

上海市教育委员会重点课程建设项目

王明衍 主编
洪 菁 胡建鹏 副主编

C 语言程序设计 实验指导书

C Language Programming Experiment Instructions



科学出版社

上海市教育委员会重点课程建设项目

C 语言程序设计实验指导书

王明衍 主编

洪 菁 胡建鹏 副主编



科学出版社

北 京

内 容 简 介

学习 C 语言程序设计课程不能满足于看懂书上的程序,而应当熟练地掌握程序设计的全过程,即独立编写源程序、独立上机调试、独立运行程序和分析结果。本书从指导课程教学、学习和考试的角度,以程序设计为主线,由范例和问题引入内容,由浅入深,使读者掌握程序设计的基本方法并逐步形成正确的程序设计思想,能够使用 C 语言进行程序设计并具备调试程序的能力。

本书共由 10 个实验组成。每个实验都提供了精心选择的范例,通过分析问题、讲解编程思路、解析常用算法和关键程序段示例,使读者逐步掌握程序设计的全过程。这对提高读者分析问题的能力、理解基本概念、开拓解题思路、全面提高程序设计能力会起到良好的效果。

本书可以作为 C 语言程序设计课程的实验教材,也可以作为计算机等级考试人员以及各种程序设计培训班学员的参考书。

图书在版编目(CIP)数据

C 语言程序设计实验指导书 / 王明衍主编. —北京: 科学出版社, 2012.3
ISBN 978-7-03-033361-2

I. ①C… II. ①王… III. ①C 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 006812 号

责任编辑: 潘志坚 王艳丽 / 责任校对: 陈玉凤

责任印制: 刘 学 / 封面设计: 殷 靓

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

江苏省句容市排印厂印刷

科学出版社编务公司排版制作

科学出版社发行 各地新华书店经销

*

2012 年 3 月第 一 版 开本: B5 (720 × 1000)

2012 年 3 月第一次印刷 印张: 8

字数: 138 000

定价: 25.00 元

(如有印装质量问题, 我社负责调换)

前 言

C 语言是一种编程灵活、特色鲜明的程序设计语言，是高等院校首选的计算机语言基础课程，学好这种语言可以为后续的面向对象的语言学习打好坚实的基础，C 语言除了学习必要的基本语法、算法外，更重要的是进行实际操作训练，使学习者掌握程序设计的基本方法并逐步形成正确的程序设计思想，提高使用 C 语言进行程序设计的能力，同时具备调试程序的能力。结合多年来的教学经验，根据学生的学习情况，为配合教学过程，使基于案例驱动的教学模式能在根本上促进学生有更大进步，特编写本书，以指导学生的上机操作。

本书由 10 个实验组成。每个实验都提供了精心选择的范例，通过分析问题、讲解编程思路、解析常用算法和关键程序段示例，使逐步读者掌握程序设计的全过程。

读者在实验前后应该完成以下四个部分的任务。

一、明确实验的目的

上机实验的目的，绝不仅仅是为了验证教材和讲课的内容，或者验证自己所编写的程序的正确与否。程序设计课程上机实验的目的是：

1. 加深对讲授内容的理解，尤其是一些语法规则。通过实验来掌握语法规则是行之有效的方法。
2. 熟悉所用的操作系统。
3. 学会上机调试程序。通过反复调试程序掌握根据出错信息修改程序的方法。
4. 通过调试完善程序。

二、做好实验前的准备工作

1. 了解所用的计算机系统(包括 C 编译系统)的性能和使用方法。
2. 复习和掌握与本实验有关的教学内容。
3. 准备好上机所需的程序，切忌不编程或抄别人的程序去上机。
4. 对程序中出现的应事先估计，对程序中自己有疑问的地方应先作上记

号，以便上机时给予注意。

5. 准备好调试程序和运行程序所需的数据。

三、确定实验的步骤

上机实验应该力求独立完成。上机过程中出现的问题，除了系统的问题以外，不要轻易举手问老师。尤其对“出错信息”，应善于分析判断，找出出错的行，然后检查该行或其上一行。

四、写好实验报告

实验报告应包括以下内容：

1. 实验目的。
2. 实验内容。
3. 程序清单。
4. 运行结果。
5. 对运行结果的分析，以及本次调试程序所取得的经验。

本书由上海工程技术大学电子电气学院计算中心王明衍、洪菁、胡建鹏编著。在本书的编写过程中，上海工程技术大学的赵毅、陈强、赵卫东、黄容和潘勇等对实验内容的选择与审定给予了很大的帮助，提出了一些非常宝贵的意见和建议，在此表示衷心的感谢。

