

第 1 部分 上机实训

实训一 编程环境和简单程序的运行



实训目的和实训要求

1. 了解和使用 Turbo C 2.0 集成环境编写运行简单的程序。
2. 了解和使用 Visual C++集成开发环境编写运行简单的程序。
3. 掌握 C 语言源程序的建立、编辑、修改、保存及编译和运行。



实训内容

1.1 Turbo C 2.0 编程环境

1. 假设 Turbo C 2.0 系统安装在 C 盘的 TC 文件夹下，文件结构如图 1-1 所示。

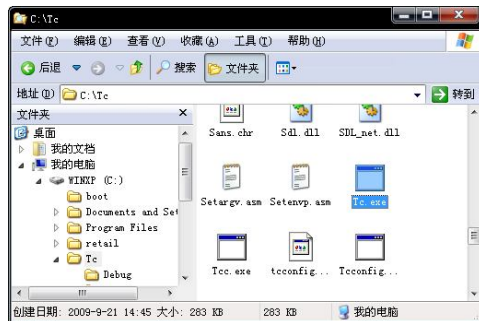


图 1-1 TC 系统文件夹

其中的 Tc.exe 是系统的主程序。双击 Tc.exe 进入编程环境，如图 1-2 所示。

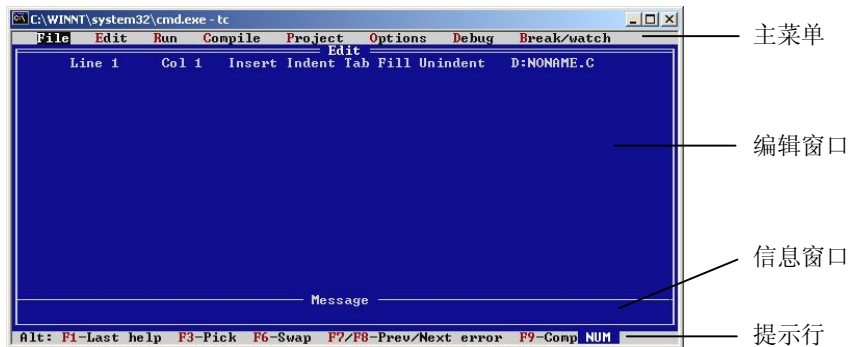


图 1-2 Turbo C 编程环境界面

观察 Turbo C 2.0 的主屏幕, 准备进行基本环境配置。

按 Alt+O 组合键选择菜单 Options, 再用光标选择 Directories 子菜单, 如图 1-3 所示。



图 1-3 设置头文件和库文件目录界面

其中 Include directories 和 Library directories 所对应的文件夹必须是实际存在的文件夹, 按图设置完成后按 Esc 键返回。再用光标键选择 Options 下的 Save options 子菜单, 按 Enter 键保存配制信息, 界面如图 1-4 所示。

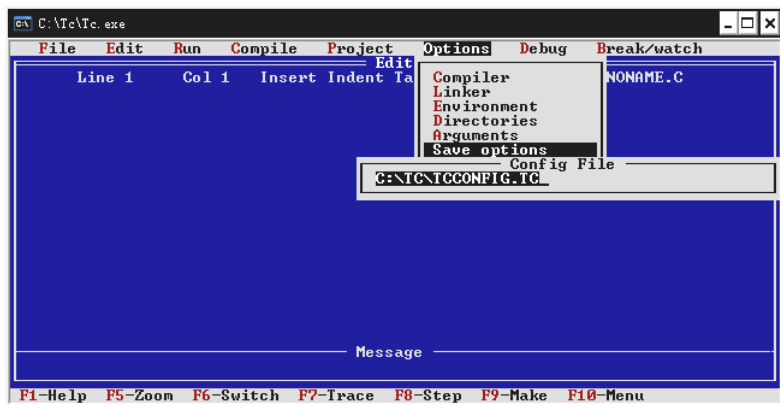


图 1-4 保存设置的界面

配置信息将保存在 TCCONFIG.TC 文件中。

2. 编辑和调试程序

在 Turbo C 2.0 编程环境下, 输入下面的程序:

```
#include <stdio.h>
void main()
{
    printf("Hello World!");
}
```

