

# 第5章 分支结构





## § 1 关系运算

eg1: 有一批非0的输入整数流，编程打印其中能被3整除的数。

条件: 数 $x$ 能被3整除吗? —  $\begin{cases} \text{是} \rightarrow \text{显示}x \\ \text{否} \rightarrow \text{过滤, 读取下一个}x \end{cases}$

eg2: 编程求方程 $ax^2+bx+c=0$ 的解。

条件:  $b^2-4ac \geq 0$ ? —  $\begin{cases} \text{是} \rightarrow \text{求二个实根} \\ \text{否} \rightarrow \text{求二个共轭的复根} \end{cases}$

eg3: 编程求 $f(x)$ 之值。

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

条件:  $x$ 等于几?



# § 1 关系运算

## 1. 关系运算符

## 2. 关系表达式: $\langle \text{表达式} \rangle [\langle \text{关系符} \rangle \langle \text{表达式} \rangle] \dots$

— 关系表达式的值是逻辑值, 关系成立为true, 不成立为false

— C中无逻辑型数据, 以**1**代表结果是**逻辑真**, 以**0**代表**逻辑假**

C算符	数学含义
<	< (小于)
<=	≤ (小于或等于)
>	> (大于)
>=	≥ (大于或等于)
==	≡ (恒等于)
!=	≠ (不等于)

eg1: 'a' < 'b' → 真 :1

eg2: (a=3) >= (b=5) → 假 :0

eg3: (2+3 > 5) != (0xff <= 0xfe) :0

eg4: (a > b) != (c <= d)



# § 1 关系运算

## 1. 关系运算符

2. 关系表达式: **<表达式>[<关系符><表达式>]...**

— 关系表达式的值是逻辑值, 关系成立为true, 不成立为false

— C中无逻辑型数据, 以**1**代表结果是**逻辑真**, 以**0**代表**逻辑假**

C算符	数学含义
<	< (小于)
<=	≤ (小于或等于)
>	> (大于)
>=	≥ (大于或等于)
==	≡ (恒等于)
!=	≠ (不等于)

eg5: 设a=3, b=2, c=1

a>b==c  $\longrightarrow$  1

d=a<=b  $\longrightarrow$  d=0

f=a>b>c  $\longrightarrow$  f=0



## § 2 逻辑运算

### 一

#### 逻辑运算符及逻辑表达式

C算符	数学含义
!	NOT —逻辑非
&&	AND—逻辑与
	OR—逻辑或

<表达式>[<逻辑符><表达式>]...

eg: !0      !!3

eg: 3&&2.0    'c'&&'d'    'c'&&'\0'

eg: -9||-1    0||'\0'

eg: 5>3&&2&&8<4-!0



## § 2 逻辑运算

### 二

### 真值表

条件a	条件b	!a	a&&b	a  b
F (0)	F (0)	T (1)	F (0)	F (0)
F (0)	T (非0)	T (1)	F (0)	T (1)
T (非0)	F (0)	F (0)	F (0)	T (1)
T (非0)	T (非0)	F (0)	T (1)	T (1)

**真假值判断：** 以**非0**值判断一个量为**真**，以**0**值判断为**假**  
**真假值的给定：** 给定**1**表示结果为**真**，给定**0**表示结果为**假**

eg: !0 → 1    !!3 → 1

eg: 3&&2.0    'c'&&'d'    'c'&&'\0'

eg: -9||-1    0||'\0'

eg: 5>3&&2&&8<4-!0





## § 2 逻辑运算

### 二

### 真值表

条件a	条件b	!a	a&&b	a  b
F (0)	F (0)	T (1)	F (0)	F (0)
F (0)	T (非0)	T (1)	F (0)	T (1)
T (非0)	F (0)	F (0)	F (0)	T (1)
T (非0)	T (非0)	F (0)	T (1)	T (1)

题1：逻辑运算符两侧运算对象的数据类型\_\_\_\_\_。

A. 只能是0或1

B. 只能是0或非0正数

C. 只能是整型或字符型数据

D. 可以是任意类型



## § 2 逻辑运算

### 二

### 真值表

条件a	条件b	!a	a&&b	a  b
F (0)	F (0)	T (1)	F (0)	F (0)
F (0)	T (非0)	T (1)	F (0)	T (1)
T (非0)	F (0)	F (0)	F (0)	T (1)
T (非0)	T (非0)	F (0)	T (1)	T (1)