本书的编著得到 2010 年上海市教育委员会重点课程建设项目的资助。

由于编者水平有限，书中的不足之处，恳请有关专家和读者批评指正。

编 者

2011 年 10 月 30 日于上海工程技术大学

目 录

前言

实验 1 熟悉 C 语言集成开发环境	1
实验 2 C 语言的三种程序结构	11
2.1 顺序结构	11
2.2 简单分支结构	15
2.3 简单循环结构	20
实验 3 分支程序设计	26
3.1 简单分支结构	26
3.2 复杂分支结构	33
实验 4 循环结构程序设计	41
4.1 基本循环语句的使用	41
4.2 嵌套循环	47
实验 5 程序结构的综合练习	55
实验 6 数组	61
6.1 一维数组	61
6.2 字符串和二维数组	68
实验 7 函数	77
实验 8 指针的应用	83
实验 9 结构体和共用体	93
实验 10 文件操作	104
参考文献	113
附录 A C 语言基本语法	114
附录 B ASCII 编码表	122

实验 1 熟悉 C 语言集成开发环境

【实验目的】

1. 熟悉 Visual C++6.0 开发环境。
2. 掌握 C 语言程序的编辑、调试及运行的过程和方法。
3. 熟悉 Visual C++6.0 集成开发环境中常见的错误信息提示。

【实验指导】

1. Visual C++6.0 开发环境

一个 C 语言源程序总是在一定的硬件和软件环境支持下通过编辑、编译、连接和运行四个过程完成的，并且分别生成源程序文件(.cpp)、目标文件(.obj)和可执行文件(.exe)。具体操作步骤如下：

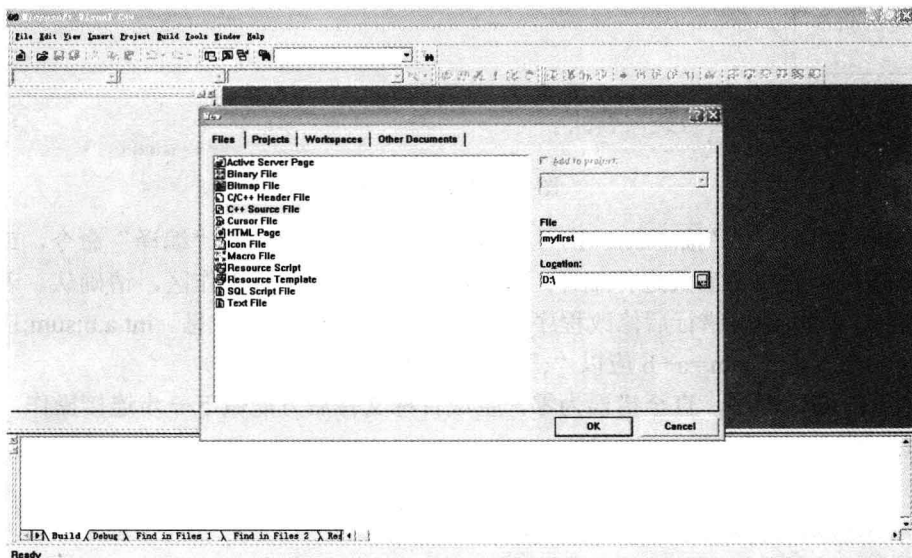


图 1.1 新建文件

(1) 双击桌面 Visual C++ 快捷方式进入 Visual C++ 开发环境, 或通过执行“开始=>程序=> Microsoft Visual C++6.0 进入。

(2) 单击“文件”菜单中的“新建”命令。

(3) 在打开的“新建”对话框中选择“文件”标签。

(4) 选择 C++ Source File, 选择文件保存位置, 然后在文件输入栏中输入文件名。如图 1.1 所示。

(5) 输入和编辑源程序。如图 1.2 所示。(提示: 注意“;”分号是表示一条 C 语句的结束, 不可缺少的, 而且必须在西文输入状态下输入。/**/是对语句的注释, 与程序运行无关, 可以不输入)

