



清华大学

腾讯会议 ID: 676 940 633
Tsinghua University

计算机程序设计基础

第4讲 选择结构程序设计 与逻辑表达式

沈瑜 (010-62782951)
shenyu@tsinghua.edu.cn

清华大学电机系

2022.10.4



前3讲回顾

main函数 **return** 语句

printf、**scanf**；头文件

整型和实型数据；字符类型；
语句（复合语句）；

算术表达式、赋值表达式、逗号表达式；

继续学习：选择结构；循环结构；



赋值过程中的类型自动转换

- 如果赋值运算符两侧的类型一致，直接赋值

`i=234; // 此前i已经被定义为int型`

- 如果赋值运算符两侧的类型不一致，自动转换

- ◆ 将浮点数赋给整型变量时，**取整**，舍弃小数部分

`i = 3.45; // 结果是 i=3`

- ◆ 整型数赋给浮点数，数值不变，浮点数形式存储

`f = 23; // 此前f被定义为float或double，相当于f=23.0`

- ◆ 字符型数据赋给整型变量，赋给**ASCII**码

`i = 'A'; // 此前i已经被定义为int型`

- ◆ **截断**现象（给允许表值范围小的变量赋值时常见）

`c=289; // c 已被定义为char型，实际c=33`





主要内容

- 选择结构的C程序设计举例
- * 用流程图表示算法
- 逻辑表达式
- 选择语句
 - 两分支选择结构: `if`语句
 - 条件表达式 `?` :
 - 多分支选择结构: `switch`语句

参考教材: 第2章、第4章





4.1 选择结构的C程序设计举例

● 例1:成绩的等级

```
#include <stdio.h>

int main()
{
    int score;

    printf("你好! 你的《计算机程序设计基础》成绩预计得多少分? \n ");
    scanf( "%d", &score );    //输入成绩
    printf("你输入的预计成绩是%d分.\n", score);
    //将刚输出的数据打印出来, 以确认是否正确输入

    char grade; //以下7/8行, 是通过选择语句, 将成绩分数转化为等级。学习选择语句时再详细研究。
    if( score >= 93 && score <=100 )
        grade = 'A';
    else if( score >= 85 && score <=92 )
        grade = 'B';
    else if( score >= 77 && score <=84 )
        grade = 'C';
    else if( score >= 70 && score <=76 )
        grade = 'D';
    else if( score <= 69 )
        grade = 'F';

    printf("你的预计成绩等级为: %c \n", grade );

    return 0;
}
```

未熟练掌握debug技能之前, 请多printf重要数据

```
C:\windows\system32\cmd
你好! 你的《计算机程序设计基础》成绩预计得多少分?
99
你输入的预计成绩是99分.
你的预计成绩等级为: A
```

```
C:\windows\system32\cmd
你好! 你的《计算机程序设计基础》成绩预计得多少分?
B
你输入的预计成绩是-858993460分.
你的预计成绩等级为: F
```

如何改错?



● 例2: 二元一次方程求解（复根的处理）

```
int main ()
{
    double a, b, c, disc, x1, x2, p, q;
    // a=1.0, b=1, c = -30; //后续改为键盘输入
    printf("Please input a, b, c:");
    scanf("%lf%lf%lf", &a, &b, &c); //按照double型输入
    printf("You just set a=%f, b=%f, c=%f\n", a, b, c);
    //将刚输出的数据打印出来, 以确认是否正确输入
    disc=b*b-4*a*c;
    p=-b/(2.0*a);
    if(disc>=0) //如果有两个实根
    {
        q=sqrt(disc)/(2.0*a);
        x1=p+q; x2=p-q;
        printf("We can find the solution:\n");
        printf("x1=%7.2f\nx2=%7.2f\n", x1, x2); //输出两个实根
    }
    else //如果有两个复根
    {
        q=sqrt(-disc)/(2.0*a);
        printf("We can find the solution:\n");
        printf("x1=%7.2f+%7.2fj\nx2=%7.2f-%7.2fj\n", p, q, p, q);
        //输出共轭复根
    }
    return 0;
}
```

