2 C 程序的调试与运行

在编写完一个 C 语言程序后，还需要检查程序中的语法错误，运行程序并查看运行结果是否正确，如果存在错误，应反复修改错误直至得到正确的运行结果。这个过程称之为程序的调试与运行。

1. 调试运行步骤

从编写一个 C 程序到运行并得到结果一般需要经过四个步骤：编辑、编译、连接、运行。此过程如图 1-5 所示。

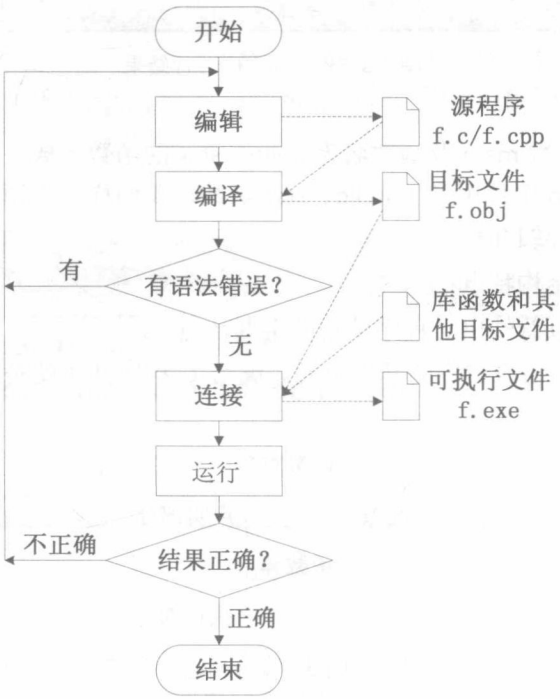


图 1-5 C 程序调试运行步骤

(1) 编辑：将编写的 C 语言源程序输入到计算机中以文件的形式保存起来。C 语言源程序的扩展名为“c”，在 Visual C++ 6.0 集成开发环境中通常将 C 程序保存为扩展名为“cpp”的文件，如“f.c”或“f.cpp”。

(2) 编译：C 语言源程序编辑好之后，应将其编译为二进制的目标代码。在编译时，还会对源程序进行语法检查，如果源程序存在语法错误，用户应该返回编辑状态，根据错误提示信息查找错误并改正，再次重新编译，直到没有语法错误，生成目标程序文件，扩展文件名为“obj”，如“f.obj”。

(3) 连接：将各个模块的二进制目标代码与系统标准模块进行连接处理，得到最终的可执行文件。连接成功将生成可执行文件，扩展文件名为“exe”，如“f.exe”。

(4) 运行：可执行文件没有语法错误，但可能有设计上的错误而导致运行结果不正确。若运行结果不正确，还需要返回到编辑状态查找并修改错误，重新编译、连接、运行，直至运行结果正确。

2. 在 Visual C++ 6.0 集成开发环境中运行 C 程序

C 语言的集成开发环境有 Turbo C、Borland C、Visual C++等。Visual C++ 6.0 功能强大、灵活性好，是目前较为流行的 C 语言集成开发环境。下面以具体的例子为例，介绍在 Visual C++ 6.0 集成开发环境中编辑、编译、连接、运行 C 程序的方法。

(1) 创建工作文件夹。

在用 Visual C++ 6.0 进行 C 程序设计时，一般先要创建一个工作文件夹，以便集中管理和查找文件。如创建文件夹“C:\源程序”。

(2) 启动 Visual C++ 6.0。

选择“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”，运行 Visual C++ 6.0，进入 Visual C++ 6.0 开发环境，如图 1-6 所示。

