

目 录

第 1 篇 C 程序设计实习指导	(1)
第 1 章 简单的 C 程序设计	(1)
第 2 章 数据运算	(8)
第 3 章 选择结构程序设计	(12)
第 4 章 循环结构程序设计	(16)
第 5 章 数组	(20)
第 6 章 指针	(25)
第 7 章 函数及其调用	(29)
第 8 章 复合数据类型	(35)
第 9 章 文件操作	(39)
第 10 章 编译预处理	(43)
第 11 章 综合练习	(45)
第 2 篇 C 程序设计模拟试题	(55)
第 1 章 程序结构	(55)
第 2 章 基本数据类型	(57)
第 3 章 运算符与表达式	(60)
第 4 章 C 语句	(64)
第 5 章 选择结构	(66)
第 6 章 循环结构	(76)
第 7 章 数组	(84)
第 8 章 指针	(92)
第 9 章 函数及其调用	(99)
第 10 章 函数间数据的传递	(111)
第 11 章 结构、位段、联合、枚举和类型定义	(120)
第 12 章 编译预处理	(126)
第 13 章 文件	(128)
第 14 章 综合题	(133)
第 15 章 程序改错题	(139)
第 16 章 编程题	(148)
附录 1 C 程序设计模拟试题参考答案	(159)
附录 2 Turbo C 2.0 编译出错信息	(166)

第 1 篇 C 程序设计实习指导

第 1 章 简单的 C 程序设计

【目的】 熟悉 Turbo C 2.0 集成环境的使用,了解用计算机解决实际问题的基本步骤,掌握 C 程序的基本格式和 C 程序的运行过程。

【实习 1】 (1 学时)

[题目] 一个笼子里关有若干只鸡和兔。某人数了一下,鸡和兔的头(用 t 表示)共 30 个,脚(用 f 表示)共 100 只。请编写程序计算笼子中的鸡和兔各多少只。

[指导]

一、先建立问题的数学模型

假设鸡为 x 只,兔为 y 只,则该问题的数学模型为:

$$a_1x + b_1y = t$$

$$a_2x + b_2y = f$$

用克莱姆法则,不难求出

$$x = (b_2t - b_1f) / (a_1b_2 - a_2b_1)$$

$$y = (a_1f - a_2t) / (a_1b_2 - a_2b_1)$$

二、编写程序

编写程序如下:

```
#include <stdio.h>
main()
{
    int x, y, f, t;
    int a1 = 1, b1 = 1, a2 = 2, b2 = 4;
    scanf("%d%d", &t, &f);
    x = (b2 * t - b1 * f) / (a1 * b2 - a2 * b1);
    y = (a1 * f - a2 * t) / (a1 * b2 - a2 * b1);
    printf("Chickens = %d, Rabbits = %d \n", x, y);
}
```

三、启动 Turbo C2.0 集成环境

集成环境如图 1-1 所示。

四、操作步骤

操作步骤如下:

1) 按任意键清除窗口中的版本信息;

