

第五章

选择语句

● 本章要点

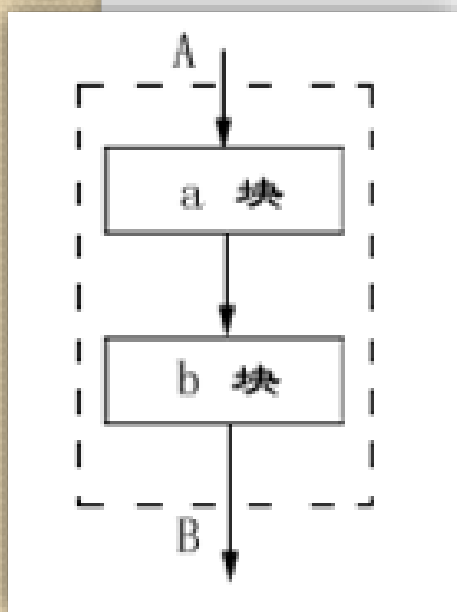
- 逻辑表达式
- if 语句
- switch 语句

C语言语句

- 语句是程序的基础，目前见过：
 - **return语句、表达式语句，以及函数调用语句**
- 相比运算符，C语言拥有的语句较少，主要分为：
 - **1.控制语句：控制程序流程**
 - **2.函数调用语句**
 - **3.表达式语句**
 - **4.其他语句：**
 - 复合语句：把几条语句组合成一条语句。
 - 空语句：不执行任何操作。

