

A decorative horizontal border at the bottom of the page featuring a repeating pattern of stylized grey leaves and stems against a light green background.

引言



在之前的章节中我们学习了选择结构程序设计。为了提升编写的程序种类和构建程序的灵活性，本章我们将学习选择结构程序设计。选择结构是对程序中某个变量或表达式的值做出判定，根据判定结果决定执行哪些语句和跳过哪些语句。为了实现选择结构的程序设计，C 语言引入了 if 语句和 switch 语句。另外，借助于条件运算符也可以实现简单的选择结构。

通过本章的学习，使读者了解并掌握 C 语言的选择结构程序设计方法，加深对 C 语言程序开发设计过程的认识，强化培养编程思路，为进一步进行 C 语言程序设计，打下坚实的基础。





任务 6 闰年表达式

1

任务描述

2

知识准备

3

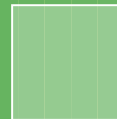
知识应用

4

任务实现



任务描述



本任务设计完成一个闰年判断程序。闰年判断条件为：

- 1 . 年份能被 4 整除而不能被 100 整除
- 2 . 年份能被 400 整除

如果满足上述条件，即为闰年，否则为平年。



知识准备



条件运算符可以实现简单的选择作用。条件表达式的值可以用在赋值语句中。关系表达式与逻辑表达式的值只有两个，即 1 或 0。C 语言用 1 表示“真”，用 0 表示“假”。

一、条件运算符

1．条件运算符（`?:`），是 C 语言中唯一需要三个操作数的运算符，它可以组成一个条件表达式，其一般形式为：

表达式 1 `?` 表达式 2 `:` 表达式 3

2．条件运算符的执行顺序。首先判断表达式 1 的值，如果数值不是 0，则计算表达式 2 的值，并将此作为条件表达式的值；如果数值是 0，则计算表达式 3 的值，并将此作为条件表达式的值。

3．条件运算符的优先级优先于赋值运算符，低于关系运算符，结合性为右结合。



知识准备



二、关系运算符与关系表达式

1. 关系运算符用于比较表达式，提出类似“a 大于 10 吗？”或“x 等于 y 吗？”这样的问题。关系表达式的一般形式：

表达式关系运算符表达式

2. 关系表达式的值，根据所描述的关系成立与否取值为 1 或 0。而不是像有些语言的真与假。即关系成立，值为 1；不成立，值为 0。见下表 4-1 关系运算符和关

表 4-1 关系运算符和关系表

运算符	符号	实例	读作	求值
大于	$>$	$5 > 3$	5 大于 3 吗？	1
大于或等于	$\geq$	$3 \geq 3$	3 大于等于 3 吗？	1
小于	$<$	$'a' < 'b'$	'a' 小于 'b' 吗？	1
小于或等于	$\leq$	$5 \leq 3$	5 小于等于 3 吗？	0
等于	$==$	$3 == 3$	3 等于 3 吗？	1
不等于	$!=$	$3 != 3$	3 不等于 3 吗？	0

3. 优先级与结合性

运算顺序由高到低为：算术运算符 $\rightarrow$ 关系运算符 $<, \leq, >, \geq \rightarrow$ 关系运算符 $==$ 和 $!=$

关系运算符的结合性：从左至右。



知识准备



三、逻辑运算符与逻辑表达式

1. C 语言有三种逻辑运算符。见表 4-2。用逻辑运算符将关系表达式或逻辑量连接起来的式子称为逻辑表达式。

表 4-2 逻辑运算符

运算符	功能	表达式
&&	逻辑与	<code>a && b</code>
	逻辑或	<code>a b</code>
!	逻辑非	<code>!a</code>

2. 一般意义，逻辑表达式成立其值为“真”，不成立其值为“假”。在 C 语言中用数值 1 代表“真”，用数值 0 代表“假”。逻辑运算的运算含义见真值表 4-3。

表 4-3 C 逻辑运算的真值表

a	b	<code>a && b</code>	<code>a b</code>	<code>!a</code>	<code>!b</code>
非 0	非 0	1	1	0	0
非 0	0	0	1	0	1
0	非 0	0	1	1	0
0	0	0	0	1	1



知识准备



3 . 逻辑运算的优先级顺序：

！（非）→ &&（与）→ ||（或）

！（非）运算为单目运算符，结合性为右结合。&&（与）、||（或）为双目运算符，结合性是左结合。

4 . C 语言逻辑运算总结：（a,b 代表一个关系表达式或逻辑量）

&& 运算：当 a ， b 都为非 0 时， a&&b 的值为 1 ，其它情况均为 0 ；

|| 运算：当 a ， b 都为 0 时， a || b 的值为 0 ，其它情况均为 1 ；

! 运算：a 为非 0 ， ! a 的值为 0 ； a 为 0 ， !a 的值为 1 。



知识应用



1. 求三个数中最大值的程序

分析：

(1) 将三个数存于 x 、 y 、 z 变量中，其中的最大数用 $\max$ 标识。通过比较，输出最大值，由于一次只能比较两个数，三个数比大小应比较两次。

(2) 第一次： x 和 y 比较，把其中的大数送入 $\max$ 变量中。

(3) 第二次： z 和 $\max$ 比较，将大数送入 $\max$ ，此时， $\max$ 中将是三个数中的最大数。

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

```
main( )
```

```
{
```

```
    float x,y,z,max;
```

```
    scanf("%f%f%f",&x,&y,&z);
```

```
    max=x>y?x:y;
```

```
    max=z>max?z:max;
```

```
    printf("max=%.2f \n",max);
```

```
}
```

