

第4章 选择结构程序设计

4.1 选择结构和条件判断

4.2 用if语句实现选择结构

4.3 关系运算符和关系表达式

4.4 逻辑运算符和逻辑表达式

4.5 条件运算符和条件表达式

4.6 选择结构的嵌套

4.7 用switch语句实现多分支选择结构

4.8 选择结构程序综合举例

4.1 选择结构和条件判断

- 由于程序处理问题的需要，在大多数程序中都会包含选择结构，需要在进行下一个操作之前先进行条件判断
- C语言有两种选择语句：
 - (1) **if**语句，实现两个分支的选择结构
 - (2) **switch**语句，实现多分支的选择结构



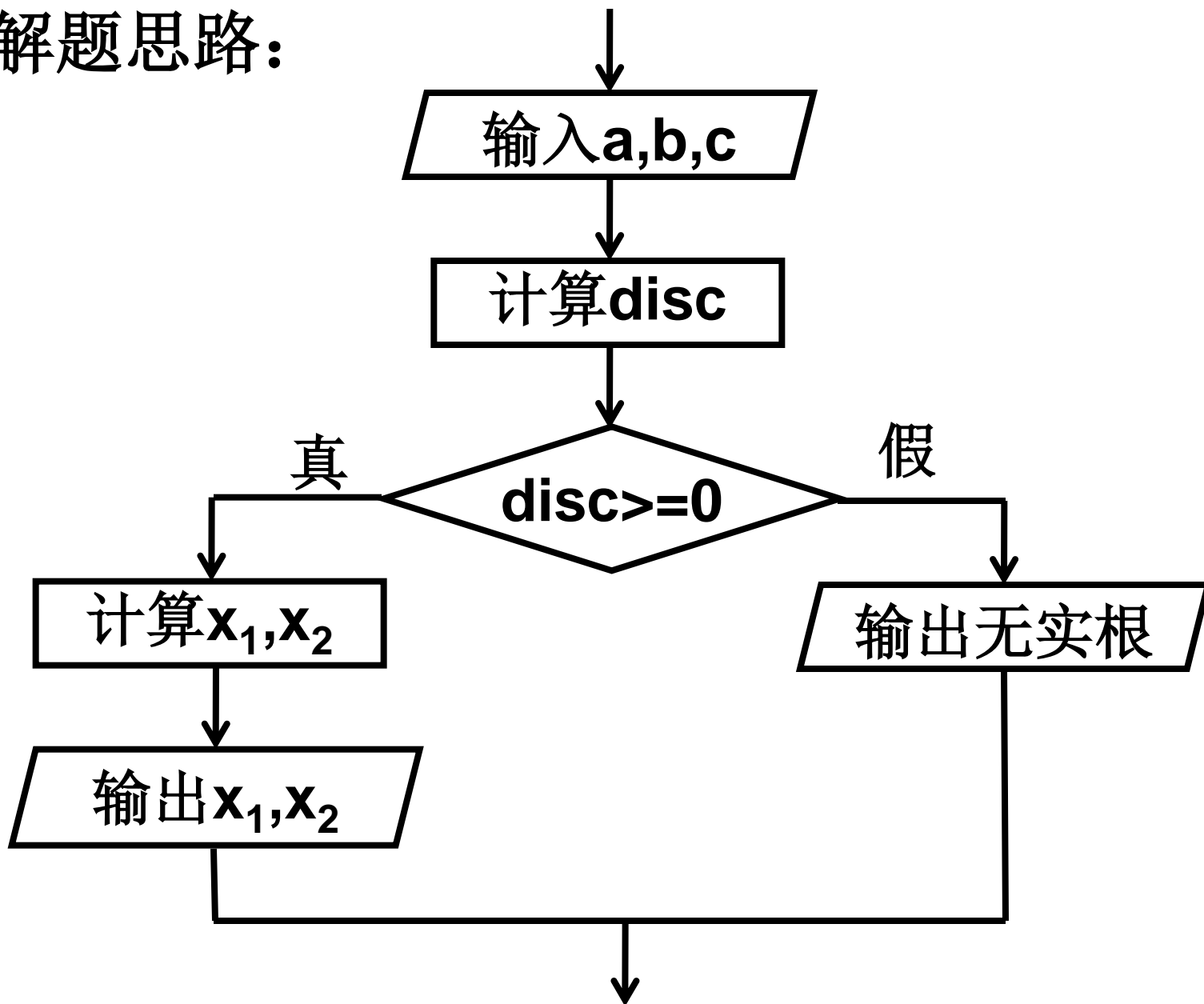
4.1 选择结构和条件判断

例**4.1** 在例**3.5**的基础上对程序进行改进。
题目要求是求 $ax^2 + bx + c = 0$ 方程的根。

由键盘输入**a,b,c**。假设**a,b,c**的值任意，并不保证 $b^2 - 4ac \geq 0$ 。需要在程序中进行判别，如果 $b^2 - 4ac \geq 0$ ，就计算并输出方程的两个实根，否则就输出“方程无实根”的信息。



➤ 解题思路:



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

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

```
int main ( )
```

```
{
```

```
    double a,b,c,disc,x1,x2,p,q;
```

```
    scanf("%lf%lf%lf",&a,&b,&c); 6 3 1
```

```
    disc=b*b-4*a*c;
```

计算 b^2-4ac ， disc的值变为-15



if (disc<0) -15<0为真

printf("has not real roots\n");

else

has not real roots

{ p=-b/(2.0*a);

q=sqrt(disc)/(2.0*a);

x1=p+q;

x2=p-q;

printf("real roots:\nx1=%7.2f\n
x2=%7.2f\n",x1,x2);

}

return 0;

}



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

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

```
int main ( )
```

```
{
```

```
double a,b,c,disc,x1,x2,p,q;
```