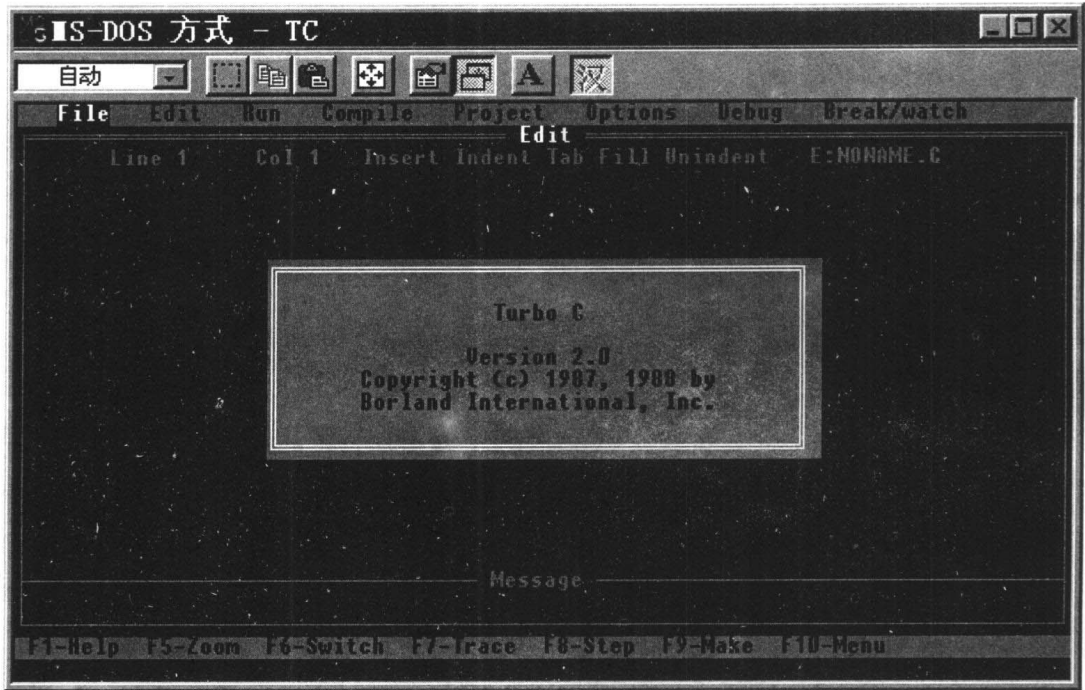


图 1-1 Turbo C 2.0 集成环境初始画面

- 2) 按 Alt + F 组合键弹出 File 菜单, 并选 New 项;
- 3) 输入源程序, 见图 1-2;

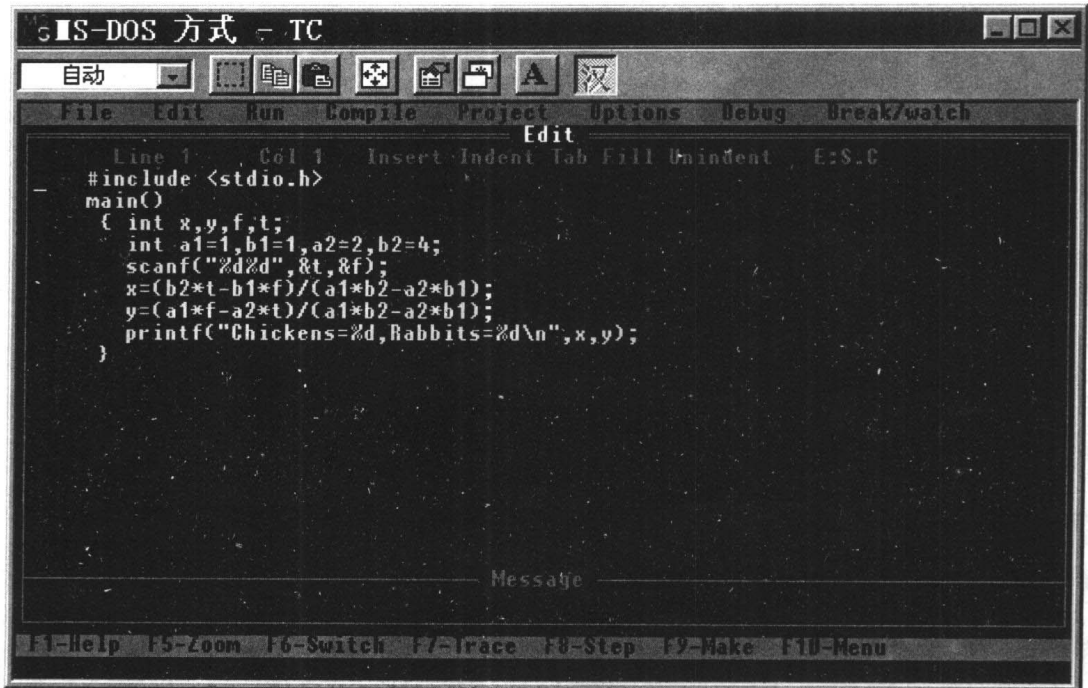


图 1-2 源程序录入

- 4) 按 Alt + F + S 组合键将源程序存盘, 将屏幕上显示的 NONAME.C 修改为希望的文件名(注意, 文件名一定要以 .C 作为扩展名);

5) 按 Alt+R 组合键及回车键编译并运行程序, 出现如图 1-3 所示的运行屏幕, 输入 t 和 f 的值(例如, t 为 50, f 为 160);

