

高等学校教材

高等学校计算机实验教学示范中心精品教材

C语言程序设计基础 实验教程

苏庆堂 胡凤珠 主 编
李洪国 王庆军 刘 莉 王环英 副主编



高等教育出版社
HIGHER EDUCATION PRESS

014002350

TP312-43

54

高等学校教材

高等学校计算机实验教学示范中心精品教材

C 语言程序设计基础实验教程

C Yuyan Chengxu Sheji Jichu Shiyān Jiaocheng

苏庆堂 胡凤珠 主编

李洪国 王庆军 刘 莉 王环英 副主编



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING



北航

C1688062

TP312-43
84

内容提要

本书是与刘启明、苏庆堂主编的《C 语言程序设计基础》(以下简称主教材)配套的实验指导教材,以综合训练为主,强化实践动手能力培养。

本书内容分为四篇,第一篇是实验篇,主要给出与主教材各章配套的实验指导;第二篇是习题篇,给出与主教材各章配套的习题及参考答案;第三篇是测试篇,提供了 10 套测试题目;第四篇是综合应用篇,设计了两个完整的案例。

本书内容通俗易懂,突出实际应用,强化关键知识点;程序调试和运行环境为 Windows 平台下的 Visual C++ 6.0。

本书可作为高等学校高级语言程序设计课程的配套教材,也可作为各类计算机程序设计的培训教材或其他从事计算机程序设计的工程技术人员的参考书。

图书在版编目(CIP)数据

C语言程序设计基础实验教程 / 苏庆堂, 胡凤珠主编.
— 北京: 高等教育出版社, 2013. 2
ISBN 978-7-04-036858-1

I. ①C… II. ①苏… ②胡… 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(2013)第013889号

策划编辑 时 阳	责任编辑 时 阳	封面设计 赵 阳	版式设计 杜微言
责任校对 张小镝	责任印制 尤 静		

出版发行	高等教育出版社	咨询电话	400-810-0598
社 址	北京市西城区德外大街 4 号	网 址	http://www.hep.edu.cn
邮政编码	100120		http://www.hep.com.cn
印 刷	化学工业出版社印刷厂	网上订购	http://www.landaco.com
开 本	787mm×1092mm 1/16		http://www.landaco.com.cn
印 张	14.25	版 次	2013 年 2 月第 1 版
字 数	340 千字	印 次	2013 年 2 月第 1 次印刷
购书热线	010-58581118	定 价	20.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换
版权所有 侵权必究
物 料 号 36858-00

前言

多年来,我们一直在进行“应用型人才培养教学内容、课程体系改革”的研究工作。本书根据教育部高等学校计算机基础课程教学指导委员会编制的《高等学校计算机基础教学发展战略研究报告暨计算机基础课程教学基本要求》编写完成,是“应用型人才培养教学内容、课程体系改革”的综合成果,为计算机程序设计课程的教学内容和课程体系改革构建了一个全新的框架。

本书是与刘启明、苏庆堂主编的《C 语言程序设计基础》配套的实验指导教材,以综合训练为主,强化实践动手能力培养。从应用型人才培养的角度来说,学生的实践能力提升是一个重要问题,需要学校和教师采取一系列有效手段来提高学生的实践能力。C 语言程序设计课程需要大量的上机练习与习题练习,这样才能理解和掌握程序设计所涉及的概念、语法规则、编程思想及程序调试方法与技巧。实践证明,通过大量的习题训练和上机实践,积累编程与调试的经验,可以迅速提高学生的编程能力和程序设计水平。

全书内容分为四篇:第一篇是实验篇,是根据主教材各章节内容编写的有关实验,旨在帮助学生循序渐进地掌握 C 语言的语法和编程的基本过程,学会设计解决问题的算法,提高调试和测试程序的能力;第二篇是习题篇,提供与主教材各章配套的习题及参考答案;第三篇是测试篇,给出了考查学生 C 语言掌握情况的综合测试,包括单项选择题、判断题、填空题和编程题等题型;第四篇是综合应用篇,编写了两个规模稍大、具有一定实用性的应用程序,通过这部分内容的学习,帮助学生了解结构化程序设计的基本方法,提高编程能力和解决实际问题的能力。

