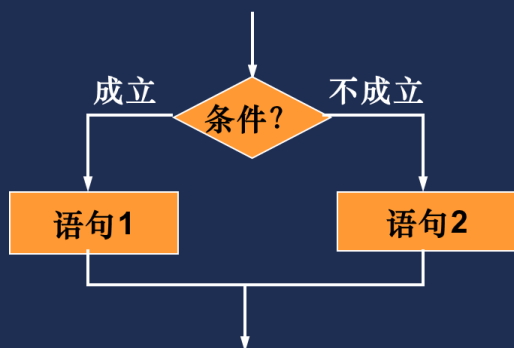


# 实验3-1、3-2

- 使用if语句编程求解两个数中的最大值。
- 从键盘输入一个数，判定其能否被7整除。



## 语法格式

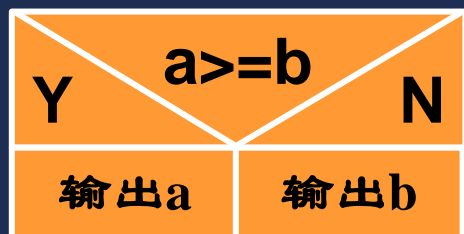
```
if(条件表达式)
{语句1;}
else
{语句2;}
```

条件表达式的括号  
后面没有分号;

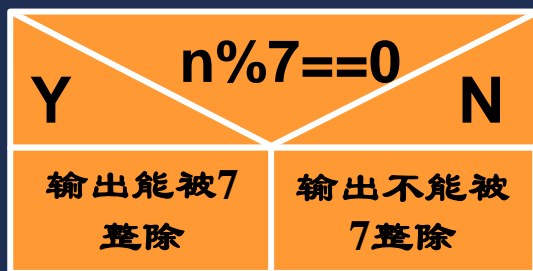
说明:

“语句1”，“语句2”可以是各种可执行的单语句(只有一条执行语句)或复合语句(有多条执行语句)。如果是单语句，花括号可省略，若是复合语句，则花括号不能省。

**else分句不能单独使用!**



实验3-1算法



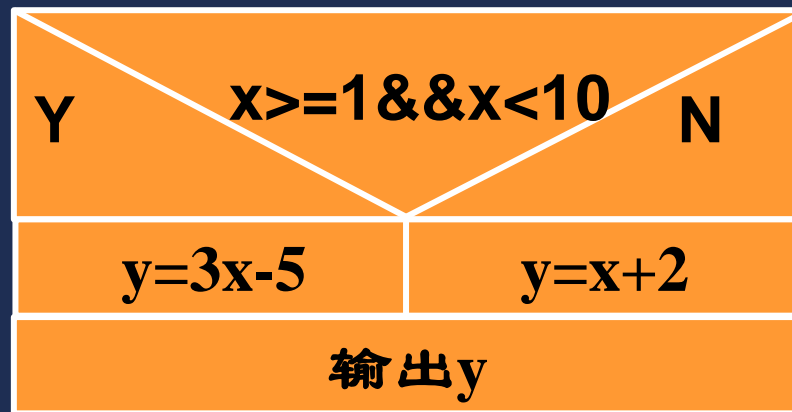
实验3-2算法

# 实验3-3

- 已知一个函数  $y = \begin{cases} 3x - 5 & (1 \leq x < 10) \\ x + 2 & (\text{其他}) \end{cases}$

