

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

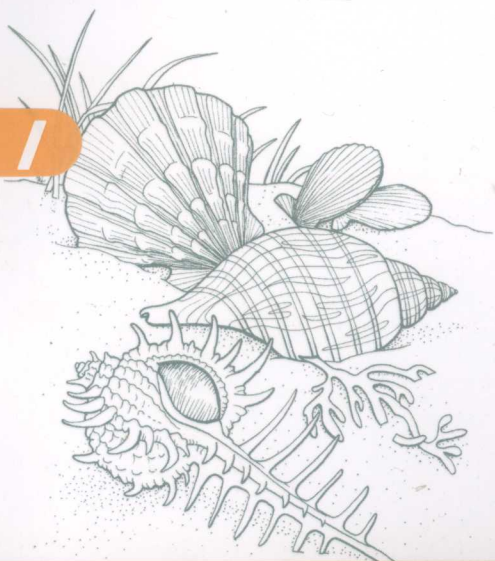
C语言程序设计 实验教程

Experiment Tutorial of C Language Programming

顾国松 主编

贾小军 骆红波 许巨定 副主编

- 精选实例、精讲算法
- 循序渐进、深入浅出
- 聚焦思维、提升能力



高校系列

 人民邮电出版社
POSTS & TELECOM PRESS

014013190

TP312C-43
856

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C语言程序设计 实验教程

Experiment Tutorial of C Language
Programming

顾国松 主编

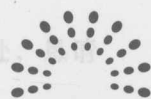
贾小军 骆红波 许巨定 副主编



北航

C1700473

TP312C-43
856



高校系列

人民邮电出版社

北京

001310310

图书在版编目(CIP)数据

C语言程序设计实验教程 / 顾国松主编. -- 北京 :
人民邮电出版社, 2014.2
21世纪高等学校计算机规划教材
ISBN 978-7-115-33834-1

I. ①C… II. ①顾… 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317529号

内 容 提 要

C语言是目前国内外使用最为广泛的程序设计语言之一,是高等学校计算机语言类课程的首选教学内容。本书是《C语言程序设计》的配套教材,重点突出程序设计的实践性。全书主要包含4部分内容。实验分析与指导:全书提供了20个典型实验,每一个实验包含实验目的、实验内容、实验分析和提高实验。习题集:旨在加强C语言知识点的训练,便于读者掌握各类基本知识。主教材习题的参考答案:通过参考答案的学习,提高读者自身的程序设计能力。等级考试例题:便于读者检测学习的效果和熟悉等级考试题型。本书以培养程序设计思维为主,以知识点训练为辅,相得益彰,条理清晰。

本书可作为高等学校理工类专业“C语言程序设计”课程的教学辅导用书,也可作为参加国家计算机等级考试(二级C语言)的辅导用书,或作为计算机程序设计爱好者的自学参考书。

-
- ◆ 主 编 顾国松
副 主 编 贾小军 骆红波 许巨定
责任编辑 武恩玉
责任印制 彭志环 杨林杰
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京中新伟业印刷有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 18
字数: 429千字
- 2014年2月第1版
2014年2月北京第1次印刷
-

定价: 37.00元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第0021号

前 言

C 语言是目前国内外使用最为广泛的程序设计语言之一, 它具有语法丰富、结构清晰、使用方便、执行效率高以及可移植性好等特点。

C 语言程序设计课程是一门实践性很强的课程, 通过程序设计、上机调试和分析测试, 不仅能使读者理解和掌握基本理论知识, 更能增加读者程序编写的能力。

本书是《C 语言程序设计》的配套实验教材, 以培养程序设计思维为主线, 注重实验实例的精选和算法分析, 全书系统地介绍了 C 语言的基本知识点和常用的编程算法。

本书的主要核心内容是实验分析与指导, 共包括 20 个实验, 每个实验都由实验目的、实验内容、实验分析和提高实验 4 部分组成。为了满足不同层次读者的需要, 本书安排了两个层次的实践内容。第 1 层次是供读者学习、模仿和验证的实验内容, 每个实验内容由多个典型实例组成, 并为每一个实例提供详细的题意解析、算法分析和程序代码。第 2 层次是本书的提高部分, 主要为不满足于完成基本实验内容的读者提供独立分析、设计和编写程序的机会。

本书以培养程序设计思维为主线, 同时兼顾 C 语言基本知识点的练习和掌握。因此, 在第 2 部分为读者提供了大量的与理论知识配套的习题集, 读者经过练习能更加深刻的理解 C 语言的基本知识。第 3 部分是教材习题的参考答案, 读者可对此进行学习和研究, 以便掌握更多的程序设计算法, 提高自身的程序设计和编写能力。