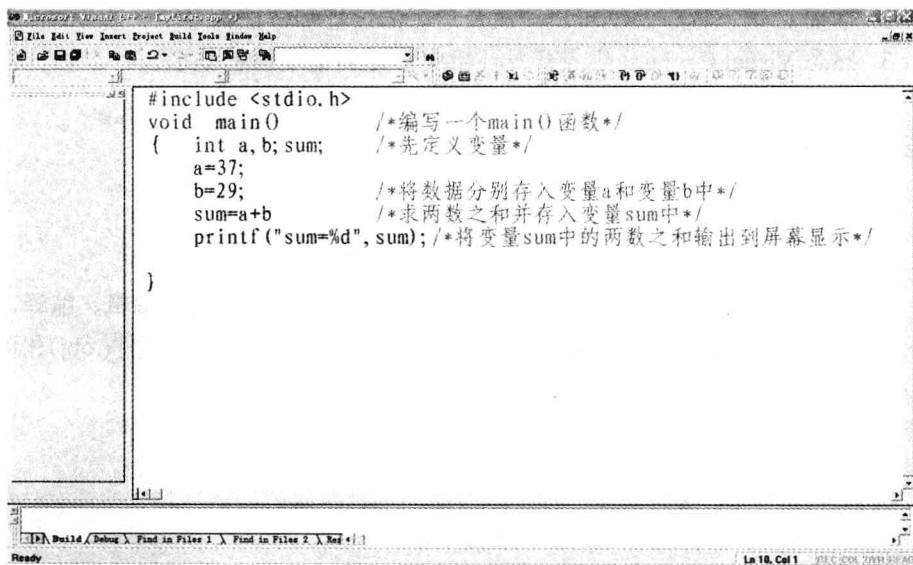


图 1.2 编辑源文件

(6) 编译程序。按“Ctrl+F7”或通过“编译”菜单中的“编译”命令, 或使用工具栏中的相应工具进行编译, 系统提示创建默认项目工作区, 请确认。若程序有错, 则找到出错行后修改程序。(提示: 注意编译出错信息。int a,b,sum;应修改为 int a,b,sum;sum=a+b 应以“;”结束, 修改为 sum=a+b;)

重新编译程序, 直至错误为零, 生成目标文件后方能做下一步连接操作, 如图 1.3 所示。

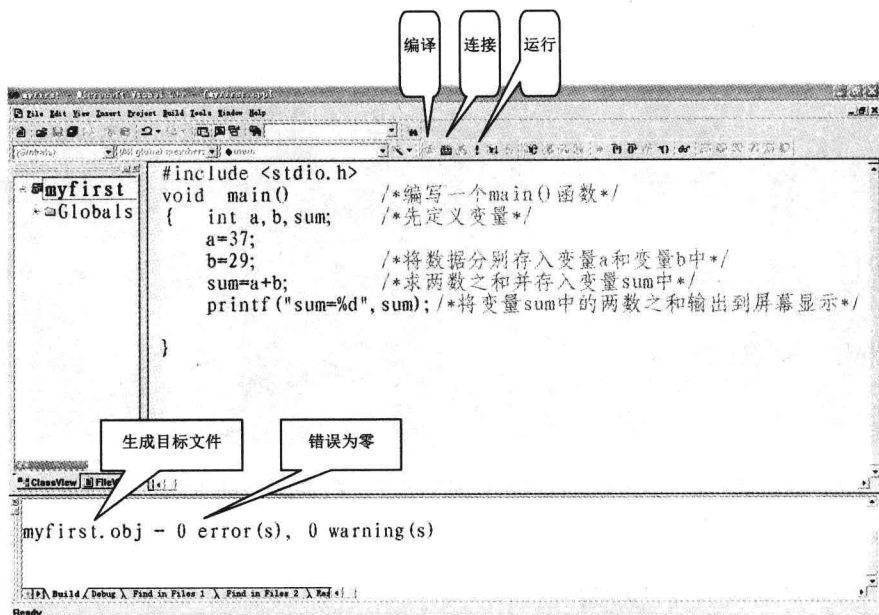


图 1.3 编译源程序

(7) 连接程序。若程序没有语法错误，则可按功能键 F7 或执行“编译”菜单中的“连接”命令或通过工具栏中的相关工具（编译工具右边工具），进行连接生成可执行文件，如图 1.4 所示。