将文件存盘 (F2), 命名为 ex1-1.c, 试编译连接 (F9)、运行 (Ctrl+F9), 并按 Alt+F5 组合键查看运行结果。

Turbo C 2.0 主要操作命令的快捷键如表 1-1 所示。

表 1-1 Turbo C 2.0 主要操作命令的快捷键

快捷键	功能
F1	激活帮助窗口，显示与当前光标所在位置有关的操作提示信息
F2	将当前文件以指定的文件名存盘
F3	装入指定文件
F4	将程序执行到光标所在的行暂停
F5	缩放当前窗口
F6	切换活动窗口
F7	调试程序，执行单步操作，可进入被调用函数
F8	调试程序，执行单步操作，不进入被调用函数
F9	编译、连接源程序，生成可执行文件
F10	激活主菜单
Esc	返回
Ctrl+F9	运行程序
Alt+F5	用户窗口，用来查看运行结果
Alt+X	退出 Turbo C
Ctrl+Y	删除光标所在的一行
Alt+F3	选择一个最近打开的文件

1.2 Visual C++ 6.0 编程环境

1. 了解 Visual C++ 界面

Visual C++ 6.0 集成开发环境被划分成 4 个主要区域：菜单栏和工具栏、工作区窗口、代码编辑窗口和输出窗口，如图 1-5 所示。

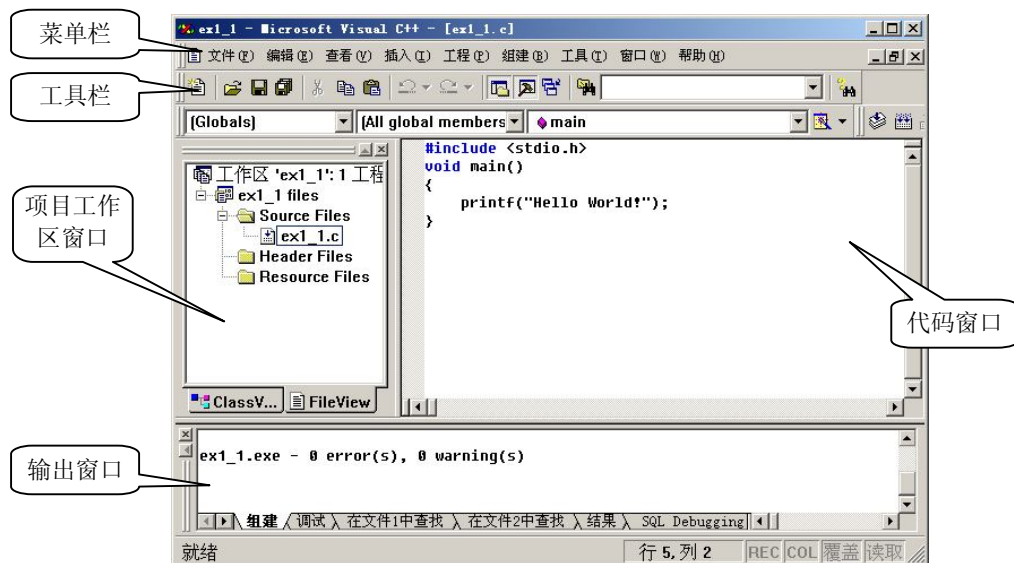


图 1-5 Visual C++ 6.0 集成开发环境

2. 编写和调试简单的 C 程序

(1) 创建空工程。

1) 选择“文件”→“新建”命令。

2) 单击“工程”选项卡,选择“Win32 Console Application (32 位控制台应用程序)”,输入工程名 ex1_1,确保单选按钮“创建新的工作空间”被选定,输入工程位置 C:\TEST\ex1_1,注意 C:\TEST 文件夹需要事先建好,如图 1-6 所示。



图 1-6 “新建”对话框

3) 在随后弹出的向导对话框中选择“一个空工程”,单击“完成”按钮,显示新建工程的有关信息。

4) 单击“确定”按钮,创建空工程结束。

此时为工程 ex1_1 创建了 C:\TEST\ex1_1 文件夹,并在其中生成了 ex1_1.dsp、ex1_1.dsw 文件和 Debug 文件夹。Debug 文件夹用于存放编译、连接过程中产生的文件。

