

《C 语言程序设计》实验指导书

2009 年 8 月

长江大学电信学院计算机课程组

前 言

C 语言是现代最流行的通用程序设计语言之一，它既具有高级程序设计语言的优点，又具有低级程序设计语言的特点，既可以用来编写系统程序，又可以用来编写应用程序。因此，C 语言正在被迅速地推广和普及。上机实验是该课程教学的一个重要环节，因此要求学生做一定数量的上机实验。本指导书可增强同学上机实验的针对性。整个教学和实验中，我们采用 VC++6.0 作为实验环境，强调学生切实培养动手实践能力，掌握调试程序的方法，通过调试理解 C 语言程序运行的过程以及 C 语言的语法规则，为后续的课程设计，计算机等级考试及其他应用做好充分的准备。

本实验指导书通过大量的实例，循序渐进地引导学生做好各章的实验。根据实验教学大纲，我们编写了十个实验，每个实验又分成五部分内容：

- (1) 实验目的；
- (2) 相关知识点
- (3) 实验内容一（偏重于：验证及程序分析）；
- (4) 实验内容二（偏重于：综合及设计类）；
- (5) 扩展内容（偏重于：灵活应用）。

其中：实验内容一要求必做，实验内容二和扩展内容可由老师选择。

在实验之前，要求学生对实验作好预习工作。在实验中，学生根据实验指导中的内容进行验证与总结，然后再去完成实验内容中安排的任务。

实验结束后，应及时提交实验报告，其内容应包括如下 6 项，具体可根据实验内容和实验要求进行增删。实验报告要求包含如下：

实验内容一：

- 1、实验题目
- 2、程序运行结果及分析

实验内容二：

- 1、实验题目
- 2、设计思路或算法分析
- 3、程序源代码
- 4、程序运行结果及分析

扩展内容：

- 1、实验题目
- 2、设计思路或算法分析
- 3、流程图
- 4、程序源代码
- 5、程序运行结果及分析
- 6、存在的问题。

目 录

实验一	C 程序的运行环境	1
实验二	数据类型、运算符和表达式	5
实验三	顺序结构和逻辑结构程序设计	10
实验四	循环控制	14
实验五	数组	18
实验六	函数一	25
实验七	函数二	29
实验八	指针	34
实验九	结构体、共用体和位运算	38
实验十	文件	53
附录 1:	常见错误提示信息的英汉对照	67
附录 2:	VISUAL C++ 6.0 集成开发环境的使用	71

实验一 C 程序的运行环境

一、 实验目的

1. 了解和使用 Visual C++ 6.0(简称 VC)集成开发环境;
2. 熟悉 VC 的控制台应用程序的开发、调试步骤和方法;
3. 掌握 C 程序的编辑、编译、连接和运行过程;
4. 了解工程、文件、工作区等基本概念;
5. 能理解常见错误的提示信息, 并根据提示进行改错。

二、 相关知识点

VC 集成开发环境;

C 语言的编辑、编译、连接和运行过程;

C 语言的调试方法和工具;

C 语言的基本组成特点。

术语: “控制台应用程序”是一个在 DOS 窗口中运行的基于字符的程序。由于这种模式的应用程序比 Windows 程序简单, 我们先选择利用 Visual C++来建立这样一个应用, 这样使得我们可以将精力先投入到学习使用 C 或者 C++编程语言, 而不需要把过多的精力投入到学习复杂的 Windows 编程中去。

三、 实验内容一（验证及程序分析）

Eg1-1:使用 Visual C++ 6.0 集成环境来编辑、编译并运行下面的示例程序:

```
#include <stdio.h>
void main()
{
printf("This is my first C program! \n");
}
```

【说明:】本题要求完成一个字符串的输出, 并且是“依葫芦画瓢”哦!

实验步骤:

(1) 启动 Visual C++开发环境

(2) 创建一个工程

- a. 单击 File 菜单中的 New 选项, 显示出新建对话框
- b. 选择 Win32 Console Application。
- c. 选择项目所在路径及输入项目的名称
- d. 在建立一个什么类型的控制台应用程序中, 选中“一个空工程”, 按“完成”

按钮, 至此, 已经建立好工程相关的文件。

(3) 建立 C++源程序文件

- a. 选菜单命令 Project|Add to Project|New, 弹出 New 对话框
- b. 在 New 对话框的 Files 选项卡中选择 C++ Source File, 并填入文件名称, 单击 OK 按钮, 完成新建 C++源程序文件

(4) 编辑 C++源程序文件

- a. 在文件编辑窗口中输入代码
- b. 完成后, 选择菜单 File|Save 保存这个文件

(5) 编译、链接并运行可执行程序

- a. 选择菜单命令 Compile, 建立目标代码文件;

如果程序有语法错误, 则屏幕下方的状态窗口中会显示错误信息, 根据这些错误信息对源程序进行修改后, 重新选择菜单命令 Compile 建立目标代码文件。

- b. 选择菜单命令 Build, 建立可执行程序

如果你正确输入了源程序, 此时便成功地生成了可执行程序。

如果程序有链接错误, 则屏幕下方的状态窗口中会显示错误信息, 根据这些错误信息对源程序进行修改后, 重新选择菜单命令 Build 建立可执行程序。

- c. 选择菜单命令 Run, 运行程序, 观察屏幕显示内容。

【说明】有时也可直接选择菜单命令 Run, 将“编译、链接并运行”几个步骤合为一起。

(6) 关闭工作空间

选择菜单命令 File|Colse WorkSpace 关闭工作空间。

Eg1-2: 使用 Visual C++ 6.0 集成环境来编辑、编译并运行下面示例程序:

(1) 源程序

```
#include <stdio.h>
void main()
{
    int a,b,temp;
    a=3;    b=4;
    temp=a;    /* 将 a 的值存储起来 */
    a=b;      /* 将 b 的值转到 a 中, 实现 b 向 a 转换 */
    b=temp;   /* 将记录 a 原始值的变量 temp, 赋值给 b, 实现 a 向 b 转换 */
    printf( "a=%d, b=%d\n", a, b);
}
```

(2) 实验步骤: 建立一个控制台应用程序项目 Exam1-2, 向其中添加一个 C++源文件

Eg1-2. cpp, 输入程序, 检查一下确认没有输入错误, 选择菜单命令编译源程序, 再选择运行程序。

(3) 运行结果:

Eg1-3:求 1+2+3+4+5+……+100 累加和

【说明:】本题要求完成经典的 C 算法——累加求和, 当我们把计算方法教给计算机后, 剩下计算的问题就让计算机来算好了, 计算机像一头老黄牛一样不厌其烦地做事情。以后会有相似的算法如“连乘”, “求级数”, “求 Π ”, “求阶乘”等。

(1) 源程序:

```
#include <stdio.h>
void main()
{
    int i, sum=0;
    for(i=1; i<=100; i++)
        sum=sum+i;      /*累加求和 */
    printf("sum=%d\n", sum);
}
```

(2) 在 VC++ 编辑环境下编辑源程序。

(3) 编译直到程序没有错误。

(4) 运行结果:

四、 实验内容二 (综合及设计类)

Eg1-4: 输入 2 个数据, 计算它们的和, 并打印输出在屏幕上。

【说明:】本题要求完成基本的键盘输入和屏幕输出的练习

```
#include <stdio.h>
void main()
{
    int a, b, s;          /* 定义变量 */
    printf(" Enter two number a, b=?\n");
    scanf("%d%d", &a, &b); /*调用标准函数, 要求用户输入 2 个数据 */
    s=a+b;
    printf("s=%d\n", s);
}
```

(2) 在 VC++ 编辑环境下编辑源程序。

(3) 编译直到程序没有错误。

(4) 从键盘输入 10 20↵, 运行结果如何?

(5) 从键盘输入 10, 20↵, 运行结果如何?

(6) 任意输入 2 个整数，之间用“空格”，或者“回车键”，或者“Tab 键”分隔，分析结果如何？

五、 扩展内容

Eg1-5: 编写一个 C 程序：输入 a, b, c 三个值，输出其中最大者。

Eg1-6: 编写一个 C 程序：计算 8!

【小结：】多分析阅读别人写的程序，多自己动手编写一些小程序，多上机调试运行程序，做到这 3 个“多”字，学习好 C 程序设计就不难了。

实验二 数据类型、运算符和表达式

一、 实验目的

- 1、掌握 C 语言数据类型，熟悉如何定义一个整型、字符型和实型的变量，以及对它们赋值的方法；
- 2、掌握不同的数据类型之间赋值的规律；
- 3、学会使用 C 的有关算术运算符，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、进一步熟悉 C 程序的编辑、编译、连接和运行的过程。

二、 相关知识点

C 语言基本数据类型包括：数值类型（整型，实型），字符类型等，是我们进行基本数据运算的基础。用定点或浮点方式表示。

常量就是不能改变的量，它们的值在程序中不可能发生改变，我们可以用一个标识符来代表一个常量，这叫做符号常量。

变量是在程序运行过程中其值可以发生改变的量。每一个变量都有名字。要注意区分变量名和变量值

常量和变量都有各种类型。如整型，实型，字符型等。

整型常量就是整数。C 语言中整数有三种表示方法：十进制整数、八进制整数或十六进制整数。

整型变量在 C 语言中能分成 4 种：基本型 `int`、短整型 `short`、长整型 `long` 和无符号型 `unsigned`。

实型常量就是我们通常所说的小数，有两种表示方式：小数形式、指数形式。

实型变量在 C 语言中有两种：`float` 和 `double`。

字符型数据包括：字符型常量（是用单引号括起来的一个字符）、字符型变量（用来存放字符常量）、字符串常量（多个字符串联在一起，用一对双引号括起来）组成。将一个字符常量存放到一个字符变量中，实际上只是把该字符的 ASCII 码存放到此字符变量中，字符串常量在内存中的存放方式有一个特殊的地方，系统会自动给每个字符串常量的结尾加上一个 `'\0'` 字符，作为字符串结束标志。

变量在使用前必须定义。定义变量的格式如下：

变量类型 变量名 1，变量名 2，...；

任何一个变量，必须先赋值，后使用。

C 语言提供了非常丰富的运算符，包括：算术运算符（`+`、`-`、`*`、`/`、`%`），关系运算符（`>`、`<`、`==`、`>=`、`<=`、`!=`），逻辑运算符 `!`、`&&(and)`、`||(or)`）、位运算符（`<<`、`>>`、`~`、`|`、`^`）、赋值运算符（`=` 及其扩展赋值运算如 `+=`、`*=`、`-=`、`/=` 等）、条件运算符（`?`：通常用条件语句）、逗号运算符（`,`）、指针运算符（`&` 和 `*`）、求字节数运算符（`sizeof`（））、强制类型转换运算符（`int`、`float(5%3)`）、分量运算符（`.->`）、下标运算符（`[]`）和其它（如函数调用运算符（`()`）。其组合运算符符合其运算优先级的限制。

三、 实验内容一（验证及程序分析）

Eg2-1: 输入并运行下面的程序

```
#include <stdio.h>
void main( )
{
    char c1,c2 ;
    c1='a';
    c2='b';
    printf("%c%c\n",c1,c2);
}
```

- (1) 运行此程序。
- (2) 在此基础上增加一条语句：
printf(“%d%d\n”,c1,c2);
再运行，并分析结果。
- (3) 将第 2 行改为：int c1,c2;
再使之运行，并观察结果。
- (4) 再将第 3、4 行改为：c1=a; c2=b;
再使之运行，分析其运行结果。
- (5) 再将第 3、4 行改为：c1= “a” ; c2= “b” ;
再使之运行，分析其运行结果。
- (6) 再将第 3、4 行改为：c1=280; c2=350; /*用大于 255 的整数*/
再使之运行，分析其运行结果。

Eg2-2: 输入并运行程序（上机前先分析程序写出结果，上机后作对照）：

```
#include <stdio.h>
void main( )
{
    char c1='a';
    char c2='b';
    char c3='c';
    char c4='\101';
    char c5='\116';
    printf("a=%cb%c\tc%c\tabc\n",c1,c2,c3);
    printf("\t\b%c %c",c4,c5);
}
```

可以改变程序中各变量的值，以便比较其差异。

Eg2-3: 输入并运行下面的程序：

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

```

{
    int a,b ;
    unsigned c,d ;
    long e,f ;
    a=100 ;
    b=-100 ;
    e=50000 ;
    f=32767 ;
    c=a ;
    d=b ;
    printf("%d,%d \n",a,b);
    printf("%u,%u \n",a,b);
    printf("%u,%u \n",c,b);
    c=a=e ;
    d=b=f ;
    printf("%d,%d \n",a,b);
    printf("%u,%u \n",c,d);
}

```

请对照程序和运行结果分析：

(1)将一个负整数赋给一个无符号的变量，会得到什么结果。画出它们在内存中的表示形式。

(2)将一个大于 32765 的整数赋给整型变量，会得到什么结果。画出它们在内存中的表示形式。

(3)将一个整数赋给无符号的变量，会得到什么结果（分别考虑该整数的值大于或等于 65535 以及小于 65535 的情况）。画出它们在内存中的表示形式。

同学们可以改变程序中各变量的值，以便比较。例如： a = 65580, b = -40000, e = 65535, f = 65580。

Eg2-4: 输入下面程序：

```

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

```

(1) 运行程序，注意 i, j, m, n 各变量的值，分别作以下改动并运行。

(2) 将程序中 m, n 的符值改为：m=i++；n=++j；再运行。

(3) 将程序改为：

```

#include <stdio.h>

```

```

void main( )
{ int i, j;
  i=8; j=10;
  printf(“%d,%d”, i++, j++);
}

```

运行并分析结果。

(4) 在 (3) 的基础上，将 printf 语句改为：

```
printf(“%d,%d”, ++i, ++j);
```

运行并分析结果。

(5) 再将 printf 语句改为：

```
printf(“%d,%d,%d,%d”, i, j, i++, j++);
```

运行并分析结果。

(6) 将程序改为：

```

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

```

运行并分析结果。

实验报告要求对照运行结果分析以上各程序的算法，及产生该结果现象的原因。

四、 实验内容二（综合及设计类）

Eg2-5: 将“China”译成密码，译码规律是：用原来字母后面的第 4 个字母代替原来的字母。例如，字母“A”后面第 4 个字母是“E”，用“E”代替“A”。因此，“China”应译为“Glmre”。

