

# 第四章

# 选择结构程序设计





# 实例分析

## ■ 实例

输入两个整数，输出较大数。

问题分析：

求较大数的模型：若

数据结构：三个变量

算法：(1) 输入变量

(2) 若  $a > b$  ,

否则,  $b$  [

(3) 输出变量 }

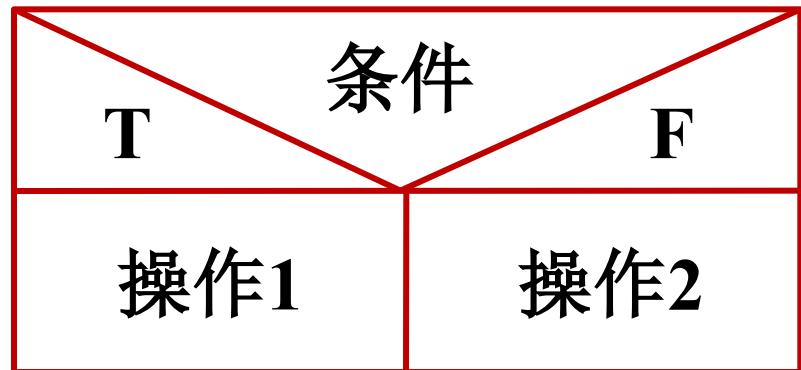
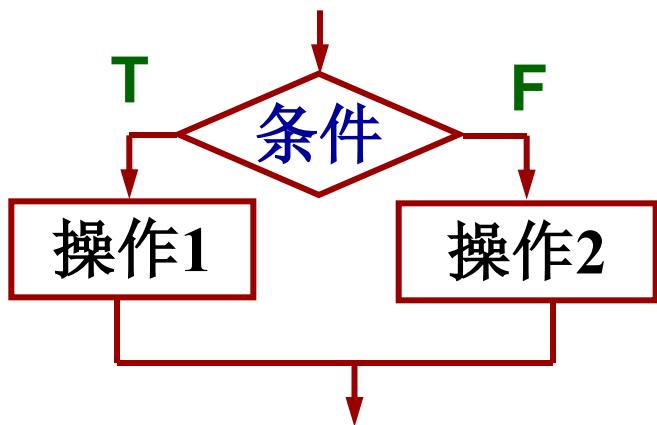
```
#include <stdio.h>
void main()
{ int a,b,m;
scanf("%d%d",&a,&b);
if(a>b)
    m=a;
else
    m=b;
printf("较大数为:%d\n",m);}
```





# 选择结构

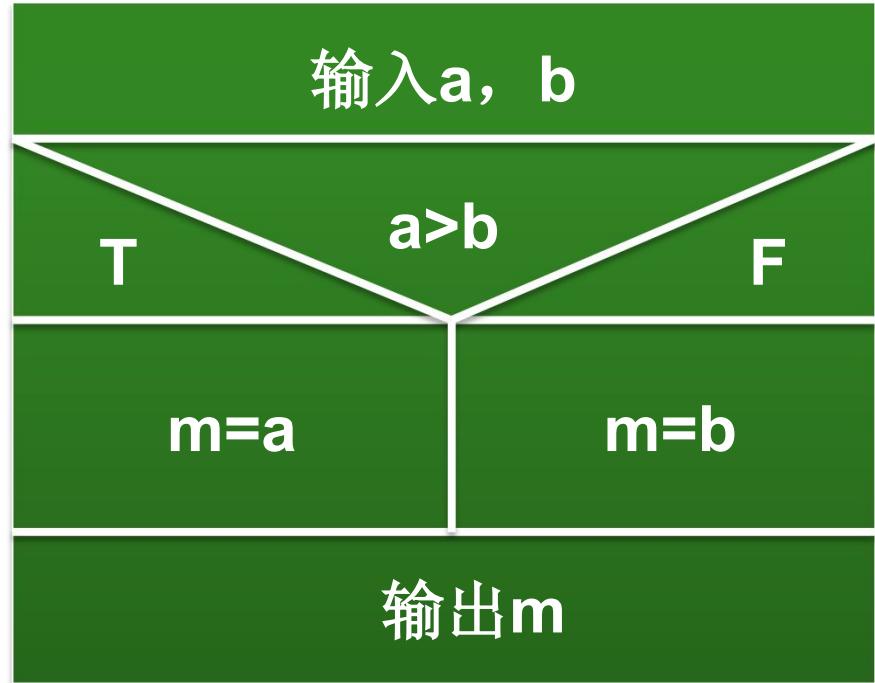
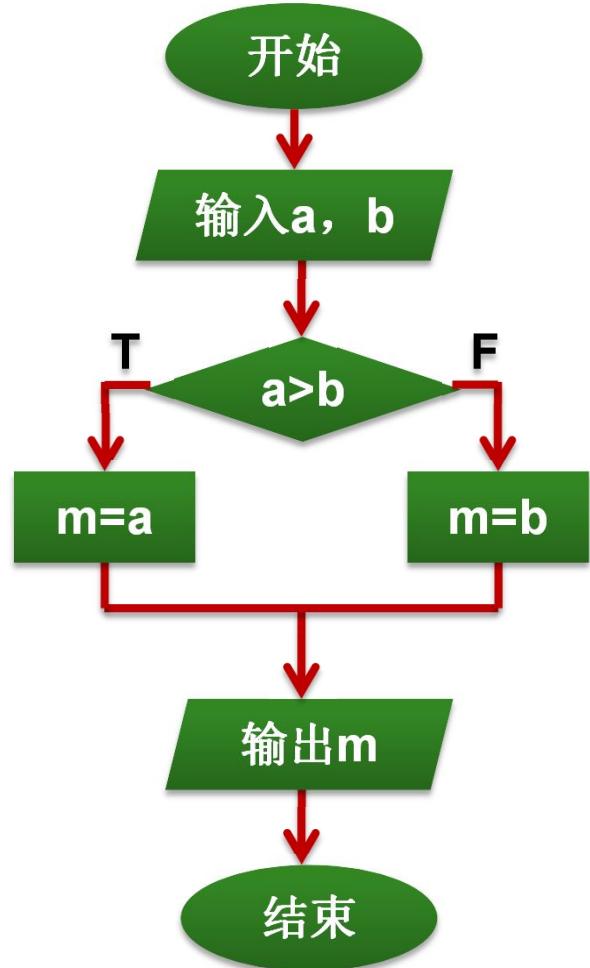
- 选择结构：  
根据一定的条件分别执行不同的操作。
- 表示方法：





# 选择结构

## 上例算法的描述





# 选择结构

## ■ C语言实现方法

**if** 语句：单分支、双分支结构

**switch**语句：多分支选择结构

条件表达式：简单

```
#include <stdio.h>
void main()
{ int a,b,m;
  scanf("%d%d",&a,&b);
  m=a>b?a:b;
  printf("较大数为:%d\n",m);
}
```





# if语句—单分支if语句

## 实例：

读入一个字符；若是小写字母，将其改为大写的，输出字母；否则原样输出该字符。

## 问题分析：

小写改大写数学模型：  $a = a - 32$

数据结构：一个变量 **a**，字符型。 单分支

过程：读入 **a**；

若 '**a**' <= **a** <= '**z**'       $a - 32 \Rightarrow a$

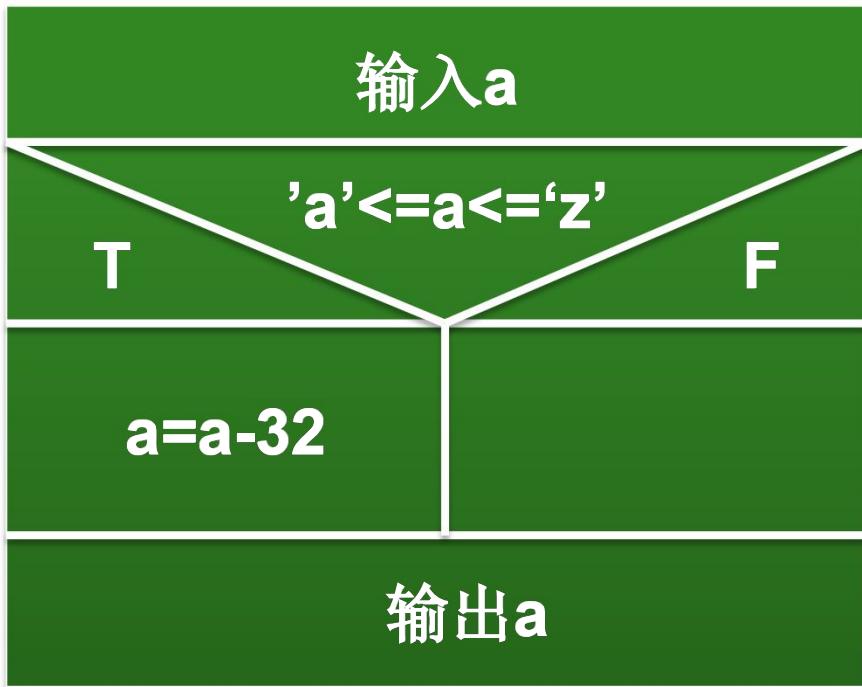
输出 **a**。





# if语句—单分支if语句

框图、程序代码



```
#include <stdio.h>
void main()
{ char a;
a=getchar();
if('a'<=a&&a<='z')
    a=a-32;
printf("%c\n",a);
}
```





# if语句—单分支if语句

## ■ 语句格式

**if(表达式)**

语句

```
if('a'<=a&&a<='z')  
    a=a-32;
```

## ■ 说明：

(1) 表达式值为非0=真， 值为0=假

(2) 语句可以是一条语句， 也可以是多条语句。



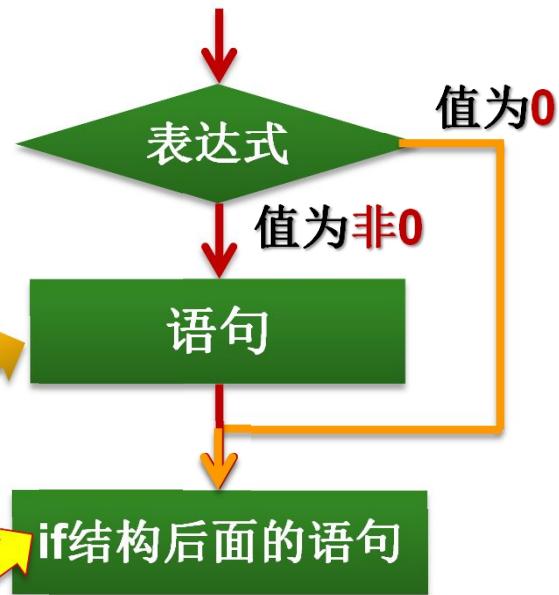


# if语句—单分支if语句

程序的功能

```
#include <stdio.h>
void main()
{int m,a,b,c,s=0;
scanf("%d",&m);
if(m>=100&&m<=999)
{a=m%10;
b=m/10%10;
c=m/100;
s=a+b+c; }
printf("s=%d\n",s);
}
```

if语句执行流程



注意

若if结构中的语句  
是一组语句时，  
用{}将其括起来。





# if语句—双分支if语句

## 实例

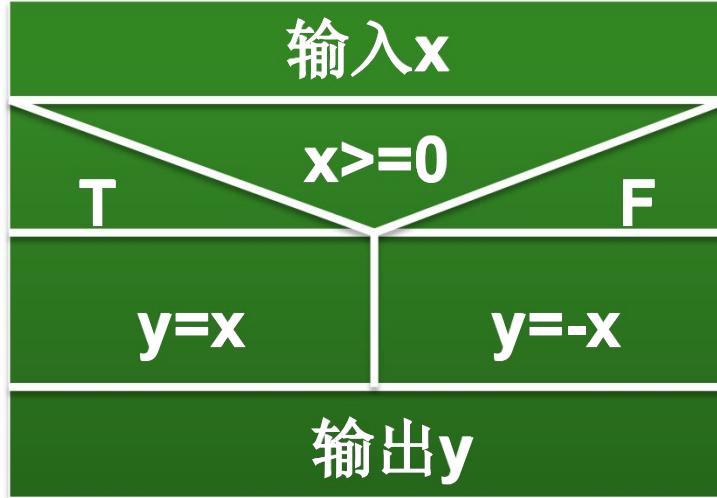
计算、输出 $|x|$ 的值。

## 问题分析

数学模型:  $|x| = \begin{cases} x & x \geq 0 \\ -x & x < 0 \end{cases}$

数据结构: 两个整型变量x、y

算法:



```
#include <stdio.h>
void main()
{ int x,y;
scanf("%d",&x);
if(x>=0)
    y=x;
else
    y=-x;
printf("%d\n",y);
}
```





# if语句—双分支if语句

## ■ 语句格式

**if(表达式)**

语句1;

**else**

语句2;

**if(  $x \geq 0$  )**

**y=x;**

**else**

**y=-x;**

## 说明

语句1、语句2可以是一条单语句，也可以是复合语句。





# if语句—双分支if语句

- 例：输入三边长。若能构成三角形，计算三角形的面积；否则输出“error”。

## 问题分析

数学模型：

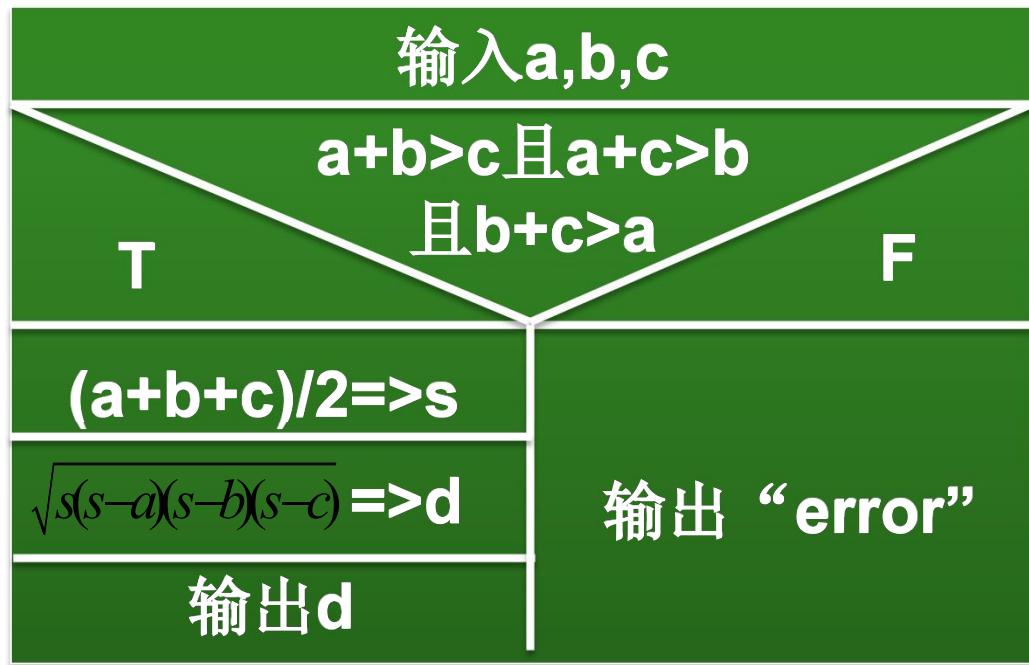
$$s = (a+b+c)/2$$

$$d = \sqrt{s(s-a)(s-b)(s-c)}$$

数据结构：

5个实型变量

算法：





# if语句—双分支if语句

```
#include <stdio.h>
#include <math.h>
void main()
{float a,b,c,s,d;
scanf("%f%f%f",&a,&b,&c);
if(a+b>c&&b+c>a&&a+c>b)
{s=(a+b+c)/2;
d=sqrt(s*(s-a)*(s-b)*(s-c));
printf("面积=%f\n",d); }
else
printf("error\n");
}
```





# 条件的表示

## ■ 关系表达式

读入一个成绩，输出“通过”或“不通过”。

## ■ 逻辑表达式

判断某年是否是闰年。

年份能被**400**整除

年份能被**4**整除，但不能被**100**整除。

## ■ 其它表达式

判断两个整型数是否相等

## ■ 变量

分析程序段的功能

`if(x)`

`y=abs(x)/x;`

`else`

`y=0;`





# 多分支结构

■ 实例：计算下列函数值。

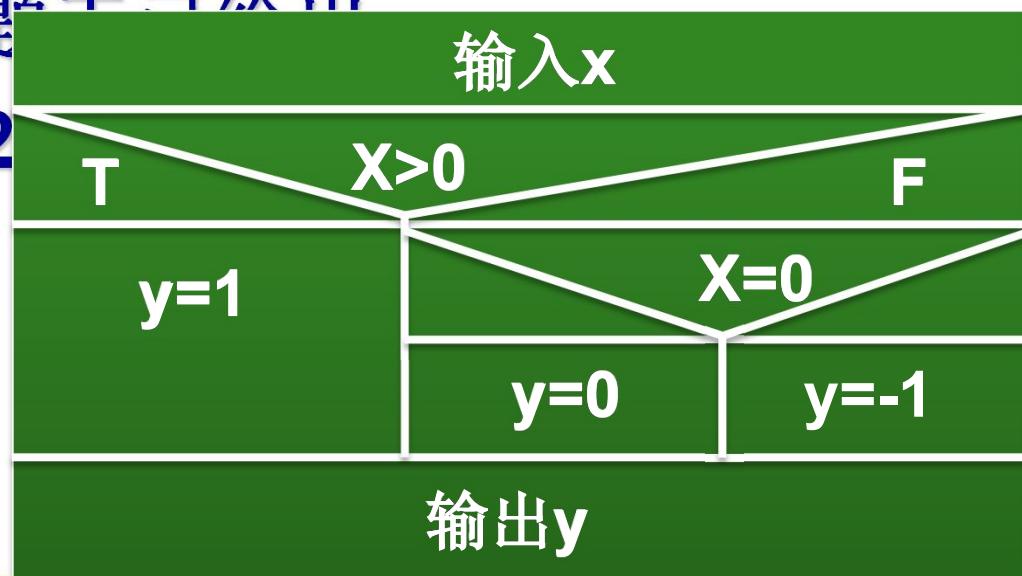
$$y = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

## 问题分析

数学模型：~~题干已给出~~

数据结构：~~2~~

算法：





# 多分支结构—多分支if语句

```
#include <stdio.h>
void main()
{int x,y;
scanf("%d",&x);
if(x>0)
    y=1;
else if(x==0)
    y=0;
else
    y=-1;
printf("%d\n",y);
}
```

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3)** 语句3;

.....