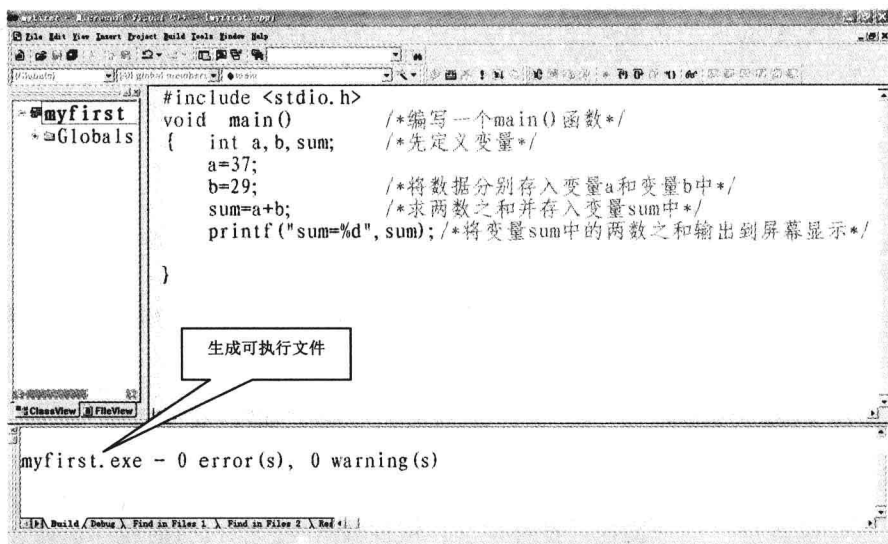


图 1.4 连接源程序

(8) 运行程序。按组合键“Ctrl+F5”，或通过“编译”菜单中的执行命令，或通过工具栏中的“!”工具运行程序。运行结果如图 1.5 所示。



图 1.5 显示运行结果

(9) 新建下一个程序前，先要关闭工作区，重复步骤 2~8。操作方法：“文件”菜单（File）中的“关闭工作区”命令（Close Workspace）。

2. 调试程序的方法

程序调试主要有两种方法，即静态调试和动态调试。

1) 程序的静态调试

程序的静态调试就是在程序编写完以后，检查程序中的语法规则和逻辑结构的正确性。它贯穿在编译、连接和运行的整个过程中。根据程序编译、连接时计算机给出的错误信息对程序进行修改，再对程序进行编译、连接，直到错误信息为零后方可运行程序。

运行过程中出现的错误大体可分为两类：

(1) 运行程序时给出出错信息。多与数据的输入、输出格式有关，以及对文件的操作错误有关。

(2) 运行结果不正常或不正确。此时需要进行动态调试，找到这些错误并改正它们。

2) 程序的动态调试

使用调试命令会使程序运行变得缓慢下来，可以控制程序的运行，可以在任何时刻中断程序、单步执行、查看变量、检查调用栈。通过查看程序流程和当前的运行状态，能有效的帮助你找出错误。通常，程序的动态调试方法有“分段隔

离”、“设置断点”、“跟踪打印”等。

在 VC “组建”(Build) 菜单下的“开始调试”中有 4 条专用的调试命令：“开始调试”(Go),“调试到下一句”(Step into),“调试到光标所在位置”(Run to Cursor)和 Attach to process...

常用的调试快捷键:

- F5: 开始调试
- Shift+F5: 停止调试
- F10: 调试到下一句, 这里是单步跟踪
- F11: 调试到下一句, 跟进函数内部
- Shift+F11: 从当前函数中跳出
- Ctrl+F10: 调试到光标所在位置
- F9: 设置(取消)断点
- Alt+F9: 高级断点设置

