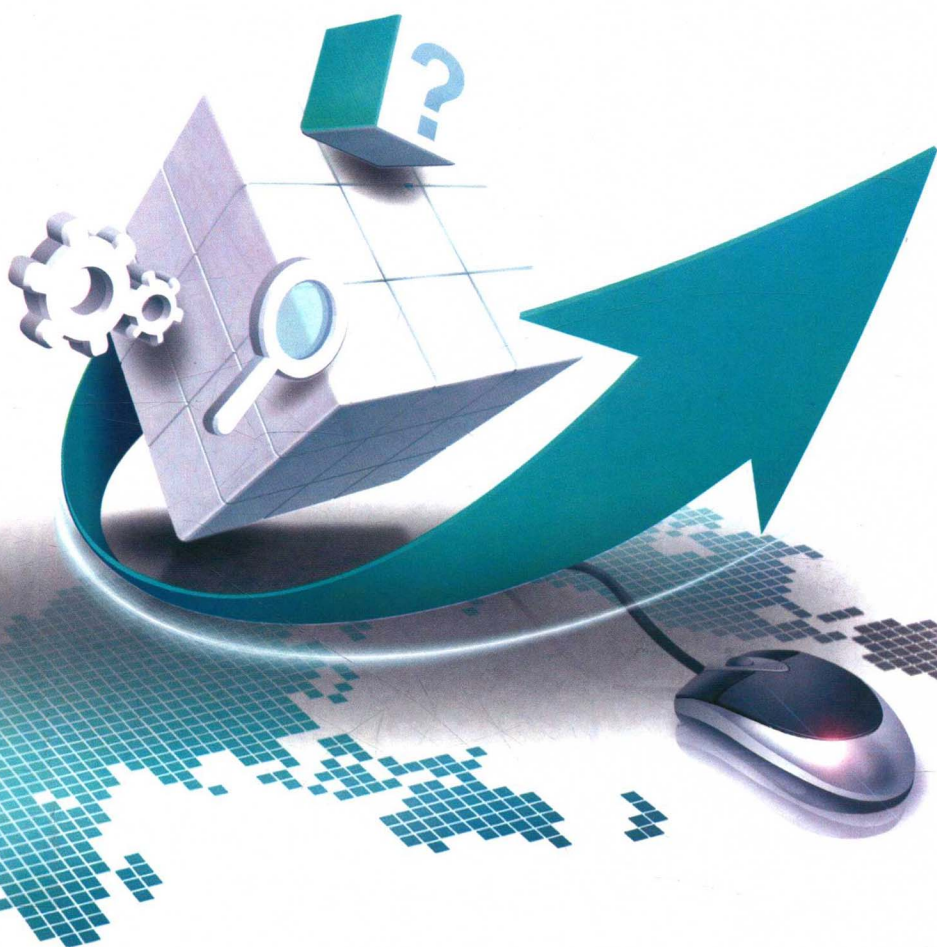


高等学校电子信息类“十三五”规划教材

C 语言实践与提高

夏海英 梁艳 宋树祥 ◆ 编著



西安电子科技大学出版社
<http://www.xduph.com>

高等学校电子信息类“十三五”规划教材

C语言实践与提高

夏海英 梁 艳 宋树祥 编著

西安电子科技大学出版社

★ 内容简介 ★

本书包括 VC 环境下 C 程序的运行与调试、数据描述与顺序程序设计、选择程序设计、循环程序设计、数组程序设计、函数程序设计、指针程序设计、结构体程序设计、文件程序设计共 9 章内容。除第 1 章外,其余各章均包括实践目的、程序运行结果分析、程序填空题、程序设计题、程序提高题和小结等主要部分。在程序设计题部分,本书给出了详细的算法提示和编程思路。

本书可作为电子信息类专业本科生的 C 语言实践指导书。

图书在版编目(CIP)数据

C 语言实践与提高/夏海英,梁艳,宋树祥编著. —西安:西安电子科技大学出版社,2017.9
(高等学校电子信息类“十三五”规划教材)