题2：设有“int a;”，以下不能正确表达数学关系 $10 < a < 15$ 的表达式是\_\_\_\_\_。

A.  $10 < a < 15$

B.  $a == 11 || a == 12 || a == 13 || a == 14$

C.  $a > 10 \&\& a < 15$

D.  $!(a \leq 10) \&\& !(a \geq 15)$





## § 2 逻辑运算

### 二

### 真值表

条件a	条件b	!a	a&&b	a  b
F (0)	F (0)	T (1)	F (0)	F (0)
F (0)	T (非0)	T (1)	F (0)	T (1)
T (非0)	F (0)	F (0)	F (0)	T (1)
T (非0)	T (非0)	F (0)	T (1)	T (1)

**优化原则：**求解逻辑表达式时，并非所有逻辑算符均被执行，只有在必须执行下一逻辑算符才能求出解时，才执行该算符

题3：以下程序段的输出结果是\_\_\_\_\_。

```
a=2,b=1,c=7,d=9,m=1,n=1;
```

```
(m=a<=b)&&(n=c>d);    printf("m=%d,n=%d\n",m,n);
```



## § 2 逻辑运算

### 二

### 真值表

题4：有int x=1,y=1,z=1,c;执行c=--x&&--y||--z;后，x,y,z的值是\_\_\_\_\_。

题5：有int x=1,y=1,z=1,c;执行c=x--&&--y||--z;后，x,y,z,c的值是\_\_\_\_\_。

题6：有int x=1,y=1,z=1,c;执行c=x--&&y--||--z;后，x,y,z,c的值是\_\_\_\_\_。



## § 2 逻辑运算

### 二

### 真值表

题7：以下程序段的输出结果是\_\_\_\_\_。

```
int a=0,b=2,c=3;  
++a||b++&& c++;  
printf(“%d,%d,%d”, a,b,c);
```

题8：以下程序段的输出结果是\_\_\_\_\_。

```
int a=0,b=2,c=3;  
a++||b++&& c++;  
printf(“%d,%d,%d”, a,b,c);
```

题9：以下程序段的输出结果是\_\_\_\_\_。

```
int a=0,b=0,c=3;  
a||b++&& c++;  
printf(“%d,%d,%d”, a,b,c);
```

