第四章 选择结构程序设计

在程序中经常需要计算机进行逻辑判断，然后根据逻辑判断的不同结果进行相应的处理。例如，求 x 的绝对值，若 x≥0，则直接取 x；若 x＜0，则取－x。这种根据某种条件的成立与否而采用不同的程序段进行处理的程序结构称为选择结构。

通常选择结构有两个分支，条件为“真”，执行 A 程序段，否则执行 B 程序段。如图 4-1 所示。有时，两个分支仍不能完全描述实际问题。如一元二次方程 $ax^2+bx+c=0$ 的求解，需要考虑根公式大于 0、等于 0、小于 0 三种情况。又如，为学生成绩划分等级（A 等：90—100，B 等：80—89，C 等：60—79，D 等：0—59），需要根据学生的成绩情况，分成四个分支分别处理。这样的程序结构称为多分支选择结构。

在选择结构中，首先计算条件表达式的值，根据其逻辑值的真与假执行不同分支程序段。条件表达式的值是逻辑值。条件表达式可以是关系表达式、逻辑表达式、值为逻辑型或可看作逻辑型值的其他表达式。最常见的是关系表达式。

C 语言提供了进行逻辑判断的选择语句 if 和 switch，利用它们可以方便地实现选择结构程序。

本章将介绍选择语句及选择结构程序设计。

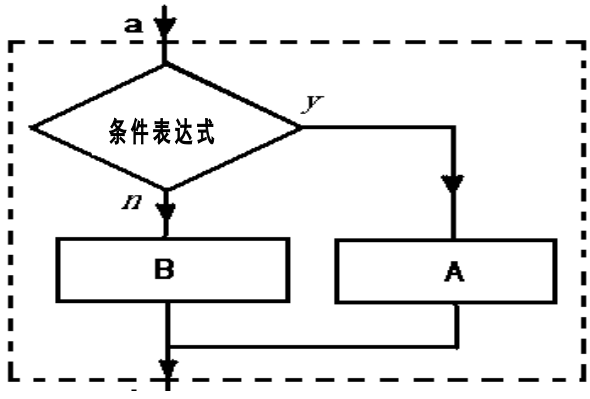


图 4-1　选择结构流程图

4.1 关系运算符和关系表达式

在程序中经常需要比较两个量的大小关系， 以决定程序下一步的工作。比较两个量的运算符称为关系运算符。

4.1.1 关系运算符及其运算优先次序

C 语言中提供了以下六种关系运算：

< 小于	<= 小于或等于	> 大于
>= 大于或等于	== 等于	!= 不等于

关系运算符都是双目运算符，其结合性均为左结合。<、<=、>、>=的优先级相同，高于==和!=，==和!=的优先级相同。

关系运算符的优先级低于算术运算符，高于赋值运算符。

注意：在 C 语言中，等于运算符是“==”，不同于赋值运算符“=”。

例如：

a==b　　判断变量 a 的值和变量 b 的值是否相等

a=b　　将变量 b 的值赋予变量 a

4.1.2 关系表达式

由关系运算符连接而成的表达式称为关系表达式。

关系运算符的两边可以是算术表达式、关系表达式，逻辑表达式、赋值表达式、字符表达式等。如 a+b>c-d、x>3/2、'a'+1<c、-i-5*j==k+1 等都是合法的关系表达式。也允许出现嵌套的情况，如 a>(b>c)、a!=(c==d)等。

关系运算的结果是一个逻辑值，即“真”或“假”。在 C 语言中，没有专门的逻辑类型数据，而是以“1”代表逻辑“真”，以“0”代表逻辑“假”。 关系运算的结果也可以参与其它数据类型的运算。

求关系表达式的值时要注意运算优先级、结合性以及逻辑结果的取值只能是 0 和 1。

【例 4.1】设 a=1，b=6，c=7，求下列关系表达式的值。

（1）c>a+b

（2）c=b>a

答案如下：

（1）值为 0。c>a+b 等价于 c>（a+b），逻辑结果为假。

（2）值为 1。关系运算符的优先级大于赋值运算符，将 b>a 的结果 1 赋给 c。

4.2 逻辑运算符和逻辑表达式

逻辑运算符用于对逻辑值进行运算，运算结果仍为逻辑值。

4.2.1 逻辑运算符及其运算优先次序

C 语言中提供了以下三种逻辑运算符：

&& 与运算　　|| 或运算　　! 非运算

当 a 与 b 的值为不同组合时，各种逻辑运算结果如表 4-1 所示。

表 4-1　逻辑运算结果表

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

0	0	1	1	0	0

与运算符“&&”和或运算符“||”均为双目运算符，具有左结合性。 非运算符“!”为单目运算符，具有右结合性。逻辑运算符中，非运算符“!”运算优先级最高，其次是与运算符“&&”，或运算符“||”的级别最低。

逻辑运算符“&&”和“||”的优先级低于关系运算符，“！”高于算术运算符。各运算符优先级由高到低可归纳为：

“！”>算术运算符>关系运算符>“&&”>“||”>赋值运算符

例如：

a>b && c>d 等价于(a>b) && (c>d)

!b==c||d<a 等价于((!b)==c)||(d<a)

a+b>c && x+y<b 等价于((a+b)>c) && ((x+y)<b)

