实验 3　顺序结构程序设计

3.1　实 验 目 的

(1) 熟悉赋值运算符的使用，能根据需要构建相应的赋值表达式，掌握两变量交换数据的方法。

(2) 继续熟悉整数相除、取余运算及数据类型转换等内容，能实现四舍五入保留指定位小数的算法。

(3) 熟悉常用数学函数的使用。

(4) 通过样例加深对 printf()常用格式控制符功能的理解，掌握 printf()的使用。

(5) 掌握 scanf()的使用，能正确输入数据。

(6) 掌握顺序结构程序设计的方法，能够画出传统的流程图和 N-S 流程图。

3.2　知 识 要 点

1. 赋值运算符和表达式

编写程序时，给变量赋值是基本操作，必须熟练掌握。

1) 赋值运算符与表达式

赋值运算符为＝。

表达式格式为“变量＝表达式”。

功能：先计算表达式的值，再赋给左边的变量，即把表达式的值存入左边变量所标识的存储单元中。

说明：

(1) ＝是“赋值”的含义，不是数学中的“等于”。例如，n＝n＋1 是将变量 n 存储单元的值加 1 后存回到该单元，这种写法经常使用，应掌握。

(2) 左边必须是左值(即能存放数据的单元)，通常为变量，不能是常量，a＋b＝c 是错误的。表达式 x＝y 执行后，改变的是 x 值，而 y 值保持不变。

(3) 赋值号＝两边的数据类型要求相同，若不同，则在赋值前自动把右边表达式的值转换为与左边类型相同的值，再赋给左边变量。

优先级：只高于逗号运算符，比其他运算符级别都低。

结合性：从右到左。

赋值表达式除了给左边变量赋值外，表达式本身也有值，其值为左边变量的值，也就是说：表达式 x＝y＝z＝0 是允许的，相当于 x＝(y＝(z＝0))，即先给 z 赋 0，再赋表达式 z＝0 的值(也是 0)给 y，这样 z、y 的值都是 0；接着再将 y 值赋给 x，因此，x 值也为 0，相当于执行了：z＝0；y＝0；x＝0；三条语句。

注意：变量初始化与赋值语句的区别，为什么语句“int a＝b＝c＝5;”是错误的？

2) 复合赋值运算符与表达式

运算符：＋＝、－＝、＊＝、/＝、％＝等。

功能：把右边表达式的值同左边变量的值进行相应运算后，再把运算结果赋给左边的变量。该复合赋值表达式的值也就是左边变量的值。

例如，“x＋＝y;”相当于“x＝x＋y;”，“x－＝y;”相当于“x＝x－y;”。

优点：简洁(可读性也不差)，编译速度快。

2. 逗号运算符与表达式

若把多个表达式用逗号(,)分开，就构成了逗号表达式。

格式：

```
表达式 1,表达式 2,…,表达式 n
```

运算过程：从左到右依次计算各个分表达式的值，整个逗号表达式的类型和值就是最右端表达式的类型和值。

优先级：最后一级，低于所有运算符。

例如：

```
x＝(3＋4,5.6＜2,4&&1,3.2－0.6);
```

该表达式的值就是最后表达式 3.2－0.6 的值，即 2.6，数据类型为 double。

3. 使用数学函数构建表达式

对于一些常用的数学运算，系统已设计好专门的函数来实现相应功能，只要加语句“＃include 头文件”，就可以在程序中直接调用数学函数来构建表达式。在头文件 math.h 中包含如表 3-1 所示的函数声明。

表 3-1 常用数学函数

函数名称	函数原型	数学式子	功能
实数绝对值函数	double fabs(double x)	|x|	返回实数 x 的绝对值
正弦函数	double sin(double x)	sin$x$(x 为弧度)	返回弧度为 x 的正弦值
余弦函数	double cos(double x)	cos$x$(x 为弧度)	返回弧度为 x 的余弦值
正切函数	double tan(double x)	tg$x$(x 为弧度)	返回弧度为 x 的正切值
平方根函数	double sqrt(double x)	$\sqrt{x}$(x＞＝0)	返回 x 的算术平方根
指数函数	double exp(double x)	e^x(e＝2.718 282)	返回 e^x 的值
幂函数	double pow(double x,double y)	x^y	返回 x^y 的值
自然对数函数	double log(double x)	ln x(x＞0)	返回以 e 为底 x 的对数
对数函数	double log10(double x)	log10x(x＞0)	返回以 10 为底 x 的对数

4. 程序流程图

编写一个程序，最重要的是确定其算法（即解决问题的方法和步骤）。可用程序流程图来表示算法，常用的流程图如下。

1）传统的流程图

用一些图框（如图 3-1 所示）来表示各种操作。特点是形象直观，易于理解。

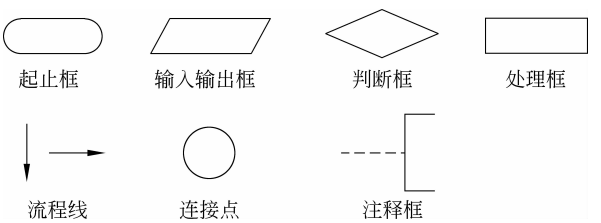

图 3-1　流程图的常见图框

图 3-2 是一个传统流程图的例子，其目的是判断一个大于或等于 3 的正整数是否为“素数”。

2）N-S 流程图

对传统流程图进行了改进，去除了箭头，全部算法写在一个矩形框内，由从属于它的多个子框构成，又称为盒图。图 3-3 是 N-S 的例子，功能与图 3-2 相同，但更加简单，明了。

