第3章 选择结构

第1章和第2章已经介绍了顺序结构的程序设计，在大多数情况下，由于数据处理的需要，程序不会是单一的顺序结构，而是顺序、选择、循环3种结构的复杂组合。

在生活中，人们在做某件事情时经常要根据某些条件来做出不同的抉择，在程序设计中也需要通过对一些条件的判断来选择处理不同的数据或完成特定的功能，这就是条件语句的任务。本章介绍如何用C语言的条件语句实现选择结构。为此先介绍用来描述条件的关系和逻辑表达式，接着介绍实现选择结构的条件语句(if和switch)。if语句有简单选择结构、二路选择结构、多路选择结构这3种形式。if语句可以通过嵌套形式实现复杂的选择结构。switch语句能方便地实现更简明的多路选择。

视频讲解

3.1 工程师岗位面试(关系运算符和逻辑运算符)

上海一家IT企业招聘C/C++开发工程师，满足以下任职要求者可以获得面试机会。

(1) 学历要求：计算机相关专业毕业，硕士及以上学历。

(2) 工作年限：具有两年以上C/C++开发工程师岗位从业经验。

(3) 业务要求：能熟练使用或精通C/C++和数据压缩算法。

3.1.1 分析与设计

这家企业招聘C/C++开发工程师的条件和描述这些条件的表达式如表3.1所示。

表3.1 任职要求和对应的表达式

招聘条件	表达式
硕士、博士	eduction=='M' ‖ eduction=='D'
在C/C++开发工程师岗位从业两年以上	working_life>2
对C/C++和数据压缩算法能熟练使用或精通	skill_level>2 && skill_level<=4

学历：G.本科 M.硕士 D.博士；工作年限：用整型数表示；熟练程度：1.初学 2.一般 3.熟练 4.精通

满足以上3个条件，即相应表达式的值为真，那么求职者就可以接到面试通知。从以上面试条件中可以观察到，有的条件涉及多个判断(子条件)，而这些判断之间存在着一定的逻辑关系，后面介绍的关系运算和逻辑运算就用来描述它们的逻辑关系。另外，程序中的if语句用于控制求职者是否满足面试条件，并输出相应的结果。

例 3-1 谁会得到面试机会。

```
#include "stdio.h"
void main()
{
    char eduction;
    int working_life, skill_level;
    printf("_______求职信息_______\n");
    printf("学历(G.本科 M.硕士 D.博士): ");
    scanf("%c",&eduction);
    if(eduction == 'M' || eduction == 'D')                    /*筛查学历*/
    {
        printf("C/C++开发工程师任职年限: ");
        scanf("%d", &working_life);
        if(working_life > 2)                                  /*筛查工作年限*/
        {
            printf("数据压缩算法(1.初学 2.一般 3.熟练 4.精通):");
            scanf("%d",&skill_level);
            if(skill_level > 2&&skill_level <= 4)             /*筛查业务能力*/
                printf("恭喜你获得面试机会!\n");              /*符合所有条件*/
            else
                printf("抱歉,你熟练程度不够!\n");
        }
        else
            printf("抱歉,你任职年限不够!\n");
    }
    else
          printf("抱歉,你学历尚浅!\n");
}
```

运行结果：

```
_______求职信息_______
学历(G.本科 M.硕士 D.博士):M
C/C++开发工程师任职年限:3
数据压缩算法(1.初学 2.一般 3.熟练 4.精通):4
恭喜你获得面试机会!
```

程序运行后的显示效果如图 3.1 所示。

图 3.1 程序运行结果

3.1.2 关系运算符和关系表达式

关系表达式是由关系运算符、括号、常量、变量和函数等运算对象构成的式子。在 C 语

言中关系运算符共有 6 种，如表 3.2 所示。

表 3.2　关系运算符

关系运算符	含　义	优 先 级	结 合 性
<	小于	这几个运算符优先级相同，但比下面高	左结合
<=	小于或等于		
>	大于		
>=	大于或等于		
==	等于	这两个运算符优先级相同，但比上面低	
!=	不等于		

通过关系运算符可以写出关系表达式，关系表达式的一般形式如下：

表达式　关系运算符　表达式

例 3-1 中出现的 working_life>2 就是关系表达式。由于关系运算符的优先级低于算术运算符，高于赋值运算符，所以下面的关系表达式有其相应的等价式子。

$a>b+c$　　等价于　$a>(b+c)$

$a>b==c$　等价于　$(a>b)==c$

$a=b>c$　　等价于　$a=(b>c)$

关系运算实际上是比较运算。关系表达式的值是一个逻辑值，即“真”或“假”。在 C 语言中没有专用的逻辑型数据，因此常用整型变量来存放关系的运算结果。若关系表达式成立，则该表达式的值为“真”，用整数“1”表示；若关系表达式不成立，则该表达式的值为“假”，用整数“0”表示。例如，关系表达式 2==3 的值为 0，5>3 的值为 1，(a=3)>(b=5) 的值为 0。

3.1.3　逻辑运算符和逻辑表达式

逻辑表达式是由逻辑运算符、括号和其他运算对象构成的式子，它的值反映了逻辑运算的结果。C 语言中的逻辑运算符有 3 种，如表 3.3 所示。

表 3.3　逻辑运算符

逻辑运算符	含　义	优 先 级	结 合 性
!	逻辑非，单目	高于 && 和 ||	右结合
&&	逻辑与，双目	高于 ||	左结合
||	逻辑或，双目	低于 &&	左结合

通过逻辑运算符可以写出逻辑表达式，逻辑表达式的一般形式如下：

表达式　逻辑运算符　表达式　或　逻辑运算符　表达式

例如，a || b 和 age>25 && sex=='f'都是逻辑表达式。

运算符“&&”和“||”的优先级低于算术运算符和关系运算符，而运算符“!”的优先级高于算术运算符和关系运算符。在例 3-1 中用 3 个逻辑表达式来选择符合条件的求职者。3 种逻辑运算符的运算规则如下。