未熟练掌握debug技能之前, 请多printf重要数据

实根的处理语句块

复根的处理语句块



● 例3: 从数字月份, 转换到繁体中文月份

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int nMonth;
```

```
    printf("请输入月份 (1-12): \n");
```

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

```
    switch(nMonth)
```

```
    {
```

```
    case 1:
```

```
        printf("壹月");
```

```
        break;
```

```
    case 2:
```

```
        printf("贰月");
```

```
        break;
```

```
    case 3:
```

```
        printf("叁月");
```

```
        break;
```

```
    case 4:
```

```
        printf("肆月");
```

```
        break;
```

```
    case 5:
```

```
        printf("伍月");
```

```
        break;
```

```
    case 6:
```

```
        printf("陆月");
```

```
        break;
```

```
    case 7:
```

```
        printf("柒月");
```

```
        break;
```

```
    case 8:
```

```
        printf("捌月");
```

```
        break;
```

```
    case 9:
```

```
        printf("玖月");
```

```
        break;
```

```
    case 10:
```

```
        printf("拾月");
```

```
        break;
```

```
    case 11:
```

```
        printf("拾壹月");
```

```
        break;
```

```
    case 12:
```

```
        printf("拾贰月");
```

```
        break;
```

```
    default:
```

```
        printf("我不懂是几月啊? !");
```

```
        return -1;
```

```
    }
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```

请输入月份 (1-12):
9
玖月

请输入月份 (1-12):
0
我不懂是几月啊? !





● 例4: Yes/No答问

```
#include <stdio.h>

int main()
{
    char cInput;

    printf("张三, 你的作业交了吗? 【交了的话请输入Y或y, 没交的话请输入N或n】 \n");
    scanf("%c", &cInput );

    if( cInput == 'Y' || cInput == 'y' )
        printf("谢谢! 经确认: 张三同学的作业已经交了! \n");
    else
        if( cInput == 'N' || cInput == 'n' )
            printf("遗憾啊! 经确认: 张三同学的作业还没交呢! \n");
        else
            printf("抱歉! 我不懂你说什么耶? ! \n");

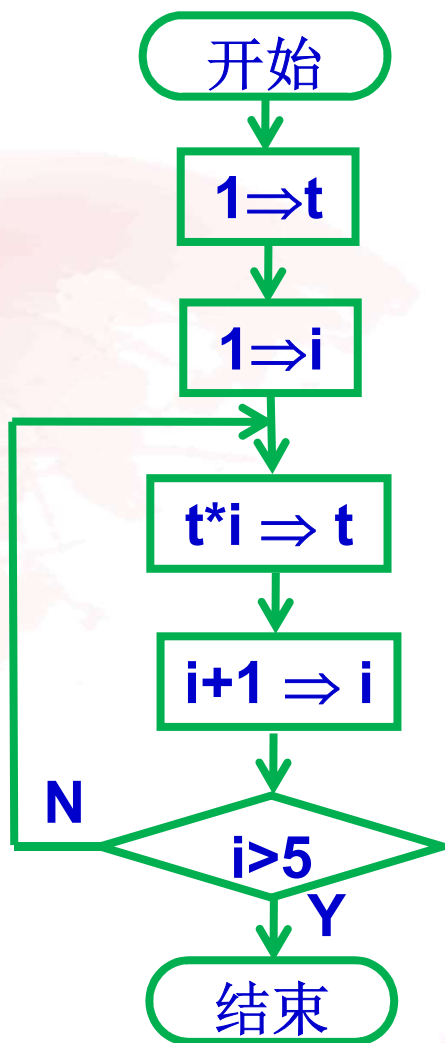
    return 0;
}
```




4.2 用流程图表示算法

1. 读懂流程图（示例）

例1：求 $1 \times 2 \times 3 \times 4 \times 5$
(教材p17, 例2.1)