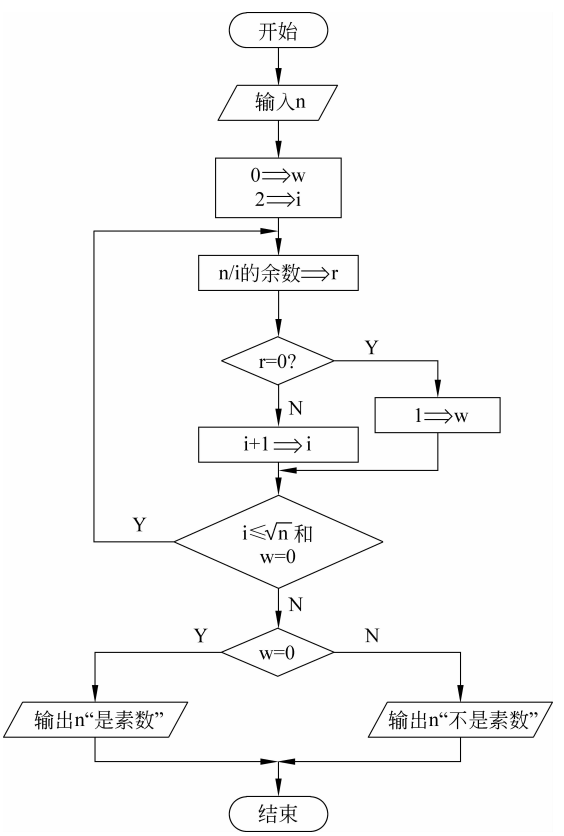

图 3-2　传统流程图的例子

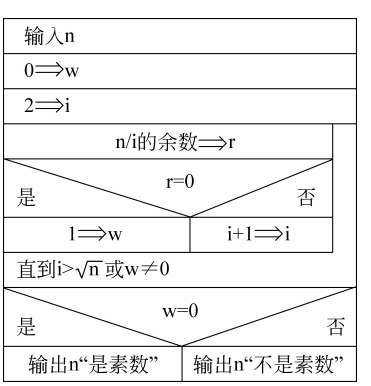

图 3-3　N-S 流程图的例子

3）程序的三种基本结构

顺序结构：从上到下顺序执行，既不重复也不跳过语句的执行。

选择结构：根据条件，选择某一模块的语句执行。

循环结构：根据条件，重复执行某一模块语句，重复的次数根据条件决定。

5. 数据的输入输出

输入输出是C语言程序中最基本的操作之一，几乎每一个C程序都包含输入输出功能。但是，C语言本身不提供输入输出语句，输入和输出操作是由C标准函数库中的函数来实现的。常用的输出函数有printf()、putchar()等，输入函数有scanf()、getchar()等。

要使用这些输入输出函数，需要在程序文件的开头加上预编译指令#include<stdio.h>。

1）使用printf()输出数据

格式：

```
printf(格式控制符,输出项表)
```

例如：

```
printf("i=%d,c=%c\n",i,c);
```

说明：

(1) 格式控制符的形式为：

```
%[对齐方式][宽度][.精度]格式描述符
```

这里的方括号表示该项可省略，百分号和格式描述符是必需的。常用的格式描述符如表3-2所示。

表3-2 常用的格式描述符

格式描述符	功　能	格式描述符	功　能
d	输出带符号的十进制数	f或lf	以小数形式输出实数，默认带6位小数
c	以字符形式输出，只输出一个字符	s	输出字符串
u或U	输出无符号的整数	o或O	输出八进制整数
x或X	输出十六进制整数	e或E	以指数形式输出实数

对齐方式为一时表示左对齐，省略则为右对齐。宽度是指数据输出的最小宽度，当数据不足指定宽度时，通常用空格补齐。当输出小数时，精度是指小数位数；若是字符串，则是指截取字符串的长度。

(2) 输出项表：可以是常量、变量或表达式列表(用逗号分开)，是输出的具体内容。

2）使用scanf()输入数据

格式：

```
scantf(格式控制符,地址列表)
```

例如：

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

说明：

(1) 格式控制符与 printf()的要求类似，加小写字母 l 用于输入长整型(如%ld)或 double 类型(如%lf)。还可以在其中插入一些附加字符。

例如：

```
scanf("a=%d,b=%d,c=%d",&a,&b,&c);
```

在输入数据时，也需要采用同样格式来输入，如输入“a＝10，b＝50，c＝100”并按 Enter 键。

(2) 地址列表：可以是变量的地址，或字符串的首地址。

(3) 常见错误：

① 不是地址列表：如“scanf("%f%f%f",a,b,c);”。

② 输入内容与插入的附加字符不对应：如“scanf("a＝%f,b＝%f,c＝%f",&a,&b,&c);”。

输入 1 3 2↙ 或 a＝1 b＝3 c＝2↙

③ 输入字符型数据时出现错位现象：如“scanf("%c%c%c",&c1,&c2,&c3);”。

输入 a b c↙ (将空格作为有效字符输入)