本书内容通俗易懂,突出实际应用,强化关键知识点;程序调试和运行环境为 Windows 平台下的 Visual C++ 6.0。

本书由苏庆堂、胡凤珠担任主编,李洪国、王庆军、刘莉、王环英担任副主编。

在本书的编写、出版过程中,我们得到许多专家的精心指点和热情帮助。教学指导委员会先后两次在我校召开计算机课程教学研讨会,清华大学、北京大学、中国人民大学、复旦大学、南京大学、中国科学技术大学等近百所高校参加,专家学者提出了很多宝贵意见。高等教育出版社为本书出版做了大量的工作,在此一并表示衷心感谢。

尽管我们为本书编写付出了很大努力,并希望能成就一部精品。但限于作者水平,书中难免有疏漏和不妥之处,敬请广大读者不吝赐教。

编者

2012 年 12 月

目 录

第 1 篇 实验篇 1

第 1 章 C 语言概述	1
第 2 章 C 语言程序基础	7
第 3 章 选择结构程序设计	13
第 4 章 循环结构程序设计	18
第 5 章 函数与宏定义	24
第 6 章 数组	30
第 7 章 指针	36
第 8 章 结构体与共用体	41
第 9 章 文件	45

第 2 篇 习题篇 51

第 1 章 C 语言概述	51
参考答案	53
第 2 章 C 语言程序基础	54
参考答案	56
第 3 章 选择结构程序设计	57
参考答案	65
第 4 章 循环结构程序设计	69
参考答案	76
第 5 章 函数与宏定义	79
参考答案	83
第 6 章 数组	85
参考答案	89
第 7 章 指针	91
参考答案	94
第 8 章 结构体与共用体	96
参考答案	102

第 9 章 文件	104
参考答案	108

第 3 篇 测试篇 111

测试一	111
测试一参考答案	117
测试二	119
测试二参考答案	126
测试三	128
测试三参考答案	133
测试四	134
测试四参考答案	138
测试五	140
测试五参考答案	144
测试六	146
测试六参考答案	151
测试七	153
测试七参考答案	160
测试八	162
测试八参考答案	167
测试九	169
测试九参考答案	174
测试十	176
测试十参考答案	180

第 4 篇 综合应用篇 183

4.1 餐馆订餐系统	183
4.2 幼儿园管理系统	200

参考文献 221

第 1 篇 实 验 篇

第 1 章 C 语言概述

【实验目的】

1. 熟悉 Visual C++ 6.0 开发工具的使用。
2. 学习编写简单的 C 语言程序。
3. 掌握 C 语言程序的编辑、编译及运行等过程。
4. 掌握 C 语言程序的单步跟踪调试方法及常用的编译方法。

【实验内容】

1. 启动 Visual C++ 6.0，熟悉其窗口组成。
2. 编写一个简单的程序，在屏幕上打印输出 Welcome to YanTai!，学会新建、编辑、编译及运行 C 语言程序。
3. 输入两个数 a、b，打印输出 $s=a*b$ 的值。单步跟踪 s 值的变化情况，掌握程序的单步跟踪调试方法。

【实验步骤】

【实验 1-1】 启动 Visual C++ 6.0，熟悉其窗口组成。

(1) 在 Windows XP 系统中，选择任务栏中“开始”|“程序”| Microsoft Visual Studio 6.0 | Microsoft Visual C++ 6.0，就可以启动 Visual C++ 6.0，如图 1.1.1 所示。

(2) 窗口组成。

- ① 编辑窗口：设计、处理源程序代码及项目资源的工作区。
- ② 输出窗口：显示编译及调试的结果，方便用户修改程序的错误。错误提示信息包括错误条数、错误所在位置和错误原因等。
- ③ 工作区窗口：主要包含项目文件、资源等基本信息。

【实验 1-2】 编写一个简单的 C 语言程序，在屏幕上输出 Welcome to YanTai!。

(1) 新建文件。打开 Microsoft Visual C++ 6.0 的操作界面后，选择主菜单中的 File|New 命令，打开如图 1.1.2 所示的 New 对话框。

在对话框中选择 Files 选项卡，在下面列出的选项中选择 C++ Source File，表示要建立新的

C++源程序文件，在右边相应的 File 文本框中输入文件名。输入的文件名应包含扩展名.c，如果不指定扩展名，系统默认指定为 C++源程序文件的扩展名.cpp。假设文件名为 c1.c，然后在 Location 列表框中选择文件的保存位置 C:\。最后单击 OK 按钮，将在 C 盘根目录下建立一个名为 c1.c 的 C 语言程序源文件。