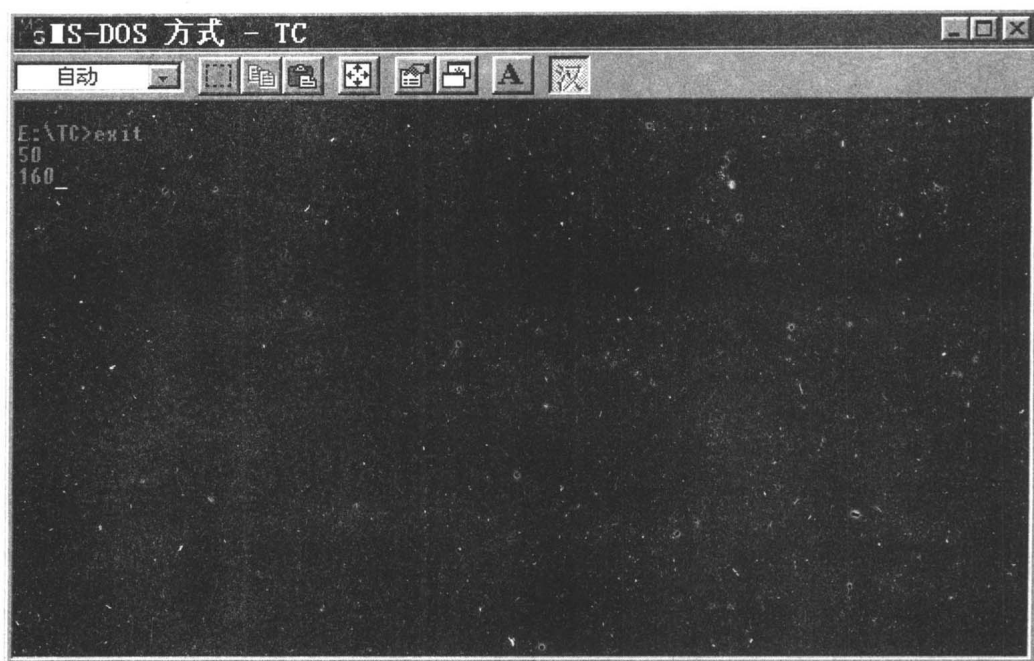


图 1-3 运行屏幕

6) 按 Alt+F+O 组合键(按住 Alt 键, 再依次按 F 键和 O 键)可以观看运行结果, 如图 1-4 所示。

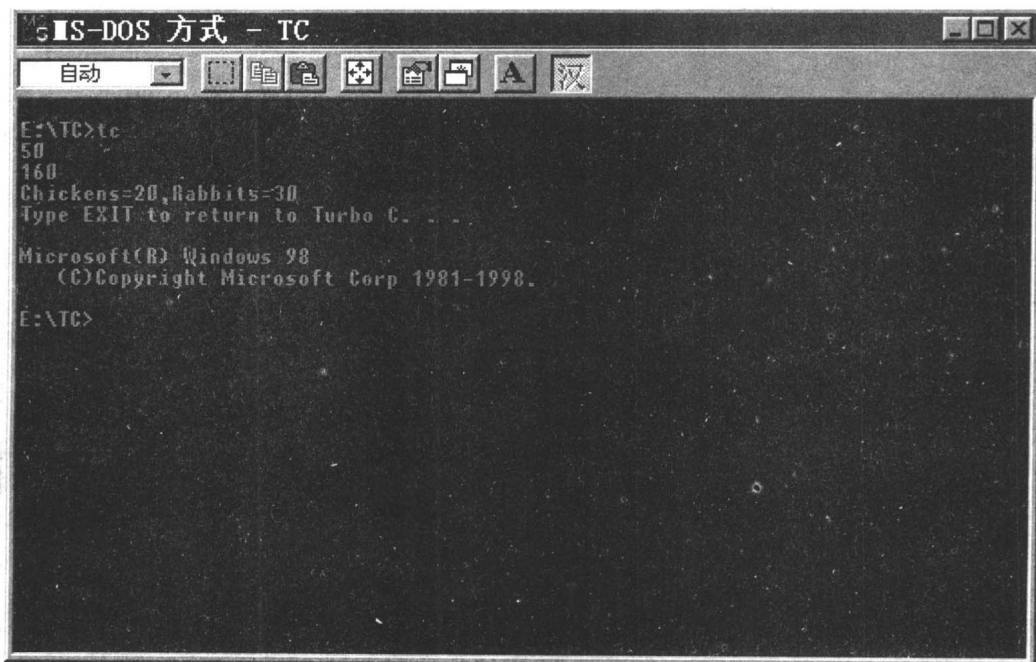


图 1-4 运行结果

7)如果要返回 Turbo C 2.0 集成环境,可以在 DOS 系统提示符下键入 EXIT 命令,如果要退出 Turbo C 2.0 集成环境,返回操作系统,可以按 Alt + F + Q 组合键或者直接关闭 Turbo C 2.0 集成环境窗口。

【实习 2】 (1 学时)

[题目] 下面程序中每个注释行后面的某行中有一个错误,请调试纠正。

```
#include <stdio.h>
main()
{ int a;
  /* found */
  float b;
  scanf("%d%f", a, b);
  /* found */
  printf("a = %d, b = %f \n", a, b);
  printf("a + b = %d \n", a + b);
}
```

[指导]

1. 启动 Turbo C 2.0 集成环境。
2. 输入源程序。
3. 按 Alt + R 进行源程序的编译和运行,系统自动进入运行屏幕。这时,应输入 a 和 b 的值(例如分别为 25 和 34.5)。
4. 按 Alt + F + O 观看运行屏幕,可以看到屏幕显示(图 1-5)。