(1) 请编一程序，用赋初值的方法使 c1、c2、c3、c4、c5 五个变量的值分别为‘c’、‘h’、‘i’、‘n’、‘a’，经过运算，使 c1、c2、c3、c4、c5 分别变为‘G’、‘l’、‘m’、‘r’、‘e’，并输出，分析是否符合要求。

(2) 改变 c1、c2、c3、c4、c5、的初值为：‘T’、‘o’、‘d’、‘a’、‘y’，对译码规律作如下补充：‘W’用‘A’代替，‘X’用‘B’代替，‘Y’用‘C’代替，‘Z’用‘D’代替。修改程序并运行。

(3) 将译码规律修改为：将字母用它前面的第 4 个字母代替，例如：‘E’用‘A’代替，‘Z’用‘U’代替，‘D’用‘Z’代替，‘C’用‘Y’代替，‘B’用‘X’代替，‘A’用‘V’代替。修改程序并运行。

Eg2-6: 求下列表达式的值，并编写程序，上机进行验证。

① $x+a\%3*(int)(x+y)\%2/4$ 设 $x=2.5, a=7, y=4.7$

② $(float)(a+b)/2+(int)x\%(int)y$ 设 $a=7, b=3, x=3.5, y=2.5$

Eg2-7: 求下列表达式的值，并编写程序，上机进行验证。

设 $a=12, n=5$ 。

① $a+=a$

② $a-=2$

③ $a*=2+3$

④ $a/=a+a$

实验三 顺序结构和逻辑结构程序设计

一、 实验目的

- 1、熟练掌握 C 语言中赋值语句的使用方法；
- 2、掌握格式输入输出 scanf() 函数、printf() 函数的调用语法；整型、实型、字符型数据的输入输出格式。
- 3、掌握字符输入输出函数 getchar()、putchar() 函数的调用语法。
- 4、熟练掌握 if 语句的 3 种形式及对应的语法规则、执行流程，switch 语句的语法规则、执行流程。
- 5、掌握 6 个关系运算符、3 个逻辑运算符的基本概念、基本属性、运算规则。

二、 相关知识点

scanf()函数 和 printf()函数的意义及用法；
getchar()函数 和 putchar()函数的意义及用法；
gets()函数 和 puts()函数的意义及用法；
if 语句的常见三种形式；
switch 语句的语法规则及执行过程；
break 语句和 continue 语句的使用。

三、 实验内容一（验证及程序分析）

Eg3-1: 从键盘输入两个 0 到 127 的整数，求两数的平方差并输出其值以及这两个整数的 ASCII 码对应的字符。

(1) 算法分析：标准 ASCII 码值范围是 0 到 127，所以在这范围内的数可以以字符形式输出，实现方法是用格式输出函数并用格式符 c 控制输出。

(2) 源程序：

```
#include <stdio.h>
void main( )
{
    int x,y,sub ;
    scanf("%d%d",&x,&y);          /* 变量前一定要加取地址符& */
    sub=x*x-y*y ;
    printf("%c,%c,%d\n",x,y,sub);
}
```

(3) 在 VC++ 编辑环境下编辑源程序。

(4) 编译直到程序没有错误。

(5) 从键盘输入

100 50↵，运行结果如何？

(6) 输出结果：

Eg3-2: 输入三角形的三边长，判断这个三角形是否是直角三角形。

(1) 算法分析：直角三角形斜边最长，要先找出三边中最长的边，判断最长边的平方是否等于其余两边的平方和，若相等就是直角三角形。

(2) 源程序：

```
#include <stdio.h>
void main( )
{
    int a,b,c,t; /* 三边设为 a,b,c,t 是用于交换的中间变量 */
    scanf("%d,%d,%d",&a,&b,&c);
    if(a<b)
    { /* a 中放 a,b 中较长边 */
        t=a;   a=b;   b=t;
    }
    if(a<c)
    { /* a 中放 a,b,c 中的最长边 */
        t=a;   a=c;   c=t;
    }
    if(a*a==b*b+c*c)
        printf("Y");
    else
        printf("N");
}
```

(3) 在 VC++ 编辑环境下编辑源程序。

(4) 编译直到程序没有错误。

(5) 从键盘输入：3，4，5↵，运行结果如何？

(6) 从键盘输入：30，40，90↵，运行结果如何？

四、 实验内容二（综合及设计类）

Eg3-3: 有一函数

$$y = \begin{cases} x & (x < 1) \\ 2x - 1 & (1 \leq x < 10) \\ 3x - 11 & (x \geq 10) \end{cases}$$

用 scanf 函数输入 x 的值，求 y 值。

(1) 算法提示：

- 1) 定义实变量 x,y
- 2) 提示输入 "Please enter x:"
- 3) 读入 x
- 4) 判断 x 所在的区间，对应出 y 的计算公式并求值（进一步细化）
- 5) 打印结果。

(2) 编写程序:

(3) 测试结果:

x	y
-1	
5	
10	

Eg3-4: 给出一个百分制成绩, 要求输出成绩等级“ A ”、“ B ”、“ C ”、“ D ”、“ E ”。90 分以上为“ A ”, 81~89 分为“ B ”, 70~79 为“ C ”, 60~69 为“ D ”, 60 分以下为“ E ”。当输入数据大于 100 或小于 0 时, 通知用户“输入数据错”, 程序结束。请分别用 if 语句和 switch 语句加以实现。

(1) 算法提示:

- 1) 定义百分制成绩(实变量)score
- 2) 提示输入 "Please enter score:"
- 3) 读入 score
- 4) 判断 score 所在的区间, 对应出 score 的等级并打印相应等级(进一步细化)。

(2) 用 if 语句实现的源程序:

(3) 用 switch 语句实现的源程序:

(4) 测试结果:

score	打印等级
65	
54	
77	
89	
92	
100	

五. 扩展内容 (可选)

Eg3-5: 输入一个三位数, 若此数是水仙花数输出“Y”, 否则输出“N”, 若输入值不是三位数输出“data error”。

提示: 水仙花数是一个三位数, 组成这个三位数的三个数字的立方和与这个三位数相等。如: $153=1^3+5^3+3^3$ 。判断是否是水仙花数需把构成三位数的三个数字分离出来并存入变量。

Eg3-6: 输入两个操作数和一个运算符, 编程实现简单的计算器。

Eg3-7: 编程求圆周长、圆面积、圆球表面积、圆球体积、圆柱体积。在程序中调用 scanf() 函数通过键盘输入半径, 圆柱高, 输出计算结果, 输出要求有文字说明, 答案精确到小数点后两位。

提示:

1) 定义符号常量 PI

2) 计算圆周长、圆面积、圆球表面积、圆球体积、圆柱体积公式分别表示为: $\text{circle}=2*\text{PI}*r$, $\text{area}=\text{PI}*r*r$, $\text{surface}=4*\text{PI}*r*r$, $\text{globe}=\text{PI}*r*r*r*4/3$, $\text{column}=\text{area}*h$

实验四 循环控制

一、 实验目的

- 1、熟练掌握 for 语句、while 语句、do-while 语句实现循环的方法；
- 2、理解循环嵌套及其使用方法；
- 3、掌握 break 语句与 continue 语句的使用；
- 4、掌握用循环实现一些常用算法（如穷举、迭代、递推等）。

二、 相关知识点

for 语句是 C 语言中最常用的循环结构。一般形式为：

```
for (表达式 1; 表达式 2; 表达式 3) 语句;
```

执行过程如下：

- 1、计算表达式 1。
- 2、计算表达式 2。如为“真”（非 0），则执行 for 后的语句；如为“假”，则跳出循环体。
- 3、如表达式 2 为“真”，则执行完循环体后，计算表达式 3 的值。转回步骤 2。

最常用的形式是：

```
for(循环变量赋初值; 判断条件; 循环变量增值)
    循环体语句;
```

while 语句用来实现“当型”循环结构，就是当满足某个条件时进行循环。形式为：

```
while (表达式)
    循环体语句;
```

当表达式为“真”（非 0 值）时反复执行后面的语句。特点是先判断，后执行。一般来说，while 中的语句是复合语句，用 {} 括起来，叫做循环体。

do-while 语句用来实现“直到型”循环，就是进行循环直到某个条件不满足。形式为：

```
do
    循环体语句;
while (表达式);
```

当表达式为“真”（非 0 值）时反复执行 do 后面的语句，特点是先执行，后判断（注意区分与 while 语句的区别）。

while 语句和 do-while 语句可以很容易的实现互相转换。一般情况下，如果 while 语句和 do-while 语句的循环体部分一样，则它们的运行结果也是一样的。但是，如果判断条件一开始就不满足，则运行结果不一样，因为此时 while 语句不执行循环体，而 do-while 语句要执行一次。

辅助控制语句 break、continue 语句

break 功能：在循环语句和 switch 语句中,终止并跳出循环体或开关体。**break** 只能终止并跳出最近一层的结构，**break** 不能用于循环语句和 switch 语句之外的任何其它语句之中。

continue 功能：结束本次循环，跳过循环体中尚未执行的语句，进行下一次是否执行循环体的判断，仅用于循环语句中。

三、 实验内容一（验证及程序分析）

Eg4-1: 输入一组整数，统计其中奇数偶数个数，直到遇到回车为止。

（1）算法分析：可设一整型变量 x，循环的输入整数的值，若 $x\%2==0$ 即为偶数，否则是奇数。

提示：循环条件可在 while 语句后的括号中出现，也可以用 break 语句控制。

（2）根据分析结果写出源程序。

程序一：

```
#include <stdio.h>
void main( )
{
    int x,xo=0,xj=0;    /* 将偶数个数 xo 与奇数个数 xj 赋 0 值 */
    scanf("%d",&x);    /* 先输入一个整数 */
    while(x!='\n')    /* 输入不是回车时循环 */
    {
        if (x%2==0)
            xo=xo+1 ;
        else
            xj=xj+1 ;
        scanf("%d",&x);    /* 循环输入其余整数 */
    }

    printf("xo:%d,xj:%d",xo,xj);
}
```

程序二：

```
#include <stdio.h>
void main( )
{
    int x,xo=0,xj=0 ;

    while(1)
    {
        scanf("%d",&x);
        if(x=='\0')    break ; /* 在循环内部用 break 语句控制循环条件
        if(x%2==0)    xo=xo+1 ;
```

```

        else xj=xj+1 ;
    }
    printf("xo:%d,xj:%d",xo,xj);
}

```

(3) 在 VC++编辑环境下编辑源程序。

(4) 编译直到没有错误。

(5) 运行并输入：

13 16 7 23 68 99 86 \0✓

(6) 查看结果

Eg4-2: 打印如下图案

```

          *
        * * *
      * * * * *
    * * * * * *
  
```

(1) 算法分析：从图案中找出行数，空格数，星号数间的关系如下：

行数	空格数	星号数
1	3	1
2	2	3
3	1	5
4	0	7
i	4-i	2*i-1

可用双重循环控制整个图案的输出。若用循环变量 i, j 分别控制外层，内层循环，则 i 的取值从 1 到 4，表示行数，在每行中要确定前面空格个数和星号个数，空格个数是 4-i，星号个数是 2*i-1。

(2) 根据分析结果写出源程序

```

#include <stdio.h>
void main()
{
    int i,j;    /*定义循环控制变量 */
    for(i=1;i<=4;i++)
    {
        for(j=1;j<=4-i;j++)
            printf(" ");    /* 输出 4-i 个空格 */
        for(j=1;j<=2*i-1;j++)
            printf(" * ");
        printf("\n");
    }
}

```

(3) 在 VC++编辑环境下编辑源程序。

- (4) 编译直到没有错误。
- (5) 运行。因不需输入，故此程序运行时仅是屏幕闪一下就又回到编辑环境了。
- (6) 查看结果。输出题目要求的图案。

实验报告要求对照各程序运行结果分析以上各程序的算法（框图表示），解释产生该结果现象的原因。

四、 实验内容二（综合及设计类）

Eg4-3: 分别用 while、do-while、for 语句编程，求数列前 20 项之和：2/1, 3/2, 5/3, 8/5, 13/8 ...

Eg4-4: 一球从 100 米高度自由落下，每次落地后反跳回原高度的一半，再落下。求它在第 10 次落地时，共经过多少米？第 10 次反弹多高？

Eg4-5: 输入两个正整数 m 和 n，求它们的最大公约数和最小公倍数。

(1) 算法分析：

最大公约数和最小公倍数可以利用如下 3 条性质实现：

性质 1：如果 $a > b$ ，则 a 和 b 的最大公约数与 $a-b$ 和 b 的最大公约数相同；

性质 2：如果 $a < b$ ，则 a 和 b 的最大公约数与 a 和 $b-a$ 的最大公约数相同；

性质 3：如果 $a = b$ ，则 a 和 b 的最大公约数与 a 和 b 的值相同；

(2) 编程

(3) 运行

在运行时，输入的值 $m > n$ ，观察结果是否正确。

再输入时，使 $m < n$ ，观察结果是否正确。

修改程序，不论 m 和 n 为何值（包括负整数），都能得到正确结果。

Eg4-6: 【兔子问题】把费波拉契(Fibonacci)数列的前 50 个数输出。

实验报告要求根据题目要求设计解决问题的算法，画出其流程框图，写出满足 C 语言规范的源程序，在 VC++ 环境下调试运行，对照运行结果分析产生该结果现象的原因。

五. 扩展内容

Eg4-7: 素数问题：把 100~2000 之间的素数输出。

Eg4-8: 闰年问题：判断 1000—2000 年之间所有的闰年。

Eg4-9: 用牛顿迭代法求下面方程在 1.5 附近的根：

$$2x^3 - 4x^2 + 3x - 6 = 0$$

在能力所及的情况下完成扩展内容的算法分析，程序编写，并利用实验时间上机调试完成运行，并将其运行过程及结果在实验报告中体现。

【小结】 3 种循环的异同点？能否相互转换？for 循环控制单元的表达式与 while 循环中表达式是如何对应的？

实验五 数组

一、实验目的

- 1、熟练掌握一维数组的基本概念，如何定义一维数组，如何初始化一维数组，输入输出、引用数组元素等方法；
- 2、熟练掌握字符数组和字符串函数的使用；
- 3、掌握二维数组的相关操作：引用数组元素、行（列）求和，行（列）最大最小值，整个数组的输入输出等。；
- 4、掌握与数组有关的常用算法，如找最大最小值、排序、首尾颠倒等；
- 5、掌握与二维数组有关的算法，如找最大最小值（或位置）、转置数组等。

二、相关知识点

数组的定义；
数组元素的引用；
数组的赋值；
数组的首址；

三、 实验内容一（验证及程序分析）

Eg5-1: 用一维数组计算多个（小于等于 20 个数）单精度实数的平均值（一维数组的应用）

（1）算法分析：先定义一个大小为 20 的浮点数组,程序运行时先将数据加总变量初设为 0，并输入数据。经过 if 选择语句来判断输入的数是否超过 20，当输入小于 20 的时候，执行数据输入循环内的程序，否则执行输出“Please input a number less than 20”的消息。循环中的程序功能为读取用户输入的数据，并累加至 Summary 变量中。平均值的求法为数据总和除以数据个数。

（2）源程序：

```
#include <stdio.h>
void main()
{   int count=0;
    int i;
    float number[20];
    float summary;
    float average;
    summary=0;

    printf("Please input the number of data:");    //输入要计算的个数
    scanf("%d",&count );

    //判断合法个数
    if(count<20)
```

```

    {
        for(i=0;i<count;i++)
        { //输入 count 个实数
            printf("Please input a number:");
            scanf("%f",&number[i]);
            //累加求和
            summary+=number[i];
        }
        //计算平均值
        average=summary/count;
        // 输出平均值
        printf("The average is %5.2f\n",average);
    }
    else
        printf("Please input a number less than 20.");
}

```

(3) 输入数据:

(4) 输出结果:

(5) 同学们可以试着输入大于 20 的数字, 看看结果有什么变化?

