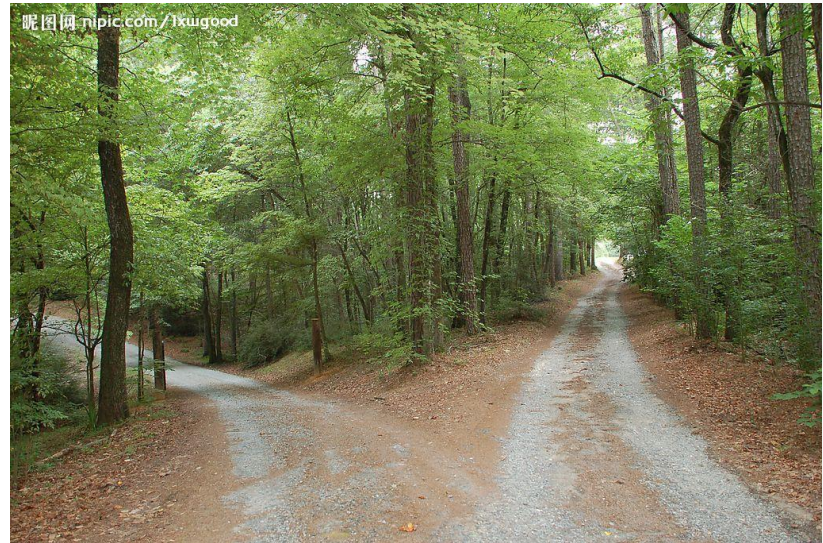


选择结构程序设计

程 淼

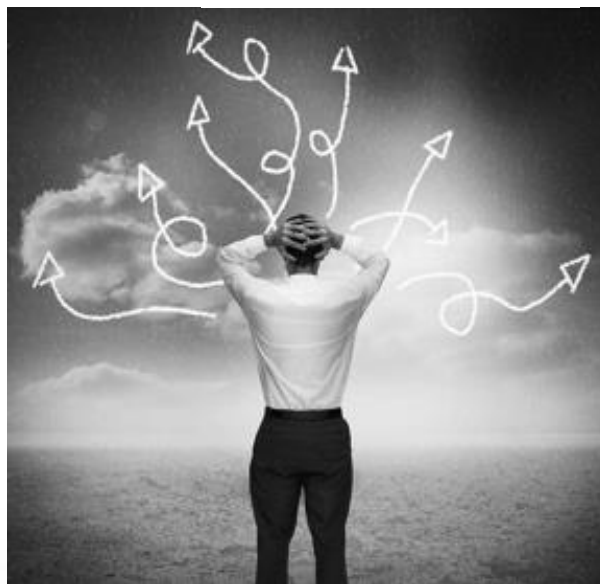
选择结构

- There are always choices !



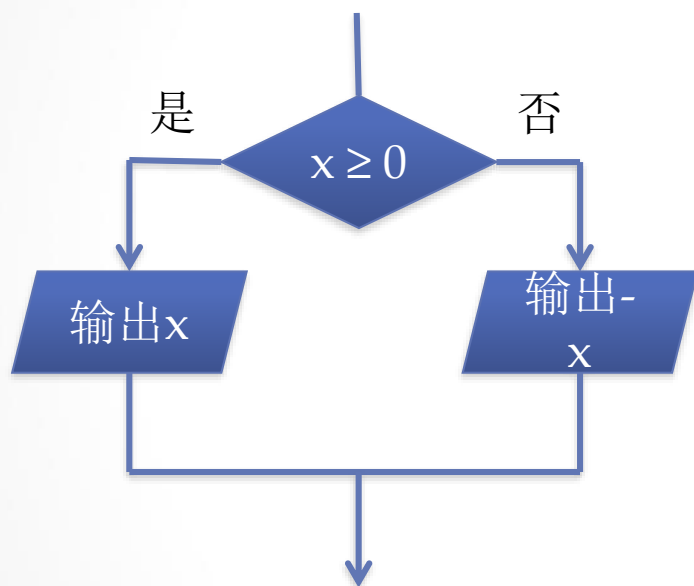
选择结构

- 逻辑世界：Yes or No ?