运行时输入：10 20 30 ✓

输出：max=30.00



知识应用



2 . 写出下面程序段的运行结果

```
#include<stdio.h>
main( ) {
int a,b=10; char c='A';
printf("%d,%d\n",a=5,b==5);
printf("%d\n",c+3<'D');
printf("%d,%d\n",a=b==a+8,3>2>1);
}
```

说明：

(1) 赋值表达式的值是被赋变量的值。因此赋值表达式 $a=5$ 的值是 a 的值，是 5。 $b==5$ 是关系表达式，不成立，其值是 0。

(2) 字符型数据相比较时，是比较它们的 ASCII 值。

(3) 表达式 $a==b=a+8$ 共有 3 个运算符，根据优先级先计算 “ $a+8$ ”： $a=b==13$ ，然后计算 “ $b==13$ ”： $a=0$ ，最后计算 “ $a=0$ ”，结果是 0。

(4) $3>2>1$ 根据从左至右的结合性，先计算 “ $3>2$ ”，结果是 1，再计算 “ $1>1$ ”，结果是 0。因此 $3>2>1$ 的值是 0。在 C 语言中关系表达式的值为 0 是不成立的意思。而在数学上 $3>2>1$ 是成立的。可见，我们在描述条件时要用 C 的思维而不是数学的思维。

运行结果：

5,0

0

0,0



知识应用



3 . 用一个表达式判断字符型变量 ch 是否是一个大写英文字母

分析：如果你写成了如下的关系表达式： $'A' \leq ch \leq 'Z'$ ，这正是我们接下来要分析的问题。C 语言在对这个表达式求值时，根据关系运算符从左至右的结合性，首先求出 $'A' \leq ch$ 的值，结果是 0 或 1，然后再求 $0 \leq 'Z'$ 或 $1 \leq 'Z'$ ，这两个表达式的结果都是 1，即无论是否是大写字母，该表达式都成立。因此是错误的判断条件。

正确表示出来，必须先分解成两个简单的条件，即： $ch \geq 'A'$ 和 $ch \leq 'Z'$ ，然后再用逻辑运算符连接起来，构成一个逻辑表达式。

根据与运算的含义，本题表示为： $ch \geq 'A' \&\& ch \leq 'Z'$ 。

4 . 若有变量 $\text{int } a=3, b=2, c=1$, 则表达式 $a-b < c \parallel b == c$ 的值是 ()

分析：表达式中共有四个运算符，优先级由高到低的顺序是：“-” 高于 “<” 高于 “==” 高于 “ $\parallel$ ”，因此求解过程为：计算 $a-b \rightarrow$ 计算 $a-b < c \rightarrow$ 计算 $b == c \rightarrow$ 计算整体表达式的值，最后结果为 0。



任务实施



一、任务流程分解

1. 流程描述：程序开始提示用户输入要判断的年份，然后判断输入的年份是否符合闰年条件，如果符合，输出闰年，如果不符合，输出平年。

(1) 程序初始化分析：定义年份变量

(2) 数据录入分析：用户输入要判断的年份

(3) 数据处理分析：判断用户输入的年份是否符合闰年规则

(4) 输出结果分析：

结果1：符合闰年判断规则，输出闰年；

结果2：不符合闰年判断规则，输出平年。



任务实施



二、代码实现

```
#include <stdio.h>
main ( )
{
    int year;
    printf(" 请输入要
判断的年份 : ");

    scanf("%d",&year);
    if(year
%4==0&&year%100!=0||
year%400==0)
        printf
(" 闰年 \n");
    else
        printf
(" 平年 \n");
}
```

三、结果演示

输入 2016 ，输出闰年的结果，程序结果演示如图 4-1 （ 1 ）所示。

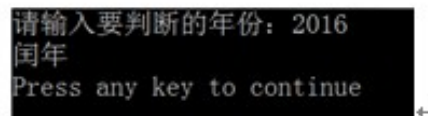


图 4-1 演示结果界面（1）

输入 2000，输出闰年的结果，程序结果演示如图 4-1 （2）所示。

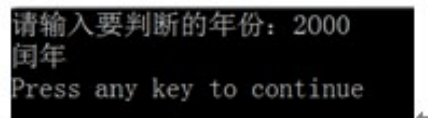


图 4-1 演示结果界面（2）

输入 2014，输出平年的结果，程序结果演示如图 4-1 （3）所示。



图 4-1 演示结果界面（3）





任务 7 判定业务员级别

1

任务描述

2

知识准备

3

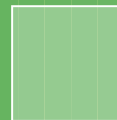
知识应用

4

任务实现



任务描述



本任务设计完成一个判定业务员**级别程序**。保险公司根据月签单金额将业务员划分为 5 个级别。规则是：月签单 10 万元以上为“金牌保险员”；月签单 8 万元以上为“银牌保险员”；月签单 6 万元以上为“铜牌保险员”，月签单 4 万元以上为“铁牌保险员”，否则“红牌警示”。程序运行后，输入业务员月签单金额，输出该业务员对应的级别。