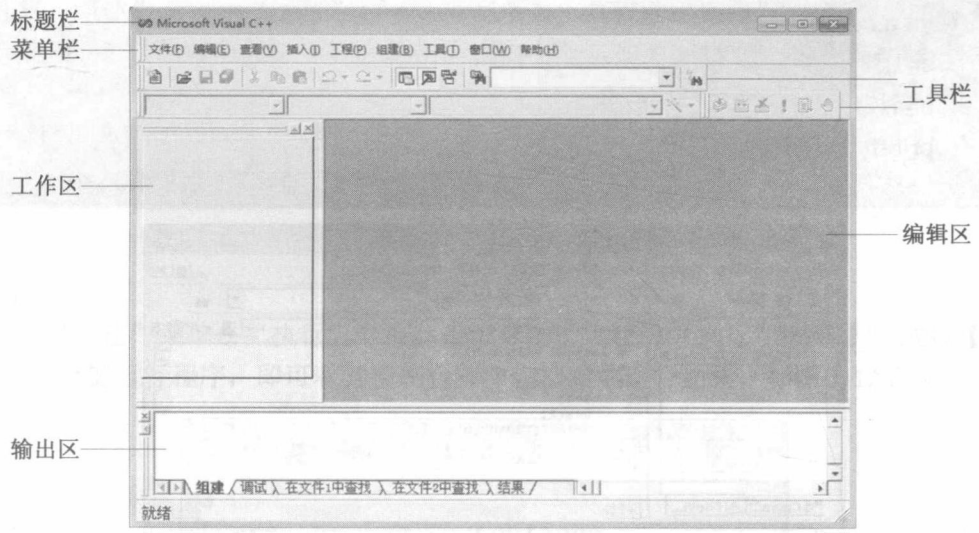


图 1-6　Visual C++ 6.0 开发环境

(3) 创建 C 源程序文件。

选择“文件”菜单→“新建”命令，或直接按【Ctrl】+【N】组合键打开“新建”对话

框。选择“文件”选项卡中的“C++ Source File”，在右边的“文件”文本框中输入文件名，如“1-4”；单击“目录”文本框右侧的 ... 按钮修改保存文件的位置，如“C:\源程序”，如图 1-7 所示；单击“确定”按钮，即可进入 Visual C++ 6.0 的代码编辑区编辑程序。

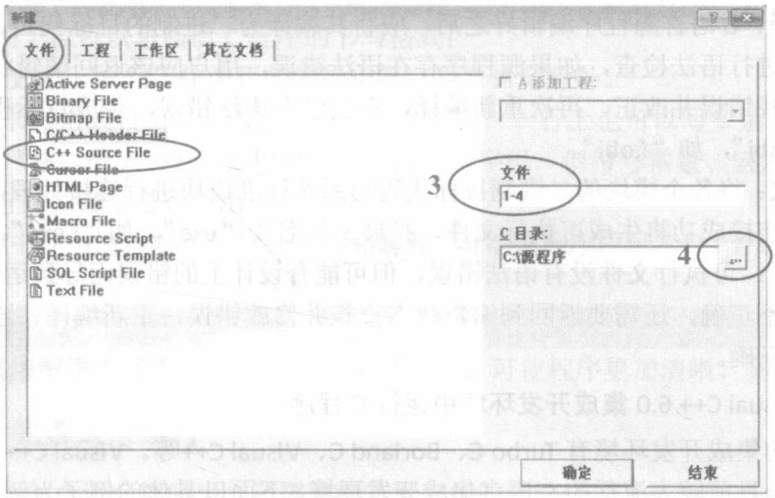


图 1-7　新建 C 程序

(4) 编辑代码并保存。

① 编辑代码：在代码编辑区输入 1-4.cpp 的源代码，完成后如图 1-8 所示。源代码如下：

【例 1-4】　编辑、编译、连接、运行如下 C 程序。

```
# include <stdio.h>
main()
{    int a,b;
     a=1;
     b=a+2;
     printf("%d\n",b);
}
```