(2) 创建 C 源文件。

1) 选择“文件”→“新建”命令。

2) 单击“文件”选项卡,选择 C++ Source File,并输入源程序文件名 ex1_1.c,如图 1-7 所示。

3) 输入、编辑源程序。

在这个阶段,C:\TEST\ex1_1 文件夹中创建了 ex1_1.c。

(3) 编译、连接和运行。

选择“编译”→“执行 ex1_1.exe”命令进行编译、连接和运行,会在输出区中显示有关信息。若程序有错,则进行编辑,如图 1-8 所示。

编译、连接和运行可以分别进行。

1) 编译 (Ctrl+F7)。

选择“编译”→“编译 ex1_1.c”命令,编译结果显示在输出区中,如果没有错误,则生成 ex1_1.obj。

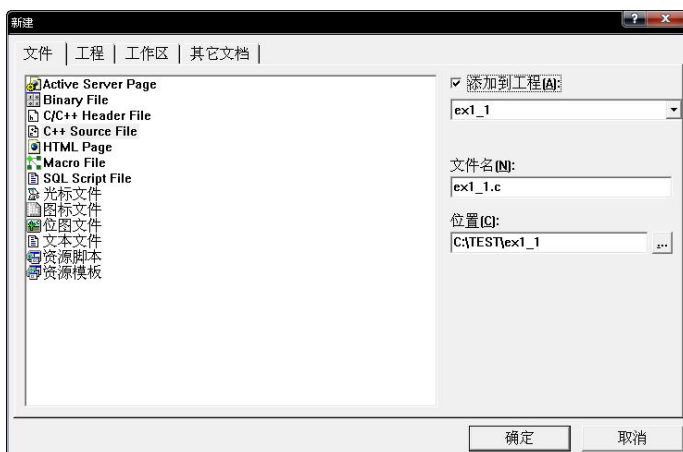


图 1-7 “新建”对话框的“文件”选项卡

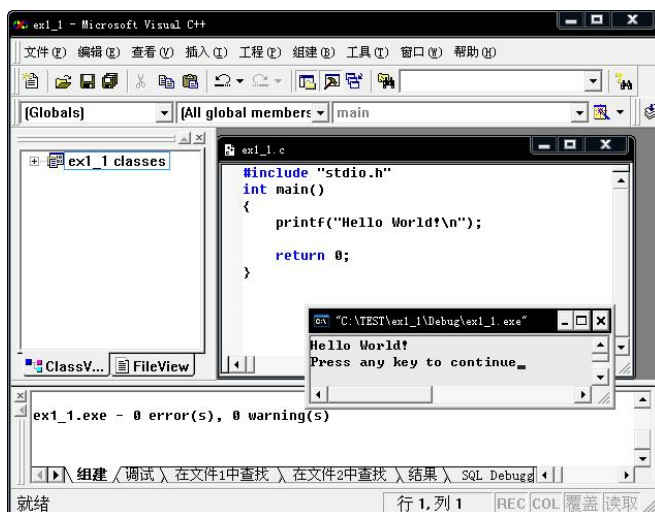


图 1-8 编译和运行的界面

2) 连接 (F7)。

选择“编译”→“构建 ex1_1.exe”命令，连接信息显示在输出区中，如果没有错误，则生成 ex1_1.exe。

3) 运行 (Ctrl+F5)。

选择“编译”→“执行 ex1_1.exe”命令。

在 C:\TEST\ex1_1\Debug 中生成了 ex1_1.obj、ex1_1.exe 等文件。ex1_1.obj 是编译后产生的目标代码文件，ex1_1.exe 是最终生成的可执行文件。

至此，一个简单 C 程序的编写、调试过程结束，所产生的文件如图 1-9 所示。

ex1_1.c 文件是最重要的一个文件，源程序就保存在这个文件中，其他文件一般都是系统自动生成的。但是，在 Visual C++ 中，仅有 .c 文件是不能直接编译、连接的，需要首先用“构建”命令让系统自动创建一个工程并将 ex1_1.c 文件加入到该工程中，然后才能执行各种操作。因此，程序员可以只复制 .c 文件，若要复制整个工程的文件夹，也请删除 Debug 文

文件夹,因为它占用相当多的存储空间。

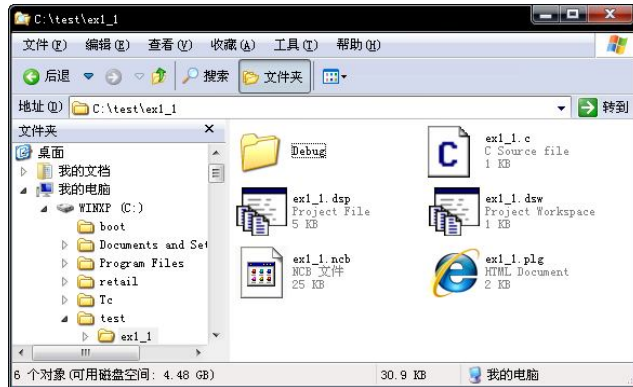


图 1-9 ex1_1 工程的文件夹

1.3 利用 Turbo C 2.0 或 Visual C++ 6.0 编辑和调试下面的程序

1. 已知圆的半径,求圆的周长和面积。

```
#include "stdio.h"
void main()
{
    int r;
    float l,s;
    r=5;
    l=2*3.14159*r;
    s=3.14159*r*r;
    printf("r=%d,l=%f,s=%f\n",r,l,s);
}
```

运行结果如图 1-10 所示。

```
r=5,l=31.415899,s=78.539750
Press any key to continue_
```

图 1-10 程序 1.1 的运行结果

2. 输入矩形的两个边长,求矩形的面积。

```
#include "stdio.h"
double area(double a, double b); /*声明子函数*/
void main()
{
    double a,b;
    double s;
    printf("Please input a,b:");
    scanf("%lf%lf",&a,&b); /*输入矩形的两个边长*/
    s=area(a,b); /*调用函数 area*/
    printf("area is %lf\n",s); /*输出矩形的面积*/
}
```

```

double area(double a, double b)          /*定义子函数求矩形的面积*/
{
    double s;
    s=a*b;
    return s;                            /*返回矩形面积的值*/
}

```

运行结果如图 1-11 所示。

```

Please input a,b:10 20
area is 200.000000
Press any key to continue

```

图 1-11 程序 1.2 的运行结果

注意其中 10,20 是运行后由用户从键盘输入的，输入后按 Enter 键。

3. 模仿上面的程序，设计程序计算并输出一个圆的面积。

实训二 数据类型、运算符和表达式



实训目的和实训要求

1. 掌握 C 语言基本标识符的写法，能正确命名变量和文件。
2. 掌握 C 语言数据类型的基本概念，熟悉如何定义整型、字符型、实型变量。
3. 学会使用 C 语言的算术运算符、赋值运算符、逗号运算符，能正确地运用它们构成表达式。
4. 掌握不同数据类型的数据混合运算时，数据类型的转换规则。



实训内容

2.1 分析下面的程序，并上机验证运行结果

```

#include "stdio.h"
void main()
{
    int a, b, c;
    unsigned u, v;
    a=32767+1; b=131071; c= -98304;
    u=131071; v= -98304;
    printf("a=%d, b=%d, c=%d\n u=%u, v=%u\n", a, b, c, u, v);
}

```

程序运行结果如图 1-12 所示。

```

a=-32768, b=-1, c=-32768      a=32768, b=131071, c=-98304
u=65535, v=32768             u=131071, v=4294868992
Press any key to continue

```

图 1-12 程序 2.1 的运行结果

图中, 左侧是在 Turbo C 下运行的结果, 右侧是在 Visual C++ 下运行的结果。请读者分析其中不同的原因 (将 `u`、`v` 的类型改为 `long` 再试试)。

2.2 分析下面的程序, 并上机验证运行结果

```
#include <stdio.h>
void main()
{
    int i=5,j;

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

    i=5;
    j=(++i)+(++i)+(i++);
    printf("i=%d,j=%d\n",i,j);

    i=5;
    printf("i++=%d,i++=%d\n",i++,i++);
    printf("i=%d\n",i);

    i=5;
    printf("++i=%d,++i=%d\n",++i,++i);
    printf("i=%d\n",i);

    i=5;j=6;
    printf("i+j=%d,j++=%d\n",i+j,j++);
    printf("i=%d,j=%d\n",i,j);

    i=5;j=6;
    printf("i+j=%d,j++=%d\n",i+j,++j);
    printf("i=%d,j=%d\n",i,j);
}
```