知识准备



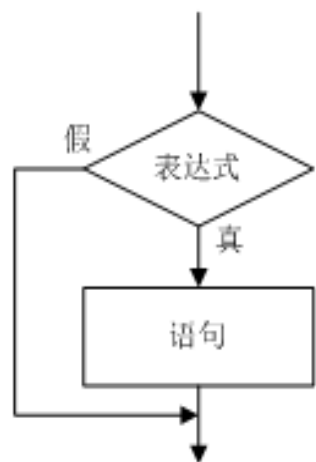
C 语言的 if 语句是根据给定的条件进行判断，以决定执行哪些语句和跳过哪些语句不执行。if 语句有三种基本形式：

一、if 语句的第一种形式：

if (表达式)

语句；

执行过程：先计算表达式的值，如果结果为非 0 值，则执行其中的“语句”，如果求值是 0，则不执行“语句”。无论哪一种情况，下一步都要执行 if 语句之后的代码。流程图见图 4-2。



实例：if (x>y)
printf("%d",x);
printf("ok");

如果有定义 x =7;y=5; 则输出： 7ok

如果有定义 x =5;y=7; 则输出： ok

图 4-2 if 语句第一种形式流程图



知识准备



二、if 语句的第二种形式：

if (表达式)

语句 1 ；

else

语句 2;

执行过程：先计算表达式的值，如果结果为非 0 值，则执行语句 1 而不执行语句 2；如果求值是 0，则执行语句 2 而不执行语句 1。流程图见图 4-3。

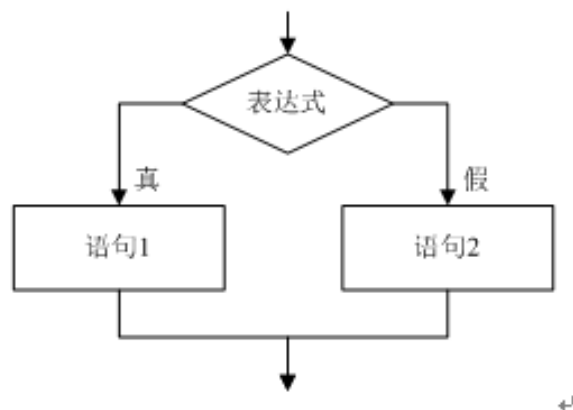


图 4-3if 语句第一种形式流程图



知识准备



三、if 语句的第三种形式：

```
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 n+1;
```

执行过程：如果表达式 1 的结果为非 0 值，则执行语句 1；否则，判定表达式 2 的值，若为非 0，则执行语句 2；否则.....否则，执行语句 n+1。流程图见图 4-4。

分号是 C 语句的必要成份，谈到语句就一定包含了一个分号在末尾。因此上述 if 语句中的第二、第三种形式中，在每个 else 前面有一个分号。这三种形式每一个都是一条 c 语句，因此整个语句结束处有一个分号。这三种形式中的“语句”又都可以是复合语句。其中 () 内的表达式一般为逻辑表达式和关系表达式，但可以是任意类型。如：

```
if ('A')  
    printf("A"); /* 'A' 的值非 0，printf( ) 函数得到执行，输出 " A"*/
```