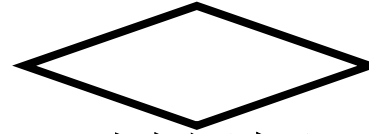
流程图基本符号



起止框



输入输出框



判断框



处理框



流程线



连接点



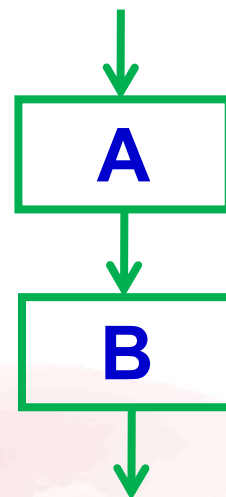
注释框

意大利面条现象
a bowl of spaghetti

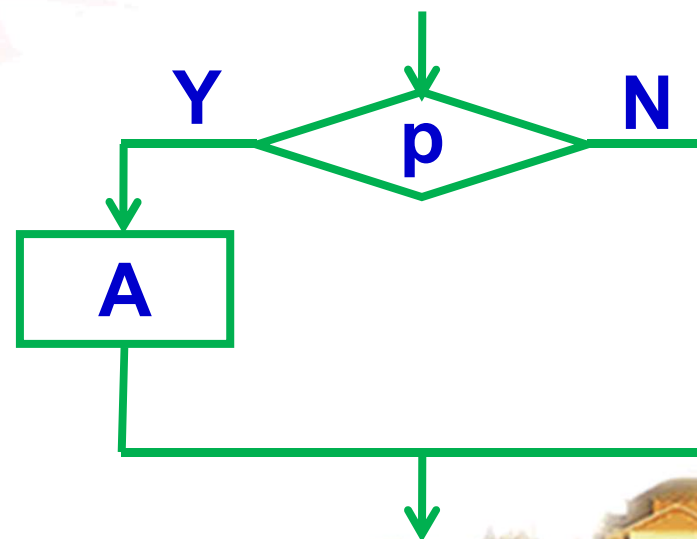
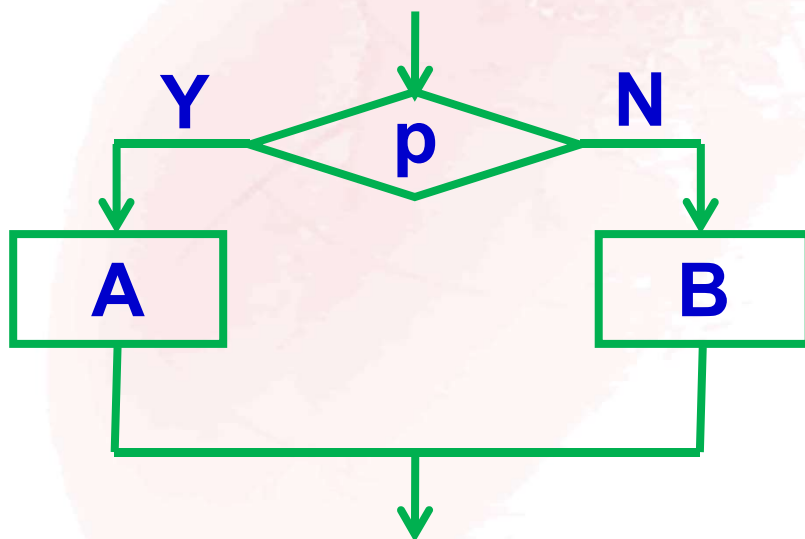