（1）$a\&\&b$：当 a 和 b 都为真时结果为真，否则结果为假。

（2）$a \| b$：当 a 和 b 至少一个为真时结果为真，当 a 和 b 都为假时结果为假。

（3）$!a$：当 a 为真时结果为假，当 a 为假时结果为真。

表 3.4 为逻辑运算的真值表，它表示 a 和 b 的值在不同组合下各逻辑运算的值。

表 3.4　逻辑运算的真值表

a	b	$!a$	$!b$	$a\&\&b$	$a \| b$
真	真	假	假	真	真
真	假	假	真	假	真
假	真	真	假	假	真
假	假	真	真	假	假

例 3-2　设 $a=4$，$b=5$，$c=5$，写出下面各逻辑表达式的值。

$a+b<c\&\&b==c$　　值为 0

$a \| b+c\&\&b-c$　　值为 1

$!(a>b)\&\&!c \| 1$　　值为 1

$!(x=a)\&\&(y=b)\&\&0$　　值为 0

$!(a+b)+c-1\&\&b+c/2$　　值为 1

说明：

（1）逻辑表达式中任何非零的数值都被作为“真”。

（2）在逻辑表达式的求解中，并不是所有的逻辑运算符都会被执行。

① $a\&\&b\&\&c$：只有 a 为真时才需要判断 b 的值，只有 a 和 b 都为真时才需要判断 c 的值。

② $a \| b \| c$：只要 a 为真，就不必判断 b 和 c 的值，只有 a 为假时才需要判断 b。a 和 b 都为假时才判断 c。

当左值可以决定整个表达式的值时就不再求右边表达式的值，因此对于与（&&）运算来说，左值为 0，就不再继续后面的运算；对于或（||）运算来说，左值为 1，就不再继续后面的运算，这样可以提高运行速度，有的书中把这一特点叫作“逻辑运算的短路特性”，例如：

$$(m=a>b)\&\&(n=c>d)$$

当 $a=1$，$b=2$，$c=3$，$d=4$，m 和 n 的原值为 1 时，由于“$a>b$”的值为 0，因此 $m=0$，即可判断出表达式 $(m=a>b)\&\&(n=c>d)$ 的值为 0，而不必再求“$n=c>d$”的值，因此 n 的值不是 0 而仍保持原值 1。

3.1.4　条件运算符和条件表达式

条件表达式是用条件运算符（?:）把 3 个表达式连接起来的式子，其形式如下：

表达式 1?表达式 2：表达式 3

条件运算符（?:）要求 3 个操作数，所以也称为三目运算符。条件表达式的求解过程是先判断表达式 1 的值是否为真（非 0），若为真，则计算表达式 2，表达式 2 的值就是整个条件表达式的值；若为假（0），则计算表达式 3，表达式 3 的值就是整个条件表达式的值。

例 3-3 输入学生的卷面成绩和平时成绩，判断学生的总评成绩是否及格（总评成绩＝卷面成绩×60%＋平时成绩×40%，且总评成绩 60 分以上为及格）。

```
#include "stdio.h"
void main()
{
    int g1,g2,grade;
    printf("请输入卷面成绩和平时成绩: ");
    scanf("%d%d",&g1,&g2);     /*不能打负分*/
    grade=(int)(g1*0.6+g2*0.4);
    printf("考试结果: %s,%d\n", grade>=60 ? "及格":"不及格",grade);
}
```

运行结果：

```
请输入卷面成绩和平时成绩: 85  80
考试结果: 及格,83
```

例 3-4 将输入的字符由大写字母转换成小写字母。

分析：使用条件表达式“ch＝(ch>＝'A'&&ch<＝'Z')？(ch＋32):ch”来判断是否为大写字母，如果字符变量 ch 的值为大写字母，则条件表达式的值为“ch＋32”，即相应的小写字母，32 是小写字母和大写字母 ASCII 码的差值。如果 ch 的值不是大写字母，则条件表达式的值为 ch，即不进行转换。

```
#include "stdio.h"
void main ()
{
    char ch;
    printf("请输入一个字符: ");
    scanf("%c",&ch);                          /*输入一个字符*/
    ch=(ch>='A'&&ch<='Z')?(ch+32):ch;         /*将输入的大写字母转换为小写字母*/
    printf("%c\n",ch);                        /*输出最后得到的字符*/
}
```

运行结果如下：

```
请输入一个字符: a
a
```

请注意条件运算符的结合性为右结合。例如“int a＝1,b＝4,c＝3,d＝2;”，条件表达式 a<b?a:c<d?c:d 的值是什么？由于条件运算符的右结合性，a<b?a:c<d?c:d 相当于 a<b?a:(c<d?c:d)，因为 a<b 为真，所以整个条件表达式的值就为 a 的值 1。如果 a<b 为假，那么再去计算后面的(c<d?c:d)，这里 a<b 为真，因此就不必计算后面的条件表达式(c<d?c:d)。

3.2 判断身材是否标准（if 语句）

输入某人的身高和体重，按照下面的方法确定此人的体重是标准、过胖还是过瘦。

(1) 标准体重＝(身高－110)kg。

（2）超过标准体重 5kg 为过胖。

（3）低于标准体重 5kg 为过瘦。

3.2.1 分析与设计

视频讲解

在设计该程序的代码时，使用 weight 代表体重、height 代表身高，因此标准体重计算方法为 height－110。若某人的体重超过标准体重 5kg，即 weight＞height－105，那么表明此人超重；若某人的体重低于标准体重 5kg，即 weight＜height－115，那么表明此人过瘦；如果某人的体重为 height－105≤weight≤ height－115，表明此人的身材标准。

例 3-5 你的身材标准吗？

```
#include "stdio.h"
void main()
{
    float height,weight;                        /*height 为身高变量,weight 为体重变量*/
    printf("请输入你的身高和体重: ");
    scanf("%f,%f",&height,&weight);
    if(height>0&&weight>0)                      /*height、weight 不可为负数*/
    {
        if(weight>height-105)                   /*判断是超重、标准还是偏轻*/
            printf("你的体重超重,请注意减肥\n");
        else if(weight<height-115)
            printf("你的体重偏轻,请注意营养\n");
        else
            printf("你的体重刚好,请保持\n");
    }
    else
        printf("输入的数据不合格\n");
}
```

运行结果 1 如下：

```
请输入你的身高和体重: 165,50
你的体重刚好,请保持
```

运行结果 2 如下：

```
请输入你的身高和体重: 160,60
你的体重超重,请注意减肥
```

运行结果 3 如下：

```
请输入你的身高和体重: 175,50
你的体重偏轻,请注意营养
```

3.2.2 if 语句

视频讲解

if 语句也称为条件语句，用于实现程序的选择结构。if 语句通过判断给定的条件（一般是关系表达式或逻辑表达式）是否成立来控制执行不同的程序语句，完成相应的功能。if 语句有 3 种语法形式，构成了 3 种选择结构。

