

# 第4章 选择结构程序设计（2）

  
S.IST@XMU

# 复习回顾

Ø 上次课的内容：

u 选择结构基本概念

u 基本if语句

l if

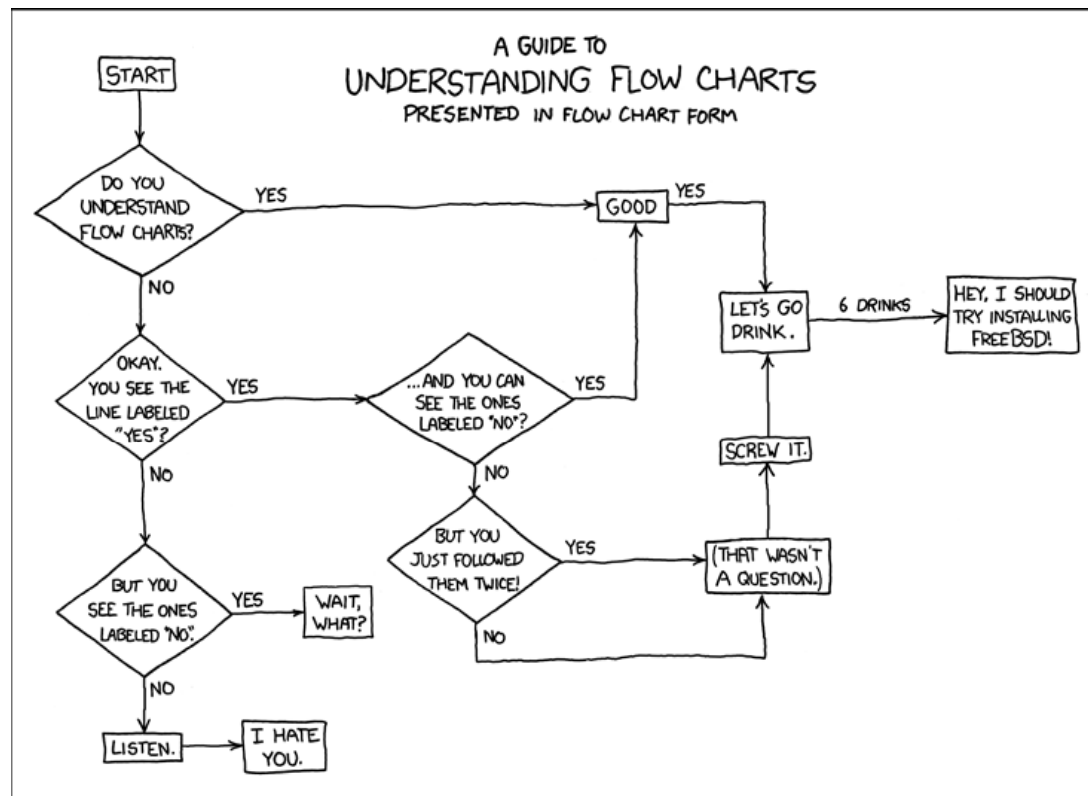
l if...else...