设置断点可以让程序在你希望的位置停下来。然后进行单步调试查看过程的正确性。该方法适用于对某个函数或某段代码的正确性产生怀疑时。

也可以如图 1.6 所示来打开 debug 工具条。

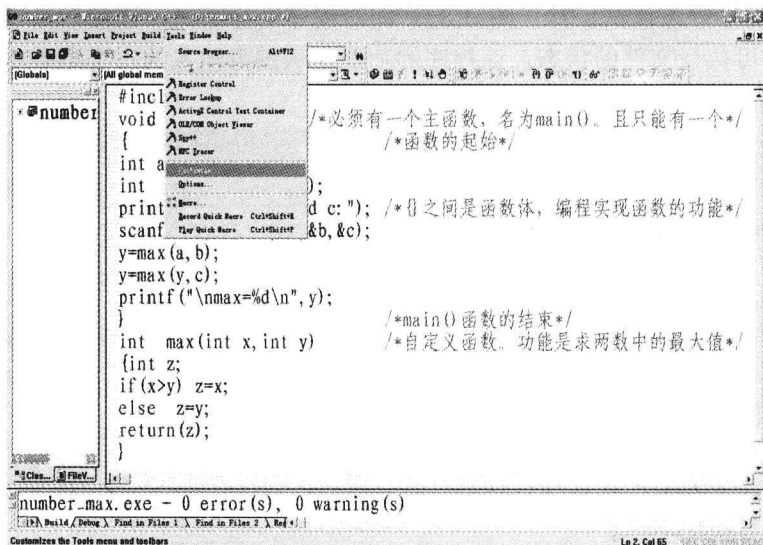


图 1.6 打开 debug 工具条

选择 debug 选项后点击 close 按钮。如图 1.7 所示显示 debug 工具条, 其第一行是控制程序运行的工具, 第二行是观察当前数据的工具, 可方便程序的调试。

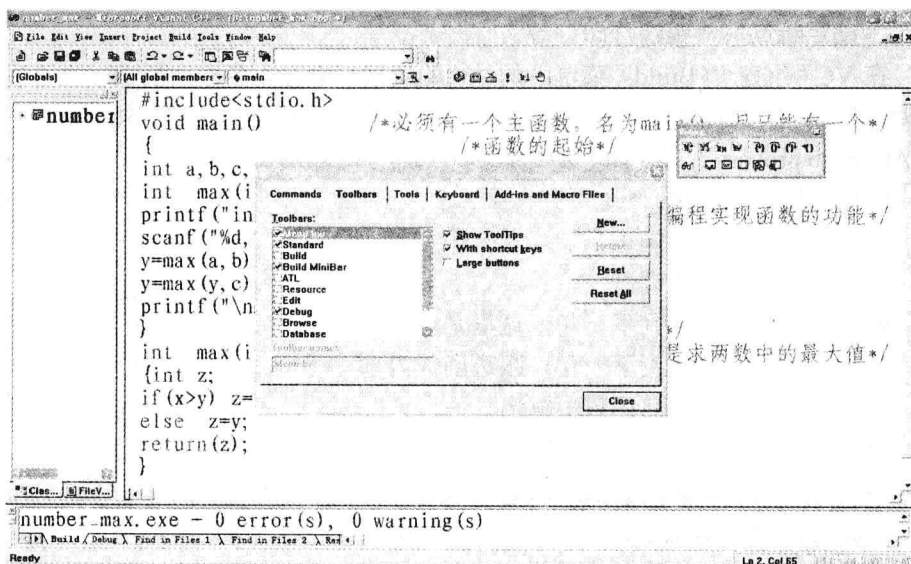


图 1.7 debug 工具条

动态调试的具体操作步骤如下:

(1) 程序静态调试正常 (图 1.8)。即对程序进行编译、连接后错误为零, 自动生成了可执行文件的状态。该程序的功能是从键盘输入的三个整数中找最大数并输出。

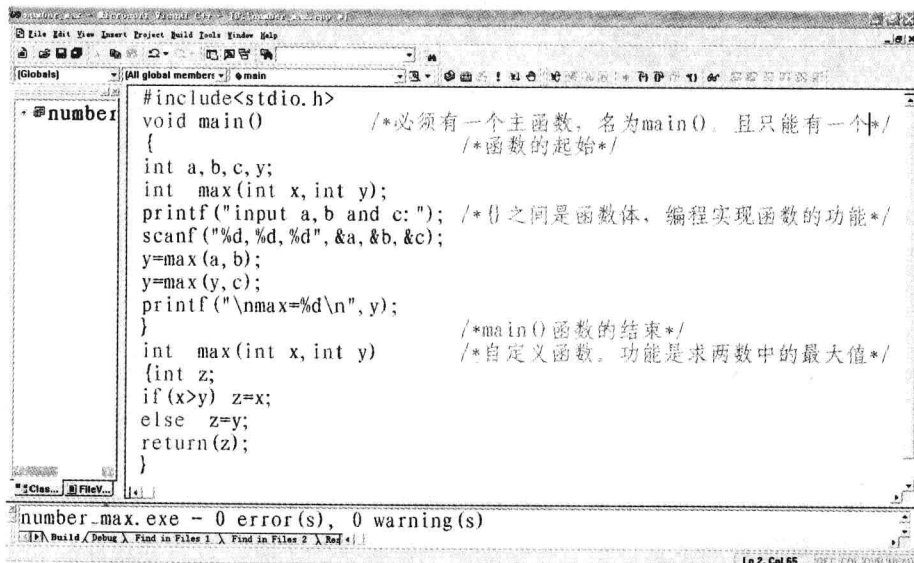


图 1.8 静态调试

(2) 在 $y=\max(a,b)$; 处使程序停止运行, 然后逐条运行, 观察 y 变量的数据变化并感受程序运行的流程。

方法一: 将光标放在 $y=\max(a,b)$; 处, 执行“组建”(Build) 菜单下, “开始调试”中的“调试到光标所在位置”(Run to Cursor), 如图 1.9 所示。