知识准备

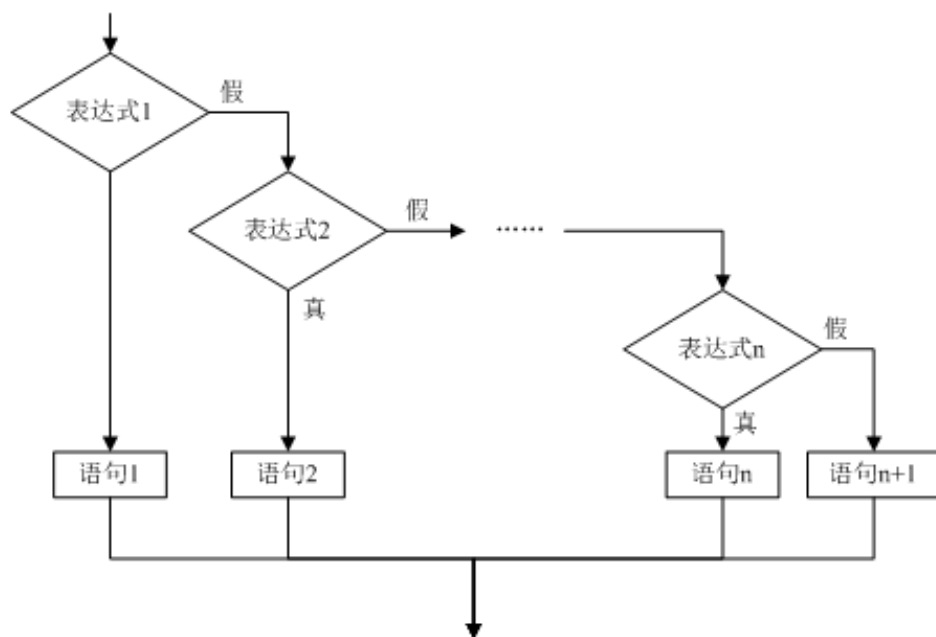


图 4-4if 语句第一种形式流程图

说明：

- 1 . if 语句中的“表达式”必须用“(”和“)”括起来。
- 2 . else 子句是 if 语句的一部分，必须与 if 配对使用，不能单独使用。
- 3 . 在三种形式的 if 语句中，if 关键字之后均为表达式。该表达式通常是逻辑表达式或关系表达式，但也可以是其它表达式。只要表达式的值为非 0，即为“真”。其后的语句总是要执行的。
- 4 . if 语句的第一、第二种形式实现的是两者选一的情形，若有三种以上情况则需要用第三种形式的 if 语句。



知识应用



1 . 输入一个数 x , 要求不使用 `abs()` 函数 , 输出其绝对值。

分析：首先输入一个数给 x 变量 , 然后进行判断 , `if` 语句中的条件为 $x < 0$, 当条件成立时 , 可用 $(-x)$ 求出其绝对值 , 否则输出 x 即可。

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

```
main()
```

```
{
```

```
    float x;
```

```
    printf(" 请输入一个数 : ");
```

```
    scanf("%f",&x);
```

```
    if (x<0)
```

```
    {
```

```
        printf("%f\n", -x);
```

```
    }
```

```
    else
```

```
    {
```

```
        printf("%f\n",x);
```

```
    }
```

```
}
```



知识应用



2. 输入两个整数，将大数存在 x 中，小数存在 y 中。

分析：将两个数输入后依次存在 x, y 变量中。如果 $(x \geq y)$ 符合题中要求输出即可，如果 $(x < y)$ ，则需要交换 x 和 y 中的数。交换两个变量的值，需要用到第三个变量如 $temp$ ，其方法是：先将两个变量中的一个，如 x 的值存入第三个变量（ $temp$ ），然后将另一个变量（ y ）的值存入第一个变量（ x ），最后将存放在第三个变量中的原第一个变量的值存入第二个变量。下面的图 4-5，有助于你理解交换两个变量的三条语句。

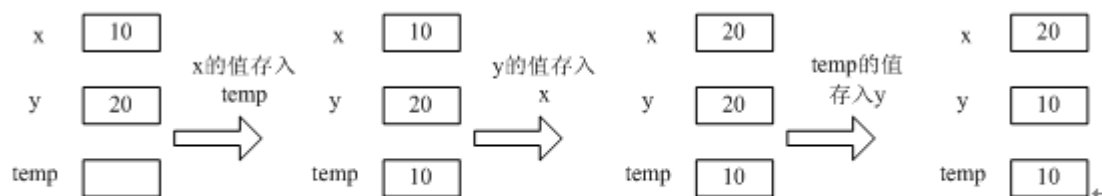


图 4-5 两个数交换

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

```
main() {  
    int x,y,temp;  
    printf(" 请输入两个数 : ");  
    scanf("%d,%d",&x,&y);  
    if(x>=y) {  
        printf("x 的值为 : %d\n",x);  
        printf("y 的值为 : %d\n",y);  
    }  
    else {  
        temp = x;  
        x = y;  
        y = temp;  
        printf("x 的值为 : %d\n",x);  
        printf("y 的值为 : %d\n",y);  
    }  
}
```

任务准备



一、任务流程分解

1. 流程描述：首先输入业务员完成金额 m 万元；然后按照规则对 m 进行判断，如果 $m > 10$ ，输出“金牌保险员”，否则如果 $m > 8$ ，输出“银牌保险员”，否则如果 $m > 6$ ，输出“铜牌保险员”，否则如果 $m > 4$ ，输出“铁牌保险员”，否则如果 $m \geq 0$ ，输出“红牌警示”，否则输出“输入的金额有误！”。