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

2 4 1

```
disc=b*b-4*a*c;
```

计算 b^2-4ac ， disc的值变为8



```
if (disc<0)      8<0为假
    printf("has not real roots\n");
```

```
else
```

```
{ p=-b/(2.0*a);    p的值变为-1
  q=sqrt(disc)/(2.0*a); q的值变为0.71
  x1=p+q;         x1的值变为-0.29
  x2=p-q;         x2的值变为-1.71
  printf("real roots:\nx1=%7.2f\n
        x2=%7.2f\n",x1,x2);
```

```
}
```

```
return 0;
```

```
}
```

```
real roots:
x1=  -0.29
x2=  -1.71
```




```
if (disc<0)
    printf("has not real roots\n");
else
{
    p=-b/(2.0*a);
    q=sqrt(disc)/(2.0*a);
    x1=p+q;
    x2=p-q;
    printf("real roots:\nx1=%7.2f\n
           x2=%7.2f\n",x1,x2);
}
```

```
return 0;
```

```
}
```

选择结构，用if语句实现的



```
if (disc<0)
    printf("has not real roots\n");
```

```
else
```

```
{  p=-b/(2.0*a);
    q=sqrt(disc)/(2.0*a);
    x1=p+q;
    x2=p-q;
    printf("real roots:\nx1=%7.2f\n
           x2=%7.2f\n",x1,x2);
}
```

```
return 0;
```

```
}
```

复合语句



4.2 用if语句实现选择结构

4.2.1 用if语句处理选择结构举例

4.2.2 if语句的一般形式



4.2.1 用if语句处理选择结构举例

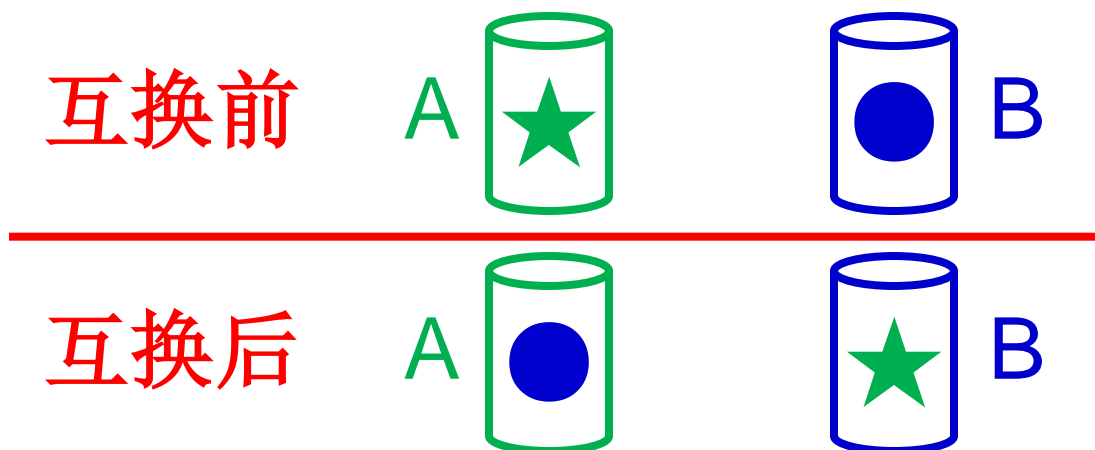
例4.2 输入两个实数，按数值由小到大的顺序输出这两个数。



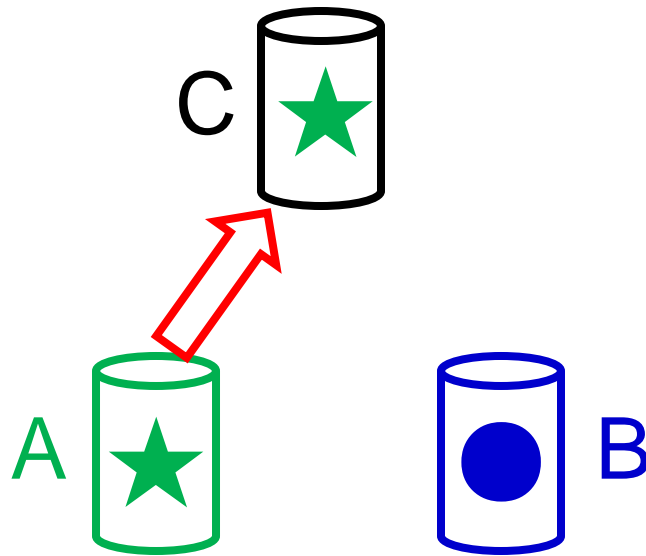
4.2.1 用if语句处理选择结构举例

➤ 解题思路:

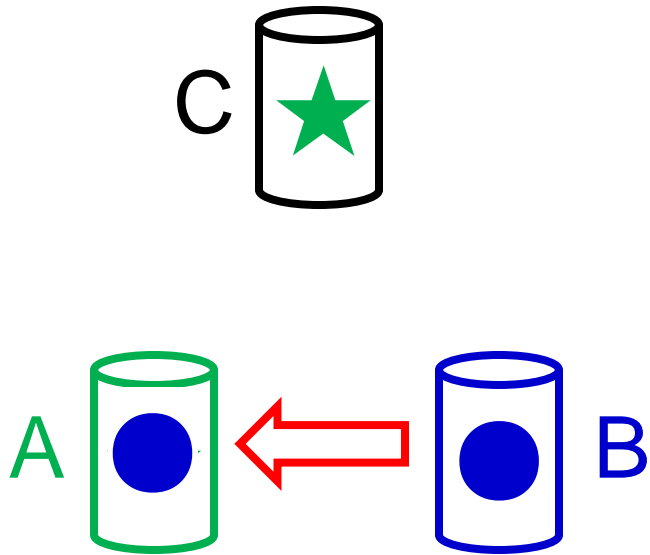
- ◆ 只需要做一次比较，然后进行一次交换即可
- ◆ 用**if**语句实现条件判断
- ◆ 关键是怎样实现两个变量值的互换