④ 试图用格式控制符、转义字符等来控制输入格式：如“scanf("%5.2f\n",&a);”。

3) 字符数据的输入输出

常用的输入输出语句有“putchar(字符型数据或整数)”，输出一个字符；[变量＝]getchar()，输入一个字符，通常存放到某一存储单元中。

3.3　实验内容与步骤

(1) (基础题)编写程序，将 10 000 秒转换成以“xx 时 xx 分 xx 秒”格式输出。

提示： 可考虑整数的/、%运算。

(2) (基础题)编程实现：先定义两个整数变量，然后输入两个值，再交换这两个变量的值，最后输出交换后的新值，如图 3-4 所示。

```
inter a,b:10,30
交换前, a=10, b=30
交换后, a=30, b=10
```

图 3-4　程序运行结果

问题：

① 语句组：a＝b；b＝a；能交换 a、b 的值吗？

② 若不能，如何改进？

③ 画出程序的传统流程图。

(3) (基础题)运行下列程序，体会 printf()中“格式控制符”的用法，并回答相关问题。

```
#include<stdio.h>
int main( )
{
    int k=1234;
    double f=12345.0123456789;
    char * p="China";
```

```
    printf ("%%d格式符: \n");
    printf ("%d\n",k);
    printf ("%6d\n",k);
    printf ("%06d\n",k);
    printf ("%2d\n\n",k);

    printf ("%%f格式符: \n");
    printf ("%f\n",f);
    printf ("%lf\n",f);
    printf ("%15f\n",f);
    printf ("%15.4f\n",f);
    printf ("%-15f\n",f);
    printf ("%-15.4f\n",f);
    printf ("%.2f\n",f);
    printf ("%30.20f\n\n",f);

    printf ("%%e格式符: \n");
    printf ("%e\n",f);
    printf ("%15e\n",f);
    printf ("%15.4e\n",f);
    printf ("%-15e\n",f);
    printf ("%-15.4e\n",f);
    printf ("%.2e\n",f);
    printf ("%30.20e\n\n",f);

    printf ("%%s格式符: \n");
    printf ("%s\n",p);
    printf ("%10s\n",p);
    printf ("%-10s\n\n",p);

    return 0;
}
```

问题：请说明格式控制符%d、%f、%e、%s的基本用法。

(4)(基础题)分析、运行下列程序,要让各变量得到对应的值a=3,b=7,x=8.5,y=71.82,c1='A',c2='a'。请问在键盘上该如何输入？然后回答相关问题。

```
#include<stdio.h>
int main( )
{
    int a,b;
    float x,y;
    char c1,c2;
    scanf("a=%d,b=%d",&a,&b);
    scanf("%f%e",&x,&y);
```

```
    scanf("%c%c",&c1,&c2);

    printf("a=%d,b=%d\n",a,b);
    printf("x=%f,y=%f\n",x,y);
    printf("c1=%c,c2=%c\n\n",c1,c2);
    return 0;
}
```

问题：

① scanf()函数应如何书写？

② 输入不同类型数据时，应注意什么？

(5)(提高题)以下程序实现的功能是：输入一个 double 类型的数据，使该数保留两位小数，对第三位小数进行四舍五入处理，然后输出此数，查验处理是否正确。请根据注释和运行结果截图填充程序。

```
#include<stdio.h>
int main()
{
    double x;
    printf("Enter x=");
    scanf("%lf",&x);

    printf("(1) x=%f.....原始数据\n",x);
    printf("(2) x=%.2f.....格式控制数据\n",x);

    x=_____;                    //x扩大 100 倍
    x=_____;                    //x增加 0.5
    x=_____;                    //对 x 取整后再赋值给 x
    x=_____;                    //x缩小 100 倍
    printf("(3) x=%f.....处理后数据\n",x);

    return 0;
}
```

程序运行结果如图 3-5 所示。

图 3-5　程序运行结果截图

(6)(提高题)改错题：以下程序有多处错误，若按下列截图所示格式输入输出数据，请在程序的相应位置上改正错误。

程序代码如下：

```
main
{
    double a,b,c,s,v;
    printf(input a,b,c :\n);
```

```
    scanf("%d %d %d",a,b,c);
    s=a*b;                                  //计算长方形面积
    v=a*b*c;                                //计算长方体积
    printf("%d %d %d",a,b,c);
    printf("s=%f\n",s,"v=%d\n",v);
}
```