从键盘输入X的值，求解并输出相应的y值。

- 算法思想：设置变量X和y，根据X的取值范围不同y有不同的值。注意X的取值范围  $1 \leq x < 10$  如何表示？



# 实验3-4

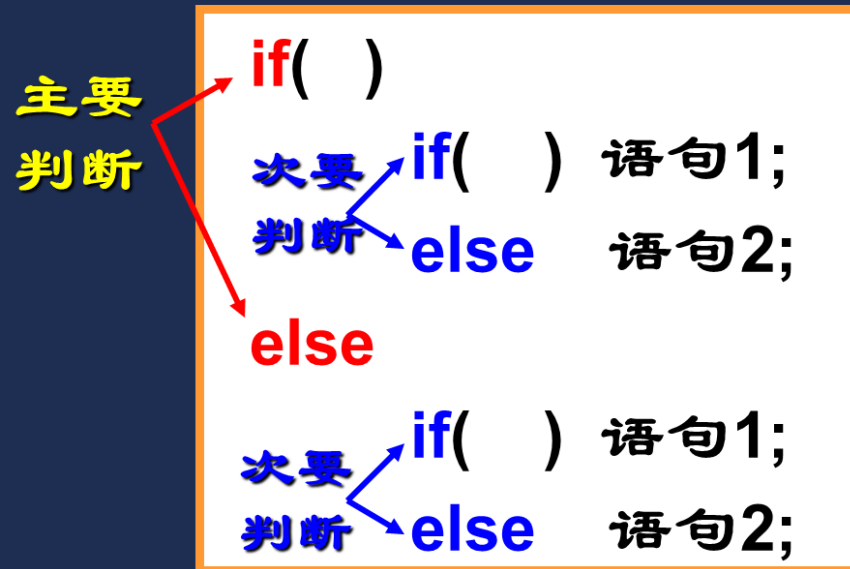
- 李某购买一批打印纸，若一次性购买100包以下，每包20元；若一次性购买100包以上（包含100包），则每包16元，编写程序，计算李某购买N包打印纸需要多少钱？
- 算法思想：将购买打印纸的数量定义为变量N，每包单价为price，N的值从键盘输入；price的值根据N的大小进行确定，最后计算出所需的总价格total。

$$\text{price} = \begin{cases} 20 & (N < 100) \\ 16 & (N \geq 100) \end{cases}$$

N >= 100	
Y	N
price=16	price=20
total=N*price	
输出total	

# 嵌套选择结构

- 嵌套的选择结构，即在条件成立时执行的语句部分或在条件不成立时执行的语句部分中包含另一个选择结构的实现语句。
- 由外层选择结构执行的判断，称为**主要判断**。
- 由内层选择结构执行的判断，称为**次要判断**。次要判定依赖于主要判定。



# 注意

如果if和else的数目不一样，else总是与它上面的最近的未配对的if配对，也可加花括号来确定配对关系。

例：

```
if( )  
if( ) 语句1;  
else  
if( ) 语句1;  
else 语句2;
```

例：

```
if( )  
    if( ) 语句1;  
    else  
        if( ) 语句1;  
        else 语句2;
```

例：

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

# 实验3-5

- 个税计算问题。根据国家的纳税政策，个人应税起征点为月收入3500元；不超过起征点1500元的，超过部分按3%交纳个人所得税；超过起征点1500至4500元的，超过部分按10%交纳个人所得税。编写一个程序，输入某人的应税月收入，计算并输出纳税后的实际收入。

- 算法思想：设变量M表示收入，tax表示税收。

$$\text{tax} = \begin{cases} 0 & (M \leq 3500) \\ (M - 3500) * 0.03 & (3500 < M \leq 5000) \\ 1500 * 0.03 + (M - 5000) * 0.1 & (5000 < M \leq 8000) \\ \text{输出“暂时不考虑”} & (M > 8000) \end{cases}$$

# 实验3-5

- 算法思想：设变量M表示收入，tax表示税收。

$$\text{tax} = \begin{cases} 0 & (M \leq 3500) \\ (M - 3500) * 0.03 & (3500 < M \leq 5000) \\ 1500 * 0.03 + (M - 5000) * 0.1 & (5000 < M \leq 8000) \\ \text{输出“暂时不考虑”} & (M > 8000) \end{cases}$$

Y	M≤3500			N	
tax=0	M≥3500&&M≤5000			N	
输出 tax, M	tax=	M>5000&&M≤8000		输出“暂时不考虑!”	
	(M-3500)*0.03	Y			N
	M=M-tax	tax=1500*0.03+			
		(M-5000)*0.1			
		M=M-tax			
	输出tax,M	输出tax,M			