ISBN 978-7-5606-4673-2

I. ① C… II. ① 夏… ② 梁… ③ 宋… III. ① C 语言—程序设计 IV. ① TP312.8

中国版本图书馆 CIP 数据核字(2017)第 210894 号

策 划 姚 磊

责任编辑 阎 彬

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 2017 年 9 月第 1 版 2017 年 9 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 9.5

字 数 219 千字

印 数 1~1000 册

定 价 24.00 元

ISBN 978 - 7 - 5606 - 4673 - 2/TP

XDUP 4965001-1

如有印装问题可调换



前 言

本书属于实验实践指导用书，面向电子信息类专业的本科生，可作为“C 语言程序设计”课程实践教学活动的活动用书。

市面上，C 语言程序设计相关书籍非常丰富，有 C 语言程序设计课堂教学用书系列、C 语言能力提升系列、C 语言实验教材系列等。这些书籍各有各的特点，基本涵盖了 C 语言学习的方方面面，但能够在 C 语言基本编程能力训练与能力提升训练间有效衔接的书籍比较少。编写本书最初的想法来源于编者所讲授的“C 语言实践”这门课程。这门课程是电子信息工程专业非常重要的一门选修课，包含 36 个课时，目的是提升学生的 C 语言编程能力，为后续学习电子信息工程的各门课程打下良好基础。为了充分调动学生学习的主动性，编者认真总结了现有的各种竞赛和创新实践项目，鼓励学生参加蓝桥杯全国软件和信息技术专业人才大赛，并对相应的 C 语言的知识点做培训。在选择培训教材时，编者发现现有教材很难满足教学要求。因此，编者开始自己总结“C 语言实践”课程教学的内容，并经过 2013—2016 年这 4 年的不断修正与增减，形成此书。

本书以 Microsoft Visual Studio 2008 作为 C 语言程序开发环境，包括 VC 环境下 C 程序的运行与调试、数据描述与顺序程序设计、选择程序设计、循环程序设计、数组程序设计、函数程序设计、指针程序设计、结构体程序设计、文件程序设计共 9 章内容。除第 1 章外，其余各章均包括实践目的、程序运行结果分析、程序填空题、程序设计题、程序提高题和小结等主要部分。在程序设计题部分，本书给出了详细的算法提示和编程思路，可使初学者快速掌握 C 语言程序设计的方法。这些内容涉及的是 C 语言的基础内容，要求学生必须熟练掌握。每个章节的实践提高内容，大多选自国内外程序设计大赛中的真

题，对提高学生编程思维能力是个很好的挑战。

在内容的选择上，本书参考了以往的 C 语言实验训练内容，也融入较多的各类程序设计大赛中新的题型，包含了基本 C 语言实验教学内容和 C 语言能力提升训练两部分。因此，在学习本书的过程中，建议从程序分析题开始，逐渐深入，以达到好的学习效果。

夏海英

2017 年 7 月

目 录

第 1 章 VC 环境下 C 程序的运行与调试.....	1
1.1 Microsoft Visual Studio 2008 运行环境介绍.....	1
1.2 C 程序开发	1
1.2.1 新建一个 C 程序	2
1.2.2 程序的编译与运行	3
1.3 C 程序调试	4
1.3.1 设置断点	5
1.3.2 调试命令	5
1.3.3 查看变量	6
1.4 小结	6
习题一	7
第 2 章 数据描述与顺序程序设计	8
2.1 实践目的	8
2.2 程序运行结果分析	8
2.3 程序填空题	11
2.4 程序设计题	14
2.5 程序提高题	16
2.6 小结	18
习题二	18
第 3 章 选择程序设计	20
3.1 实践目的	20
3.2 选择程序运行结果分析	20
3.3 程序填空题	25
3.4 程序设计题	28
3.5 程序提高题	32
3.6 小结	32
习题三	32