运行程序, 结果如图 1-13 所示。

```
i=7, j=12
i=8, j=21
i++=6, i++=5
i=7
++i=7, ++i=6
i=7
i+j=12, j++=6
i=5, j=7
i+j=12, j++=7
i=5, j=7
```

(a) 在 Turbo C 下运行

```
i=7, j=12
i=8, j=21
i++=5, i++=5
i=7
++i=7, ++i=6
i=7
i+j=11, j++=6
i=5, j=7
i+j=12, j++=7
i=5, j=7
```

(b) 在 Visual C++ 下运行

图 1-13 程序 2.2 的运行结果

2.3 分析下面的程序, 并上机验证运行结果

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



```

{
    char c;
    int a;
    double d;

    /*自动类型转换*/
    d = 100+'A'+3.1415926;
    printf("d=%lf\n",d);

    /*存在赋值类型转换*/
    c = 65;
    d = c;
    printf("c=%c,d=%lf\n",c,d);

    /*强制类型转换*/
    d = 3.14159;
    a = (int)d;
    printf("a=%d\n",a);
}

```

运行结果如图 1-14 所示。

```

d=168.141593
c =A,d=65.000000
a=3
Press any key to continue

```

图 1-14 程序 2.3 的运行结果



1. 分析下列程序的运行结果。

```

#include <stdio.h>
void main()
{
    float x, y;
    x = 1234567890000000000.0;
    y = x + 0.12345;
    printf("x=%f, y=%f\n", x, y);

    x = 3.1415926;
    y = x - 0.0001;
    printf("x=%.3f, y=%.3f\n", x, y);

    y = x + 0.0000005;
    printf("x=%.7f, y=%.7f\n", x, y);
}

```

2. 编程实现，不借助中间变量交换 a 和 b 中的值。（提示：a=a+b;b=a-b;a=a-b;）

实训三 简单的 C 程序设计



实训目的和实训要求

1. 掌握 C 语言中赋值表达式和赋值语句的使用。
2. 正确理解、选择输入格式字符所控制的数据对象、数据输入 (输出) 形式和方法。
3. 掌握输出函数 printf() 和 putchar() 的用法, 能正确使用各种输出格式。
4. 掌握输入函数 scanf() 和 getchar() 的用法, 能正确使用各种输入格式。



实训内容

3.1 输入并运行下面的程序。

```
#include <stdio.h>
void main()
{
    char c='A';
    int a = 65 , b = -100;
    long d = 100 ;
    float x = 3.141592631415 ,y = -3141592631.415;
    double dx = 3.141592631415;
    printf("c=%d, c=%c, c=%x\n",c,c,c);
    printf("a=%d, a=%x, a=%o,a=%c\n",a,a,a,a);
    printf("a=%d, a=%10d,a=%-10d, a=%+d\n",a,a,a,a);
    printf("b=%d, b=%10d,b=%-10d, b=%+d\n",b,b,b,b);
    printf("d=%d, d=%ld,d=%x, d=%lx\n",d,d,d,d);

    printf("x=%f,x=%6.f,x=%0.3f,x=%6.3f,x=%10.3fn",x,x,x,x,x);
    printf("y=%f,y=%6.f,y=%10.f\n",y,y,y);

    printf("dx=%f,dx=%6.f,dx=%0.3f,dx=%6.3f,dx=%10.3fn",dx,dx,dx,dx,dx);
    printf("x=%0.8f,dx=%0.8fn",x,dx);
}

```

运行结果如图 1-15 和图 1-16 所示。

```
c=65, c=A, c=41
a=65, a=41, a=101, a=A
a=65, a=65, a=65, a=65, a=+65
b=-100, b=-100, b=-100, b=-100
d=100, d=6553600, d=0, d=64
x=3.141593, x=3.141593, x=3.142, x= 3.142, x= 3.142
y=-3141592576.000000, y=-3141592576.000000, y=-3141592576.000000
dx=3.141593, dx=3.141593, dx=3.142, dx= 3.142, dx= 3.142
x=3.14159274, dx=3.14159263
```