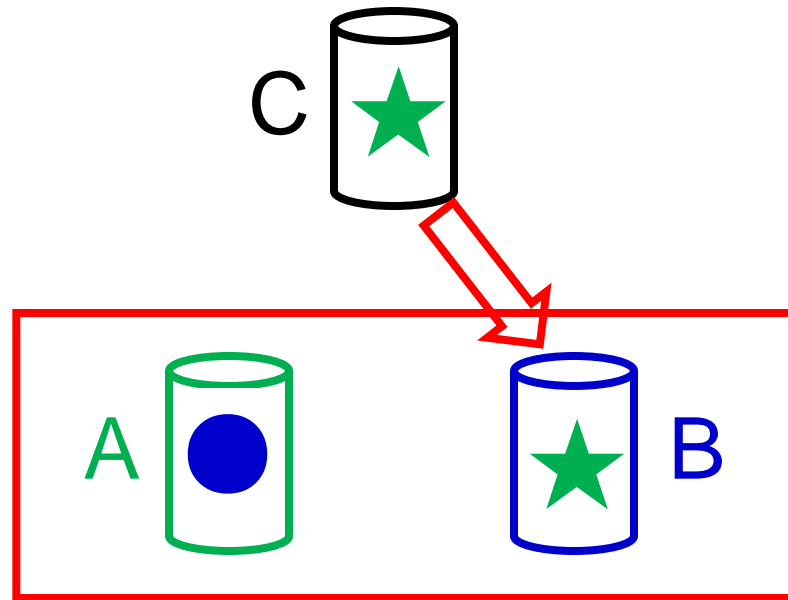
4.2.1 用if语句处理选择结构举例



4.2.1 用if语句处理选择结构举例



4.2.1 用if语句处理选择结构举例




```
#include <stdio.h>
int main()
{ float a,b,t;
  scanf("%f,%f",&a,&b);
  if(a>b) 如果a>b
  {
    t=a;
    a=b;
    b=t;
  }
  printf("%5.2f,%5.2f\n",a,b);
  return 0;
}
```

将a和b的值互换

```
3.6,-3.2
-3.20, 3.60
```



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

```
int main()
```

```
{ float a,b,t;
```

```
    scanf("%f,%f",&a,&b);
```

```
    if(a>b)
```

```
    { t=a;
```

```
      a=b;
```

```
      b=t;
```

```
    }
```

选择结构，用if语句实现的

```
    printf("%5.2f,%5.2f\n",a,b);
```

```
    return 0;
```

```
}
```



例4.3 输入**3**个数**a**，**b**，**c**，要求按由小到大的顺序输出。



➤ 解题思路： 可以先用伪代码写出算法：

◆ **if $a > b$** , **a**和**b**对换 (**a**是**a**、**b**中的小者)

◆ **if $a > c$** , **a**和**c**对换 (**a**是三者中最小者)

◆ **if $b > c$** , **b**和**c**对换 (**b**是三者中次小者)

◆ 顺序输出**a**, **b**, **c**



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

```
int main()
```

```
{ float a,b,c,t;
```

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

```
if(a>b)    如果 a>b, 将a和b对换
```

```
{ t=a; a=b; b=t; } a是a、b中的小者
```

```
if(a>c)
```

```
{ t=a; a=c; c=t; }
```

```
if(b>c)
```

```
{ t=b; b=c; c=t; }
```

```
printf("%5.2f,%5.2f,%5.2f\n",a,b,c);
```

```
return 0;
```

```
}
```



```
#include <stdio.h>
int main()
{ float a,b,c,t;
  scanf("%f,%f,%f",&a,&b,&c);
  if(a>b)
  { t=a; a=b; b=t; }
  if(a>c)    如果 a>c, 将a和c对换
  { t=a; a=c; c=t; }  a是三者中的小者
  if(b>c)
  { t=b; b=c; c=t; }
  printf("%5.2f,%5.2f,%5.2f\n",a,b,c);
  return 0;
}
```



```
#include <stdio.h>
int main()
{ float a,b,c,t;
  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)    如果 b>c, 将b和c对换
  { t=b; b=c; c=t; }    b是三者中的次小者
  printf("%5.2f,%5.2f,%5.2f\n",a,b,c);
  return 0;
}
```

3,7,1

1.00, 3.00, 7.00



4.2.2 if语句的一般形式

if (表达式) 语句1
[else 语句2]

关系表达式
逻辑表达式
数值表达式

方括号内的部分为可选的



4.2.2 if语句的一般形式

if (表达式) **语句1**

[else 语句2]

简单的语句
复合语句
另一个if语句等



最常用的**3**种**if**语句形式:

1. if (表达式) 语句1 (没有**else**子句)

2. if (表达式) 语句1
else 语句2 (有**else**子句)

3. 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m+1

(在**else**部分又嵌套了多层的**if**语句)



(1)if（表达式） 语句1

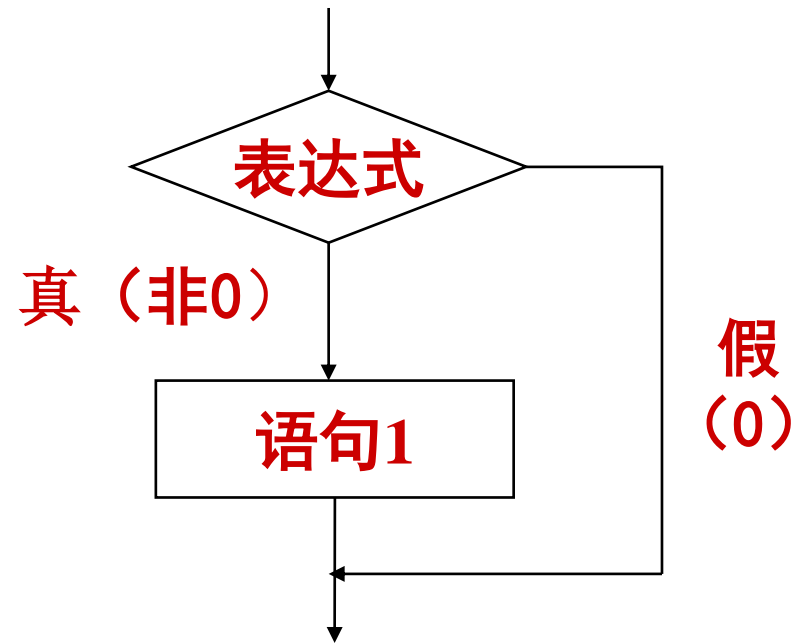
例：

```
if(x>y)
{
```

```
printf("%d",x);
x=0;
```

```
}
```