选择结构

- 一个例子：输入一个数，要求输出其绝对值

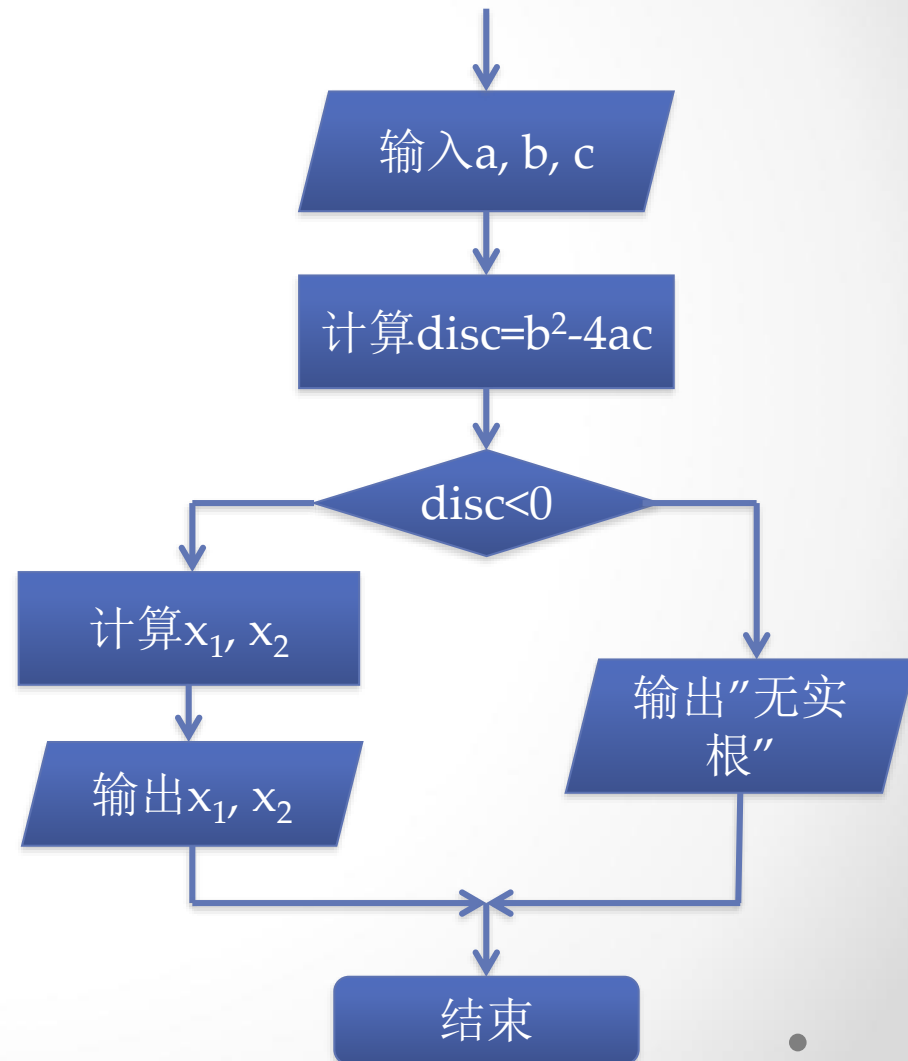


```
if (x >= 0)
    printf("%d", x);
else
    printf("%d", -x);
```