第4章 循环程序设计	34
4.1 实践目的	34
4.2 循环程序运行结果分析	34
4.3 程序填空题	38
4.4 程序设计题	43
4.5 程序提高题	48
4.6 小结	54
习题四	54
第5章 数组程序设计	56
5.1 实践目的	56
5.2 数组程序运行结果分析	56
5.3 程序填空题	60
5.4 程序设计题	68
5.5 程序提高题	72
5.6 小结	78
习题五	78
第6章 函数程序设计	79
6.1 实践目的	79
6.2 函数程序运行结果分析	79
6.3 程序填空题	84
6.4 程序设计题	91
6.5 程序提高题	99
6.6 小结	106
习题六	106
第7章 指针程序设计	108
7.1 实践目的	108
7.2 指针程序运行结果分析	108
7.3 程序填空题	113
7.4 程序设计题	120
7.5 程序提高题	123
7.6 小结	129
习题七	129
第8章 结构体程序设计	130
8.1 实践目的	130
8.2 结构体程序运行结果分析	130

8.3 程序填空题	131
8.4 程序设计题	133
8.5 程序提高题	135
8.6 小结	137
习题八	137
 第 9 章 文件程序设计	 138
9.1 实践目的	138
9.2 文件程序运行结果分析	138
9.3 程序填空题	140
9.4 程序设计题	142
9.5 程序提高题	143
9.6 小结	143
习题九	143

第 1 章

VC 环境下 C 程序的运行与调试

1.1 Microsoft Visual Studio 2008 运行环境介绍

Microsoft Visual Studio 2008(简称 VC9)是面向 Windows Vista、Office 2007、Web 2.0 的开发工具，为我们提供了一个集程序创建、编辑、编译、调试等诸多功能于一体的集成开发环境(IDE)。VC9 集成开发环境功能强大，不仅提供了大量的向导(Wizard)，还有完备的帮助功能(MSDN)。所以，初学者在学习 C 语言编程时，并不需要全面了解开发环境的全部功能，可以在安装 VC9 时选择完全安装 MSDN，然后在遇到问题时再去查阅 MSDN 中的相关说明。

通过“开始”菜单或桌面快捷方式启动 Visual C++ 进入集成开发环境，如图 1-1 所示。

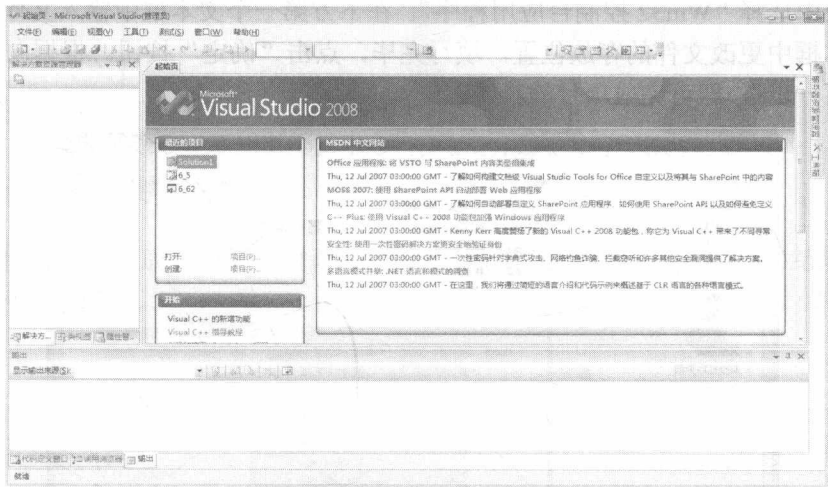


图 1-1 VC9 集成开发环境

跟 VC6.0 不同，VC9 支持 Visual C++、C#、VB 等语言的程序开发。首次启动 VC9，会有一个选择编译环境的界面，选择 Visual C++ 即可。