三种基本结构

(1) 顺序结构



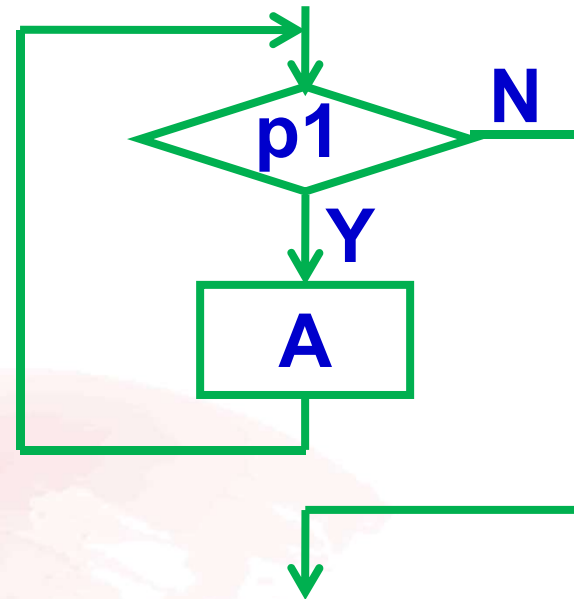
(2) 选择结构



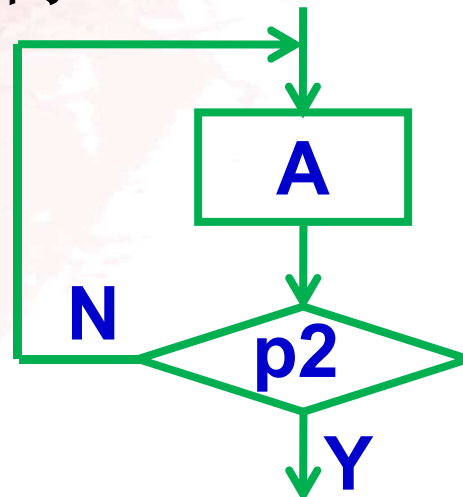
三种基本结构

(3) 循环结构

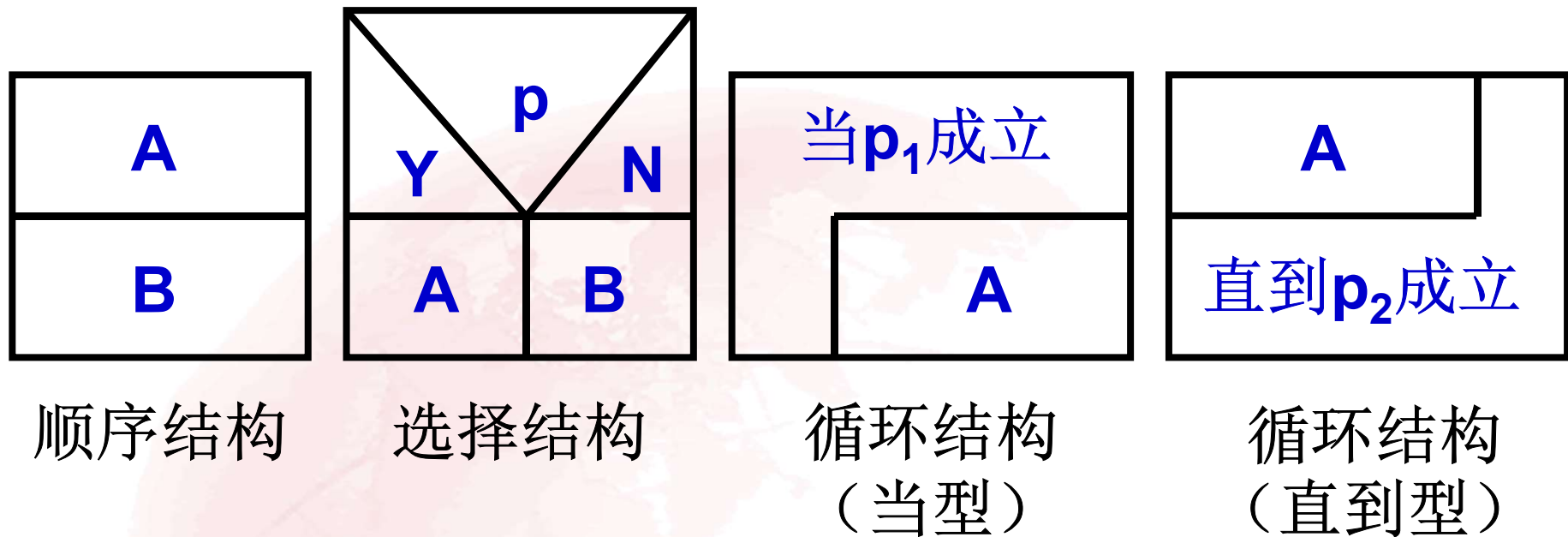