实验 4　选择结构程序设计

4.1　实 验 目 的

(1) 熟悉 C 语言逻辑值“真”、“假”的表示。

(2) 掌握关系运算符的种类、运算优先级、运算结果的类型，清楚一个关系表达式对应的相反式。

(3) 掌握逻辑运算符的种类、运算优先级、结合性、操作数与运算结果的类型，能根据要求构建相应的逻辑表达式，理解逻辑运算中“短路”现象。

(4) 掌握 if 语句的用法，能够根据要求熟练使用单分支、双分支、多分支(嵌套)结构。

(5) 掌握 switch 语句的格式、功能及注意事项。

(6) 熟悉条件运算符和条件表达式的基本用法。

4.2　知 识 要 点

在生活中有很多需要判断和选择的情景。例如，天冷了，就要多穿衣服；若生病了，就要吃药或者去看病……在利用计算机求解问题的过程中，也需要根据条件的不同，做出相应选择。

1. C 语言的逻辑值

在计算机高级语言中，条件判断的结果是用逻辑值来表示的。

逻辑值只有两个：分别用“真”、“假”表示。早期的 C 语言没有专门的逻辑值，对用户提供的逻辑值，计算机将所有非 0 值视为“真”，0 值表示“假”；当由计算机输出逻辑值时，“真”用 1 表示，“假”用 0 表示。后来的 C99 有了表示逻辑值的 bool 型，分别用 true、false 表示“真”、“假”，其内部仍是分别用 1 和 0 表示。

2. 关系运算

关系运算用来比较两个操作数的大小，其结果为逻辑值(“真”或“假”)。进行关系运算的关键是构造合适的“关系表达式”。

运算符有 6 个，分别是＜(小于)、＜＝(小于等于)、＞(大于)、＞＝(大于等于)、＝＝(等于)、！＝(不等于)。

关系表达式：由一个关系运算符连接前后两个表达式而构成的式子，其格式如下。

操作数1 关系运算符 操作数2

例如，10>5是关系表达式，其结果为“真”(即1)，而10<5结果为“假”(即0)。

说明：

(1) 关系运算的操作数可以是各种基本数据类型(如int、char、double、float等)。

(2) 比较大小时，数值型以其大小、英文字母以其ASCII码、汉字以它的机内码为准进行比较，如'a'>'A'的值为“真”(即为1)。

(3) 由于计算机精度有限，通常浮点数不进行相等(==)判断。

优先级：在无括号的情况下，<、<=、>、>=运算符的级别要高于==、!=；算术运算符的优先级要高于关系运算符，关系运算符的优先级要高于赋值运算符。例如，'a'-3>10的结果为1，计算过程是：先得到'a'的ASCII码(97)，再减去3得到算术表达式的值(94)，最后再与10比较大小。

结合性：是指当表达式中出现两个或两个以上相同优先级的操作符的情况下，用于指定运算是从左向右结合还是从右向左结合。关系运算的结合性是从左到右。

运算与反运算：在6个关系运算符中，可以分为<与>=、>与<=、==与!=三组，在两个操作数保持不变的情况下，每组中的两个运算符构成的运算互为反运算，当某一种关系表达式的运算结果为“真”(即为1)时，它的反运算结果必然为“假”(即为0)，反之也成立。也就是说，对于同一问题的求解，可以从两个相反角度去考虑，这体现了程序设计方法的多样性、复杂性。

3. 逻辑运算

如果要构造更为复杂的表达式，就可以使用逻辑表达式。

逻辑表达式：将逻辑型数据与逻辑运算符连接而成的式子。

逻辑运算符有3个，分别是：

1) !(逻辑非，单目运算，即只有一个操作数)

运算规则：当操作数为“真”(即非0)时，运算结果为“假”(即0)；当操作数为“假”(即0)时，运算结果为“真”(即1)。

2) &&(逻辑与，双目运算，有两个操作数)

运算规则：只有两个操作数同时为“真”(非0)时，运算结果才为“真”(即为1)，否则均为“假”(即为0)。也就是说，只要两个操作数中有一个为“假”(即为0)时，运算结果就一定为“假”(即为0)。

3) ||(逻辑或，双目运算，有两个操作数)

运算规则：只有两个操作数同时为“假”(即0)时，运算结果才为“假”(即0)，否则均为“真”(即1)，也就是说，只要两个操作数中有一个为“真”(非0)时，运算结果就一定为“真”(即为1)。

a	b	!a	a&&b	a||b
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

图4-1 逻辑运算符的运算规则

三种运算符的运算规则如图4-1所示。

结合性：从左到右。

优先级：下列运算符中优先级从高到低依次是：! 运算符、算术运算符(*、/、%、+、-)、关系

运算符(>、>=、<、<=、==、!=)、&& 运算符、‖运算符、赋值运算符(=、+=、-=、*=、/=、≪=、≫=等)。

问题：数学式 0<x<5 该如何书写呢？答案：x>0 && x<5。为什么？

设 a、b 是逻辑型变量，则下列三个式子成立：

!!a==a、!(a&&b)==!a‖!b、!(a‖b)==!a&&!b

构造逻辑表达式举例如下。

【例 4-1】 字符 ch 为英文字母或数字字符。

分析：英文字母有大小写之分，可利用其 ASCII 码大小来构造表达式；大写字母、小写字母、数字字符三者之间是“或”关系。

逻辑表达式：

(ch>='A'&&ch<='Z')||(ch>='a'&&ch<='z')||(ch>='0'&&ch<='9')