选择结构

- 再看一个例子：要求解得 $ax^2+bx+c=0$ 方程的根，由键盘输入 a, b, c 。



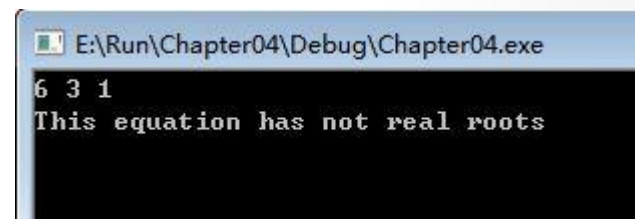
选择结构

- 再看一个例子：要求解得 $ax^2+bx+c=0$ 方程的根，由键盘输入a, b, c。

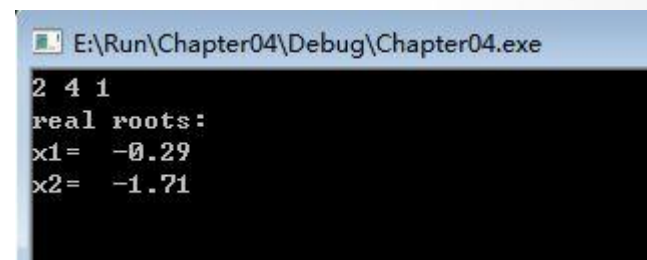
```
#include <stdio.h>
#include <math.h>

int main()
{
    double a, b, c, disc, x1, x2, p, q;
    scanf("%lf%lf%lf", &a, &b, &c);
    disc = b*b - 4*a*c;
    if (disc < 0)
        printf("This equation 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\nx2=%7.2f\n", x1, x2);
    }

    return 0;
}
```



```
E:\Run\Chapter04\Debug\Chapter04.exe
6 3 1
This equation has not real roots
```



```
E:\Run\Chapter04\Debug\Chapter04.exe
2 4 1
real roots:
x1=  -0.29
x2=  -1.71
```

if语句

- if语句的一般形式:

- if (表达式) 语句1
- [else 语句2]

if (表达式) 语句 1

if (表达式)
语句 1
else
语句 2

```
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
```

if语句

- 输入两个实数，按代数值由小到大的顺序输出这两个数
 - 将两个变量的值互换，不能把两个变量直接互相赋值
 - $a = b;$
 - $b = a;$

```
#include <stdio.h>
int main()
{
    float a, b, t;
    scanf("%f, %f", &a, &b);

    //a = b;
    if (a > b)
    {
        t = a;
        a = b;
        b = t;
    }

    printf("%.2f, %.2f\n", a, b);
    return 0;
}
```


if语句

- 输入3个数a, b, c, 要求按由小到大的顺序输出

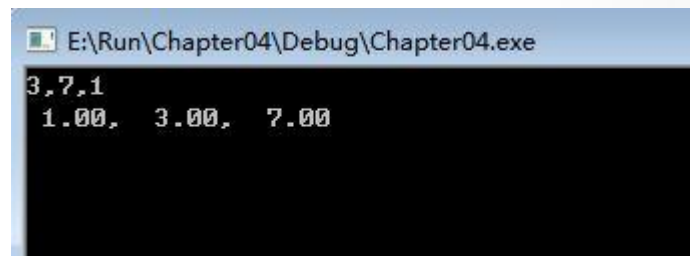
```
#include <stdio.h>

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)
    {
        t = b;
        b = c;
        c = t;
    }

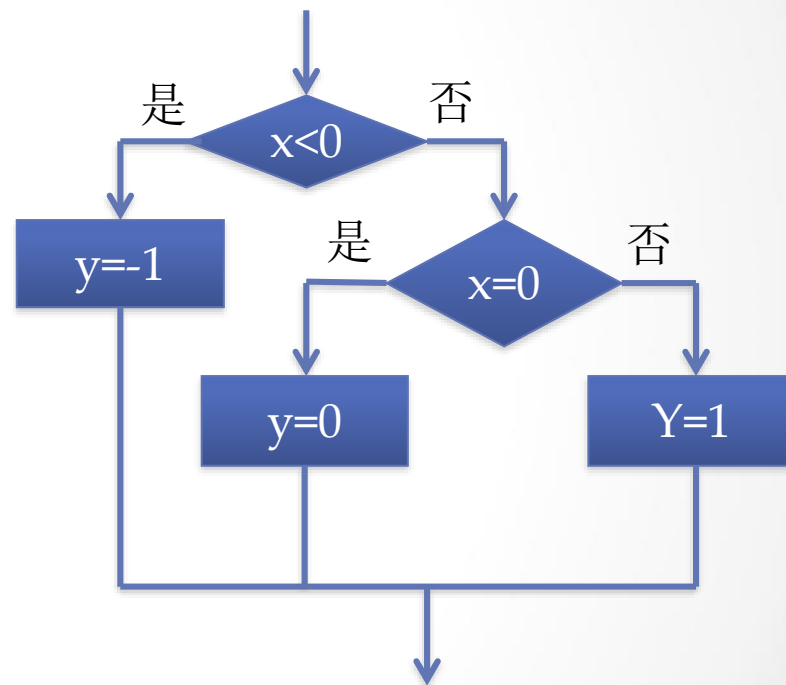
    printf("%5.2f, %5.2f, %5.2f\n", a, b, c);
    return 0;
}
```



if语句

- 用if语句表示阶跃函数

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



关系运算符


- 关系运算符及其优先级

- <
 - <=
 - >
 - >=
 - ==
 - !=
- 优先级相同(高)
- 优先级相同(低)