1. 简单选择结构

简单选择结构也称为单分支结构，语句形式如下：

```
if(表达式)
  语句;
```

该语句的功能是如果表达式的值为真(非 0 值)，则执行语句；如果表达式的值为假(0 值)，则跳过该语句继续执行后续程序。其执行过程如图 3.2 所示。

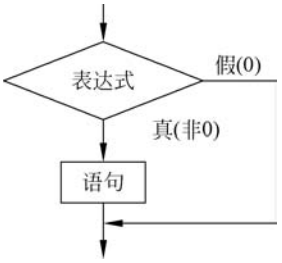

图 3.2　简单选择结构的执行过程

例如：

```
if(x>y)
    printf("%d\n",x);
```

说明：if(表达式)这行后面不要添分号，因为在这里 if 关键字这行和其后面一行是联系在一起而构成一条完整的语句，加分号就把该语句变为两条语句了。

例如：

```
if(x>y);                    /*这是一条语句,";"是语句结束符*/
    printf("%d",x);         /*这是另一条语句*/
```

例 3-6　超市在打折促销商品，若购物金额在 500～1000 元打 8 折，超过 1000 元打 7 折，低于 500 元不打折，编程序输入购物金额，计算并输出折扣金额。

分析：首先设购物金额为 total，按照题目要求打折范围被划分为 3 段，即若 0＜total＜500，不打折；若 500≤total≤1000，打 8 折；若 total＞1000，打 7 折。

```
#include "stdio.h"
void main ()
{
    float total;                                    /*total 为购物金额*/
    printf("请输入购物金额: ");
    scanf("%f",&total);                             /*输入购物金额*/
    if(0<total && total<500)
        printf("折扣后金额: %8.2f\n", total);       /*无折扣*/
    if(500<= total && total<=1000)
        printf("折扣后金额: %8.2f\n", total*0.8);   /*打 8 折*/
    if(total>1000)
        printf("折扣后金额: %8.2f\n", total*0.7);   /*打 7 折*/
    if(total<=0)
        printf("数据非法\n");
}
```

运行结果 1 如下：

```
请输入购物金额: 400
折扣后金额: 400.00
```

运行结果 2 如下：

```
请输入购物金额：800
折扣后金额：640.00
```

运行结果 3 如下：

```
请输入购物金额：2000
折扣后金额：1400.00
```

2. 二路选择结构

二路选择结构也称为双分支结构，语句形式如下：

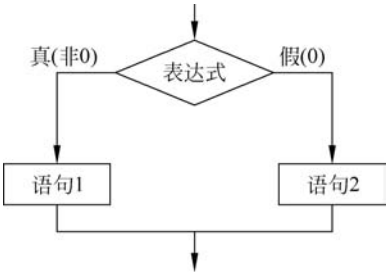

图 3.3　二路选择结构的执行过程

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

该语句的功能是如果表达式的值为真(非 0 值)，则执行语句 1，否则执行语句 2。其执行过程如图 3.3 所示。

说明：语句 1 和语句 2 可以是一个语句也可以是复合语句(用大括号括起来的多个语句)。

例 3-7　目前，中国男性的就业年龄为 16～60 岁，输入某人的年龄，判断他是否适合就业。

```
#include "stdio.h"
void main ()
{
    int age;                                          /* age 为年龄 */
    printf("请输入你的年龄：");
    scanf("%d",&age);                                 /* 输入年龄,正数 */
    if(age<16)
        printf("你还太年轻,不宜工作\n");
    else
    {
        if(16<=age && age<=60)
            printf("你适合就业\n");
        else
            printf("你可以安享退休生活了\n");
    }
}
```

运行结果 1 如下：

```
请输入你的年龄：15
你还太年轻,不宜工作
```

运行结果 2 如下：

```
请输入你的年龄：20
你适合就业
```

运行结果 3 如下：

```
请输入你的年龄：61
你可以安享退休生活了
```

例 3-8 输入年份，判断该年是否为闰年，若是闰年则输出“Y”，否则输出“N”。

分析：若年份能被 4 整除但不能被 100 整除，则该年是闰年；年号能被 400 整除也是闰年。由此，判断闰年的逻辑表达式如下：

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

```
#include "stdio.h"
void main()
{
    int year;
    printf("请输入一个年份：");                          /* 输入年份到变量 year 中 */
    scanf("%d",&year);
    if((year % 4 == 0&&year % 100!= 0) || year % 400 == 0)
        printf("Y\n");                                   /* 是闰年 */
    else
        printf("N\n");                                   /* 不是闰年 */
}
```

运行结果 1 如下：

```
请输入一个年份：2011
N
```

运行结果 2 如下：

```
请输入一个年份：2004
Y
```

3. 多路选择结构

多路选择结构也称为多分支结构，语句形式如下：

```
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 m)
    语句 m;
[else
    语句 n; ]
```

该语句将从上至下依次判断各表达式的值，当出现某个表达式的值为真时则执行其对应的语句，然后跳到整个 if 语句之外继续执行后续程序。如果所有的表达式值均为假，则执行 else 后的语句 n，接着继续执行后续程序。其执行过程如图 3.4 所示。

例如：

```
if (number > 500)
    cost = 0.15;
else if(number > 300)
    cost = 0.10;
else if(number > 100)
    cost = 0.075;
else if(number > 50)
    cost = 0.05;
else
    cost = 0;
```

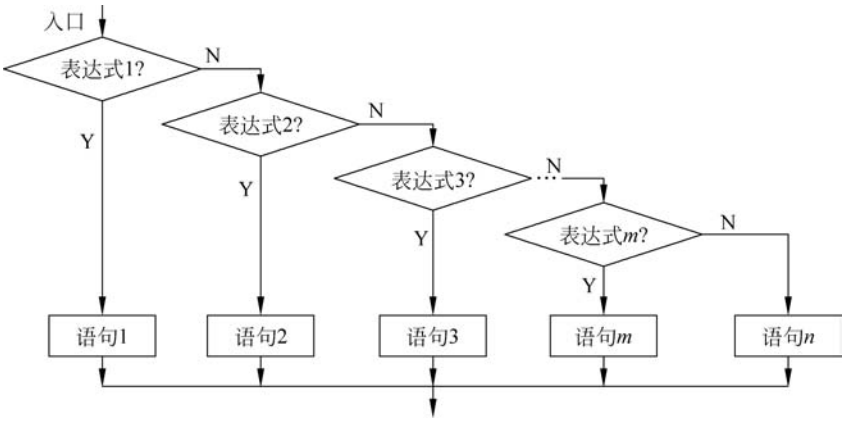

图 3.4　多路选择结构的执行过程

例 3-9　某公司对销售人员的工资实行底薪 1000 元加月销售额提成的政策，销售额与提成的比例关系如表 3.5 所示(单位：元)。

表 3.5　销售额与提成的比例关系表

销售额 sales	提 成 比 例	销售额 sales	提 成 比 例
sales＜10000	没有提成	50000≤sales＜100000	3%
10000≤sales＜50000	2%	sales≥100000	4%