1.2 C 程序开发

一个完整的 C 程序有四个开发步骤：编辑、编译、连接和运行程序。

如果经过测试，可执行文件运行后达到了预期的设计目的，这个 C 语言程序的开发工作便到此完成了。如果运行出错，说明程序处理的逻辑存在问题，需要再次回到编辑环境，针对程序出现的逻辑错误进一步检查、调试、修改源程序，重复编辑→编译→连接→运行的过程，直到取得预期结果为止。

1.2.1 新建一个 C 程序

选择下拉菜单：文件→新建→项目，如图 1-2 所示。

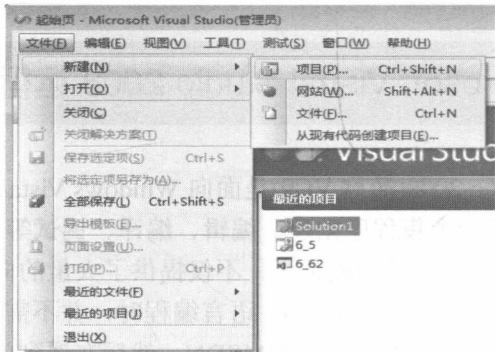


图 1-2 文件的新建

点击“项目”选项，弹出如图 1-3 所示对话框，在左边项目类型框中选择“Win32”，在右边模板中选择“Win32 控制台应用程序”，在下方第一个文本框中输入文件的名称，在第二个文本框中更改文件的存放位置，填写完毕，点击“确定”按钮。根据向导提示，选择两次“下一步”，即可完成项目的新建。

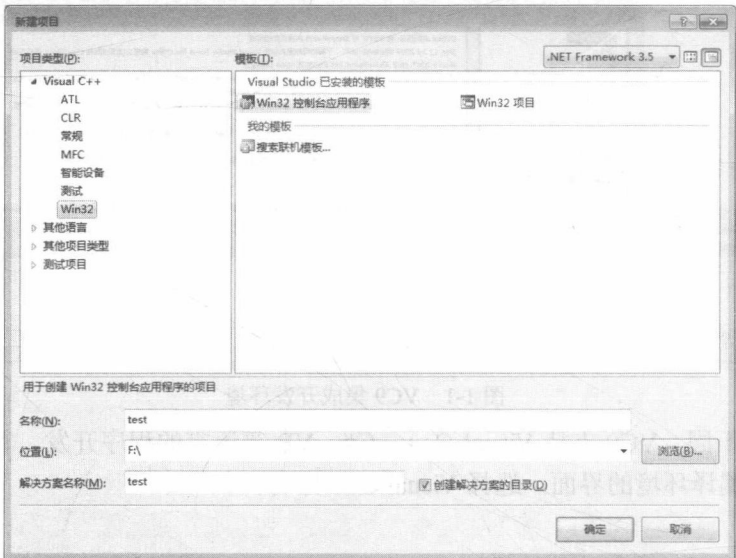


图 1-3 新建项目

项目新建完成后，界面如图 1-4 所示。双击左边“解决方案”中对应的后缀名为 .cpp 的文件，即可进入文件的编辑界面。程序的编辑可在 test.cpp 中完成。

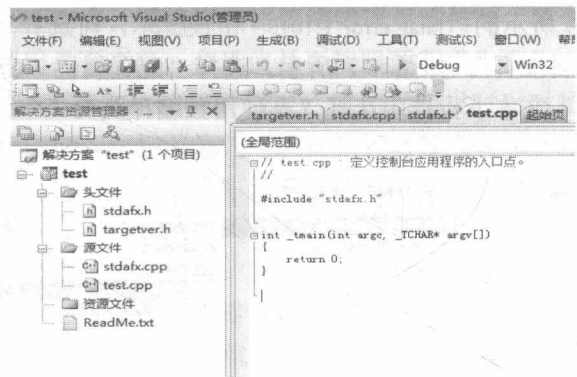


图 1-4 文件编辑界面

如果喜欢 VC6 下的编辑风格，可将左边“解决方案”中除了 test.cpp 文件外的其他文件全部删除(选中文件，按下“Delete”键即可完成文件删除)。双击 test.cpp，清空其编辑界面，根据程序需求，编辑完整的 C 程序代码，如图 1-5 所示。