嵌套的选择结构

if( )

else

if( )

else

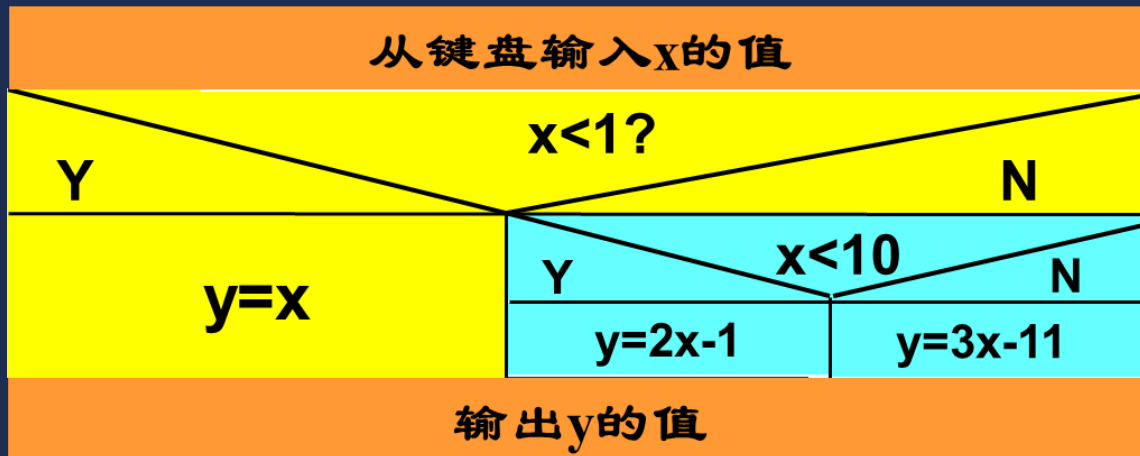
if( )

else

# 实验4-1

- 已知一函数，输入x的值，求解并输出相应的y值。

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



嵌套的选择结构

if( )

else

if( )