本书注重程序设计算法的精细剖析, 既注重培养读者设计程序的能力, 又提倡良好的程序设计风格。全书以最新的 C99 为标准, 每个实例的源程序代码均在 C Free 5.0 环境下调试通过, 可直接使用。另外, 本书提供了所有案例的数据及运行结果, 方便读者参考及研究。

本书由多位优秀教师在多年 C 语言教学与程序设计实践的基础上, 结合多次编写相关讲义和教材的经验总结而成, 同时本书在编写过程中也参考了大量书籍, 得到了许多同行的帮助与支持, 在此向他们表示衷心的感谢。本书由顾国松任主编, 贾小军、骆红波、许巨定任副主编。实验 1 由方玫编写, 实验 2 由许巨定编写, 实验 3、4、5 和实验 14 由顾国松编写、实验 6、7、8、18 和实验 19 由贾小军编写、实验 9、10、11 和实验 20 由骆红波编写, 实验 12、13 和等级考试试题由潘云燕编写和整理、实验 15、16 和实验 17 由刘锦萍编写。全书由顾国松统稿。

由于时间仓促, 编者才疏学浅, 书中难免存在不足或者遗漏之处, 恳请广大读者提出修改建议。

编 者

2013 年 10 月

目 录

第一部分 实验分析与指导

实验 1 C 语言程序的运行环境和运行方法	2	实验 11 变量的作用域与存储类别	54
实验 2 数据运算	7	实验 12 指针变量	59
实验 3 顺序结构与分支结构程序设计	13	实验 13 指针与数组	63
实验 4 循环结构程序设计	18	实验 14 预处理命令	69
实验 5 循环综合程序设计	23	实验 15 结构体变量初始化与引用	74
实验 6 一维数组	29	实验 16 结构体与函数	80
实验 7 二维数组	35	实验 17 位运算	87
实验 8 字符数组	40	实验 18 文件读写	91
实验 9 函数的基本用法	46	实验 19 文件定位与检测	98
实验 10 函数的嵌套与递归	50	实验 20 C 语言程序设计综合应用	102

第二部分 习题集

习题 1 概述	110	习题 6 指针	177
习题 2 C 语言程序设计基础	112	习题 7 编译预处理	194
习题 3 程序控制结构	121	习题 8 结构体与共用体	198
习题 4 数组	144	习题 9 位运算	205
习题 5 函数	160	习题 10 文件	208

第三部分 主教材各章习题参考答案

第 1 章 C 语言概述	212	第 7 章 编译预处理	237
第 2 章 C 语言程序设计基础	213	第 8 章 结构体与共用体	239
第 3 章 程序控制结构	214	第 9 章 位运算	245
第 4 章 数组	220	第 10 章 文件	247
第 5 章 函数	224	第 11 章 C 语言程序设计综合案例	253
第 6 章 指针	230		

第四部分 等级考试模拟题与参考答案

计算机等级考试模拟题 1 (二级 C)	258	计算机等级考试模拟题 2 (二级 C)	269
计算机等级考试 C 语言上机模拟题 1	265	计算机等级考试 C 语言上机模拟题 2	277

参考文献

第一部分

实验分析与指导

本部分共包含 20 个实验，每个实验由实验目的、实验内容、实验步骤和提高实验组成。

实验 1 C 语言程序的运行环境和运行方法

实验 2 数据运算

实验 3 顺序结构与分支结构程序设计

实验 4 循环结构程序设计

实验 5 循环综合程序设计

实验 6 一维数组

实验 7 二维数组

实验 8 字符数组

实验 9 函数的基本用法

实验 10 函数的嵌套与递归

实验 11 变量的作用域与存储类别

实验 12 指针变量

实验 13 指针与数组

实验 14 预处理命令

实验 15 结构体变量初始化与引用

实验 16 结构体与函数

实验 17 位运算

实验 18 文件读写

实验 19 文件定位与检测

实验 20 C 语言程序设计综合应用

实验 1

C 语言程序的运行环境和运行方法

1.1 实验目的

1. 熟悉 Visual C++ 6.0、C-Free 5.0 两种 C 语言程序设计环境。
2. 掌握在两种运行环境下如何编辑、编译、运行和连接一个 C 程序。
3. 通过运行简单的 C 程序，认识 C 语言程序的结构特点，学习程序的基本编写方法。

