



第4章 逻辑运算和选择结构



第一节 关系运算符和关系表达式

关系运算实际上就是一种“比较运算”，
即

用关系运算符对两个运算量进行比较，比较他们之间的“大小关系”。

例： 关系表达式：

$y > 3$

用于判断 y 是否大于3



1. 关系运算符及其优先次序

C语言中的关系运算符

运算符	名称	示例	示例说明	优先次序
<	小于	$a < b$	a小于b	优先级相同 (高)
<=	小于或等于	$a \leq b$	a小于或等于b	
>	大于	$a > b$	a大于b	
>=	大于或等于	$a \geq b$	a大于或等于b	
==	等于	$a == b$	a等于b	优先级相同 (低)
!=	不等于	$a != b$	a不等于b	



关于优先次序的说明如下：

(1) 关系、算术、赋值运算符的优先级关系如下：

算术运算符（高） \rightarrow 关系运算符（中） \rightarrow 赋值运算符（低）

(2) 关系运算符的结合方向是“自左向右”

例：

$c+d > a+b \Rightarrow (c+d) > (a+b)$

$a > b != c \Rightarrow (a > b) != c$

$a == b < x \Rightarrow a == (b < x)$

$a = b >= c \Rightarrow a = (b >= c)$

$x != a * b < c \Rightarrow x != ((a * b) < c)$



2. 关系表达式

例：

$x < y$, $2 * a + b < c - d$, $(x = 2) \leq (y = 4)$, $'a' < 'b'$

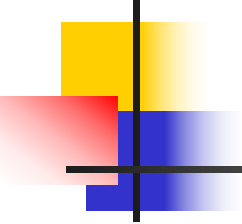
在C语言中，关系表达式的结果为1或0。

当关系表达式成立时，其结果为1；

否则值为0。

例： $3 < 2$ 的值为 0

$5! = 81$ 的值为 1



例：试求 $x=5>3>1$ 的值

分析： $x=5>3>1 \Rightarrow x=(5>3)>1$
 $=1>1$
 $=0$



第二节 逻辑运算符和逻辑表达式

- 关系表达式只适用于描述单一的条件，
如： $x > 0$
- 对于较复杂的复合条件，
如：“ x 大于0且不等于2”，就需要逻辑运算符才能描述，即 $x > 0 \ \&\& \ x \neq 2$ ，其中“ $\&\&$ ”是C语言中的逻辑运算符。

C语言中的逻辑运算符

运算符	名称	示例	示例说明	优先次序
!	逻辑非	!a	a非	<div> <div></div> <div>↑</div> <div>高</div> <div>低</div> </div>
&&	逻辑与	a&&b	a与b	
	逻辑或	a b	a或b	



关于优先次序的说明如下：

(1) 逻辑运算符的结合方向是“自左向右”。

(2) 运算符的优先级

运算符	优先次序
!(非)	 高 低
算术运算符	
关系运算符	
&&和	
赋值运算符	



2. 逻辑表达式

一个逻辑表达式中可以包含多个逻辑运算符。

如： `x > y && a <= b - !c`

在C语言中, 当参与逻辑运算量为**非零值**表示“真”,
整数零表示“假”。

C语言编译系统在给出运算结果时, 以数值**1**代表“真”,
以数值**0**代表“假”。



运算规则

- 只有a, b同为真, $a \& \& b$ 才为真, 否则为假
- 只要a, b中任一为真, $a || b$ 就为真
- 若a为真, 则!a为假; a为假, 则!a为真



逻辑运算规则表

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

例： 设 $x=9$, $y=3$,

则逻辑表达式: $x>y\&\&!x$ 的值为 0



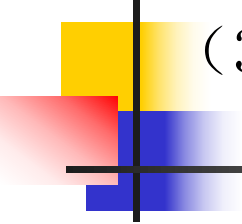
逻辑运算注意点:

(1) 参与逻辑运算的量不但可以是0和1，也可以是0和非0的任何类型的数据，如：整型、字符型、实型等。

例： ‘A’&&'B' , 值为1

(2) 在处理一个复杂表达式时，对于表达式中不同位置上出现的数值，应区分哪些是作为数值运算或关系运算，哪些作为逻辑运算的对象。

例： $x > y \&\& a \mid \mid b \leq c - !d$



(3) 在逻辑表达式求解中，并不是所有逻辑运算符都执行，有时只需执行一部分运算就可以得出结果。

例1: `int a=1, b=0, c;`
`c=a || ++b;`
`c=? b=?`

例2: `int a=1, b=0, c;`
`c=b&&a++;`
`c=? a=?`



第三节 选择结构

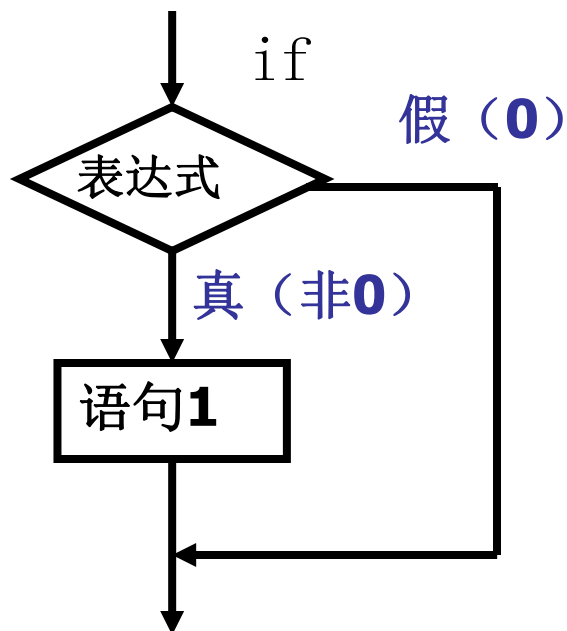
1. if语句的三种形式

- if 语句
- if-else 语句
- if-else-if 语句



if语句

格式: `if (表达式) 语句 ;`

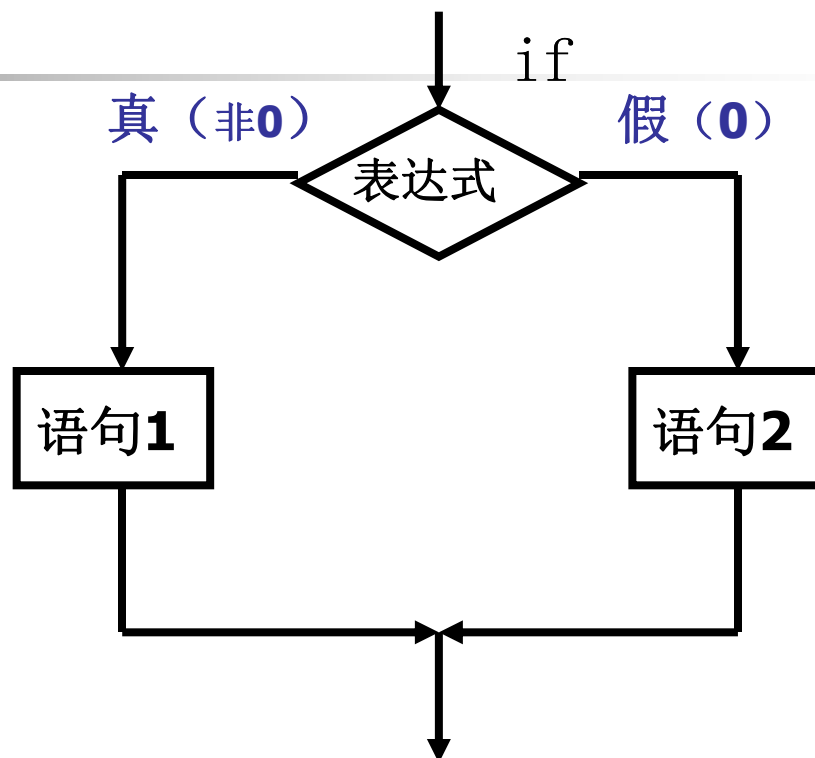


例: `if(x>y) printf("x>y");`
输入两个整数, 求最大值

[返回](#)

if-else语句

格式: `if (表达式)`
 语句1 ;
 else
 语句2 ;



例:输入两个整数, 求最大值

三个整数, 求最大值?