【例 4-2】 year 表示年份，写出判定是否为“闰年”的表达式。

分析：根据历法知识，如果年份 year 满足下列条件之一即为闰年。

① 能被 4 整除但不能被 100 整除。

② 能被 400 整除。

逻辑表达式：

(year%4==0)&&(year%100!=0)||(year%400==0)

思考题：设学生有五门课程，分别用 sc1、sc2、sc3、sc4、sc5 表示成绩，请问怎样构造满足下列不同要求的表达式。

① 平均成绩 80 分及 80 分以上。

② 每门成绩均在 80 分及 80 分以上。

③ 没有一门课程成绩在 80 分或 80 分以上。

逻辑表达式的“短路”现象：

(1) 在形如□&& □&& □&&…的表达式中，只要前面有一个表达式(用符号□表示)的值为“假”，则整个表达式的值就为“假”，此后各表达式不再计算，因为它们的值无论是“真”还是“假”，都不会影响整个表达式的运算结果，这称为 && 运算符的“短路”现象。

例如，a=0，b=1，计算表达式 a++&& b++的值。

计算过程：先取 a 值 0，a 再增加 1，因为是 && 运算，整个表达式值为 0，表达式 b++不再计算，b 值保持不变(即 b 仍是 1)。

(2) 在形如□|| □|| □||…的表达式中，只要前面有一个表达式(用符号□表示)的值为“真”，则整个表达式的值也就为“真”，后面各表达式的值也不必再计算，理由也是后续表达式的值不会影响整个表达式的运算结果，这称为||运算符的“短路”现象。

例如，a=1，b=1，计算表达式 a++ || b++的值。

计算过程：先取 a 值 1，a 再增加 1，因为是||运算，整个表达式值为 1，表达式 b++不再计算，b 值保持不变(即 b 仍是 1)。

提示：在计算逻辑表达式值时，应考虑到上述两种“短路”现象的存在，以免造成麻烦。

4. if语句

1）基本形式

```
if (表达式)
    语句1
else
    语句2
```

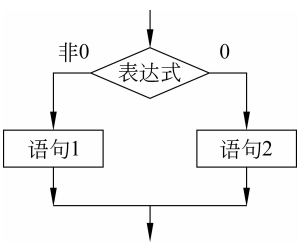

图 4-2　if语句执行过程

说明：

（1）此处的表达式可以是算术表达式、关系表达式、逻辑表达式、赋值表达式、字符表达式等，其值为“真”（即非0）或“假”（即0）。

（2）表达式外的括号（ ）一定要有，不能省略。

（3）语句1、语句2后应加分号（;），如果有多条语句，就可用大括号{ }括住，构成复合语句。

if语句执行过程如图4-2所示。

【例4-3】 输入x值，输出分段函数 $y=\begin{cases} x-1 & (x<0) \\ x+5 & (x\geqslant 0) \end{cases}$ 的值。

编程思路：x值以0为界，采用if…else…语句分成两种情况，很容易实现。

程序代码如下：

```
#include<stdio.h>
int main()
{
    double x,y;
    printf("请输入x的值:");
    scanf("%lf",&x);
    if (x<0)
    {
        y=x-1;
    }   else
    {
        y=x+5;
    }
    printf("x=%f,y=%f\n\n",x,y);
    return 0;
}
```

```
请输入x的值:10
x=10.000000, y=15.000000
```

图 4-3　例4-3程序运行结果

程序运行结果如图4-3所示。

2）单分支if语句

如果if语句省略else部分，就形成了单分支语句，格式为：

```
if  (表达式)
    语句
```

执行过程如图 4-4 所示。

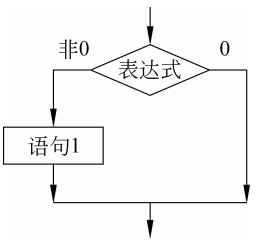

图 4-4　单分支 if 语句执行过程

【例 4-4】 输入 3 个数 a、b、c，要求按由小到大的顺序输出。

编程思路：使用单分支 if 语句进行两两比较，若不满足顺序就交换这两个变量的值，即 a、b 先进行比较，较小值放入 a，较大值放入 b；再将 a、c 进行比较，较小值放入 a，较大值放入 c，这样 a 存放的就是最小值；接着将 b、c 进行比较，较小值放入 b，较大值放入 c。最后，依次输出 a、b、c 即可。

程序代码如下：

```
#include<stdio.h>
int main()
{
    float a,b,c,t;
    printf("请输入三个实数:");
    scanf("%f%f%f",&a,&b,&c);
    if(a>b)
    { t=a;a=b;b=t;}
    if(a> c)
    {t=a;a=c;c=t;}
    if(b> c)
    {t=b;b=c;c=t;}
    printf("从小到大:%f,%f,%f\n",a,b,c);
    return 0;
}
```

程序运行结果如图 4-5 所示。

```
请输入三个实数:1.2 -8.95 0.5
从小到大:-8.950000, 0.500000, 1.200000
```