方法二: 在 $y=\max(a,b)$; 处设置断点 (F9), 执行“组建”(Build) 菜单下, “开始调试”中的“开始调试”(Go)。

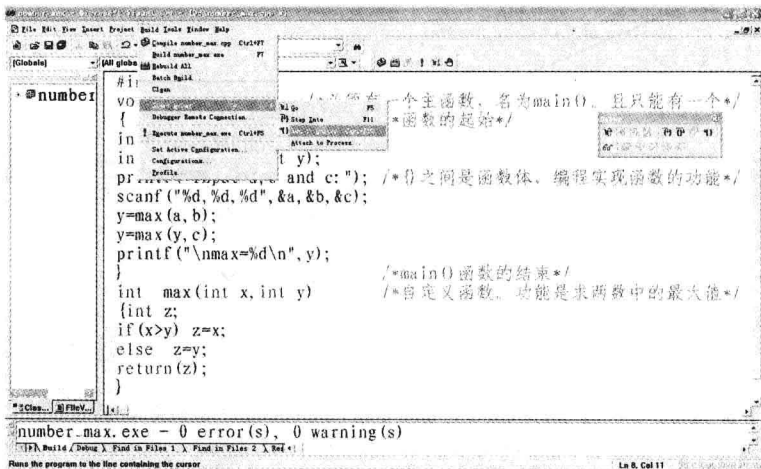


图 1.9 选择调试命令

(3) 当程序执行 $\text{scanf}("%d, \%d, \%d", \&a, \&b, \&c);$ 语句时, 需要从键盘输入三个整数, 分别赋值于 a, b, c 三个变量。如图 1.10 所示。