1.2 实验内容

1. 熟悉 C-Free 集成环境的使用。进入 C-Free5.0，编辑运行程序，实现输出一行信息。
源程序如下：

```
#include <stdio.h> /*包含标准库的信息*/
int main() /*定义名为 main 的函数，它不接受参考值*/
{ /*main 函数的语句都被包含在花括号中*/
    printf("Hello Kitty\n"); /*main 函数调用库函数 printf 显示打印的序列*/
    return 0; /*其中\n 代表换行符*/
}
```

注：/*...*/表示对程序的注释，包含在/*与*/之间的内容将被编译器忽略。

将程序修改如下，观察输出结果。

```
#include <stdio.h>
int main()
{
    printf("Hello");
    printf(" Kitty");
    printf("\n");
    return 0;
}
```

若将 printf("\n");删除，观察程序运行结果。

2. Visual C++6.0 运行环境的使用。进入 VC 环境，编辑运行上面的程序。
3. 在两种环境中分别编辑运行程序，实现两个整数的加法运算。

```
#include <stdio.h>
int main()
```

```

{
    int a,b,sum;                /*定义整型变量 a,b,sum*/
    a=110; b=120;              /*对变量 a,b 赋值*/
    sum = a+b;                 /*计算 sum 的值*/
    printf("sum=%d",sum);      /*将 sum 的值以整型的方式输出*/
    return 0;
}

```

将以上程序修改如下, 实现从键盘上输入任意两个整数, 完成加法运算。

```

#include <stdio.h>
int main()
{
    int a,b,sum;                /*定义整型变量 a,b,sum*/
    scanf("%d,%d",&a,&b);      /*从键盘上输入两个数给 a,b 赋值*/
    sum = a+b;                 /*计算 sum 的值*/
    printf("sum=%d",sum);      /*将 sum 的值以整型的方式输出*/
    return 0;
}

```

注: 输入的两个数之间用逗号间隔, 如输入 4, 5。

4. 程序的简单语法错误。修改上面的程序, 观察运行结果。

(1) 去掉 printf 语句末尾的分号, 运行程序, 观察错误信息。

(2) 去掉 scanf 语句中的 “&” 符号, 运行程序, 观察错误信息。

(3) 将程序中的表达式 “sum = a+b” 修改为 “Sum = a+b”, 运行程序, 观察错误信息。

1.3 实验步骤

1. 实验内容 1

(1) 双击打开 C-Free 快捷方式, 运行 C-Free。

(2) 单击工具栏上的 “新建” 按钮, 这样就会产生一个新的代码编辑窗口, 在编辑区中输入程序。

(3) 在主菜单上, 单击 “文件→保存” 菜单, 将程序保存成扩展名为 .c 的文件 (如保存为实验 1-1.c 文件) 及选择保存的位置 (如 D: 盘)。

(4) 对文件进行编译, 在主菜单上, 单击 “构建→运行” 菜单或直接按【F5】键运行程序, 如图 1-1 所示; 若程序正确, 输出结果如图 1-2 所示。

2. 实验内容 2

使用 Visual C++6.0 编辑运行程序, 新建一个文件有两种方法。

方法一: 新建文件, 编译后让系统自动生成一个工程。

方法二: 先新建工程后新建文件。

下面我们详细介绍这两种方法。

方法一: 系统自动生成工程文件。

(1) 新建文件。选择 File→New, 之后选择 File→C++ Source File, 输入文件名, 扩展名为 .c, 同时设置文件存放的路径, 如图 1-3 所示。