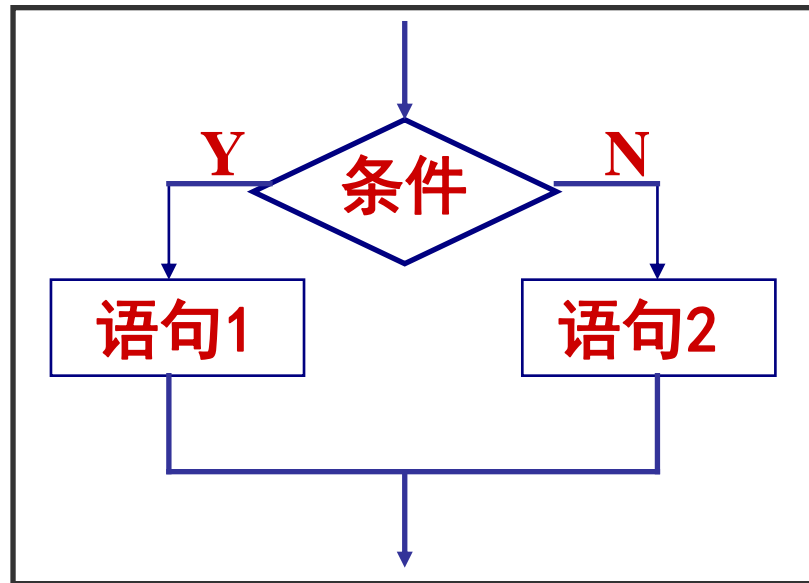
.....



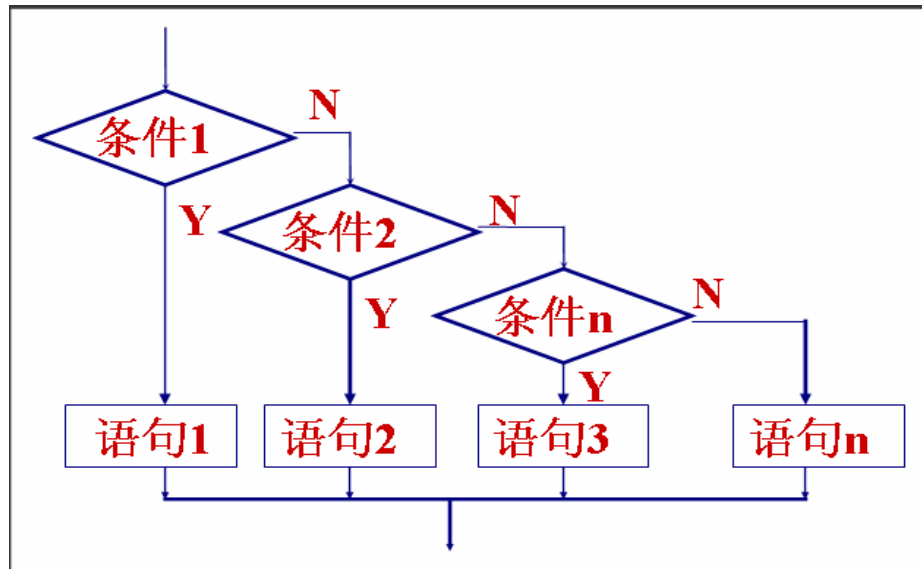
**(2)if(表达式) {语句1}
 else {语句2}**

例:

**if (x>y) printf("%d",x);
else printf("%d",y);**



(3)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



```
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; 等价于
```

```
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;
```

分号不能丢



➤说明:

- (1)**整个**if**语句可写在多行上，也可写在一行上
但都是一个整体，属于同一个语句
- (2)**“语句**1**” ... “语句**m**”是**if**中的内嵌语句
内嵌语句也可以是一个**if**语句
- (3)**“语句**1**” ... “语句**m**”可以是简单的语句
，也可以是复合语句



4.3关系运算符和关系表达式

4.3.1关系运算符及其优先次序

4.3.2 关系表达式



4.3.1 关系运算符及其优先次序

➤ 关系运算符:

用来对两个数值进行比较的比较运算符

➤ C 语言提供 6 种关系运算符:

① $<$ (小于) ② \leq (小于或等于)

③ $>$ (大于) ④ \geq (大于或等于)

优先级相同 (高)

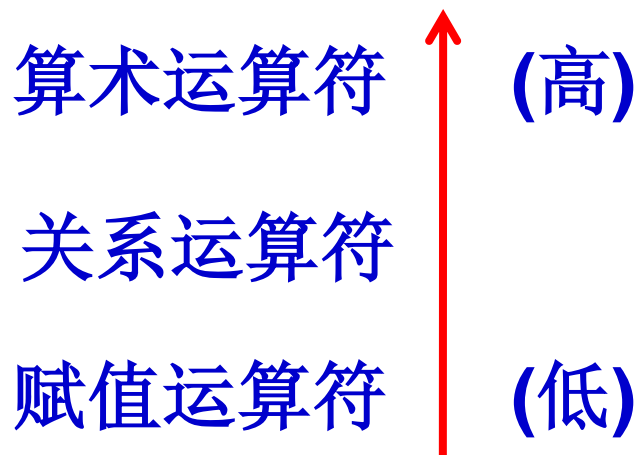
⑤ $==$ (等于) ⑥ $!=$ (不等于)

优先级相同 (低)



4.3.1 关系运算符及其优先次序

➤ 关系、算术、赋值运算符的优先级



4.3.1 关系运算符及其优先次序

Why?