图 1-15 程序 3.1 的运行结果 (Turbo C)

```

c=65, c=A, c=41
a=65, a=41, a=101, a=A
a=65, a=        65, a=65        , a+=65
b=-100, b=      -100, b=-100   , b=-100
d=100, d=100, d=64, d=64
x=3.141593, x=    3, x=3.142, x= 3.142, x=    3.142
y=-3141592576.000000, y=-3141592576, y=-3141592576
dx=3.141593, dx=    3, dx=3.142, dx= 3.142, dx=    3.142
x=3.14159274, dx=3.14159263

```

图 1-16 程序 3.1 的运行结果 (Visual C++)

修改程序，将其中的变量定义及初始化部分改为：

```

char c=66;
short int a = 65 , b = -100;
long d = -100 ;
float x = 314159263.1415 , y = -3.141592631415;
double dx = 3141592631415926;

```

观察程序运行结果并思考为什么。

3.2 输入并运行下面的程序。

```

#include <stdio.h>
void main()
{
    char c;
    int a,b;
    float x,y;
    double dx,dy;
    printf("1.Input a,b(100 -200):");
    scanf("%d%d",&a,&b);
    printf("a=%d,b=%d\n",a,b);

    printf("2.Input a,b(100,-200):");
    scanf("%d,%d",&a,&b);
    printf("a=%d,b=%d\n",a,b);

    printf("3.Input a,b,c(100 -200A):");
    scanf("%d%d%c",&a,&b,&c);
    printf("a=%d,b=%d,c=%c\n",a,b,c);

    printf("4.Input a,b,c(100,-200,9):");
    scanf("%d,%d,%c",&a,&b,&c);
    printf("a=%d,b=%d,c=%c\n",a,b,c);

    printf("5.Input a,c,b(100A-200):");
    scanf("%d%c%d",&a,&c,&b);
    printf("a=%d,b=%d,c=%c\n",a,b,c);

    printf("6.Input a,b(1112222):");
    scanf("%3d%4d",&a,&b);

```

```

printf("a=%d,b=%d\n",a,b);

printf("7.Input a,b(1112223333):");
scanf("%3d%*3d%4d",&a,&b);
printf("a=%d,b=%d\n",a,b);

printf("8.Input x,y(3.1415926 31415926):");
scanf("%f%f",&x,&y);
printf("x=%f,y=%f\n",x,y);

printf("9.Input dx,dy(3.1415926 31415926):");
scanf("%lf%lf",&dx,&dy);
printf("dx=%lf,dy=%lf\n",dx,dy);

printf("10.Input x,c,y(3.1415926A31415926):");
scanf("%f%c,%f",&x,&c,&y);
printf("x=%f,y=%f,c=%c\n",x,y,c);
}

```

程序运行结果如图 1-17 所示。

```

1.Input a,b<100 -200>:100 -200
a=100,b=-200
2.Input a,b<100,-200>:100,-200
a=100,b=-200
3.Input a,b,c<100 -200A>:100 -200A
a=100,b=-200,c=A
4.Input a,b,c<100,-200,9>:100,-200,9
a=100,b=-200,c=9
5.Input a,c,b<100A-200>:100A-200
a=100,b=-200,c=A
6.Input a,b<1112222>:1112222
a=111,b=2222
7.Input a,b<1112223333>:1112223333
a=111,b=3333
8.Input x,y<3.1415926 31415926>:3.1415926 31415926
x=3.141593,y=31415926.000000
9.Input dx,dy<3.1415926 31415926 31415926>:3.1415926 31415926
dx=3.141593,dy=31415926.000000
10.Input x,c,y<3.1415926A31415926>:3.1415916A3.1415926
x=3.141592,y=31415926.000000,c=A

```

图 1-17 程序 3.2 的运行结果



思考

- scanf()与 getchar(), printf()与 putchar()各有什么相同点和不同点?
- scanf("%3d%*3d%4d",&a,&b);中“%*3d”是什么意思?
- 设计程序,输入三角形的三条边,求该三角形的面积。
- 设计程序,求方程 $x^2-2x-11=0$ 的根。