else

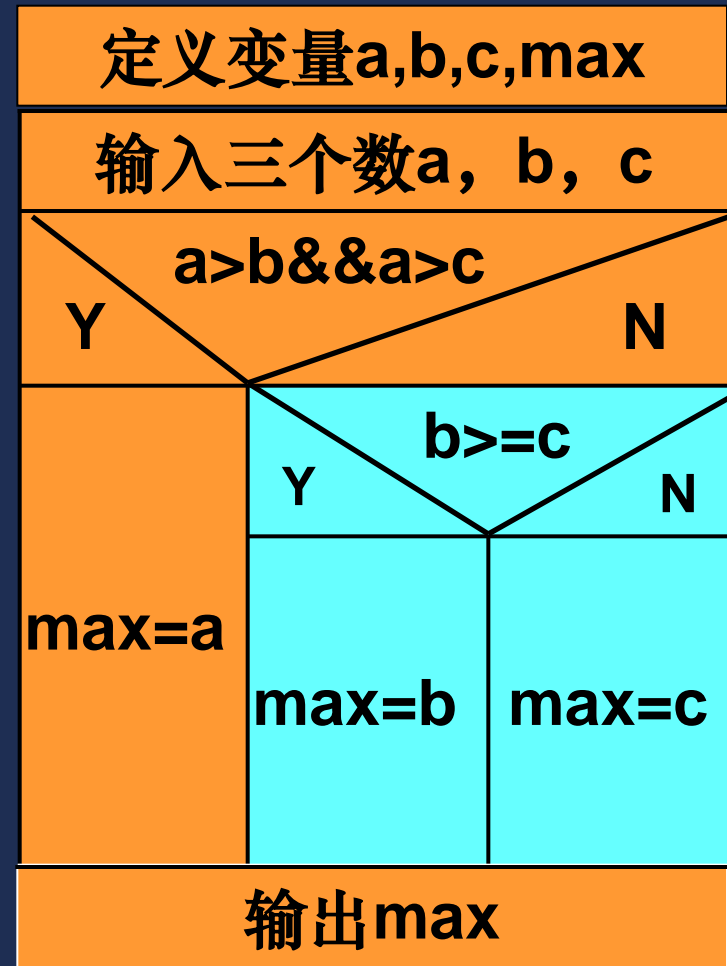


# 实验4-2 方法一

- 从键盘任意输入三个数，计算并输出它们的最大数。

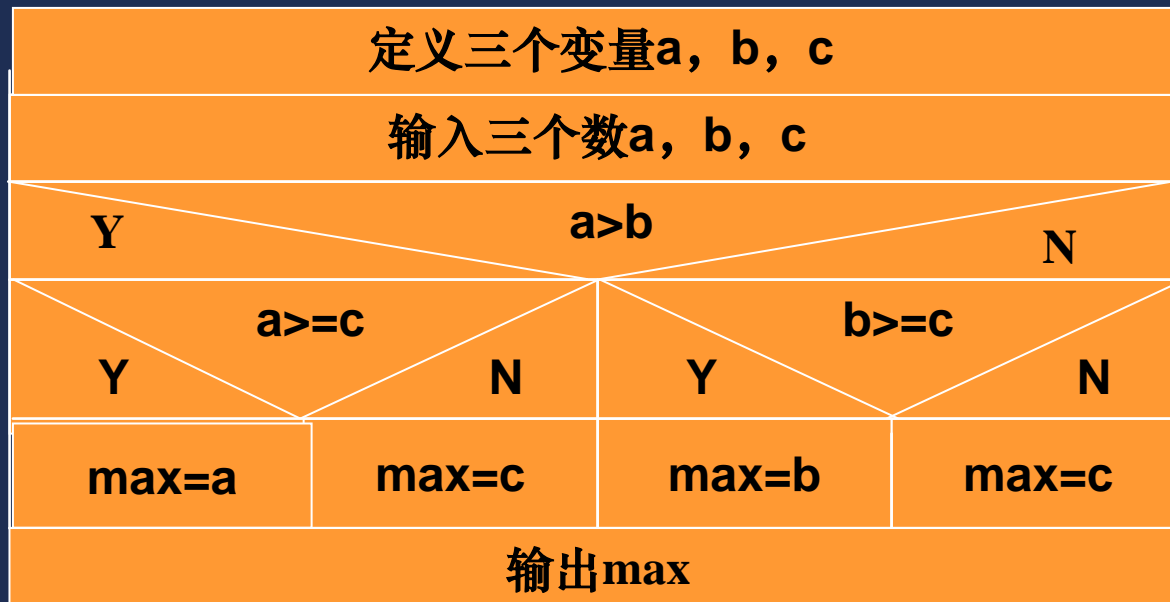
- 思路分析：

假设用 $a$ ， $b$ ， $c$ 分别表示输入的三个数。首先判断 $a$ 是否为最大值，如果是，就直接输出判断结果；如果不是，则最大值必然在 $b$ 和 $c$ 中，此时再判断 $b$ 和 $c$ 的大小关系就可以得出整个判断结果。



# 实验4-2 方法二

- 从键盘任意输入三个数，计算并输出它们的最大数。
- **分析：**可以先在a和b中找出一个较大值出来；然后再用这个较大值与剩下的数c进行比较，其中的较大者即为三个数中的最大者。



# 实验4-3

- 某工厂有如下规定：工龄不到一年的（工龄为0），无年休假；工龄为1~5年的，年休假为7天；工龄为6~10年的，年休假为14天；工龄为11~20年的年休假为21天；工龄20年以上的，年休假为28天。编程实现：输入工龄，输出对应的年休天数。要求用if...else的嵌套结构实现。

工龄	休假天数
不足一年(0)	0
1~5	7
6~10	14
11~20	21
20年以上	28

嵌套的选择结构

```
if(age==0 )
```

```
else
```

```
    if( age>=1 &&age<=5)
```

```
    else
```

```
        if(age>=6&&age<=10)
```

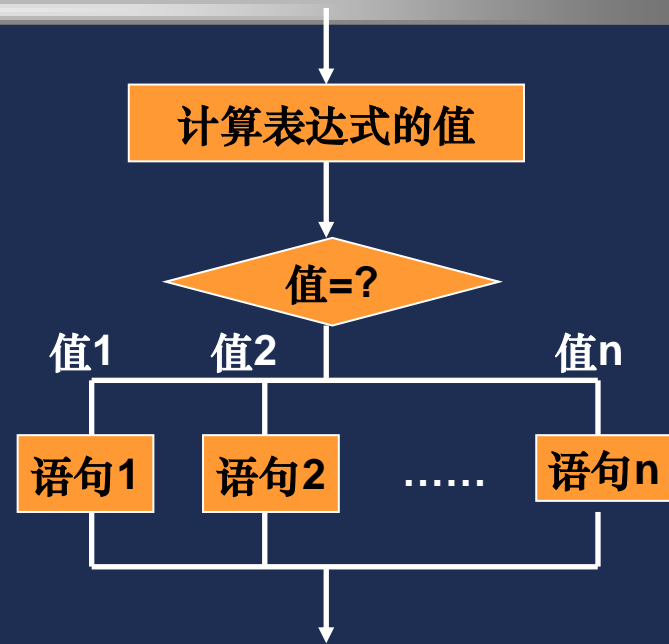
```
        else
```

```
            if(age>=11&&age<=20)
```

```
            else
```