**else if(表达式m)** 语句m;

**else** 语句m+1;





# 多分支结构—多分支if语句

■ 例：把一个百分制成绩转换成五分制成绩。

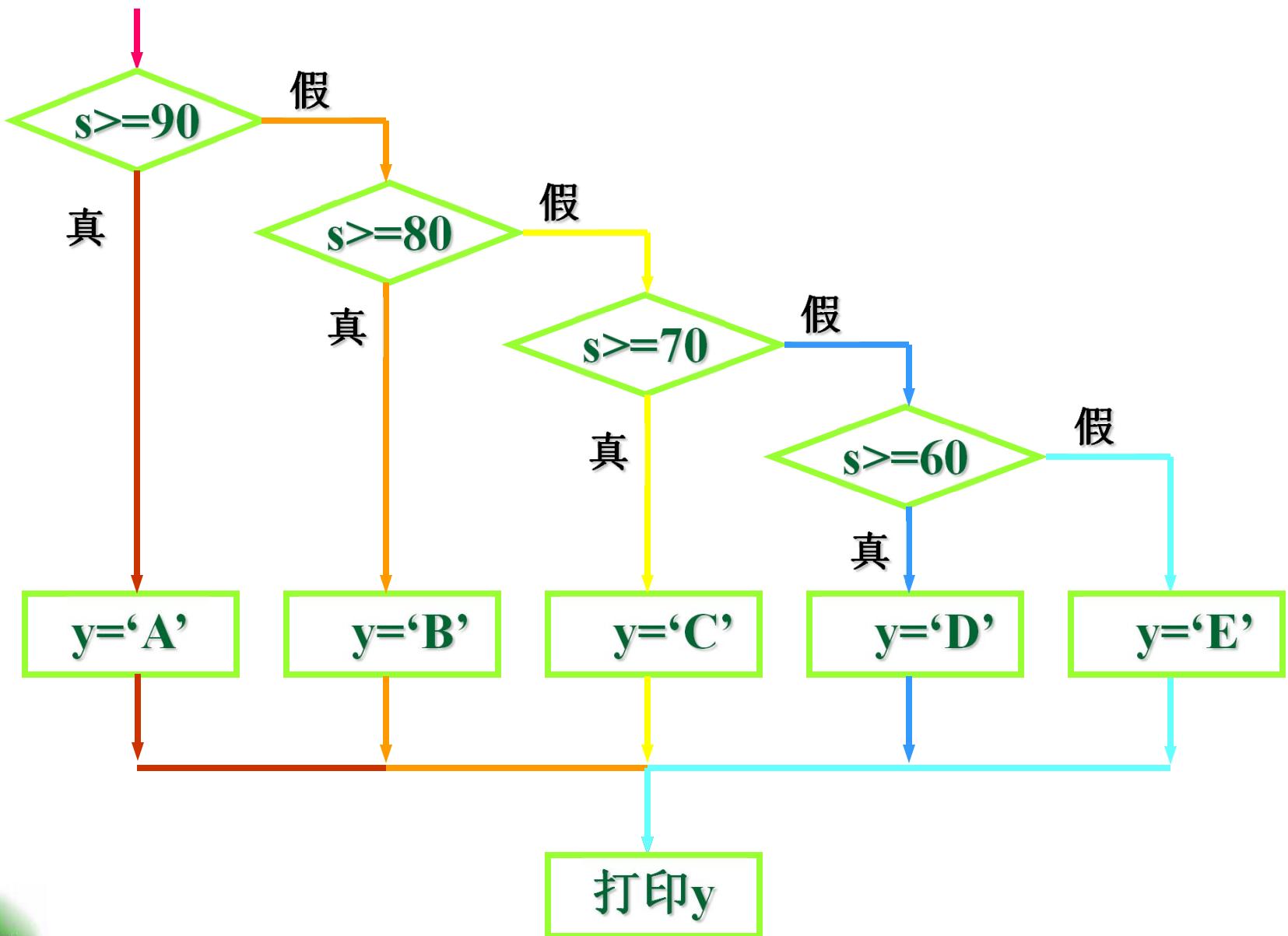
转换规则：

<b>100~90</b>	<b>89~80</b>	<b>79~70</b>	<b>69~60</b>	<b>59~0</b>
优秀	良好	中等	及格	不及格

5个分支。

转换过程：







# 多分支结构—多分支if语句

```
#include "stdio.h"  
void main( )  
{int s;char y;  
scanf("%d",&s);  
if(s>=90) y='A';  
else if(s>=80) y='B';  
else if(s>=70) y='C';  
else if(s>=60) y='D';  
else y='E';  
printf("成绩为: %c\n",y);}
```

多分支if语句的弱点：

分支增多，程序可读性差。





# 多分支结构—switch语句

```
#include "stdio.h"
void main( )
{ int s;char y;
scanf("%d",&s);
switch(s/10)
{case 10: y='A';break;
 case 9: y='A';break;
 case 8: y='B';break;
 case 7: y='C';break;
 case 6: y='D';break;
default: y='E';break;
}
printf("成绩为: %c\n",y);}
```

特点:

**break:**

中止switch语句的执行。



没有**break**

程序如何执行? ,  
能不能间与?





# 多分支结构—switch语句

整型  
字符型

整型  
字符型

## 语句格式

**switch (表达式)**

{ case

case

case

default :

}

常量 1:

常量 2:

⋮

常量 n:

语句1

语句2

语句n

语句n+1





# 多分支结构—switch语句

## ■ 执行过程

(1) 求解表达式的值

(2) 寻找入口：

(3) 执行相应的语句

有**break**: 结束**switch**语句

无**break**: 继续执行其下面分句





# 多分支结构—switch语句

例：输入一个0~6的整数，转换成星期输出。

0	1	2	3	4	5	6	其它
输出 星期日	输出 星期一	输出 星期二	输出 星期三	输出 星期四	输出 星期五	输出 星期六	输出 error





# 多分支结构—switch语句

```
#include <stdio.h>
void main( )
{ int day;
    scanf("%d",&day);
    switch (day)
    { case 0: printf( "Sunday\n"); break;
        case 1: printf( "Monday \n"); break;
        case 2: printf( "Tuesday \n") ; break;
        case 3: printf( "Wednesday \n"); break;
        case 4: printf( "Thursday \n"); break;
        case 5: printf( "Friday \n"); break;
        case 6: printf( "Saturday \n"); break;
        default: printf( " Input Error "); break; }
}
```





# 多分支结构—switch语句

分析程序运行结果

```
#include "stdio.h"
void main()
{int x=1,y=0,a=0,b=0;
switch(x)
{ case 1: switch(y)
{case 0: a++;break;
 case 1:b++;break;
}
case 2: a++;b++;break;
case 3: a++;b++;
}
printf("a=%d,b=%d\n",a,b);
}
```





