第 3 章

数据的运算与输入输出

3.1 知识要点

- 运算符的构成及运用。
- 表达式的构成及运算规则。
- 数据的输入与输出。

【知识点讲解】 用运算符和括号将运算对象(常量、变量和函数等)连接起来的、符合 C 语言语法规则的式子,称为表达式。C 语言提供有丰富的运算符,构成多种表达式,主要有算术表达式、赋值表达式、关系表达式、逻辑表达式、条件表达式、逗号表达式。

把数据从计算机内部送到计算机的外设上的操作称为“输出”。相反,从计算机外部设备将数据送入计算机内部的操作称为“输入”。输入、输出是对数据的重要操作。没有输出的程序是没有用的,没有输入的程序是缺乏灵活性的。在程序运行时,由用户临时输入所需的数据,可以提高程序的通用性和灵活性。C 语言本身不像其他高级语言一样有输入和输出语句,其输入和输出是由标准的输入输出函数完成的。在使用系统库函数时,要用预编译命令“#include”将有关的“头文件”包含到用户源程序中。本次实验涉及的知识点关系如图 3-1 所示。

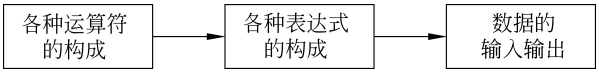

图 3-1 本次实验涉及的知识点关系图

3.2 上机实验

1. 实验目的

- 学会使用 C 语言算术运算符、关系运算符、逻辑运算符、赋值运算符、逗号运算符、条件运算符,掌握 C 语言中表达式的概念,理解相应运算符的运算规则和在运算过程中数据类型的转换规则。
- 掌握赋值语句的使用方法。
- 掌握字符输入输出函数的使用形式及调用方法。
- 掌握格式输入输出函数的使用形式及调用方法,能够正确使用常用格式控制符。

2. 实验参考

【例 3-1】　股票发行价格计算中，可以使用不同的方法进行定价，其中一种方法为分析法：股票价格＝((资产总值－负债总值)/投入资本总额)×每股面值。

请编写一个程序，给定资产总值、总负债值、投入资本总额和每股面值，计算股票价格(精确到小数点后 2 位)。

【源程序】

```
#include <stdio.h>
int main()
{
    double assets, debt, invest, value, pricing;
    printf("请输入上市公司资产总值,负债总值,投入资本总额,每股面值(用空格分开):\n");
    scanf("%lf%lf%lf%lf",&assets,&debt,&invest,&value);
    pricing=(assets-debt)/invest*value;
    printf("上市公司资产总值 assets=%.2e\n 负债总值 debt=%.2e\n 投入资本总额
            invest=%.2e\n 每股面值
        value=%.2e\n",assets,debt,invest,value);
    printf("计算得到股票价格为 pricing=%.2f\n",pricing);
    return 0;
}
```

【程序输入】

```
请输入上市公司资产总值,负债总值,投入资本总额,每股面值(用空格分开):
500000000 90000000 100000000 4<CR>
```

【程序输出】

```
上市公司资产总值 assets=5.00e+008
负债总值 debt=9.00e+007
投入资本总额 invest=1.00e+008
每股面值 value=4.00e+000
计算得到股票价格为 pricing=16.40
```

【程序分析】　根据股票价格计算公式：股票价格＝((资产总值－负债总值)/投入资本总额)×每股面值，计算股票定价。由于结果需要精确到小数点后 2 位，因此变量都用 double 数据类型来存储，而且资产数值较大，所以用%e 来控制输出。%.2e 表示使用科学记数法，实数后面保留 2 位小数位。还有其他修饰符，例如："%5f"就是要保证结果的最小宽度为 5，"%-f"表示输出字符靠左。

【例 3-2】　编写程序判断年数是闰年还是平年，闰年输出为 1，平年输出为 0。闰年判断公式：((year%4==0)&&(year%100!=0))‖(year%400==0)。

【源程序】

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

```
{
    unsigned int year=0;
    int res=0;
    printf("请输入年份: ");
    scanf("%d",&year);
    res=((year%4==0)&&(year%100!=0))‖(year%400==0);
    printf("res=1 是闰年\nres=0 是平年\n");
    printf("结果是:\nres=%d\n",res);
    return 0;
}
```