① 当型循环结构



② 直到型循环结构

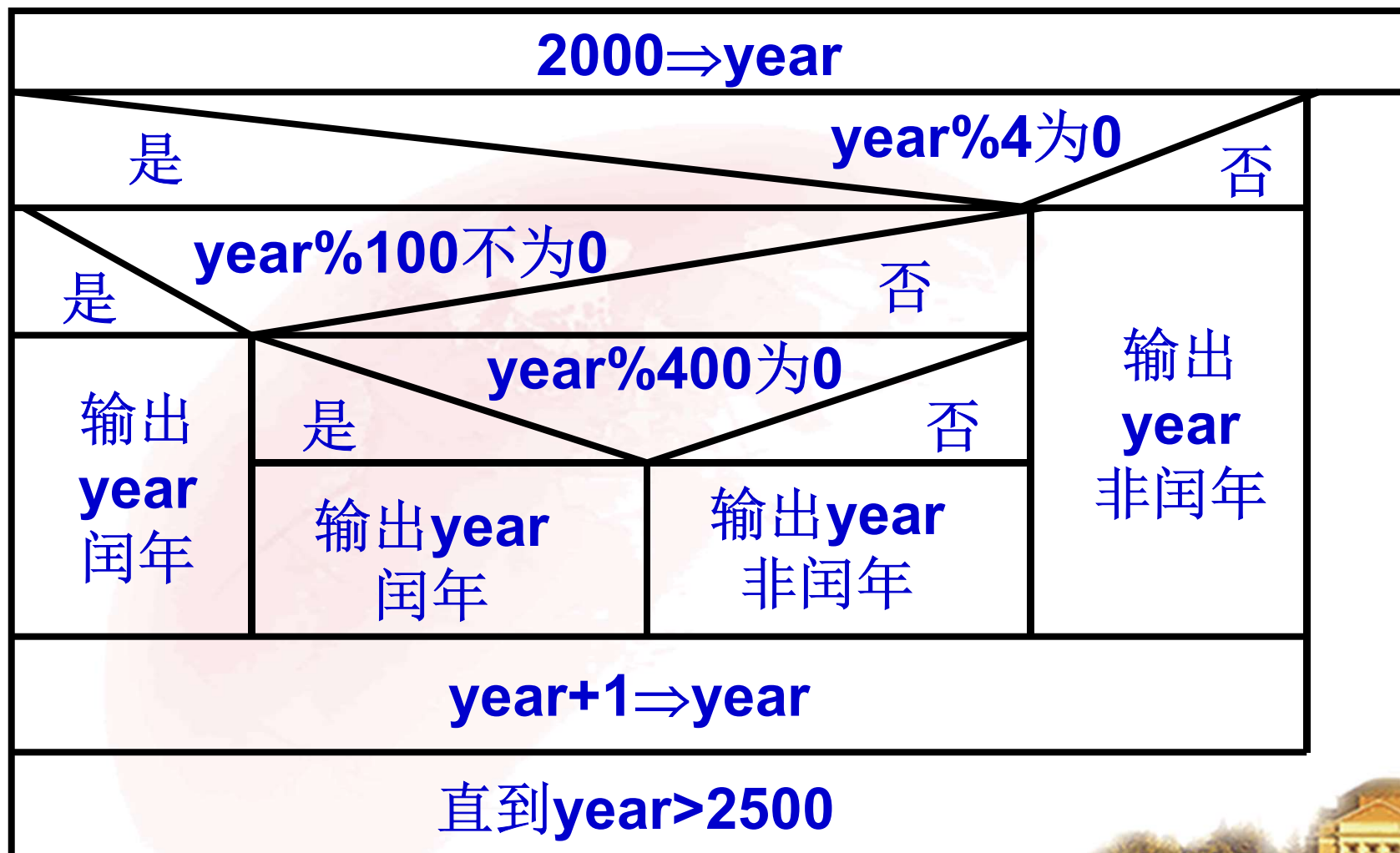


2. N-S结构化流程图（盒图）



读懂N-S流程图（示例）

例：从2000到2500年，输出所有闰年（教材p19，例2.3）





4.3 逻辑表达式

1. 二值逻辑的真值

逻辑运算的结果:

– 假: 0 **false** 真: 1 **true**

布尔类型/逻辑类型变量

```
bool b1 = false;
```

```
b1=true;
```

非0, 为真

例

注: **bool, false, true**在C语言不支持, **C++**中支持

标准C语言中可这么写: **int b1=0; b1=1;**



2. 关系运算符及其优先级

- 关系运算符

对两个数值进行比较的关系运算符，结果为逻辑值

- C 语言提供 6 种关系运算符：

① $<$ (小于) ② \leq (小于或等于)

③ $>$ (大于) ④ \geq (大于或等于)

例

优先级相同 (高)

(判等运算符)

⑤ $==$ (等于) ⑥ $!=$ (不等于)


优先级相同 (低)

注意区别：赋值 $=$ 与 判等 $==$



- 关系、算术、赋值运算符的优先级

算术运算符 (高)
关系运算符
判等运算符
赋值运算符 (低)



$c \geq a + b$ 等效于 $c \geq (a + b)$

$a \geq b == c$ 等效于 $(a \geq b) == c$

$a == b \leq c$ 等效于 $a == (b \leq c)$

$a = b \geq c$ 等效于 $a = (b \geq c)$



3. 逻辑运算符及其优先级

- 对逻辑量（含关系表达式结果）进行运算
- 3种逻辑运算符：
 - &&**（逻辑与） **||**（逻辑或） **!**（逻辑非）
- **&&**和**||**是双目(元)运算符
- **!**是一目(元)运算符

➤ 判断年龄在**13**至**17**岁之内？

age>=**13** **&&** **age**<=**17**

➤ 判断年龄小于**12**或大于**65**？

age<**12** **||** **age**>**65**

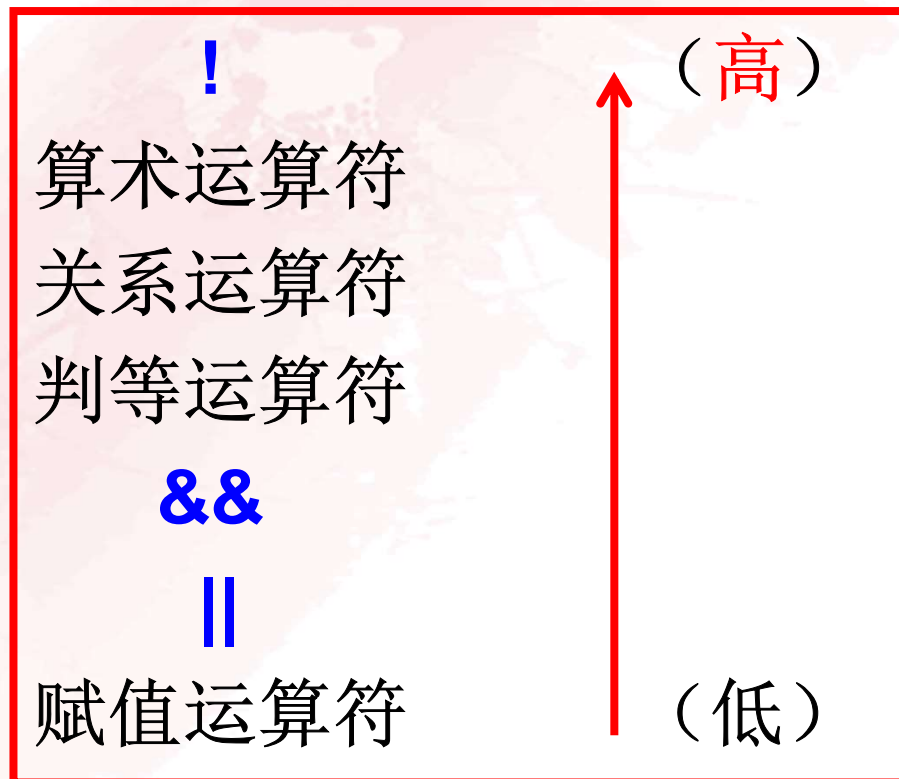


逻辑运算的优先级

➤逻辑运算符的优先次序

! → **&&** → **||** (**!**为三者中最高)