$c > a + b$ 等效于 **$c > (a + b)$**

$a > b == c$ 等效于 **$(a > b) == c$**

$a == b < c$ 等效于 **$a == (b < c)$**

$a = b > c$ 等效于 **$a = (b > c)$**



4.3.2 关系表达式

➤ 关系表达式

- ◆ 用关系运算符将两个数值或数值表达式连接起来的式子
- ◆ 关系表达式的值是一个逻辑值，即“真”或“假”
- ◆ 在C的逻辑运算中，以“1”代表“真”，以“0”代表“假”



4.4 逻辑运算符和逻辑表达式

4.4.1 逻辑运算符及其优先次序

4.4.2 逻辑表达式

4.4.3 逻辑型变量



4.4.1 逻辑运算符及其优先次序

➤ 3种逻辑运算符:

&& (逻辑与) **||** (逻辑或) **!** (逻辑非)

➤ **&&**和**||**是双目(元)运算符

➤ **!**是一目(元)运算符

➤ 逻辑表达式

◆ 用逻辑运算符将关系表达式或其他逻辑量连接起来的式子



4.4.1 逻辑运算符及其优先次序

➤ 判断年龄是否在**13**至**17**岁之间

age >= 13 && age <= 17

➤ 判断年龄是否小于**12**或大于**65**?

age < 12 || age > 65



4.4.1 逻辑运算符及其优先次序

➤ 逻辑运算的真值表（真1，假0）

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

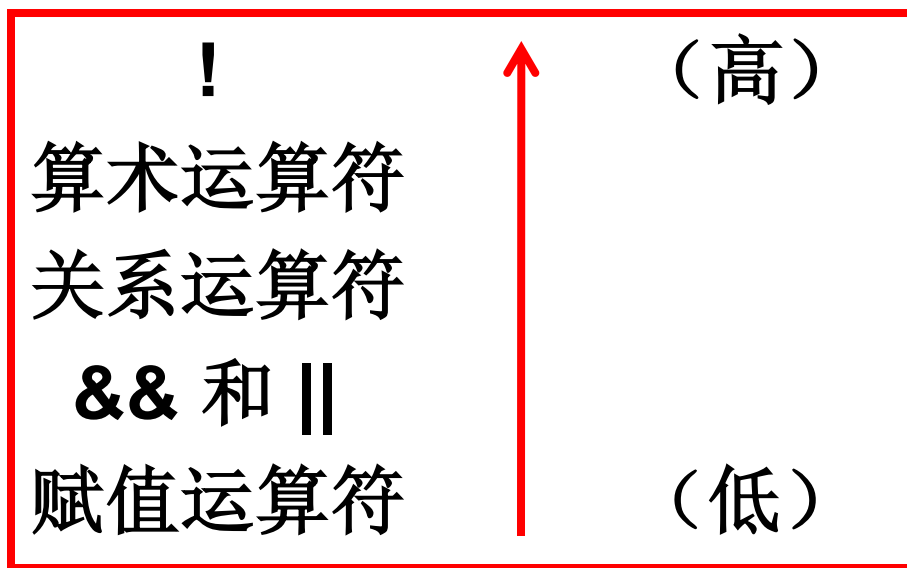


4.4.1 逻辑运算符及其优先次序

➤ 逻辑运算符的优先次序

! → && → || (!为三者中最高)

➤ 与其他运算符的优先次序



4.4.2 逻辑表达式

- 逻辑表达式的值应该是逻辑量“真”或“假”
- 编译系统在表示逻辑运算结果时
 - ◆ 以数值**1**代表“真”，以**0**代表“假”
- 但在判断一个量是否为“真”时
 - ◆ 以**0**代表“假”，以非**0**代表“真”
- 注意：将一个非零的数值认作为“真”



4.4.2 逻辑表达式

(1) 若 $a=4$ ，则 $!a$ 的值为0

(2) 若 $a=4$ ， $b=5$ ，则 $a \&\& b$ 的值为1

(3) a 和 b 值分别为4和5，则 $a||b$ 的值为1

(4) a 和 b 值分别为4和5，则 $!a||b$ 的值为1

(5) $4 \&\& 0 || 2$ 的值为1

(6) 让 x 的结果 在 a ， b 同真，或 c ， d 同真时，
为真，其余情况为假

$x = (a\&\&b)|| (c\&\&d)$



4.4.2 逻辑表达式

➤ 修改后的逻辑运算真值表

a	b	! a	! b	a && b	a b
非0	非0	0	0	1	1
非0	0	0	1	0	1
0	非0	1	0	0	1
0	0	1	1	0	0



4.4.2 逻辑表达式

- 判别某一年是否闰年，用逻辑表达式表示
- 闰年的条件是符合下面二者之一：

① 能被**4**整除，但不能被**100**整除，如**2008**

② 能被**400**整除，如**2000**

◆ **$(\text{year \% } 4 == 0 \ \&\& \ \text{year \% } 100 != 0)$**
 $|| \ \text{year \% } 400 == 0$

◆ 如果表达式值为**1**，则闰年；否则为非闰年



4.5 条件运算符和条件表达式

- 有一种**if**语句，当被判别的表达式的值为“真”或“假”时，都执行一个赋值语句且向同一个变量赋值

➤ 如：

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

条件运算符

$\text{max} = (\text{a} > \text{b}) ? \text{a} : \text{b};$



4.5 条件运算符和条件表达式

➤ 有一种**if**语句，当被判别的表达式的值为“真”或“假”时，都执行一个赋值语句且向同一个变量赋值

➤ 如：**if (a>b)**

```
        max=a;  
else  
        max=b;
```

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

条件表达式



4.5 条件运算符和条件表达式

➤ 条件表达式的一般形式为

表达式 1 ? 表达式 2 : 表达式 3



4.5 条件运算符和条件表达式

➤ 条件运算符的执行顺序：

◆ 求解表达式**1**

◆ 若为非**0**（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

◆ 若表达式**1**的值为**0**（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



4.5 条件运算符和条件表达式

- 条件运算符优先于赋值运算符
- 条件运算符的结合方向为“自右至左”
- 以下为合法的使用方法：
 - ◆ **`a > b ? (max=a):(max=b);`**
 - ◆ **`a > b ? printf("%d",a): printf("%d",b);`**



4.5 条件运算符和条件表达式

例4.4 输入一个字符，判别它是否大写字母，如果是，将它转换成小写字母；如果不是，不转换。然后输出最后得到的字符。



4.5 条件运算符和条件表达式

- 解题思路：用条件表达式来处理，当字母是大写时，转换成小写字母，否则不转换



4.5 条件运算符和条件表达式

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

```
int main()
```

```
{
```

```
    char ch;
```

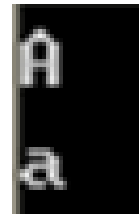
```
    scanf("%c",&ch);
```

```
    ch=(ch>='A' && ch<='Z')?(ch+32):ch;
```

```
    printf("%c\n",ch);
```

```
    return 0;
```

```
}
```