图 4-5　例 4-4 程序运行结果

说明：两个变量（如 a、b）要交换所存数据，直接使用语句“a＝b；b＝a；”不能实现。需要借用第三个变量（如 t）来存放中间值，正确的代码为“t＝a；a＝b；b＝t；”请注意赋值的先后顺序。

思考题：生活中两个杯子分别盛满水和油，如何进行交换呢？

3）if 语句的嵌套

if…else…语句实现两分支功能，如果再在其中的 if 部分或 else 部分嵌入 if 语句能构成多分支语句。由于嵌入的位置不同，嵌入的 if 语句既可以包含 else 部分，也可以省略 else 部分，从而构成比较复杂的嵌套关系，如图 4-6～图 4-12 所示。

说明：

（1）C 语言规定：else 子句总是与前面最接近的不带 else 子句的 if 语句相结合，与书写格式有无缩进毫无关系；若要改变这种结合关系，可以加括号。

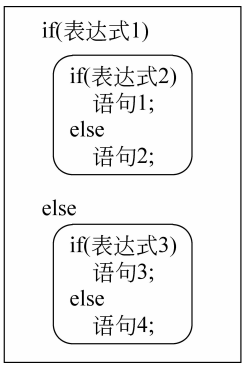

图 4-6　在 if、else 子句分别嵌入 if…else… 语句（4 分支）

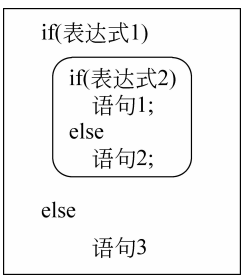

图 4-7　只在 if 子句嵌入 if…else… 语句（3 分支）

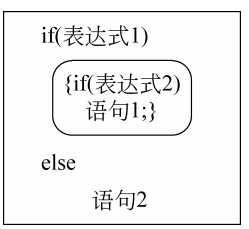

图 4-8　只在 if 子句嵌入单分支 if 语句（有括号）（2 分支）

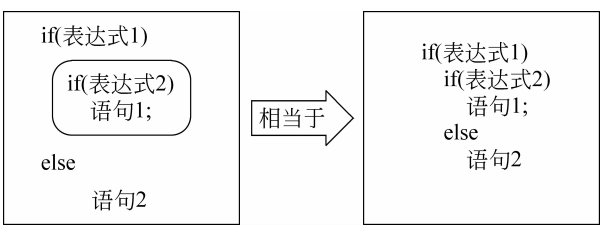

图 4-9　只在 if 子句嵌入单分支 if 语句（无括号）（2 分支）

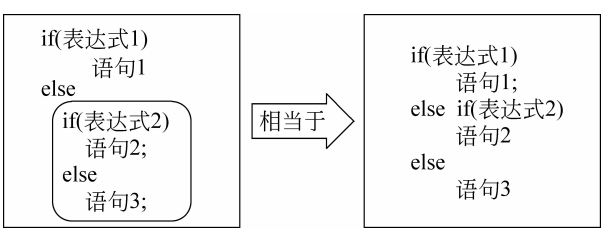

图 4-10　只在 else 子句嵌入双分支 if 语句（3 分支）

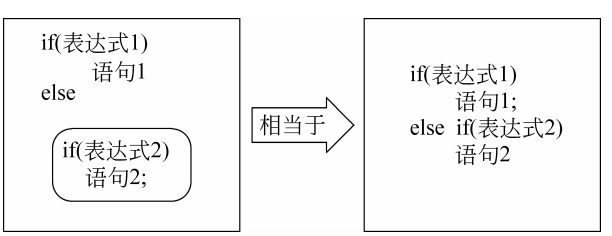

图 4-11　只在 else 子句嵌入单分支 if 语句（2 分支）

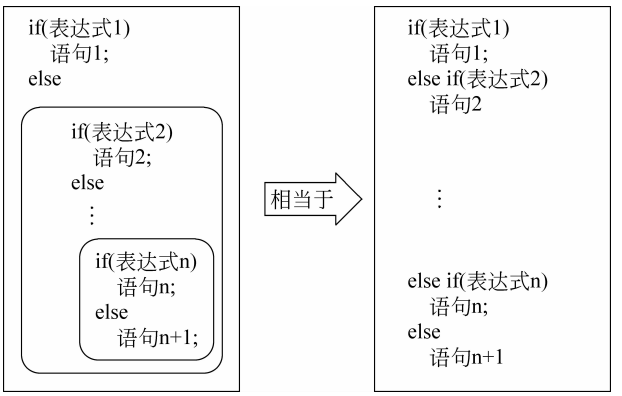

图 4-12　只在 else 子句中嵌套多重 if…else…语句(n＋1 分支)

(2) 如果所有的嵌套部分都放在 else 部分,就能形成多分支结构,这类嵌套结构清晰,可读性强,在编程中大量使用。

【例 4-5】 输入学生成绩,输出相应等级(90～100 为“优秀”, 80～90 为“良好”, 70～80 为“中等”,60～70 为“及格”,60 以下为“不及格”)。