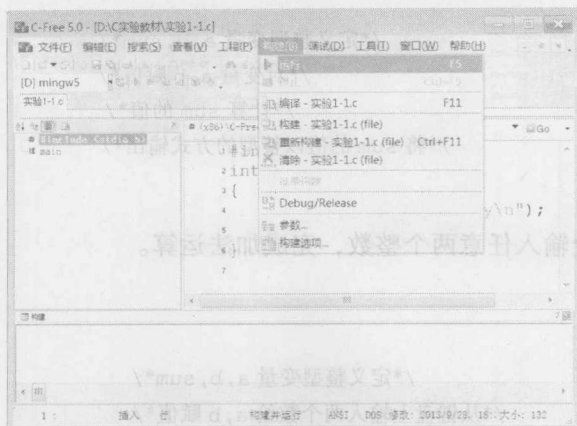


图 1-1 运行程序

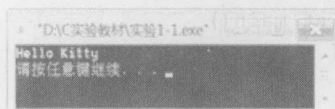


图 1-2 运行结果

(2) 输入程序。将程序输入到编辑区。

(3) 编译程序。选择 Build→Compile。弹出如图 1-4 所示对话框。单击“是”按钮，自动生成默认的工程空间，如图 1-5 所示。

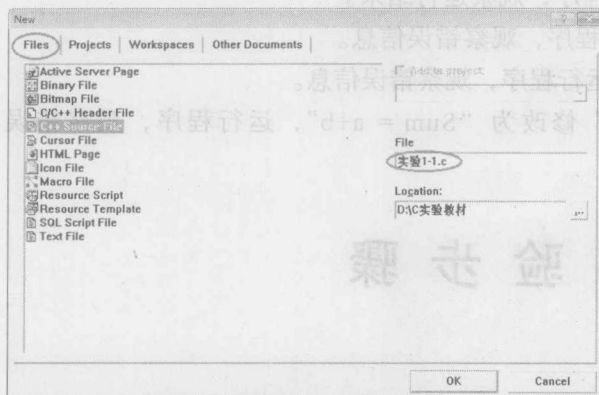


图 1-3 选择 File→C++ Source File

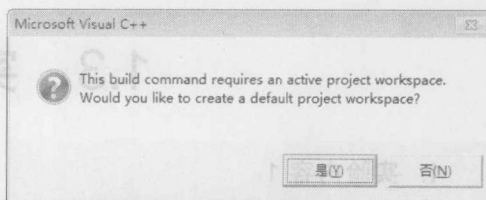


图 1-4 “生成默认工程空间”对话框

(4) 连接程序。选择 Build→Rebuild All。若有错，修改错误到无错误为止。

(5) 运行程序。选择 Build→Execute (或【Ctrl+F5】)。输出结果如图 1-6 所示。

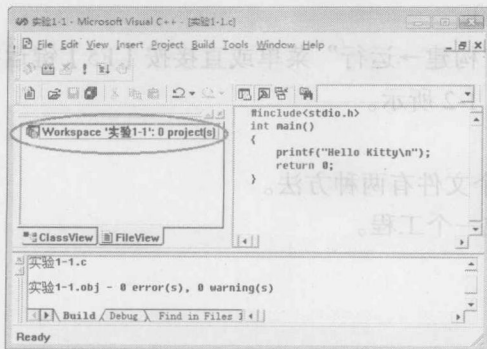


图 1-5 左边画圈处查看自动生成的工程空间

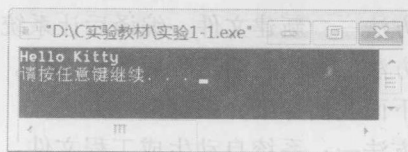


图 1-6 查看结果

方法二：先新建工程，后新建文件。

(1) 新建工程。选择 File→New，选择 Projects 选项卡下的 Win32 Console Application，如图 1-7 所示。

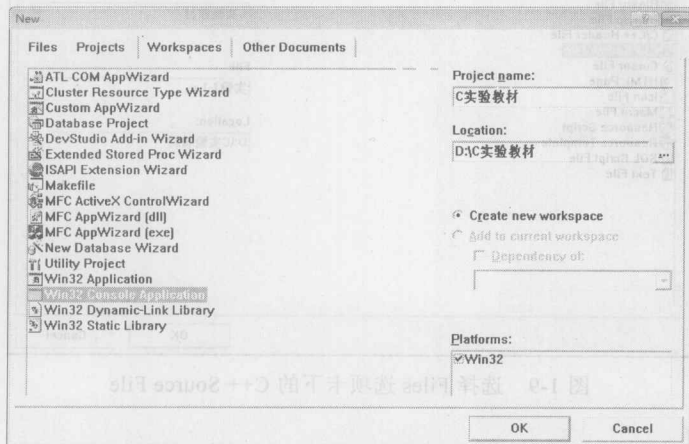


图 1-7 选择 Win32 Console Application

在 Project name 下输入您的工程名（如 C 实验教材），并选择该工程文件所存放的路径（如 D:\），最后单击 OK 按钮。

根据自己需求选择，可选第一项新建一个空的工程，如图 1-8 所示。单击 Finish 按钮完成新建工程。

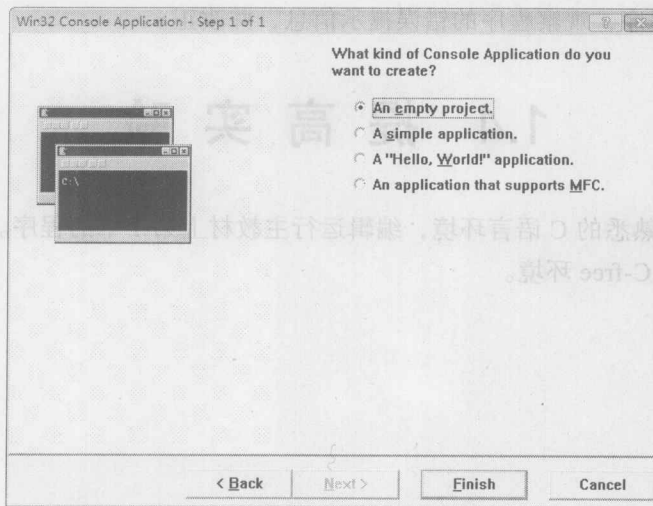


图 1-8 默认新建一个空工程