4.2.2 逻辑表达式

逻辑表达式是用逻辑运算符将关系表达式或逻辑值连接起来构成的式子。

逻辑表达式的运算结果为1（真）或0（假）。系统处理时，以0代表“假”，以非0代表“真”。

注意：

（1）关系运算符不能连用。如数学关系式10<y<30，应写成y>10&&y<30。

（2）逻辑表达式可能会没有完全运算。如表达式x&&y&&z只有当x为真时，才要判别y的值，只有x和y都为真时，才要判别z的值。只要x为假就不必判别y和z的值。

或运算也存在这样的问题。表达式x||y||z只有当x为假时，才要判别y的值，只有x和y都为假时，才要判别z的值。只要x为真就不必判别y和z的值。

【例4.2】设a=1，b=3，c=5，d=8，x=1，y=1，则表达式(x=a>b)&&(y=c>d)执行后x和y的值为多少？

解：x=0，y=1。因为a>b的值为0，所以x=0，该表达式的值必为0，y=c>d未被执行。

4.3 if语句

使用if语句可以构成分支结构程序。C语言的if语句有三种基本形式：单分支、双分支与多分支。

4.3.1 单分支if语句

格式：

if(表达式) 语句;

语句的执行顺序是：如果表达式的值为真，则执行其后的语句， 否则不执行该语句。执行流程如图 4-2 所示。

说明：

（1）if 关键字之后的条件表达式两边必须加圆括号，语句之后必须加分号。

（2）条件表达式的值为真时，只执行其后的一条语句，若要执行多条语句应使用花括号“{}”把这些语句括起来构成复合语句。

（3)条件表达式通常是逻辑表达式或关系表达式，也可以是其它表达式(赋值表达式等)，甚至还可以是一个常量或变量。

例如，下列语句形式都是允许的。

```
if(a=3)c=10;
if(b)a=1;
if(3)c=10;
if( ‘a’ )c=10;
```

按照 if 语句的执行过程，都将先计算表达式的值，值为非 0，即为逻辑“真”，值为 0，则为逻辑“假”。 if(a=3)c=10;和 if(3)c=10;语句中表达式的值永远为 3，非 0，所以条件为“真”，if(‘a’)c=10;语句中表达式的值也非 0，条件也为“假”，所以其后的语句 c=10;总是被执行。

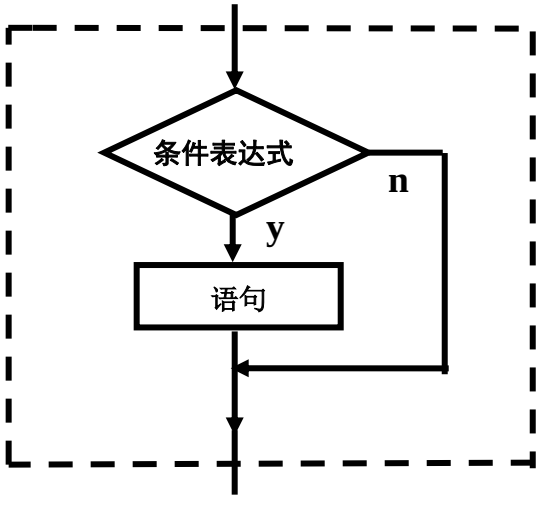


图 4-2 单分支条件语句执行流程图

【例 4.3】阅读程序，写出程序结果。

程序如下：

```
#include <stdio.h>
void main()
{   int a=1,b=2,t=0;
if (a=0) t=a;a=b;b=t;
    printf(“%d,%d\n”,a,b);
}
```

运行结果为：

函数体第二行 if (a=0) t=a;a=b;b=t;中，t=a;a=b;b=t;三条语句没有用“{}”括起来，所以等价于：

```
if (a=0) t=a;
a=b;b=t;
```

执行时，0 赋给 a，条件为假，顺次执行 a=b;b=t;，因此 a=2，b=0。

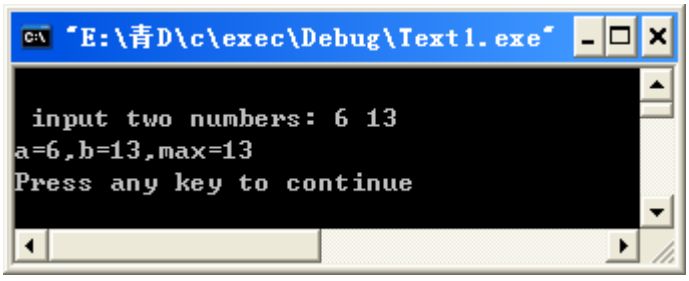
【例 4.4】编写程序完成：输入两个整数，输出其中的大数。

分析：本例中，设置变量 max 以及存放两个整数的变量 a 和 b，将其中一个数（变量 a 的值）赋予变量 max；再用 if 语句判别 max 与另一个数 b 的大小，如果 max 小于 b，则把 b 的值赋予 max，即保证 max 中为大者；最后输出 max 的值。这种方法也适用于从更多的数中找出最大者，只要不断地比较 max 与其它各数，使 max 中始终为大者即可。

程序如下：

```
#include <stdio.h>
void main()
{
    int a,b,max;
    printf("\n input two numbers: ");
    scanf("%d%d",&a,&b);
    max=a;
    if (max<b) max=b;
    printf("a=%d,b=%d,max=%d\n",a,b,max);
}
```

运行结果为：



程序运行后，输入 6 和 13，并以空格隔开后回车，即可输出结果。

思考：（1）从两个数中找出最小数如何编写程序？

（2）若从三个或更多的数中找出最大数应如何编程？

4.3.2 双分支 if 语句

格式：

```
if(表达式)
    语句 1;
else
    语句 2;
```

语句的执行顺序是：如果表达式的值为真，则执行语句 1，否则执行语句 2 。语句 1 和语句 2 执行后均转去执行本 if 语句的下一语句。执行流程如图 4-3 所示。

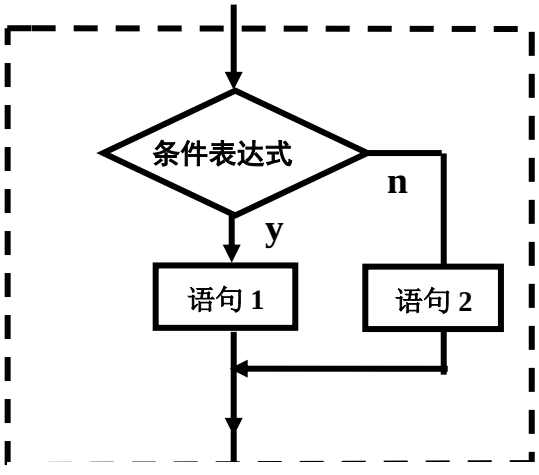


图 4-3 双分支条件语句执行流程图

【例 4.5】输入两个整数 a 和 b，按由大到小的顺序输出这两个数。

分析：本例根据 a、b 变量的大小有不同的输出顺序，两种情况分别在两个分支中实现。

程序如下：

```
#include <stdio.h>
void main()
{
    int a, b;
    printf("input two numbers: ");
    scanf("%d%d",&a,&b);
    if(a>b)
        printf("%d,%d\n",a,b);
```

```
    else
        printf("%d,%d\n",b,a);
}
```

运行结果为：

程序运行后，输入 10 和 20 回车，即可得到结果。

注意：虽然关键字 else 后没有写条件表达式，但其隐含了条件“if 后条件表达式的值为假”。例 4.5 中隐含了条件“a≤b”。

【例 4.6】求四个整数中，偶数的和与奇数的个数。

分析：

（1）判别某数 a 是否偶数实际上是判别该数能否被 2 整除，即 a%2==0 成立否。扩展开来，判断某数是否能被 k 整除的条件表达式为：a%k==0。

（2）求累加和与统计个数方法类似。本例中，设置 sum 变量存放偶数的和。求和前，先将 sum 清 0，然后通过赋值语句 sum=sum+a;把判断出的偶数 a 累加到 sum 中。赋值号右侧的 sum 是本次累加以前的结果，加上新值 a，再赋给 sum，得到新的累加和。这是求累加和的基本方法。设置 num 变量统计奇数的个数，其处理方法类同变量 sum 的处理。

程序如下：

```
#include "stdio.h"
void main()
{   int a,b,c,d;
    int sum=0,num=0;
    scanf ("%d,%d,%d,%d", &a,&b,&c,&d);
    if(a%2==0)sum=sum+a; else num=num+1;
    if(b%2==0)sum=sum+b; else num=num+1;
    if(c%2==0)sum=sum+c; else num=num+1;
    if(d%2==0)sum=sum+d; else num=num+1;
    printf("a=%d,b=%d,c=%d,d=%d\n",a,b,c,d);
    printf ("sum=%d,num=%d\n",sum,num);
}
```

运行结果为：

输入：12,3,27,8↙

输出：a=12,b=3,c=27,d=8

sum=20,num=2

思考：如果求若干个数的乘积又应如何处理？

4.3.3 多分支 if 语句

格式：

```
if(表达式 1)
    语句 1;
else if(表达式 2)
    语句 2;
…
else if(表达式 m)
    语句 m;
else
    语句 n;
```

语句的执行顺序是：依次判断各表达式的值，当某个表达式的值为真时， 则执行其后对应的语句，然后转到整个 if 语句的下一语句继续执行后续程序。 如果所有的表达式均为假，则执行 else 后的语句 n ，再转下一语句继续执行后续程序。语句的执行过程如图 4-4 所示。

【例 4.7】编程完成判别键盘输入字符的类别。

分析：字符的类别可以根据字符的 ASCII 码来判别。由 ASCII 码表可知 ASCII 码值小于 32 的为控制字符；在“0”和“9”之间的为数字；在“A”和“Z”之间的为大写字母；在“a”和“z”之间的为小写字母；其余则为其它字符。本例是一个多分支选择问题，由 getchar() 函数接收键盘键入字符，用 if-elseif 语句实现判断并给出不同的输出结果。

程序如下：

```
#include"stdio.h"
void main()
{char c;
    printf("input a character: ");
    c=getchar();
    if(c<32)
        printf("This is a control character\n");
    else if(c>='0'&&c<='9')
```

```
        printf("This is a digit\n");
    else if(c>='A'&&c<='Z')
        printf("This is a capital letter\n");
    else if(c>='a'&&c<='z')
        printf("This is a small letter\n");
    else
        printf("This is an other character\n");
}
```

运行结果为：

显示：input a character:

输入：a↙

输出：This is a small letter

注意：

（1）判断变量 c 是否大写字母，不能写成'A'<=c<='Z'。这种形式编译系统并不报错，理解为（'A'<=c）<='Z'，所以，先判断'A'<=c 成立否，然后判断该逻辑值是否<='Z'，可能导致不在该范围内的字符也会得到逻辑“真”。

（2）除控制字符的判断使用 ASCII 码值外，其它可视字符直接判断，不需要查找他们的 ASCII 码值。

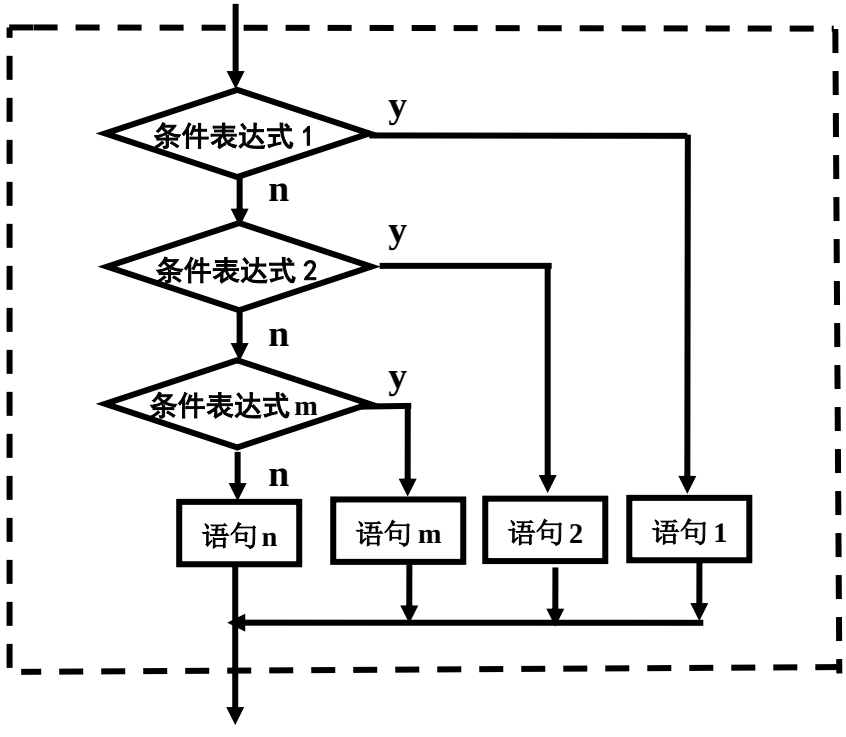


图 4-4 多分支条件语句执行流程图

【例 4.8】输入某学生的成绩，输出该学生的成绩和等级。等级划分：90-100 分为 A 级、80-89 分为 B 级、60-79 分为 C 级、0-59 分为 D 级。

分析：首先提示用户输入成绩，当所输入的成绩不在合理范围(0≤score≤100)时，提示输入错误，程序执行结束。成绩在合理范围内时，满足 score≥90 相当于 100≥score≥90，属于 A 级；不满足 score≥90，score<90 自然满足，只要 score≥80 相当于 90> score≥80，属于 B 级；……。依次类推，可以处理其它分支情况。

程序如下：

```
#include "stdio.h"
void main()
{     int score;
      printf("Please input score:\n");
      scanf("%d",&score);
      if(score>100|| score<0)
            printf("score error!\n");
      else if(score>=90)
            printf ("score=%d,grade A\n", score);
      else if (score>=80)
            printf ("score=%d,grade B\n", score);
      else if (score>=60)
            printf ("score=%d,grade C\n", score);
      else
            printf ("score=%d,grade D\n", score);
}
```

运行结果为：

显示：Please input score:
输入：120↙
输出：score error!
显示：Please input score:

输入：76↙
输出：score=76,grade C
显示：Please input score:
输入：34↙
输出：score=34,grade D

以上是 3 次运行的结果。

4.3.4 i f 语句的嵌套

当 if 语句中的执行语句又是 if 语句时，则构成了 if 语句的嵌套。利用 if 语句的嵌套可以实现多分支结构。其一般形式如下：

```
if(表达式)
    if 语句;
else
    if 语句;
```

说明：

（1）允许嵌套的 if 语句是 if 型或 if-else 型，这时应注意 if 和 else 的配对问题。为避免二义性，C 语言规定，else 总是与它前面最近的未配对的 if 配对。例如有如下结构：

```
if(表达式 1)
if(表达式 2)
语句 1;
else
语句 2;
```

应该将其理解为结构①，而不是结构②，因为 else 应与距其最近的第 2 个 if 配对。

结构①：

```
if(表达式 1)
  { if(表达式 2)
        语句 1;
    else
        语句 2; }
```

结构②：

```
if(表达式 1)
    {if(表达式 2)
        语句 1; }
else
    语句 2;
```

结构③：

```
if(表达式 1)
if(表达式 2)
        语句 1;
else
    语句 2;
```

（2）恰当地使用“{}”构成复合语句，可以构造不同的嵌套结构。例如上述结构②。

（3）通过在语句前加空格可增加多分支结构的可读性，但不会改变程序的结构。例如上述结构③不可理解为结构②，结构③的功能等同于结构①。

（4）一般情况下，应尽量较少使用 if 语句的嵌套结构，多分支结构用 if-elseif 语句实现，程序更便于阅读理解，结构也更加清晰。

【例 4.9】计算如下分段函数：

$$y=\begin{cases} x-5 & (x>15) \\ 5+2x & (0<x\leqslant 15) \\ 0 & (x\leqslant 0) \end{cases}$$

程序如下：

```
#include "stdio.h"
void main()
{     int x,y;
      scanf ("%d",&x);
      if(x>15)
          y=x-5;
      else                                          /*  x≤15 */
          if(x>0)y=5+2*x;
          else y=0;
      printf ("x=%d,y=%d\n",x,y);
}
```

运行结果为：

输入：10↙

输出：x=10,y=25

分析：本例中求 y 的多分支程序段可以有多种写法，下述程序段①～⑤都是正确的。

程序段①：

```
if(x>0)
  if(x>15)y=x-5;
  else y=5+2*x;
else
  y=0;
```

程序段②：

```
if(x<=0)
  y=0;
else
  if(x>15)y=x-5;
  else y=5+2*x;
```

程序段③：

```
if(x<=0)
    y=0;
else if(x>15)
    y=x-5;
else
    y=5+2*x;
```

程序段④：

```
y=0;
if(x<=15)
    {if(x>0)y=5+2*x;}
else y=x-5;
```

程序段⑤：

```
y=0;
if(x>0)
    if(x<=15)y=5+2*x;
    else y=x-5;
```

4.4 条件运算符和条件表达式

在条件语句的各分支中，如果只执行单个赋值语句，使用条件运算符构成的条件表达式来实现，不仅程序代码简洁，而且可以提高程序的运行效率。

条件运算符为“?”和“:”，是一个三目运算符，即有三个参与运算的量。由条件运算符组成条件表达式，一般形式为：

表达式 1?表达式 2:表达式 3

执行过程为：先求解表达式 1，如果表达式 1 的值为非 0（真），则求解表达式 2 并将其值作为整个条件表达式的值，否则求解表达式 3 并将其值作为整个条件表达式的值。

例如，找出变量 a 与 b 中大数，用条件表达式可写为：

```
max=(a>b)?a:b;
```

该语句的语义是：如果 a>b 为真，则把 a 的值赋予 max，否则把 b 的值 赋予 max。

说明：

（1）条件运算符?和：是一对运算符，不能分开单独使用。

（2）条件运算符的运算优先级低于逻辑运算符、关系运算符和算术运算符，但高于赋值运算符。因此，语句 max=(a>b)?a:b;可以去掉括号写为 max=a>b?a:b;，条件表达式 a>b?a:b+1 应理解为 a>b?a:(b+1)而不是(a>b?a:b)+1。

（3）条件运算符可以嵌套使用构成多分支。例如求变量 x 的符号可表达为：

```
sgn=x>0?1:(x==0?0:-1)
```

(4)条件运算符的结合方向是自右至左。例如:a>b?a:c>d?c:d应理解为a>b?a:(c>d?c:d)。

【例 4.10】输入一个字符，如为大写字母则转换为小写字母，最后输出该字符。

分析：小写字母与大写字母的 ASCII 码值相差 32。如变量 str 的值为大写字母，则加上 32 后即可转换为小写字母，否则维持不变。

程序如下：

```
#include "stdio.h"
void main()
```

```
{   char str;
    scanf("%c",&str);
    str=(str>=' A' &&str<=' Z' )?(str+32):str;
    printf("%c\n",str);
}
```

运行结果为：

输入：C↙

输出：c

4.5 switch语句和goto语句

本节介绍多分支选择结构语句switch语句和无条件转向语句goto语句。

4.5.1 switch语句

switch语句是C语言提供的一种多分支选择语句，用来实现多分支选择结构。一些多分支问题用switch语句来处理程序的结构更简洁，可读性更强。

格式：

```
switch(表达式)
{
    case 常量表达式1: 语句1;
    case 常量表达式2: 语句2;
    …
    case 常量表达式n: 语句n;
    default : 语句n+1;
}
```

语句的执行顺序是：计算switch后表达式的值，并逐个与case后常量表达式的值相比较，当表达式的值与某个常量表达式的值相等时，执行其后的语句，然后不再进行判断，继续执行下面所有case后的语句。 如果表达式的值与所有case后常量表达式的值均不相等，则执行default后的语句。

说明：

（1）switch后面的表达式可以是C语言中任意合法表达式，其两边的括号不能省略。

（2）case后面的常量表达式中不能出现变量，且类型必须是整型、字符型或枚举型，各个常量表达式的值不能重复，且与case间要有空格隔开。

（3）default 为可选项，其与 case 语句的出现顺序没有严格限制，可出现在任何位置，不会影响执行结果。

（4）case 后面的语句可以是一条或多条语句，也可以为空。为多条语句时，可以不用“{}”将它们括起来；为空时，程序执行到此会自动向下顺序执行，实现多个 case 共用一组执行语句。

例如，下面程序段当表达式的值为 8、9、10 时，均可输出“成绩优良”字样。

```
  …
case 10:
case 9:
case 8:printf(“成绩优良”);break;
case 7:
  …
```

（5）执行完一个 case 后面的语句后，会继续顺序执行下面各 case 语句，且不再进行条件判断，因此应注意使用 break 语句跳出 switch 语句。

break 语句格式：

```
break;
```

功能：终止其所在的 switch 语句或循环语句的执行。

【例 4.11】输入一个整数，如在 1～7 间输出对应的星期英文单词，否则提示出错。

分析：下列程序段①只有最后一个分支的结果是正确的，使用 break 语句改进后的程序段②可以完成程序功能。

程序①如下：

```
#include "stdio.h"
void main(){
  int num;
  printf("input a integer number: ");
  scanf("%d",&num);
  switch (num){
  case 1:printf("Monday\n");
  case 2:printf("Tuesday\n");
  case 3:printf("Wednesday\n");
  case 4:printf("Thursday\n");
  case 5:printf("Friday\n");
  case 6:printf("Saturday\n");
  case 7:printf("Sunday\n");
  default:printf("error\n");}
```

程序②如下：

```
#include "stdio.h"
void main(){
  int num;
  printf("input a integer number: ");
  scanf("%d",&num);
  switch (num){
  case 1:printf("Monday\n");break;
  case 2:printf("Tuesday\n"); break;
  case 3:printf("Wednesday\n"); break;
  case 4:printf("Thursday\n"); break;
  case 5:printf("Friday\n"); break;
  case 6:printf("Saturday\n"); break;
  case 7:printf("Sunday\n"); break;
  default:printf("error\n");}
}
```

运行结果为：

显示：input a integer number:

输入：6↙

输出：Saturday

Sunday

error

运行结果为：

显示：input a integer number:

输入：5↙

输出：Friday

【例 4.12】编写计算器程序。由用户输入运算数和四则运算符，最后输出计算结果。

分析：本例可用于四则运算求值。使用 switch 语句判断运算符，然后运算输出结果。当输入运算符不是+、-、*、/时提示错误。

程序如下：

```
#include "stdio.h"
void main()
{float a,b,s;
      char c;
      printf("input expression: a+(-,*,/)b \n");
      scanf("%f%c%f",&a,&c,&b);
      switch(c)
      {
            case '+': s=a+b;break;
            case '-': s=a-b;break;
            case '*': s=a*b;break;
            case '/': s=a/b;break;
            default: printf("input error\n");exit();
      }
      printf("%.2f%c%.2f=%.2f\n",a,c,b,s);
}
```

运行结果为：

显示：input expression: a+(-,*,/)b

输入：6+7↙

输出：6.00+7.00=13.00

上述多分支程序段也可以改写为：

```
switch(c)
{
      case '+': printf("%f\n",a+b);break;
      case '-': printf("%f\n",a-b);break;
      case '*': printf("%f\n",a*b);break;
      case '/': printf("%f\n",a/b);break;
      default: printf("input error\n");
}
```

4.5.2 goto 语句

goto 语句是无条件转向语句。

格式：

```
goto 语句标号;
```

功能：将程序的执行转向语句标号所在的位置。

说明：

（1）语句标号的命名规则与变量名相同，标号以冒号结尾。

（2）语句标号应与 goto 语句出现在同一函数中。

（3）goto 语句可以方便、快速地转到指定的任意位置继续执行，这种任意性破坏了程序的顺序流程，会使程序的可读性、可维护性变差，因而结构化程序设计中不提倡使用 goto 语句。

【例 4.13】输入 5 个数，统计负数的个数并输出。

分析：本例通过 goto 语句实现跳转，重复执行同一段程序处理 5 个数据，设置变量 num 记录处理数据的个数，当 5 个数处理完后就不再重复。每次输入一个数 data，如果是负数，则使 count 变量加 1，记录负数个数。最后输出结果。

程序如下：

```
#include "stdio.h"
void main()
{float data;
      int count=0,num=1;
      n1:
```

```
        printf("input data %d:\n",num);
        scanf("%f",&data);
        if(data<0)count++;
        num++;
        if(num<=5)goto n1;                    /*如果 num<=5，转回标号 n1 处*/
        printf("count=%d\n",count);
    }
```

程序运行结果如图 4-5 所示。

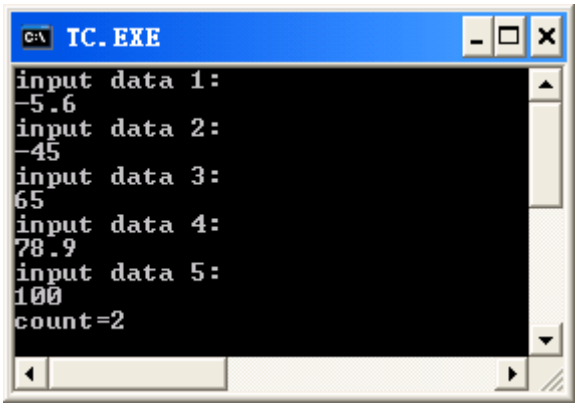


图 4-5 例 4.13 运行结果图

4.6 选择结构程序举例

本节通过几个典型的例子来学习选择结构程序的实际应用。

【例 4.14】输入 3 个数，按从大到小的顺序输出。

分析：设 3 个数分别是 a、b 和 c，把它们中最大者存放在 a 中，把次大者存放在 b 中，c 中存放最小者，最后依次输出 a、b 和 c。本例实际完成 3 个数的排序操作。

将 3 个数中的最大数存入 a，需比较两次，每次比较均使 a 中存放两者中的大数，先 a 与 b 比，再 a 与 c 比。b 中存入次大数，是在剩下的两个数 b 与 c 中找大者，比较一次。

在每次比较处理中，因为不能丢失任何数据，所以不能简单进行数据赋值，应互换数据。a 与 b 中数据互换使用 3 条赋值语句通过中间变量 t 来完成，即 t=a; a=b; b=t;。

程序如下：

```
#include "stdio.h"
```

```
void main()
{  int a,b,c,t;
  printf("Please input a,b,c:\n");
   scanf("%d,%d,%d",&a,&b,&c);
   if(a<b){t=a;a=b;b=t;}          /* a 和 b 的值交换，a 中存放较大的数*/
   if(a<c){t=a;a=c;b=t;}          /* a 和 c 的值交换，a 中存放最大数*/
   if(b<c){t=b;b=c;c=t;}          /* b 和 c 的值交换，c 中存放最小数*/
   printf("%d,%d,%d\n",a,b,c);
}
```

运行结果如下：

输入：16，80，7↙

输出：80,16,7

思考：如有 5 个数需按由小到大顺序输出应如何编写程序？

【例 4.15】输入两个整数，若它们的平方和大于 100，则输出该平方和的百位数以上（包括百位数字）的各位数字，否则输出两个整数的和。

分析：有一个整数 m，取其百位数以上(含百位数)数字的表达式为 m/100，取百位数以下(不含百位数)数字的表达式为 m%100。

有效地使用运算符“/”、“%”及“*”可以进行数据分解与组合。例如，由两位数 a 和 b 构成数据 c，要求将数据 a 的十位和个位数依次放在数据 c 的个位和十位上，数据 b 的十位和个位数依次放在数据 c 的百位和千位上，表达式为：

```
c=(b%10)*1000+(b/10)*100+(a%10)*10+a/10
```

程序如下：

```
#include "stdio.h"
void main()
{ int a,b,c,d;
  printf("Please input a,b\n");
  scanf("%d,%d",&a,&b);
  c=a*a+b*b;
  if(c>100)
    {  d=c/100;
       printf("%d,%d\n", c,d);
    }
  else
    printf("a+b=%d\n",a+b);
```

```
}
```

运行结果如下：

输入：15，53↙

输出：3034,30

输入：3,9↙

输出：a+b=12

【例 4.16】求一元二次方程 $ax^2+bx+c=0(a\neq0)$的根。

分析：当 a≠0 时，一元二次方程 $ax^2+bx+c=0$ 的解根据判别式有三种情况：

（1）若 b*b-4*a*c>0，有两个不相等的实根；

（2）若 b*b-4*a*c=0，有两个相等的实根；

（3）若 b*b-4*a*c<0，有两个共扼复根。

程序如下：

```
#include "stdio.h"
#include "math.h"
void main()
{    float a,b,c,pb,temp,x1,x2;
     printf( "Input a,b,c\n");
     scanf("%f,%f,%f",&a,&b,&c);
     if (a==0)                          /*如 a=0，非一元一次方程*/
          printf("input data error!\n");
     else                                    /* 解一元二次方程 */
         {   pb=b*b-4*a*c;
             temp=2*a;
             x1=-b/temp;
             if(pb>0)                         /* 输出两个不等实根 */
                  { x2=sqrt(pb)/temp;
                    printf(" real root:\n ");
                    printf("r00t1=%f,r00t2=%f\n",x1+x2,x1-x2);}
              else if(pb==0)                  /* 输出两个相等实根 */
                  printf("root=%f\n",x1);
              else                                    /* 输出复根 */
                  {  x2=sqrt(-pb)/temp;
                    printf("complex root:\n");
                     printf("root1=%f+%fi\n", x1,x2);
```

```
                printf ("root2=%f-%fi\n", x1,x2);
            }
        }
    }
```

运行结果如下：

输入：0,10,5↙

输出：input data error!

输入：1,2,5↙

输出：complex root:

root1= -1.000000 + 2.000000i

root2= -1.000000 - 2.000000i

输入：1,5,2↙

输出：real root:

root1=-0.438447,root2=-4.561553

【例 4.17】运输公司为每种货物制订了单位运费 p，实际运费 f 根据货物重量 w、运输里程 s 而不同，当运输里程达到 250 公里及以上会得到运费折扣 d，路程越远折扣越大。运费计算公式为：f=p×w×s×(1-d)。假如折扣标准如下，请编写程序计算实际运费。

250≤s＜500，2%；500≤s＜1000，5%；1000≤s＜2000，8%；2000≤s＜3000，10%；3000≤s，15%。

分析：本例的关键问题是判断运输里程的范围，进而得到相应的运费折扣，再根据公式计算实际运费。是一个多分支问题。可以看到折扣变化起点均是 250 的倍数，因此将 s/250，记为 h，得到如下折扣标准：

1≤h＜2，2%；2≤h＜4，5%；4≤h＜8，8%；8≤h＜12，10%；12≤h，15%。

程序如下：

```
#include "stdio.h"
#include "math.h"
void main()
{    float p,f,w,s,d;
     int h;
     printf( "Input p,w,s\n");
     scanf("%f,%f,%f",&p,&w,&s);
     h=(int)s/250;
      switch(h)
      {     case 0:d=0;break;
```

```
            case 1:d=0.02;break;
            case 2:
            case 3:d=0.05;break;
            case 4:
            case 5:
            case 6:
            case 7:d=0.08;break;
            case 8:
            case 9:
            case 10:
            case 11:d=0.1;break;
            default:d=0.15;break;
        }
        f=p*w*s*(1-d);
         printf("s=%.2f,d=%.2f,f=%15.2f\n",s,d,f);
    }
```

运行结果如下：

输入：100,25,476↙

输出：s=476.00,d=0.02,f=　　1166200.00

习题四

一、单选题

1. if 语句的控制条件 。

A. 只能用关系表达式　　B. 只能用关系表达式或逻辑表达式

C. 只能用逻辑表达式　　D. 可以用任何表达式

2. 以下运算符中优先级最低的为：

A. &&　　B. &　　C. ||　　D. =

3. 已知 char c='A' ; int i=1, j; 执行语句 j=!c&&i++; 后，i 和 j 的值是:

A. 1, 1　　B. 1,0　　C. 2,1　　D. 2,0

4. 以下错误的 if 语句是：

A.if (x>y);　　B.if (x= =y) x+=y;

C.if (x!=y) scanf("%d",&x) else scanf("%d",&y);　　D.if (x<y){x++;y++;}

5. C 语言对嵌套 if 语句的规定是：else 总是与_____配对。

A.其之前最近的 if　　B.第一个 if

C.缩进位置相同的 if　　D.其前面最近的且尚未配对的 if

6. 已知 int x,a,b;则下列选项中错误的 if 语句是：

A.if (a=b) x++;　　B.if (a=<b) x++;

C.if (a-b)　x++;　　D.if (x) x++;

7. 在下面的条件语句中（其中 s1 和 s2 表示是 C 语言的语句），只有一个在功能上与其它 3 条语句不等价，它是：

A.if (a) s1;else s2;　　B.if (a= =0) s2; else s1;

C.if (a!=0) s1; else s2;　　D.if (a= =0) s1;else s2;

8. 下列关于 switch 语句和 break 语句的结论中，正确的是：

A.break 语句是 switch 语句中的一部分

B.switch 语句中可以根据需要使用或不使用 break 语句

C.在 switch 语句中必须使用 break 语句

D.break 语句只能用在 switch 语句中

9. 若要求在 if 后一对圆括号中表示条件 "a 不等于 0 成立"，则能正确表示这一关系的表达式为：

A.a<>0　　B.!a　　C.a=0　　D.a

10. 已知 int w=3, x=10, z=7; 则执行下面语句后的输出结果为:

printf("%d", x>10?+100:x-10);

```
printf("%d",w++||z++);
printf("%d",!w>z);
printf("%d",w&&z);
```

A.0111　　B.11111　　C.0101　　D.0100

11. 当 c 的值不为 0 时，在下列选项中能够将 c 的值赋给变量 a、b 的是:

A.c=b=a;　　B.(a=c)||(b=c) ;

C.(a=c)&&(b=c);　　D.a=c=b;

12. 以下程序的输出结果是：

```
main()
{ float x=2,y;
if(x<0) y=0;
else if(x<5&&!x) y=1/(x+2);
else if(x<10) y=1/x;
else y=10;
printf("%f\n",y);
}
```

A. 0.000000 B. 0.250000 C. 0.500000 D. 10.000000

13. 执行下列程序段后，x、y 和 z 的值分别是：

```
int x=10,y=20,z=30;
if(x>y) z=x;x=y;y=z;
```

A. 10,20,30　　B. 20,30,30　　C. 20,30,10　　D. 20,30,20

14. 以下程序的输出结果是：

```
main( )
{ int w=4,x=3,y=2,z=1;
if(x>y&&!(z==w))printf("%d\n",(w<x?w:z<y?z:x));
else printf("%d\n", (w>x?w:z>y?z:x));
}
```

A. 1　　B. 2　　C. 3　　D. 4

15. 若 a 和 b 均是整型变量，以下正确的 switch 语句是：

```
A. switch (a/b)
{ case 1: case 3.2: y=a+b; break ;
case 0: case 5: y=a-b;
}
```

```
B. switch (a*a+b*b);
{case 3:
 case 1: y=a+b; break ;
  case 0: y=b-a; break; }
```

```
C. switch a
```

```
D. switch(a+b)
```

```
{ default : x=a+b;
case 10 : y=a-b;break;
case 11 : y=a*d; break; }
```

```
{case 10: x=a+b; break;
case 11: y=a-b; break;
}
```

二、填空题

1. 在C语言中，用____表示逻辑值“真”，用____表示逻辑值“假”。
2. “if(!k)a=3;”语句中的!k改写为_____，其功能不变。
3. 表达“a=b或a<c”的条件表达式为_____。
4. 表达“若|x|>4，则输出x，否则输出：error！”的if语句是_____。
5. 能正确表达“当x的值是[1,10]或[200,210]范围内的奇数时，输出x”的if语句是___ 。
6. 下列程序段的输出是_____ 。

```
int i=0,k=100,j=4;
if (i+j) k=(i=j)?(i=1):(i=i+j);
printf ("k=%d\n",k);
```

7. 以下程序的输出是_____ 。

```
main( )
{ int a=0, b=0, c=0;
if (a=b+c) printf ("*** a=%d\n", a);
else printf ("$$$ a=%d\n", a);
}
```

8. 下列程序的输出结果是_____ 。

```
#include " stdio.h"
main()
{ int x=1, y=0, a=0, b=0;
switch (x)
{ case 1: switch (y)
{ case 0: a++; break;
case 1: b++; break;
}
case 2: a++; b++;
}
printf ("a=%d, b=%d\n" , a, b);
}
```

9.下面程序的运行结果是_____。

```
#include "stdio.h"
main()
{  int a=1,b=2,c=3;
   if(a>b)
     if(a>c)
        printf("%d",a);
      else printf("%d",b);
    printf("%d",c);
}
```

10. 若下列程序执行后 t 的值为 4 ，则执行时输入 a,b 的值范围是_____ 。

```
#include "stdio.h"
main()
{ int a, b, s=1, t=1;
scanf ("%d, %d", &a, &b);
if (a>0) s+=1;
if (a>b) t+=s;
else if(a==b) t=5;
else t = 2*s;
printf ("s=%d, t=%d\n", s,t);
}
```

三、编程题

1. 编写程序，输入一个年份，判断其是否闰年。为闰年的条件是：该年份能被 400 整除或能被 4 整除，但不能被 100 整除。

2. 有一函数：

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

分别用：（1）简单 if 语句 （2）嵌套的 if 语句（3）if-else 语句 （4）switch 语句编写程序，要求输入 x 的值，输出 y 的值。

3. 编写程序，输入 3 个整数，判断它们是否能够构成三角形，若能构成三角形，则输出三角形的类型（等边、等腰或一般三角形）。

4. 编写程序，加密数据。方法：对给定数值，每一位数字均加 2，且在[0，9]范围内，若加密后某位数字大于 9，则取其被 10 除的余数。如：6987 加密后为 8109。

5. 将下列程序用 switch 语句改写，并使其功能不变。

```
main()
{ int x, y;
scanf ("%d",&x);
if ( x<20 ) y = 1;
if ( x<30 ) y = 2;
if ( x<40 ) y = 3;
if ( x<50 ) y = 4;
if ( x<60 ) y = 5;
else y = 6;
printf("x=%d,y=%d\n",x,y);
}
```

6. 某商店为促销推出如下让利销售方案,其中 M 为购买金额，N 为让利百分比。

M<100, N=0； 100<=M<200, N=1.5%；200<=M<300, N=2.5%； 300<=M<400, N=3.5%；400<=M<500, N=4.5%； 500<=M<600, N=5.5%；M>600, N=6 %；

编写程序，对输入的购买金额，输出顾客购买金额、实际支付的金额和返还的金额。