A
a



4.6 选择结构的嵌套

➤ 在**if**语句中又包含一个或多个**if**语句称为**if**语句的嵌套

➤ 一般形式:

if()

if() 语句1
else 语句2

else

if() 语句3
else 语句4

在同一个花括号内，**else**总是与它上面最近的未配对的**if**配对

内嵌**if**



4.6 选择结构的嵌套

- 在**if**语句中又包含一个或多个**if**语句称为**if**语句的嵌套

if ()

{

if () 语句**1**

}

else语句**2**

内嵌**if**

{ }限定了内嵌**if**范围



4.6 选择结构的嵌套

例4.5有一函数：

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

编一程序，输入一个**x**值，要求输出相应的**y**值。



4.6 选择结构的嵌套

➤ 解题思路:

- ◆ 用**if**语句检查**x**的值，根据**x**的值决定赋予**y**的值
- ◆ 由于**y**的可能值不是两个而是三个，因此不可能只用一个简单的(无内嵌**if**)的**if**语句来实现



4.6

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

```
if(x<0)    y = -1;
```

```
if(x==0)   y = 0;
```

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

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

➤ 解题思路

(1) 先后用5个独立的if语句处理:

输入x

若 $x < 0$, 则 $y = -1$

若 $x = 0$, 则 $y = 0$

若 $x > 0$, 则 $y = 1$

输出x和y

```
-5
x=-5,y=-1
```



4.6

➤ 解题思路

(2) 用一

输入 x

若 $x < 0$, 则 $y = -1$

否则

若 $x = 0$, 则 $y = 0$

否则 $y = 1$

输出 x 和 y

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

```
if(x<0) y=-1;
```

```
else
```

```
    if(x==0) y=0;
```

```
    else y=1;
```

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

```
-5
x=-5,y=-1
```

```
5
x=5,y=1
```



4.6

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

```
if(x<0) y=-1;
```

```
else
```

提倡内嵌if放在else中

```
if(x==0) y=0;
```

```
else y=1;
```

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

➤ 解题思路

(2) 用一

输入x

若 $x < 0$ 则 $y = -1$

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

```
if (x>=0)
```

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

```
    else y=0;
```

```
else y=-1;
```

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



4.7 用switch语句实现多分支选择结构

➤ **switch**语句用来实现多分支选择结构

◆ 学生成绩分类

85分以上为'**A**'等

70~84分为'**B**'等

60~69分为'**C**'等

.....

◆ 人口统计分类

按年龄分为老、中、青、少、儿童



➤ **switch**语句的作用是根据表达式的值，使流程跳转到不同的语句

➤ **switch**语句的一般形式：

switch (表达式) 整数类型(包括字符型)

{ **case** 常量1 : 语句1

case 常量2 : 语句2

 :
 :
 :

case 常量n : 语句n

default : 语句n+1

}



➤ **switch**语句的作用是根据表达式的值，使流程跳转到不同的语句

➤ **switch**语句的一般形式：

switch (表达式)

```
{ case 常量1 : 语句1
    case 常量2 : 语句2
      ⋮      ⋮      ⋮      不能相同
    case 常量n : 语句n
    default   : 语句n+1
}
```



说明:

- (1)**当表达式的值与某一个**case**后面的常量表达式的值相等时，就执行此**case**后面的语句，若所有的**case**中的常量表达式的值都没有与表达式的值匹配的，就执行**default**后面的语句。
- (2)**每一个**case**的常量表达式的值必须互不相同，否则就会出现互相矛盾的现象（对表达式的同一个值，有两种或多种执行方案）。
- (3)**各个**case**和**default**的出现次序不影响执行结果。例如，可以先出现“**default: ...**”，再出现**case' D': ...**，然后是**case' A: ...**。

(4) 执行完一个**case**后面的语句后，流程控制转移到下一个**case**继续执行。

“**case**常量表达式”只是起语句标号作用，并不是在条件判断。在执行 **switch**语句时，根据**switch**后面表达式的值找到匹配的入口标号，从此标号开始执行下去，不再进行判断。应该在每一个**case**后，用一个**break**语句来终止该语句，否则自动执行下一个**case**语句。

(5) 多个**case**语句可以共用一组执行语句。

```
scanf("%c",&grade);  
printf("Your score:");  
switch(grade)
```

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

```
A  
Your score:85~100  
70~84  
60~69  
<60  
enter data error!
```



```
scanf("%c",&grade);  
printf("Your score:");  
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("enter data error!\n");  
}
```



```
scanf("%c",&grade);  
printf("Your score:");  
switch(grade)  
{ case 'A':  
  case 'B':  
  case 'C': printf("60~69\n");break;  
  case 'D': printf("<60\n");break;  
  default: printf("enter data error!\n");  
}
```

```
A  
Your score:60~69
```



4.7 用switch语句实现多分支选择结构

例4.6 要求按照考试成绩的等级输出百分制分数段，**A**等为**85**分以上，**B**等为**70~84**分，**C**等为**60~69**分，**D**等为**60**分以下。成绩的等级由键盘输入。



4.7 用switch语句实现多分支选择结构

➤ 解题思路:

- ◆ 判断出这是一个多分支选择问题
- ◆ 根据百分制分数将学生成绩分为**4**个等级
- ◆ 如果用**if**语句，至少要用**3**层嵌套的**if**，进行**3**次检查判断
- ◆ 用**switch**语句进行一次检查即可得到结果



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

```
int main()
```

```
{ char grade;
```

```
scanf("%c",&grade);
```

```
printf("Your score:");
```

```
switch(grade)  值为A
```

```
{ 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("enter data error!\n");
```

```
}
```

```
return 0;
```

```
}
```

```
A  
Your score:85~100
```



```
#include <stdio.h>
int main()
{ char grade;
  scanf("%c",&grade);
  printf("Your score:");
  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("enter data error!\n");
  }
  return 0;
}
```

不能少




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

```
int main()
```

```
{ char grade;
```

```
scanf("%c",&grade);
```

```
printf("Your score:");
```

```
switch(grade)  值为C
```

```
{ 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("enter data error!\n");
```

```
}
```

```
return 0;
```

```
}
```

```
C  
Your score:60~69
```



```
#include <stdio.h>
int main()
{ char grade;
  scanf("%c",&grade);
  printf("Your score:");
  switch(grade)  值为F
  { 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("enter data error!\n");
  }
  return 0;
}
```

```
F
Your score:enter data error!
```