编程思路:先确定各分界点成绩(如 90、80、70 等)属于哪一个区域? 再采用 if…else if …else …实现多分支。

程序代码如下:

```
#include<stdio.h>
int main()
{
    int score;
    printf("请输入成绩:");
    scanf("%d",&score);
    if (score>=90)
        printf("  %d-->优秀\n",score);
    else if (score>=80)
        printf("  %d-->良好\n",score);
    else if(score>=70)
        printf("  %d-->中等\n",score);
    else if(score>=60)
        printf("  %d-->及格\n",score);
    else
        printf("  %d-->不及格\n",score);
    return 0;
}
```

程序运行结果如图 4-13 所示。

```
请输入成绩:85
 85-->良好
```

图 4-13　例 4-5 程序运行结果

5. 条件表达式

从名字上不难判断出其功能,能根据条件是否满足来

决定表达式的值，实现双分支 if 语句类似功能。

条件运算符：?:（唯一的三目运算符，即有三个操作数）。

条件表达式格式："表达式 1? 表达式 2 ：表达式 3"。

执行顺序：先求解表示条件的表达式 1 的值，如果为"真"，则求解表达式 2，此时表达式 2 的值就作为整个条件表达式的值，假若表达式 1 的值为"假"，则求解表达式 3，表达式 3 的值就是整个条件表达式的值。

优先级：高于赋值运算符，低于关系运算符和算术运算符。

结合性：从右到左。

【例 4-6】 求 x 的绝对值。

分析：正数和 0 的绝对值是它本身，负数的绝对值是其相反数。条件表达式为"(x>=0)? x:(-x)"。

【例 4-7】 求两个数 a、b 的较小值。

分析：先比较两个数的大小，取其中的较小值。条件表达式为"(a<=b)? a:b"。

思考题：

① 上述表达式还能写成其他形式吗？

② 与 if…else…语句相比，条件表达式有什么优势？

6. switch 语句

基本形式：

```
switch (表达式)
{
    case 常量表达式 1:
        语句 1
        ⋮
        [break]
    case 常量表达式 2:
        语句 2
        ⋮
        [break]
        ⋮
    [default:
        语句 n+1]
}
```

功能：根据表达式值，选择不同语句进行执行，实现多分支功能，类似于电风扇的多档开关。

执行过程：

(1) 先计算"表达式"的值，假定为 M，若它不是整型，系统只取其整数部分作为结果值。

(2) 依次计算每个常量表达式的值，假定它们的值依次为 M1，M2，…，同样若它们的值不是整型，则自动转换为整型。

（3）让 M 依次同 M1，M2，…进行比较，一旦遇到 M 与某个值 Mi 相等，就从对应标号的语句开始向下执行，如果没有碰到 break 语句，将一直执行到右大括号为止结束整个 switch 语句的执行；若遇到 break 语句，则结束 switch 语句的执行。当 M 与所有值 Mi 都不同，则执行 default 子句的语句，直至结束；若无 default 语句，则什么都不做。执行过程如图 4-14 所示。

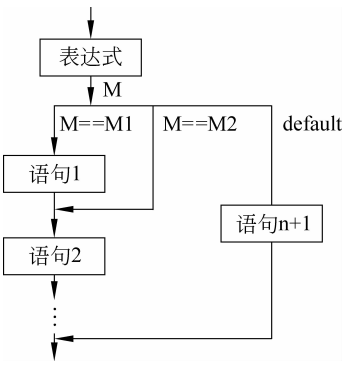

图 4-14　switch 语句执行过程

说明：

（1）表达式通常为整型或字符型，选择合适的表达式是构造 switch 语句的关键。

（2）break 语句应根据情况来设置。

（3）default 子句是可选的。

例 4-5 还可用 switch 语句来改写，关键点是构造表达式、设置各 case 子句的常量值。程序代码如下：

```
#include<stdio.h>
int main()
{
    int score;
    printf("请输入成绩:");
    scanf("%d",&score);
    switch(score/10)
    {
    case 10:
    case 9:
        printf(" %d-->优秀\n",score);break;
    case 8:
        printf(" %d-->良好\n",score);break;
    case 7:
        printf(" %d-->中等\n",score);break;
    case 6:
        printf(" %d-->及格\n",score);break;
    default:
        printf(" %d-->不及格\n",score);
    }
    return 0;
}
```

4.3　实验内容与步骤

（1）（基础题）分析、运行下列程序，验证逻辑值、关系表达式值，并回答相关问题。

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

```
int main()
{
    int x=15,y=5,z=50;
    printf("x=%d,y=%d,z=%d\n",x,y,z);
    printf("x>y? %d\n",x>y);
    printf("x<=y? %d\n",x<=y);
    printf("x+y<z? %d\n",x+y<z);
    printf("z-30>=x+y? %d\n",z-30>=x+y);
    printf("y==z-30>x? %d\n",y==z-30>x);
    return 0;
}
```