分析：

(1) 定义变量 sales 用来存放销售人员当月的销售额。

(2) 提示用户输入销售人员当月的销售额，并用 scanf()函数接受用户输入的销售人员当月的销售额。

(3) 根据表 3.5 的规则计算销售人员当月的工资(底薪＋销售额×提成比例)，并输出。

```
#include "stdio.h"
void main()
{
    const float B_PAY = 1000.0;                    /*定义底薪*/
    float sales,income;                            /*定义销售额、工资*/
    printf("请输入销售人员的销售额: ");
```

```
    scanf("%f",&sales);
    if(sales<0)
    {
        printf("输入的数据非法\n");
        return;
    }
    if(sales<10000)
        income= B_PAY;
    else if(sales<50000)
        income= B_PAY +sales*(float)0.02;
    else if(sales<100000)
        income= B_PAY +sales*(float)0.03;
    else
        income= B_PAY +sales*(float)0.04;
    printf("本月工资= %.2f \n ", income);
}
```

运行结果 1 如下：

```
请输入销售人员的销售额: -1000
输入的数据非法
```

运行结果 2 如下：

```
请输入销售人员的销售额: 5000
本月工资=1000.00
```

运行结果 3 如下：

```
请输入销售人员的销售额: 20000
本月工资=1400.00
```

运行结果 4 如下：

```
请输入销售人员的销售额: 60000
本月工资=2800.00
```

运行结果 5 如下：

```
请输入销售人员的销售额: 200000
本月工资=9000.00
```

说明：

(1) 多路选择结构为多选一的结构。

(2) 如果所有条件均不成立，也不需要完成任何操作，则可以省略 else 子句。

对 if 语句的几点说明如下：

(1) 在 if 语句的 3 种形式中，在 if 后面都有表达式，一般为混合型表达式，有时为单纯的逻辑表达式、关系表达式或算术表达式。

(2) 在第 2、第 3 种形式的 if 语句中，在每个 else 前面有一个分号，整个语句结束处有一个分号。

(3) 在 if 和 else 后面可以只含有一个内嵌的操作语句，也可以有多个操作语句，但要用

大括号将几个语句括起来成为一个复合语句，此时不用加分号。

3.2.3 if 语句的嵌套

3.2.2 节讲解的 3 种形式是 if 语句的基本形式，如果在 if 语句中又包含一个或多个 if 语句则称为 if 语句的嵌套。其一般形式如下：

```
if(表达式)                                    /* 外层 */
    if 语句;                                  /* 内层 */
```

或者

```
if(表达式)                                    /* 外层 */
    if 语句;                                  /* 内层 */
else
    if 语句;                                  /* 内层 */
```

例 3-10 学校进行成绩分级管理，取消分数制，改为成绩分级评定。具体办法是小于 60 分为 E 类；60～70 分（不含 70 分）为 D 类；70～80 分（不含 80 分）为 C 类；80～90 分（不含 90 分）为 B 类；90 分以上为 A 类。设计一个程序，对输入的成绩进行等级划分。

分析：如果用多分支 if 语句进行判断，过程如下。

（1）看输入的成绩是否小于 60 分，若是则为 E 类，否则转（2）。

（2）看输入的成绩是否小于 70 分，若是则为 D 类，否则转（3）。

（3）看输入的成绩是否小于 80 分，若是则为 C 类，否则转（4）。

（4）看输入的成绩是否小于 90 分，若是则为 B 类，否则转（5）。

（5）该成绩为 A 类。

应用第 3 种 if 语句的执行过程如图 3.5 所示。

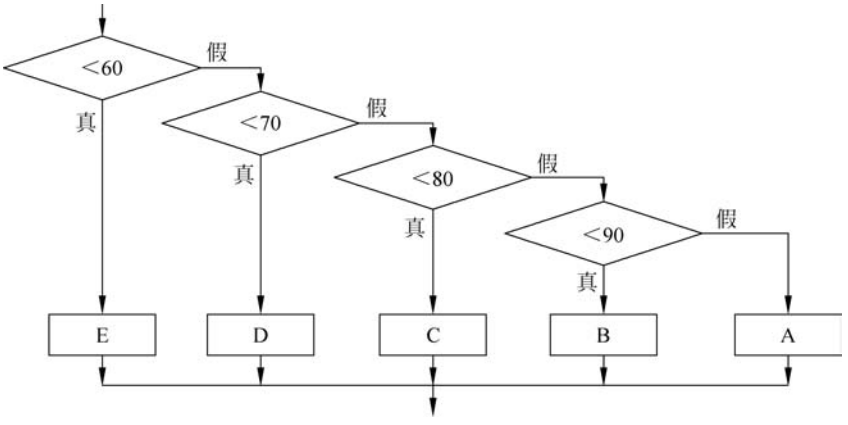

图 3.5 应用第 3 种 if 语句的执行过程

通过该流程图可以看出，如果一个学生的成绩为 95 分，那么至少要在该程序中进行 4 次比较才能得到最终结果。倘若同学们的成绩普遍都很好，那么用这个程序进行成绩等级划分的效率是不高的，因为它最多要比较 4 次才能得到最终结果。如果应用 if 语句的嵌套，就可以减少比较次数，从而提高程序的效率，应用 if 嵌套语句的执行过程如图 3.6 所示。

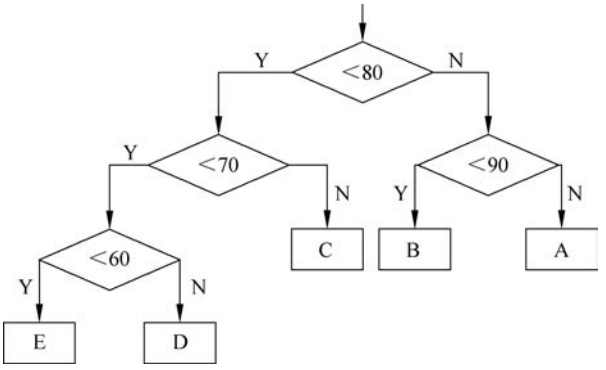

图 3.6　应用 if 嵌套语句的执行过程

从该流程图中不难看出，应用 if 嵌套语句可以减少程序的平均比较次数。因为它最多只要比较 3 次就能得到最终结果。同样是成绩普遍很好的同学，应用这个程序来进行成绩等级划分的效率要比图 3.5 所示的效率高。如果在大型系统中，需要比较的次数更多，那么这个程序的优势将更加明显。具体代码如下：

```
#include "stdio.h"
void main()
{
    int score;
    printf("请输入成绩: ");
    scanf("%d",&score);                    /*输入成绩存储到变量 score 中*/
    if(score<80)                           /*判断成绩是否小于 80 分*/
        if(score<70)                       /*判断成绩是否小于 70 分*/
            if(score<60)                   /*判断成绩是否小于 60 分*/
                printf("E\n");             /*如果成绩小于 60 分,则对应等级为 E*/
            else
                printf("D\n");             /*如果成绩小于 70 分,则对应等级为 D*/
        else
            printf("C\n");                 /*如果成绩小于 80 分,则对应等级为 C*/
    else                                   /*判断成绩是否小于 90 分*/
        if(score<90)                       /*如果成绩小于 90 分,则对应等级为 B*/
            printf("B\n");
        else                               /*如果成绩不小于 90 分,则对应等级为 A*/
            printf("A\n");
}
```