返回

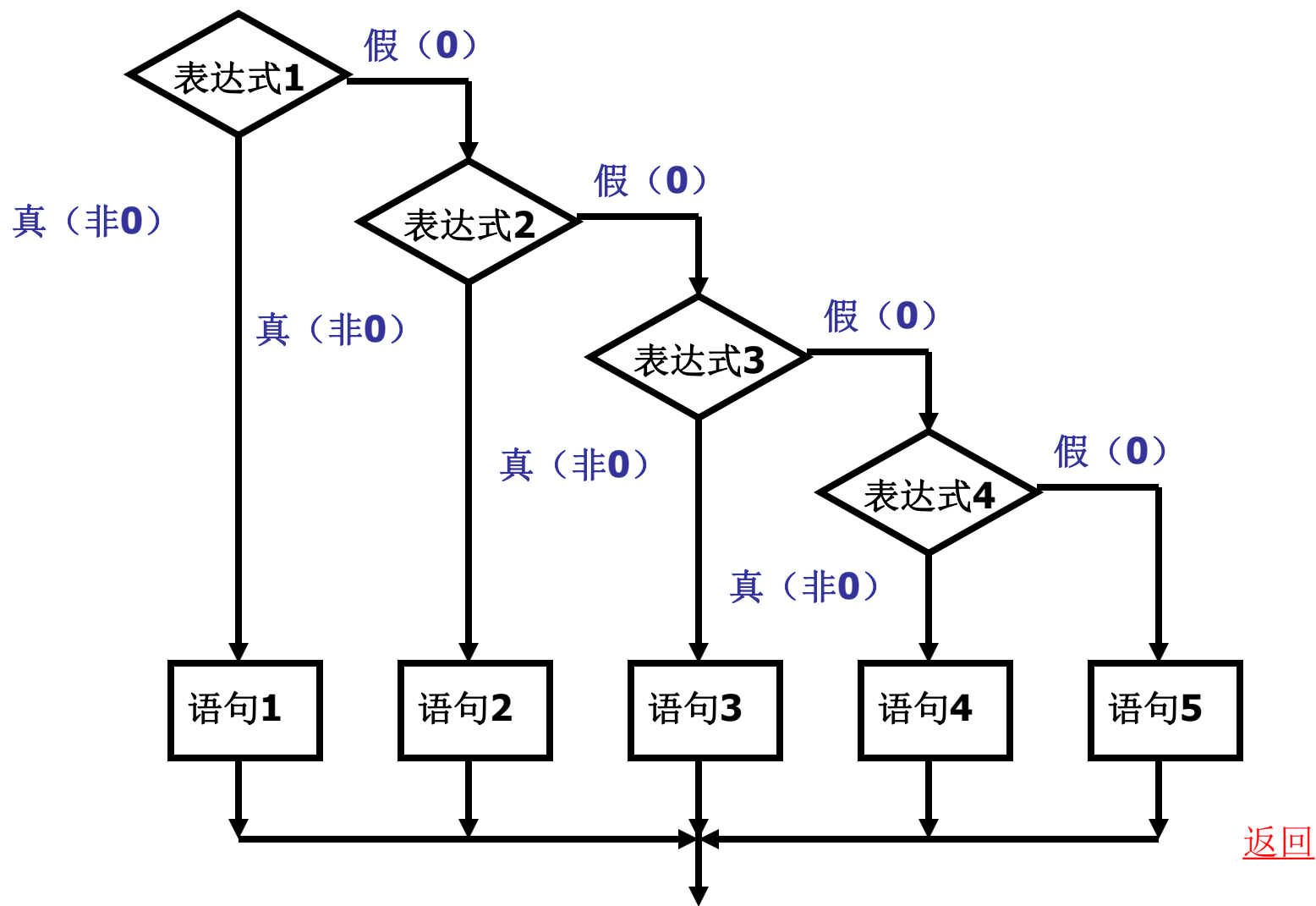


if-else-if语句

格式: if (表达式1)
 语句1 ;
 else if (表达式2)
 语句2 ;

 else if (表达式n-1)
 语句n-1 ;
 else
 语句n ;

if-else-if语句





程序举例

编写一个程序：要求输入一个学生的考试成绩，输出其分数和对应的等级。

共分5个等级：

小于60分为 “E”；

60~69分为 “D”；

70~79分为 “C”；

80~89分为 “B”；

90分以上的为 “A”。

```
if(score<60) grade="E";  
else if(score<70) grade="D";  
else if(score<80) grade="C";  
else if(score<90) grade="B";  
else grade="A"
```



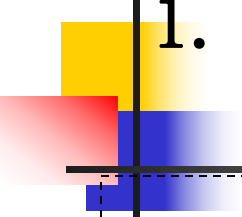
if语句注意点:

(1) if后的表达式一般为关系或逻辑表达式, 也可有其他表达式, 甚至可为变量

```
int x=1, a=0;  
if(a==0) x=3;  
if(a=0) x=4;  
if(a) x=5;  
x=?
```

(2) if后语句为单语句, 若要执行多个语句则必须组合形成复合语句。

```
int x=3;  
if(x>4) ;  
x=2;  
x++;  
x=?
```



1. if语句的嵌套

```
if(表达式1)
```

```
    if(表达式2)
```

```
        语句1;
```

```
    else
```

```
        语句2;
```

```
else
```

```
    语句3;
```

```
if(表达式1)
```

```
    语句1;
```

```
else
```

```
    if(表达式2)
```

```
        语句2;
```

```
    else
```

```
        语句3;
```

```
if(表达式1)
```

```
    if(表达式1)
```

```
        语句1;
```

```
    else
```

```
        语句2;
```

```
else
```

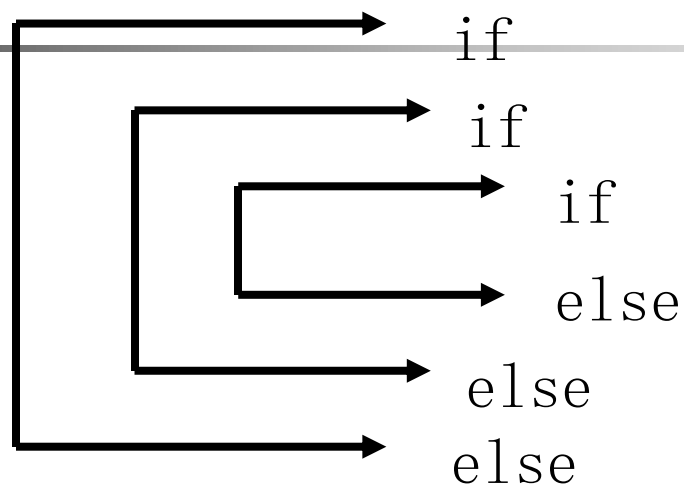
```
    if(表达式2)
```

```
        语句3;
```

```
    else
```

```
        语句4;
```

if和else配对规则：else总是与它前一个最近的if配对



若要求else与前面的if配对，则要用到复合语句。

如： if(表达式1)

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



如： 分析下面两段程序z的值

```
1.  if(n>0)                /* a=3, b=5, n=2 , z=100*/  
    if(a>b) z=a;  
    else z=b;
```

```
2.  if(n>0)  
    {if (a>b) z=a;}  
    else z=b;
```




第四节 条件运算符

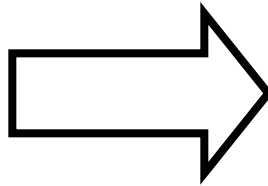
一般形式： 表达式1?表达式2:表达式3

例： if (x>y)

max=x;

else

max=y;



max=(x>y)?x:y;

条件运算符说明：

- (1) 条件运算符优先级高于赋值运算符，但低于算术运算符、关系运算符、逻辑运算符。
- (2) 条件运算符的结合方向：“自右向左”。

例： $a > b ? a : c > d ? c : d \Rightarrow a > b ? a : (c > d ? c : d)$

- (3) 表达式1、表达式2、表达式3类型可以不同，条件表达式的值取两者较高的类型。

例： $x > y ? 1 : 1.5 \quad /*x=2, y=1*/$

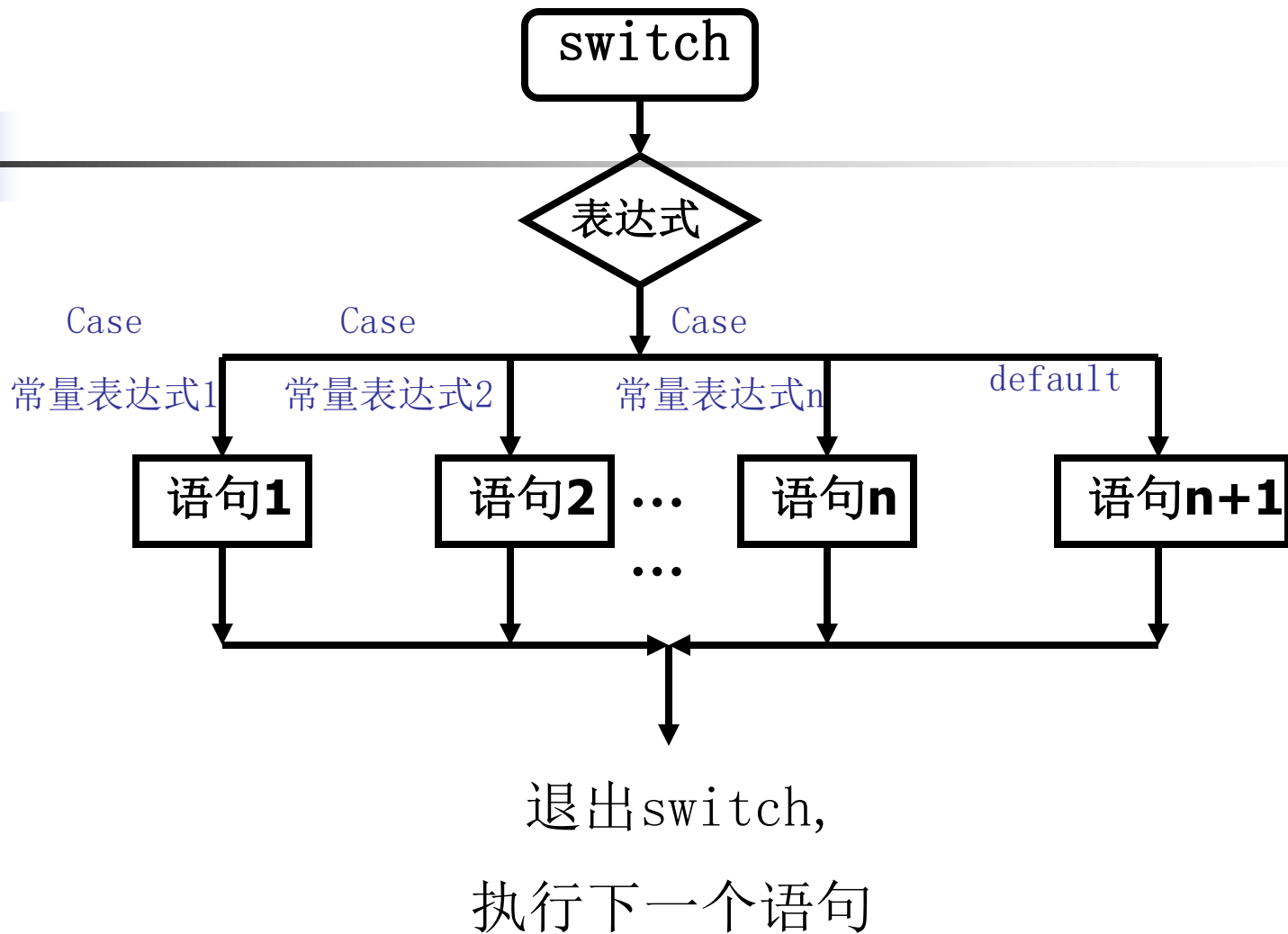
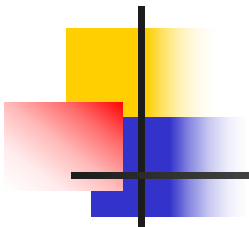


第五节 switch-case语句

Switch-case语句用于处理多分支结构的条件选择语句，又称“开关语句”。

一般形式： `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; break;  
}
```



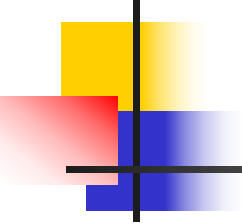


注意点:

- (1) 各个case常量表达式不一定按其值的大小顺序排列，但各个case后的常量表达式必须是不同的值。

例:

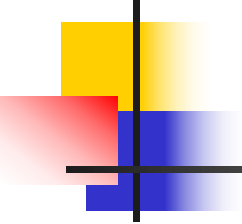
```
switch(ch)
{
    case'A':语句1;    break;
    case'C':语句2;    break;
    case'B':语句3;    break;
    case'A':语句4;    break;
    default:语句5;
}
```



(2) 在switch-case语句中，多个case可以共用一条执行语句。

例：

```
.....  
case'A':  
case'B':  
case'C': printf(">60");break;  
.....
```



(3) 若case后面的语句执行完后没有break语句，程序将顺序执行下面的case分支，直到遇到break语句，跳出switch-case语句或switch-case语句结束。

例：

.....

```
switch(a>b?1:2)  /* a=3, b=6 */
{
    case 1: a=100;
    case 2: b=a+b;
    case 3: a=b+a;
    default: b--;
}
printf("%d, &d\n", a, b);
```

结果： 12, 8

(4) switch-case语句的嵌套

```
例:  #include "stdio.h"
      main()
      {   int x=1, y=0, a=0, b=0;
          switch(x)
          {   case 1:
                  switch(y)
                  {   case 0:a++;break;
                      case 1:b++;break;
                  }   /*注意该位置没有break*/
              case 2:
                  a++;b++; break;
          }
          printf("a=%d, b=%d\n", a, b);
      }
```

结果 a=2, b=1



程序举例

1. 求一个整数的绝对值。

```
main()
{
    int a;
    scanf("%d",&a);
    if(a<0)
        a=-a;
    printf("%d",a);
}
```



程序举例

2. 输入2个实数，按代数值由小到大输出。

```
main()
{
    int a,b,t;
    scanf("%d,%d",&a,&b);
    if(a>b)
    {
        t=a;a=b;b=t;
    }
    printf("%d,%d",a,b);
}
```



程序举例

3. 计算分段函数 $y = \begin{cases} x^2 - 2 & x \geq 0 \\ x & x < 0 \end{cases}$ 的值

```
main()
{
    float x,y;
    scanf("%f",&x);
    if(x>=0)
        y=x*x-2;
    else
        y=x;
    printf("%.2f",y);
}
```



程序举例

4. 编写一个程序：要求输入一个学生的考试成绩，输出其分数和对应的等级，用switch-case实现。

共分5个等级：

小于60分为 “E”；

60~69分为 “D”；

70~79分为 “C”；

80~89分为 “B”；

90分以上的为 “A”。