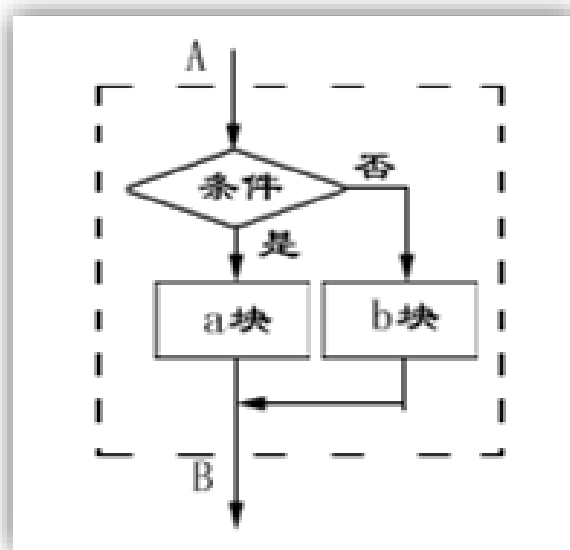
程序流程

- 程序解决问题的思路、或步骤



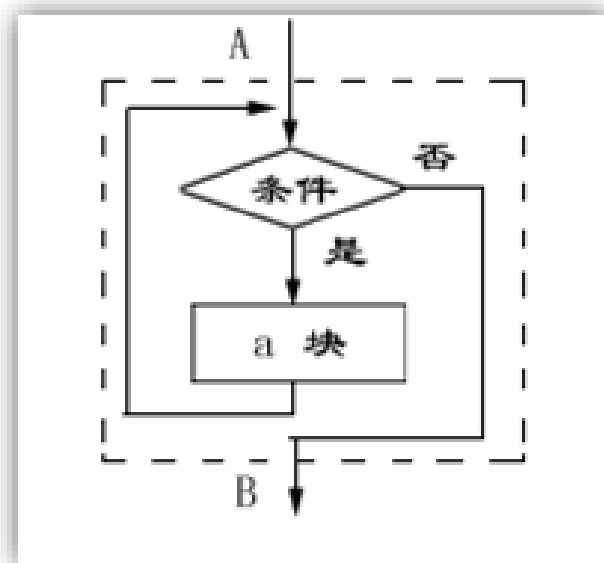
顺序结构

走直路
eg. 打饭, 打菜



选择结构

岔路
eg. 男女生打饭



循环结构

回头路
eg. 排队打饭

(流程) 控制语句

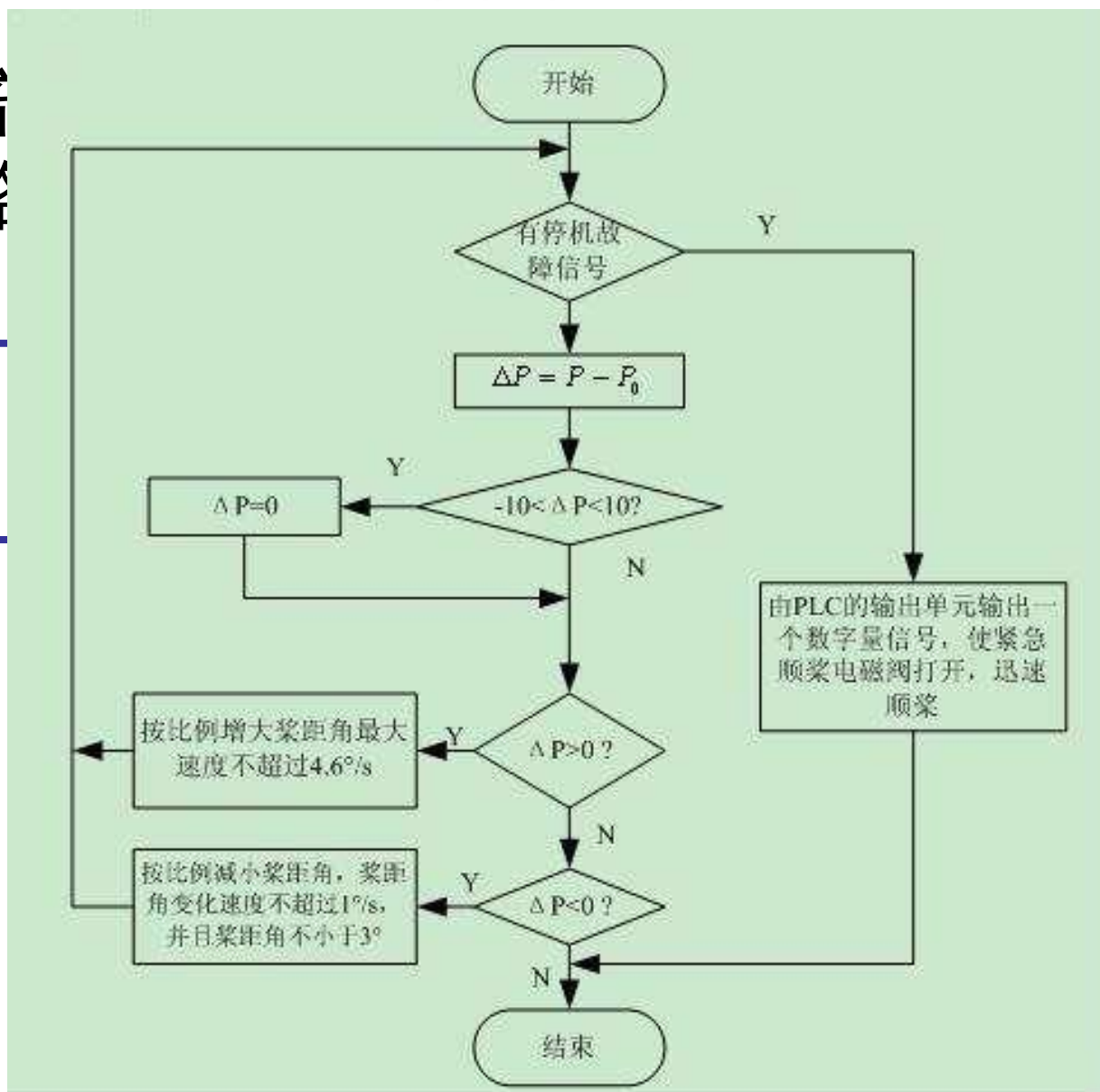
● 流程

○

○

○

○



序的骨

o语句,

控制语句如何控制流程？

- 条件判断
 - if（条件判断表达式），eg. 想吃面，身高低于120cm，高富帅
 - while（条件判断表达式），eg. 有人排队，试卷统分完
- 条件判断表达式——逻辑表达式

逻辑表达式用于设置流程控制条件
控制语句+逻辑表达式+要执行操作

5.1 逻辑表达式

- 逻辑表达式：描述一种判断
 - eg. 清水河校区是一座寺庙
 - 结果：正确与否
 - 判断题“勾叉”——对错
- 值为“真(true)”或“假(false)”
 - 真：确实是寺庙
 - 假：不是寺庙
- 许多编程语言具有“布尔”或“逻辑”类型
 - 只有两值：真、假。
- C语言：
 - 0：假
 - 1：真