(1) 程序初始化分析：定义业务员完成金额 m

(2) 数据录入分析：输入完成金额 m

(3) 数据处理分析：按照公司制定规则，对完成金额 m 进行判断

(4) 输出结果分析：

结果 1： $m > 10$ ，金牌保险员；

结果 2： $m > 8$ ，银牌保险员；

结果 3： $m > 6$ ，铜牌保险员；

结果 4： $m > 4$ ，铁牌保险员；

结果 5： $m \geq 0$ ，红牌警示；

结果 6： m 取其他值，提示输入的金额有误



任务准备



二、代码实现

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

```
main() {
```

```
    float m;
```

```
    printf(" 请输入完成金额 ( 万元 ) : ");
```

```
    scanf("%f",&m);
```

```
    if(m>=10)
```

```
        printf(" 金牌保险员 \n");
```

```
    else if(m>=8)
```

```
        printf(" 银牌保险员 \n");
```

```
    else
```

```
        if(m>=6)
```

```
            printf(" 铜牌保险员 \n");
```

```
        else
```

```
            if(m>=4)
```

```
                printf(" 铁牌保险员 \n");
```

```
            else
```

```
                if(m>=0)
```

```
                    printf(" 红牌警示 \n");
```

```
                else
```

```
                    printf(" 您输入的金额不正确 ! \n");
```

```
}
```



任务准备



三、结果演示

输入 15 万元，输出金牌保险员的结果，程序结果演示如图 4-6（1）所示。

```
请输入完成金额(万元): 15  
金牌保险员  
Press any key to continue
```

图 4-6 演示结果界面（1）

输入 9 万元，输出银牌保险员的结果，程序结果演示如图 4-6（2）所示。

```
请输入完成金额(万元): 9  
银牌保险员  
Press any key to continue
```

图 4-6 演示结果界面（2）

输入 7 万元，输出铜牌保险员的结果，程序结果演示如图 4-6（3）所示。

```
请输入完成金额(万元): 7  
铜牌保险员  
Press any key to continue
```

图 4-6 演示结果界面（3）



任务准备



输入 5 万元，输出铁牌保险员的结果，程序结果演示如图 4-6（4）所示。↵

```
请输入完成金额(万元): 5
铁牌保险员
Press any key to continue
```

图 4-6 演示结果界面（4）↵

输入 3 万元，输出红牌警示的结果，程序结果演示如图 4-6（5）所示。↵

```
请输入完成金额(万元): 3
红牌警示
Press any key to continue
```

图 4-6 演示结果界面（5）↵

输入 -2 万元，输出您输入的金额不正确的结果，程序结果演示如图 4-6（6）所示。↵

```
请输入完成金额(万元): -2
您输入的金额不正确!
Press any key to continue
```

图 4-6 演示结果界面（6）↵





任务 8 标准体重

1

任务描述

2

知识准备

3

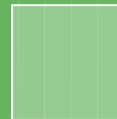
知识应用

4

任务实现



任务描述



本任务设计完成一个标准体重判断程序。程序运行后，按照提示信息，用户输入性别、身高（**cm**）和体重（**kg**）。男性的标准体重为：身高 -105；女性的标准体重为：身高 -110。设体重与标准体重上、下偏差 2kg 均属标准体重，否则为不标准体重。



知识准备



一、if 语句嵌套形式

在一个 if 语句中又包含另一个 if 语句，从而构成了 if 语句的嵌套使用。内嵌的 if 语句既可以嵌套在 if 子句中，也可以嵌套在 else 子句中。常用 if 语句嵌套形式如下：

(1) if 嵌套 if...else

```
if ( )  
    if ( ) 语句 1 ; /* 内层的 if...else 语句 */  
    else 语句 2 ;
```

(2) if...else 嵌套 if...else

```
if ( )  
    if ( ) 语句 1; /* 内层的 if...else 语句 */  
else 语句 2;  
else  
    if ( ) 语句 3; /* 内层的 if...else 语句 */  
else 语句 4;
```

说明：if 语句的嵌套形式不需要刻意追求，而是在解决实际问题过程中随着问题的需求而设计的。



知识准备



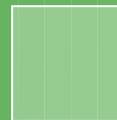
二、if 与 else 配对规则

上述形式中，因为书写上同一层次的 if 和 else 有相同的缩进，所以 if 和 else 的配对关系还很清晰。但值得一提的是，它们的配对关系，决不是依据有相同的缩进来判定的。if 和 else 的配对关系正是 if 嵌套程序的解读关键。if 与 else 的配对规则：

- 1 . else 总是与它上面最近的尚未与 else 配对的 if 配对。
- 2 . 如果 if 与 else 的数目不一样，为了实现程序设计的意图，可以使用加花括号来明确配对关系。



知识应用



1 . 输入 3 个数，输出其中最大值。

分析：从键盘输入 3 个数（ a, b, c ），首先比较 a 、 b 的大小，如果 $a > b$ ，那么继续比较 a 与 c 的大小，如果 $a > c$ ，最大值 $\max = a$ ；否则最大值 $\max = c$ ；如果 $a \leq b$ ，那么继续比较 b 与 c 的大小，如果 $b > c$ ，最大值 $\max = b$ ；否则最大值

```
max=c;
#include <stdio.h>
main() {
    int a,b,c,max;
    printf(" 请输入三个数： ");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b)
        if(a>c)
            max=a;
        else
            max=c;
    else
        if(b>c)
            max=b;
        else
            max=c;
    printf(" 最大值为： %d\n",max);
}
```