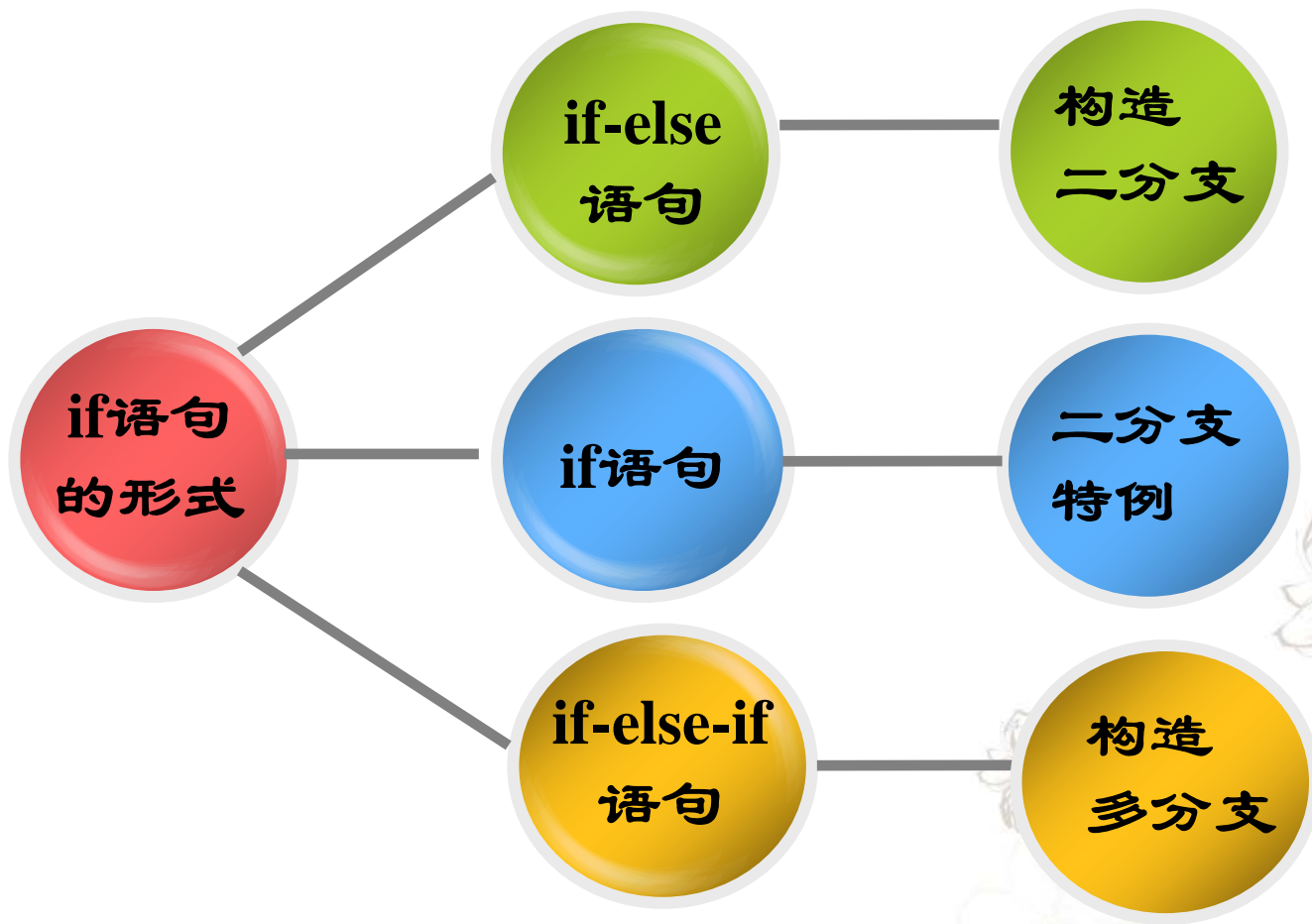


## § 2 逻辑运算

eg1: 有一批非0的输入整数流, 编程打印其中能被3整除的数

```
#include <stdio.h>
main( )
{ int x;
  r: scanf("%d",&x);
    if(x==0) goto end;
    if(x%3==0) printf("x=%d\n",x);
    goto r;
  end: ;
}
```

## § 3 if语句



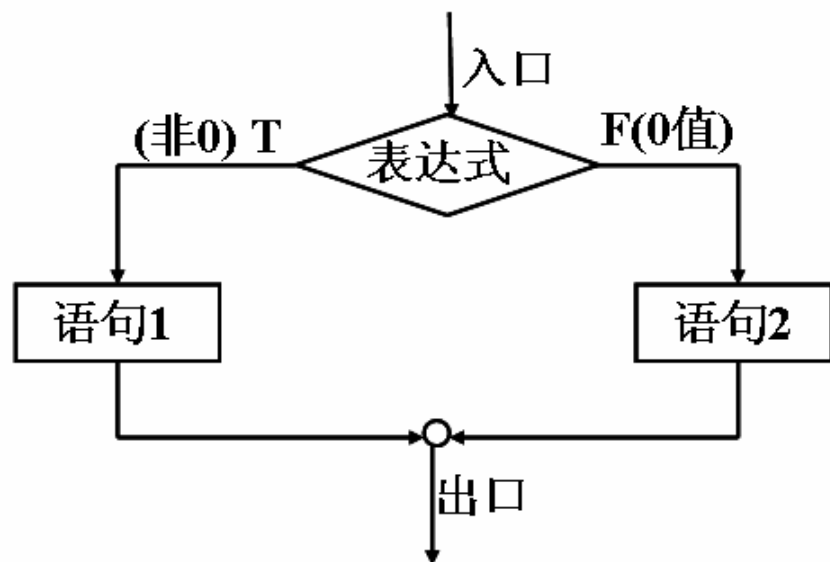
## § 3 if语句

### 1.if-else语句

**if**(表达式) 语句1 **else** 语句2

### 2.if语句

**if**(表达式) 语句



eg1:下列程序的运行结果是 **F**。

```
main( )  
{ if(2*2==5<2*2==4)  
    printf("T");  
  else  
    printf("F"); }
```



## § 3 if语句

### 1.if-else语句

**if(表达式) 语句1 else 语句2**

### 2.if语句

**if(表达式) 语句**

### 3.if-else-if 语句

eg1:下列程序的运行结果是 **F**。

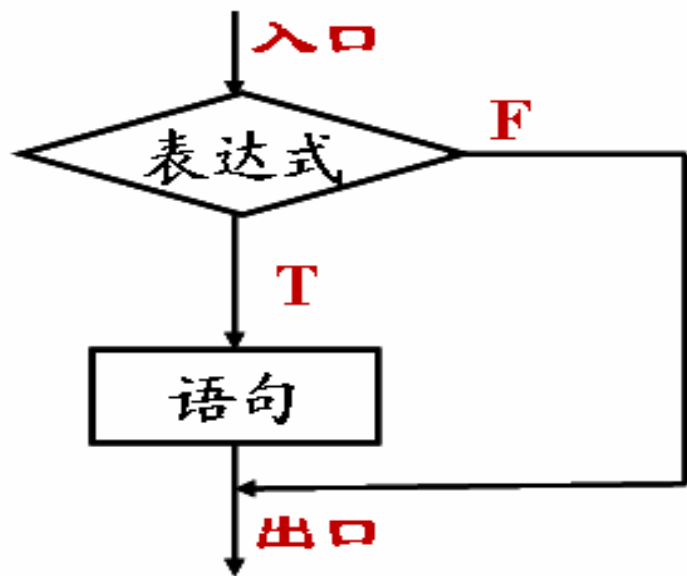
```
main( )  
{ if(2*2==5<2*2==4)  
    printf("T");  
  else  
    printf("F"); }
```