关系运算符

- 关系运算符的优先级低于算术运算符
- 关系运算符的优先级高于赋值运算符
 - $c > a + b$ 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 $a = (b > c)$

算术运算符	(高)
关系运算符	
赋值运算符	(低)



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，称**关系表达式**
 - $a > b$, $a + b > b + c$, $(a = 3) > (b = 5)$, $'a' < 'b'$, $(a > b) > (b < c)$
 - $(a > b) == c$
 - $b + c < a$

逻辑运算符

- 由几个给定简单条件组成的复合条件
 - 如果星期六不下雨，我去公园玩
 - 是否星期六
 - 是否下雨
 - 既满足“今天是星期六”，也满足“下雨”
 - 参加少年运动会的年龄限制为13~17岁
 - 年龄 $\text{age} \geq 13$
 - 年龄 $\text{age} \leq 17$
 - $\text{age} \geq 13 \text{ AND } \text{age} \leq 17$

逻辑运算符

- 逻辑运算符
 - 与(AND)，或(OR)，非(NOT)

运算符	含义	举例	说明
&&	逻辑与	a && b	如果a和b都为真，则结果为真，否则为假
	逻辑或	a b	如果a和b有一个以上为真，则结果为真，二者都为假时，结果为假
!	逻辑非	!a	如果a为假，则!a为真，如果a为真，则!a为假

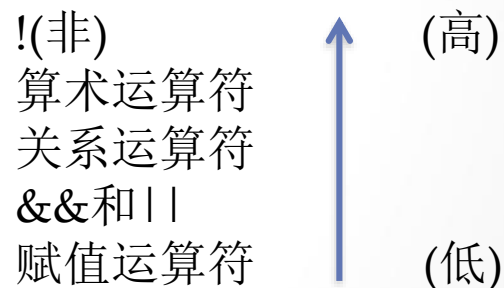
逻辑运算符

- 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

逻辑运算符

- 在一个逻辑表达式中如果包含多个逻辑运算符，按以下的优先次序：
 - !(非) \rightarrow &&(与) \rightarrow ||(或)，即"!"为三者中最高的
 - 逻辑运算符中的"&&"和"||"低于关系运算符，"! "高于算术运算符
- 例子
 - (a>b) && (x>y)
 - (a==b) && (x==y)
 - (!a) || (a>b)



逻辑表达式

- 逻辑表达式的值应该是一个逻辑量"真"或"假"
- C语言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"
 - $a=4$ ，则 $!a$ 的值为0
 - $a=4$ ， $b=5$ ，则 $a\&\&b$ 的值为1
 - a 和 b 值分别为4和5， $a||b$ 的值为1
 - a 和 b 值分别为4和5， $!a||b$ 的值为1
 - $4\&\&0||2$ 的值为1
- $5>3 \&\& 8<4-!0$

逻辑表达式

- 在逻辑表达式的求解中，并不是所有的逻辑运算符都被执行，只是在必须执行下一个逻辑运算符才能求出表达式的解时，才执行该运算符
 - `a && b && c`
 - `a || b || c`

逻辑表达式

- 判别用year表示的某一年是否闰年，可以用一个逻辑表达式来表示。闰年的条件是符合下面二者之一：
 - 能被4整除，但不能被100整除
 - 能被400整除
- `(year%4 == 0 && year%100 != 0) || year%400 == 0`

逻辑型变量

- 可以将关系运算和逻辑运算的结果存到一个逻辑型变量中，以便于分析和运算
- 定义逻辑变量用类型符_Bool

```
float score;  
scanf("%f", &score);  
_Bool a, b;  
a = score >= 60;  
b = score <= 69;  
if (a && b)    printf("The grade is C\n");
```