图 1-8　在编辑区编辑源程序

② 保存源程序文件：选择“文件”菜单→“保存”命令(“另存为”命令可修改文件名或文件存放位置)，也可单击工具栏中的“保存”按钮 来保存文件。

(5) 编译。

选择“组建”菜单(由于汉化版本不同，有的版本此处为“编译”菜单)→“编译”命令，也可单击工具栏的“编译”按钮，或按【Ctrl】+【F7】组合键，在弹出的“创建默认项目工作空间”对话框中选择“是”按钮，如图 1-9 所示，然后系统开始对当前的源程序进行编译。

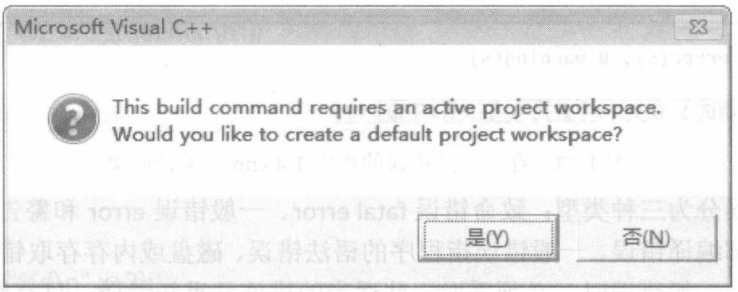


图 1-9　“创建默认项目工作空间”对话框

若编译过程中没有发现语法错误，则在输出区窗口中显示如图 1-10 所示的编译信息。

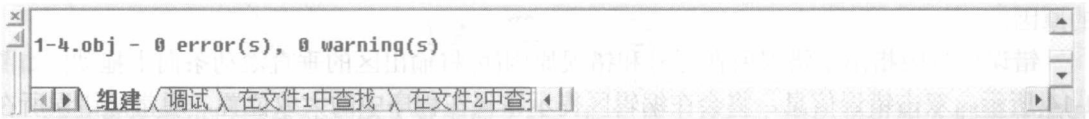


图 1-10　输出区窗口中的编译信息

(6) 连接。

选择“组建”菜单→“组建”命令，也可单击工具栏的“连接”按钮，或按【F7】键进行连接。若连接没有错误，则在输出区窗口中显示如图 1-11 所示的连接信息。

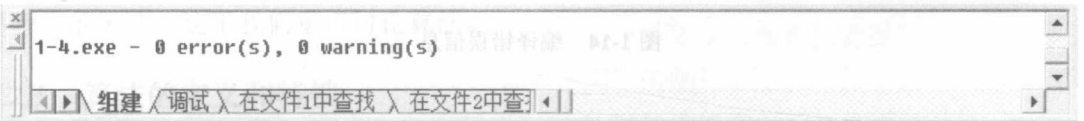


图 1-11　输出区窗口中的连接信息

(7) 运行。

选择“组建”菜单→“执行”命令，也可单击工具栏的“执行”按钮，或按【Ctrl】+【F5】组合键运行程序，即可看到控制台程序窗口中的运行结果，如图 1-12 所示。