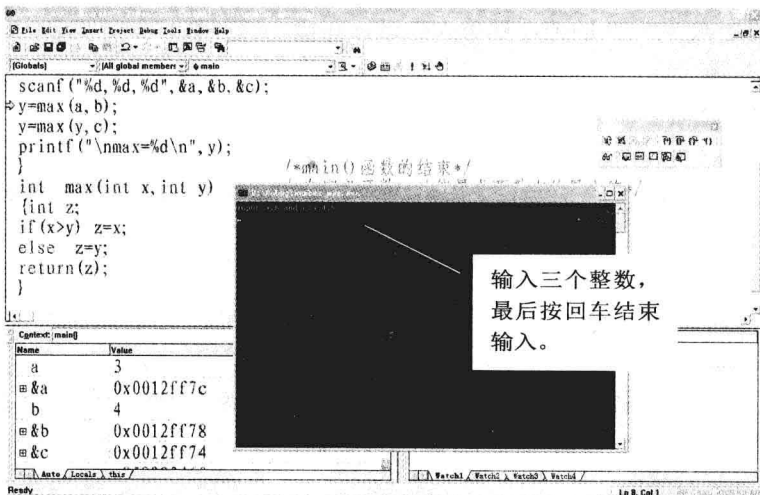


图 1.10 数据输入

(4) 当程序在指定位置停止时,可观察变量的数据。如图 1.11 所示。

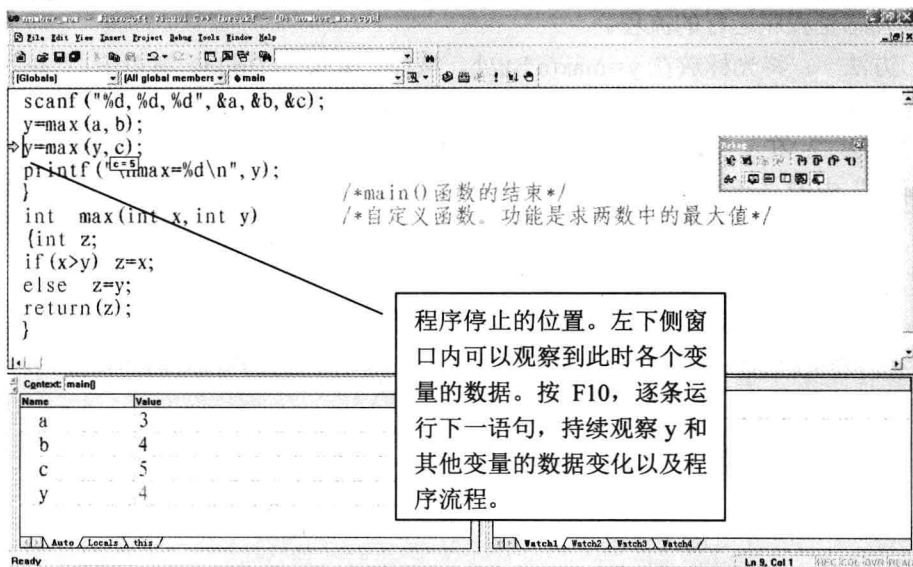


图 1.11 动态调试

(5) 通过查看程序流程和数据变化可以帮助你发现错误。结束调试,可执行“编译”菜单中的停止调试命令 (stop debug)。

【实验内容】

(1) 编程求 $37+29$ 的值,熟悉 VC 运行环境。(提示:可参见实验指导 1 和 2 程序的静态调试)

输入输出示例

输出显示: sum=66

(2) 从键盘输入三个整数,求出最大数并输出。熟悉 VC 的动态调试。(提示:可参见实验指导 2 程序的动态调试)

输入输出示例

输入: input a,b,c:3,4,5

输出: max=5

(3) 通过上机实践,运行下列程序并分析输出结果。

a. 程序代码

```
#include<stdio.h>
```

```
void main()
```

```
{  
    char    ch='a';  
    printf("%c 对应的 ASCII 码是 : %d\n",ch,ch);  
}
```

程序输出结果:

分析输出结果:

b. 程序代码

```
#include<stdio.h>  
void main()  
{  
    int    a=168;  
    float  b=123.456;  
    float  x=35.567,y;  
    printf("a=%5d\n",a);  
    printf("a=%-5d\n",a);  
    printf("b=%6.2f\n",b);  
    printf("b=%e\n",b);  
    y=(int)(x*100+0.5)/100.0;  
    printf("x=%f,y=%f\n",x,y);  
}
```

程序输出结果:

分析输出结果:

c. 程序代码

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

```
Int      a=168;
printf("十进制数%d 对应的八进制数是%o\n",a,a);
printf("十进制数%d 对应的十六进制数是%x\n",a,a);
}
```

程序输出结果:

分析输出结果:

【实验结果与分析】

将源程序、运行结果和思考题分析以及实验中遇到的问题和解决问题的方法，写在实验报告上。

思考题

1. 将实验内容 1 程序中的表达式“a+b”的“a”改为“A”，然后编译程序，分析原因。
2. 将实验内容 1 程序中的表达式“a+b”修改为“a-b”，然后运行程序，分析结果。
3. 指出下面程序中的错误并修改，写出执行结果。

```
#include<stdio.h>
void main()
{   int      inum;
    float    fnum=123.45;
    printf("a=%d\tf=%0.2f\n",inum,fnum);
}
```

实验2 C语言的三种程序结构

2.1 顺序结构

【实验目的】

1. 理解C语言程序的三种基本结构。
2. 掌握变量定义和基本数据处理。
3. 掌握输入输出函数的功能、格式及使用方法，设计简单的顺序结构程序。

【实验指导】

1. 本实验适用的知识点、语法和语句

1) C语言程序的三种基本结构

程序设计提倡清晰的结构，其基本思路是将一个复杂问题的求解过程划分为若干阶段，每个阶段合理选用顺序、选择或循环这三种控制结构，使要处理的问题都容易被理解和处理。

顺序结构只要按照解决问题的顺序写出相应的语句就行，它的执行顺序是自上而下，依次执行。

选择程序结构用于判断给定的条件，根据判断的结果判断某些条件，根据判断的结果来控制程序的流程。

循环结构可以减少源程序重复书写的工作量，用来描述重复执行某段算法的问题，这是程序设计中最能发挥计算机特长的程序结构。

2) 变量定义

格式：数据类型[变量1，变量2，…，变量n];

功能：变量是在程序运行过程中，其值可以被改变的量，以上语句就是定义