知识应用



2 . 三角形形状判断

分析：从键盘输入 3 个数 (a,b,c)，如果这 3 个数能构成一个三角形，则输出该三角形的形状信息。

(1) 按照三角形构成规则 (两边之和大于第三边)，判断是否能构成三角形，若能构成，继续判断三角形形状；若不能构成三角形，输出 “不能构成三角形”。

(2) 判断三角形形状：如果三边相等 (a==b&&b==c)，那么输出 “等边三角形”；否则，继续判断，如果两边相等 (a==b||b==c||a==c)，那么输出 “等腰三角形”；否则，输出 “普通三角形”。

```
#include <stdio.h>
main ( ) {
    float a,b,c;
    printf(" 请输入三角形三边长 ");
    scanf("%f%f%f",&a,&b,&c);

    if(a>0&&b>0&&c>0&&a+b>c&&a+c>b&&b+c>a)
    /* 三角形判定条件 */
    {
        if(a==b&&b==c)/* 三边长
        相等 */
            printf(" 等边
            三角形 ");
        else{
            if(a==b||b==c||a==c)/* 两边相等 */
                printf(" 等腰三角形 ");
            else
                printf(" 普通三角形 ");
        }
        else
            printf(" 不能构成三角形 ");
    }
}
```



任务实施



一、任务流程分解

1. 流程描述：开始程序提示用户输入性别（1 男 0 女），如果性别为“男”，则继续输入身高和体重，男性的标准体重 = 身高 - 105，然后判断输入体重与标准体重差值是否为 2，如果差值为 2，则输出“正常”，否则输出“不正常”；如果性别为女，则继续输入身高和体重，女性的标准体重 = 身高 - 110，然后判断输入体重与标准体重差值是否为 2，如果差值为 2，则输出“正常”，否则输出“不正常”；如果输入不是“男”，“女”，则提示“性别输入不正确”。

（1）程序初始化分析：定义身高、性别、体重及标准体重变量

（2）数据录入分析：用户输入性别、身高和体重

（3）数据处理分析：判断相关信息是否符合标准体重条件

（4）输出结果分析：

结果 1：男性，标准体重正常；

结果 2：男性，标准体重不正常；

结果 3：女性，标准体重正常；

结果 4：女性，标准体重不正常；

结果 5：其他，输入性别不正确。



任务实施



二、知识扩展

1 . 引入头文件：`#include<math.h>`

2 . 使用方式：

```
#include<math.h>
```

```
fabs(bztz-tz)<=2;/* 判断两个变量之差的绝对值值是否小于等于 2*/
```



任务实施



三、代码实现

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

```
#include <math.h>
```

```
main ( )
```

```
{
```

```
    float sg,tz,bztz;
```

```
    int sex;
```

```
    printf(" 请输入性别 :(1 男 0 女)");
```

```
    scanf("%d",&sex);
```

```
    if(sex==1)
```

```
    {
```

```
        printf(" 请输入身高 (cm) 和体重 (kg) : ");
```

```
            scanf("%f%f",&sg,&tz);
```

```
            bztz=sg-105;
```

```
            if(fabs(bztz-tz)<=2)
```

```
                printf(" 正常 !\n");
```

```
            else
```

```
                printf(" 不正常 !\n");
```

```
    }
```

```
    else
```

```
        if(sex==0)
```

```
        {
```

```
            printf(" 请输入身高 (cm) 和体重 (kg) : ");
```

```
            scanf("%f%f",&sg,&tz);
```

```
            bztz=sg-110;
```

```
            if(fabs(bztz-tz)<=2)
```

```
                printf(" 正常 !\n");
```

```
            else
```

```
                printf(" 不正常 !\n");
```

```
            }
```

```
        else
```

```
            printf(" 性别输
```

```
入错误 !\n");
```

```
    }
```



任务实施



四、结果演示

输入性别 1，身高 175，体重 70，输出正常的结果，程序结果演示如图 4-7（1）所示。

```
请输入性别:(1男 0女)1
请输入身高(cm)和体重(kg): 175 70
正常!
Press any key to continue
```

图 4-7 演示结果界面（1）

输入性别 1，身高 175，体重 75，输出不正常的结果，程序结果演示如图 4-7（2）所示。

```
请输入性别:(1男 0女)1
请输入身高(cm)和体重(kg): 175 75
不正常!
Press any key to continue
```

图 4-7 演示结果界面（2）

输入性别 0，身高 165，体重 55，输出正常的结果，程序结果演示如图 4-7（3）所示。

```
请输入性别:(1男 0女)0
请输入身高(cm)和体重(kg): 165 55
正常!
Press any key to continue
```

图 4-7 演示结果界面（3）



任务实施



输入性别 0，身高 165，体重 55，输出正常的结果，程序结果演示如图 4-7（3）所示。

```
请输入性别:(1男 0女)0
请输入身高(cm)和体重(kg): 165 55
正常!
Press any key to continue
```