(2) 新建文件。选择 File→New。之后在 File 选项卡下选择 C++ Source File，输入文件名，如实验 1-1.c，如图 1-9 所示。请注意：若不写扩展名.c 系统也会给文件一个扩展名.cpp，但是即使在程序完成正确的情况下，编译会出现错误。原因是 VC++ 的编译器认为.c 的为 C 程序，.cpp 的为 C++程序，C 程序与 C++程序中同样的函数在编译后的 obj 文件中的 symbol 是不同的，所以以 C 方式编译的 obj 文件与以 C++方式编译的 obj 文件无法成功链接。

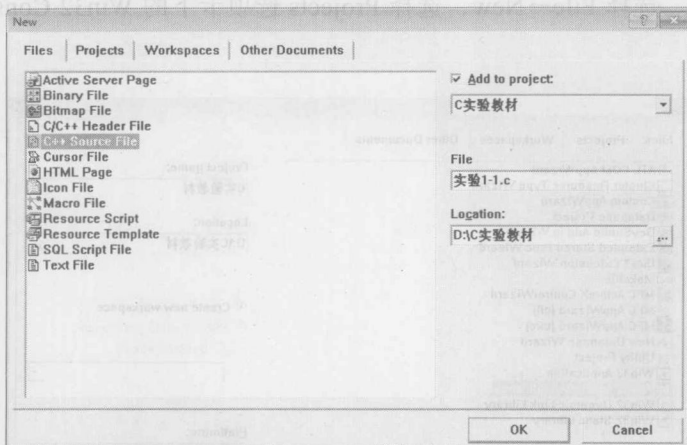


图 1-9 选择 Files 选项卡下的 C++ Source File

- (3) 输入程序。
- (4) 编译程序。选择 Build→Compile。
- (5) 连接程序。选择 Build→Rebuild (或【F7】)。
- (6) 运行程序。选择 Build→Execute (或【Ctrl+F5】)。

3. 实验内容 3

在两种环境中分别编辑运行实现两个整数加法运算的程序，并按要求修改程序。

4. 实验内容 4

根据要求修改程序，观察程序的错误提示信息，并改正。

1.4 提高实验

1. 选择一种较熟悉的 C 语言环境，编辑运行主教材上第 1 章的程序。
2. 进一步熟悉 C-free 环境。

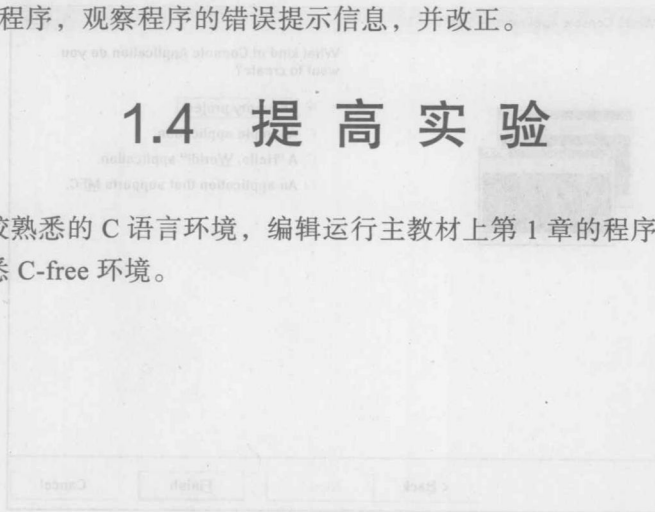


图 1-8 新建一个工程

实验 2

数据运算

2.1 实验目的

1. 了解 C 语言中数据类型的意义。
2. 理解常用运算符的意义。
3. 掌握 C 语言表达式的书写规则。
4. 掌握 C 语言程序基本输入、输出的格式。
5. 掌握逗号运算和条件运算的基本规则。
6. 掌握关系运算和逻辑运算的基本规则。

2.2 实验内容

1. 验证整型数据与字符型数据之间的互用性及限制。
2. 观察分析整型数据、无符号整型数据、长整型数据的使用。
3. 自加(++)和自减(--)运算符的使用。
4. 各种数据类型的综合使用。
5. 输入、输出语句的使用。
6. 逗号运算表达式和条件运算表达式的使用。
7. 关系运算表达式和逻辑运算表达式的使用。