l else if

u 关系运算符和表达式

u 逻辑运算符和逻辑表达式

u 初学者建议借助流程图规划分支结构，入门练习如上



# 关于实验课（地点：实验楼401）

Ø OJ ( Online Judge ) 常用于ACM国际大学生程序设计竞赛，用机器自动判定程序解答正确与否

Ø 游戏规则：**必须严格按照样例的格式进行输入输出。**

- scanf的格式控制符必须与样例的规定完全相符

- 输入前不要打印提示信息

- 输出格式多一个字符少一个字符都不行

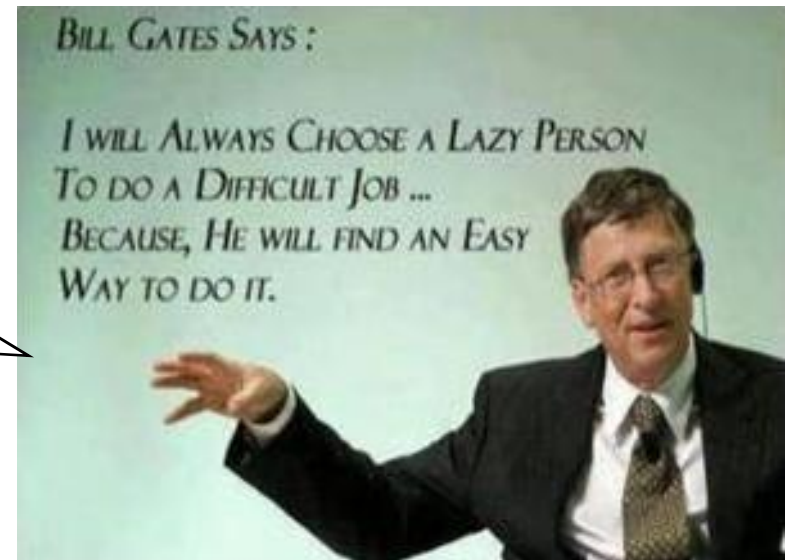
- 每行行首不应有空格

- 每行输出一般以回车符结束

**! Be careful !**

# 一个优秀程序员应该是个懒人？

我总是希望找一个懒人来  
做一件困难的工作...  
因为他们总能找到**简便**  
**的解决办法。**



## Ø 目前见过的C语言里的偷懒

`u i=i+1;` 可以写成 `++i;`

`u a=a+b;` 可以写成 `a+=b;`

`u if (n!=0)` 可以写成 `if(n)`

`u` 还有接下来的**条件运算符和条件表达式...**



# 条件运算符

Ø C语言中唯一的三元运算符

Ø 用来简洁地实现类似于if...else...的语句结构

Ø 如

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

条件运算符

$\text{max} = (\text{a} > \text{b}) ? \text{a} : \text{b};$

**注意：**条件运算符是一对，如 $\text{max} = \text{a} > \text{b} ? \text{a};$ （缺少：）或 $\text{max} = \text{a} : \text{b};$ （缺少？）都是错的

# 条件表达式

Ø C语言中唯一的三元运算符

Ø 用来简洁地实现类似于if...else...的语句结构

Ø 如 if (a > b)

max = a;

else

max = b;

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

条件表达式

# 条件表达式的形式和执行顺序

Ø 条件表达式的一般形式为

表达式 1 ? 表达式 2 : 表达式 3

Ø 条件运算符的执行顺序：

1. 求解表达式1

2. 若表达式1的值为非0（真）则求解表达式2，此时表达式2的值就作为整个条件表达式的值

3. 若表达式1的值为0（假），则求解表达式3，表达式3的值就是整个条件表达式的值

u 如  $x > 0 ? 2 : -2$ ，如果 $x$ 大于0，整个表达式结果为2；  
否则，整个表达式结果为-2。

# 条件运算符的优先级和结合性

Ø 条件运算符的**优先级**高于赋值运算符，但是低于关系运算符和算术运算符。结合性是**自右向左**。

└ 例如  $a < b ? a : c > d ? c : d$  (**不推荐这么写！**)

└ 相当于  $a < b ? a : (c > d ? c : d)$

└ 如有  $c = a > b ? a : b + 1$

└ 相当于  $c = a > b ? a : (b + 1)$

└ **而不是**  $c = (a > b ? a : b) + 1$



# 这样的条件表达式也是合法的

Ø 例如

```
⌈ a > b ? (max=a) : (max=b);
```

相当于

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

```
⌈ a > b ? printf("%d",a) : printf("%d",b);
```

相当于

```
if (a > b) { printf("%d",a); }  
else { printf("%d",b); }
```

Ø **建议**尽量将条件表达式的使用局限于赋值语句

Ø **不建议**用条件表达式实现复杂的分支结构

# 条件表达式应用实例

Ø 输入一个字符，判别它是否大写字母，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（要求用条件表达式处理）

```
1. #include <stdio.h>
2. int main()
3. {
4.     char ch;
5.     scanf("%c",&ch);
6.     ch = (ch>='A' && ch<='Z')?(ch+32):ch;
7.     printf("%c\n",ch);
8.     return 0;
9. }
```



# if语句的嵌套

Ø 在if...else语句的语法中，**语句1**和**语句2**可以是任何可执行语句，当然也可以是if语句。这就形成if语句的嵌套。通过if语句的嵌套，可以表示多分支结构，形式例如：

形式1

```
1. if(表达式1)
2. {
3.     if(表达式1.1)
4.         语句1.1
5. }
6. else
7.     语句2
```

形式2

```
1. if(表达式1)
2. {
3.     if(表达式1.1)
4.         语句1.1
5.     else
6.         语句1.2
7. }
8. else
9.     语句2
```

形式3

```
1. if(表达式1)
2. {
3.     if(表达式1.1)
4.         语句1.1
5.     else
6.         语句1.2
7. }
8. else
9. {
10.    if(表达式2.1)
11.        语句2.1
12.    else
13.        语句2.2
14. }
```

# if与else的配对

Ø if语句的嵌套很容易混淆，要注意if与else的配对关系

Ø C语句规定，else总是与它上面最近的一个未配对的if配对

执行顺序？

```
if(表达式1)
if(表达式2)
    语句1
else
if(表达式3)
    语句2
else
    语句3
```

实际执行

```
if(表达式1)
{
    if(表达式2)
        语句1
    else
    {
        if(表达式3)
            语句2
        else
            语句3
    }
}
```

错误理解

```
if(表达式1)
{
    if(表达式2)
        语句1
}
else
{
    if(表达式3)
        语句2
    else
        语句3
}
```

若想这样执行，请  
适当添加大括号

```
if(表达式1)
{
    if(表达式2)
        语句1
}
else
{
    if(表达式3)
        语句2
    else
        语句3
}
```

# 关于if嵌套语句练习

Ø 阅读下面的程序，注意else配对问题并分析其执行结果

```
1. #include <stdio.h>
2. int main()
3. {
4.     int x=4;
5.     if (x>6)
6.         if (x<12)
7.             ++x;
8.     else
9.         --x;
10.    printf("x=%d:\n",x);
11.    return 0;
12. }
```

**【运行结果】 x=4**

```
1. #include <stdio.h>
2. int main()
3. {
4.     int x=4;
5.     if (x>6)
6.     {
7.         if (x<12)
8.         {
9.             ++x;
10.        }
11.        else
12.        {
13.            --x;
14.        }
15.    }
16.    printf("x=%d:\n",x);
17.    return 0;
18. }
```