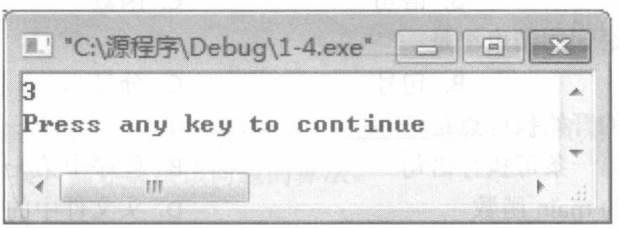


图 1-12　程序 1-4.cpp 的运行结果

(8) 关闭工作空间。

完成了对程序的操作后，应保存好已经建立的程序与数据，并关闭工作空间。选择“文件”菜单→“关闭工作空间”命令，可关闭工作空间。

关闭工作空间后，可重复以上步骤，进行其他程序文件的操作。

3. 编译错误的处理

若在编译过程中系统发现程序存在编译错误，会将错误信息、显示在输出区窗口中，如图 1-13 所示。

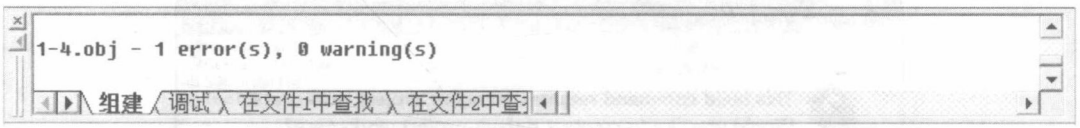


图 1-13　存在语法错误的程序 1-4.cpp 的编译结果

源程序错误分为三种类型：致命错误 fatal error、一般错误 error 和警告 warning。致命错误通常是内部编译错误。一般错误指程序的语法错误、磁盘或内存存取错误或命令错误。发生致命错误及一般错误时，必须采取一些适当的措施并重新编译，只有修改这两类错误后，程序才能继续连接、运行。警告并不阻止编译进行，它指出一些值得怀疑的情况，可以说 warning 是编译器为程序员提供的友善建议和意见，我们应当认真查看产生 warning 的原因。

错误信息中指出了错误所在行号和错误原因(可将输出区的垂直滚动条向上拖动)，如图 1-14 所示。双击错误信息，将会在编辑区提示错误在程序中的大致位置，通常是错误所在的行或附近。查找出错误，修改后重新编译直至成功，方可连接、运行。

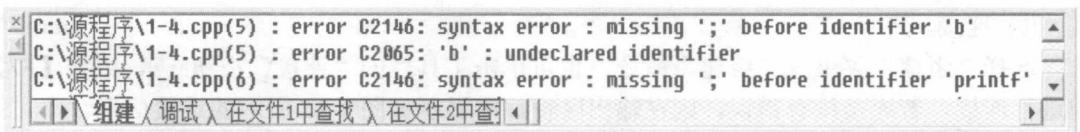


图 1-14　编译错误信息

习　题

1. 选择题

(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) C 语言程序语句的结束符是_____。

A. 逗号　　B. 句号　　C. 分号　　D. 括号

(3) C 语言程序的开始执行点是_____。

A. 程序中第一条可执行语句　　B. 程序中第一个函数

C. 程序中的 main 函数　　D. 头文件中的第一个函数

(4) 完成 C 源程序编辑后，到生成执行文件，必须经过的步骤依次为_____。

A. 连接、编译　　B. 编译、连接　　C. 连接、运行　　D. 运行

(5) C 语言源程序连接后，产生文件的扩展名是_____。

A. c　　B. obj　　C. exe　　D. h

2. 写出以下程序的运行结果

(1)
```
# include <stdio.h>
main()
{  printf("I love China!\n");
   printf("We are students.\n");
}
```

(2)
```
# include <stdio.h>
main()
{  int x;
   x=10;
   printf("%d\n",x+2);
}
```

1.2 算　法

计算机尽管可以完成许多极其复杂的工作，但实质上这些工作都是按照人们事先编好的程序进行的，所以人们常把程序称为计算机的灵魂。有这样一个著名的公式：

程序 = 算法 + 数据结构

这个公式说明：对于面向过程的程序设计语言而言，程序由算法和数据结构两大要素构成。其中，数据结构是指数据的组织和表示形式；而算法就是进行操作的方法和步骤，是程序的灵魂。这里我们重点讨论算法。

1.2.1 算法的定义和特性

1. 算法的定义

算法是为解决一个具体问题而采用的确定、有效的方法和操作步骤。

【例 1-5】　已知圆的半径为 10，求其面积。描述解决该问题的算法。

算法描述如下：

(1) 半径 r=10；

(2) 面积 $s=\pi r^2$；

(3) 输出面积 s，结束。

【例 1-6】　有三只杯子：A 杯中装满酒，B 杯中装满醋，C 杯是空杯。利用 C 杯将 A 杯与 B 杯装的东西对换。描述解决该问题的算法。

算法描述如下：

(1) A 杯倒入 C 杯；