运行结果 1 如下：

```
请输入成绩: 59
E
```

运行结果 2 如下：

```
请输入成绩: 68
D
```

运行结果 3 如下：

```
请输入成绩: 72
C
```

运行结果 4 如下：

```
请输入成绩: 86
B
```

运行结果 5 如下：

```
请输入成绩: 93
A
```

在 if 嵌套语句结构中要注意 if 和 else 的配对，配对原则为 else 总是与它上面最近的同一复合语句中的未配对的 if 语句配对。以下程序段得到的配对关系为第①个 if 和第③个 else 配对；第②个 if 和第②个 else 配对；第③个 if 和第①个 else 配对；第④个 if 和第④个 else 配对。

```
①if(score<80)
    ②if(score<70)
        ③if(score<60)
            printf("E\n");
        ①else
            printf("D\n");
        ②else
            printf("C\n");
③else
④if(score<90)
    printf("B\n");
④else
    printf("A\n");
```

注意：配对时将从第一个 else 开始向上寻找与其配对的 if 语句。

3.2.4 实战演练

某邮局对邮寄包裹有以下规定：若包裹的长、宽、高任一尺寸超过 1m 或质量超过 30kg，不予邮寄；对可以邮寄的包裹每件收取手续费 0.5 元，再加上根据表 3.6 按质量 weight 计算的邮资，请编写程序计算包裹的邮寄资费。

表 3.6 按质量计算的邮资标准

质量(kg)	收费标准(元/kg)
weight≤10	1.00
10<weight≤20	0.90
20<weight≤30	0.80

请读者根据题意将程序补充完整。

```
#include "stdio.h"
void main()
{
    float length,width,height,weight,postage,r ; /* length、height、width 分别为长、高、宽;
weight 为包裹的质量; postage 为应付邮资; r 为能否邮寄的标志变量 */
    printf("请输入包裹的长、宽、高、质量: ");
    scanf("%f%f%f%f",&length,&width,&height,&weight);
```

```
    if(________)                    /* 长、宽、高任一尺寸超过 1m 或质量超过 30kg,不予邮寄 */
        r = -1;
    else if(________)               /* 如果质量<=10kg,收费标准为 1.00 元/kg */
        r = 1.00;
    else if(________)               /* 如果质量<=20kg,收费标准为 0.90 元/kg */
        r = 0.90;
    else if(________)               /* 如果质量<=30kg,收费标准为 0.80 元/kg */
        r = 0.80;
    if(________)                    /* 如果 r == -1,不予邮寄 */
        printf("包裹不符合要求,不予邮寄。\n");
    else                            /* 计算所付邮资 */
    {
        postage = ________;
        printf("你应该支付的邮资为: %2f 元.\n",postage);
    }
}
```

运行结果如下：

```
请输入包裹的长、宽、高、质量: 0.50  0.80  1.00  6.00
你应该支付的邮资为: 6.50 元
```

3.3　顾客点餐(switch 语句)

视频讲解

快餐店推出新套餐系列供顾客点餐，可供选择的套餐有 4 种：A 餐，超值全家桶＋大薯，78.00 元；B 餐，超值全家桶＋大鸡米花，85.00 元；C 餐，超值全家桶＋新奥尔良烤鸡腿堡，83.50 元；D 餐，超值全家桶＋葡式蛋挞或黄桃蛋挞，93.00 元。请编程实现顾客点餐和价格结算功能。

3.3.1　分析与设计

程序可以先显示 4 种套餐的价目表，然后接收从键盘输入的套餐号，确认输入无误再接收点餐套数，根据套餐号确定相应的单价，然后通过单价和数量计算顾客需支付的金额并输出。如果输入的数据不符合要求，则应给出提示。该题目可以用 3.2 节介绍的多路选择结构实现，但用 switch 语句同样可以实现多路选择，而且更加简洁、易读。

例 3-11　顾客点快餐。

```
#include "stdio.h"
void main()
{
    char meal;
    int n = 0;
    float price;
    printf("     套餐及价目表\n");              /* 显示价目表 */
    printf("A 餐,超值全家桶 + 大薯,78.00 元\n");
    printf("B 餐,超值全家桶 + 大鸡米花,85.00 元\n");
    printf("C 餐,超值全家桶 + 新奥尔良烤鸡腿堡,83.50 元\n");
    printf("D 餐,超值全家桶 + 葡式蛋挞/黄桃蛋挞,93.00 元\n");
```

```
    printf("请输入套餐种类: ");
    scanf("%c",&meal);                                  /*从键盘输入套餐种类*/
    if(meal>='A'&&meal<='D')
    {
        switch(meal)                                    /*根据套餐种类确定单价*/
        {
        case 'A': price=78.00;
            break;
        case 'B': price=85.00;
            break;
        case 'C': price=83.50;
            break;
        case 'D': price=93.00;
            break;
        }
        printf("请输入点餐套数: ");
        scanf("%d",&n);                                 /*从键盘输入点餐套数*/
        if(n<=0)
            printf("点餐数量不对!\n");
        else
            printf("请支付%.2f元\n",price*n);           /*计算支付金额并输出*/
    }
    else
        printf("只有以上4种套餐,请输入A～D\n");

}
```

运行结果：

```
          套餐及价目表
A餐,超值全家桶+大薯,78.00元
B餐,超值全家桶+大鸡米花,85.00元
C餐,超值全家桶+新奥尔良烤鸡腿堡,83.50元
D餐,超值全家桶+葡式蛋挞/黄桃蛋挞,93.00元
请输入套餐种类: A
请输入点餐套数: 2
请支付156.00元
```