图 4-7 演示结果界面（3）

输入性别 0，身高 165，体重 50，输出不正常的结果，程序结果演示如图 4-7（4）所示。

```
请输入性别:(1男 0女)0
请输入身高(cm)和体重(kg): 165 50
不正常!
Press any key to continue
```

图 4-7 演示结果界面（4）

输入性别 2，输出性别输入错误的结果，程序结果演示如图 4-7（5）所示。

```
请输入性别:(1男 0女)2
性别输入错误!
Press any key to continue
```

图 4-7 演示结果界面（5）





任务 9 学生管理系统菜单

1

任务描述

2

知识准备

3

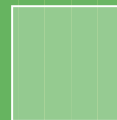
知识应用

4

任务实现



任务准备



本任务设计完成一个学生管理系统菜单界面。程序运行后，显示学生管理系统菜单界面，按照提示信息，用户输入相应数字，选择具体操作功能；根据用户输入的数字，提示用户选择的操作；如果用户输入的数字不在可选功能范围内，提示用户“输入不正确”。菜单界面上显示“1 学生学籍管理”、“2 增加学生信息”、“3 删除学生信息”、“4 修改学生信息”、“5 查询学生信息”、“6 学生成绩统计操作”、“0 退出系统”等内容。



知识准备



为了使循环控制更加灵活，C 语言提供了 switch 语句，便于设计多选一的程序。

一、switch 语句格式

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

二、switch 语句的执行

计算表达式 1 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，然后不再进行判断，继续执行后面所有 case 后的语句。如表达式的值与所有 case 后的常量表达式均不相同时，则执行 default 后的语句。



知识准备



三、switch 语句使用注意事项

- 1 . switch 后圆括号内“表达式”的类型，可以是任意的能求得一个整数值，如 int 类型或 char 类型值的表达式。
- 2 . 每一个 case 的常量表达式的值必须互不相同。
- 3 . 各个 case 和 default 的顺序可以任意，不影响执行结果。
- 4 . 多个 case 可以共用一组语句。

如： case 'A ':

case 'B ':

case 'C':printf(" 成绩合格 ");

break; /* 也可以将这三个 case 写在一行上 */

- 5 . 在 case 后，如果有一个以上执行语句，可以不用 {} 括起来。当然加上 {} 也可以。
- 6 . 各 case 和 default 子句的先后顺序可以变动，而不会影响程序执行结果。
- 7 . default 子句可以省略不用。
- 8 . switch 语句几乎和 break 分不开，尽管从语法上没有硬性的规定，但任何一个用到 switch 语句的实用程序，都离不开 break 语句。因为本质上，case 语句只是一个入口，并没有判断的功能，如果没有 break，程序就会毫无阻碍地长驱直入，而不具备了分支的作用。



知识应用



1 . 设计一个简单的计算器 , 能进行加减乘除运算。

分析 :

- (1) 将输入的参加运算的两个数保存在 x,y 变量中。
- (2) 输入的运算符是 + 、 - 、 * 、 / 中的一个 , 将其保存在字符变量 oper 中。
- (3) 用 switch 语句根据输入的不同运算符采用不同的计算公式。

```
#include <stdio.h>
main() {
    float x,y,result;
    char oper;
    printf(" 请输入两个数和一个运算符 ");
    scanf("%f%c%f",&x,&oper,&y);
    switch(oper) {
    case '+': result=x+y;      break;
    case '-': result=x-y;      break;
    case '*': result=x*y;      break;
    case '/': result=x/y;
                break;
    }
    printf("%.2f%c%.2f = %.2f\n",x,oper,y,result );
}
```





一、任务流程分解

游戏流程描述：程序开始运行后，按照提示信息，用户输入 1 进入的学生学籍管理菜单、输入 2-5 进入增删改查学生信息菜单、输入 6 进入学生成绩统计操作菜单、输入 0 退出系统。通过 switch 语句控制菜单选项，如果用户输入的信息都不等于 case 后面的常量表达式，则执行 default 语句。

- (1) 程序初始化分析：打印菜单选项
- (2) 数据录入分析：用户输入菜单选择数字
- (3) 数据处理分析：使用 switch-case 语句判断用户输入
- (4) 输出结果分析：

结果 1：您选择的是 1. 学生学籍管理；

结果 2：您选择的是 2. 增加学生信息；

结果 3：您选择的是 3. 删除学生信息；

结果 4：您选择的是 4. 修改学生信息；

结果 5：您选择的是 5. 查询学生信息；

结果 6：您选择的是 6. 学生成绩统计操作；

结果 7：谢谢使用！再见！；

结果 8：按键错误！请重新选择！。



任务实施

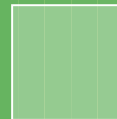


二、代码实现

```
#include<stdio.h>
main()
{
    int choice;/* 用户选择变量 */
    printf("*****& 学生成绩文件管理 &*****\n\n");
    printf("***** 请输入所需操作 *****\n");/* 选择功能菜单 */
    printf("1. 学生学籍管理 \n");
    printf("2. 增加学生信息 \n");
    printf("3. 删除学生信息 \n");
    printf("4. 修改学生信息 \n");
    printf("5. 查询学生信息 \n");
    printf("6. 学生成绩统计操作 \n");
    printf("0. 退出系统 \n");
    printf("*****\n");
    printf(" 请选择 :");
    scanf("%d", &choice);
```