2.3 实验步骤

1. 实验内容 1

按照要求修改程序，观察程序结果。程序如下：

```
/* 实验 2-1_1.C --- data type(char) definition */
#include <stdio.h>
int main()
{
```

```

char c1, c2;
c1 = 'a';
c2 = 'b';
printf("%c %c\n", c1, c2);
return 0;
}

```

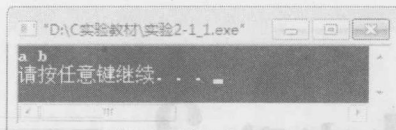


图 2-1 实验 2-1_1.C 的运行结果

程序运行结果如图 2-1 所示, c1、c2 都按照字符格式输出。

修改 1: 在此基础上增加一个语句: `printf("%d %d\n", c1, c2);`, 再运行, 并分析结果。程序如下:

```

/* 实验 2-1_2.C --- data type(char) definition */
#include <stdio.h>
int main()
{
    char c1, c2;
    c1 = 'a';
    c2 = 'b';
    printf("%c %c\n", c1, c2);
    printf("%d %d\n", c1, c2);
    return 0;
}

```

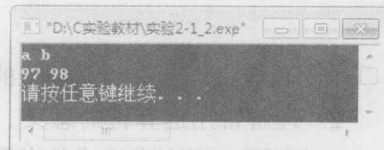


图 2-2 实验 2-1_2.C 的运行结果

运行结果如图 2-2 所示, c1、c2 也可以按照整数格式输出。(字母 a 的 ASCII 码是 97)

修改 2: 再将第 2 行 `char c1, c2;` 改为: `int c1, c2;` 再运行, 并分析结果。程序如下:

```

/* 实验 2-1_3.C --- data type(char) definition */
#include <stdio.h>
int main()
{
    int c1, c2;
    c1 = 'a';
    c2 = 'b';
    printf("%c %c\n", c1, c2);
    printf("%d %d\n", c1, c2);
    return 0;
}

```

运行结果如图 2-3 所示。c1、c2 可以按照字符格式、整数格式输出。

修改 3: 再将第 3、4 行改为:

```

c1=a; /* 不用单引号 */
c2=b;

```

再运行, 分析其运行结果。编译通不过, 不能运行。

修改 4: 再将第 3、4 行改为:

```

c1="a"; /* 用双引号 */
c2="b";

```

再运行, 分析其运行结果。编译通不过, 不能运行。

修改 5: 再将第 3、4 行改为:

```

c1=300; /* 输入大于 255 的整数 */
c2=400;

```

再运行, 分析其运行结果。

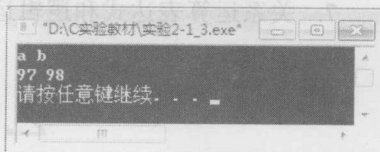


图 2-3 实验 2-1_3.C 的运行结果

程序如下:

```
/* 实验 2-1_4.C --- data type(char) definition */
#include <stdio.h>
int main()
{
    int c1,c2;
    c1=300;
    c2=400;
    printf("%c %c\n", c1, c2);
    printf("%d %d\n", c1, c2);
    return 0;
}
```

运行结果如图 2-4 所示, c1、c2 按照字符格式输出时, 只保留 1 个字节的数值, 并输出和 ASCII 相对应的字符。

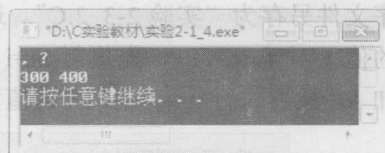


图 2-4 实验 2-1_4.C 的运行结果

2. 实验内容 2

对照程序和运行结果进行分析。

程序如下:

```
/* 实验 2-2.C --- data and expression */
#include <stdio.h>
int main()
{
    int i,j,m,n;
    scanf("i=%d,j=%d",&i,&j);
    m=(i+1,j+2);
    n=i>j?i-1:j+1;
    printf("i=%d,j=%d\n",i,j);
    printf("m=%d,n=%d\n",m,n);
    return 0;
}
```

修改 1: 设 $i=10$, $j=90$, 该程序的运行结果如图 2-5 所示。 m 的值为 92 (即 $j+2$), n 的值为 91 (即 $j+1$, 因为 $i>j$ 不成立)。

修改 2: 设 $i=80$, $j=20$, 该程序的运行结果如图 2-6 所示。 m 的值为 22 (即 $j+2$), n 的值为 79 (即 $i-1$, 因为此时 $i>j$ 成立)。