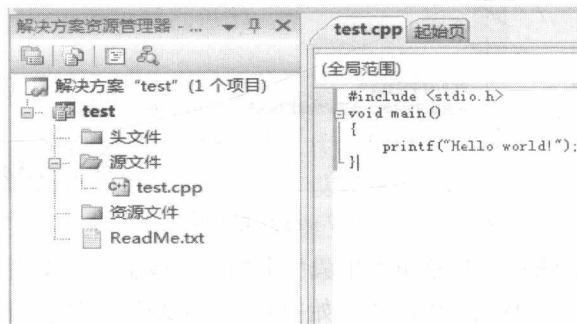


图 1-5 一个完整的 C 程序代码

1.2.2 程序的编译与运行

选择下拉菜单“生成”中的“编译”，即可完成 C 程序的编译。编译的结果显示在输出框中，如图 1-6 所示。

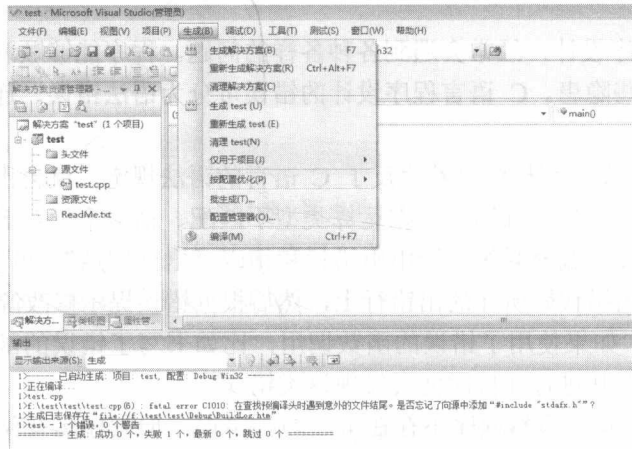


图 1-6 程序的编译

如果出现如图 1-6 所示的“fatal error C1010”错误提示,可选择下拉菜单“项目”中的“属性”,打开“test 属性页”,如图 1-7 所示。选择“配置属性→C/C++→预编译头”,在“创建/使用预编译头”文本下拉框中选择“不使用预编译头”。