【程序输入】

请输入年份：1975<CR>

【程序输出】

```
res=1 是闰年
res=0 是平年
结果是:
res=0
```

【程序分析】 由于年份不可能出现负值,因而年份 year 定义为 unsigned int 类型。程序中使用了算术运算符、关系运算符和逻辑运算符相结合来判断是闰年还是平年。优先级:算术运算符高于关系运算符高于逻辑运算符(除了!以外),在逻辑运算中还有可能发生“短路现象”,例如,输入“2000”2000％4＝＝0 为“真”,且 2000％100＝＝0 为“真”后面是或运算“‖”,所以 year％400 不用再做判断,就可知道该表达式为真,这就是一种“短路”现象。

【例 3-3】 请编写一个程序,使用 getchar()和 putchar(c)函数实现字符数据的输入、输出。

【源程序】

```
#include <stdio.h>
int main()
{
    char a,b,c;
    a=getchar();
    b=getchar();
    c=getchar();
    putchar(a); putchar(b); putchar(c);
    putchar('\n');
    putchar('A'); putchar('\101'); putchar(65);
    return 0;
}
```

【程序输入】

BOY<CR>

【程序输出】

```
BOY
AAA
```

【程序分析】 getchar 函数的作用是从标准输入设备获得一个字符，其一般形式为 getchar()。在输入过程中，getchar()也会将空格、回车符当作一般字符输入。例如，上面例题中，如果输入内容为"B＜空格＞O＜空格＞Y"，则会分别将字符'B'＜空格＞'O'传递给变量 a，b，c，输出的结果则为"B＜空格＞O"。putchar 函数的作用是向标准输出设备（通常是显示器或打印机）输出一个字符。其一般形式为 putchar(c)，输出字符变量 c 的值，c 可以是字符型变量、整型变量、转义字符。在使用该函数时，应在程序前使用预编译命令：#include ＜stdio.h＞。

3. 实验内容

编写程序并上机调试运行。

(1) 设有变量定义如下：

```
int i=6, j=12;
double x=3.28, y=90;
```

希望得到如下输出结果：

```
i=6    j=c
x=3.280000E+000    y=90
```

请编程实现。

(2) 某种物品每年的折旧费的线性计算方法如下：

折旧费＝(购买价格－废品价值)/使用年限

请编写一个程序，当输入某物品的购买价格、使用年限和废品价值时，程序能计算出其在某一年折旧后的价值(结果保留两位小数)。

(3) 编写程序实现以下功能。计算在贷款第一个月、第二个月及第三个月后需要还款的金额。

贷款金额：20 000.00

年贷款利率：6.0 %

每个月还款金额：386.66

第一个月剩余的需还款金额：19 713.34

第二个月剩余的需还款金额：19 425.25

第三个月剩余的需还款金额：19 135.71

说明：所有数额有效位数保持在小数点后 2 位。

提示：每个月，剩余的贷款金额为总数减去每个月的还款金额，但是每个月剩余的贷款金额要加上按照月贷款利率计算出来的利息。月贷款利率为年贷款利率除以 12。

(4) 编写程序实现以下功能，使用平均分摊法计算融资租赁租金。

每次支付的租金＝((租赁设备购置成本－预计残值)＋租赁期间利息＋租赁期间手续

费)/租金支付次数。

例如：某企业于2016年1月1日从租赁公司租一设备价值100 000元，租期为5年，预计租赁期满残值6000元，归租赁公司，年利率9%，手续费是设备价格的2%（一次性收取）。租金一年付一次，则每次需要支付的租金为：((100 000－6000)＋(100 000×(1＋9%)^5－100 000)＋100 000×2%)/5＝29 972(元)。

3.3 实验过程中的常见问题与解决方法

【问题3-1】 提示信息中出现scanf或者printf函数未定义的错误提示。

造成这类问题的原因往往是忘记写文件包含命令＃include ＜stdio.h＞，或者包含写法出错。例如：

```
#include <stdio.h>;                    //添加了多余的";"号
#include <stdio.h>, <math.h>           //不正确的书写方式
#include <stdio.h, math>               //不正确的书写方式
```

解决方法：编译预处理命令以＃开头，末尾不加分号；一个include命令只能指定一个被包含文件。如果需要包含多个文件，则需用多个include命令，且一个命令应单独占一行。

【问题3-2】 未识别的字符类错误提示。

造成该类错误的原因有多种，根本原因是使用了C语言不接受的字符。例如：

(1) 程序中混淆使用中文字符和英文字符。

C语言在语法上不支持中文字符，所有的关键字和语法结构都需要使用英文字符。如果在字符串和注释以外的地方使用全角字符，特别是全角的标点符号，如逗号、分号等，与半角符号外观很像，不容易区分，会产生此类错误。