程序运行效果如图 3.7 所示。

图 3.7　顾客点餐结果

当然还可以用数组实现更灵活的设计，在学习第 5 章后大家可以尝试一下。

3.3.2 switch 语句

switch 语句的一般形式如下：

```
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;
…
case 常量表达式 n: 语句 n;
                   break;
[default:语句 n + 1;]
}
```

switch 语句的功能是先计算 switch 后面的表达式的值，再依次与 n 个常量表达式的值进行比较，当表达式的值与某个 case 后的常量表达式的值相等时执行该 case 后的语句，然后执行 break 语句跳出 switch 语句。如果所有常量表达式的值都不等于 switch 后面表达式的值，则执行 default 后的语句。例如：

```
switch(grade)
{
case 'A ': printf("85～100\n");
       break;
case 'B ': printf ("70～84\n");
       break;
case 'C ': printf ("60～69\n");
       break;
case 'D ': printf ("< 60\n");
       break;
default: printf ("error\n");
}
```

switch 语句的执行过程如图 3.8 所示。

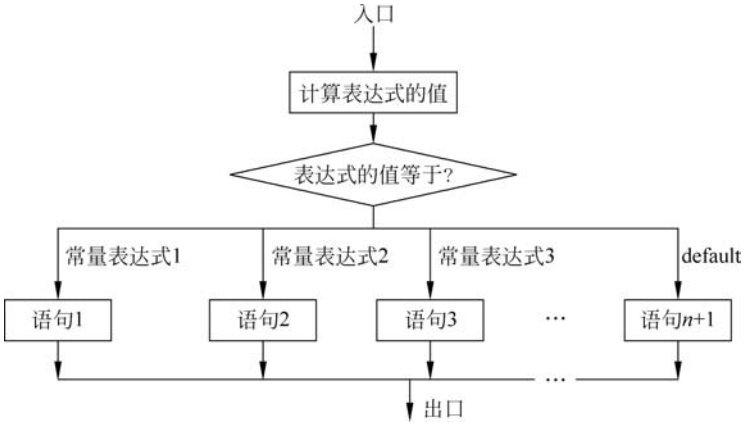

图 3.8 switch 语句的执行过程

注意：在每一个 case 中都应使用 break 语句提供一个出口，使流程跳出 switch 语句。若没有使用 break 语句，则从第一个满足条件的 case 后面的语句起所有 case 后的语句都会被执行，直至遇到 break 语句，这种情况叫作落空。请看下面无 break 语句的例子(例 3-13)。

例 3-12 某足球队按照运动员的体重配给含蛋白质和脂肪食物，标准为：体重 50～69kg 配高蛋白、中度脂肪食物；70～89kg 配高蛋白、低脂肪食物；90～120kg 只配高蛋白食物。编程为输入的运动员体重选择符合标准的食物配给方案。

```
#include "stdio.h"
void main()
{
    int w;
    printf("请输入运动员的体重: ");
    scanf("%d",&w);
    if(w<50 || w>120)
        printf("体重不符合标准,转行吧!\n");
    else
    {
        switch(w/10)
        {
        case 5:
        case 6:printf("配高蛋白、中度脂肪食物\n");break;
        case 7:
        case 8:printf("配高蛋白、低脂肪食物\n");break;
        default:printf("只配高蛋白食物\n");
        }
    }
}
```

例 3-13 case 后无 break 语句的例子(落空)。

```
#include "stdio.h"
void main()
{
    int c=8;
    switch(c<10?1:c<25?2:c<35?3:4)      /*通过条件表达式确定 switch 表达式的值*/
    {
    case 1: printf("%d℃ 有点冷\n",c);
    case 2: printf("%d℃ 正合适\n",c);
    case 3: printf("%d℃ 有点热\n",c);
    default: printf("%d℃ 太热了\n",c);
    }
}
```

c 的值为 8，则计算 switch 后的条件表达式 $c<10?1:c<25?2:c<35?3:4$，其值为 1，对应的 case 语句为 case 1。由于每个 case 语句后都没有 break 语句，则将从 case 1 开始后面的所有语句都会被执行。因此运行结果如下：

```
8℃有点冷
8℃正合适
8℃有点热
8℃太热了
```

3.3.3 使用 switch 语句的注意事项

用户在使用 switch 语句时应注意以下 6 个问题。

(1) case 后的常量表达式的值应该与 switch 后面的表达式的值类型一致,并且跟在 switch 后面的表达式的值和所有 case 后面的常量表达式的值都必须为整型或字符型,不允许为浮点型。

(2) 同一个 switch 语句中所有 case 后面的常量表达式的值应互不相同,否则会出现值存在语法错误。

(3) 当 switch 表达式的值与某个 case 子句中的常量表达式的值相匹配时,就执行此 case 子句中的内嵌语句,若所有的 case 子句中的常量表达式的值都不能与 switch 表达式的值匹配,就执行 default 子句的内嵌语句。

(4) 各 case 和 default 子句的先后顺序变动不会影响程序的执行结果,如果 default 子句前置,后边要加 break 语句结果才正确,只有最后的分支语句可以不加 break 语句且不影响结果。

(5) 多个 case 可以共用一组执行语句。例如:

```
case 4:
case 5:
case 6:
case 7:
case 8:
case 9: printf("很遗憾,没有奖品给你");
```

当输入 4、5、6、7、8、9 时都在屏幕上显示“很遗憾,没有奖品给你”。

(6) 每个 case 后面可以是一个语句,也可以是多个语句,还可以没有语句。当是多个语句时,可以不用大括号括起来。

3.3.4 多路选择结构的比较

switch 结构和 if-else if-else 多路选择结构都能实现多分支选择结构,但两者各有优势。if-else if-else 多路选择结构的控制能力比 switch 的条件控制更强,if-else if-else 多路选择结构可以依照各种逻辑运算的结果进行流程控制,而 switch 只能进行==(相等)判断,并且只能是整数判断。而 switch 的结构比 if-else if-else 多路选择结构更清晰。

注意:两者都要尽量避免用得过多、过长,尤其不要嵌套层数太多,否则会大大增加程序的分支,使逻辑关系显得混乱,不易维护,易出错。

3.3.5 实战演练

编程查找途经某学院的公交车的路线信息。用户输入第几路公交车,程序将输出其途

经路线上的主要站点信息。表 3.7 是途经学院的公交车路线信息。

表 3.7　公交车路线信息

公交车路数	线路途经的主要站点
70 路	西南林业大学、石闸立交、环城北路、小菜园立交、黄土坡、津桥学院、海源寺
83 路	昆明北站、塘子巷、交三桥、莲花宾馆、黄土坡、国际花园、津桥学院、海源寺
116 路	南屏街、昆师路、麻园、黄土坡、高新区(津桥学院)、昌源路、滇缅大道
K13 路	南屏街、小西门、西园路口、洪家营、高新区(津桥学院)、商院路、海源寺