# 缩排对if-else配对无任何影响

Ø 即使把代码排版成下面这样，运行顺序和结果并不会改变

```
1. #include <stdio.h>
2. int main()
3. {
4.     int x=4;
5.     if (x>6)
6.         if (x<12)
7.             ++x;
8.     else
9.         --x;
10.    printf("x=%d:\n",x);
11.    return 0;
12. }
```

**【运行结果】 x=4**

```
1. #include <stdio.h>
2. int main()
3. {
4.     int x=4;
5.     if (x>6)
6.     {
7.         if (x<12)
8.         {
9.             ++x;
10.        }
11.        else
12.        {
13.            --x;
14.        }
15.    }
16.    printf("x=%d:\n",x);
17.    return 0;
18. }
```

# 应该用{ } 明确if-else配对

Ø 下面两个代码的运行顺序和结果就有区别了

```
1. #include <stdio.h>
2. int main()
3. {
4.     int x=4;
5.     if (x>6)
6.     {
7.         if (x<12)
8.         {
9.             ++x;
10.        }
11.    }
12.    else
13.    {
14.        --x;
15.    }
16.    printf("x=%d:\n",x);
17.    return 0;
18. }
```

**【运行结果】 x=3**

```
1. #include <stdio.h>
2. int main()
3. {
4.     int x=4;
5.     if (x>6)
6.     {
7.         if (x<12)
8.         {
9.             ++x;
10.        }
11.        else
12.        {
13.            --x;
14.        }
15.    }
16.    printf("x=%d:\n",x);
17.    return 0;
18. }
```

**【运行结果】 x=4**

# if嵌套语句应用实例：分段函数

Ø 有一函数：

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

编一程序，输入一个x值，要求输出相应的y值。

## Ø 解题思路：

- u 用if语句检查x的值，根据x的值决定赋予y的值
- u 由于y的可能值不是两个而是三个，因此不可能只用一个简单的(无内嵌if)的if语句来实现



# if嵌套语句应用实例：分段函数

Ø 解题方法1：先后用3个独立的if语句处理

输入x

若  $x < 0$ , 则  $y = -1$

若  $x = 0$ , 则  $y = 0$

若  $x > 0$ , 则  $y = 1$

输出x和y

```
.....  
scanf("%d",&x);  
if (x<0)  
    y = -1;  
if (x==0)  
    y = 0;  
if (x>0)  
    y = 1;  
printf("x=%d,y=%d\n",x,y);  
.....
```

# if嵌套语句应用实例：分段函数

## Ø解题方法2：用一个嵌套的if语句处理

输入x

若 $x < 0$ , 则 $y = -1$

否则

若 $x = 0$ , 则 $y = 0$

否则  $y = 1$

输出x和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);  
.....
```

# if嵌套语句的编程风格

```
1. #include <stdio.h>
2. int main()
3. {
4.     double x,y;
5.     scanf("%d",&x);
6.     if (x<0)
7.     {
8.         y=-1;
9.     }
10.    else
11.    {
12.        if (x==0)
13.        {
14.            y=0;
15.        }
16.        else
17.        {
18.            y=1;
19.        }
20.    }
21.    printf("x=%d,y=%d\n",x,y);
22.    return 0;
23. }
```

Ø 分支语句坚持用大括号标记起止位置（课件上有些地方没用到是因为空间不够！）

Ø 内嵌if语句相对外层大括号应缩进

Ø 提倡内嵌if语句放在外层else分支中

Ø 内嵌if语句本身也应保持锯齿形缩进形状

# 选择结构分支太多怎么办

Ø 比如



# switch语句应用实例：分数段

Ø 要求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。成绩的等级由键盘输入。

Ø 解题思路：

- u 判断出这是一个多分支选择问题

- u 根据学生成绩的4个等级设置相应的分数段

- u 如果用if嵌套，要用3层嵌套的if，进行3次检查判断

- u 用switch语句进行一次检查即可得到结果

# switch语句应用实例：分数段

```
1. #include <stdio.h>
2. int main()
3. {
4.     char grade;
5.     scanf("%c",&grade);
6.     printf("Your score:");
7.     switch (grade)  值为A
8.     {
9.         case 'A': printf("85~100\n"); break;
10.        case 'B': printf("70~84\n"); break;
11.        case 'C': printf("60~69\n"); break;
12.        case 'D': printf("<60\n"); break;
13.        default:  printf("enter data error!\n");
14.    }
15.    return 0;
16.}
```

A  
Your score:85~100

# switch语句应用实例：分数段

```
1. #include <stdio.h>
2. int main()
3. {
4.     char grade;
5.     scanf("%c",&grade);
6.     printf("Your score:");
7.     switch (grade)
8.     {
9.         case 'A': printf("85~100\n"); break;
10.        case 'B': printf("70~84\n"); break;
11.        case 'C': printf("60~69\n"); break;
12.        case 'D': printf("<60\n"); break;
13.        default: printf("enter data error!\n");
14.    }
15.    return 0;
16.}
```



C  
Your score:60~69

值为C

如果switch后面表达式的值与case后面的常量相同，流程就跳转到此case后面的语句执行

# switch语句应用实例：分数段