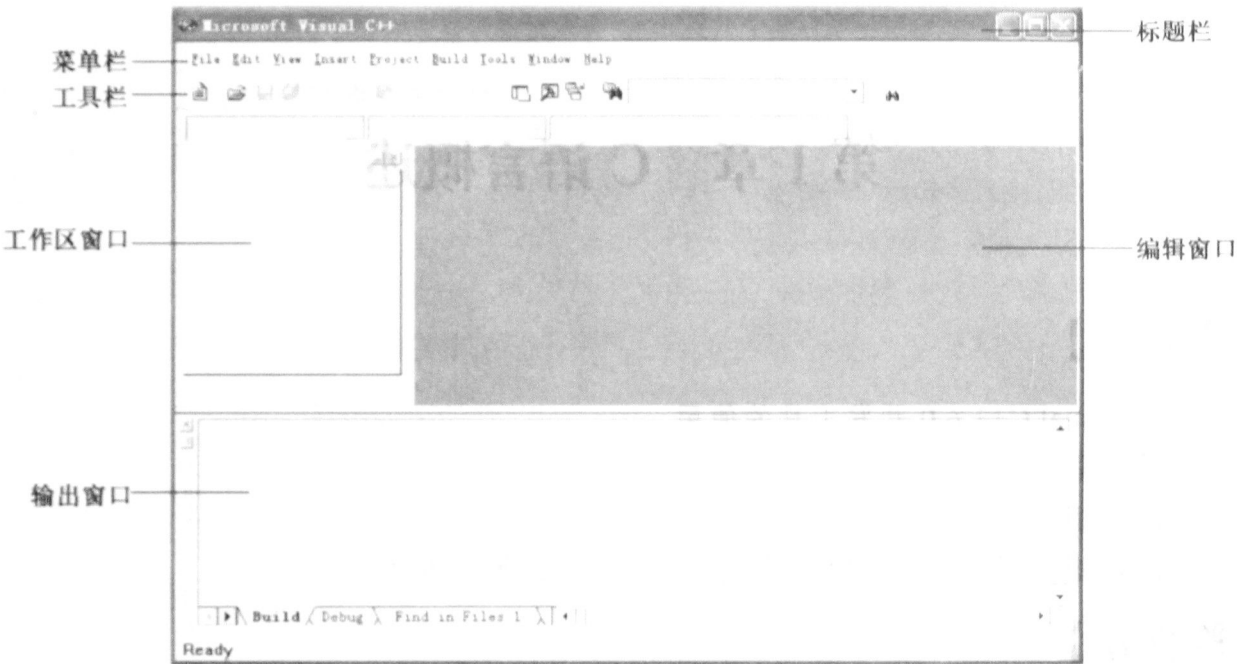


图 1.1.1 Visual C++ 6.0 主窗口界面

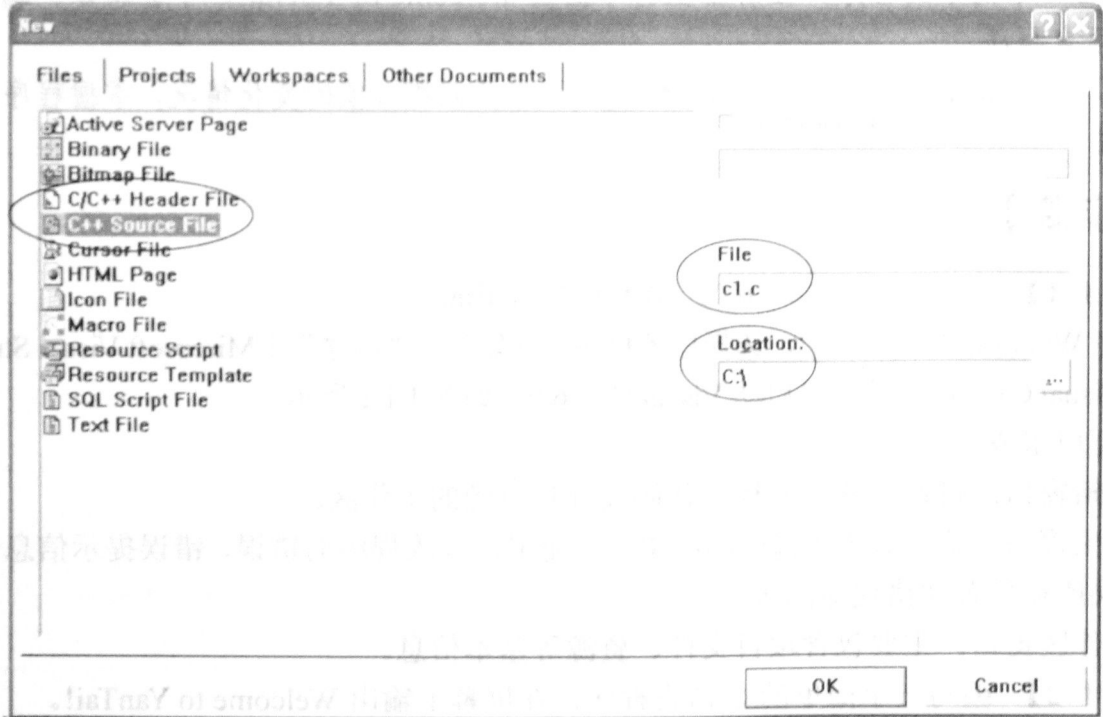


图 1.1.2 New 对话框

(2) 编辑并保存源文件。在如图 1.1.3 所示的程序编辑区内输入下面的源代码。

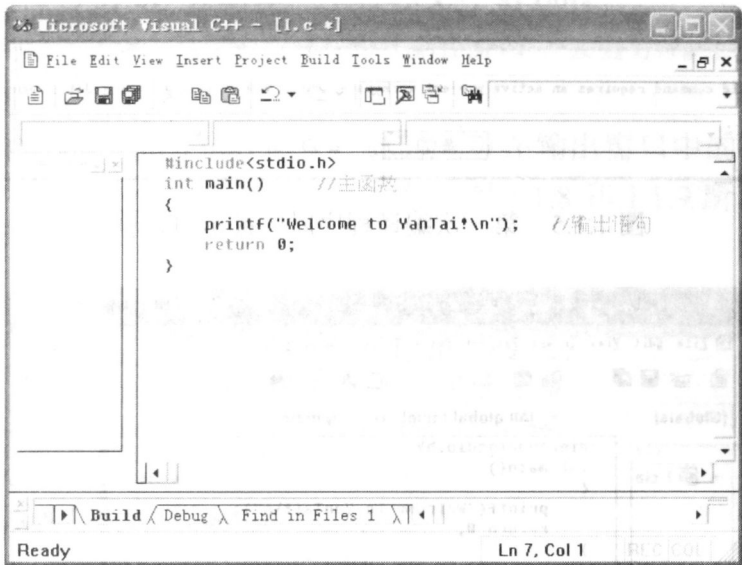


图 1.1.3 程序编辑窗口

```
#include <stdio.h>
int main ( )
{
    printf ("Welcome to YanTai! \n");
    return 0;
}
```

选择 File | Save 命令，保存编辑好的源文件。

(3) 编译程序。选择 Build | Compile c1.c 命令，如图 1.1.4 所示。



图 1.1.4 编译程序

如果是第一次编译源程序，会弹出如图 1.1.5 所示的对话框，单击“是”按钮，开始编译。

编译时，输出窗口中将显示相应的编译说明。如果源程序没有词法和语法错误，输出窗口中将显示“c1.obj - 0 error(s), 0 warning(s)”，如图 1.1.6 所示，说明编译没有错误，生成目标文件 c1.obj。