# if嵌套结构

- 在if语句的某个分支中又完整地包含了另外一个if语句，称为if的嵌套。
- 用于解决多分支问题

例如：求下列函数值。

$$f(x) = \begin{cases} x & x < 1 \\ 2x-1 & 1 \leq x < 10 \\ 3x-11 & x \geq 10 \end{cases}$$

```
#include "stdio.h"
void main()
{ float x,y;
  scanf("%f", &x);
  if(x>=1)
    if(x>=10)
      y=3*x-11;
    else
      y=2*x-1;
  else y=x;
  printf("y=%f\n", y); }
```





# if嵌套结构

## ■ 形式:

**if (表达式1)**

**if (表达式2)** 语句2

**else** 语句3

**else**

**if (表达式3)** 语句4

**else** 语句5

内层

说明

可以同时出现，  
可以单独出现。





# if嵌套结构

**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3

**else**

**if (表达式2)** 语句1

**else** 语句2

**if (表达式1)**

**if (表达式2)** 语句1

**else** 语句2

关键

清楚**if**与**else**  
的对应关系





# if嵌套结构

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

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

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





# if嵌套结构

```
#include "stdio.h"
void main()
{ int x,y;
  scanf("%d", &x);
  if(x>0)
    y=1;
  else
    if(x==0)
      y=0;
    else
      y=-1;
  printf("y=%d\n", y);
```

```
#include "stdio.h"
void main()
{int a=2,b=-1,c=2;
  if(a<b)
    if(b<1)
      c=0;
    else
      c+=1;
  printf("%d\n",c);}
```

与哪个  
if匹配



# if嵌套结构

## 说明

- ① 可以在**if** 分支中嵌套，  
也可在**else**分支中嵌套。
- ② 嵌套时注意**if**与**else**的对应关系。  
**else**子句与**if** 的匹配原则：  
与在它上面、距它最近、  
且尚未匹配的**if** 配对。
- ③为明确匹配关系，避免匹配错误，  
建议将内嵌的**if** 语句用花括号括起来。





# 程序举例

## 分析程序运行结果

输入分别是：

A、B、C、D、  
E时，程序运行结果  
分别是什么？

关键点：

- 1) 表达式`++c`的值。
- 2) 子句后没有`break`，  
程序执行的顺序。

思考：什么时候执行  
`case 'A':`子句？

```
#include<stdio.h>
void main()
{ int k=1;
char cm;
cm=getchar();
switch(++cm)
{case 'A': k++;break;
case 'B': k--;
case 'C': k+=2;
case 'D': k=k%2;
case 'E': k=k*10;break;
default:k=k/3;}
k++;
printf("k=%d\n",k);}
```





# 程序举例

```
#include "stdio.h"  
void main( )  
{ float x,y; char z;  
scanf("%f%c%f",&x,&z,&y);  
switch(z)  
{case '+':printf("%f\n",x+y);break;  
case '-': printf("%f\n",x-y);break;  
case '*': printf("%f\n",x*y);break;  
case '/': printf("%f\n",x/y);break;  
default: printf("Input error!\n");  
}
```

- 1) 程序运行时可能  
出现什么问题?
- 2) 如何完善程序?





# 程序举例

■ 编程：输入年份，判断该年是否是闰年？

解决方法：

使用双分支if语句。

闰年条件： $(y \% 4 == 0 \& \& y \% 100 != 0) || y \% 400 == 0$

使用多分支结构。

闰年的条件：

{ 年份不能被4整除：不是；

{ 能被4整除，{ 但不能被100整除：是

{ 能100整除，{ 能被400整除：是

{ 否则：不是





# 程序举例

## ■ 算法的关键

1) 如何实现多分支?

使用嵌套if语句

2) 如何记录是闰年或不是闰年?

使用标记变量:

$f=1$  是闰年;  $f=0$  不是闰年



# 程序举例

## ■ 算法

读入年份=>y

{如果y不能被4整除: f=0

{否则: {如果y不能被100整除}

{否则: {如果y能被400整除  
    {否则: f=0

如果f=1: 输出“是闰年”

否则: 输出“不是闰年”

```
#include <stdio.h>
void main()
{int y,f;
scanf("%d",&y);
if(y%4!=0)
    f=0;
else
    if(y%100!=0)
        f=1;
    else
        if(y%400==0)
            f=1;
        else
            f=0;
if(f)
    printf("是闰年。 \n");
else
    printf("不是闰年。 \n"); }
```



# 本章小结