eg2: `if(3) printf("%c",'3');`

eg3: `if('a') printf("%d",'a');`

eg4: `a=1;`

`if(a=3) printf("ok");/*warning*/`





## § 3 if语句

### 1.if-else语句

**if(表达式) 语句1 else 语**

### 2.if语句

**if(表达式) 语句**

### 3.if-else-if 语句

**if(表达式1) 语句1**

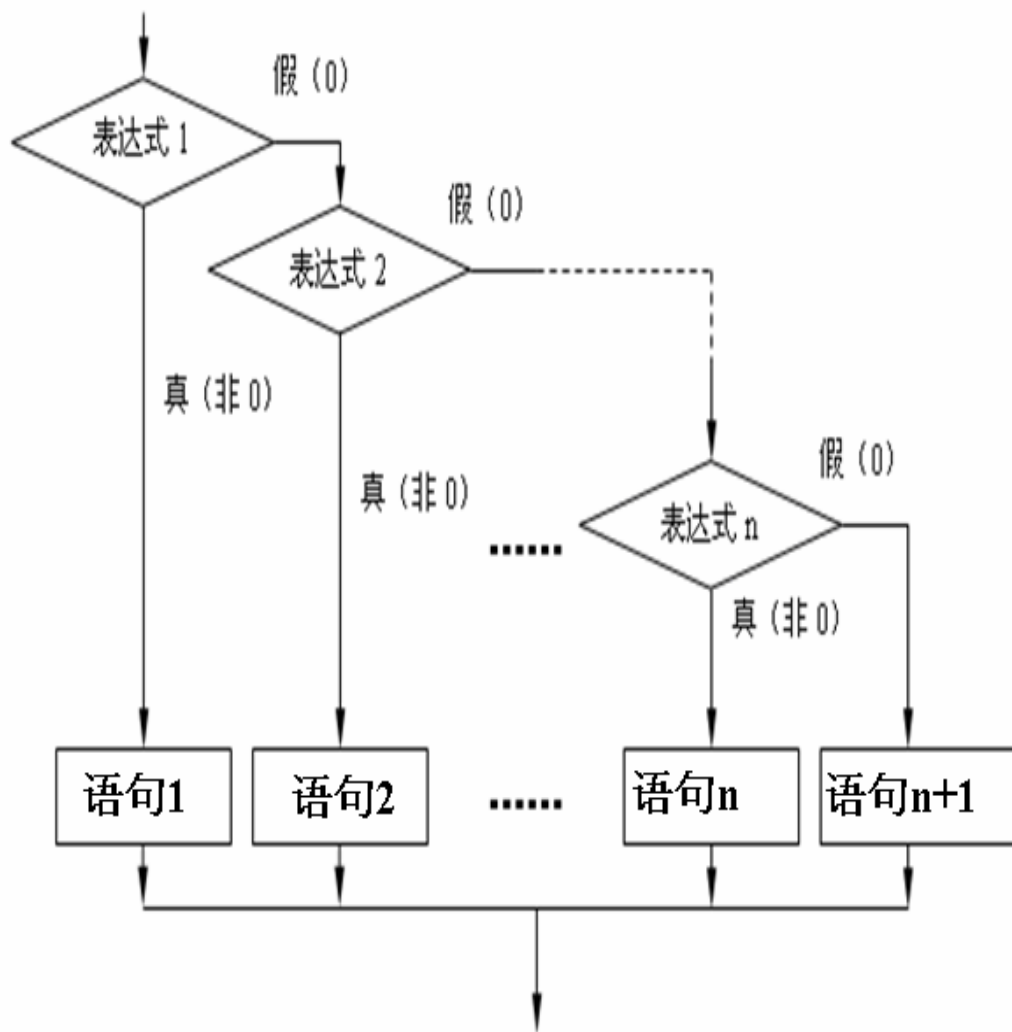
**else if(表达式2) 语句2**

**else if(表达式3) 语句3**

.....