Eg5-2: 分别求一个 5×5 矩阵的所有行之和、列之和、两条对角线上的元素之和。(二维数组的应用)

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

【提示】Eg5-8: 检验并打印魔方矩阵也建立在此题的基础上, 有兴趣的同学可以试一试。

(1) 算法分析:

在下面的 5×5 阶矩阵中, 先将这个矩阵中的元素读到一个二维整型数组中, 对每一行、每一列的元素之和分别定义一个一维整型数组, 对每一对角线上的元素之和只须定义 2 个变量。

(2) 源程序

```

#include <stdio.h>
#define N 5
void main( )
{
    int i, j;
    int x[N][N] = {{ 17,24,1,8,15},{ 23,5,7,14,16}, { 4,6,13,20,22},
                  { 10,12,19,21,3},{ 11,18,25,2,9}};
    int rowSum[N], colSum[N], diagSum1, diagSum2;
    int flag = 1;
    // 行求和
    for (i=0; i<N; i++)
    {
        rowSum[i] = 0;

```

```

        for (j=0; j<N; j++)
        {
            rowSum[i] = rowSum[i] + x[i][j];
        }
    }
//列求和
    for (j=0; j<N; j++)
    {
        colSum[j] = 0;
        for (i=0; i<N; i++)
        {
            colSum[j] = colSum[j] + x[i][j];
        }
    }
//主对角线求和
    diagSum1 = 0;
    for (j=0; j<N; j++)
    {
        diagSum1 = diagSum1 + x[j][j];
    }
//次对角线求和
    diagSum2 = 0;
    for (j=0; j<N; j++)
    {
        diagSum2 = diagSum2 + x[j][N-1-j];
    }
//输出行之和
    for (i=0; i<N; i++)
    {
        printf("%4d", rowSum[i]);
    }
    printf("\n");
//输出列之和
    for (i=0; i<N; i++)
    {
        printf("%4d", colSum[i]);
    }
    printf("\n");
//输出主对角线和次对角线之和
    printf("%4d %4d\n", diagSum1, diagSum2);
}

```

(3) 运行结果:

Eg5-3:输入一长度不超过 10 的字符串，判断它是否回文（字符串的应用）

如：ab22ba, level 均为回文。

(1) **算法分析：**回文其实就是字符串中的字符对称相等，所以就要判断对称的字符是否都相等，需要用循环来控制。可设一字符数组 a，长度为 10，用来存放此字符串，再设循环变量 i, j，兼做数组下标，进入循环时 i 的值为 0，指向第零个元素，j 的值为数组长度减 1，指向最后一个元素，判断 a[i] 和 a[j] 是否相等，如出现不等就退出循环，下次循环时 i 增 1，j 减 1，继续判断 a[i] 和 a[j] 的关系，若没出现过不等的情况就是

回文。

长度为偶数 ab22ba, 当 $i=3$, $j=2$ 时停止

长度为奇数 level, 当 $i=2$, $j=2$ 时停止, 故 $i < j$ 为循环条件。

(2) 源程序:

```
#include <string.h>    /* 程序中要用到字符串处理函数 */
#include <stdio.h>
void main( )
{
    char a[10];
    int i,j,f=1; /* f 用做标志, 若发现某一 a[i]与 a[j]不等, 就置 f 为 0, 退出
                循环, 若 f 为 1 就是回文, 否则不是*/
    scanf("%s",a); /*用格式符 s 整体输入字符串, 因 a 为数组名, 已代表数
                组起始地址, 所以输入时不必加取地址符 */
    for(i=0,j=strlen(a)-1;i<j;i++,j--)
    {
        if(a[i]!=a[j]) f=0;
        if(!f) break; /* 若 f 为 0 提前退出循环, 不是回文 */
    }
    if (f) printf("Yes\n");
    else printf("No\n");
}
```

(3) 在 VC++ 编辑环境下编辑源程序。

(4) 编译直到没有错误。

(5) 运行, 输入: wh33hw✓

(6) 查看输出结果:

四、 实验内容二（综合及设计类）

Eg5-4: 用选择法对 10 个整数排序。10 个整数用 scanf 函数输入。

(1) **算法分析:** 选择法的排序思想是从所有元素中选择一个最小元素放在 $a[0]$ (即让最小元素 $a[j]$ 与 $a[0]$ 交换, j 变量是用来记录最小数的下标值的), 作为第一轮; 第二轮是从 $a[1]$ 开始到最后的各元素中再选择一个最小元素, 放在 $a[1]$ 中...以下类推。

显然, n 个数要进行 $n-1$ 轮, 在第一轮中要比较 $n-1$ 次, 第二轮中比较 $n-2$ 次, ..., 第 i 轮中比较 $n-i$ 次。

(2) 编写程序:

(3) 输入 10 个数据:

(4) 运行结果:

Eg5-5: 打印杨辉三角形前 10 行

创建二维数组 $a[N][N]$, i, j


```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
.....

```

(1) 算法提示: 流程图----->

(2) 编写程序:

(3) 运行结果:

for i=0, i < N, ++i	
	a[i][0]=1
	a[i][i]=1
for i=2; i < N; ++i	
	for j=1; j < i; ++j
	a[i][j] = a[i-1][j-1] + a[i-1][j]
for i=0; i < N; ++i	
	for j=0; j <= i; ++j
	printf("%6d", a[i][j])
	printf("\n")

五. 扩展内容(选做)

Eg5-6: 编一个程序, 将两个字符串 s1 和 s2 进行比较。如果 s1>s2, 输出一个正数; s1=s2, 输出 0; s1<s2, 输出一个负数。不要用 strcmp 函数。两个字符串用 gets 函数读入, 输出的正数或负数的绝对值应是相比较的两个字符串相应字符的 ASCII 码的差值。

(1) **算法分析:** 两个字符串从第一个字符开始一一对应进行比较, 直到不相等的那个字符, 从而求得它们的 ASCII 码的差值。比如, ‘A’ 与 ‘C’ 相比, 由于 ‘A’ < ‘C’, 应输出负数, 由于 ‘A’ 与 ‘C’ 的 ASCII 码的差值为 2, 因此, 应输出 “-2”。同理, “And” 和 “Aid” 比较, 根据第 2 个字符比较结果, ‘n’ 比 ‘i’ 大 5, 因此应输出 ‘5’。

(2) 编写程序:

(3) 测试结果:

S1	S2	结果
abcd	abcd	
Abcd123	abdefds	
cdefds	A123evr	

Eg5-7: 检验并打印魔方矩阵 (二维数组的应用)

(1) 算法分析:

在下面的 5×5 阶魔方矩阵中, 每一行、每一列、每一对角线上的元素之和都是相等的, 试编写程序将这些魔方矩阵中的元素读到一个二维整型数组中, 然后检验其是否为魔方矩阵, 并将其按如下格式显示到屏幕上。

```

17  24  1   8   15
23  5   7  14  16
4   6  13  20  22
10  12  19  21  3
11  18  25  2   9

```

(2) 源程序

```
#include <stdio.h>
#define N 5
void main( )
{
    int i, j;
    int x[N][N] = {{ 17,24,1,8,15},{ 23,5,7,14,16}, { 4,6,13,20,22},
                    { 10,12,19,21,3},{ 11,18,25,2,9}};
    int rowSum[N], colSum[N], diagSum1, diagSum2;
    int flag = 1;
    // 行求和
    for (i=0; i<N; i++)
    {
        rowSum[i] = 0;
        for (j=0; j<N; j++)
        {
            rowSum[i] = rowSum[i] + x[i][j];
        }
    }
    //列求和
    for (j=0; j<N; j++)
    {
        colSum[j] = 0;
        for (i=0; i<N; i++)
        {
            colSum[j] = colSum[j] + x[i][j];
        }
    }
    //主对角线求和
    diagSum1 = 0;
    for (j=0; j<N; j++)
    {
        diagSum1 = diagSum1 + x[j][j];
    }
    //次对角线求和
    diagSum2 = 0;
    for (j=0; j<N; j++)
    {
        diagSum2 = diagSum2 + x[j][N-1-j];
    }
    // 判断主对角和次对角元素之和是否相等
    if (diagSum1 != diagSum2)
    {
        flag = 0;
    }
    else
    {
        for (i=0; i<N; i++)
        {
            //再判断行和列元素之和是否相等
            if ((rowSum[i] != diagSum1) || (colSum[i] != diagSum1))
                flag = 0;
        }
    }
}
```

```

// 判是否为魔方矩阵，进行输出
if (flag)
{
    printf("It is a magic square!\n");
    for (i=0; i<N; i++)
    {
        for (j=0; j<N; j++)
        {
            printf("%4d", x[i][j]);
        }
        printf("\n");
    }
}

```

(3) 运行结果:

【小结】 如何定义一维数组？如何初始化一维数组？`int a[10]`；定义了几个数组元素？其数组元素分别是什么？存储空间如何分配？如何随机访问任意一个数组元素？

实验六 函数一

一、 实验目的

- 1、掌握函数的定义方法；
- 2、掌握函数的调用及函数声明；
- 3、掌握函数实参与形参的对应关系及“值传递”的方式；
- 4、掌握函数实参与形参的对应关系及“地址传递”的方式；

二、 相关知识点（比较详细）

函数定义的一般形式；

函数的参数（形参、实参）；

函数的返回值；

函数的调用（值传递，地址传递）；

三、 实验内容一（验证及程序分析）

Eg6-2: 判别一个整数数组中各元素的值，若大于 0 则输出该值，若小于等于 0 则输出 0 值。

（知识点：数组元素作参数，值传递）

（1） 算法分析：本程序中首先定义一个无返回值函数 cmp，并说明其形参 v 为整型变量。在函数体中根据 v 值输出相应的结果。在 main 函数中用一个 for 语句输入数组各元素，每输入一个就以该元素作实参调用一次 cmp 函数，即把 a[i] 的值传送给形参 v，供 cmp 函数使用。

（2） 程序如下：

```
#include <stdio.h>
void cmp(int v)
{
    if(v>0)
        printf("%d ",v);
    else
        printf("%d ",0);
}
void main()
{
    int a[5],i;
    printf("input 5 numbers\n");
    for(i=0;i<5;i++)
    {
        scanf("%d",&a[i]);/*每输入一个数，就对它进行与零的比较*/
        cmp(a[i]);
    }
}
```

```

    }
}

```

(3) 在 VC++ 编辑环境下输入源程序。

(4) 编译直到程序无错误。

(5) 运行，输入：

input 5 numbers

(6) 查看运行结果。

Eg6-3: 输入 10 个学生的姓名和学号，要求按学号从小到大顺序排序，姓名顺序也随之调整。

(知识点：数组名作为函数参数，地址传递)

(1) 算法分析：

input 函数完成 10 个学生的数据的录入，sort 函数的作用是选择法排序。

(2) 程序如下：

```

#include <stdio.h>
#include <string.h>
#define N 10
void input(int num[],char name[N][8]) /*定义函数 input, 完成输入数据的功能*/
{
    int i ;
    for(i=0;i<N;i++)
    {
        printf("\n input NO.:");
        scanf("%d",&num[i]);
        printf("input name:");
        getchar();
        gets(name[i]);
    }
}

void sort(int num[],char name[N][8]) /*定义函数 sort, 完成排序的功能*/
{
    int i,j,min,temp1 ;
    char temp2[8];

    for(i=0;i<N-1;i++) /*外循环控制轮次*/
    {
        min=i ;
        for(j=i;j<N;j++)
            if(num[min]>num[j]) /*内循环控制本轮中共执行的次数*/
                min=j ; /*每次进行比较, 变量 j 用来记录元素下标的最小值*/
        temp1=num[i]; /*假设如果为真, 则以下语句完成交换*/
        strcpy(temp2,name[i]);
        num[i]=num[min];
    }
}

```

```

        strcpy(name[i],name[min]);
        num[min]=temp1 ;
        strcpy(name[min],temp2);
    }
    printf("\n result:\n");
    for(i=0;i<N;i++)
        printf("\n %5d% 10s",num[i],name[i]); /*按排好的顺序输出 10 个数据*/
}

void main( )
{
    int num[N],number ;
    char name[N][8];
    input(num,name); /*调用输入函数 input*/
    sort(num,name); /*调用排序函数 sort*/
}

```

(3) 在 VC++ 编译环境下输入源程序代码。

(4) 编译直到程序没有错误。

(5) 运行，输入：

```

input NO. :1✓
input name: Li✓
input NO. : 2✓
input name: Wang✓
input NO. : 5✓
input name: Liu✓
input NO. : 8✓
input name: Ma✓
input NO. : 4✓
input name: Chen✓
input NO. : 10✓
input name: Zhou✓
input NO. : 12✓
input name: Zhang✓
input NO. : 6✓
input name: Xie✓
input NO. : 23✓
input name: Yuan✓
input NO. : 34✓
input name: Lu✓