➤与其他运算符的优先次序



逻辑表达式

- 逻辑表达式的值应该是逻辑量“真”或“假”
- 编译系统在表示逻辑运算结果时
 - 以数值**1**代表“真”，以**0**代表“假”
- 自左向右扫描判断逻辑表达式
- 但在判断一个量是否为“真”时
 - 视 **0** 为“假”，视 **非0** 为“真”
- 注意：将一个非零的数值视为“真”




逻辑表达式举例

- (1) 若 $a=4$ ，则 $!a$ 的值为0
- (2) 若 $a=4$ ， $b=5$ ，则 $a \&\& b$ 的值为1
- (3) a 和 b 值分别为4和5，则 $a||b$ 的值为1
- (4) a 和 b 值分别为4和5，则 $!a||b$ 的值为1
- (5) $4 \&\& 0 || 2$ 的值为1




逻辑短路

➤一旦能够明确无误地确定整个表达式的值，就不再计算表达式余下部分



对“&&”运算符,如果前一操作数为假,则不再理会后一操作数;



对“||”运算符,如果前一操作数为真,则不再理会后一操作数;

有 `int a=0, b=0;`

`if((a=4) || (b=0))`

`if((a==4) || (b==0))`

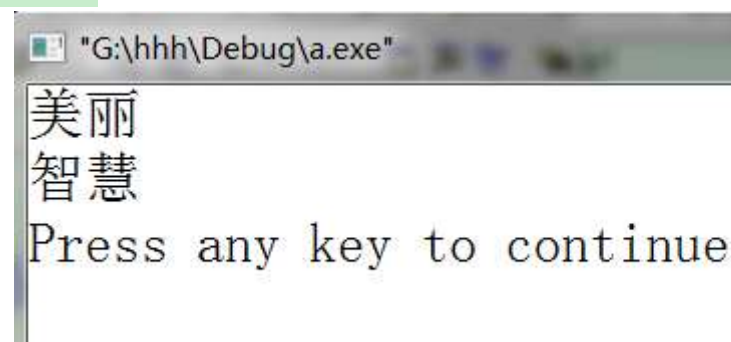
逻辑短路



`if((b==0) || (a=4))`

逻辑表达式举例

```
void main() {  
    int a = 4, b = 5;  
    if ( !a && (a=0)>b) printf("真诚\n");  
    if ( !a || (a=0)>b)   printf("善良\n");  
    if ( !a && a<b) printf("美丽\n");  
    if ( !a || a<b)   printf("智慧\n");  
}
```



逻辑表达式举例

a=4, b=5, 则以下表达式为真的是: (多选题)

A

!a

B

a&&b

C

a||b

D

!a||b

E

a&&0 || b

提交



4.3 if语句

1. if语句的一般形式

if (表达式) 语句1
[else 语句2]

关系表达式
逻辑表达式
数值表达式

方括号内的部分是可选的



if语句的一般形式

if (表达式) 语句1

[else 语句2]

简单的语句
复合语句
另一个if语句等



2. 最常用的3种if语句形式

1) if (表达式) 语句1 (没有else子句)

2) if (表达式) 语句1

else 语句2 (有else子句)

3) 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

(在else部分又嵌套了多层的if语句)

级联式if书写形式



```
if( score >= 93 && score <=100 )  
    grade = 'A';  
else if( score >= 85 && score <=92 )  
    grade = 'B';  
else if( score >= 77 && score <=84 )  
    grade = 'C';  
else if( score >= 70 && score <=76 )  
    grade = 'D';  
else if( score <= 69 )  
    grade = 'F';
```

➤说明:

- (1)整个if语句可写在多行上，也可写在一行上
但都是一个整体，属于同一个语句
- (2)“语句1”...“语句m”是if中的内嵌语句
内嵌语句也可以是一个if语句
- (3)“语句1”...“语句m”可以是简单的语句，也
可以是复合语句



级联式if书写形式

```
if(score >= 93 && score <=100) grade = 'A';  
else if (score >= 85 && score <=92 ) grade = 'B';  
else if (score >= 77 && score <=84 ) grade = 'C';  
else if (score >= 70 && score <=76 ) grade = 'D';  
else if (score <=69 ) grade = 'F';
```

```
if(score >= 93 && score <=100) grade = 'A';  
else  
    if (score >= 85 && score <=92 ) grade = 'B';  
    else  
        if (score >= 70 && score <=76 ) grade = 'D';  
        else if (score <=69 ) grade = 'F';
```