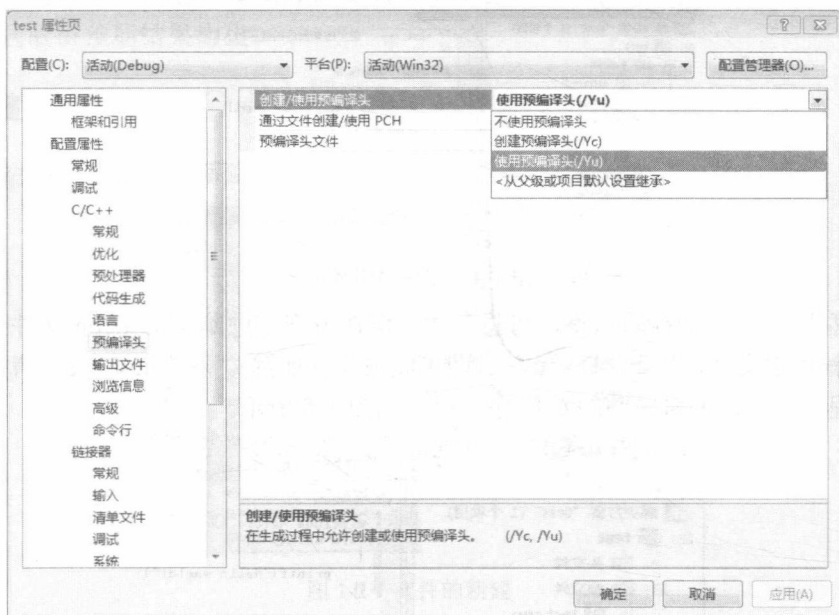


图 1-7 test 属性页

程序编译成功后,选择下拉菜单“生成”中的“生成解决方案”,即可完成程序的编译与连接。选择下拉菜单“调试”中的“开始执行(不调试)”,即能看到程序的运行结果。

除了以上的操作外,点击快捷工具栏中的  按钮,也可以实现程序的编译、连接与运行。

1.3 C 程序调试

初学 C 语言程序设计,经常会遇到各种各样的问题。程序能够顺利运行,并非大功告成,也可能存在某些隐患。C 语言程序设计的错误可分为语法错误、连接错误、逻辑错误和运行错误。

(1) 语法错误。在编写程序时若违反了 C 语言的语法规则,就会出现语法错误。语法不正确、关键词拼错、标点漏写、数据运算类型不匹配、括号不配对等都属于语法错误,在进入程序编译阶段,编译系统会给出出错行和相应“出错信息”。可以双击错误提示行,将光标快速定位到出错代码所在的出错行上,然后根据错误提示修改源程序,排除错误。

(2) 连接错误。如果使用了错误的函数调用,比如书写了错误的函数名或不存在的函数名,编译系统在对其进行连接时便会发现这类错误。

(3) 逻辑错误。有时虽然程序不存在上述两种错误,但程序运行结果就是与预期效果不符。这就可能是逻辑错误了。产生逻辑错误往往是因为程序采用的算法有问题,或编写

的程序逻辑与算法不完全吻合。逻辑错误比语法错误更难排除,需要程序员对程序逐步调试,检测循环、分支调用是否正确,变量值是否按照预期产生变化。

(4) 运行错误。程序不存在上述三种错误,但运行结果时有时错,这就是运行错误。运行错误往往是由于程序的容错性不高,可能在设计时仅考虑了一部分数据的情况,对于其他数据就不能适用了。例如打开文件时没有检测打开是否成功就开始对文件进行读写,结果程序运行时,如果文件能够顺利打开,程序运行正确,反之则程序运行出错。要避免这种类型的错误,需要对程序反复测试,完备算法,使程序能够适应各种情况的数据。

为了方便程序员排除程序中的逻辑错误,VC9 提供了强大的调试功能。每当我们创建一个新的 VC9 工程项目时,默认状态就是 Debug(调试)版本。调试版本会执行编译命令 `_D_DEBUG`,将头文件的调试语句 `ifdef` 分支代码添加到可执行文件中;同时加入的调试信息可以让开发人员观察变量,单步执行程序。由于调试版本包含了大量信息,所以生成的 Debug 版本可执行文件容量会远远大于 Release(发行)版本。

当出现逻辑错误或者运行错误的时候,就需要进行调试了,以下是具体的调试步骤。

1.3.1 设置断点

VC9 可以在程序中设置断点,跟踪程序实际执行流程。设置断点后,可以按“F5”功能键启动 Debug 模式,程序会在断点处停止。接着可以单步执行程序,观察各变量的值如何变化,确认程序是否按照设想的方式运行。设置断点的方法是:将光标停在要被暂停的那一行代码上,点击右键,选择“断点”→“插入断点”(或者直接单击要暂停的那行的最左边一列),如图 1-8 所示,断点所在代码行的最左边出现了一个深红色的实心圆点,这表示断点设置成功。