**else if(表达式n) 语句n**

**[else 语句n+1]**







## § 3 if语句

### 1.if-else语句

**if**(表达式) 语句1 **else** 语句2

### 2.if语句

**if**(表达式) 语句

### 3.if-else-if 语句

**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]

### 4.说明

- 条件中,  $!=0$ 可以缺省
- if的各个分支只能是1个语句, 而不能是一组语句
- 各分支语句i可以是if语句, 从而形成if 嵌套
- else总是与其上最近的if匹配, 可用{ }改变这种匹配性

eg5:**if**(a\*b**!=0**) printf(“a×b≠0”);

等价

eg5:**if**(a\*b) printf(“a×b≠0”);



## § 3 if语句

eg6: 编程求 $\triangle abc$ 之面积

```
#include <stdio.h>
#include <math.h>
main( )
{ float a,b,c,s,area;
  scanf("%f,%f,%f",&a,&b,&c);
  if(a+b>c&&a+c>b&&b+c>a)
  { s=(a+b+c)/2.0;
    area=sqrt(s*(s-a)*(s-b)*(s-c));
    printf("\na=%f,b=%f,c=%f,area=%f\n",a,b,c,area);
  }
  else printf("\n Error!Input a,b,c again,Please!");
}
```



## § 3 if语句

例1: `if(exp) a++;`中与`exp`完全等价的表达式是 **D**。

- A. `exp==1`
- B. `exp!=1`
- C. `!exp`
- D. `exp!=0`

例2: 以下关于if语句的错误描述是 **B**。

- A. 条件表达式可以是任意表达式
- B. 条件表达式只能是关系表达式或逻辑表达式
- C. 条件表达式的括号不可以省略
- D. 与else配对的if语句是其之前最近的未配对的if语句



## § 3 if语句

例3：有函数

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

判断下列解法的正确性。

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);
```

```
}
```

正确

main( )

```
{ int x,y;
```

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

```
y=-1;
```

```
if(x!=0)
```

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

```
else y=0;
```

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

```
}
```

错误



## § 3 if语句

**例4：**对成绩g分档显示：90~100为A等，80~89为B等，70~79为C等，60~69为D等，若小于60为F等并给出提示信息。

```
main( )
{ int g,need;
  scanf("%d",&g);
  printf("\n The student\'s grade is:");
  if(g>=90)      printf("A\n");
  else if(g>=80)  printf("B\n");
  else if(g>=70)  printf("C\n");
  else if(g>=60)  printf("D\n");
  else { printf("F\n"); need=60-g;
        printf("The student requires %d points to pass!",need);
      }
}
```



## § 3 if语句

例5：输入三个数，按由小到大的顺序将其排序。

```
main( )  
{ float a,b,c,temp;  
  scanf("%f,%f,%f",&a,&b,&c);  
  printf("\n Before sorted,a=%.2f,b=%.2f,c=%.2f\n",a,b,c);  
  if(a>b) {temp=a; a=b; b=temp;} /*a为a,b中小者*/  
  if(a>c) {temp=a; a=c; c=temp;} /*a为a,b,c中小者*/  
  if(b>c) {temp=b; b=c; c=temp;} /*b为b,c中小者*/  
  printf("\n After sorted,a=%.2f,b=%.2f,c=%.2f\n",a,b,c);  
}
```



## § 3 if语句

**例6：编程判某年year是否是闰年**





## § 3 if语句

**判闰年条件：** ①year能被4整除的同时不能被100整除；  
② year能被400整除