```

(6) 查看运行结果

理解验证程序的工作流程，分析其算法，对照结果解释产生该结果现象的原因。

四、 实验内容二（综合及设计类）

Eg6-3: 输入 10 个学生 5 门课的成绩，分别用函数求：

- (1) 每个学生平均分；
- (2) 每门课的平均分；
- (3) 找出最高的分数所对应的学生和成绩。

根据题目要求设计解决问题的算法，画出其流程框图，写出满足 C 语言规范的源程序，在 VC++ 环境下调试运行，对照运行结果分析其结果。

五、扩展内容

Eg6-4: 求两个整数的最大公约数和最小公倍数。设计一个求最大公约数的函数和一个最小公倍数的函数，实现 **Eg4-4** 的任务；

Eg6-5: 设计一个判素数的函数，实现 **Eg4-5** 的任务；

Eg6-6: 设计一个判闰年的函数，实现 **Eg4-6** 的任务；

【小结】 调用库函数时应在程序开头添加什么命令？什么是 void 函数？void 函数和有返回值函数的调用形式有什么不同？什么是值参数？值参数对应的实参有什么规定？

实验七 函数二

一、 实验目的

- 1、掌握函数的嵌套调用和递归调用方法；
- 2、掌握全局变量和局部变量、动态变量和静态变量的概念和使用方法；
- 3、学习对多文件程序的编译和运行。

二、 相关知识点

递归函数的嵌套调用规则；

局部、全局变量的作用域和使用时机；

四种存储属性 AUTO、REGISTER、EXTERN 和 STATIC 变量的生存期和使用时机；

三、 实验内容一（验证及程序分析）

Eg7-1: Hanoi（汉诺）塔问题（函数递归调用）

这是一个典型的只有用递归方法（而不可能用其他方法）解决的问题。问题是这样的：有三根针 A、B、C。A 针上有 64 个盘子，盘子大小不等，大的在下，小的在上。要求把这 64 个盘子从 A 针移到 C 针，在移动过程中可以借助 B 针，每次只允许移动一个盘，且在移动过程中在三根针上都保持大盘在下，小盘在上。要求编程序打印出移动的步骤。

（1）算法思想：将 n 个盘子从 A 针移到 C 针可以分解为以下三个步骤：

- ①将 A 上 n-1 个盘借助 C 针先移到 B 针上。
- ②把 A 针上剩下的一个盘移到 C 针上。
- ③将 n-1 个盘从 B 针借助于 A 针移到 C 针上。

（2）源程序：

```
#include <stdio.h>

void move(char getone, char putone)
{
    printf("%c- ->%c\n",getone,putone);
}

void hanoi(int n, char one, char two, char three)
{
    /*将 n 个盘从 one 借助 two， 移到 three*/
    if(n==1)    move(one,three);
    else
    {
        hanoi(n-1,one,three,two);
        move(one,three);
        hanoi(n-1,two,one,three);
    }
}
```



```

    }
    void main( )
    {
        int m ;
        printf("input the number of disks:");
        scanf("%d",&m);
        printf("The step to moving%3 d disks : \n ", m);
        hanoi(m,'A','B','C');
    }

```

(3) 运行结果:

Eg7-2: 利用函数的递归调用实现某数的阶乘 (函数递归调用)

(1) 源程序:

```

#include <stdio.h>
int fac(int n)
{    //求阶乘函数
    int f;
    if(n<0)  printf("n<0,data error!");
    else if(n==0||n==1)
        f=1;
    else
        f=fac(n-1)*n;
    return(f);
}
void main()
{
    int n, y;
    printf("Input a integer number:");
    scanf("%d",&n);
    y=fac(n);
    printf("%d!=%15d",n,y);
}

```

(2) 运行结果:

Eg7-3: 分析下面程序的运行结果，为什么出现这种结果？

(1) 源程序

```

#include <stdio.h>
void fun(int i, int j);
void main( )
{
    int i, j, x, y, n, g;
    i=2, j=3, g=x=5, y=9, n=7;
    fun(n, 6);
    printf("g=%d;i=%d;j=%d;\n", g, i, j);
}

```

```

        printf("x=%d;y=%d\n", x, y);
        fun(n, 6);
    }
void fun(int i, int j)
{
    int x, y, g;
    g=8; x=7; y=2;
    printf("g=%d;i=%d;j=%d;\n", g, i, j);
    printf("x=%d;y=%d\n", x, y);
    x=8, y=6;
}

```

(2) 运行结果:

(3) 结果分析:

Eg7-4: 分析下面程序的运行结果，为什么出现这种结果？

(1) 源程序:

```

#include <stdio.h>
void incx(void);
void incy(void);
void main( )
{
    incx( );
    incy( );
    incx( );
    incy( );
    incx( );
    incy( );
}

void incx(void)
{
    int x=0;
    printf("x=%d\t", ++x);
}

void incy(void)
{
    static int y=0;
    printf("\ny=%d\n", ++y);
}

```

(2) 运行结果:

(3) 结果分析:

四、 实验内容二（综合及设计类）

Eg7-5: 用递归算法处理 Fibonacci 数列问题，计算 Fibonacci 数列的第 7 项 $f(7)$ 。

递归公式：

$$f(n) = \begin{cases} 1 & (n=1) \\ 1 & (n=2) \\ f(n-1)+f(n-2) & (n>2) \end{cases}$$

注意：1、函数 $f(n)$ 是一个有返回值的函数。

2、在主函数中提供参数 n 的值。

五. 扩展内容

Eg7-6: 用递归法将一个整数 n 转换成字符串。例如输入整数“789”，应输出字符序列‘7’‘8’‘9’。 n 的位数不固定，可以是任意位数的整数。

（1）算法提示：将一个整数 n 分解成字符序列，首先将问题划分成两个部分，一部分是终止递归的条件，另一部分是继续更深层次递归的条件：

- 当 $n/10 \neq 0$ 时，将 $n/10$ 作为参数继续更深层次递归。
- 否则如果当 $n/10 = 0$ 时（1 位数，可以直接转换），把整数 n 转换成单个字符并输出， $c=n\%10+'0'$ ，打印 c

注意：1、函数 `convert(int n)` 完成输出功能，是一个 `void` 函数。

2、在主函数中提供参数 n 的值。

（2）源程序：

（3）输入 $n=789$ ，输出：

（4）输入 $n=12345$ ，输出：

Eg7-7: 用递归法求 n 阶勒让德多项式的值，递归公式为：

$$P_n(x) = \begin{cases} 1 & (n=0) \\ x & (n=1) \\ ((2n-1) \cdot x \cdot P_{n-1}(x) - P_{n-2}(x)) / n & (n>1) \end{cases}$$

Eg7-8: 求两个整数的最大公约数和最小公倍数（**Eg4-4** 和 **Eg6-4** 的扩展）。

（1）不用全局变量，分别用两个函数求最大公约数和最小公倍数。两个整数在主函数中输入，并传送给函数 1，求出的最大公约数返回主函数，然后再与两个整数一起作为实参传递给函数 2，以求出最小公倍数，再返回到主函数输出最大公约数和最小公倍数。

（2）用全局变量的方法，分别用两个函数求最大公约数和最小公倍数，但其值不由

函数带回。将最大公约数和最小公倍数都设为全局变量，在主函数中输出它们的值。

(3) 将两个函数单独存放在 `file1.c` 中，主函数存放在 `file2.c` 中，用课本中介绍的方法对它们进行编译和运行。

【小结：】递归算法是利用函数处理问题的技术。递归函数每一次调用都保存了形参和变量的值；通过结束递推调用返回上一次调用，实现回代过程，从而解决原始问题。

实验八 指针

一、实验目的

- 1、掌握宏定义、文件包含的方法，了解条件编译的作用与使用方法；
- 2、掌握指针的概念，会定义和使用指针变量；
- 3、了解或掌握指针与数组的关系，指针与数组有关的算术运算、比较运算。
- 4、能正确使用数组的指针和指向数组的指针变量；
- 5、能正确使用字符串的指针和指向字符串的指针变量。

二、相关知识点

宏定义，带参数的宏定义，不带参数的宏定义；
地址和指针的概念；
数组和指针的关系；
字符串和指针的关系；

三、实验内容一（验证及程序分析）

Eg8-1: 输入两个整数，并使其从大到小输出，用指针变量实现数的比较。

(1) 源程序：

```
#include <stdio.h>
void main()
{   int *p1,*p2,*p,a,b;
    scanf("%d,%d",&a,&b);
    p1=&a; p2=&b;
    if(a<b)
    { p=p1; p1=p2; p2=p;}
    printf("a=%d,b=%d\n",a,b);
    printf("max=%d,min=%d\n",*p1,*p2);
}
```

(2) 输入数据：

(3) 结果输出：

Eg8-2: 输入两个整数，并使其从大到小输出，用函数实现数的交换

(1) 源程序：

```
#include <stdio.h>
void swap(int *p1, int *p2)
{   int p;
    p=*p1;
```

```

        *p1=*p2;
        *p2=p;
    }
void main()
{   int a,b;
    int *p,*q;
    scanf("%d,%d",&a,&b);
    p=&a;  q=&b;
    if(a<b) swap(p,q);
    printf("\n%d,%d\n",a,b);
}
(2) 输入数据:
(3) 结果输出:
(4) 如果将 swap 函数修改为如下形式, 分析如何调试和修改?
void swap(int *p1, int *p2)
{   int *p;
    *p=*p1;
    *p1=*p2;
    *p2=*p;
}

```

Eg8-3:用指针法输入 12 个数, 然后按每行 4 个数输出。

(1) 算法分析: 定义一个整型数组和一个整型指针, 这样通过数组就可以静态分配内存空间, 存储数据; 然后将指针与数组相关, 使指针指向与数组相同的首地址处, 这样就可以通过指针或者数组都可以对存储空间加以操作。

(2) 源程序:

```

#include <stdio.h>
void main()
{   int j,k,a[12],*p;
    p=a;                                //使指针 p 指向与数组 a 相同的首地址处
    for(j=0;j<12;j++)
        scanf("%d",p++);               //移动 P 的位置, 输入数据
    p=a;                                //指针重定位
    for(j=0;j<12;j++)
    {
        if(j%4==0)
            printf("\n");               //按每行 4 个数输出
        printf("%4d",*p++);
    }
    printf("\n");
}

```

(3) 输入数据:

(4) 结果输出:

Eg8-4: 分别用下标法、地址法和指针变量 p 访问数组 $a[5]=\{1,3,5,7,9\}$, 用这三种方法输出数组各元素的值, 每种方法输出的五个元素在一行上。

(1) 源程序:

```
#define S(a,b)  a*b
#include <stdio.h>
void main( )
{
    int c,d,t ;
    scanf("%d, %d",&c,&d);
    t=S(c+d,c-d);    /*宏展开后代入变量 c,d 的值, 从而求得变量 t 的值*/
    printf("%d",t);
}
```

(2) 输入数据:

(3) 结果输出:

(4) 分析本题难点:

①宏展开后的表达式应该为: $c+d*c-d$

②代入 c, d 的值 (例如输入 2, 3), 则 $t=2+3*2-3=5$

③易错的地方是结果为: $(2+3) * (2-3) = -6$ (无形中加了括号)

假设此题变化为: #indefi ne S (a, b) (a)*(b), 则正确结果才是-6

通过此题应该掌握带参的宏展开的实质: 不仅仅是作简单的置换, 还要进行参数的替换。

四、 实验内容二 (综合及设计类)

Eg8-5: 从键盘输入十个整数, 要求用冒泡法 (或选择法) 实现从大到小的排列输出。

EG8-6: 练习指针数组: 有三个字符串 “DATA STRUCTURE”、“COPUTER DESIGN”、“C PROGRAM”, 请按字符顺序输出这三个字符串。(要求用指针数组指向这三个字符串。)

五. 扩展内容

Eg8-7: 将指针作为函数参数: 一个数组有 10 个元素 {1, 8, 10, 2, -5, 0, 7, 15, 4, -5}, 利用指针作为函数参数编程, 输出数组中最大和最小的元素值。

Eg8-8:练习数组指针作为函数参数: 求 3×4 的二维数组 {1, 3, 5, 7, 9, 11, 13, 17, 19, 21, 23, 25} 中的所有元素之和。

Eg8-9:练习指针与字符串: 在一行字符串中删去指定的字符。例如, 要求在一行文字:

“I have 150 Yuan!!” 中删去字符 “0”，使其变为 “I have 15 Yuan!!”。

实验九 结构体、共用体和位运算

一、 实验目的

- 1、掌握结构体类型变量的定义和使用；
- 2、掌握结构体类型数组的概念和使用；
- 3、掌握链表的概念，初步学会对链表进行操作；
- 4、掌握共用体的概念与使用；
- 5、掌握按位运算的概念和方法，学会使用位运算符；
- 6、学会通过位运算实现对某些位的操作。

二、 相关知识点

C 语言提供了一种如果用简单变量来分别代表属性，难以反映出他们之间的内在联系数据类型称为结构体。如学生姓名、编号、性别、年龄、各科成绩、地址等。他们是同一个处理对象，学生的属性，在这之间，即有字符型、也有长整、短整型、实型等各种数据类型。

例：

Num	name	sex	age	score	addr
10010	Li fum	m	18	88.5	beijin
整型	字符型	字符	整型	实型	字符型

```
struct student
{
    int num;
    char name[20];
    char sex;
    short int age;
    float score;
    char addr[30];
}
```

上面就定义了一个结构体类型，**struct** 是关键字，结构体类型是 **student** 。其中有 6 个不同的数据项。

结构体类型不同于基本数据类型的特点：

- (1) 由若干个数据项组成，每个数据项称为一个结构体的成员，也可称为“域”。
- (2) 结构体类型并非只能有一种，而可以有千千万万。

```
struct 结构体名
{
    成员项表列
};
```

- (3) 定义一个结构体类型，并不意味着系统将分配一段内存单元来存放各数据项成员。

因为这仅仅只定义了类型。

(4) 结构体类型需用户自己定义。

位运算符和位运算

三、 实验内容一（验证及程序分析）

Eg9-1: 编写一个函数 `print`，打印一个学生的成绩数组，该数组中有 5 个学生的数据记录，每个记录包括 `num`、`name`、`score[3]`，用主函数输入这些记录，用 `print` 函数输出这些记录。

(1) 算法分析：依据题意，先定义一个包含有三个成员项的结构体数组，在主函数中利用循环依次输入数据，并调用函数 `print`，完成输出数据的功能。

(2) 程序如下：

```
#define N 5
#include <stdio.h>
struct student
{
    char num[6];
    char name[8];
    int score[4]; /*注意，考虑到下标从 1 开始，有三门课程成绩 score[3]所以设定
分数的数组长度加 1, score[4]，防止数据溢出*/
} stu[N];
void main()
{
    int i,j;
    void print(struct student stu[N]);
    for(i=0;i<N;i++)
    {
        printf("\nInput score of student %d:\n",i+1);
        printf("NO.:");
        scanf("%s",stu[i].num);
        printf("name:");
        scanf("%s",stu[i].name);
        for(j=1;j<=3;j++)
        {
            printf("score %d:",j);
            scanf("%d",&stu[i].score[j]);
        }
        printf("\n");
    }
    print(stu);
}
```

```

void print(struct student stu[N])
{ // 输出模块
    int i,j;
    printf("\n NO.    name    score1    score2    score3\n");
    for(i=0;i<N;i++)
    {
        printf("%5s%10s",stu[i].num,stu[i].name);
        for(j=1;j<=3;j++)
            printf("%9d",stu[i].score[j]);
        printf("\n");
    }
}

```

(3) 在 VC++编辑环境下输入源程序。

(4) 编译直到程序无错误。

(5) 运行输入:

Input score of student 1:

NO. :101✓

name:Li✓

score1:90✓

score2:79✓

score3:89✓

Input score of student 2:

NO. :102✓

name:Ma✓

score1:97✓

score2:90✓

score3:58✓

Input score of student 3:

NO. :103✓

name:Wang✓

score1:77✓

score2:70✓

score3:78✓

Input score of student 4:

NO. :104✓

name:Fun✓

score1:56✓

score2:89✓

score3:65✓

Input score of student 5:

NO. :105✓

name:Xue✓

score1:87✓

```
score2:65✓
```

```
score3:63✓
```

(6) 查看运行结果

Eg9-2: 设计一个函数，使给出一个数的原码，能得到该数的补码。

(1) 算法分析：一个正数的补码等于该数原码，一个负数的补码等于该数的反码加 1。

(2) 编写源程序

```
#include <stdio.h>
void main( )
{
    unsigned int a ;
    unsigned int getbits(unsigned);
    printf("\nInput an octal number:");
    scanf("%o",&a);
    printf("result:%o\n",getbits(a));
}

unsigned int getbits(unsigned value) /*求一个二进制的补码函数*/
{
    unsigned int z ;
    z=value&0100000 ;
    if(z==0100000)
        z=~ value+1 ;
    else
        z=value ;
    return(z);
}
```

(3) 运行情况如下：

①Input an octal number:2345✓

result:2345

②Input an octal number:152525✓

result:25253

实验报告要求对照各程序运行结果分析以上各程序的算法（框图表示），并对其结果进行分析。

四、 实验内容二（综合及设计类）

根据题目要求设计算法，编程并上机调试运行。

Eg9-3: 有 10 个学生，每个学生的数据包括学号、姓名、3 门课的成绩，从键盘输入 10 个学生的数据，要求打印出 3 门课的总平均成绩。

Eg9-4: 编写一个函数 getbits，从一个 16 位的单元中取出某几位（即该几位保留原值，其余位为 0）。函数调用形式为：

```
getbits(value, n1, n2)
```

value 为该 16 位（两个字节）单元中的数据值，n1 为欲取出的起始位，n2 为欲取出的结束位。如：

```
getbits(0101675,5,8)
```

表示对八进制 101675 这个数，取出它从左面起的第 5 位到第 8 位。

实验报告要求根据题目要求设计解决问题的算法，画出其流程框图，写出满足 C 语言规范的源程序，在 VC++ 环境下调试运行，分析运行结果。

五、扩展内容

理解扩展内容所涉及的相关知识点，对照下面的程序代码，理解其编程思路，并能举一反三。

Eg9-5: 通讯录管理系统:

该系统通过文本菜单进行操作，功能包括：创建通讯录、显示记录、查询记录、修改记录、添加记录、删除记录和记录排序等，各功能模块均采用独立的函数来表示，通过主函数直接或是间接调用，特别注意的是，通讯录数据采用结构体定义和管理，体会这样做的优越性。

(1) 程序如下：

```
#define N 100
#include <stdio.h>
#include <string.h>
#include <conio.h>
struct student
{
    char num[10];
    char name[10];
    char tel[10];
};

/* 函数声明 */
void myprint();/* 显示菜单 */
int mycreat(struct student*p,int n);/* 输入通讯录*/
void mydisplay(struct student*p,int n);/* 显示通讯录 */
void mysearch(struct student*p,int n);/* 查找 */
void mymodify(struct student*p,int n);/* 修改通讯录 */
int myadd(struct student*p,int n);/* 增加通讯录 */
int mydelete(struct student*p,int n);/* 删除 */
void mysort(struct student*p,int n);/* 排序 */
void sch_num(struct student*p,int n);/* 按学号查找 */
void sch_name(struct student*p,int n);/* 按姓名查找 */
void main()
```

```

{
    char choose,yes_no ;
    struct student record[N]; /* 存放通讯录信息,共 100 条 */
    int total ; /* 通讯录总数 */
    do
    {
        myprint();          /* 显示菜单 */
        printf("          请选择:          ");
        choose=getchar();
        switch(choose)
        {
            case '1' :
                total=mycreat(record,N);
                break ;
            case '2' :
                mydisplay(record,total);
                break ;
            case '3' :
                mysearch(record,total);
                break ;
            case '4' :
                mymodify(record,total);
                break ;
            case '5' :
                total=myadd(record,total);
                break ;
            case '6' :
                total=mydelete(record,total);
                break ;
            case '7' :
                mysort(record,total);
                break ;
            case '0' :
                break ;
            default :
                printf("\n          %c 为非法选项!\n",choose);
        }
        if(choose=='0') break ;
        printf("\n          要继续选择吗(Y/N)?");
        do
        {
            yes_no=getchar();
        }while(yes_no!='Y'&&yes_no!='y'&&yes_no!='N'&&yes_no!='n');
    }while(yes_no!='Y'||yes_no!='y');
}

```

```

}

void myprint()
{ // 显示菜单信息
    printf("\n\n\n\n\n");
    printf("                                \n");
    printf("        请输入选项编号(0--7)          \n");
    printf("            1---创建通讯录              \n");
    printf("            2---显示通讯录              \n");
    printf("            3---查询通讯录              \n");
    printf("            4---修改通讯录              \n");
    printf("            5---添加通讯录              \n");
    printf("            6---删除通讯录              \n");
    printf("            7---排序通讯录              \n");
    printf("            0---退出                    \n");
    printf("                                \n");
}

int mycreat(struct student*p,int n)
{ // 输入学生信息
    int i=0 ;
    int m ;
    printf("请输入要输入的学生人数(<%d):  ",n);
    scanf("%d",&m);
    while(i<m)
    {
        printf("\n");
        printf("请输入第%d 个记录:\n",i+1);
        printf("学号:");
        do
        {
            gets(p->num);
        }
        while(strcmp(p->num,"")==0);
        printf("姓名:");
        gets(p->name);
        printf("电话号码:");
        gets(p->tel);
        p++;
        i++;
    }
    return m ;
}

```

```

void mydisplay(struct student*p,int n)
{ // 显示学生信息
    printf("          学号          姓名          电话号码\n");
    while(n>0)
    {
        printf("%23s%15s%15s\n",p->num,p->name,p->tel);
        p++;
        n--;
    }
}

```

```

void mymodify(struct student*p,int n)
{ // 修改学生信息
    int i ;
    char nam[10];
    mydisplay(p,n);
    printf("\n 请输入要修改记录的姓名:");
    gets(nam);
    for(i=0;i<n;i++,p++)
    if(strcmp(nam,p->name)==0)break ;
    if(i==n)
    printf("无此人!\n");
    else
    {
        printf("请输入正确的学号:");
        do
        {
            gets(p->num);
        }
        while(strcmp(p->num,"")==0);
        printf("请输入正确的姓名:");
        gets(p->name);
        printf("请输入正确的电话号码:");
        gets(p->tel);
    }
}

```

```

void mysearch(struct student*p,int n)
{ // 查找指定学生
    char c ;
    mydisplay(p,n);
    printf("\n 按学号查询(h),还是按姓名查询(m)?");

```



```

        c=getchar();
        if(c=='h'||c=='H')
            sch_num(p,n);
        else if(c=='m'||c=='M')
            sch_name(p,n);
        else
            printf("\n 非法字符!\n");
    }

void sch_num(struct student*p, int n)
{
    // 按学号查询
    int i,flag=0 ;
    char nm[10];
    printf("\n 请输入要查询记录的学号:");
    gets(nm);
    for(i=0;i<n;i++)
        if(strcmp(nm,p[i].num)==0)
        {
            printf("          学号      姓名      电话号码\n");
            printf("%23s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
            flag=1 ;
        }
    if(flag==0)
        printf("无此人!\n");
}

void sch_name(struct student*p,int n)
{
    // 按姓名查询
    int i,flag=0 ;
    char nam[10];
    printf("\n 请输入要查询记录的姓名:");
    gets(nam);
    for(i=0;i<n;i++)
        if(strcmp(nam,p[i].name)==0)
        {
            printf("          学号      姓名      电话号码\n");
            printf("%23s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
            flag=1 ;
        }
    if(flag==0)
        printf("无此人!\n");
}

```

```

int myadd(struct student*p,int n)
{ // 增加学生信息
    int i=0 ;
    char c ;
    struct student t,*q ;
    printf("添加前的记录: \n");
    printf("          学号          姓名          电话号码\n");
    i=0 ;
    /* 显示原有记录信息 */
    while(i<n)
    {
        printf("%23s%15s%15s\n",q->num,q->name,q->tel);
        i++;
        q++;
    }

    do
    {
        printf("请输入新记录的学号: \n");
        do
        {
            gets(t.num);
        } while(strcmp(t.num,"")==0);
        /* 学号不能为空 */

        printf("请输入新记录的姓名: ");
        gets(t.name);
        printf("请输入新记录的电话号码: ");
        gets(t.tel);

        for(i=0;i<n;i++)
        if(strcmp(t.num,(p+i)->num)==0||strcmp(t.tel,(p+i)->tel)==0)
        {
            printf("学号或电话号码重,要重新输入吗(Y/N)?");
            c=getchar();
            putchar('\n');
            break ;
        }

        /* 找到最后, 没有重复 */
        if(i==n)
        {

```

```

        *(p+i)=t ;
        printf("添加后的记录:\n");
        printf("          学号          姓名          电话号码\n");
        for(i=0;i<n+1;i++)
            printf("%23s%15s%15s\n",(p+i)->num,(p+i)->name,p[i].tel);
        break ;
    }
} while(c=='y'||c=='Y');

return n+1 ;
}

int mydelete(struct student*p,int n)
{ // 删除学生信息
    int i,x ;
    char c,nm[10],nam[10];

    printf("删除前的记录: \n");
    printf("          学号          姓名          电话号码\n");
    for(i=0;i<n;i++)
        printf("%23s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
    printf("按学号删除(h)， 还是按姓名删除(m)?");
    c=getchar();
    if(c=='h'||c=='H')
    {
        printf("\n 请输入要删除记录的学号: ");
        gets(nm);
        for(x=0;x<n;x++)
            if(strcmp(nm,p[x].name)==0)
                break ;
        /* 说明提前结束循环，找到人 */
        if(x<n)
        {
            for(i=x;i<n-1;i++)
            {
                p[i]=p[i+1];
            }
            printf("删除后的记录: \n");
            printf("          学号          姓名          电话号码\n");
            for(i=0;i<n-1;i++)
                printf("%23s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
        }
        else

```

```

        printf("无此人\n");
    }
    else
    if(c=='m' || c=='M')
    {
        printf("\n 请输入要删除记录的姓名: ");
        gets(nam);
        for(x=0;x<n;x++)
            if(strcmp(nam,p[x].name)==0)
                break ;
        if(x<n)
        {
            for(i=x;i<N-1;i++)
            {
                p[i]=p[i+1];

            }
            printf("删除后的记录: \n");
            printf("          学号          姓名          电话号码\n");
            for(i=0;i<n-1;i++)
                printf("%23s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
        }
        else
            printf("无此人\n");
    }

    else
        printf("\n 非法字符\n");

    return n-1 ;
}

void mysort(struct student*p,int n)
{ // 排序函数
    int j,i,k ;
    char c1,c2 ;
    struct student temp ;
    printf("排序前的记录:\n");
    printf("          学号          姓名          电话号码\n");
    for(i=0;i<n;i++)
        printf("%23s%15s%15s\n",p[i].num,p[i].name,p[i].tel);

    printf("按学号排序(h), 还是按姓名排序(m)?");

```

```

c1=getchar();
if(c1=='h'||c1=='H')
{
    printf("\n 按升序排序 (s), 还是按降序排序 (j) ?");
    c2=getchar();
    /* 升序 */
    if(c2=='s'||c2=='S')
    {
        for(i=0;i<n-1;i++)
        {
            k=i ;
            for(j=i;j<n;j++)
            if(strcmp(p[k].name,p[j].name)<0)
            k=j ;
            temp=p[k];
            p[k]=p[i];
            p[i]=temp ;

        }
        printf("排序后的记录:\n");
        printf("          学号          姓名          电话号码\n");
        for(i=0;i<n;i++)
            printf("%23s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
    }
    /* 降序 */
    else if(c2=='j'||c2=='J')
    {
        for(i=0;i<n-1;i++)
        {
            k=i ;
            for(j=0;j<n;j++)
            if(strcmp(p[k].name,p[j].name)<0)
            k=j ;
            temp=p[k];
            p[k]=p[i];
            p[i]=temp ;

        }
        printf("排序后的记录:\n");
        printf("          学号          姓名          电话号码\n");
        for(i=0;i<n;i++)
            printf("%23s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
    }
}