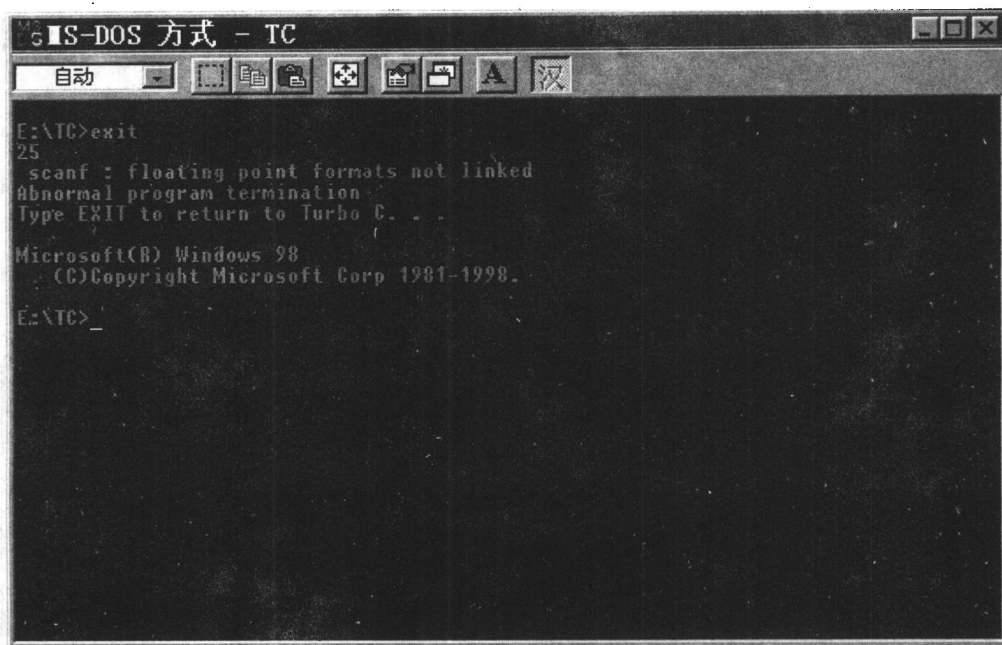


图 1-5 程序非正常运行的提示

5. 根据屏幕提示可知,编译时因为浮点数的格式不对而未能连接成功,程序非正常终止。这时,可以用 exit 命令返回集成环境,按 Alt + C 组合键将源程序先编译成 .obj 文件,以便发现错误。结果系统给出了 6 处警告信息,见图 1-6。按任意键后,可看到具体的错误信息及错误所在行,错误的性质均为“变量未定义”,分别出现在第 6、8、9 行,见图 1-7。