```
main( )  
{ int year, leap;  
  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"); }
```

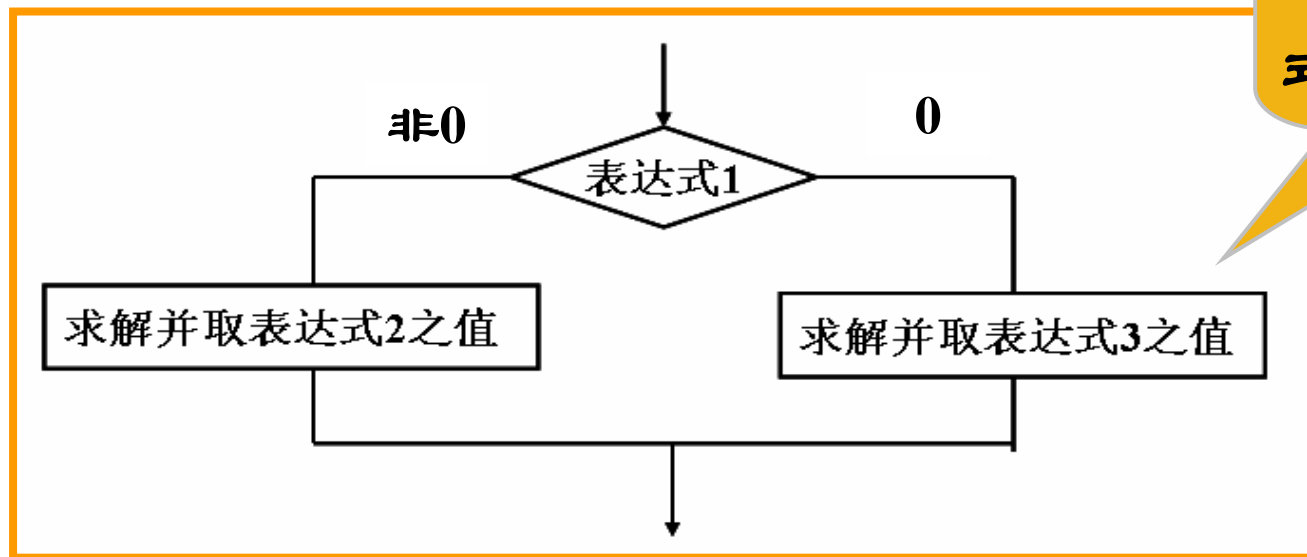
if(year%4!=0) leap=0;  
else if(year%100!=0) leap=1;  
else if(year%400!=0) leap=0;  
else leap=1;





## § 4 条件运算符?:

表达式1? 表达式2:表达式3



结果的类型为  
式2、式3高者

eg1:  $x?(y?1:0):0$

$x\&\&y$

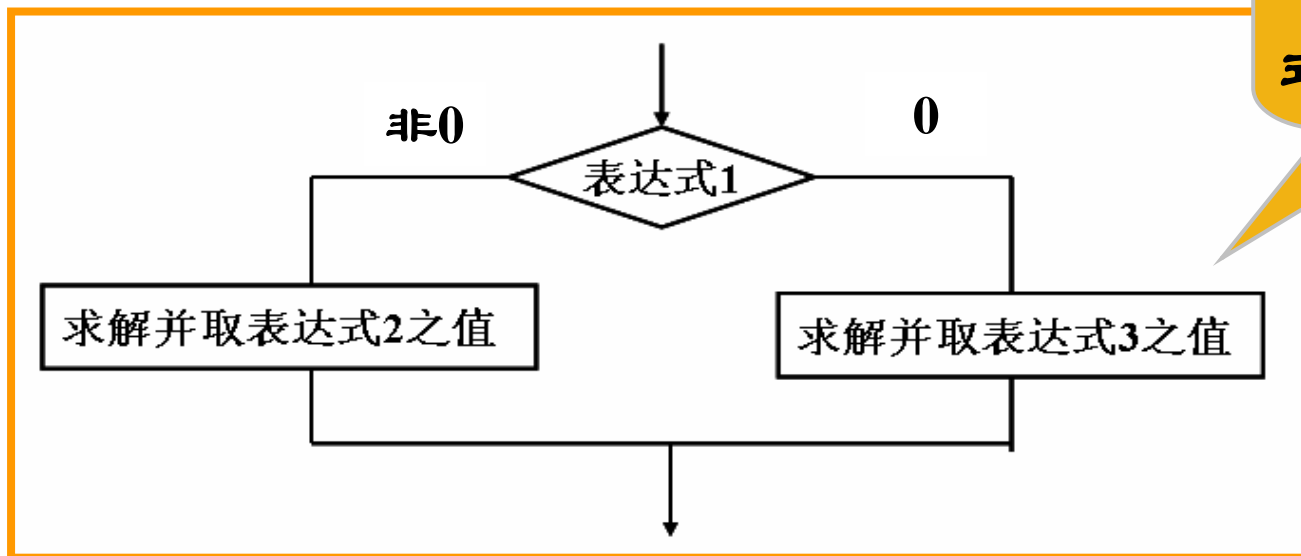
eg2:  $x?1:(y?1:0)$

$x||y$



## § 4 条件运算符?:

表达式1? 表达式2:表达式3



结果的类型为  
式2、式3高者

eg3: main( )  
{ int k=4,a=3,b=2,c=1;  
printf(“\n%d\n”,k<a ? k :(c<b ? c :a));  
}



## § 4 条件运算符?:

题1: 设有说明“`int x,y,z,m=10,n=5;`”,执行下面语句:

```
x=(--m==n++)?--m:++n; y=m++; z=n;
```

则x,y,z的值分别为\_\_\_\_\_。

A. 7,9,7      B. 5,9,7      C. 6,11,5      D. 10,11,10

题2: 设有说明“`int x=2,y=3;`”,则`++x>y--?x:y`的值为\_\_\_\_\_。

A. 1      B. 2      C. 3      D. 4

题3: 若有说明语句: `int x=1,y=0;`则表达式

`x-->(y+x)?5:25>y++?'1':'2'`的值是\_\_\_\_\_。

题4: 执行以下程序段后, 变量a,b,c的值是\_\_\_\_\_。

```
int a,b,c ,x=10,y=9;
```

```
a=(--x==y++)?--x:++y; b=x++; c=y;
```





## § 4 条件运算符?:

例：写一个函数模拟tolower,将读入的字母从大写转化为小写输出,非大写字母原样输出。