任务实施



```
switch(choice) /* 多重选择实现功能不同的功能 */
{ case 1:          printf(" 您选择的是 1. 学生学籍管理 \n");
  break;
  case 2:          printf(" 您选择的是 2. 增加学生信息 \n");
  break;
  case 3:          printf(" 您选择的是 3. 删除学生信息 \n");
  break;
  case 4:          printf(" 您选择的是 4. 修改学生信息 \n");
  break;
  case 5:          printf(" 您选择的是 5. 查询学生信息 \n");
  break;
  case 6:          printf(" 您选择的是 6. 学生成绩统计操作 \n");
  break;
  case 0:          printf("\n 谢谢使用 ! 再见 !\n");
  break;
  default:
    printf("\n 按键错误 ! 请重新选择 !\n");
} /* 结束 switch */
```



任务实施



三、结果演示

输入 1, 输出您选择的是 1. 学生学籍管理的结果, 程序结果演示如图 4-8 (1) 所示。

```
*****&学生成绩文件管理&*****  
*****请输入所需操作*****  
1. 学生学籍管理  
2. 增加学生信息  
3. 删除学生信息  
4. 修改学生信息  
5. 查询学生信息  
6. 学生成绩统计操作  
0. 退出系统  
*****  
请选择:1  
您选择的是1. 学生学籍管理  
Press any key to continue
```

图 4-8 演示结果界面 (1)

输入 6, 输出您选择的是 6. 学生成绩统计操作的结果, 程序结果演示如图 4-8 (2) 所示。

```
*****&学生成绩文件管理&*****  
*****请输入所需操作*****  
1. 学生学籍管理  
2. 增加学生信息  
3. 删除学生信息  
4. 修改学生信息  
5. 查询学生信息  
6. 学生成绩统计操作  
0. 退出系统  
*****  
请选择:6  
您选择的是6. 学生成绩统计操作  
Press any key to continue
```

图 4-8 演示结果界面 (2)



任务实施



输入 0，输出谢谢使用！再见！的结果，程序结果演示如图 4-8（3）所示。

```
*****&学生成绩文件管理&*****  
  
*****请输入所需操作*****  
1. 学生学籍管理  
2. 增加学生信息  
3. 删除学生信息  
4. 修改学生信息  
5. 查询学生信息  
6. 学生成绩统计操作  
0. 退出系统  
*****  
请选择:0  
  
谢谢使用!再见!  
Press any key to continue
```

图 4-8 演示结果界面（3）

输入 9，输出按键错误！请重新选择！的结果，程序结果演示如图 4-8（4）所示。

```
*****&学生成绩文件管理&*****  
  
*****请输入所需操作*****  
1. 学生学籍管理  
2. 增加学生信息  
3. 删除学生信息  
4. 修改学生信息  
5. 查询学生信息  
6. 学生成绩统计操作  
0. 退出系统  
*****  
请选择:9  
  
按键错误!请重新选择!  
Press any key to continue
```

图 4-8 演示结果界面（4）



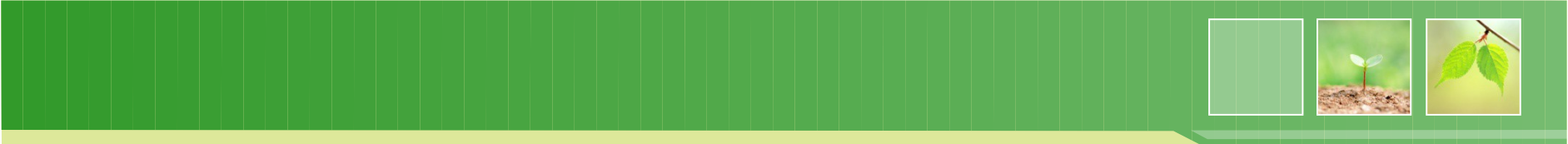
本章总结



前面章节中介绍的基础知识，能够完成顺序执行的控制程序。本章开始体现 C 语言的强大之处，在执行过程中根据用户输入的数据或计算的结果，来确定下一步的操作。本章学习了如何利用条件运算符来比较变量，再使用 if、if-else 和 switch 语句来控制程序执行过程。根据不同的执行过程，会有不同的结果输出。

本章重点内容是条件语句的格式、选择结构及分支的嵌套。难点内容为分支结构的产生和条件表达式的确定。选择结构是根据“条件”来决定选择哪一组语句，编程前，首先对要解决的问题进行逻辑分析，再确立每个分支点的判定条件，并确定每个分支各自的出入通道，要把各种情况所对应的处理语句都列出来。





C 语言程序设计



Thank You!