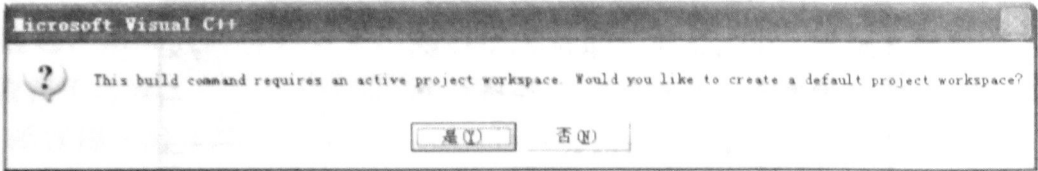


图 1.1.5 第一次编译程序时的提示对话框

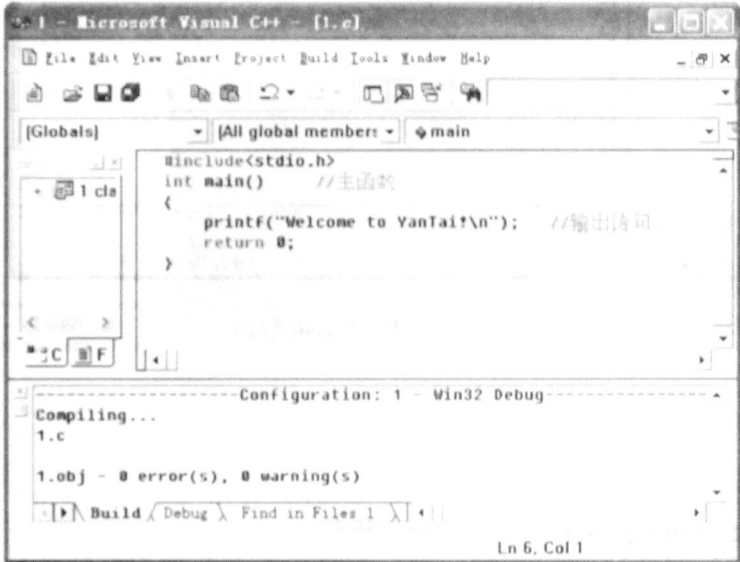


图 1.1.6 程序编译无错

如果源程序有词法和语法错误，编译时系统会给出错误和警告信息，并在下方的输出窗口中显示出来。

(4) 链接与运行。源程序编译好后会生成一个目标文件 `c1.obj`，目标文件不能被计算机直接执行，必须将目标文件和相关的库函数或目标程序链接成一个可执行文件，才能被计算机执行。选择 `Build | Build c1.obj` 命令，生成可执行文件 `c1.exe`，该文件与 `c1.obj` 保存在同一文件夹下。

(5) 选择 `Build | Execute c1.exe` 命令运行程序；程序运行后，将显示一个控制台程序窗口，显示程序的输出结果，如图 1.1.7 所示。

【实验 1-3】 输入两个数 `a`、`b`，打印输出两个数乘积 `s` 的值。单步跟踪 `s` 值的变化情况，掌握程序的单步跟踪调试方法。

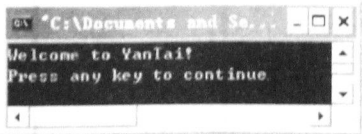


图 1.1.7 程序运行结果

(1) 新建文件，输入如下代码：

```
# include <stdio.h>
int main( )    // 主函数
{
    int a,b,s;
    printf("输入 a,b 两个数\n");
    scanf ("%d%d",&a,&b);
    s=a*b;
    printf("s 的结果是: %d\n",s);
    return 0;
}
```

输入完后，保存文件为 `s1.c`，并编译产生目标文件 `s1.obj`。

(2) 单步跟踪调试程序。单步跟踪调试时，输出窗口中会显示每个变量的当前值，即实时跟踪变量的值，所以称“单步跟踪调试”。

选择 **Build | Start Debug | Step Into** 命令，此时，可在输出窗口中的 **Name** 处添加要跟踪的变量，则单步跟踪时，**Value** 处会显示变量的当前值。图 1.1.8 和 1.1.9 所示分别为添加跟踪变量前后的效果。