```
#include <stdio.h>

main( )
{ char ch;

  ch=getchar( );

  ch=ch>='A' && ch<='Z'? ch+'a'-'A' :ch;

  putchar(ch);

}
```



## § 5 switch语句

错误  
流程

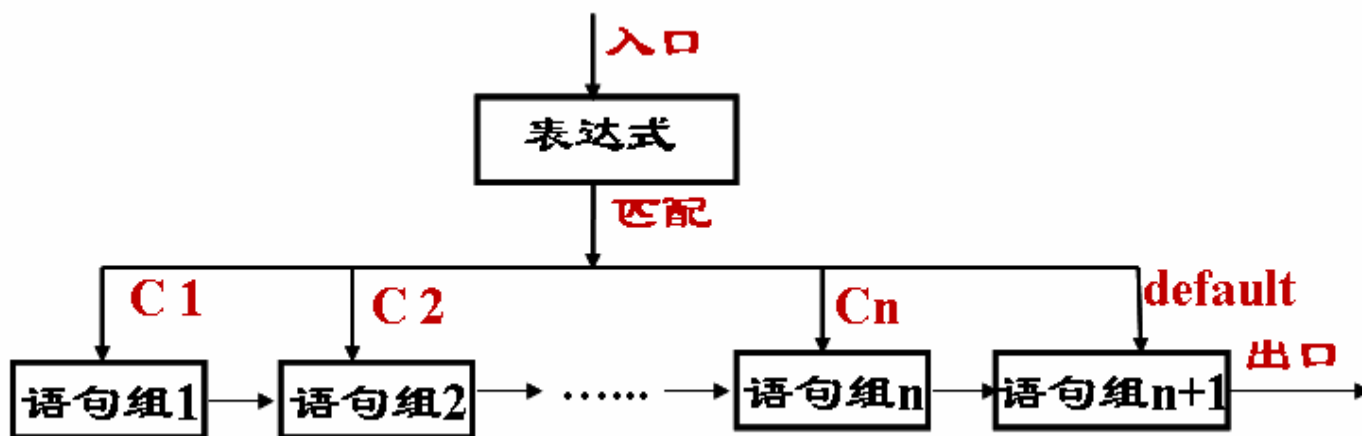
**switch**(表达式)

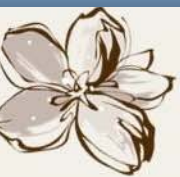
```
{ case 常量表达式1: 语句组1break;  
  case 常量表达式2: 语句组2break;  
  .....  
  case 常量表达式n: 语句组nbreak;  
  [default:      语句组n+1]  
}
```

**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");  
}
```

eg:由成绩等级grade输出分数段。  
A等: 85~100  
B等: 70~84  
C等: 60~69  
D等: <60





## § 5 switch语句

错误  
流程

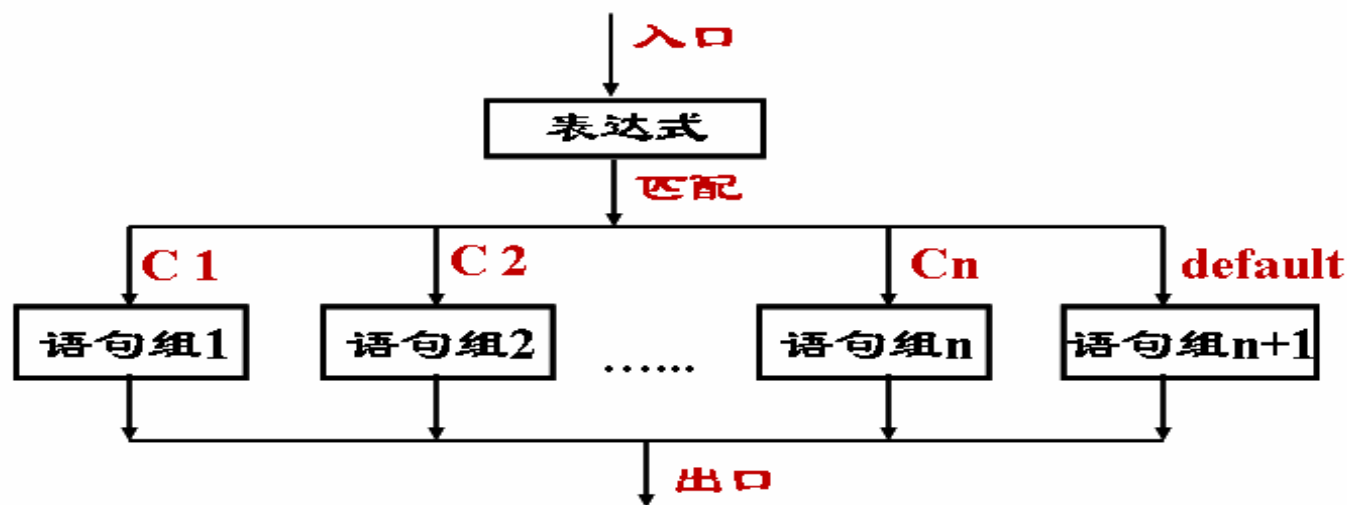
**switch**(表达式)