提示：输入公交车路数，用 switch 语句选择匹配表 3.7 中的路数，然后输出途经路线信息。若输入的路数不在该表之列，则给出提示。

3.3.6　综合设计(简单界面设计)

例 3-14　从键盘上输入两个数和一个运算符(+：加，-：减，*：乘，/：除)，从而构成一个二元运算表达式，计算其值并输出运算结果。

分析：首先输入两个数和运算符，然后根据运算符进行相应的运算，但做除法时应判断除数是否非法(为 0)，如果运算符不是+、-、*、/，则同样是非法。

```
#include "stdio.h"
void main()
{
    float n1,n2;                          /*存放两个数的变量*/
    int tag=0;                            /*运算合法标志,0: 合法,1: 非法*/
    char op;                              /*运算符变量*/
    float result;                         /*运算结果变量*/
    printf("_______表达式计算_______\n");
    printf("输入两个数和一个运算符(+: 加,-: 减,*: 乘,/: 除),\n");
    printf("构成简单表达式,例如,3.5+7.2\n");
    scanf("%f%c%f",&n1,&op,&n2);
    switch(op)
    {
    case '+': result=n1+n2;
        break;
    case '-': result=n1-n2;
        break;
    case '*': result=n1*n2;
        break;
    case '/': if(!n2)                     /*判断除数是否为0*/
            {
                  printf("除数是0,非法!\n");
                  tag=1;                  /*置运算非法标志*/
            }
            else
                result=n1/n2;
            break;
    default: printf("运算符非法!\n");
          tag=1;                          /*置运算非法标志*/
    }
```

```
    if(!tag)
        printf("运算结果: %.2f %c %.2f = %.2f\n",n1,op,n2,result);
}
```

运行结果：

```
_______表达式计算_______
输入两个数和一个运算符(+:加,-:减,*:乘,/:除),
构成简单表达式,例如,3.5+7.2
3.5+7.2
运算结果: 3.50+7.20=10.70
```

程序运行效果如图 3.9 所示。

图 3.9　表达式运算结果

3.4　小结

本章详细介绍了选择结构，用于实现选择结构的语句是 if 语句和 switch 语句。它们根据语句中的条件判断的结果选择所要执行的程序分支，其中条件可以用表达式来描述，如关系表达式和逻辑表达式等。

if 语句有 3 种形式，分别如下。

(1) 简单选择结构。

语句形式为

```
if(表达式)
  语句;
```

(2) 二路选择结构。

语句形式为

```
if(表达式)
  语句 1;
else
  语句 2;
```

(3) 多路选择结构。

语句形式为

```
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 m)
    语句 m;
else
    语句 n;
```

switch 语句形式如下：

```
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;
    …
case 常量表达式 n: 语句 n;
    break;
default:语句 n + 1;
}
```

习　题　3

1. 选择题

(1) 用逻辑表达式表示“大于 10 而小于 20 的数”，正确的是(　　)。

A. 10＜x＜ 20　　B. x＞10 ‖ x＜ 20

C. x＞10 &x＜ 20　　D. !(x＜=10 ‖ x＞=20)

(2) $x=1, y=1, z=1$，执行表达式 w＝＋＋x ‖ ＋＋y&&＋＋z 后 x、y、z 的值分别为(　　)。

A. $x=2$，$y=1$，$z=1$　　B. $x=2$，$y=2$，$z=2$

C. $x=1$，$y=1$，$z=1$　　D. $x=2$，$y=2$，$z=1$

(3) 已知“int a ＝ 10，b ＝ 11，c ＝ 12;”，表达式(a＋b)＜c&&b＝＝c 的值是(　　)。

A. 2　　B. 0　　C. －2　　D. 1

(4) 为了避免在嵌套的条件语句 if-else 中产生歧义，C 语言规定的 if-else 语句的匹配原则是(　　)。

A. else 子句与所排位置相同的 if 配对

B. else 子句与其之前最近的尚未配对的 if 配对

C. else 子句与其之后最近的 if 配对

D. else 子句与同一行上的 if 配对

(5) 判断 char 型变量 ch 是否为大写字母的正确表达式是(　　)。

A. 'A'<=ch<='Z'　　B. (ch>='A')&(ch<='Z')

C. (ch>='A')&&(ch<='Z')　　D. ('A'<=ch)and('Z'<=ch)

(6) 为表示关系 $x \geqslant y \geqslant z$,应使用的 C 语言表达式为(　　)。

A. (x>=y)&&(y>=z)　　B. (x>=y)and (y>=z)

C. x>=y>=z　　D. (x>=y)&(y>=z)

(7) 设 x、y 和 z 是 int 型变量,且 $x=3$,$y=4$,$z=5$,则下面表达式中值为 0 的是(　　)。

A. x&&y　　B. x<=y

C. x‖y+z&&y-z　　D. !((x<y)&&!z‖1)

(8) 以下运算符中优先级最低的运算符为(　　)。

A. &&　　B. &　　C. !=　　D. ‖

(9) 下列程序段执行后 x 的值为(　　)。

```
int x,y=5;
x=++y;
if(x==y)
    x*=2;
if(x>y)
    x++;
else
    x=y-1;
```

A. 5　　B. 10　　C. 13　　D. 9

(10) 根据下面的程序段判断 x 的取值在(　　)范围内时将打印字符串“第二”。

```
if(x>0)
    printf("第一");
else if(x>-3)
    printf("第二");
else
    printf("第三");
```

A. $x>0$　　B. $x>-3$

C. $x<=-3$　　D. $x<=0$&$x>-3$

(11) 已知“int x=10,y=20,z=30;”,以下语句执行后 x、y、z 的值是(　　)。

```
if(x>y)
    z=x;
x=y;
y=z;
printf("%d,%d,%d",x,y,z);
```

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

(12) 以下关于逻辑运算符两侧运算对象的叙述正确的是(　　)。

A. 只能是整数 0 或 1　　B. 只能是整数 0 或非 0 整数

C. 只能是整数 0 或正整数　　D. 可以是任意合法的表达式

（13）下列程序段的输出结果是（　　）。

```
int a = 2,b = 3,c = 4,d = 5;
int m = 2,n = 2;
a = (m = a > b)&&(n = c > d) + 5;
printf(" % d, % d",m,n);
```