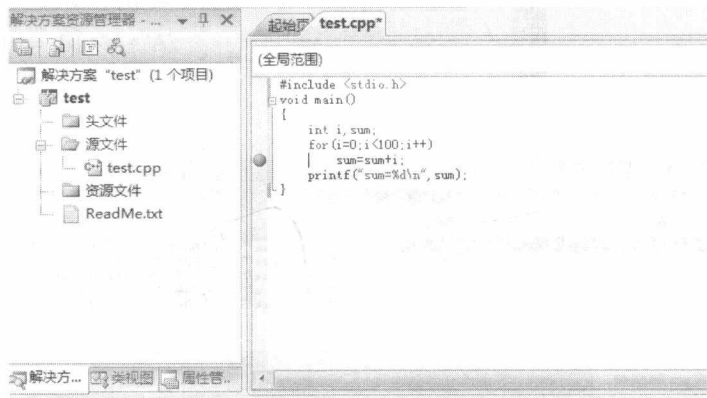


图 1-8 断点的设置

若想取消某行的断点,则把光标放在此行,点击右键,选择“断点”→“删除断点”,即可删除该行的断点(或直接单击实心圆点)。

1.3.2 调试命令

选择下拉菜单“调试”中的“启动调试”,程序运行至断点处停止。调试命令的快捷键如下:

F5: 从当前语句开始运行程序，直到程序结束或至断点处；

F10: 单步执行，跳过所调用的函数；

F11: 单步执行，跟踪所调用的函数；

Ctrl + F10: 运行程序至光标所在行；

Shift + F5: 终止调试，程序返回编辑状态。

在调试模式下，程序停止在某条语句，该条语句左边就会出现一个黄色的小箭头。单步执行可随时查看变量，检查调用情况。比如，按“F5”功能键进入调试模式，程序运行到断点处暂停；不断按“F10”功能键，就可以一行一行地执行程序，直到程序运行结束。

1.3.3 查看变量

单步调试程序的过程中，可以在下方的局部变量窗口和监视窗口动态地查看变量的值，如图 1-9 所示。局部变量窗口自动显示当前运行上下文中的各个变量的值，而监视窗口内只显示在此监视窗口输入的变量或表达式的值。随着程序的逐步运行，也可以直接用鼠标指向程序中的变量查看其值。

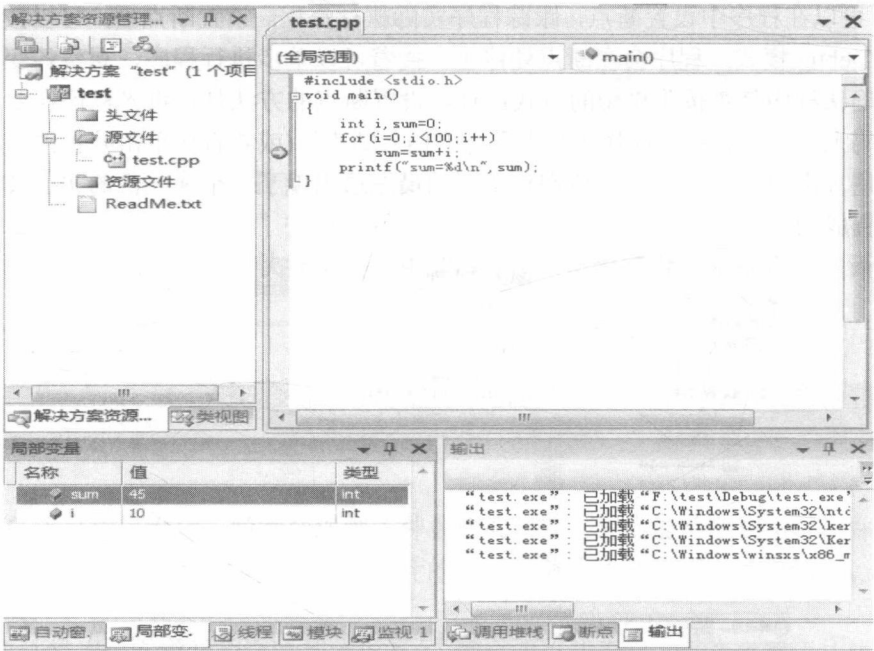


图 1-9 查看变量

1.4 小 结

本章首先通过介绍 VC9 的运行环境，说明了如何创建一个 C 程序，同时对第一个 C 程序进行了解释分析，演示了一个 C 程序运行必须经过的四个步骤：编辑、编译、连接和