实训四 选择结构程序设计



实训目的和实训要求

1. 学会使用 C 语言的关系运算符和逻辑运算符，并能运用这些运算符构成表达式。
2. 理解 C 语言对逻辑型数据的表示方法。
3. 熟练掌握各种类型的 if 语句和 switch 语句。
4. 熟悉条件运算符和 if 语句之间的转化方法。
5. 熟悉选择结构的程序段的执行过程。



实训内容

- 4.1 分析下面的程序，并上机验证运行结果。

```
#include <stdio.h>
void main()
{
    int a,b;
    a=b=0;
    a || ++b; printf("%d,%d\n",a,b);
    a=b=1;
    a || ++b; printf("%d,%d\n",a,b);
    a=b=0;
    a && ++b; printf("%d,%d\n",a,b);
    a=b=1;
    a && ++b; printf("%d,%d\n",a,b);
}
```

程序运行结果如图 1-18 所示。

```
0,1
1,1
0,0
1,2
```

图 1-18 程序 4.1 的运行结果

- 4.2 分析下面的程序，并上机验证运行结果。

```
#include <stdio.h>
int show(int n,int a) /*n 用于显示调用次序，a 用作返回值*/
{
    printf("(%d,%d)",n,a);
    return a;
}
void main()
{
```

```

int a;
a = show(1,1) || show(2,2) && show(3,3); printf("a=%d\n",a);
a = show(1,0) || show(2,0) && show(3,1); printf("a=%d\n",a);
a = show(1,0) || show(2,1) && show(3,2); printf("a=%d\n",a);
}

```

程序运行结果如图 1-19 所示。

```

<1,1>a=1
<1,0><2,0>a=0
<1,0><2,1><3,2>a=1

```

图 1-19 程序 4.2 的运行结果

4.3 输入出租车类型和里程，计算打车的费用。计算方式为：3 公里以内 8 元；3 公里以上 0 车型每公里 1.5 元，1 车型每公里 2 元。

```

#include <stdio.h>
void main()
{
    int taxiType;
    float s;
    float money;
    printf("Input taxi type(0,1):");
    scanf("%d",&taxiType);

    printf("Input s:");
    scanf("%f",&s);

    if(s < 3)
        money = 8;
    else
        if(taxiType == 0)
            money = 8 + (s-3)*1.5;
        else
            money = 8 + (s-3)*2;

    printf("money=%.2f\n",money);
}

```

分别输入车型、里程为 0、2.5，0、6 和 1、6 后，三次运行的结果如图 1-20 所示。

```

Input taxi type(0,1):0
Input s:2.5
money=8.00

Input taxi type(0,1):0
Input s:6
money=12.50

Input taxi type(0,1):1
Input s:6
money=14.00

```

图 1-20 程序 4.3 的运行结果

4.4 输入一个正整数作为年份，编程判断该年是不是闰年。若是输出“YES”，否则输出“NO”。

【分析】满足下面条件之一即为闰年：

- 能被 4 整除，但不能被 100 整除
- 能被 400 整除

```
#include "stdio.h"
void main()
{
    int year;
    scanf("%d",&year);
    if(year%400 == 0 || ( year%4 == 0 && year%100 != 0))
        printf("Yes\n");
    else
        printf("NO\n");
}
```

4.5 输入成绩 score，判断成绩的等级。

```
#include <stdio.h>
void main()
{
    int a;
    printf("Input score:");
    scanf("%d",& score);
    if(score <0 || score >100)
        printf("Input data error\n");
    else
        switch(score /10)
        {
            case 10:
            case 9 : printf("A\n");break;
            case 8 : printf("B\n");break;
            case 7 : printf("C\n");break;
            case 6 : printf("D\n");break;
            default: printf("E\n");
        }
}
```

调试并运行程序，分别输入 115、95、85、75、65、55、-5，观察运行结果。如果在程序中删除其中的某个 break，再次观察运行结果有什么不同。



1. 输入 2010 年的一个月份，输出这个月的天数。（2010 年为平年）
2. 从键盘输入一个 4 位整数，如 1234，分别输出其个位数、十位数、百位数和千位数。
3. 编写程序实现两个数的交换。