# 多路(开关)switch选择语句 P<sub>60</sub>

```
switch (表达式)
{
    case 值1: 语句1;
    case 值2: 语句2;
        ...
    case 值n: 语句n;
    default: 语句;
}
```



1. 先计算switch语句中表达式的值，此表达式可以包含常量、变量、函数和运算符的组合，结果应该是整型或字符型。
2. 每个case后的值称为“开关常数”，为整数或字符型常量，值不能相同，只起入口标号的作用。
3. 寻找与表达式值匹配的case语句，由此开始执行此case以下的所有语句。如果没有找到与之匹配的case语句，则从default语句开始执行。

# break语句

P<sub>60</sub>

```
switch (表达式)
{
case值1: 语句;
           break;
case值2: 语句;
           break;
           ⋮
case值n: 语句序列n;
           break;
default: 语句;
}
```

如果让计算机在执行了匹配的case语句后退出switch，不再执行switch中其他case或default语句，需要在每个case分支的最后加break语句，用来结束整个的switch选择结构。

# 实验4-4

- 文具厂销售笔袋，按照客户的购买数量不同，给出了不同的优惠政策，详见下表。要求编程实现：输入客户购买笔袋数量，输出购买总金额。**要求用switch语句实现。**

数量 (个)	单价 (元/个)
1~49	38
50~99	35
100~199	32
200~299	30
300及以上	28

数量 (个)	$c=num/50$	单价 (元/个)
1~49	0	38
50~99	1	35
100~199	2~3	32
200~299	4~5	30
300及以上	6及以上	28

switch语句要求条件表达式取值为确定的若干个开关量，而不能使用关系表达式。但是分析发现，数量num的分段点均是50的倍数，因此，将数量除以50，取整数商c，可得到若干整数值， $c=num/50$

# 实验4-4

数量 (个)	$c=num/50$	单价 (元/个)
1~49	0	38
50~99	1	35
100~199	2~3	32
200~299	4~5	30
300及以上	6及以上	28

```
int c;  
c=num/50;  
switch(c)  
{  
    case 0: price=38;break;  
    case 1: price=35;break;  
    case 2:  
    case 3: price=32;break;  
    case 4:  
    case 5: price=30;break;  
    default : price=28;  
}  
total=num*price;
```

## 4-5 模拟二人猜拳游戏：1.剪刀 2.石头 3.布。

算法分析：由于有两个游戏者，**每个游戏者都有三种可能的输入**，对**第一个游戏者的某种输入**（如输入1），再**对应第二个游戏者的3种不同输入**就可以得到相应的输赢信息，因此要使用**switch嵌套语句**。

定义变量 int first, second;

从键盘输入 first和 second的值（1:剪刀；2:石头；3: 布）；

```
switch(first)
```

```
{
```

```
case 1: switch(second)
```

```
{
```

```
case 1: cout<<"剪刀平剪刀."<<endl; break;
```

```
case 2: cout<<"石头胜剪刀."<<endl; break;
```

```
case 3: cout<<"剪刀胜布."<<endl; break;
```

```
default : cout<<" 第二次输入有误!"<<endl ; break;
```

```
}
```

```
break;
```

```
case 2: switch(second) { ..... }
```

```
break;
```

```
case 3: switch(second) { ..... }
```

```
break;
```

```
default : cout<<"第一次输入有误!"<<endl;
```

```
}
```



**补充：** 输入某年某月某日，判断这一天是这一年的第几天。  
**提示：** 先根据输入的年份y，确定是否为闰年（闰年算法可参考流程图），然后再判断m是几月（可用swith语句），根据常识（一月大，二月平，三月大.....）即可获知天数。