分号不能丢

3. 选择结构的嵌套

- 在**if**语句中又包含一个或多个**if**语句称为**if**语句的嵌套
- 一般形式:

```
if ( )  
    if ( ) 语句1  
    else 语句2  
else  
    if ( ) 语句3  
    else 语句4
```

else总是与它上面最近的未配对的**if**配对

内嵌**if**



3. 选择结构的嵌套

- 在if语句中又包含一个或多个if语句称为if语句的嵌套

if ()

{

if () 语句1

}

else 语句2

内嵌if

{ }限定了内嵌if范围





4.4 条件运算符和表达式

1. 条件运算符

- 有一种**if**语句，当被判别的表达式的值为“真”或“假”时，都执行一个赋值语句且向同一个变量赋值

➤ 如：

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

条件运算符

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

条件表达式

➤ 条件表达式的一般形式为

表达式 1 ? 表达式 2 : 表达式 3

➤ 条件运算符的执行顺序：

◆ 求解表达式1

◆ 若为非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

◆ 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



➤ 以下为合法的使用方法:

◆ `a>b ? (max=a):(max=b);`

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

➤ 条件运算符优先于赋值运算符

➤ 条件运算符的结合方向为“自右至左”

`(x>y? printf("1"):(c==0)?printf("2"):printf("3"))`



➤ 例：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。

```
#include <stdio.h>
```

```
int main()
```

```
{ char ch;
```

```
scanf("%c",&ch);
```

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

```
printf("%c\n",ch);
```

```
return 0;
```

```
}
```





4.6 switch实现多分支选择结构

1. 条件运算符

```
switch (表达式) {  
    case 常量1:  
        statements  
        break;  
    case 常量2:  
        statements  
        break;  
    default:  
        statements  
}
```

switch语句能做到的，**if**语句都能做到。但当分支太多时，用**switch**来写，程序会更简洁！



case子语句

```
switch (表达式) {  
    case 常量1:  
        statements ;  
        break;  
    case 常量2:  
        statements ;  
        break;  
    default:  
        statements ;  
}
```

1) 表达式的值与此常量表达式相等时，就从此case顺序执行

2) 常量类型任意。一般是整型或字符型

3) statements可以是单个语句或多个语句

4) 碰到“break;”跳出此分支结构，switch语句执行结束

5) 若此处break语句漏写，则程序继续往下执行

case仅仅是语句标号，开始执行的入口



●例3: 从数字月份, 转换到繁体中文月份

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int nMonth;
```

```
    printf("请输入月份 (1-12) : \n");
```

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

```
    switch(nMonth)
```

```
    {
```

```
    case 1:
```

```
        printf("壹月");
```

```
        break;
```

```
    case 2:
```

```
        printf("貳月");
```

```
        break;
```

```
    case 3:
```

```
        printf("叁月");
```

```
        break;
```

```
    case 4:
```

```
        printf("肆月");
```

```
        break;
```

```
    case 5:
```

```
        printf("伍月");
```

```
        break;
```

```
    case 6:
```

```
        printf("陆月");
```

```
        break;
```

```
    case 7:
```

```
        printf("柒月");
```

```
        break;
```

```
    case 8:
```

```
        printf("捌月");
```

```
        break;
```

```
    case 9:
```

```
        printf("玖月");
```

```
        break;
```

```
    case 10:
```

```
        printf("拾月");
```

```
        break;
```

```
    case 11:
```

```
        printf("拾壹月");
```

```
        break;
```

```
    case 12:
```

```
        printf("拾贰月");
```

```
        break;
```

```
    default:
```

```
        printf("我不懂是几月啊? !");
```

```
        return -1;
```

```
    }
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```

请输入月份 (1-12) :
9
玖月

请输入月份 (1-12) :
0
我不懂是几月啊? !

```
x=0;y=0;c=0;
```

```
(x>y? printf("1") : (c==0)?printf("2"):printf("3"))
```

输出结果是?

☐ A 1

☒ B 2

☐ C 3

提交