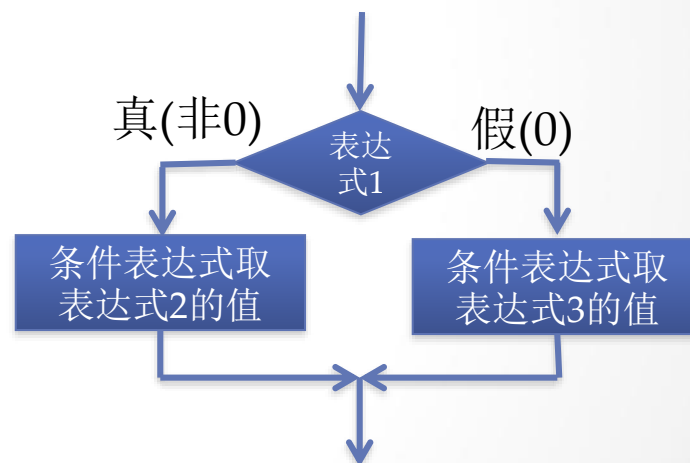
条件运算符

- 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

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

- 表达式1? 表达式2: 表达式3

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



条件运算符

- 条件运算符优先于赋值运算符
 - $\text{max} = (\text{a} > \text{b}) ? \text{a} : \text{b}$
- 条件表达式的值，把它赋给一个变量max
 - $\text{a} > \text{b} ? (\text{max} = \text{a}) : (\text{max} = \text{b})$

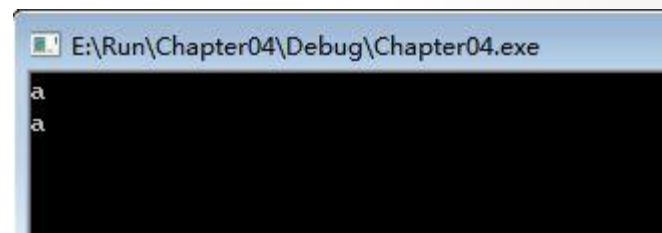
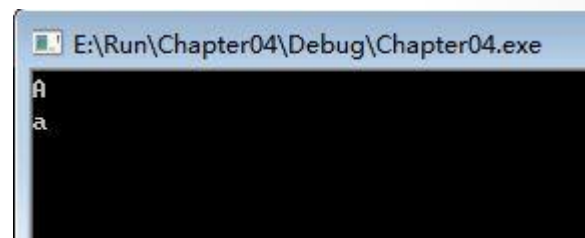
条件运算符

- 一个例子：输入一个字符，判别它是否为大写字母，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

```
#include <stdio.h>

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

    return 0;
}
```



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  
else  
    if( )  语句 3  
    else  语句 4
```

} 内嵌if

} 内嵌if

选择结构的嵌套

- 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
- 解题思路：
 - 先后用3个独立的if语句处理
 - 用一个嵌套的if语句处理

选择结构的嵌套

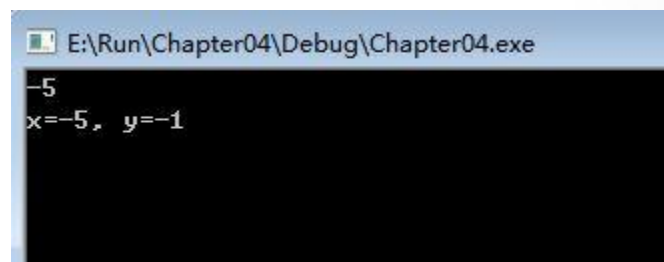
- 编写程序

```
#include <stdio.h>

int main()
{
    int x, 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);

    return 0;
}
```



```
E:\Run\Chapter04\Debug\Chapter04.exe
-5
x=-5, y=-1
```

选择结构的嵌套

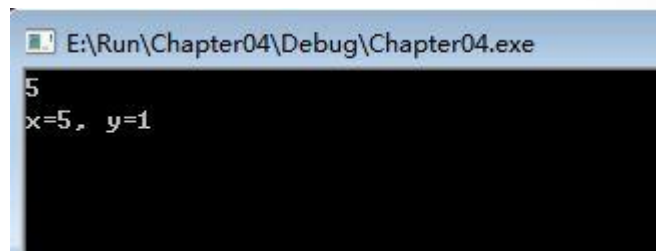
- 编写程序

```
#include <stdio.h>

int main()
{
    int x, y;
    scanf("%d", &x);