```
1. #include <stdio.h>
2. int main()
3. {
4.     char grade;
5.     scanf("%c",&grade);
6.     printf("Your score:");
7.     switch (grade)
8.     {
9.         case 'A': printf("85~100\n"); break;
10.        case 'B': printf("70~84\n"); break;
11.        case 'C': printf("60~69\n"); break;
12.        case 'D': printf("<60\n"); break;
13.        default: printf("enter data error!\n");
14.    }
15.    return 0;
16.}
```

如果没有与switch后面表达式的值与匹配的case常量，流程就跳转到default后面的语句执行，如果没有default，则不执行任何语句，流程跳转到switch语句的下一个语句执行。

值为F

```
F
Your score:enter data error!
```



# switch语句应用实例：分数段

```
1. #include <stdio.h>
2. int main()
3. {
4.     char grade;
5.     scanf("%c",&grade);
6.     printf("Your score:");
7.     switch (grade)
8.     {
9.         case 'A': printf("85~100\n"); break;
10.        case 'B': printf("70~84\n"); break;
11.        case 'C': printf("60~69\n"); break;
12.        case 'D': printf("<60\n"); break;
13.        default:  printf("enter data error!\n");
14.    }
15.    return 0;
16.}
```

不能少!

# switch语句应用实例：分数段

```
1. #include <stdio.h>
2. int main()
3. {
4.     char grade;
5.     scanf("%c",&grade);
6.     printf("Your score:");
7.     switch (grade)
8.     {
9.         case 'A': printf("85~100\n"); break;
10.        case 'B': printf("70~84\n"); break;
11.        case 'C': printf("60~69\n"); break;
12.        case 'D': printf("<60\n"); break;
13.        default:  printf("enter data error!\n");
14.    }
15.    return 0;
16.}
```

```
A
Your score:85~100
70~84
60~69
<60
enter data error!
```

**case只有标记作用，不进行判断检查，如果没有break，将连续执行后续case语句，直至遇到下一个break或整个switch语句结束**

# switch语句应用实例：分数段

```
1. #include <stdio.h>
2. int main()
3. {
4.     char grade;
5.     scanf("%c",&grade);
6.     printf("Your score:");
7.     switch (grade)
8.     {
9.         case 'A': printf("85~100\n"); break;
10.        case 'B': printf("70~84\n"); break;
11.        case 'C': printf("60~69\n"); break;
12.        case 'D': printf("<60\n"); break;
13.        default: printf("enter data error!\n");
14.    }
15.    return 0;
16.}
```



A  
Your score:60~69

# switch语句应用实例：分数段

```
1. #include <stdio.h>
2. int main()
3. {
4.     char grade;
5.     scanf("%c",&grade);
6.     printf("Your score:");
7.     switch (grade)
8.     {
9.         case 'a':
10.        case 'A': printf("85~100\n"); break;
11.        case 'b':
12.        case 'B': printf("70~84\n"); break;
13.        case 'c':
14.        case 'C': printf("60~69\n"); break;
15.        case 'D': printf("<60\n"); break;
16.        default: printf("enter data error!\n");
17.    }
18.    return 0;
19.}
```

输入a或A

Your score:85~100

输入d

Your score:enter data error!

多个case标号可共用一组执行语句

# switch语句应用实例：分数段

```
1. #include <stdio.h>
2. int main()
3. {
4.     char grade;
5.     scanf("%c",&grade);
6.     printf("Your score:");
7.     switch (grade)
8.     {
9.         case 'A': printf("85~100\n"); break;
10.        case 'B': printf("70~84\n"); break;
11.        case 'C': printf("60~69\n"); break;
12.        case 'D': printf("<60\n"); break;
13.        default:  printf("enter data error!\n");
14.    }
15.    return 0;
16.}
```

此行位置有问题，  
应如何修改？



F  
Your score:enter data error!

# switch语句应用实例：分数段

```
1. #include <stdio.h>
2. int main()
3. {
4.     char grade;
5.     scanf("%c",&grade);
6.     printf("Your score:");
7.     switch (grade)
8.     {
9.         case 'A':
10.            printf("Your score:");
11.            printf("85~100\n");
12.            break;
13.        case 'B':
14.            printf("Your score:");
15.            printf("70~84\n");
16.            break;
17.        case 'C':
18.            printf("Your score:60~69\n");
19.            break;
20.        case 'D':
21.            printf("Your score:<60\n"); break;
22.        default: printf("enter data error!\n");
23.    }
24.    return 0;
25.}
```

每个case后面可有多个语句，此处视为一个复合语句，可以没有花括号

# switch语句的作用和注意事项

Ø 根据表达式的值，使流程跳转到不同的语句

Ø switch语句的一般形式：

switch ( **表达式** ) **必须是整数类型(包括字符型)**

```
{  
    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  
}
```

# switch语句的作用和注意事项

Ø 根据表达式的值，使流程跳转到不同的语句

Ø switch语句的一般形式：

```
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
```

```
}
```

**必须是常量，且不能相同**



# switch语句的作用和注意事项

Ø 根据表达式的值，使流程跳转到不同的语句

Ø switch语句的一般形式：

```
switch ( 表达式 )
```

```
{
```

```
    default      : 语句n+1
```

```
    case 常量2   : 语句2
```

```
    case 常量1   : 语句1
```

```
        |       |       |
```

```
    case 常量n   : 语句n
```

```
}
```

**各个case标号出现次序  
不影响执行结果**

# switch语句的编程风格

```
1. #include <stdio.h>
2. int main()
3. {
4.     char grade;
5.     scanf("%c",&grade);
6.     printf("Your score:");
7.     switch (grade)
8.     {
9.         → case 'A':
10.            → printf("Your score:");
11.            → printf("85~100\n");
12.            → break;
13.         → case 'B':
14.            printf("Your score:");
15.            printf("70~84\n");
16.            break;
17.         → case 'C':
18.            printf("Your score:60~69\n");
19.            break;
20.         → case 'D': printf("Your score:<60\n"); break;
21.         → default: printf("enter data error!\n");
22.     }
23.     return 0;
24. }
```

Ø switch表达式之后应有花括号，独立成行且垂直对齐

Ø 各标号应缩进且垂直对齐

Ø case语句有较多语句时则应相对标号缩进，本身应保持锯齿形状

Ø case语句较短时可与标号同一行

# 选择结构程序设计方法

- Ø **第一步**：确定分支条件
- Ø **第二步**：按照题目要求画出与分支结构相关的流程图
- Ø **第三步**：根据前面的流程图选择实现方式
- Ø **第四步**：按照选择的实现方式写出若干程序块
- Ø **第五步**：将所有程序块按照逻辑进行组合，并加上需要包含的头文件等其他内容，形成最终的程序。

# 选择结构程序综合举例：闰年

Ø 写一程序，判断某一年是否闰年

Ø 解题思路：判定闰年需满足两个条件：①能被4整除，但不能被100整除。或②能被400整除

u **第一步**：确定分支条件

l 条件1：年份是否可以被4整除。

l 条件2：若1成立，年份是否可以被100整除。

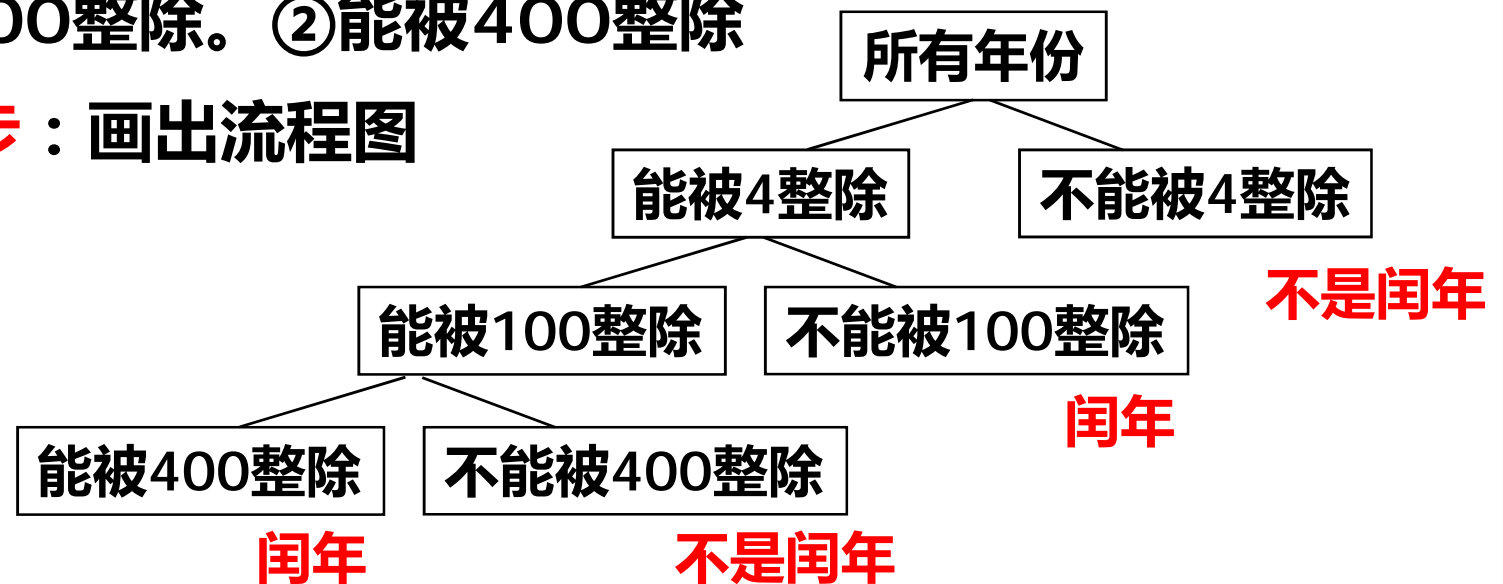
l 条件3：若2成立，年份是否可以被400整除。

# 选择结构程序综合举例：闰年

Ø 写一程序，判断某一年是否闰年

Ø 解题思路：判定闰年需满足两个条件：①能被4整除，但不能被100整除。②能被400整除

u 第二步：画出流程图



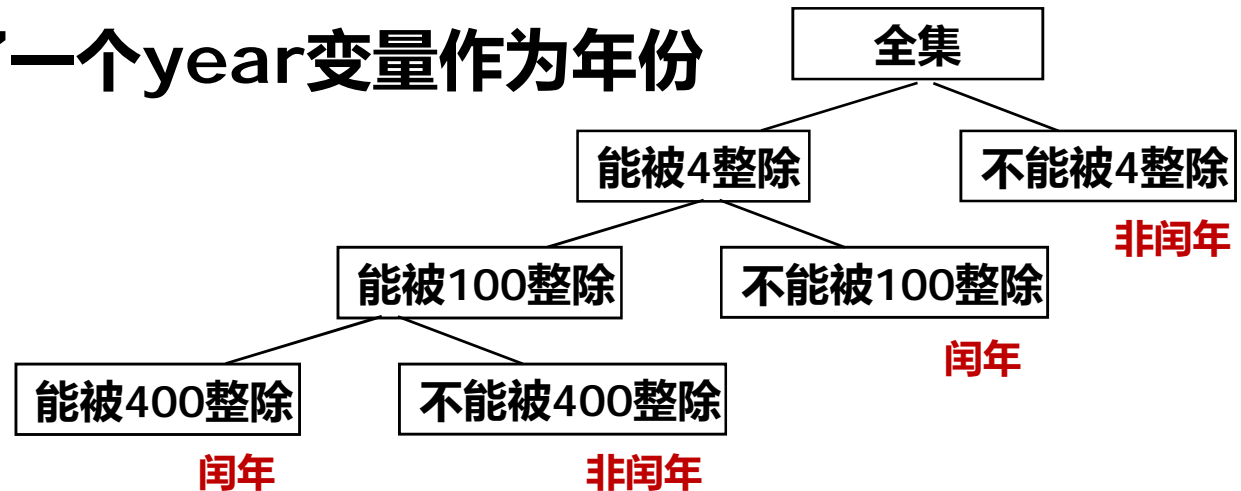
u 第三步：选择实现方式。

l 从流程图可见，可使用if嵌套语句。

# 选择结构程序综合举例：闰年

## 第四步：写出语句块

假设我们定义了一个year变量作为年份



程序块1