图 2-5 实验 2-2.C 的运行结果之一

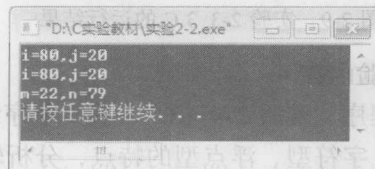


图 2-6 实验 2-2.C 的运行结果之二

3. 实验内容 3

按照要求修改程序, 观察程序结果, 分析自增(++)和自减(--)运算规律。

程序如下:

```
/* 实验 2-3_1.C --- ++, -- */
#include <stdio.h>
int main()
{
    int a,b;
    float c;
    char c1,c2;
    a=5; b=6;
    c1='a'; c2='b';
}
```

```

inti,j,m=0,n=0;
i=8;
j=10;
m+=i;
n-=j;
printf("i=%d,j=%d,m=%d,n=%d\n",i,j,m,n);
return 0;
}

```

运行该程序（实验 2-3_1.C），其结果如图 2-7 所示。观察 i、j、m、n 各变量的值。

修改 1：将语句“m+=i;”改为“m=i++;”，再将语句“n-=j;”改为“n=++j;”，并将程序文件另存为“实验 2-3_2.C”。运行该程序，其结果如图 2-8 所示。观察 i、j、m、n 各变量的值（对于 m=i++，先执行 m=i，此时 i 的值为 8，然后 i 自加 1；对于 n=++j，先执行 j 自加 1，然后执行 n=j，此时 j 的值为 11）。

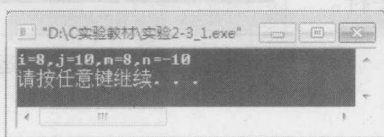


图 2-7 实验 2-3_1.C 的运行结果

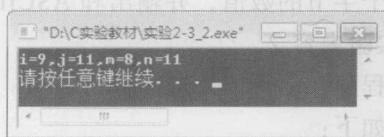


图 2-8 实验 2-3_2.C 的运行结果

修改 2：将程序“实验 2-3_1.C”中的 printf 语句改为：

```
printf("%d,%d,%d,%d\n", ++i, ++j, m, n);
```

并将程序文件另存为“实验 2-3_3.C”。运行该程序，其结果如图 2-9 所示。观察 i、j、m、n 各变量的值，并与图 2-7 的运行结果进行比较。（在输出时，i 先自加 1，然后输出 i 的值 9；j 也先自加 1，然后输出 j 的值 11。）

修改 3：再将程序“实验 2-3_3.C”中的语句“m+=i;”改为“m+=i++;”、语句“n-=j;”改为“n-=--j;”，并将程序文件另存为“实验 2-3_4.C”。运行该程序，其结果如图 2-10 所示。观察 i、j、m、n 各变量的值，并与图 2-9 的运行结果进行比较（先执行 m=i，即 m=8；然后 i 有 2 次自加 1，输出 i 的值 10；j 先自减 1，使得 n 为-9，然后 j 自加 1，最后输出 j 的值 10）。

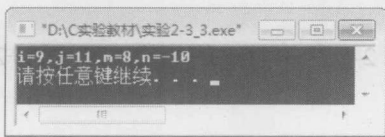


图 2-9 实验 2-3_3.C 的运行结果

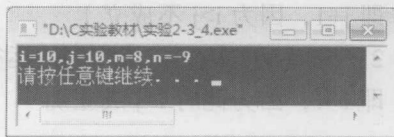


图 2-10 实验 2-3_4.C 的运行结果

4. 实验内容 4

运行程序“实验 2-4_1.C”，观察程序结果并分析不同数据类型的混合使用规则（提示，根据整型、字符型、浮点型的特点，分析输出结果）。

程序如下：

```

/* 实验 2-4_1.C --- data type */
#include <stdio.h>
int main()
{
    int a,b;
    float d,e;
    char c1,c2;
    a=61; b=62;
    c1='a'; c2='b';
}

```

```

d=3.56; e=-6.87;
printf("a=%d,b=%d,c1=%c\n",a,b,c1);
printf("c2=%c,d=%6.2f,e=%6.2f\n",c2,d,e);
return 0;
}

```

程序“实验 2-4_1.C”的执行结果如图 2-11 所示。

运行程序“实验 2-4_2.C”，观察程序结果并分析（提示：假设输入的字符为 x 和 y。根据整型、无符号整型、字符输入/输出函数的特点，分析输出结果）。

程序如下：

```

/* 实验 2-4_2.C --- data type */
#include <stdio.h>
int main()
{
    int a, b;
    char c1, c2;
    unsigned int p, q;
    a=50000; b=-60000;
    c1=getchar(); c2=getchar();
    p=32768; q=800000000;
    printf("a=%d,b=%d\n",a,b);
    printf("p=%u,q=%u\n",p,q);
    putchar(c1);putchar(c2);
    printf("\n");
    return 0;
}

```

程序“实验 2-4_2.C”的执行结果如图 2-12 所示。

5. 实验内容 5

按照要求修改程序，观察程序结果并分析逻辑表达式的使用规则。

程序如下：

```

/* 实验 2-5_1.C --- logical expression */
#include <stdio.h>
int main()
{
    int a,b,c;
    a=10;
    b=20;
    c=0;
    printf("%d,%d\n",!a*b,!c);
    printf("%d,%d\n",a&&b-10||c,a+c>b&&b>a);
    printf("%d,%d\n",a||(c=a+b),c);
    return 0;
}

```

运行程序“实验 2-5_1.C”，其执行结果如图 2-13 所示（提示，逻辑值“真”用 1 表示，逻辑值“假”用 0 表示。反过来，0 表示“假”，非 0 表示“真”。没有执行语句“c=a+b”）。

修改 1：使 b=0，c=20，将程序文件另存为“实验 2-5_2.C”。再运行该程序并分析结果。其执行结果如图 2-14 所示（提示：!a*b 等价于 (!a)*b，没有执行语句“c=a+b”）。

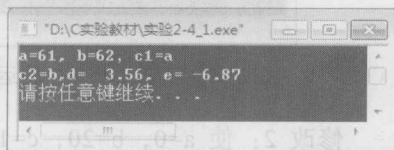
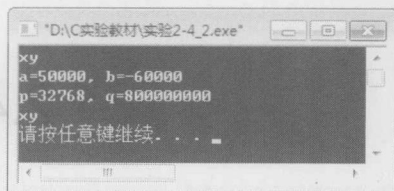


图 2-11 实验 2-4_1.C 的运行结果



图 2-12 实验 2-4_2.C 的运行结果



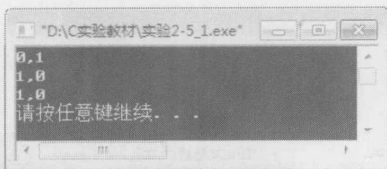


图 2-13 实验 2-5_1.C 的运行结果

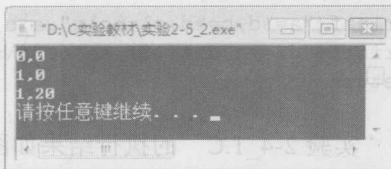


图 2-14 实验 2-5_2.C 的运行结果

修改 2: 使 $a=0$, $b=20$, $c=10$, 将程序文件另存为“实验 2-5_3.C”。再运行该程序并分析结果。其执行结果如图 2-15 所示 (提示, `printf("%d,%d\n", a|(c=a+b), c);`语句, 先输出 c 的值 10, 然后执行 $c=a+b$, 使 c 的值为 20)。

修改 3: 使 $a=0$, $b=20$, $c=0$, 将程序文件另存为“实验 2-5_4.C”。再运行该程序并分析结果。其执行结果如图 2-16 所示 (提示, `printf("%d,%d\n", a|(c=a+b), c);`语句, 先输出 c 的值 0, 然后执行 $c=a+b$, 使 c 的值为 20)。

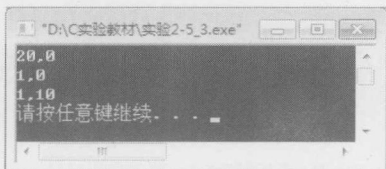


图 2-15 实验 2-5_3.C 的运行结果

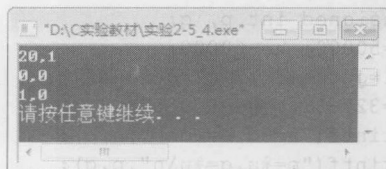


图 2-16 实验 2-5_4.C 的运行结果

2.4 提高实验

1. 设 $a=0$, $b=1$, $c=2$, 设计一个简单的程序, 验证下面两个表达式的结果。

表达式 1: $a==10 \&\& b|| c==5$

表达式 2: $++a \&\& b-- \&\& (c=5)$

2. 设 m 、 n 为整数, 设计一个简单的程序, 根据 $m>n$ 、 $m=n$ 、 $m<n$ 的不同情况, 验证下面两个表达式的结果。

表达式 1: $i=m+1, j=n-3, k=m+n$

表达式 2: $m \geq n ? m+50 : n-18$