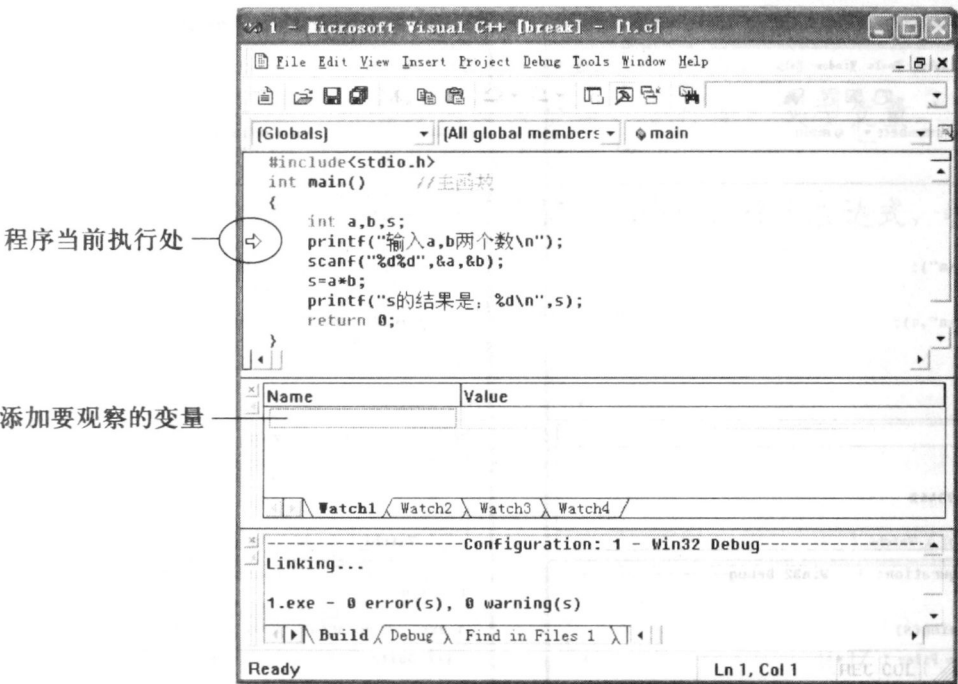


图 1.1.8 添加跟踪变量前的界面

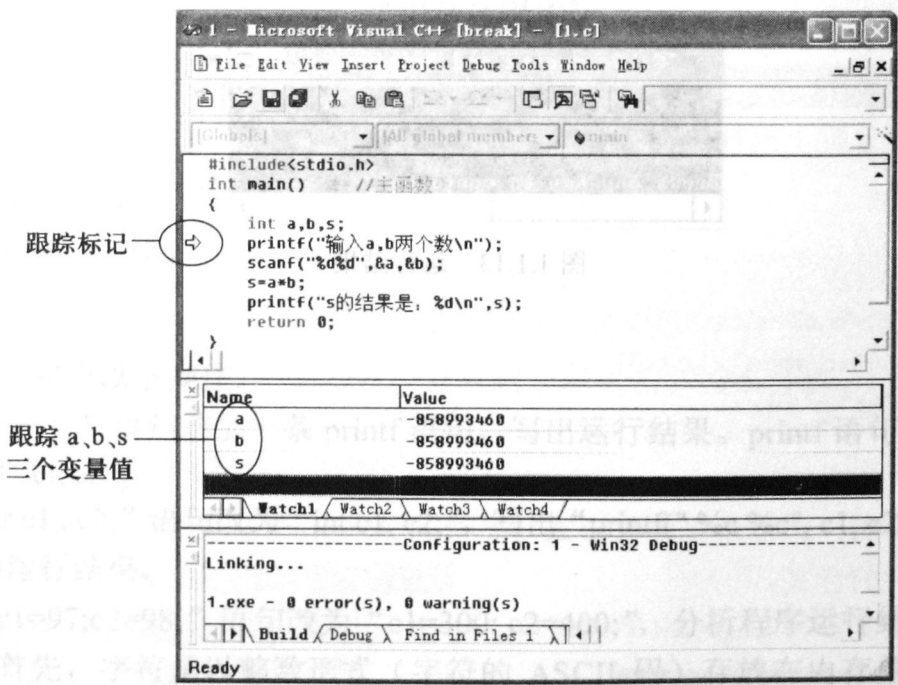


图 1.1.9 添加跟踪变量后的界面

添加跟踪变量后，可反复按快捷键 **F10**，每按一次 **F10** 键，程序就执行一步，可以从跟踪

标记的移动来观察程序执行所在的位置。当程序执行到 `scanf` 语句时，在控制台程序窗口中输入变量 `a`、`b` 的值，如输入 `a` 的值为 4，`b` 的值为 5，如图 1.1.10 所示。输入后，回到程序的编译窗口，继续按 F10 键单步跟踪。通过观察窗口可以观察变量 `a`、`b` 和 `s` 的值，如图 1.1.11 和图 1.1.12 所示。单步跟踪到程序结束后，控制台程序窗口会显示程序的输出结果，如图 1.1.13 所示。

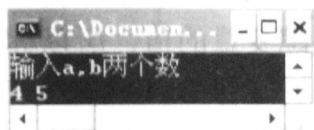


图 1.1.10 输入 `a`、`b` 的值



图 1.1.11 单步跟踪 (1)



图 1.1.12 单步跟踪 (2)

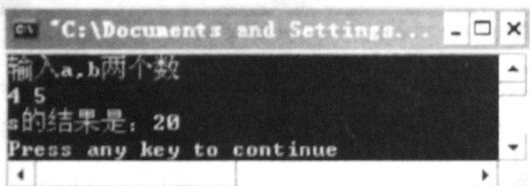


图 1.1.13 运行结果

第2章 C 语言程序基础

【实验目的】

- 1. 运行简单的 C 语言程序，初步了解 C 语言程序的特征。
- 2. 掌握 C 语言的数据类型，熟悉如何定义整型、字符型、实型变量，以及对它们进行赋值的方法；了解以上数据类型输出时所用的格式转换符。
- 3. 学会使用 C 语言中的算术运算符，以及包含这些运算符的表达式，特别是自加（++）和自减（--）运算符的使用。
- 4. 熟悉 printf 函数以及 scanf 函数的基本使用方法。