```
.....  
if (year%4==0)  
    继续判断（嵌入程序块2）;  
else  
    输出“不是闰年”;  
.....
```

程序块2

```
.....  
if (year%100==0)  
    继续判断（嵌入程序块3）;  
else  
    输出“是闰年”;  
.....
```

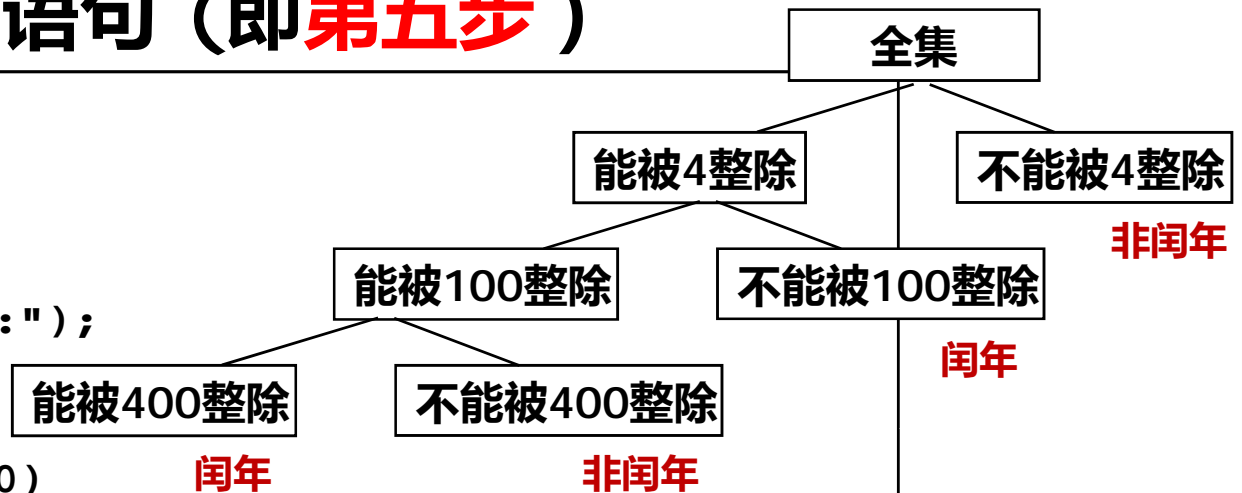
程序块3