执行。程序编译和运行中，初学者总会遇到各种各样的问题，因此本章还简要介绍了 C 程序错误的种类及解决方法，详细说明了程序调试中设置断点、单步执行、查看变量等操作。本章为读者学习并理解 C 程序设计与运行、熟悉开发环境奠定了基础。

习 题 一

1. 在 VC9 集成开发环境中编辑、编译、连接并执行输出“Hello World!”程序。
2. 设计一个 C 程序，可实现 100 以内奇数的求和。
3. 设计一个 C 程序，输入一个整数，判断其是否是素数。
4. 设计一个 C 程序，输入 10 个整数，计算正数的和与平均值。
5. 下面有一段程序，可将字符串中的小写字母转换为大写字母。程序中有两处错误，请调试改正之，但不得增删语句。

```
#include<stdio.h>
int main(void)
{
    char s[255];
    int i;
    scanf("%s",s);
    for (i=0; s[i];i++)
        if(s[i] >= 'b' && s[i] <= 'z')
            s[i] = s[i] + 'a' - 'A';
    printf("%s",s);
    return 0;
}
```

第2章

数据描述与顺序程序设计

2.1 实践目的

- (1) 进一步掌握运行一个 C 语言程序的方法和步骤。
- (2) 清楚 C 语言的运算符、标识符的区别。
- (3) 掌握 C 语言的数据类型，能够定义整型、实型、字符型变量，熟悉变量的赋值方法。
- (4) 熟练掌握数据输入的方式、数据输出的格式及各种数据输入或输出控制格式的含义。
- (5) 学会使用 C 的运算符以及用这些运算符组成的表达式，特别是自加(++)和自减(--)运算符的使用。

2.2 程序运行结果分析

- (1) 输入并运行下面的程序，分析其运行结果。

```
#include <stdio.h>

int main()
{
    char c1,c2;  c1=46;c2=47;
    printf("%3c%3c", c1,c2);
    printf("%3d%3d", c1,c2);
    return 0;
}
```

将程序第二行改为：int c1,c2; 再运行，分析其结果。

注：本例体现出 C 语言的灵活性，即整型变量与字符型变量可以相互转换。

- (2) 输入并运行下面的程序，分析其运行结果，特别注意输出 c1, c2 的值是什么，并思考为什么。

```
#include <stdio.h>

int main( )
{
    int a,b; float c,d; long e,f; unsigned int u,v; char c1,c2;
    scanf("%d,%d",&a,&b);
    scanf("%f,%f", &c,&d);
    scanf("%ld,%ld",&e,&f);
    scanf("%o,%o",&u,&v);
    scanf("%c,%c", &c1,&c2);
    printf("\n");
    printf("a=%4d,b=%4d\n",a,b);
    printf("c=%8.2f,d=%8.2f\n",c,d);
    printf("e=%16ld,f=%16ld\n",e,f);
    printf("u=%o,v=%o\n",u,v);
    printf("c1=%c,c2=%c\n",c1,c2);
    return 0;
}
```

(3) 先分析下面程序的运行结果，然后再上机运行，看结果是否一致。

```
#include <stdio.h>

int main( )
{
    int x,y,z;
    x=y=z=3;
    y=x++-1; printf("%4d%4d", x, y);
    y=++x-1; printf("%4d%4d", x, y);
    y=z--+1; printf("%4d%4d", z, y);
    y=--z+1; printf("%4d%4d", z, y);
    return 0;
}
```

(4) 分析以下程序，写出运行结果。

```
#include <stdio.h>

int main( )
{
    char c1='a',c2='b',c3='c',c4='\101',c5='\116';
    printf("a%c b%c\tabc\n",c1,c2,c3);
    printf("\t\b%c %c",c4,c5);
    return 0; }
```