【实验内容】

- 1. C 语言数据类型的使用。
- 2. 数据输出的格式。
- 3. 算术运算符、算术表达式的使用。
- 4. printf 函数和 scanf 函数的使用。

【实验步骤】

【实验 2-1】 输入并运行下列程序，写出运行结果。

```
#include <stdio.h>
int main ()
{
    char c1, c2;
    c1=97;c2=98;
    printf ("%c %c", c1, c2);
    return 0;
}
```

在此基础上，完成以下操作。

(1) 在“return 0;”语句前加一条 printf 语句，写出运行结果。printf 语句如下：

```
printf ("%d,%d", c1, c2 );
```

(2) 将“char c1, c2;”语句改为“int c1, c2;”，写出“printf(" %c %c", c1, c2); printf(" %d,%d", c1, c2);”语句的运行结果。

(3) 再将“c1=97;c2=98;”语句改为“c1=300; c2=400;”，分析程序运行结果。

解题思路：首先，字符是以整数形式（字符的 ASCII 码）存放在内存单元中的；其次，字符的 ASCII 码用 7 位二进制位表示，即取值范围是 0~127，所以越界会出现一些意想不到的结果。

程序分析：先将 ASCII 码值为 97 的字符'a'赋值给 c1，再将 ASCII 码值为 98 的字符'b'赋值给 c2，以字符形式输出。输出结果如下：

```
a bPress any key to continue
```

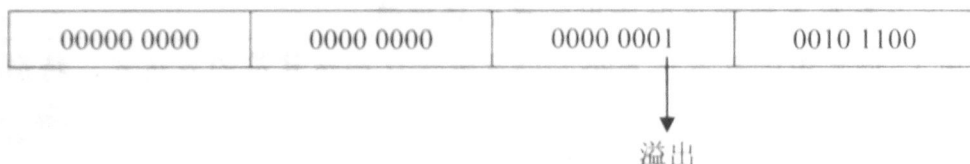
(1) 以整数形式输出字符 c1,c2。输出结果如下：

```
a b 97,98Press any key to continue
```

(2) 将 c1,c2 定义为整型，其输出结果如下：

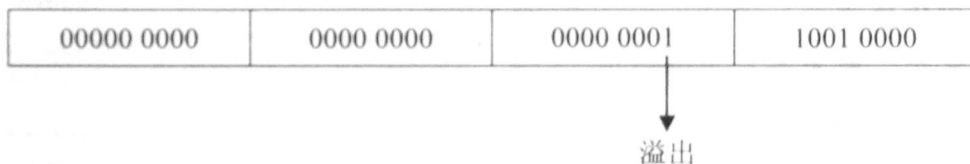
```
a b 97,98Press any key to continue
```

(3) 赋给 c1,c2 的值若超出字符型数据的范围，例如，当 c1=300 时，数据在内存中的存放形式如下：



由于整型数据在内存中占 1 B，故二进制形式转换为 0010 1100，即 44。ASCII 码值是 44 的十进制数对应的字符是','。

同理，当 c2=400 时，数据在内存中的存放形式如下：



则整型变量 c2 的二进制形式是 1001 0000，是负数，以补码形式存放在内存中，其原码表示的数值是-112，由于 ASCII 码值的范围是 0~127，所以输出的是一个未知结果'?'。因此，其输出结果如下：

```
. ?300,400Press any key to continue
```

【实验 2-2】 分析下列程序，写出运行结果，再输入计算机运行，将输出的结果与分析得到的结果进行比较。

```
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;
}
```

解题思路：首先，字符是以整数形式（字符的 ASCII 码）存放在内存单元中的；其次，要清楚常用转义字符的含义。

程序分析：转义字符'\t'表示水平制表符，'\b'表示退格，'\n'表示换行。输出结果如下：

```
aa bb abc
A NPress any key to continue
```

【实验 2-3】 自加、自减运算符的使用。输入并分析以下程序，写出运行结果。

```
#include <stdio.h>
int main ()
{
    int i, j, m, n;
    i=8; j=10;
    m=++i; n=j++;
    printf ("%d,%d,%d,%d", i, j, m, n);
    return 0;
}
```