A. 0,2　　B. 2,2　　C. 0,0　　D. 1,1

（14）下列叙述中正确的是（　　）。

A. break 语句只能用于 switch 语句

B. 在 switch 语句中必须使用 default

C. break 语句必须与 switch 语句中的 case 配对使用

D. 在 switch 语句中不一定使用 break 语句

2. 填空题

（1）在 C 语言中用_____表示逻辑值“真”，用_____表示逻辑值“假”。

（2）以下程序执行后的输出结果是_____。

```
#include "stdio.h"
void main()
{
    int x = 10,y = 20 ,t = 0;
    if(x == y)
        t = x;
    x = y;
    y = t;
    printf(" % d, % d\n",x,y);
}
```

（3）以下程序的功能是将输入的一个小写字母循环后移 5 个位置后输出。例如，'a'变成'f'，'w'变成'b'，请补充语句。

```
#include "stdio.h"
void main()
{
    char c;
    scanf(" % c",&c);
    if(c >= 'a'&&c <= 'u')
        ____①____;
    else if (c >= 'v'&&c <= 'z')
        ____②____;
    printf(" % c\n",c);
}
```

（4）将下面的程序段改为条件表达式为_____。

```
int max,a,b;
if(a > b)
    max = a;
```

```
else
    max = b;
```

(5) 有以下程序段,正确的数学函数关系是_____。

```
if(x == 0)
    y = 0;
else if(x > 0)
    y = 1;
else
    y = -1;
```

(6) 与条件表达式"x=k?i++:i--",等价的语句是_____。

(7) 请写出以下程序的输出结果: _____。

```
#include "stdio.h"
void main()
{
    int a = 100;
    if(a > 100)
        printf("%d\n",a > 100);
    else
        printf("%d\n",a <= 100);
}
```

3. 读程序写结果题

(1) 以下程序的输出结果是_____。

```
#include "stdio.h"
void main()
{
    int a = 0,i;
    switch(i)
    {
    case 0:
    case 3: a += 2;
    case 1:
    case 2: a += 3;
    default: a += 5;
    }
    printf("%d\n",a);
}
```

(2) 以下程序的输出结果是_____。

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

```
    a = 2;b = 7;c = 5;
    switch(a > 0)
    {
    case 1:
        switch(b < 10)
        {
        case 1:printf("@");break;
        case 0:printf("!");break;
        }
    case 0:
        printf(" * ");break;
    default:printf("&");
    }
}
```

（3）以下程序的输出结果是_____。

```
#include "stdio.h"
void main()
{
    int a = 1,b = 2,d = 0;
    if(a == 1)
        d = 1;
    else if(b!= 3)
        d = 3;
    else
        d = 4;
    printf(" % d\n",d);
}
```

（4）以下程序的输出结果是_____。

```
#include "stdio.h"
void main()
{
    int x = 1,y = 0,a = 0,b = 0;
    switch( x )
    {
    case 1:
        switch( y )
        {
        case 1: a++; break;
        case 0: b++;
        case 2: b++; break;
        }
    case 2: a++; b++; break;
    case 3: a++; b++;
    }
    printf("\n a = % d, b = % d\n", a, b);
}
```

(5) 以下程序的输出结果是_____。

```
#include "stdio.h"
void main()
{
    int k1 = 1,k2 = 2,k3 = 3,x = 15;
    if(!k1)
        x--;
    else if(k2)
        if(k3)
            x = 4;
        else
            x = 3;
    printf("x = %d\n",x);
}
```

(6) 以下程序的输出结果是_____。

```
#include "stdio.h"
void main()
{
    int a = -1,b = 4,k;
    k = (a++<=0)&&(!b--<=0);
    printf("%d,%d,%d\n",k,a,b);
}
```

4. 编程题

(1) 编写一个程序,实现以下功能：从键盘输入两个整数,检查第一个数是否能被第二个数整除,并输出判断结果。

(2) 从键盘输入三角形 3 条边的边长,判断是否能构成三角形,若能,则输出该三角形的面积及类型(等边、等腰、直角、一般),否则输出“不能构成三角形”。

(3) 求方程 $ax^2+bx+c=0$ 的解。

基本的算法如下：

① $a=0$,不是二次方程。

② $b^2-4ac=0$,有两个相等实根。

③ $b^2-4ac>0$,有两个不等实根。

④ $b^2-4ac<0$,有两个共轭复根。

(4) 编程判断所输入整数的奇偶性。

(5) 某加油站有 a、b、c 3 种汽油,单价分别为 6.12、5.95、5.75(元/千克),同时提供“自动加油”“手工加油”两种服务模式,分别给予 2%和 5%的优惠。编写程序实现以下功能：当用户输入加油量、汽油品种和服务类型后输出应付金额。

(6) 企业发放的奖金根据利润提成,利润 i 小于或等于 10 万元,奖金可提成 10%；利润大于 10 万元,小于或等于 20 万元,即 $100000<i\leqslant200000$ 时,小于 10 万元的部分按 10%提成,大于 10 万元的部分可提成 7.5%；$200000<i\leqslant400000$ 时,小于 20 万的部分仍按上述办法提成(下同),大于 20 万元的部分按 5%提成；$400000<i\leqslant600000$ 时,大于 40 万元的部分按 3%提成；$600000<i\leqslant1000000$ 时,大于 60 万元的部分按 1.5%提成；$i>1000000$

时，大于 100 万元的部分按 1%提成。从键盘输入当月利润 i，求应发奖金总数。要求：

① 用 if 语句编写程序。

② 用 switch 语句编写程序。

本章实验实训

【实验目的】

(1) 熟练掌握条件表达式的运用方法。

(2) 熟练掌握 if 语句的运用方法。

(3) 熟练掌握 switch 语句的运用方法。

(4) 学会选择结构程序的设计。

(5) 学会简单界面设计方法。

【实验内容及步骤】

编写程序，在屏幕上显示一张如下格式的学生活动安排表：

```
********** 我每天的活动安排 **********
        选项号      活动信息
          1         起床、洗漱、晨跑
          2         早餐
          3         上课
          4         午餐
          5         实验室做实验
          6         踢足球
          7         晚餐
          8         图书馆
          9         睡觉
请输入选项号(1～9):
```

当操作人员根据以上提示输入某选项号后，程序显示活动时间和活动信息。活动与时间的对照关系如下：

1	6:30	起床、洗漱、晨跑
2	7:30	早餐
3	8:00	上课
4	12:00	午餐
5	14:00	实验室做实验
6	17:00	踢足球
7	18:10	晚餐
8	19:00	图书馆
9	23:00	睡觉

注意：若输入了非法的选项号应该提示选项出错。