问题：

① C语言逻辑值“真”、“假”如何表示？输入、输出时有什么不同？

② 关系运算符包含哪几个？它们的优先级如何？

③ 关系表达式的运算结果是什么？

④ 哪些关系运算符可构成互为相反运算的式子？它们的运算结果有什么关联？

(2)(基础题)分析、运行下列程序，验证逻辑表达式的值，并回答相关问题。

```
#include<stdio.h>
int main()
{
    int a=3,b=4,c=5;
    int x,y,z;
    printf("a=%d,b=%d,c=%d\n",a,b,c);
    printf("a+b>c&&b==c ? %d\n",a+b>c&&b==c);
    printf("!a||!c||b ? %d\n",!a||!c||b);
    printf("a||b+c&&b>c ? %d\n",a||b+c&&b>c);
    printf("a* b&&c+a ? %d\n\n",a* b&&c+a);

    printf("执行 x=a<b||c++后,x=%d,a=%d,b=%d,c=%d\n",x=a<b||c++,a,b,c);
    printf("执行 y=a>b&&c++后,y=%d,a=%d,b=%d,c=%d\n",y=a>b&&c++,a,b,c);
    printf("执行 z=a||b++||c++后,z=%d,a=%d,b=%d,c=%d\n",z=a||b++||c++,a,b,c);
    return 0;
}
```

问题：

① 逻辑运算符包含哪几个？它们的优先级如何？它的运算优先级高于算术运算符、赋值运算符吗？

② 逻辑表达式的操作数、运算结果是什么？

③ 什么是逻辑运算中“短路”现象？这会带来什么影响？

(3)(基础题)以下程序的功能是：输入学生4门课程的成绩，然后根据要求构建相应的逻辑表达式，之后计算这些表达式的值并输出。请根据程序相关提示填写所缺代码，再

运行该程序予以验证，最后回答相关问题。

```
#include<stdio.h>
int main()
{
    double sc1,sc2,sc3,sc4;
    printf("请输入学生的 4 门课程成绩:\n");
    scanf("_________________",_________________);
    printf("sc1=%f,sc2=%f,sc3=%f,sc4=%f\n",sc1,sc2,sc3,sc4);
    printf("4 门课程的平均成绩大于等于 80? %s\n",_____________?"是":"否");
    printf("4 门课程中每门的成绩均大于等于 80? %s\n",_____________?"是":"否");
    printf("4 门课程中至少有一门的成绩大于等于 80? %s\n",_________?"是":"否");
    printf("4 门课程中没有一门的成绩大于等于 80? %s\n",_________?"是":"否");
    printf("4 门课程中至少有两门的成绩大于等于 80? %s\n",_________?"是":"否");
    return 0;
}
```

问题：

① 在程序中的什么位置使用了“条件表达式”？

② 条件运算符有几个操作数？条件表达式如何执行？

(4)(基础题)假设 26 个大写字母和 26 个小写字母分别首尾相连组成两个不同的环，编程实现：输入一个字母，输出它的前一个和后一个字母。例如，输入'A'，输出'Z'和'B'；输入'a'，输出'z'和'b'。

(5)(基础题)编写一个程序：输入一个正整数，先判断其是奇数还是偶数，再进一步判断能否被 3 整除，运行界面如图 4-15 所示。

图 4-15　程序运行结果

提示：

① 判断奇偶性是指能否被 2 整除，可考虑用%运算符取余数，再判断是否为 0。

② 程序运行有 4 种可能结果，可用 if…else…语句嵌套来实现：在外层考虑奇偶性，在内嵌的 if…else…中考虑能否被 3 整除。

(6)(基础题)对于下列函数：

$$y=\begin{cases} x & (-5<x<0) \\ x-1 & (x=0) \\ x+1 & (0<x<10) \end{cases}$$

编写程序，要求输入 x 的值，输出 y 值。

提示：可以选择如下方法之一来编写程序。

① 多个 if 语句(不含 else 部分)。

② 嵌套的 if 语句。

③ if…else if…else 语句。

④ 嵌套的条件表达式。

(7)(提高题)以下程序实现的功能是：利用系统函数 rand 产生两个 0～99 的随机整数，之后进行算术四则运算(加、减、乘、除)，用户先输入运算符，接着输入对应运算的结果，最后由程序来判断是否正确，并输出相应信息。

分析、运行该程序，体会 switch 语句和随机函数的用法，并回答相关问题。

程序代码如下：

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main()
{
    int a,b,result=-1,input=0;
    char op;
    /* 系统函数 rand: 产生 0~32767 随机整数,头文件是 stdlib.h
    系统函数 srand(int seed):seed 相同,产生随机数中也相同,头文件是 stdlib.h
    系统函数 time(0):返回系统时间的总秒数,头文件是 time.h
    */

    srand(time(0));
    a=rand()%100;
    b=rand()%100;
    printf("输入算术运算的运算符(+、-、x、/): ");      //'*'不可用,改用'x'(小写)
    op=getchar();

    switch(op)
    {
        case '+':
            result=a+b;
            printf("%d +%d=",a,b);
            scanf("%d",&input);
            break;
        case '-':
            result=a-b;
            printf("%d -%d=",a,b);
            scanf("%d",&input);
            break;
        case 'x':
            result=a*b;
            printf("%d* %d=",a,b);
            scanf("%d",&input);
```