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

```
int main()
```

```
{ char grade;
```

```
scanf("%c",&grade);
```

```
printf("Your score:");
```

此行位置有问题，
应如何修改？

```
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("enter data error!\n");
```

```
}
```

```
return 0;
```

```
printf("Your score:");
```

```
}
```



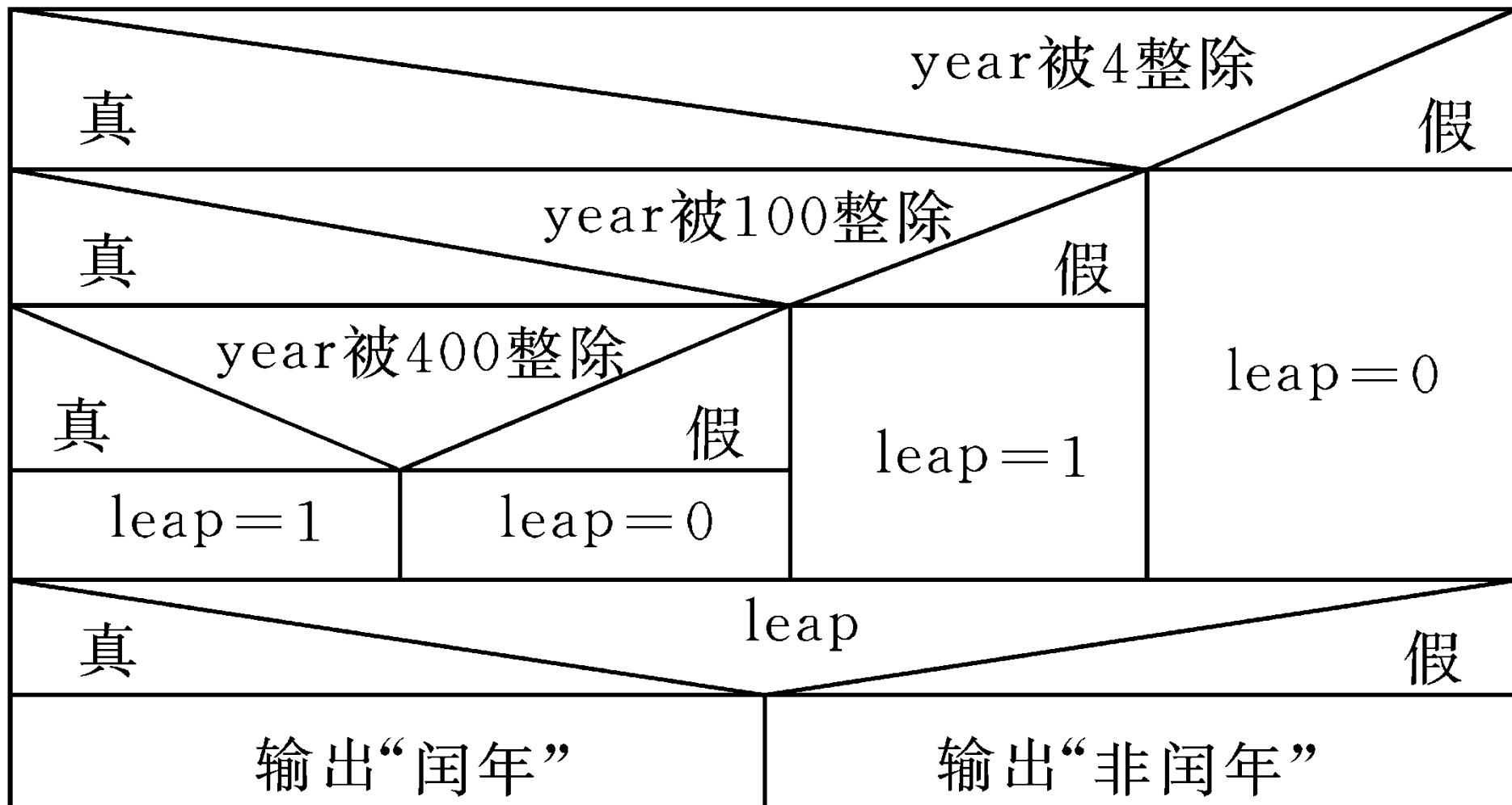
4.8选择结构程序综合举例

例4.8 写一程序，判断某一年是否闰年。

- 解题思路：在前面已介绍过判别闰年的规则
- 用变量**leap**代表是否闰年的信息。若闰年，令**leap=1**；非闰年，**leap=0**。最后判断**leap**是否为1（真），若是，则输出“闰年”信息



4.8选择结构程序综合举例



```
#include <stdio.h>
int main()
{int year,leap; 标志变量
    printf("enter year:"); scanf("%d",&year);
    if (year%4==0)
        if(year%100==0)
            if(year%400==0)    leap=1;
            else    leap=0;
        else    leap=1;
    else    leap=0;    与if (leap!=0)含义相同
    if (leap)    printf("%d is ",year);
    else    printf("%d is not ",year);
    printf("a leap year.\n");
    return 0;
}
```



```
2012
2012 is a leap year.
```

```
> 2100
2100 is not a leap year.
```

```
{int year, leap;
    printf("enter year:"); scanf("%d",&year);
    if (year%4==0)
        if(year%100==0)
            if(year%400==0)    leap=1;
            else    leap=0;
        else    leap=1;
    else    leap=0;
    if (leap)    printf("%d is ",year);
    else    printf("%d is not ",year);
    printf("a leap year.\n");
    return 0;
}
```

采取锯齿形式



```

#include <stdio.h>
int main()
{int year, leap;
  printf("enter year:"); scanf("%d",&year);
  if (year%4==0)
    if(year%100==0)
      if(year%400==0)
        leap=1;
      else
        leap=0;
    else if (year%100!=0)
      leap=0;
    else if(year%400!=0)
      leap=0;
    else
      leap=1;
  if (leap)
    printf("leap year");
  else printf("not leap year");
  printf("\n");
  return 0;
}