将以上程序分别做如下改动，先分析程序运行结果，再运行程序进行验证。

(1) 将语句“`m=++i; n=j++;`”改为“`m=i++; n=++j;`”。

(2) 将程序做如下修改：

```
#include <stdio.h>
int main ()
{
    int i, j;
    i=8;
    j=10;
    printf ("%d,%d", i++, j++);
    return 0;
}
```

(3) 在(2)的基础上，将 `printf` 语句改为“`printf ("%d,%d", ++i, ++j);`”。

(4) 在(2)的基础上，将 `printf` 语句改为“`printf ("%d,%d,%d,%d", i, j, i++, j++);`”。

(5) 将程序做如下修改：

```
int main ()
{
    int i, j, m=0, n=0;
    i=8; j=10;
    m+=i++; n--=--j;
    printf ("%d,%d,%d,%d", i, j, m, n);
    return 0;
}
```

解题思路：自加运算 `n++` 表示 `n=n+1`；自减运算 `n--` 表示 `n=n-1`。

程序分析：当“`++`”或“`--`”运算符在变量前时，表示先自加或自减，再赋值；而当“`++`”或“`--`”运算符在变量后时，表示先赋值，再自加或自减。输出结果如下：

9,11,9,10 Press any key to continue.

改动程序后，输出结果分别如下：

(1) **9,11,8,11** Press any key to continue.

(2) **8,10** Press any key to continue

(3) **9,11** Press any key to continue.

(4) **8,10,8,10** Press any key to continue

(5) **9,9,8,-9** Press any key to continue.

【实验 2-4】 输入并运行如下程序，注意观察运行结果。

```
#include <stdio.h>

int main ( )
{
    int a,b;
    float c;
    scanf ("%d%d%f",&a,&b,&c);
    printf ("a=%d\n",a);
    printf ("b=%d\n",b);
    printf ("c=%f\n",c);
    return 0;
}
```

说明：运行这个程序时，分别按下面 4 种格式输入数据，注意观察各自的运行结果。

(1) 第一种格式：

10 25 4.72 ✓

(2) 第二种格式：

10 ✓

25 ✓

4.72 ✓

(3) 第三种格式：

10 ✓

4.72 ✓

(4) 第四种格式：

10, 25, 4.72 ✓ (这种输入格式是错误的，想一想错在什么地方？)

解题思路：注意输入函数 scanf 的使用。

程序分析：

(1) 当输入的数据之间用空格分隔时，若输入内容与函数 scanf 中要求的输入个数一致，则输出结果为：

```
10 25 4.72
a=10
b=25
c=4.720000
Press any key to continue
```

(2) 当输入数据之间用“✓”分隔时，若输入内容与函数 scanf 中要求的输入个数一致，则输出结果为：

```
a=10
b=25
c=4.720000
Press any key to continue
```

(3) 当输入数据之间用“,”分隔时，若输入内容与函数 scanf 中要求的输入个数不一致，则根据数据类型将输入数据存入每个变量地址空间，输出结果为：

```
10
4.72
a=10
b=4
c=0.720000
Press any key to continue
```

(4) 当输入数据之间用“,”分隔时, 由于输入内容与函数 scanf 中要求的输入格式不一致, 输出结果不能正确显示, 输出结果为:

```
10,25,4.72
a=10
b=-858993460
c=-107374176.000000
Press any key to continue
```

【实验 2-5】 完善程序。下面程序的功能是: 根据商品的原价和折扣率, 计算商品的实际售价。在程序中的横线处填写正确的语句或表达式, 使程序完整。上机调试程序, 使程序的运行结果与书中的结果一致。

```
# include <stdio.h>
int main ( )
{
    double cost,percent,c;
    printf ("Please Enter the First Price: ");
    scanf ("_____",&cost);
    printf (______);
    scanf ("_____",&percent);
    c=cost*percent;
    printf (_____,c);
    return 0;
}
```

解题思路: 注意输入函数 scanf 和输出函数 printf 的使用。

编写程序:

```
# include <stdio.h>
int main ( )
{
    double cost,percent,c;
    printf ("Please Enter the First Price: ");
    scanf ("%lf",&cost);
    printf ("Enter the Discount: ");
    scanf ("%lf",&percent);
    c=cost*percent;
    printf ("Actual Price is:%f\n",c);
    return 0;
}
```

程序运行结果:

