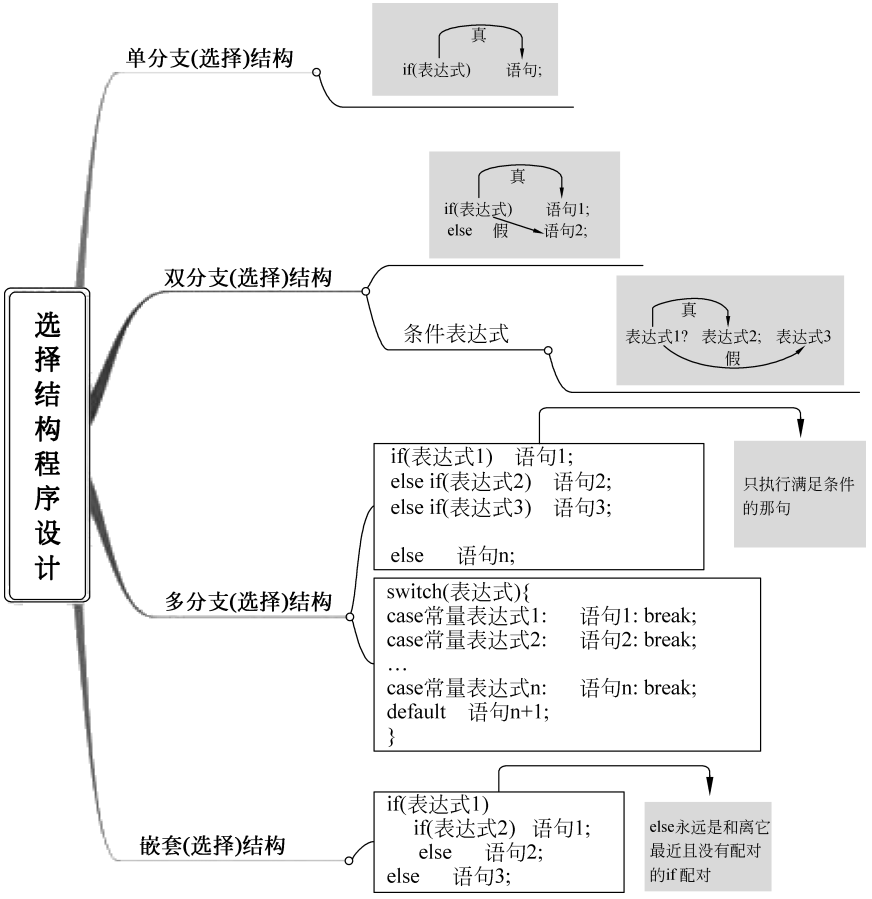
第3章 选择结构程序设计

选择结构思维导图



学习任务与目标

1. 掌握选择结构的基本思想方法；
2. 掌握条件的表示方法；
3. 掌握关系运算符和关系表达式，逻辑运算符和逻辑表达式的用法；
4. 学会用 if 语句实现选择结构；

5. 学会用 switch 语句实现多分支结构；
6. 熟悉 if 语句的嵌套使用方法；
7. 能用选择结构设计程序解决实际问题。

3.1　选择结构

3.1.1　引例：BMI 判断成年人是否肥胖

1. 描述问题

利用 BMI 公式：BMI＝体重/身高^2，计算成年人是否肥胖。

BMI(Body Mass Index，身体质量指数，简称体质指数)是国际上常用的衡量人体胖瘦程度以及是否健康的一个标准。

根据 WHO(世界卫生组织)规定的标准，亚洲人的 BMI 若高于 23 则属于超重。WHO 的标准不太适合中国人的体质程度，为此制定中国参考标准如表 3-1 所示。

表 3-1　不同标准成年人的 BMI

WHO 标准	亚洲标准	中国标准	体质程度	相关疾病发病危险性
＜18.5	—	—	偏瘦	低(但其他疾病危险性增加)
18.5～24.9	18.5～22.9	18.5～23.9	正常	平均水平
≥25	≥23	≥24	超重	
25.0～29.9	23～24.9	24～27.9	偏胖	增加
30.0～34.9	25～29.9	≥28	肥胖	中度增加
35.0～39.9	≥30	—	重度肥胖	严重增加
≥40.0	—	—	极度肥胖	极度严重增加

说明：中国成年人的肥胖标准：BMI≥24 为超重，BMI≥28 为肥胖，理想的体重指数是 22。男性腰围大于或等于 85cm、女性腰围大于或等于 80cm 为腰部肥胖标准。

2. 分析解决问题

(1) 输入你的身高和体重，分别用变量 tall 和 weight 表示；
(2) 利用公式计算 BMI 值，该值用变量 fat 表示；
(3) fat 的值与表 3-1 给定的 BMI 范围比较，输出体质程度。
参考程序如下：

```
#include <stdio.h>
void main ( )
{
    float tall,weight,fat;
    printf("please input your tall(m) and weight(kg):\n");
    scanf("%f%f",&tall,&weight);
    fat = weight/(tall * tall);
```

```
    if (fat <= 18.5)
       printf("BMI = %f, you are too thin !\n", fat);
    else if (fat <= 23.9)
          printf("BMI = %f, you are standard! \n", fat);
       else if (fat <= 27.9)
                printf("BMI = %f, you are fat! \n", fat);
            else
                 printf("BMI = %f, you are too fat! \n", fat);
}
```

【讨论】 每位成年人都适用 BMI 指数吗？例如以下人群：

(1) 运动员；

(2) 正在做身体特种训练的人；

(3) 怀孕或哺乳期的妇女；

(4) 身体虚弱或久坐不动的老年人。

如何完善上述程序？

3.1.2 选择结构的思想方法

回顾第 2 章顺序结构的基本思想方法，使用一些表达式语句、输入输出函数，可以编写简单的程序。然而，要解决多种条件下比较复杂的问题，仅靠顺序结构是不够的。例如，根据工资收入情况计算个人所得税、十字路口交通信号灯“红灯停，绿灯行”等问题的特点是先判断再选择。根据给定的条件与现有的条件进行分析、比较和判断，并按判断后的不同情况进行不同的处理，这是选择结构(又称分支结构)程序设计的思想方法。

3.1.3 选择结构程序设计步骤

选择结构程序设计应考虑两个方面的问题，即两个步骤：一是如何表示条件；二是实现选择结构需要哪些语句。在 C 语言中，一般用关系表达式或逻辑表达式表示条件，用 if 语句或 switch 语句实现选择结构。

3.2 关系运算

通常用关系运算符比较两个量的关系，以决定程序下一步的工作。由关系运算符连接构成的表达式称为关系运算表达式。日常生活中，人们的逻辑推理对应着 C 语言中的关系或逻辑运算。

3.2.1 关系运算符及其优先级

C 语言有以下六种关系运算符：

(1) ＜，小于；

(2) ＜＝，小于或等于；

(3) >,大于;

(4) >=,大于或等于;

(5) ==,等于;

(6) !=,不等于。

关系运算符属于双目运算符,其结合性均为左结合。在关系运算中,<、<=、>、>=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。

3.2.2　关系表达式

关系表达式的一般形式为:

表达式 关系运算符 表达式

例如:

```
a+b>c-d
x>3/2
'a'+1<c
-i-5*j==k+1
```

都是合法的关系表达式。

表达式中可以又包含关系表达式,即允许出现关系表达式嵌套的情况。例如:

```
a>(b>c)
a!=(c==d)
```

关系表达式的值是真或假,分别用 1 和 0 表示。例如:5>0 的值为真,即为 1;(a=3)>(b=5)由于 3>5 不成立,其值为假,即为 0。

【例 3-1】 输出关系表达式的值。

参考程序如下:

```
#include<math.h>
void main()
{
  char c='a';
  int i=1,j=2,k=3;
  float x=3e+5,y=0.85;
  printf("%d,%d\n",'a'+5<c,-i-2*j>=k+1);
                                        /*字符变量以它对应的 ASCII 码值参与运算*/
  printf("%d,%d\n",1<j<5,x-5.25<=x+y);
  printf("%d,%d\n",i+j+k==-2*j,k==j==i+5);
/*根据关系运算符的左结合性,k==j==i+5,先判断 k==j,不成立,其值为 0,再判断 0==i+5,
   也不成立*/
}
```

注意:

(1) 由于浮点数在计算机中不能非常准确地表示,因此,判断两个浮点数是否相等,通

常不使用关系运算符,而是利用区间判断方法来实现。例如,判断 x 是否等于 3.0017,可利用逻辑表达式 x>3.0016 && x<3.0018 判断,当逻辑表达式为真时,可以认为 x 等于 3.0017。

又如:

```
1.0/3.0*3.0==1.0
```

可改写为

```
fabs(1.0/3.0*3.0-1.0)<1e-6
```

应避免对实数做等于 0 或不等于 0 的判断。

(2) 应严格区分“=”与“==”运算符。

3.3 逻辑运算

3.3.1 逻辑运算符及其优先级

C 语言提供三种逻辑运算符:&&、||、!。

! 为单目运算符,具有右结合性;&& 与|| 均为双目运算符,具有左结合性。三种逻辑运算符和其他运算符的优先级如图 3-1 所示。

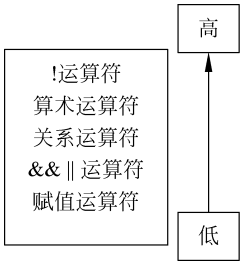


图 3-1 运算符优先级高低顺序

例如,下列运算按照运算符的优先顺序可以得出等价形式:

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)

再例如 5>3&&8<4-! 0 自左向右运算过程如下。

第一步:计算 5>3 的逻辑值为 1;

第二步:计算! 0 的逻辑值为 1;

第三步:计算 4-1 的值为 3;

第四步:计算 8<3 的逻辑值为 0;

第五步:判断 1&& 的 0 逻辑值为 0。

3.3.2 逻辑表达式

逻辑表达式的一般形式为:

表达式 逻辑运算符 表达式

其中,表达式又可以是逻辑表达式,从而组成逻辑表达式的嵌套形式。

例如:

```
(a&&b)&&c
```

根据逻辑运算符的左结合性，也可写为如下等价形式：

```
a&&b&&c
```

逻辑表达式的值是式中各种逻辑运算的最后值。

3.3.3　逻辑运算表达式的值

逻辑运算表达式的值分为真和假两种，分别用 1 和 0 表示。逻辑运算求值规则如下。

1. &&（与运算）

当参与运算的两个量都为真时，结果为真，否则为假。例如：

```
5 > 0 && 4 > 2
```

由于 5＞0 为真，4＞2 也为真，因此进行与运算的结果为真。

2. ||（或运算）

参与运算的两个量只要有一个为真，结果为真；当两个量都为假时，结果为假。例如：

```
5 > 0||5 > 8
```

由于 5＞0 为真，不必判断 5＞8，因此，或运算的结果为真。

3. !（非运算）

参与运算的量为真时，进行非运算，结果为假；参与运算的量为假时，进行非运算，结果为真。

例如：

```
!(5 > 0)
```

结果为假。

C 语言编译系统判断一个量是为真还是假时，以 0 代表假，以非 0 的值作为真。例如，5&&3 的值为真，即为 1，因为 5 和 3 均为非 0。又如 5||0 的值为真，即为 1。

当逻辑运算符两边表达式的值为不同的组合时，逻辑运算得到的结果不同，表 3-2 列出了逻辑运算的真值表。

表 3-2　逻辑运算的真值表

a	b	! a	! b	a&&b	a||b
真(1)	真(1)	假(0)	假(0)	真(1)	真(1)
真(1)	假(0)	假(0)	真(1)	假(0)	真(1)
假(0)	真(1)	真(1)	假(0)	假(0)	真(1)
假(0)	假(0)	真(1)	真(1)	假(0)	假(0)

【例 3-2】 分析逻辑运算程序的运行结果。

参考程序如下：

```
#include <stdio.h>
#include <math.h>
void main()
{
   char c = 'a';
   int i = 1,j = 2,k = 3;
   float x = 3.5,y = 0.85;
   printf("%d,%d\n",!x*!y,!!!x);
   printf("%d,%d\n",x||i&&j-3,i<j&&x<y);
   printf("%d,%d\n",i==5&&c&&(j=8),x+y||i+j+k);
}
```

分析：

(1) 对于第 2 个 printf 语句中的输出项 x||i && j－3，先计算 j－3 的值为非 0，再求 i && j－3 的逻辑值为 1，故 x||i&&j－3 的逻辑值为 1；对于输出项 i<j&&x<y，由于 i<j 的值为 1，而 x<y 为 0，故表达式的值为 1、0，进行与运算，最后输出结果为 0。

(2) 对于第 3 个 printf 语句中的输出项 i==5&&c&&(j=8)，由于 i==5 为假，即值为 0，该表达式由两个与运算组成，所以整个表达式的值为 0。对于式 x+ y||i+j+k，由于 x+y 的值为非 0，故整个或表达式的值为 1。

【谨记】 逻辑运算的短路特性。

(1) 短路特性。

逻辑表达式求解时，并非所有的逻辑运算都被执行，只有在必须执行下一个逻辑运算才能求出表达式的解时，才执行该逻辑运算。例如：

```
a&&b&&c      /* 只有 a 为真时，才判别 b 的值，只在 a、b 都为真时，才判别 c 的值 */
a||b||c      /* 只有 a 为假时，才判别 b 的值；只在 a、b 都为假时，才判别 c 的值 */
```

举例：若“a=1;b=2;c=3;d=4;m=1;n=1;”进行(m=a>b)&&(n=c>d)运算，则只执行 m=a>b，得到 m=0。而不执行 n=c>d，所以 n=1。

(2) 复杂逻辑运算。

例如，判别闰年的条件用复杂逻辑运算表示。

① 能被 4 整除但不能被 100 整除：(year % 4 == 0)&&(year % 100!= 0)

② 能被 400 整除：year % 400 == 0

综合起来：((year % 4 == 0)&&(year % 100!= 0))||year % 400 == 0

优化逻辑运算：(year % 4 == 0&&year % 100!= 0)||year % 400 == 0

3.4 if 语句

if 语句的功能是：根据给定的条件进行判断，以决定执行某个分支程序段。用 if 语句可以构成选择结构。

3.4.1　if 语句的三种形式

1. if 单分支结构(基本形式)

if(表达式) 语句

其语义是：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。if 单分支结构执行过程如图 3-2 所示。

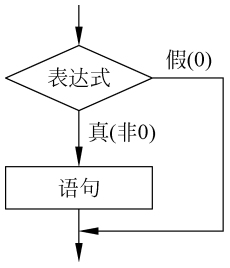


图 3-2　if 单分支结构执行过程

【例 3-3】 用 if 语句求两个整数中的较大者。

参考程序如下：

```
#include <stdio.h>
#include <math.h>
void main()
{
    int a,b,max;
    printf("input two numbers: \n ");
    scanf("%d%d",&a,&b);
    max=a;                    /*假定 a>b ,max=a*/
    if (max<b) max=b;         /*判定 max 与 b 的大小关系*/
    printf("max= %d",max); /*max 保存的值即 a 与 b 中的较大者*/
}
```

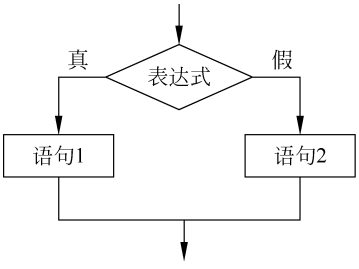


图 3-3　if 双分支结构的执行过程

2. if 双分支结构(常用形式)

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

if 双分支结构的语义：如果表达式的值为真,则执行语句 1,否则执行语句 2。

执行过程如图 3-3 所示。

【例 3-4】 用 if-else 语句改写例 3-3。

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

3. if 多分支结构(嵌套形式)

当有多个条件选择时,可用 if-else-if 语句,其一般形式为:

```
if(表达式 1)
     语句 1;
else if(表达式 2)
          语句 2;
     else if(表达式 3)
               语句 3;
                 …
          else if(表达式 n)
                    语句 n;
               else 语句 m;
```

其语义是:依次判断表达式的值,当出现某个值为真时,则执行其对应的语句;然后跳到整个 if 语句之外继续执行程序。如果所有的表达式均为假,则执行语句 m。继续执行后续程序。

if-else-if 语句的执行过程如图 3-4 所示。

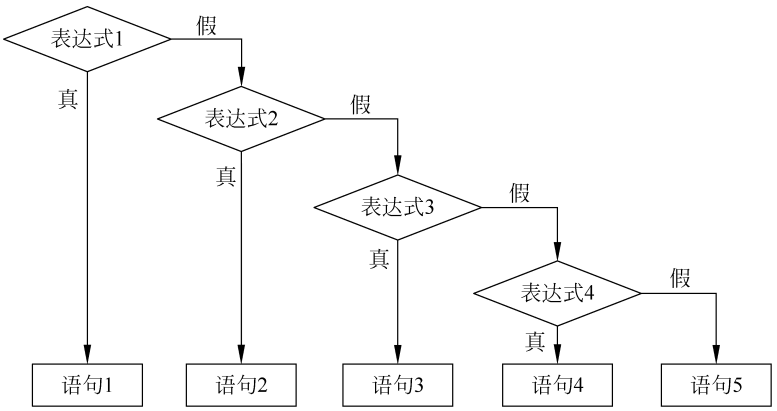


图 3-4 if-else-if 语句的执行过程

【例 3-5】 判定学生的成绩分数 85～100,75～84,60～74,0～59 分别对应的等级:优秀(A)、良好(B)、合格(C)、不及格(D)。要求从键盘输入百分制分数,用 if-else-if 语句实现对应的输出等级。例如,输入 71,输出显示“及格(C)”。

```
#include<stdio.h>
void main()
{
    int score;
    printf("input a student's score: ");
    scanf("%d",&score);
    if(score>100||score<0)
        printf("This is a invalid score\n");
    else if(score>=85)
```

```
            printf("优秀(A)\n");
        else if(score >= 75)
                printf("良好(B)\n");
            else if(score >= 60)
                    printf("及格(C)\n");
                else
                    printf("不及格(D)\n");
}
```

这是一个典型的多分支选择的问题，可以根据输入数字的大小判断输入数值所在的范围，分别输出不同的等级。

在 if 语句的三种形式中，if 条件后面通常为单个语句，如果要在满足条件时执行一组（多个）语句，则必须把这一组语句用{}括起来组成一个复合语句。注意在}之后不能再加分号。例如：

```
if(a > b)
    {a++;
    b++;}
else
    {a = 0;
     b = 10;}
```

【看一看】 if 表达式的类型。

if 关键字之后均为表达式，表达式类型任意。例如：

```
if(a = 5)   语句 1;     /* 表达式 a = 5 的值为非 0,所以执行其后的语句 1 */
if('b')     语句 2;     /* 若'b'的值永远为非 0,则执行其后的语句 2 */
if(5)       语句 3;     /* 5 的值为非 0,所以执行其后的语句 3 */
```

都是允许的。只要表达式的值为非 0，即为真，则执行 if 后面的语句。

又如，程序段：

```
if(a = b)
     printf("%d",a);
else
     printf("a = 0");
```

的语义是把 b 值赋予 a，若 a 为非 0 则输出该值，否则输出 a＝0。

3.4.2　if 语句的嵌套

当 if 语句中的执行语句又是 if 语句时，构成 if 语句嵌套的情形。

if 语句嵌套的一般形式为：

```
if(表达式)
    if 语句;
```

或者为

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

嵌套中的 if 语句可以是 if-else 型,这将会出现多个 if 和多个 else 的情况,这时要特别注意 if 和 else 的配对问题。

例如:

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

其中的 else 究竟是与哪一个 if 配对?

第一种应理解为:

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

第二种应理解为:

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

为了避免这种歧义,C 语言规定,else 总是与它前面最近的且尚未配对的 if 配对,因此对上述情况应按第一种情况理解。

【例 3-6】 用 if 语句的嵌套结构比较两个数的大小。

参考程序如下:

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

采用 if 嵌套结构比较两个数 a、b 的大小,实际上有 a>b、a<b 或 a=b 三种选择。因此,一般情况下较少使用 if 语句的嵌套结构,而是采用 if-else-if 语句完成,而且程序更加清晰。

【例 3-7】 用 if-else-if 语句改写例 3-6。

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

3.4.3 条件运算符和条件表达式

条件运算符?:是 C 语言中唯一的一个三目运算符,即有三个参与运算的量。

如果在条件语句中只执行单个的赋值语句,则使用条件表达式实现,能使程序简洁,提高运行效率。

条件表达式的一般形式为:

表达式 1? 表达式 2:表达式 3

其求值规则为:如果表达式 1 的值为真,则以表达式 2 的值作为整个条件表达式的值,否则以表达式 3 的值作为整个条件表达式的值。

条件表达式通常用于赋值语句中。例如:

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

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

可改写为 if 语句的等价形式:

```
if(a>b)  max=a;
   else max=b;
```

使用条件表达式时,应注意:

(1) 条件运算符? 和:是一对运算符,不能分开单独使用。

(2) 条件运算符的优先级低于关系运算符和算术运算符,高于赋值运算符。

因此

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

可以去掉括号而写为

```
max=a>b?a:b
```

(3) 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)
```

这是条件表达式嵌套的情形,即其中的表达式 3 又是一个条件表达式。读者可以用条件表达式改写例 3-6。

3.5 switch 语句

在 C 语言中,通常用 switch 语句实现多分支选择结构。

1. switch 语句形式

switch 语句的一般形式为:

```
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;
}
```

其语义是:计算表达式的值,并逐个与其后的常量表达式值相比较,当表达式的值与某个常量表达式的值相等时,即执行其后的语句,不再进行判断,继续执行剩下 case 后的所有语句。若表达式的值与所有 case 后的常量表达式均不相同时,则执行 default 语句。

2. 使用 switch 语句注意事项

(1) case 后只能是常量表达式,且各常量表达式的值不能相同,否则出现错误;

(2) 各 case 和 default 子句的先后顺序可以变动,不会影响程序执行结果;

(3) default 子句可以省略;

(4) case 后允许有多个语句,可以不用{}括起来;

(5) 多个 case 可共用一组执行语句。例如:

```
case 'A':
case 'B':
case 'C': printf("score > 60\n");
          break;
…
```

【例 3-8】 用 switch 语句改写例 3-5,判定分数等级的程序。

参考程序如下:

```
#include"stdio.h"
void main()
{   int a;
```

```
    printf("input a score: ");
    scanf("%d",&a);
    if(a>100||a<0)
       printf("a invalid score\n");
    else
       switch (a/10)
       {
          case 10 :
          case 9 :
          case 8 : printf("优秀(A)\n");
          case 7 : printf("良好(B)\n");
          case 6 : printf("及格(C)\n");
          default : printf("不及格(D)\n");;
       }
}
```

运行结果如图 3-5 所示。

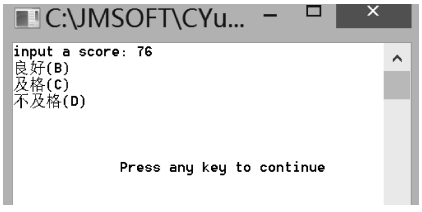


图 3-5　例 3-8 程序运行结果

程序要求输入一个分数(例如,输入 76),输出该分数对应的等级(良好)。但是,输入 76 之后,却执行了 case 7 及其之后的所有语句,显然不正确。

【想一想】 为什么会出现这种情况?如何改进程序?

switch 语句中的“case 常量表达式”只相当于一个语句标号,若表达式的值和某标号匹配则转向该标号执行,在执行完该标号语句后不能自动跳出 switch 语句,所以出现继续执行所有后面 case 语句的情况。

为了避免输出不应有的结果,在每一条 case 语句之后都增加 break 语句,使每一次执行之后均可跳出 switch 语句。添加 break 语句的程序代码如下:

```
#include"stdio.h"
void main()
{   int a;
    printf("input a score: ");
    scanf("%d",&a);
    if(a>100||a<0)
          printf("a invalid score\n");
    else switch (a/10)
          {  case 10: case 9:
             case 8 : printf("优秀\n"); break;
             case 7 : printf("良好\n"); break;
             case 6 : printf("及格\n"); break;
             default:printf("不及格\n");
          }
}
```

3. switch 语句嵌套

在 switch 语句中可以包含 switch 语句,构成 switch 语句的嵌套。

switch 嵌套程序举例如下:

```
#include <stdio.h>
void 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++; break;
      case 3: a++;b++;
   }
   printf("\na = %d,b = %d",a,b);
}
```

运行结果:a=2,b=1,请读者分析原因。

3.6 选择结构应用案例

根据工资收入情况,计算应缴个人所得税。

我国个税起征点调整历程如下。

1981 年,个人所得税正式开征,起征点为月收入 800 元,当时超过此工资收入数额的中国公民少而又少,大多数缴纳个人所得税的是外籍在华高级职员。

随着经济的发展和物价的提高,2005 年,调整个税起征点标准为 1600 元/月;2008 年,提高到 2000 元/月;2011 年,提高到 3500 元/月,专项扣除(如三险一金等)。

2018 年 10 月,个税起征点提高到 5000 元/月,专项扣除及专项附加扣除:如子女教育、继续教育、大病医疗、住房贷款利息或住房租金、赡养老人,以及允许劳务报酬、稿酬、特许权使用费等三类收入扣除 20%的费用后计算纳税。

个人所得税计算方法(公式)如下:

应缴个人所得税=(月工资-起征点-专项扣除-专项附加扣除)*税率-速算扣除数

个人所得税税率及速算扣除数如表 3-3 所示。

表 3-3 个人所得税税率及速算扣除数

级数	月应纳税所得额	税率	速算扣除数/元
1	不超过 3000 元的部分	3%	0
2	3000～12 000 元的部分	10%	105
3	12 000～25 000 元的部分	20%	555

续表

级数	月应纳税所得额	税率	速算扣除数/元
4	25 000～35 000 元的部分	25%	1005
5	35 000～55 000 元的部分	30%	2755
6	55 000～80 000 元的部分	35%	5505
7	超过 80 000 元的部分	45%	13 505

例如，某单位职工 A，2019 年 9 月份工资收入 16 000 元，个人缴纳的三险一金、住房贷款利息及赡养老人等合计 5680 元，则应纳税所得额＝16 000－5000－5680＝5320(元)。

应缴个人所得税＝5320＊20%－555＝509(元)。

当你 2023 年大学毕业参加工作，设想第一年的年薪 18 万元，个税起征点提高到 8000 元/月，三险一金、继续教育和住房租金等每月合计 4321 元，分别按 2008、2011、2018、2023 年的起征点，使用 if 语句或 switch 语句编程计算每月应缴个人所得税。

【查一查】 调研日常生活中电费、水费、煤气费、快递费的计算方法，尝试使用 if 语句或 switch 语句编写对应的计费程序。

3.7 答疑解惑

3.7.1 混合运算中的数据类型转换

疑问：在混合运算中，整型和浮点型数据可以相互转换吗？

解惑：通过如下程序说明。程序本意为 x＝0 则 y＝0，x!＝0 则 y＝1；但是，下列程序不论输出 x 的值是否为 0，y 都等于 0。

```
#include<stdio.h>
void main()
 {
     float x,y;
     scanf("%f",&x);
     if (x==0)
         y=0;
     else
       y=1;
     printf("%d",y);
}
```

如何修改程序，使输出结果符合要求？

解析：程序中定义 y 是一个浮点数，对应的输出语句应该用“printf("%f",y);”或者把已定义 y 为 float 改成 int，即 int y。

C 语言对于不同类型之间的数据转换：整型可以自动转换成浮点型；但是浮点型不能自动转换成整型，例如 x＝＝0 存在隐患，因此，不要对浮点数判断“等于”的操作。

3.7.2 if 语句的特点

疑问：下列程序运行后，不论输入什么数，都没有输出结果，原因何在？

程序如下：

```
#include <stdio.h>
void main()
{  float a;
   scanf("%f",&a);
   if (a == 123);
       printf("aaaa\n");
   else
       printf("bbbb\n");
 }
```

解惑：if 条件表达式后面不需要用分号，如果用分号表示 if 条件表达式结束，则其后面的语句都不会被执行。

if 语句的标准格式为：

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

3.7.3 switch 语句的易错点

疑问：初学者在运用 switch 语句时有哪些地方容易出错？

解惑：

(1) 在 switch 语句中有一个或多个 case 时，忘记加 break 语句；

(2) switch 语句中的 case 值不能是变量，只能是常量；

(3) case 各常量表达式的值不能相同，否则会出现错误。

知识点小结

本章介绍了选择结构程序设计的思想方法、关系运算符和关系表达式、逻辑运算符和逻辑表达式；条件运运算符“?:”，相当于根据不同情况对同一变量赋值的 if-else 语句的简写形式。阐述了 if 语句的三种选择结构：单分支 if 语句、双分支 if-else 语句、多分支 if-else-if 语句；if-else 语句在嵌套使用时应注意 else 和最近一个尚未被 else 匹配的 if 配对。多分支可用 switch 语句实现，其特点是根据一个条件表达式的多个不同值，进行多个分支结构的构造，且每个分支应用 break 语句结束，能够使程序的条理层次更加清晰。本章的重点是 if-else 语句、switch 语句，难点是 if 语句的嵌套。

习题 3

3.1　单选题

1. 设有定义“int a=1，b=2，c=3;”，以下语句执行结果与其他三个不同的是(　　)。

A. if(a>b)c=a,a=b,b=c;　　B. if(a>b){c=a,a=b,b=c;}

C. if(a>b)c=a;a=b;b=c;　　D. if(a>b){c=a;a=b;b=c;}

2. 以下程序段中，与语句“k=a>b?(b>c?1:0):0;”功能相同的是(　　)。

A.
```
if((a>b) && (b>c)) k=1;
  else k=0;
```

B.
```
if((a>b)||(b>c))k=1;
  else k=0;
```

C.
```
if(a<=b)k=0;
  else if(b<=c)k=1;
```

D.
```
if(a>b) k=1;
  else if(b>c)k=1;
  else k=0;
```

3. 以下选项中与“if(a==1)a=b; else a++;”语句功能不同的 switch 语句是(　　)。

A.
```
switch(a)
{ case 1:a=b;break;
  default : a++;
}
```

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

C.
```
switch(a)
{ default:a++;break;
  case 1:a=b;
}
```

D.
```
switch(a==1)
{   case 1:a=b;break;
    case 0:a++;
}
```

4. 有下列嵌套的 if 语句：

```
if(a<b)
  if(a<c) k=a;
  else  k=c;
else
  if(b<c) k=b;
  else  k=c;
```

与上述 if 语句等价的语句是(　　)。

A. k=(a<b)? a:b;k=(b<c)? b:c;

B. k=(a<b)? ((b<c)? a:b):((b>c)? b:c);

C. k=(a<b)? ((a<c)? a:c):((b<c)? b:c);

D. k=(a<b)? a:b;k=(a<c)? a:c;

5. 已有定义“char c;”，程序前面已在命令行中包含 ctype.h 文件，不能用于判断 c 中的字符是否为大写字母的表达式是(　　)。

A. isupper(c)　　B. 'A'<=c<='Z'

C. 'A'<=c&&c<='Z'　　D. c<=('z'-32)&&('a'-32)<=c

6. if 语句的基本形式为“if(表达式)语句”,其中“表达式”为(　　)。

A. 必须是逻辑表达式　　B. 必须是关系表达式

C. 必须是逻辑表达式或关系表达式　　D. 可以是任意合法的表达式

3.2 填空题

1. 设 x 为 int 型变量,写出一个关系表达式_____,用以判断 x 同时为 3 和 7 的倍数时,关系表达式的值为真。

2. 下列程序的功能是输出 a、b、c 三个变量中的最小值。在画线处填空。

```
#include <stdio.h>
void main( )
{
   int a,b,c,t1,t2;
   scanf("%d%d%d",&a,&b,&c);
   t1 = a < b?   _____;
   t2 = c < t1?  _____;
   printf("%d\n",t2);
}
```

3. 已有定义“char c=' ';int a=1, b;”(此处 c 的初值为空格字符),执行“b=! c &&a;”后 b 的值为_____。

4. 已知字母 A 的 ASCII 值为 65,若变量 kk 为 char 型,能正确判断 kk 的值为大写字母的表达式是_____ 。

5. 若变量已正确定义,有以下程序段:

```
int a = 3,b = 5,c = 7;
if(a > b)a = b;c = a;
if(c!= a)c = b;
printf("%d, %d, %d\n",a,b,c);
```

则输出结果是_____ 。

3.3 简答题

1. 简述选择结构程序设计的思想方法。

2. 在 C 语言中,条件如何表示?

3. 选择结构有哪些语句?简述其执行过程。

4. 简述分支语句的嵌套使用方法。

3.4 编程实战题

1. 输入一行字符,分别统计其中的英文字母、空格、数字和其他字符的个数。

2. 有一个分段函数如下,编程实现:输入 x 的值,输出对应的 y 值。

$$y=\begin{cases} x & (x<1) \\ 2x-1 & (1\leqslant x<10) \\ 3x-1 & (x\geqslant 10) \end{cases}$$

3. 给一个不多于 5 位的正整数,要求:

(1) 求出它是几位数;

(2) 分别打印每一位上的数字;

(3) 按逆序打印各位数字,例如原数为 321,应输出 123。

实验 3　选择结构程序设计

本次实验涉及逻辑运算符和逻辑表达式、关系运算符和关系表达式、条件运算符和条件表达式；if 语句和 switch 语句；选择结构程序设计。

【实验目的】

(1) 掌握逻辑运算符和逻辑表达式；
(2) 掌握关系运算符和关系表达式；
(3) 掌握条件运算符和条件表达式；
(4) 掌握 if 语句的三种形式：单分支 if 语句、双分支 if-else 语句和多分支 if-else-if 语句；
(5) 会用 switch 语句实现多分支控制语句；
(6) 掌握 break 语句在 switch 结构中的使用方法。

【实验内容】

一、基础题

1. 编程验证下列 if 语句是否正确,若有错误,分析错误的原因。

A.
```
if (x>0)
        printf("%f",x)
    else
        printf("%f",-x);
```

B.
```
if (x>0)
        { x=x+y; printf("%f",x);}
    else
        printf("%f",-x);
```

C.
```
if (x>0)
        {x=x+y;printf("%f",x);};
    else
        printf("%f",-x);
```

D.
```
if (x>0)
        {x=x+y;printf("%f",x)}
    else
        printf("%f",-x);
```

2. 输入三个整数,编程求出最大数和最小数。

提示：首先,比较两个数 a,b 的大小,并把大数存入 max,小数存入 min 中；然后,再与 c 比较,若 max 小于 c,则把 c 赋予 max,如果 c 小于 min,则把 c 赋予 min,因此保持 max 总是最大数,而 min 总是最小数；最后,输出 max 和 min 的值。

二、提高题

1. 用英文单词星期几的第一个字母来判断是星期几,如果第一个字母相同,则继续判断第二个字母,编程实现。

提示:使用多分支 switch 语句或 if 语句,若第一个字母相同,则用 case 语句或 if 语句判断第二个字母。

2. 编程输入一个整数,判断它能否被 3、5、7 整除,并输出以下信息之一:

(1) 能被其中一个数(要指出哪一个)整除;

(2) 能被其中两个数(要指出哪两个)整除;

(3) 能同时被 3、5、7 整除;

(4) 不能被 3、5、7 中任一个数整除。

3. 编程计算两个日期(日期格式:年-月-日)之间的天数。本题给出参考程序代码,要求上机运行,观察运行结果,写出算法。

参考程序如下:

```
/*求所在月份的天数*/
#include<stdio.h>
int daysMonth(int year,int month,int day)
  {
    int days[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};
    int i,sum=0;
    for(i=0;i<month;i++)
       sum+=days[i];
    if(month>2)  /*如果是闰年则2月加一天*/
        if((year%4==0)&&(year%100!=0)||(year%400==0))
            sum+=1;
    sum+=day;
    return sum;
  }
  /*交换位置,避免负值*/
  void swap(int x1,int x2)
  {
      int tmp=x1;
      x1=x2;
      x2=tmp;
  }
  long countDate(int y1,int m1,int d1,int y2,int m2,int d2)
  {
      int daysyear1,daysyear2;
      long totalDays=0;
      int total_day1;
      int tmpYear, tmpDays;
      printf("The first Date is %ld-%ld-%ld\n",y1,m1,d1);
      printf("The second Date is %ld-%ld-%ld\n",y2,m2,d2);
      if(y1>y2)
       {
         swap(y1,y2);
         swap(m1,m2);
```

```
        swap(d1,d2);
      }
    if(y1 == y2)
     {
        daysyear1 = daysMonth(y1,m1,d1);
        daysyear2 = daysMonth(y2,m2,d2);
        totalDays = abs(daysyear1 - daysyear2) + 1;
        printf("totalDays is %ld\n",totalDays - 1);
      }
    else
     {
        daysyear1 = daysMonth(y1,m1,d1);
        total_day1 = 365 - daysyear1 + 1;
        if(m1 <= 2)
         if(y1 % 4 == 0&&(y1 % 100!= 0)||(y1 % 400 == 0))
           total_day1 += 1;
        totalDays += total_day1;
        tmpYear = y1;
        while(++tmpYear < y2)
        {
            tmpDays = 365;
            if((tmpYear % 4 == 0)&&(tmpYear % 100!= 0)||(tmpYear % 400 == 0))
             tmpDays += 1;
            totalDays += tmpDays;
        }
       daysyear2 = daysMonth(y2,m2,d2);
       totalDays += daysyear2;
     }
    printf("totalDays is %ld\n",totalDays - 1);
}
void main()
{
    int y1,m1,d1,y2,m2,d2;
    printf("Plsase input Date,for example 2018 - 8 - 18,2019 - 9 - 19:\n");
    scanf("%ld - %ld - %ld,%ld - %ld - %ld",&y1,&m1,&d1,&y2,&m2,&d2);
    countDate(y1,m1,d1,y2,m2,d2);
}
```