```

```

        else printf("\n 非法字符\n");
    }

else if(c1=='m' || c1=='M')    /* 按姓名排序 */
{
    printf("\n 按升序排序 (s), 还是按降序排序 (j) ? ");
    c2=getchar();

    if(c2=='s' || c2=='S')
    {
        for(i=0;i<n-1;i++)
        {
            k=i ;
            for(j=0;j<n;j++)
                if(strcmp(p[k].name,p[j].name)<0)
                    k=j ;
            temp=p[k];
            p[k]=p[i];
            p[i]=temp ;
        }
        printf("排序后的记录:\n");
        printf("      学号      姓名      电话号码\n");
        for(i=0;i<n;i++)
            printf("%23s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
    }
    /* 降序 */
    else if(c2=='j' || c2=='J')
    {
        for(i=0;i<n-1;i++)
        {
            k=i ;
            for(j=i;j<n;j++)
                if(strcmp(p[k].name,p[j].name)<0)
                    k=j ;
            temp=p[k];
            p[k]=p[i];
            p[i]=temp ;
        }
        printf("排序后的记录:\n");
        printf("      学号      姓名      电话号码\n");
        for(i=0;i<n;i++)
            printf("%23s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
    }
}

```

```
    }  
    else printf("\n 非法字符! \n");  
  }  
  else printf("\n 非法字符! \n");  
}
```

(2) 编译直到程序无错误;
(3) 运行;

实验十 文件

一、 实验目的

- 1、掌握文件以及缓冲文件系统、文件指针的概念；
- 2、学会使用文件打开、关闭、读、写等文件操作函数；
- 3、学会用缓冲文件系统对文件进行简单的操作。

二、 相关知识点（比较详细）

文件类型指针

文件的打开与关闭

文件的读写操作函数

三、 实验内容一（验证及程序分析）

Eg10-1: 将一个磁盘文件中的信息复制到另一个磁盘文件中。

```
#include <stdio.h>
#include "stdlib.h"
void main( )
{
    FILE*in,*out ;    // 定义文件指针
    char ch,infile[10],outfile[10];
    printf("Enter the infile name:\n");
    scanf("%s",infile);
    printf("Enter the outfile name:\n");
    scanf("%s",outfile);

    if((in=fopen(infile,"r"))==NULL)
    { // 判断文件是否正确读操作
        printf("cannot open infile\n");
        exit(0);
    }

    if((out=fopen(outfile,"w"))==NULL)
    { // 判断文件是否正确写操作
        printf("cannot open outfile\n");
        exit(0);
    }

    //判文件是否结束，如果不结束，则读文件 in 的内容写入到文件 out 之中
    while(!feof(in))
```



```

        fputc(fgetc(in),out);

fclose(in);    //关闭文件
fclose(out);

}

```

运行情况如下:

```

Enter the infile name:
file1.c✓          (输入原有磁盘文件名)
Enter the outfile name:
file2.c✓          (输入新复制的磁盘文件名)

```

程序运行结果是将 file1.c 文件中的内容复制到 file2.c 中去。可以用下面命令验证:

```

c:\>type file1.c
computer and c      (file1.c 中的信息)
c:\>type file2.c
computer and c      (file2.c 中的信息)

```

以上程序是按文本文件方式处理的。也可以用此程序来复制一个二进制文件, 只需将两个 fopen 函数中的“r”和“w”分别改为“rb”和“wb”即可。

分析以上各程序的算法(用框图表示), 解释产生该结果现象的相关知识点及实现语句。

四、 实验内容二（综合及设计类）

Eg10-2: 有 10 个学生, 每个学生的数据包括学号、姓名、3 门课的成绩, 从键盘输入 10 个学生的数据, 要求打印出 3 门课的总、平均成绩, 并计算出每人的平均成绩, 将原有数据和计算出的平均分数存放在磁盘文件 stu.txt 中。

根据题目要求设计解决问题的算法, 画出其流程框图, 写出满足 C 语言规范的源程序, 在 VC++ 环境下调试运行, 对照运行结果分析其结果现象。

五、 扩展内容

理解扩展内容所涉及的相关知识点, 对照下面的程序代码, 理解其编程思路, 举一反三。

Eg10-3: 通讯录管理系统:

该系统通过文本菜单进行操作, 功能包括: 创建通讯录、显示记录、查询记录、修改记录、添加记录、删除记录和记录排序等, 各功能模块均采用独立的函数来表示, 通过主函数直接或是间接调用, 特别注意的是, 通讯录数据采用结构体定义和管理, 并可以直接从文件中读入数据或是将数据写入文件中, 体会这样做的优越性。

(1) 程序如下:

```

#define N 100
#define SIZE 30
#include <stdio.h>
#include <string.h>
#include <conio.h>
struct student
{
    char num[SIZE];
    char name[SIZE];
    char tel[SIZE];
};

/*函数声明 */
void myprint();/* 显示菜单 */
int mycreat(struct student*p,int n);/* 输入通讯录 */
void mydisplay(struct student*p,int n);/* 显示通讯录 */
void mysearch(struct student*p,int n);/* 查找 */
void mymodify(struct student*p,int n);/* 修改通讯录 */
int myadd(struct student*p,int n);/* 增加通讯录 */
int mydelete(struct student*p,int n);/* 删除 */
void mysort(struct student*p,int n);/* 排序 */
void sch_num(struct student*p,int n);/* 按学号查找 */
void sch_name(struct student*p,int n);/* 按姓名查找 */
int loadinfo(struct student*p,int n);/* 载入通讯录信息 */
int saveinfo(struct student*p,int n);/* 保存通讯录信息 */

void main()
{
    /* 文件定义 */
    char choose,yes_no ;
    struct student record[N];/* 存放通讯录信息,共 100 条 */
    int total=0,flag ;/* 通讯录总数 */

    total=loadinfo(record,N);/* 从文件读取记录信息,不超过 100 条 */
    do
    {
        myprint();/* 显示菜单 */

        printf("        请选择:        ");
        choose=getchar();
        switch(choose)
        {
            case '1' :

```

```

        total=mycreat(record,total);
        break ;
case '2' :
    mydisplay(record,total);
    break ;
case '3' :
    mysearch(record,total);
    break ;
case '4' :
    mymodify(record,total);
    break ;
case '5' :
    total=myadd(record,total);
    break ;
case '6' :
    total=mydelete(record,total);
    break ;
case '7' :
    mysort(record,total);
    break ;
case '8' :
    flag=saveinfo(record,total);
    if(flag==1)
        printf("\n 保存成功!\n");
    else
        printf("\n 保存失败!\n");
        break ;
case '0' :
    break ;
    default :
        printf("\n          %c 为非法选项!\n",choose);
    }
    if(choose=='0')break ;
    printf("\n          要继续选择吗(Y/N)?");
    do
    {
        yes_no=getchar();
    } while(yes_no!='Y'&&yes_no!='y'&&yes_no!='N'&&yes_no!='n');
    }while(yes_no!='Y'||yes_no!='y');
}

void myprint()/* 显示菜单信息 */
{

```

```

printf("\n\n\n\n");
printf("                                \n");
printf("        请输入选项编号(0--7)                \n");
printf("            1---批量输入通讯录                \n");
printf("            2---显示通讯录                    \n");
printf("            3---查询通讯录                    \n");
printf("            4---修改通讯录                    \n");
printf("            5---添加通讯录                    \n");
printf("            6---删除通讯录                    \n");
printf("            7---排序通讯录                    \n");
printf("            8---保存通讯录                    \n");
printf("            0---退出                            \n");
printf("                                \n");
}

```

```

int mycreat(struct student*p,int n) /* 批量输入学生信息 */
{
    int i=0 ;
    int m ;
    printf("添加前的记录，共有 <%d>个 : \n",n);
    printf("\n 记录号      学号      姓名      电话号码\n");
    i=0 ;
    while((i<n)&&(strcmp(p->num,"")!=0))
    {
        printf("%d:   %15s%15s%15s\n",i+1,p->num,p->name,p->tel);
        i++;
        p++;
    }

    printf("\n 请输入要输入的学生人数(< %d):  ",N-n);
    scanf("%d",&m);
    i=0 ;
    while(i<m)
    {
        printf("\n");
        printf("请输入第%d 个记录:\n",i+1);
        printf("学号:");
        do
        {
            gets(p->num);
        }while(strcmp(p->num,"")==0);
        printf("姓名:");
        gets(p->name);
    }
}

```

```

        printf("电话号码:");
        gets(p->tel);
        p++;
        i++;
    }
    return m+n ;
}

void mydisplay(struct student*p,int n) /* 显示学生信息 */
{
    /*clrscr();*/
    int i=0 ;
    printf("\n 记录号          学号          姓名          电话号码\n");
    while(i<n)
    {
        printf("%d:    %15s%15s%15s\n",i+1,p->num,p->name,p->tel);
        p++;
        i++;
    }
}

```

```

void mymodify(struct student*p,int n) /* 修改学生信息 */
{
    int i ;
    char nam[10];
    mydisplay(p,n);
    printf("\n 请输入要修改记录的姓名:");
    gets(nam);
    for(i=0;i<n;i++,p++)
    if(strcmp(nam,p->name)==0)break ;
    if(i==n)
        printf("无此人!\n");
    else
    {
        printf("请输入正确的学号:");
        do
        {
            gets(p->num);
        }while(strcmp(p->num,"")==0);
        printf("请输入正确的姓名:");
        gets(p->name);
        printf("请输入正确的电话号码:");
    }
}

```

```

        gets(p->tel);
    }
}

void mysearch(struct student*p,int n) /* 查找指定学生 */
{
    char c ;
    mydisplay(p,n);
    printf("\n 按学号查询(h),还是按姓名查询(m)?");
    c=getchar();
    if(c=='h' || c=='H')
        sch_num(p,n);
        else if(c=='m' || c=='M')
            sch_name(p,n);
    else
        printf("\n 非法字符!\n");
}

void sch_num(struct student*p,int n) /* 按学号查询 */
{
    int i,flag=0 ;
    char nm[10];
    printf("\n 请输入要查询记录的学号:");
    gets(nm);
    for(i=0;i<n;i++)
        if(strcmp(nm,p[i].num)==0)
        {
            printf("\n      学号      姓名      电话号码\n");
            printf("%15s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
            flag=1 ;
        }
    if(flag==0)
        printf("无此人!\n");
}

void sch_name(struct student*p,int n) /* 按姓名查询 */
{
    int i,flag=0 ;
    char nam[10];
    printf("\n 请输入要查询记录的姓名:");
    gets(nam);
    for(i=0;i<n;i++)
        if(strcmp(nam,p[i].name)==0)

```

```

    {
        printf("\n          学号          姓名          电话号码\n");
        printf("%15s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
        flag=1 ;
    }
    if(flag==0)
        printf("无此人!\n");
}

```

int myadd(struct student*p,int n) /* 增加学生信息 */

```

{
    int i=0 ;
    char c ;
    struct student t,*q ;
    printf("添加前的记录: \n");
    printf("\n   学号           姓名           电话号码\n");
    i=0 ;
    q=p ;
    /* 显示原有记录信息 */
    while(i<n)
    {
        printf("%15s%15s%15s\n",q->num,q->name,q->tel);
        i++;
        q++;
    }
    do
    {
        printf("请输入新记录的学号: \n");
        do
        {
            gets(t.num);
        } while(strcmp(t.num,"")==0); /* 学号不能为空 */
        printf("请输入新记录的姓名:  ");
        gets(t.name);
        printf("请输入新记录的电话号码:  ");
        gets(t.tel);
        for(i=0;i<n;i++)
            if(strcmp(t.num,(p+i)->num)==0||strcmp(t.tel,(p+i)->tel)==0)
            {
                printf("学号或电话号码重,要重新输入吗(Y/N)?");
                c=getchar();
                putchar('\n');
                break ;
            }
    }
}

```

```

    }
    /* 找到最后，没有重复 */
    if(i==n)
    {
        *(p+i)=t ;
        printf("添加后的记录:\n");
        printf("\n      学号      姓名      电话号码\n");
        for(i=0;i<n+1;i++)
        printf("%15s%15s%15s\n",(p+i)->num,(p+i)->name,p[i].tel);
        break ;
    }
}
while(c=='y'||c=='Y');
return n+1 ;
}

```

```

int mydelete(struct student*p,int n) /* 删除学生信息 */
{
    int i,x ;
    char c,nm[10],nam[10];
    printf("删除前的记录: \n");
    printf("\n      学号      姓名      电话号码\n");
    for(i=0;i<n;i++)
    printf("%15s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
    printf("按学号删除(h)， 还是按姓名删除(m)?");
    c=getchar();
    if(c=='h'||c=='H')
    {
        printf("\n 请输入要删除记录的学号: ");
        gets(nm);
        for(x=0;x<n;x++)
        if(strcmp(nm,p[x].num)==0)
        break ;
        /* 说明提前结束循环，找到人 */
        if(x<n)
        {
            for(i=x;i<n-1;i++)
            {
                p[i]=p[i+1];
            }
            printf("删除后的记录: \n");
            printf(" 学号      姓名      电话号码\n");

```



```

        for(i=0;i<n-1;i++)
            printf("% 15s% 15s% 15s\n",p[i].num,p[i].name,p[i].tel);
    }
    else
        printf("无此人\n");
}
else
    if(c=='m' || c=='M')
    {
        printf("\n 请输入要删除记录的姓名: ");
        gets(nam);
        for(x=0;x<n;x++)
            if(strcmp(nam,p[x].name)==0)
                break ;
        if(x<n)
        {
            for(i=x;i<N-1;i++)
            {
                p[i]=p[i+1];
            }
            printf("删除后的记录: \n");
            printf(" 学号          姓名          电话号码\n");
            for(i=0;i<n-1;i++)
                printf("% 15s% 15s% 15s\n",p[i].num,p[i].name,p[i].tel);
        }
        else
            printf("无此人\n");
    }
    else
        printf("\n 非法字符\n");
    return n-1 ;
}

```

```

void mysort(struct student*p,int n) /* 排序函数 */
{
    int j,i,k ;
    char c1,c2 ;
    struct student temp ;
    printf("\n 排序前的记录:\n");
    printf("\n 学号          姓名          电话号码\n");
    for(i=0;i<n;i++)
        printf("% 15s% 15s% 15s\n",p[i].num,p[i].name,p[i].tel);
    printf("\n 按学号排序(h), 还是按姓名排序(m)?");
}