```
.....  
if (year%400==0)  
    输出“是闰年”;  
else  
    输出“不是闰年”;  
.....
```

# 选择结构程序综合举例：闰年

## Ø 方法一：嵌套if语句（即第五步）

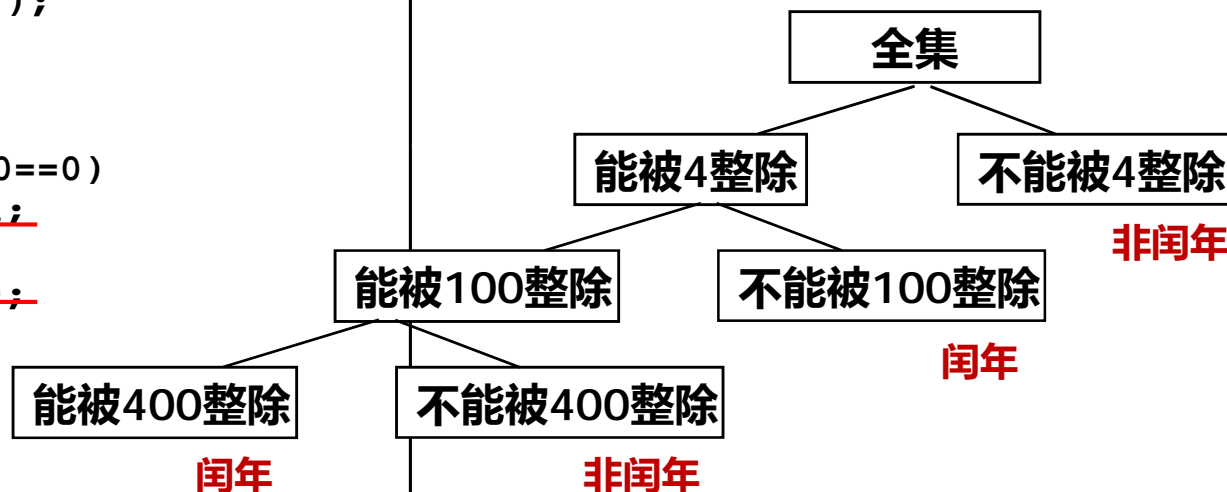
```
1. #include <stdio.h>
2. int main()
3. {
4.     int year;
5.     printf("enter year:");
6.     scanf("%d",&year);
7.     if (year%4==0)
8.         if (year%100==0)
9.             if (year%400==0)
10.                printf("%d is a leap year\n",year);
11.            else
12.                printf("%d is not a leap year\n",year);
13.        else
14.            printf("%d is a leap year\n",year);
15.    else
16.        printf("%d is not a leap year\n",year);
17.    return 0;
18.}
```



# 选择结构程序综合举例：闰年

## Ø 方法一：嵌套if语句（添加一个标记变量）

```
1. #include <stdio.h>
2. int main()
3. {
4.     int year, leap=0;
5.     printf("enter year:");
6.     scanf("%d",&year);
7.     if (year%4==0)
8.         if (year%100==0)
9.             if (year%400==0)
10.                leap = 1;
11.            else
12.                leap = 0;
13.        else
14.            leap = 1;
15.    else
16.        leap = 0;
17.    if (leap)
18.        printf("%d is ",year);
19.    else
20.        printf("%d is not ",year);
21.    printf("a leap year\n");
22.    return 0;
23.}
```

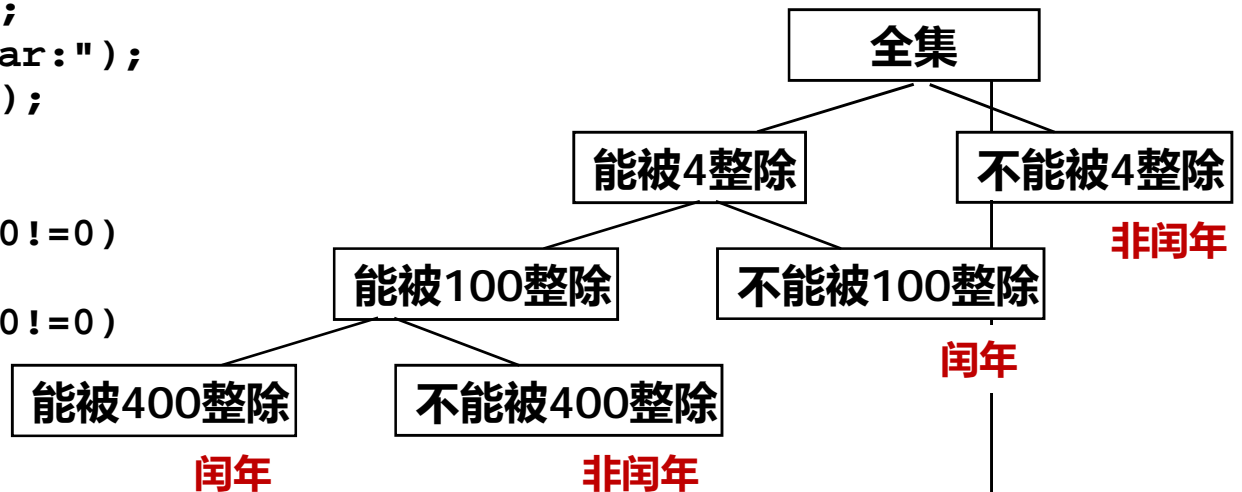




# 选择结构程序综合举例：闰年

## Ø 方法二：else if语句

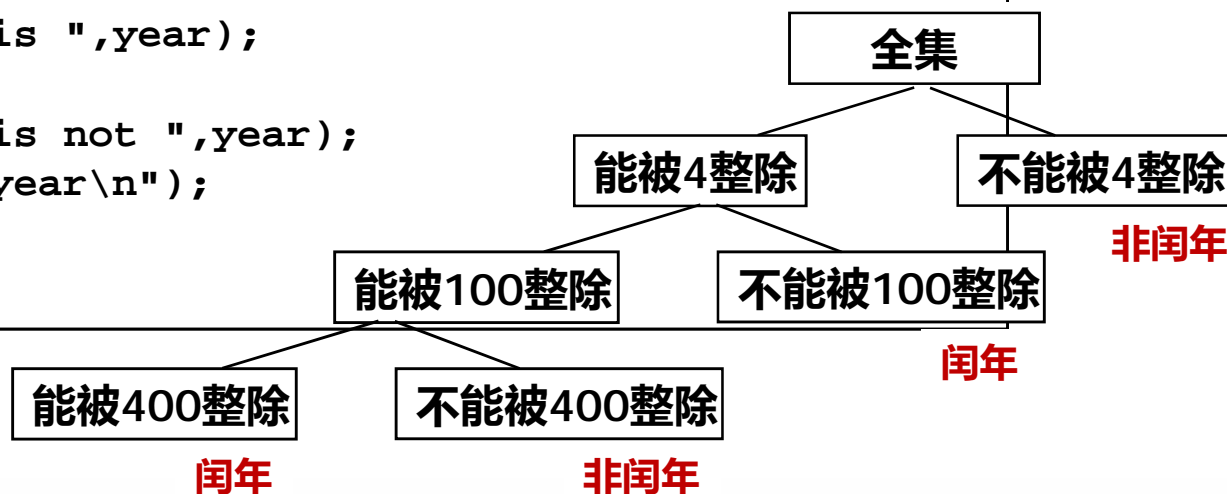
```
1. #include <stdio.h>
2. int main()
3. {
4.     int year, leap=0;
5.     printf("enter year:");
6.     scanf("%d",&year);
7.     if (year%4!=0)
8.         leap = 0;
9.     else if (year%100!=0)
10.        leap = 1;
11.    else if (year%400!=0)
12.        leap = 0;
13.    else
14.        leap = 1;
15.    if (leap)
16.        printf("%d is ",year);
17.    else
18.        printf("%d is not ",year);
19.    printf("a leap year\n");
20.    return 0;
21.}
```



# 选择结构程序综合举例：闰年

## Ø 方法三：逻辑表达式