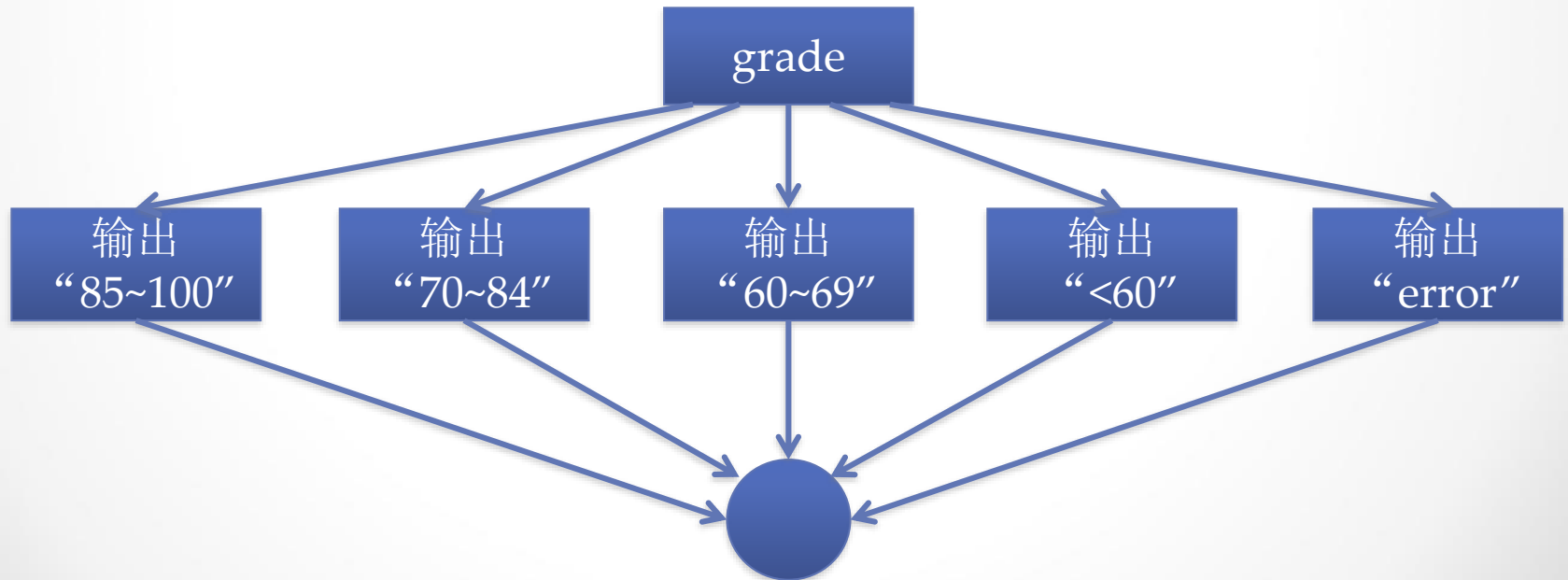
    if (x >= 0)
        if (x > 0)    y = 1;
        else y = 0;
    else    y = -1;

    printf("x=%d, y=%d\n", x, y);
    return 0;
}
```



多分支选择结构

- switch语句是多分支选择语句
- 先看一个例子：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



多分支选择结构

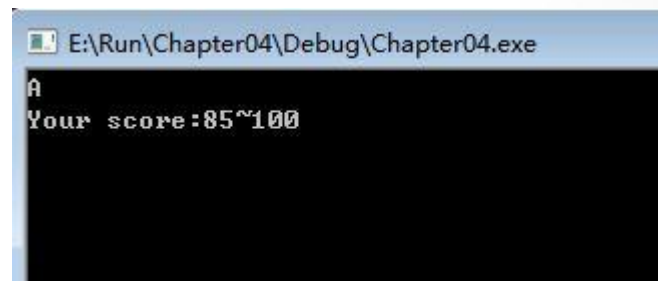
- 编写程序

```
#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;
}
```



多分支选择结构

- switch语句的一般形式如下