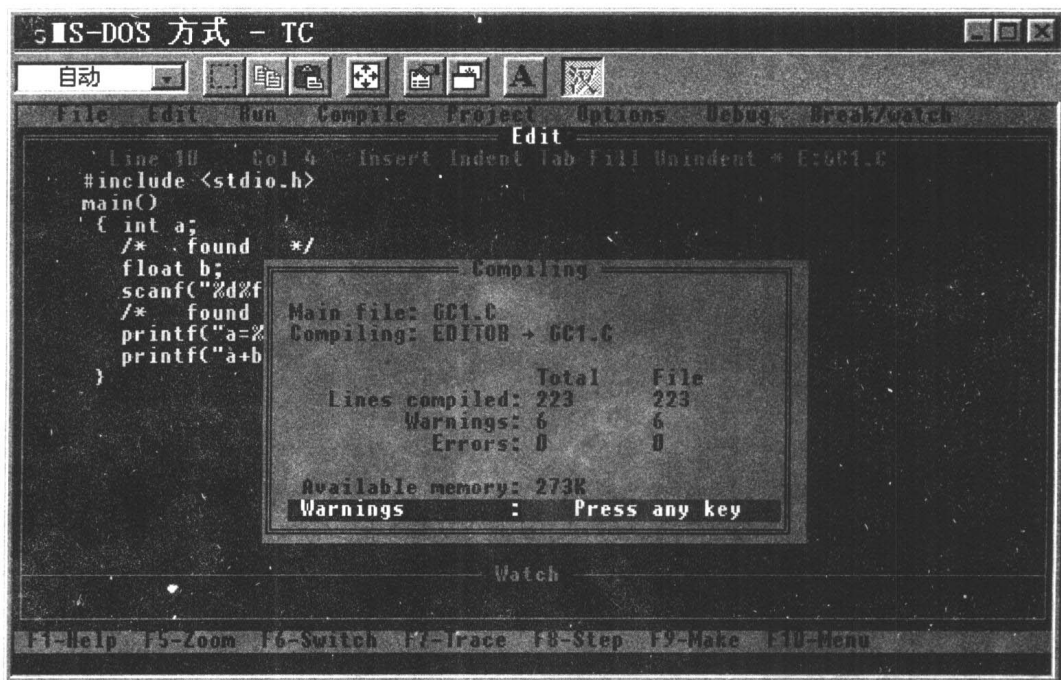


图 1-6 编译出错信息

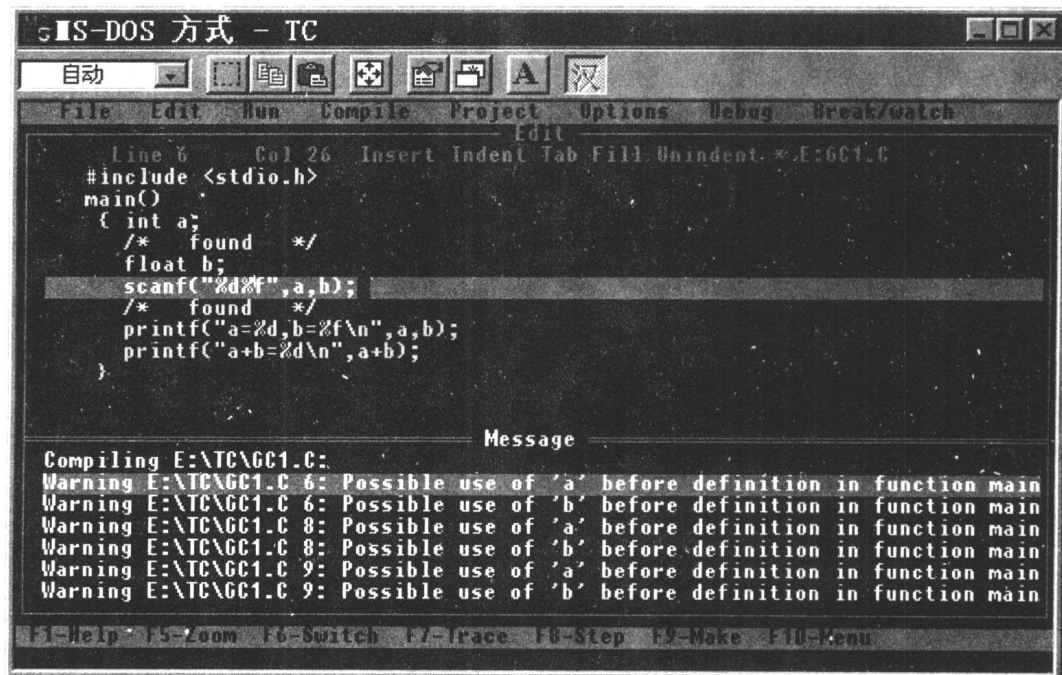


图 1-7 编译错误详细信息

6. 为了修改错误,可以按回车键,直接进入编辑状态。第6行为反白显示,这是第一个错误所在行。根据错误提示,属于数据格式问题,容易确定错误的原因是“a”和“b”不是地址量。将它们分别修改为“&a”和“&b”。由于第8、9行中的错误是因第6行的错误造成的,随着第6行错误的纠正会自动消失,因而不必修改。用 Alt + F + S 组合键将修改过的源程序存盘。