```

```

c1=getchar();
if(c1=='h'||c1=='H')
{
    printf("\n 按升序排序 (s), 还是按降序排序 (j) ?");
    c2=getchar();
    /* 升序 */
    if(c2=='s'||c2=='S')
    {
        for(i=0;i<n-1;i++)
        {
            k=i ;
            for(j=i;j<n;j++)
            if(strcmp(p[k].num,p[j].num)<0)
            k=j ;
            temp=p[k];
            p[k]=p[i];
            p[i]=temp ;

        }
        printf("\n 排序后的记录:\n");
        printf("\n   学号           姓名           电话号码\n");
        for(i=0;i<n;i++)
        printf("%15s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
    }

    else if(c2=='j'||c2=='J') /* 降序 */
    {
        for(i=0;i<n-1;i++)
        {
            k=i ;
            for(j=0;j<n;j++)
            if(strcmp(p[k].num,p[j].num)<0)
            k=j ;
            temp=p[k];
            p[k]=p[i];
            p[i]=temp ;

        }
        printf("\n 排序后的记录:\n");
        printf("\n   学号           姓名           电话号码\n");
        for(i=0;i<n;i++)
        printf("%15s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
    }
}

```

```

        else printf("\n 非法字符\n");
    }

else if(c1=='m' || c1=='M') /* 按姓名排序 */
{
    printf("\n 按升序排序 (s), 还是按降序排序 (j) ? ");
    c2=getchar();

    if(c2=='s' || c2=='S')
    {
        for(i=0;i<n-1;i++)
        {
            k=i ;
            for(j=0;j<n;j++)
                if(strcmp(p[k].name,p[j].name)<0)
                    k=j ;
            temp=p[k];
            p[k]=p[i];
            p[i]=temp ;
        }
        printf("排序后的记录:\n");
        printf("\n   学号           姓名           电话号码\n");
        for(i=0;i<n;i++)
            printf("%15s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
    }

    else if(c2=='j' || c2=='J') /* 降序 */
    {
        for(i=0;i<n-1;i++)
        {
            k=i ;
            for(j=i;j<n;j++)
                if(strcmp(p[k].name,p[j].name)<0)
                    k=j ;
            temp=p[k];
            p[k]=p[i];
            p[i]=temp ;
        }
        printf("排序后的记录:\n");
        printf("\n   学号           姓名           电话号码\n");
        for(i=0;i<n;i++)
            printf("%15s%15s%15s\n",p[i].num,p[i].name,p[i].tel);
    }
}

```

```

        }
        else printf("\n 非法字符! \n");
    }
    else printf("\n 非法字符! \n");
}

int loadinfo(struct student*p,int n) /* 载入通讯录信息 */
{
    FILE*fp ;
    int i,all=0 ;
    char filename[30]="txl.dat" ;
    fp=fopen(filename,"r");
    if(fp==NULL)
    {
        /* printf("\n***不能打开文件%s, 或%s 文件不存在!***",filename,filename);
*/
        return 0 ;
    }
    fscanf(fp,"%d\n",&all);
    if(all>N) /* 读取记录总数 */
    {
        printf("文件记录数太多，不能读取");
        return 0;
    }

    for(i=0;i<all;i++)/* 读取记录信息 */
    {
        fscanf(fp,"%s %s %s\n",p[i].num,p[i].name,p[i].tel);
    }
    fclose(fp); /* 关闭文件 */
    return all ;
}

int saveinfo(struct student*p,int n)
{
    FILE*fp ;
    int i,all=0 ;
    char filename[30]="txl.dat" ; /*文件可以用记事本打开*/
    fp=fopen(filename,"w");
    if(fp==NULL)
    {
        printf("\n***不能建立或打开文件%s!***",filename);
        return 0 ;
    }

```

```
    }  
    fprintf(fp,"%d\n",n);  
    for(i=0;i<n;i++)  
        fprintf(fp,"%s %s %s\n",p[i].num,p[i].name,p[i].tel);  
    fclose(fp);  
    return 1 ;  
}
```

(2) 编译直到程序无错误;

(3) 运行;

(4) 查看运行结果

附录 1：常见错误提示信息的英汉对照

Ambiguous operators need parentheses : 不明确的运算需要用括号括起
Ambiguous symbol 'xxx' : 不明确的符号
Argument list syntax error : 参数表语法错误
Array bounds missing : 丢失数组界限符
Array size too large : 数组尺寸太大
Bad character in parameters : 参数中有不适当的字符
Bad file name format in include directive : 包含命令中文件名格式不正确
Bad ifdef directive syntax : 编译预处理 `ifdef` 有语法错
Bad undef directive syntax : 编译预处理 `undef` 有语法错
Bit field too large : 位字段太长
Call of non-function : 调用未定义的函数
Call to function with no prototype : 调用函数时没有函数的说明
Cannot modify a const object : 不允许修改常量对象
Case outside of switch: 漏掉了 `case` 语句
Case syntax error : `Case` 语法错误
Code has no effect 代码不可达不可能执行到
Compound statement missing{ 分程序漏掉 "{"
Conflicting type modifiers 不明确的类型说明符
Constant expression required 要求常量表达式
Constant out of range in comparison 在比较中常量超出范围
Conversion may lose significant digits 转换时会丢失意义的数字
Conversion of near pointer not allowed 不允许转换近指针
Could not find file 'xxx' 找不到 XXX 文件
Declaration missing ; 说明缺少 ";"
Declaration syntax error 说明中出现语法错误
Default outside of switch Default 出现在 `switch` 语句之外
Define directive needs an identifier 定义编译预处理需要标识符
Division by zero 用零作除数
Do statement must have while Do-while 语句中缺少 `while` 部分
Enum syntax error 枚举类型语法错误
Enumeration constant syntax error 枚举常数语法错误
Error directive :xxx 错误的编译预处理命令
Error writing output file 写输出文件错误
Expression syntax error 表达式语法错误
Extra parameter in call 调用时出现多余错误
File name too long 文件名太长
Function call missing) 函数调用缺少右括号

Fuction definition out of place 函数定义位置错误
 Fuction should return a value 函数必需返回一个值
 Goto statement missing label Goto 语句没有标号
 Hexadecimal or octal constant too large 16 进制或 8 进制常数太大
 Illegal character 'x' 非法字符 x Illegal initialization 非法的初始化
 Illegal octal digit 非法的 8 进制数字
 Illegal pointer subtraction 非法的指针相减
 Illegal structure operation 非法的结构体操作
 Illegal use of floating point 非法的浮点运算
 Illegal use of pointer 指针使用非法
 Improper use of a typedefsymbol 类型定义符号使用不恰当
 In-line assembly not allowed 不允许使用行间汇编
 Incompatible storage class 存储类别不相容
 Incompatible type conversion 不相容的类型转换
 Incorrect number format 错误的数据格式
 Incorrect use of default Default 使用不当
 Invalid indirection 无效的间接运算
 Invalid pointer addition 指针相加无效
 Irreducible expression tree 无法执行的表达式运算
 Lvalue required 需要逻辑值 0 或非 0 值
 Macro argument syntax error 宏参数语法错误
 Macro expansion too long 宏的扩展以后太长
 Mismatched number of parameters in definition 定义中参数个数不匹配
 Misplaced break 此处不应出现 break 语句
 Misplaced continue 此处不应出现 continue 语句
 Misplaced decimal point 此处不应出现小数点
 Misplaced elif directive 不应编译预处理
 elif Misplaced else 此处不应出现 else
 Misplaced else directive 此处不应出现编译预处理 else
 Misplaced endif directive 此处不应出现编译预处理 endif
 Must be addressable 必须是可以编址的
 Must take address of memory location 必须存储定位的地址
 No declaration for function 'xxx' 没有函数 xxx 的说明 No stack 缺少堆栈
 No type information 没有类型信息
 Non-portable pointer assignment 不可移动的指针（地址常数）赋值
 Non-portable pointer comparison 不可移动的指针（地址常数）比较
 Non-portable pointer conversion 不可移动的指针（地址常数）转换
 Not a valid expression format type 不合法的表达式格式
 Not an allowed type 不允许使用的类型
 Numeric constant too large 数值常太大
 Out of memory 内存不够用

Parameter 'xxx' is never used 能数 xxx 没有用到
Pointer required on left side of -> 符号->的左边必须是指针
Possible use of 'xxx' before definition 在定义之前就使用了 xxx (警告)
Possibly incorrect assignment 赋值可能不正确
Redeclaration of 'xxx' 重复定义了 xxx
Redefinition of 'xxx' is not identical xxx 的两次定义不一致
Register allocation failure 寄存器定址失败
Repeat count needs an lvalue 重复计数需要逻辑值
Size of structure or array not known 结构体或数组大小不确定
Statement missing ; 语句后缺少"; "
Structure or union syntax error 结构体或联合体语法错误
Structure size too large 结构体尺寸太大
Sub scripting missing] 下标缺少右方括号
Superfluous & with function or array 函数或数组中有多余的"&"
Suspicious pointer conversion 可疑的指针转换
Symbol limit exceeded 符号超限
Too few parameters in call 函数调用时的实参少于函数的参数
Too many default cases Default 太多(switch 语句中一个)
Too many error or warning messages 错误或警告信息太多
Too many type in declaration 说明中类型太多
Too much auto memory in function 函数用到的局部存储太多
Too much global data defined in file 文件中全局数据太多
Two consecutive dots 两个连续的句点
Type mismatch in parameter xxx 参数 xxx 类型不匹配
Type mismatch in redeclaration of 'xxx' xxx 重定义的类型不匹配
Unable to create output file 'xxx' 无法建立输出文件 xxx
Unable to open include file 'xxx' 无法打开被包含的文件 xxx
Unable to open input file 'xxx' 无法打开输入文件 xxx
Undefined label 'xxx' 没有定义的标号 xxx
Undefined structure 'xxx' 没有定义的结构 xxx
Undefined symbol 'xxx' 没有定义的符号 xxx
Unexpected end of file in comment started on line xxx 从 xxx 行开始的注解
尚未结束文件不能结束
Unexpected end of file in conditional started on line xxx 从 xxx 开始的条件
语句尚未结束文件不能结束
Unknown assemble instruction 未知的汇编结构
Unknown option 未知的操作
Unknown preprocessor directive: 'xxx' 不认识的预处理命令 xxx
Unreachable code 无路可达的代码
Unterminated string or character constant 字符串缺少引号
User break 用户强行中断了程序

Void functions may not return a value Void 类型的函数不应有返回值

Wrong number of arguments : 调用函数的参数数目错

'xxx' not an argument :xxx 不是参数

'xxx' not part of structure xxx 不是结构体的一部分

xxx statement missing (xxx 语句缺少左括号

xxx statement missing) xxx 语句缺少右括号

xxx statement missing ; xxx 缺少分号

xxx' declared but never used 说明了 xxx 但没有使用

xxx' is assigned a value which is never used 给 xxx 赋了值但未用过

附录 2: Visual C++ 6.0 集成开发环境的使用

- (1) 启动 Visual C++ 6.0 集成开发环境
- (2) 选择菜单“开始/程序/Microsoft Visual Studio 6.0/Microsoft Visual C++ 6.0”,进入 Visual C++ 6.0 集成开发环境用户界面,如图 1-1 所示。

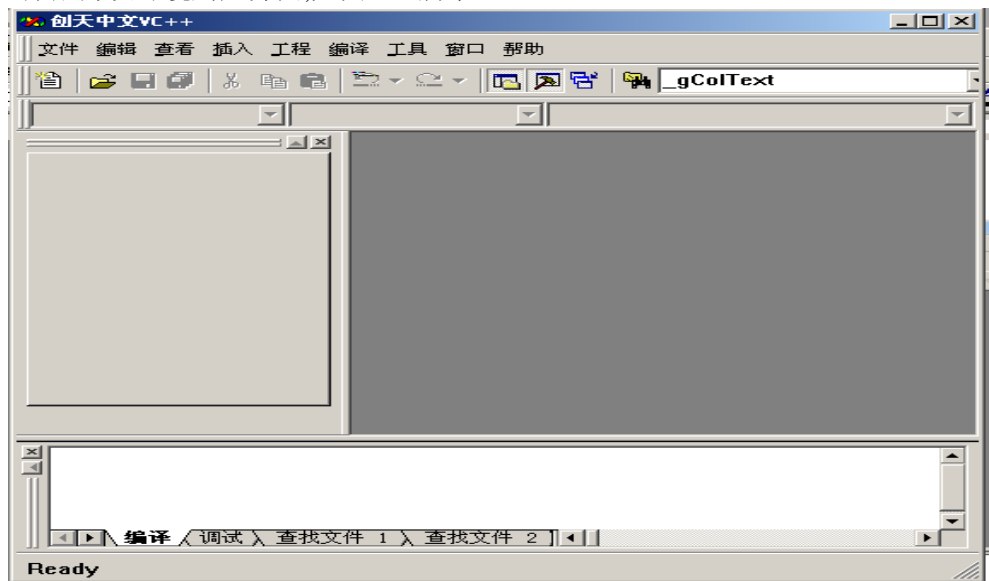


图 1-1 Visual C++ 6.0 用户界面

2. 创建新工程

- (1) 单击菜单“文件/新建”,显示“新建: 对话框,如图 1-2 所示。
- (2) 在此对话框的列表栏中,选择“Win32 Console Application”(Win32 控制台应用程序)。在“工程”栏中输入一个工程名,如 Exam1。在位置文本框中选择工程所在的路径,如图中所示 e:\C 实验\exam1。单击“确定”按钮。