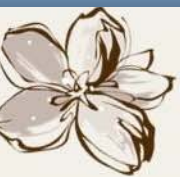
```
{ case 常量表达式1: 语句组1break;  
  case 常量表达式2: 语句组2break;  
  .....  
  case 常量表达式n: 语句组nbreak;  
  [default: 语句组n+1]  
}
```

**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");  
}
```

eg:由成绩等级grade输出分数段。  
A等: 85~100  
B等: 70~84  
C等: 60~69  
D等: <60





## § 5 switch语句

错误  
流程

**switch**(表达式)

```
{ case 常量表达式1: 语句组1break;  
  case 常量表达式2: 语句组2break;  
  .....  
  case 常量表达式n: 语句组nbreak;  
  [default: 语句组n+1]  
}
```

**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");  
}
```

**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");  
}
```

eg:由成绩等级grade输出分数段。

A等: 85~100

B等: 70~84

C等: 60~69

D等: <60





## § 5 switch语句

**switch**(表达式)

```
{ case 常量表达式1: 语句组1 break;  
  case 常量表达式2: 语句组2 break;  
  .....  
  case 常量表达式n: 语句组n break;  
  [default: 语句组n+1]  
}
```

- 对于带有break的开关语句，各个case与default可以任意次序出现，不影响执行结果。
- 各case常量必须互不相等，而多个case可以共用同一组语句。

**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");  
}
```

**switch**(grade)

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





## § 5 switch语句

**switch**(表达式)

```
{ case 常量表达式1: 语句组1break;  
  case 常量表达式2: 语句组2break;  
  .....  
  case 常量表达式n: 语句组nbreak;  
  [default:      语句组n+1]  
}
```

eg:由成绩等级grade输出  
考试结果:  
A等、B等、 C等:**PASS!**  
D等:**Fail!**

- 对于带有break的开关语句，各个case与default可以任意次序出现，不影响执行结果。
- 各case常量必须互不相等，而多个case可以共用同一组语句。

switch(grade)

```
{ case 'A':  
    case 'B':  
    case 'C': printf("PASS!\n");break;  
    case 'D': printf("Fail! \n");  
}
```



## § 5 switch语句

**例1：**当运行时输入字母A时，程序的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
main( )
{ char ch;
  ch=getchar( );
  switch(ch)
  { case 65: printf("%c",'A');
    case 66: printf("%c",'B');
    default: printf("%s\n","other");
  }
}
```

**ABother**



## § 5 switch语句

例2：以下程序的输出结果是\_\_\_\_\_。

```
main( )  
{ int a=1,b=0;  
  switch(a)  
  { case 1: switch(b)  
          { case 0: printf("**0**"); break;  
            case 1: printf("**1**"); break;  
          }  
    case 2: printf("**2**"); break;  
  }  
}
```

**\*\*0\*\*\*\*2\*\***



## § 5 switch语句

例3：编程计算货物运费f:

$$f = p \times w \times s \times (1 - d\%)$$

其中p为基本运费，w为货重，s为运货距离，d为折扣。

$s < 250\text{km}$	$c = 0$	无折扣
$250 \leq s < 500$	$c = 1$	2%折扣
$500 \leq s < 1000$	$c = 2, 3$	5%折扣
$1000 \leq s < 2000$	$c = 4, 5, 6, 7$	8%折扣
$2000 \leq s < 3000$	$c = 8, 9, 10, 11$	10%折扣
$s \geq 3000$	$c \geq 12$	15%折扣

输入p,w,s					
T			s≥3000		
F					
c=12			c=s/250		
c之值					
0	1	2,3	4,5,6,7	8,9,10,11	12
d=0	d=2	d=5	d=8	d=10	d=15
f=p*w*s*(1-d/100)					
输出f					



## § 5 switch语句

```
#include <stdio.h>
main( )
{ int c,s,d; float p,w,f;
  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; }
  f=p*w*s*(1-d/100.);
  printf("freight=%.2f\n",f); }
```

输入p,w,s					
T			s≥3000		
F					
c=12			c=s/250		
c之值					
0	1	2,3	4,5,6,7	8,9,10,11	12
d=0	d=2	d=5	d=8	d=10	d=15
f=p*w*s*(1-d/100)					
输出f					