7. 用 Alt + R 组合键编译并运行。系统自动进入运行屏幕,应重新输入数据 25 和 34.5。

8. 按 Alt + F + O 返回运行屏幕,显示如图 1-8。

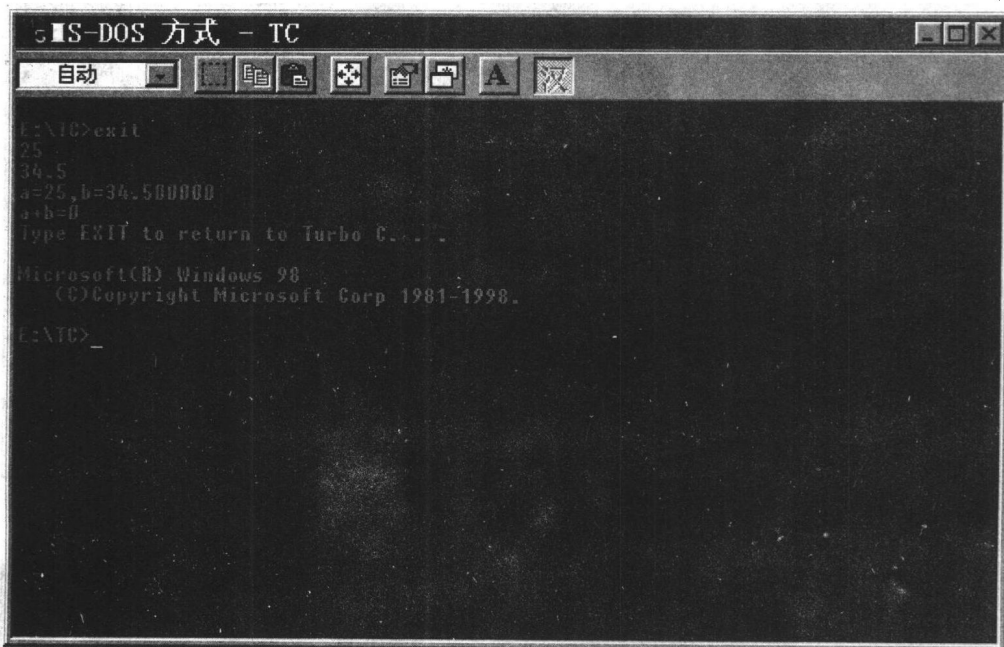


图 1-8 实习 2 题目的运行屏幕

注意到 $a + b$ 的结果是不对的,问题肯定出在输出语句上。只要将第 9 行中的“%d”改为“%f”后,重新编译和运行,就可以获得正确的结果。

【思考题】

1. 试编写一个程序,从键盘输入矩形的两条边长,计算该矩形的面积。
2. 试编写一个程序,从键盘输入变量 a 和 b 的值,将它们打印出来。然后,将两者的值进行交换,并打印交换后的 a 、 b 值。例如, a 和 b 的输入值分别是 5 和 8,交换后, a 的值为 8 而 b 的值为 5。
3. 下面的程序用来求长方体的面积和体积,请调试该程序,纠正其中的错误。

```
#include <stdio.h>
main()
{ float a, b, c, volume, area;
  /* * * * found * * * */
  scanf("%d%d%d", a, b, c)
  /* * * * found * * * */
```



```
area = 2(a * b + b * c + a * c);  
volume = a * b * c;  
/* * * * * found * * * * */  
printf("area = %.2f, volume = %.4f \n", area, volume);  
{
```

第2章 数据运算

【目的】 熟练掌握 C 语言中的各种数据类型及数据的运算;熟练掌握变量的定义方法,包括数据类型及存储类型的说明、定义的位置及其对变量的作用域和生命期的影响,以及局部变量和全局变量的概念;熟练掌握算术、赋值、关系、逻辑、测试数据长度和位运算符的优先级和结合性;熟练掌握不同类型数据间的转换和运算规则;熟练掌握赋值表达式、算术表达式、关系表达式、逻辑表达式、逗号表达式的书写方法和求值规则。

【实习 1】 (1 学时)

[题目] 运行下面的程序,分析运行结果。

```
#include <stdio.h>
main()
{   int i=5,j=5,x,y,z,a,b,c;
    char ch1,ch2;
    i++;
    printf("i= %d,j= %d \n", ++i,j++);
    x=10;
    x+=x-=x-x;
    printf("x= %d \n",x);
    x=y=z=1;
    printf("++x||++y&&++z= %d \n", ++x||++y&&++z);
    c=246;
    a=c/100%9;
    b=(-1)&(-1);
    printf("a= %d,b= %d \n",a,b);
    ch1='A'+ '5' - '3';
    ch2='A'+ '6' - 3;
    printf("ch1= %c, ch2= %c \n",ch1,ch2);
}
```

[指导]

1. 启动 Turbo C 2.0 集成环境,直接将所给源程序录入,并以 ex2.c 存盘。
2. 按 Alt + R 及回车键编译。
3. 如果输入程序时没有出错,程序就能立即运行;如果编译出错,请将源程序与本题给出的程序对照,修改正确后再执行步骤 2。
4. 程序运行后仍回到 TC 集成环境窗口,用 Alt + F + O 组合键进入 DOS 运行屏幕,可以看到程序运行结果为:

```
i=7,j=5
```

```

x = 20
++x || ++y && ++z = 1
a = 2, b = -1
ch1 = C, ch2 = t

```

5. 对程序运行结果分析如下。

1) 先看第一个输出结果。变量 i 的初值为 5, 经过赋值语句 $i++$ 后, 值为 6。在输出语句中又执行表达式 $++i$, 即先加 1 再取 i 的值, 所以 i 的最后结果为 7。变量 j 只在输出语句中执行了表达式 $j++$, 即先取 j 的值, 再使 j 加 1, 所以输出的 j 值为 5。

2) 再看第二个输出结果。变量 x 的起始值为 10, 在执行表达式 $x += x - x$ 时, 从右到左进行计算, 即先计算 $x - x$, 其值为 0, 然后计算 $x - 0$, 结果为 $x = 10$; 最后计算 $x + x$, 得到 $x = 20$ 。

3) 第三个输出结果是一个逻辑表达式的值, 结果不是 0 就是 1。经过 $++$ 运算后, x 、 y 、 z 的值均为 21, 经逻辑“与”运算后, 即 $21 \&\& 21$, 结果为 1, 再经过逻辑“或”运算, 即 $21 || 1$, 结果为 1。