```
switch (表达式)
{
    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
}
```

多分支选择结构

- switch后面的括号内的"表达式", 其值的类型应为整数类型(包括字符型)
- switch下面的花括号内是一个复合语句
- 可以没有default标号
- 各个case标号出现次序不影响执行结果
- 每一个case常量必须互不相同; 否则就会出现互相矛盾的现象

多分支选择结构

- case标号只起标记的作用：根据switch表达式的值找到匹配的入口标号，并不在此进行检查
- 在case子句中虽然包含了一个以上执行语句，但可以不必用花括号括起来，回自动顺序执行本case标号后面的所有的语句
- 多个case标号可以公用一组执行语句

多分支选择结构

- 用switch语句处理菜单命令。在许多应用程序中，用菜单对流程进行控制，例如从键盘输入一个'A'或'a'字符，就会执行A操作，输入一个'B'或'b'字符，就会执行B操作



```
#include <stdio.h>

void action1(int x, int y)
{
    printf("x+y=%d\n", x+y);
}

void action2(int x, int y)
{
    printf("x*y=%d\n", x*y);
}

int main()
{
    void action1(int, int), action2(int, int);
    char ch;
    int a = 15, b = 23;
    ch = getchar();
    switch(ch)
    {
        case 'a':
        case 'A': action1(a, b); break;
        case 'b':
        case 'B': action2(a, b); break;

        default: putchar('\a');
    }

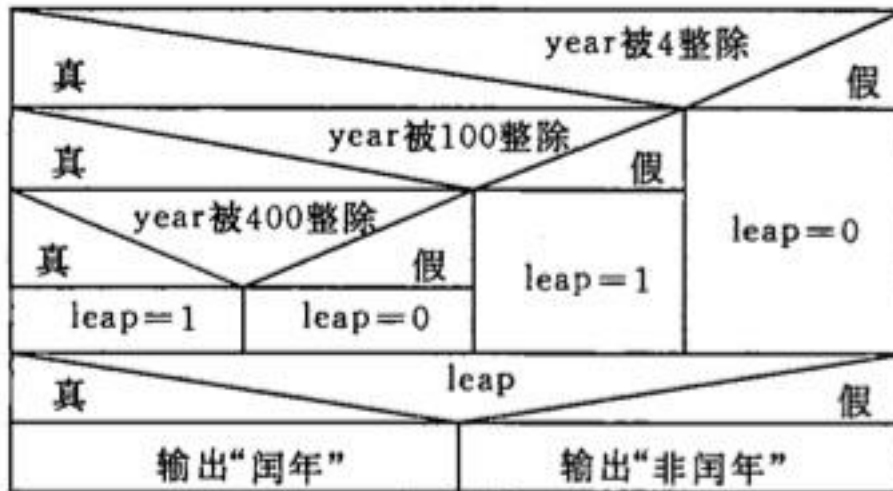
    return 0;
}
```

选择结构程序

...

选择结构程序

- 写一个程序，判断某一年是否为闰年



选择结构程序

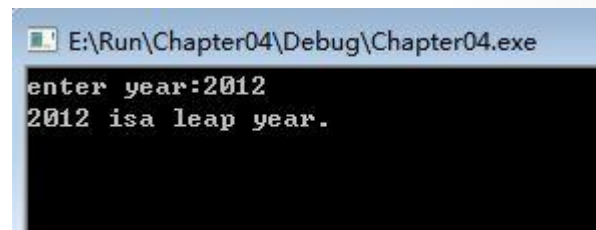
- 编写程序

```
#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;

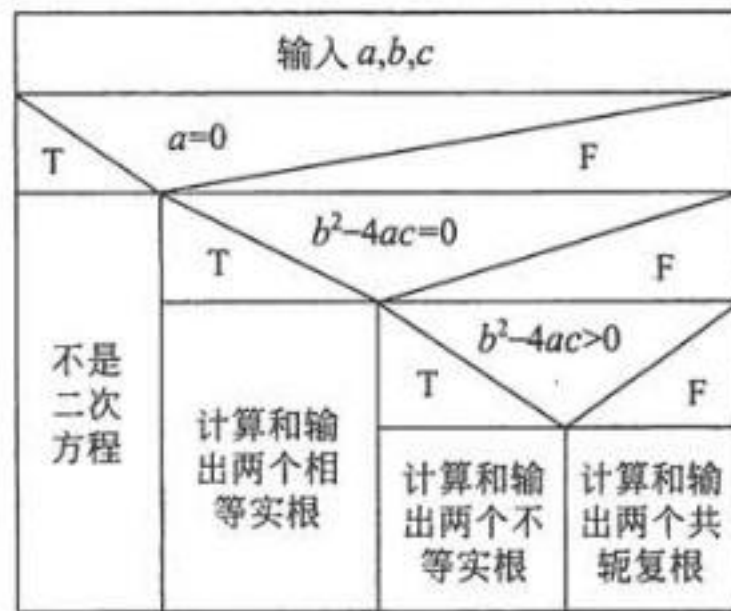
    if (leap)
        printf("%d is", year);
    else
        printf("%d is not", year);