```




```
#include <stdio.h>
int main()
{int year, leap;
  printf("enter year:"); scanf("%d",&year);
  if (year%4==0)
    if(year%100==0)
      if(year%400==0)    leap=1;
      else    leap=0;
    else    leap=1;
  else    leap=0;
```

```
if((year%4==0 && year%100!=0)
    || (year%400==0))
  leap=1;
else
  leap=0;
}
```



练习题

year 被 100 整除			
真			假
year 被 400 整除		year 被 4 整除	
真	假	真	假
leap=1	leap=0	leap=1	leap=0

例**4.9** 求 $ax^2 + bx + c = 0$ 方程的解。

➤解题思路：处理以下各情况

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。

应当以 $p+qi$ 和 $p-qi$ 的形式输出复根

其中， $p = -b/2a$ ， $q = (\sqrt{b^2 - 4ac})/2a$

➤参见教材图4.14



真	<div> <div>$a=0$</div> <div>假</div> </div>		
<div>输出“非二次方程”</div>	真	<div> <div>$b^2-4ac=0$</div> <div>假</div> </div>	
	<div> <div>输出两个相等实根：</div> <div>$-\frac{b}{2a}$</div> </div>	真	<div> <div>$b^2-4ac>0$</div> <div>假</div> </div>
		<div> $x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ $x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$ </div>	<div> <div>计算复根的实部和虚部：</div> <div>实部 $p = -\frac{b}{2a}$</div> <div>虚部 $q = \frac{\sqrt{-(b^2 - 4ac)}}{2a}$</div> </div>
		<div> <div>输出两个实根</div> <div>$x_1、x_2$</div> </div>	<div> <div>输出两个复根：</div> <div>$p+qi,$</div> <div>$p-qi$</div> </div>

```
#include <stdio.h>  
#include <math.h>  
int main()  
{  
    double a,b,c,disc,x1,x2,realpart,  
                                     imagpart;  
    scanf("%lf,%lf,%lf",&a,&b,&c);  
    printf("The equation ");  
    if(fabs(a)<=1e-6) 实型不能用 if (a==0)  
        printf("is not a quadratic\n");
```



else

{disc=b*b-4*a*c; 先算**disc**，以减少重复计算

if(fabs(disc)<=1e-6) 不能用**if (disc==0)**

**printf("has two equal roots:%8.4f\n",
-b/(2*a));**

else



```
if(disc>1e-6)  
{x1=(-b+sqrt(disc))/(2*a);  
  x2=(-b-sqrt(disc))/(2*a);  
  printf("has distinct real roots:%8.4f  
        and %8.4f\n",x1,x2);  
}  
else
```



```

{ realpart=-b/(2*a);
  imagpart=sqrt(-disc)/(2*a);
  printf(" has complex roots:\n");
  printf("%8.4f+%8.4fi\n"
          ,realpart,imagpart);
  printf("%8.4f-%8.4fi\n",
          realpart,imagpart);

```

```

}
1,2,1

```

```

The equation has two equal roots: -1.0000
return 0;

```

```

}

```




```

{ realpart=-b/(2*a);
  imagpart=sqrt(-disc)/(2*a);
  printf(" has complex roots:\n");
  printf("%8.4f+%8.4fi\n",
          realpart,imagpart);
  printf("%8.4f-%8.4fi\n",
          realpart,imagpart);

```

```

1.2.2
The equation has complex roots:
-1.0000+ 1.0000i
-1.0000- 1.0000i
}

```



```

{ realpart=-b/(2*a);
  imagpart=sqrt(-disc)/(2*a);
  printf(" has complex roots:\n");
  printf("%8.4f+%8.4fi\n",
          ,realpart,imagpart);
  printf("%8.4f-%8.4fi\n",
          realpart,imagpart);

```

```
2,6,1
```

```
The equation has distinct real roots: -0.1771 and -2.8229
```

```

}
return 0;

```

```
}
```



例4.10 运输公司对用户计算运输费用。
路程(**s km**) 越远, 每吨·千米运费越低。

➤ 标准如下:

$s < 250$	没有折扣
$250 \leq s < 500$	2%折扣
$500 \leq s < 1000$	5%折扣
$1000 \leq s < 2000$	8%折扣
$2000 \leq s < 3000$	10%折扣
$3000 \leq s$	15%折扣



➤ 解题思路:

◆ 设每吨每千米货物的基本运费为**p**，货物重为**w**，距离为**s**，折扣为**d**

◆ 总运费**f**的计算公式为 **$f = p \times w \times s \times (1 - d)$**



➤ 折扣的变化规律（参见教材图**4.15**）：

- ◆ 折扣的“变化点”都是**250**的倍数
- ◆ 在横轴上加一种坐标**c**，**c**的值为 **$s/250$**
- ◆ **c**代表**250**的倍数
- ◆ 当 **$c < 1$** 时，表示 **$s < 250$** ，无折扣
- ◆ **$1 \leq c < 2$** 时，表示 **$250 \leq s < 500$** ，折扣 **$d = 2\%$**
- ◆ **$2 \leq c < 4$** 时， **$d = 5\%$** ； **$4 \leq c < 8$** 时， **$d = 8\%$** ；
 $8 \leq c < 12$ 时， **$d = 10\%$** ； **$c \geq 12$** 时， **$d = 15\%$**



```
#include <stdio.h>
int main()
{
    int c,s;
    float p,w,d,f;
    printf("please enter
    price,weight,discount:");
    scanf("%f,%f,%d",&p,&w,&s);
    if(s>=3000) c=12; 输入单价、重量、距离
    else      c=s/250;
```



switch(c)

```
{ case 0:  d=0; break;  
  case 1:  d=2; break;  
  case 2:  
  case 3:  d=5; break;  
  case 4:  
  case 5:  
  case 6:  
  case 7:  d=8; break;  
  case 8:  
  case 9:  
  case 10:  
  case 11: d=10; break;  
  case 12: d=15; break;  
}
```



```
f = p * w * s * (1 - d / 100);  
printf("freight=%10.2f\n",f);  
return 0;  
}
```

```
please enter price,weight,discount:100,20,300  
freight= 588000.00
```