```
1. #include <stdio.h>
2. int main()
3. {
4.     int year, leap=0;
5.     if ((year%4==0 && year%100!=0) || year%400==0)
6.         leap = 1;
7.     else
8.         leap = 0;
9.     if (leap)
10.        printf("%d is ",year);
11.    else
12.        printf("%d is not ",year);
13.    printf("a leap year\n");
14.    return 0;
15.}
```



# 选择结构程序综合举例：运输

Ø 运输公司对用户计算运输费用。路程( $s$  km) 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

# 选择结构程序综合举例：运输

## Ø 解题思路：

u 设每吨每千米货物的基本运费为 $p$ ，货物重为 $w$ ，距离为 $s$ ，折扣为 $d$ ，其中 $p, w, s$ 需输入

u 总运费 $f$ 的计算公式为 $f = p \times w \times s \times (1 - d)$

u 考虑switch结构，关键是找到 $d$ 和 $s$ 之间的关系

l 折扣的“变化点”都是250的倍数，设 $c$ 的值为 $s/250$

μ 当 $c < 1$ 时，表示 $s < 250$ ，无折扣

μ  $1 \leq c < 2$ 时，表示 $250 \leq s < 500$ ，折扣 $d = 2\%$

μ  $2 \leq c < 4$ 时， $d = 5\%$ ； $4 \leq c < 8$ 时， $d = 8\%$ ；

μ  $8 \leq c < 12$ 时， $d = 10\%$ ； $c \geq 12$ 时， $d = 15\%$

# 选择结构程序综合举例：运输

```
1. #include <stdio.h>
2. int main()
3. {
4.     int c,s;
5.     float p,w,d,f;
6.     printf("please enter price,weight,discount:");
7.     scanf("%f,%f,%d",&p,&w,&s);
8.     c = (s>=3000)?12:(s/250);
9.     switch (c)
10.    {
11.        case 0:    d=0; break;
12.        case 1:    d=2; break;
13.        case 2:
14.        case 3:    d=5; break;
15.        case 4:
16.        case 5:
17.        case 6:
18.        case 7:    d=8; break;
19.        case 8:    case 9:    case 10:
20.        case 11:    d=10; break;
21.        case 12:    d=15; break;
22.    }
23.    f = p * w * s * (1 - d / 100);
24.    printf("freight=%10.2f\n",f);
25.    return 0;
26. }
```

**第6行，在scanf语句前用printf语句输出必要的提示信息，可以使用户界面变得友好，但在上机课上请按照题目规定的输入格式设计程序，不要自行添加提示信息**

# 作业 2016/10/11

**Ø 按下列要求编写程序，提交手写源代码（下次课10月14日课间交）**

1. 输入一个整数，判断它是否为偶数。如果是，则输出“yes”，否则输出“no”。
2. 一件衣服95元，若消费满300元，可打八五折。输入购买衣服的件数，输出需要支付的金额（单位：元），保留两位小数。
3. 输入三角形三边长度值（均为正整数），判断它是否能为直角三角形的三边长。如果根本无法构成三角形，则输出“not a triangle”。否则如果可以构成直角三角形，则输出“yes”，否则输出“no”。

**Ø 上机练习（不用交）：编译运行本讲义例程，教材第四章9、10、11、12**