4) 再看第四个输出结果。计算 a 的值时, 先用 246 整除 100, 结果为 2, 再用 2 与 9 取余, 即 2 除以 9 的余数, 结果仍为 2; b 的计算则是一个按位“与”运算。我们知道, -1 的二进制补码表示为“全 1”, 即 1111111111111111。两个“全 1”经按位“与”运算后仍为“全 1”, 所以结果仍为 -1 ;

5) 最后一个输出结果: 由于大写字母 A 的 ASCII 代码值为 65, 数字字符 5 和 3 的 ASCII 代码值分别为 53 和 51, 由此可以计算 $ch1$ 的值为 $65 + 53 - 51 = 67$ 。它代表大写字母 C; $ch2$ 的值为 $65 + 54 - 3 = 116$, 它代表小写字母 t。

【实习 2】 (1 学时)

[题目] 设 a 、 b 均为整型变量, 且 $b = 5$, 请编一个程序计算 $a = 2 + (b + = b + +, b + 8, + + b)$ 及 $b = (a = 32767, a + 1)$ 的值。

[指导]

当初学者对某些操作或运算没有十分把握时, 应学会通过上机编程验证。

1. 本题的参考程序为:

```

main()
{
    int a, b = 5;
    a = 2 + (b + = b + +, b + 8, + + b);
    printf("a = %d \n", a);
    b = (a = 32767, a + 1);
    printf("b = %d \n", b);
}

```

2. 录入并调试上述程序, 如果发现错误, 可根据编译中的错误提示并参照附录 2 修改源程序。

3. 对修改后的源程序要重新编译和运行。

4. 本题的运行结果为:

```
a = 14
```

b = -32768

5. 对程序运行结果分析如下:

1) 先计算逗号表达式中 $b++$ 、 $b+8$ 、 $++b$ 的值分别为 5、6+8、6+1, 因此该逗号表达式的值为 7, 然后计算圆括号中的 b 值为 $b=5+7=12$, 最后计算 a 的值为 $2+12=14$;

2) 在计算第二个表达式即 $b=(a=32767, a+1)$ 的值时, 要特别注意整型数的计算机内部表示 (32767 在计算机内存中的表示为 0111111111111111, 进行加 1 运算后, 结果为 1000000000000000。由于符号位为 1, 说明它是一个负数, 应把它视为补码, 显然它表示 -32768 而不是 +32768)。

【思考题】

1. 运行下面的程序, 并分析运行结果 (本题涉及变量的作用域和生命期)。

```
#include <stdio.h>
main()
{ int a, b, c;
  a = 1, b = 2, c = 3;
  a + = 1; b + = 1; c + = b;
  { int b, c;
    b = 4;
    c = b * 3;
    a + = c;
    printf("First: a = %d, b = %d, c = %d \n", a, b, c);
    a = a + c;
    printf("Second: a = %d, b = %d, c = %d \n", a, b, c);
  }
  printf("Third: a = %d, b = %d, c = %d \n", a, b, c);
}
```

2. 分析下列程序的输出结果 (本题主要涉及各种表达式的计算)。

```
#include "stdio.h"
main()
{ int a = 11, b = 10;
  a - = b + 1;
  printf("a = %d \n", a);
  a + = b + 1;
  printf("a = %d \n", a);
  a * = b + 1;
  printf("a = %d \n", a);
  a / = b + 1;
  printf("a = %d \n", a);
  a % = b + 1;
  printf("a = %d \n", a);
}
```

```
a << = b;
printf("a = %d \n", a);
a >> = b;
printf("a = %d \n", a);
a &= b;
printf("a = %d \n", a);
a ^= b;
printf("a = %d \n", a);
a |= b;
printf("a = %d \n", a);
a = (3 * 5, a * 4, a + 5);
printf("a = %d \n", a);
printf("%d \n", b += b + +, b + 8, + + b);
```

第3章 选择结构程序设计

【目的】 熟练掌握各种选择结构,包括 if-else 及其嵌套、if-else if-else 形式的多重选择以及 switch 形式的多重选择结构的使用。

【实习 1】 (1 学时)

[题目] 任给三条边长 a、b、c,如果能构成三角形,则计算它的面积;否则,显示不能构成三角形的信息。

【指导】

1. 判断能否构成三角形的数学依据是看是否满足两边之和大于第三边;求任意三角形的面积可以用海伦公式。

2. 本题参考程序为:

```
#include <stdio.h>
#include <math.h>
main()
{ float a, b, c, p, s;
  printf("Enter the three lines of a triangle:");
  scanf("%f%f%f", &a, &b, &c);
  if (a + b > c && b + c > a && a + c > b)
  { p = (a + b + c) / 2;
    s = sqrt(p * (p - a) * (p - b) * (p - c));
    printf("The area of the triangle is: %.2f \n", s);
  }
  else
    printf("The lines are not able to form a triangle. \n");
}
```