    printf("a leap year.\n");
    return 0;
}
```



选择结构程序

- 求 $ax^2+bx+c=0$
 - $a=0$ ，不是二次方程
 - $b^2-4ac=0$ ，有两个相等实根
 - $b^2-4ac>0$ ，有两个不等实根
 - $b^2-4ac<0$ ，有两个共轭复根



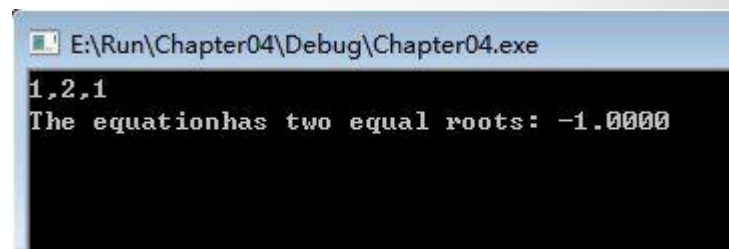
选择结构程序

- 编写程序

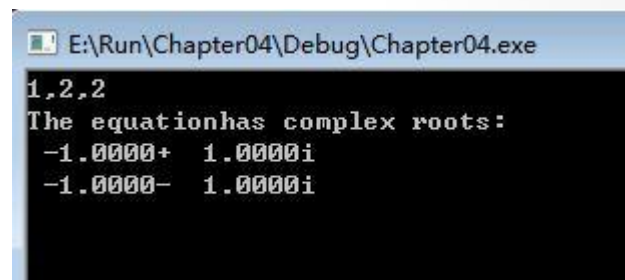
```
#include <stdio.h>
#include <math.h>

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)
        printf("is not a quadratic\n");
    else
    {
        disc = b*b - 4*a*c;
        if (fabs(disc) <= 1e-6)
            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);
            }
    }

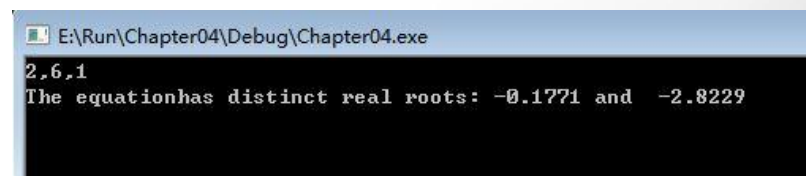
    return 0;
}
```



```
E:\Run\Chapter04\Debug\Chapter04.exe
1,2,1
The equation has two equal roots: -1.0000
```



```
E:\Run\Chapter04\Debug\Chapter04.exe
1,2,2
The equation has complex roots:
-1.0000+ 1.0000i
-1.0000- 1.0000i
```



```
E:\Run\Chapter04\Debug\Chapter04.exe
2,6,1
The equation has distinct real roots: -0.1771 and -2.8229
```

选择结构程序

- 运输公司对用户计算运输费用。路程(skm)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

选择结构程序

- 编写程序

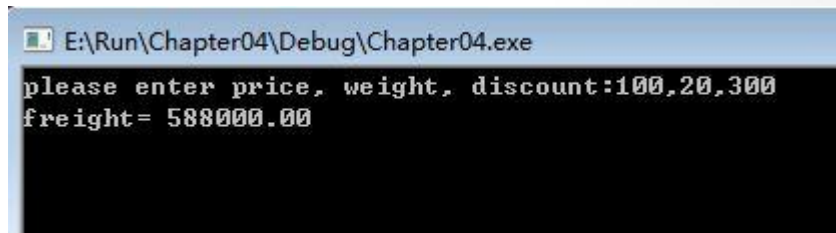
```
#include <stdio.h>

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;
    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;
}
```



```
E:\Run\Chapter04\Debug\Chapter04.exe
please enter price, weight, discount:100,20,300
freight= 588000.00
```