例如：

```
int a,b;
a=5;                                   //这里使用了全角分号
b=8;
```

(2) 定义标识符的时候使用非法的字符。

C语言的变量等命名遵守一定的规范，并不是所有的名称都可以被接受。例如：f(a)、a－b、2x都是非法的。错误的命名也会产生此类错误。

(3) 错误地使用函数名称。

例如：

```
print("Hello World");                  //print少输入一个"f"字符,会产生此类错误
```

(4) 混淆正、反斜杠。

在C语言中，正、反斜杠有不同的含义，错误地使用也会产生此类错误。

例如：

```
x=5\2;                                 //写错除号
printf("/n");                          //写错换行符
```

(5) 没有定义变量就直接使用。

在 C 语言中,变量必须先定义然后再使用,否则会产生此类错误。

例如:

```
#include <stdio.h>
int main()
{
    y=x+10;                                    //y,x 都没有定义,会产生错误
    return 0;
}
```

解决方法:在错误提示处,再重新输入一遍,就可以解决全角半角的问题,以及脱字的错误。但是命名错误、没有定义变量属于使用和字符理解错误,仔细地阅读错误提示信息,不难找到解决的方案。

【问题 3-3】 括号使用不当的错误。

C 语言中,应成对出现的符号必须配对,如:{ }、[ ]、( )、' '、" "。如果写程序时发生了不匹配的情况,就会给出这类提示。例如,大括号配对圆括号、尖括号配对大括号都会发生此类错误。

解决方法同样是仔细阅读编译器的错误提示,很快可以找到问题。

【问题 3-4】 忽略了变量的类型,进行了不合法的运算。

例如:

```
int main()
{
    float a=4.0, b=3.0;
    printf("%d", a%b);
}
```

因为“%”是取余运算,只有整型变量才可以进行取余运算,而实型变量是不允许进行“取余”运算的,所以编译时会出现错误。

解决方法:进行运算时一定要注意变量的数据类型。

【问题 3-5】 输入数据时,容易出现以下几种错误:

(1) 变量前忘记写地址符“&”。

例如:

```
scanf("%d%d", x, y);
```

虽然在编译时系统没有报告错误,但是在程序运行时会出错。

解决方法:输入数据时,变量前必须要有地址符“&”。

(2) 从键盘输入数据时,输入的格式与 scanf 中格式字符串中的格式不一致。

例如:

```
scanf("%d, %d", &x, &y);
输入数据:12<空格>36<CR>
```

输入时两个整数之间用空格分开,但是 scanf 格式字符串中的两个%d 是用逗号分开

的,所以只有x能得到数据12,而y还是随机数。

正确的输入方式是必须用逗号分开输入的两个整数,即输入12,36<CR>。

类似地,如果输入函数是scanf("x=%d, y=%d", &x, &y);则正确的输入方式是x=12,y=36<CR>。

解决方法:输入数据的格式必须与scanf中格式字符串的格式保持完全一致。

(3) 输入字符型数据时用空格、回车符来分隔两个字符数据。

例如:

```
scanf("%c%c", &ch1, &ch2);
printf("ch1=%c,ch2=%c.\n",ch1,ch2);
输入数据:A<空格>B<CR>
输出结果:ch1=A,ch2=  .  (实际上 ch2=后面是输出了一个空格)
```

因在格式字符串中两个%c之间没有空格,那么在输入数据时两个字符'A'和'B'之间也不能有空格。如果格式字符串写成"%c %c",则在输入时'A'和'B'之间可以加空格。若写成"%c,%c",则输入应为A,B<CR>。

解决方法:输入字符型数据时要注意不能用空格、回车符等来分隔两个字符数据,因为它们本身也是合法的字符。

【问题3-6】 使用格式化输入、输出函数时,格式字符与变量的数据类型不匹配。

例如以下4个程序段:

```
(1)  int x=3;
     printf("x=%f\n", x);          //输出x时,格式字符用'f'是错的
     输出结果:x=0.000000           //输出结果类型不匹配
(2)  int x;
     scanf("%f", &x);              //输入x时,格式字符用'f'是错的
     printf("x=%d\n", x);
     输入数据:3<CR>
     输出结果:x=1077936128         //输出结果不对
(3)  float x=5.6;
     printf("x=%d\n", x);          //输出x时,格式字符用'd'是错的
     输出结果:x=1610612736         //输出结果不对
(4)  float x;
     scanf("%d", &x);              //输入x时,格式字符用'd'是错的
     printf("x=%f\n", x);
     输入数据:5.6<CR>
     输出结果:x=0.000000           //输出结果不对
```

解决方法:格式字符一定要与变量的数据类型相匹配。

特别注意:long int型和double型数据输入、输出时应在对应的格式字符'd'或'f'前加字母'l'。