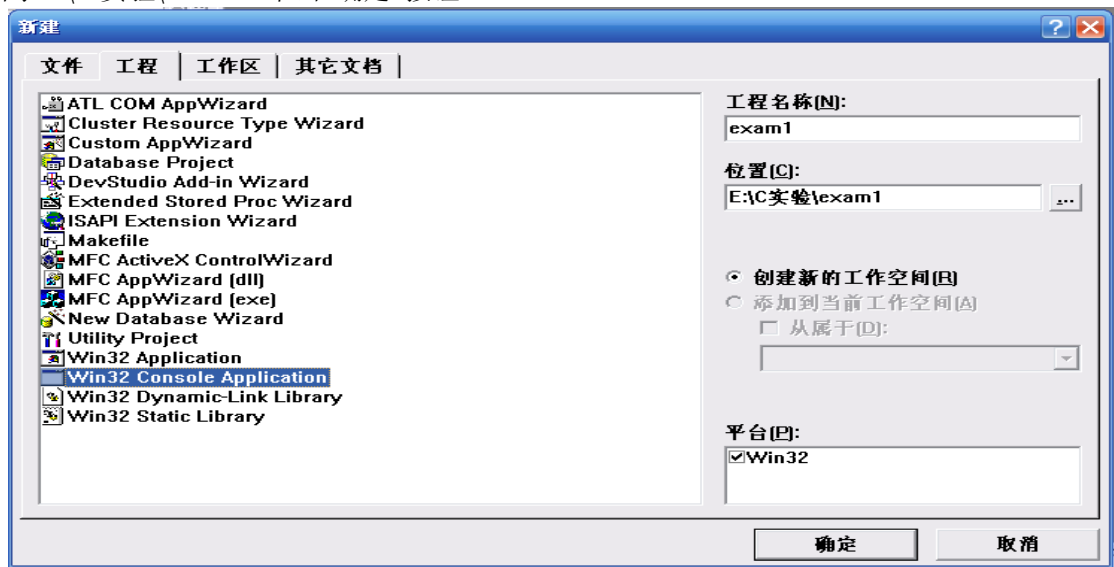


图 1-2 “新建”对话框

(3)在弹出的“Win32 Console Application-步骤共 1 步”对话框中选择“一个空工程”单选项。然后单击“完成”按钮，如图 1-3 所示。



图 1-3 “Win32 Console Application-Step 1 of 1” 对话框

(4)在“新建工程信息”对话框中单击“确认”按钮，完成工程创建过程。

3. 创建 C++源程序文件

(1) 新建 C++源程序：选择“文件”菜单项，出现一个下拉式菜单，如图 1-4 所示，再选择该菜单中的“新建”选项（热键为 Ctrl+N）。弹出“新建”窗口，在“新建”窗口文件名的文本框中输入文件名 eg1-1，如图 1-4 所示。若要编辑新的源程序，可以再次打开“文件”菜单，重新建立文件，步骤如上所述；也可以点击“文件”菜单中的“打开工作区”选项，打开一个已经存在的源文件。

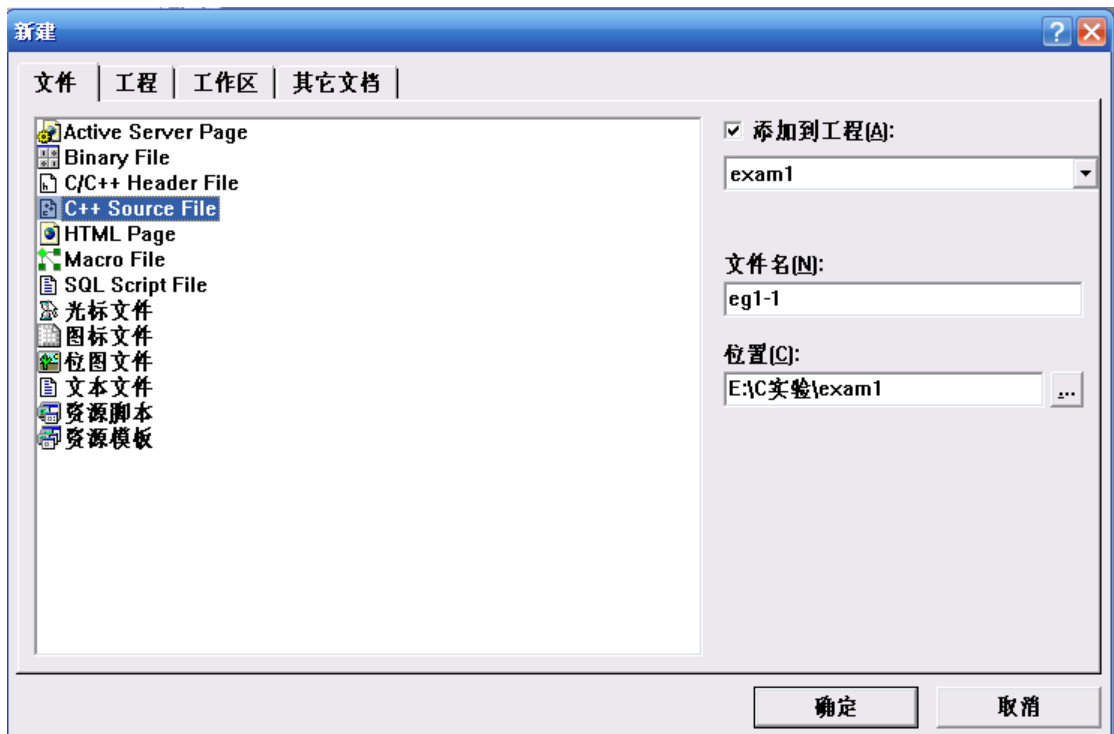


图 1-4 “新建”窗口

- (2) 生成 C++源文件：单击“确定”按钮，出现编辑屏幕，如图 1-5 所示。同时生成了一个 eg1-1.cpp 的源程序。

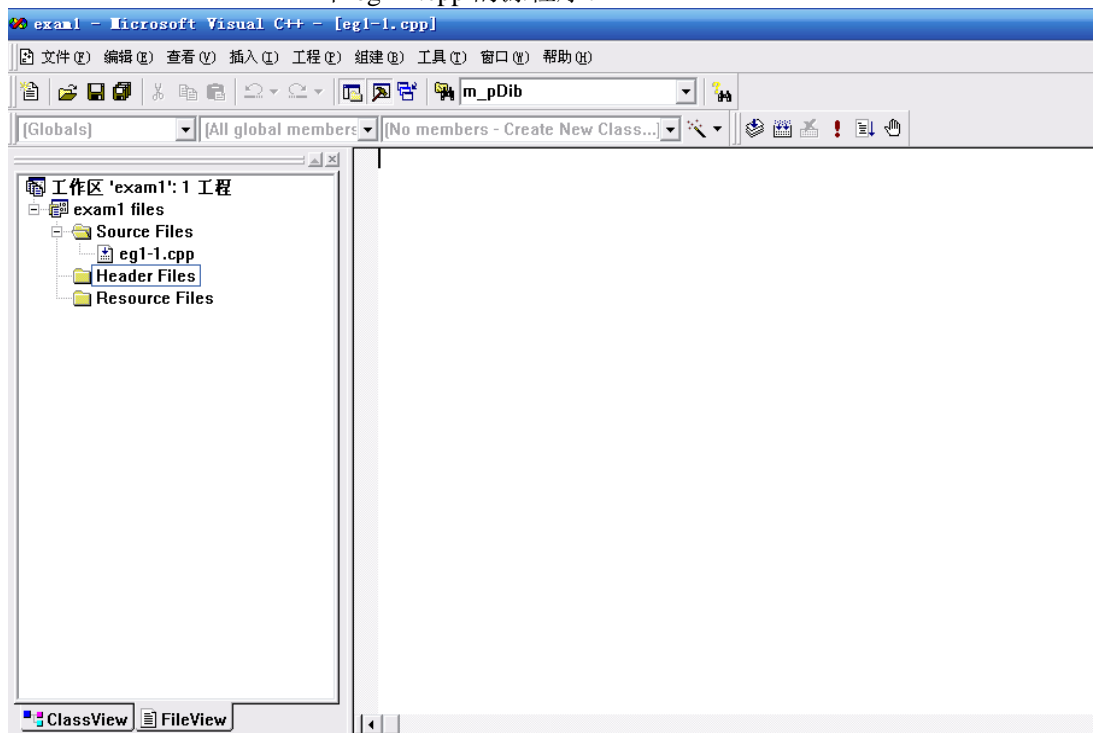


图 1-5 “C++”源程序编辑屏幕

- (3) 编辑 C++源程序：在“C++源程序编辑”窗口下编辑 C++源程序。在编辑窗口键入 C++源程序。如图 1-6 所示，键入如下源程序：

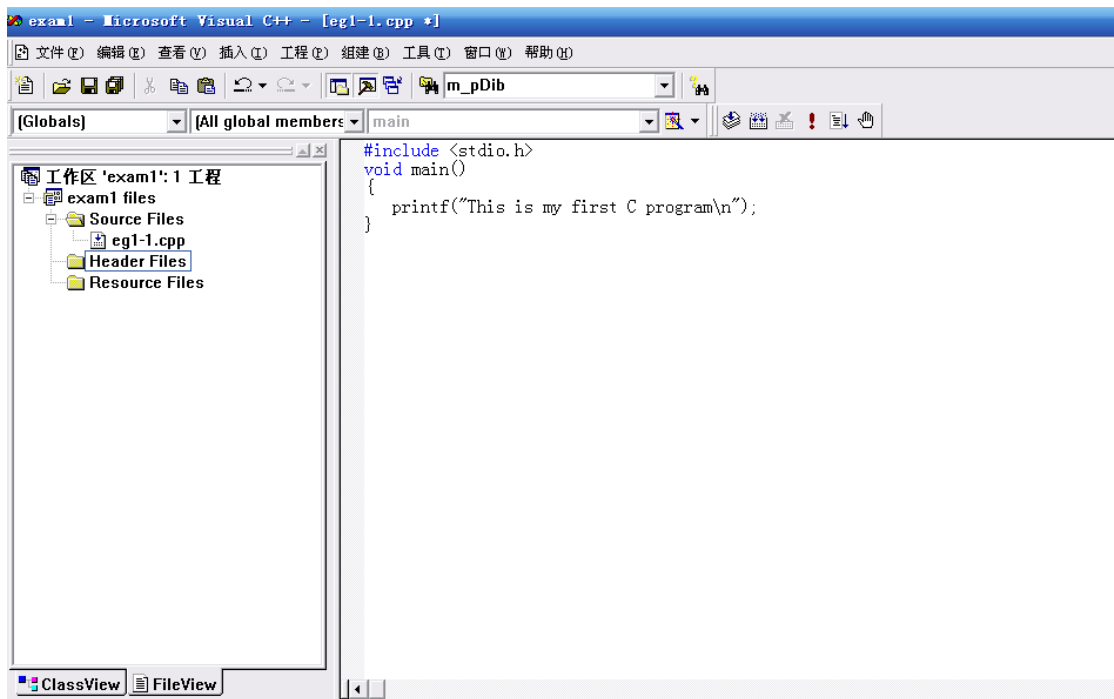


图 1-6 编辑 C++源程序

4. 编译、连接和运行源程序

(1) 编译：选择菜单项“**编译**”，出现“**编译**”的下拉菜单，在该下拉式菜单中选择“**编译 eg1-1.cpp**”菜单项，这时系统开始对当前的源程序进行编译，在编译过程中，将所发现的错误显示在屏幕下方的“**编译**”窗口中。所显示的错误信息中指出该错误所在行号和该错误的性质。用户可根据这些错误信息进行修改。上述程序的“**编译**”窗口，如图 1-7 所示。

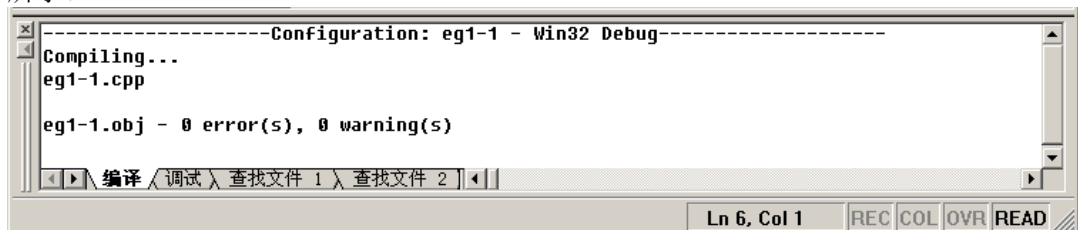


图 1-7 “编译”窗口

(2) 组建：编译无错误后，可进行连接生成可执行文件（.exe），这时选择“**组建**”下拉菜单中的“**组建 eg1-1.exe**”选项。“**编译**”窗口出现如图 1-8 所示的信息说明编译连接成功，并生成以源文件名为名字的可执行文件（eg1-1.exe）。

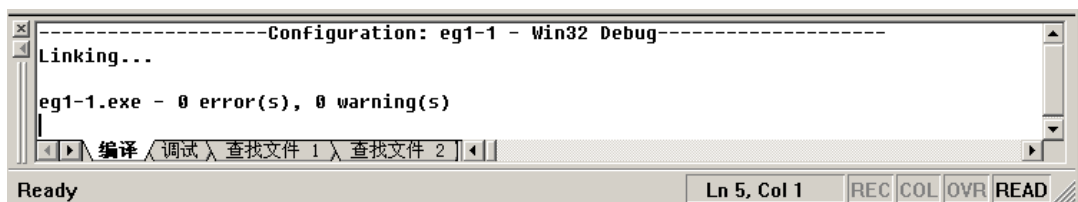


图 1-8 编译连接信息

(3) 执行：组建成功后（至少是 0 error(s)，允许有警告错误存在），选择“**组建**”菜

单项中“执行 eg1-1.exe”选项。这时，运行该可执行文件，并将结果显示在另外一个显示执行文件输出结果的窗口中，如图 1-9 所示。

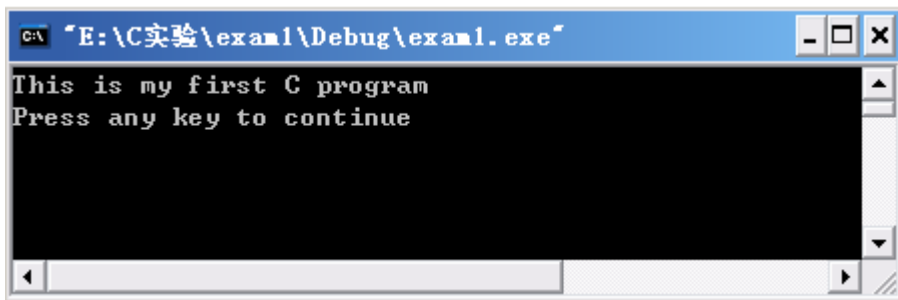


图 1-9 运行 C++程序结果

说明 1：步骤（1）和步骤（2）可以合二为一，选择“组建”下拉菜单中的“全部重建”选项。

说明 2：步骤（1）—步骤（3）中任何一步均可能出现错误，则一定要仔细查找错误后，更正，再重新完成相应的步骤，或重新“编辑->编译->连接->运行”。

说明 3：可以采用“调试工具”作为有效的手段，进行调试、跟踪、监视等等，这是培养一个好的程序员必备的“利器”。

5. 关闭打开工作区、打开工作区

单击菜单“文件/关闭工作区”，关闭工作区。

单击菜单“文件/打开工作区”，在弹出的对话框中选定“e：\C++ 实验\exam1\exam1.dsw”，单击“打开”按钮，则可打开工作区，对已建立的工程文件进行修改。