3. 编制本程序时要注意以下几点:

1) 程序中要用到 C 语言编译系统提供的求平方根函数 sqrt(), 因此要将头文件 “math.h” 包含进来;

2) if 后面的三条语句构成一条复合语句, 要用花括号括住, 否则就会出现语法错误。

【实习 2】 (1 学时)

[题目] 下面的程序用来从键盘输入一个不多于 5 位的整数, 能显示出它是几位数, 并按正反两种顺序显示出各位数字(例如输入整数 345, 应能输出 3, 345, 543)。请把程序调试正确。

```
#include <stdio.h>
main()
{ char c1, c2, c3, c4, c5;
```

```

int n;
long int x;
c1 = c2 = c3 = c4 = c5 = ' ';
scanf("%ld", &x);
if x >= 10000
    n = 5;
else if x >= 1000
    n = 4;
else if x >= 100
    n = 3;
else if x >= 10
    n = 2;
else
    n = 1;
switch (n)
    { case 5: c5 = x % 10 + '0'; x = x/10; break;
      case 4: c4 = x % 10 + '0'; x = x/10; break;
      case 3: c3 = x % 10 + '0'; x = x/10; break;
      case 2: c2 = x % 10 + '0'; x = x/10; break;
      case 1: c1 = x % 10 + '0';
    }
printf("n = %d \n", n);
printf("%c%c%c%c%c \n", c1, c2, c3, c4, c5);
printf("%c%c%c%c%c \n", c5, c4, c3, c2, c1);
}

```

[指导]

1. 程序中用一个多重选择来判断输入的整数是几位数;用 switch 语句对不同位数的数进行求各位数字的处理。

2. C 语言中的 int 型变量能存储的最大正整数为 32767,更大的数就要用 long int 型变量。

3. 程序中包含两类错误:一类是 if 语句中的关系表达式要用括号括起来;第二类是 switch 语句中不能用 break。这是因为,当 n=5 时,第一个 case 语句只求出了最末尾一位数字,其后的各位数字则分别在后续的 case 语句中求出。编程中使用 switch 语句时,通常需要用 break 切断各个 case 之间的连续性,以便分别对不同的情况进行各自的处理。而本题正好利用各个 case 之间的连续性来获得所需要的结果,这是一个特例。

4. 程序中 c1 = x % 10 + '0' 等语句的作用是将 x 除以 10 的余数转化为数字字符。假设 x = 23, 则 x % 23 + '0' 就是数字字符 '3', 而不是整数 3。

5. 本题正确的源程序为:

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

```

main()
{ char c1, c2, c3, c4, c5;
  int n;
  long int x;
  c1 = c2 = c3 = c4 = c5 = ' ';
  scanf("%ld", &x);
  if (x >= 10000)
    n = 5;
  else if (x >= 1000)
    n = 4;
  else if (x >= 100)
    n = 3;
  else if (x >= 10)
    n = 2;
  else
    n = 1;
  switch (n)
  { case 5: c5 = x % 10 + '0'; x = x / 10;
    case 4: c4 = x % 10 + '0'; x = x / 10;
    case 3: c3 = x % 10 + '0'; x = x / 10;
    case 2: c2 = x % 10 + '0'; x = x / 10;
    case 1: c1 = x % 10 + '0';
    }
  printf("n = %d \ n", n);
  printf("%c%c%c%c%c \ n", c1, c2, c3, c4, c5);
  printf("%c%c%c%c%c \ n", c5, c4, c3, c2, c1);
}

```

程序运行时,若从键盘输入 12345,则输出结果为:

```

n = 5
12345
54321

```

【思考题】

1. 用 if-else 语句的嵌套代替实习 2 题目中的 if-else if-else 结构,重新编制实习 2 题目的程序,并获得正确的结果
2. 运行下面的程序,并分析运行结果(本题涉及 switch 语句的嵌套)。

```

#include <stdio.h>
main()
{ int a = 2, b = 7, c = 5;
  switch(a > 0)

```



```

| case 1:switch(b<10)
|   | case 1:printf("@");break;
|   | case 0:printf("!");break;
|   |
| case 0:switch(c == 5)
|   | case 0:printf("*");break;
|   | case 1:printf("#");break;
|   | default:printf("% %");break;
|   |
|   default:printf("&");
|   printf("\ n");
|

```

3. 请编制一个程序, 计算当 $a=1$ 、 $b=2$ 、 $c=3$ 、 $d=4$ 时, 条件表达式 $a < b$? $a:c < d$? $a:d$ 的值是多少?