构造逻辑表达式

- 表达式：操作数+运算符
- 所需运算符：
 - 关系运算符
 - 判等运算符
 - 逻辑运算符
- Eg, 清水河是寺庙
 - **Campus_qing==temple**

5.1.1 关系运算符

- C的关系运算符如下：
 - **<** 小于
 - **>** 大于
 - **<=** 小于等于
 - **>=** 大于等于
- 关系运算符应用于表达式构成逻辑表达式：
 - **Eg. $i \leq j + 3$; height < 120 ; age > 60 ;**
 - **结果： 0（假）或 1（真）。**
- 可用于比较整数和浮点数，也可比较混合类型操作数。
 - **$3 < PI$**

关系运算符

算术运算符	(高)
关系运算符	↑
赋值运算符	(低)

- $v = i + j < k - 1$
- 等价于：
 - $v = ((i + j) < (k - 1))$
- 左结合（二元）
 - $i < j < k$ 等价于： $(i < j) < k$,
 - $i < j$ 的结果（1或0）再与 k 比较大小。
 - 与数学中意义不同（ j 介于 i 与 k 之间）
- 如果想检测 j 的值是否在 i 和 k 之间：
 - $i < j \ \&\& \ j < k$ （复杂逻辑表达式）

5.1.2 判等运算符

- C的判等运算符如下：
 - **==** 等于
 - **!=** 不等于
- 左结合，结果0（假）或 1（真）。
- 优先级低于关系运算符。
 - 表达式 **$i < j == j < k$** 相当于
 - **$(i < j) == (j < k)$**
- 优先级：高于赋值运算符
 - **$x = i == j$**

5.1.3 逻辑运算符

- 用于构造复杂逻辑表达式（复杂条件）
 - **eg. 儿童、老人免票：**
 - $\text{year} < 12$ 或 $\text{year} > 60$
 - **eg. 高富帅：**
 - $h > h_{\text{潘}}$ 且 $lk > lk_{\text{赵}}$ 且 $\text{vehicle} \geq 1$ 且 $\text{dorm} \geq 1$
- C逻辑运算符如下：

◦ &&	逻辑与(且)	二元	左结合性
◦ 	逻辑或	二元	左结合性
◦ !	逻辑非	一元	右结合性

5.1.3 逻辑运算符

- 以简单逻辑表达式（值0或1）为操作数，运算结果为：0（假）或 1（真）
- 儿童、老人免票
 - `year < 12 || year > 60`
- 高富帅
 - `h > h潘 && lk > lk赵 && vehicle >= 1 && dorm >= 1`
- 非0的操作数当作真值运算
 - `动物园ticket = 10;`
 - 入园检票：`ticket || year < 12 || year > 60`
 - `if(ticket || year < 12 || year > 60) pass;`

流程控制示例——非诚勿扰

- 如果高富帅，则留灯，否则灭灯
- if (高并且富并且帅) 留灯, else 灭灯
 if($h > h_0 \ \&\& \ p > p_0 \ \&\& \ k > k_0$)
 keep light on
 else
 trun off the light

逻辑运算符优先级次序

优先级	名称	符号	结合性
1	(后缀)自增、自减	++ --	左结合
2	(前缀)自增、自减	++ --	右结合
	一元正号、负号	+ -	
	一元逻辑运算符	!	
3	乘法类	* / %	左结合
4	加法类	+ -	左结合
5	关系运算符	> < >= <=	左结合
6	判等运算符	== !=	左结合
7	二元逻辑运算符	&&	左结合
8	赋值	= *= /= %= += -=	右结合

逻辑运算符优先级次序

$c = d > a + b \ \&\& \ a > b$

- 等效于：

$c = ((d > (a + b)) \ \&\& \ (a > b))$

逻辑运算符的运算结果

a		b	!a	!b	a&&b	a b
非0 (1)		非0 (8)	0	0	1	1
非0		0	0	1	0	1
0		非0	1	0	0	1
0		0	1	1	0	0

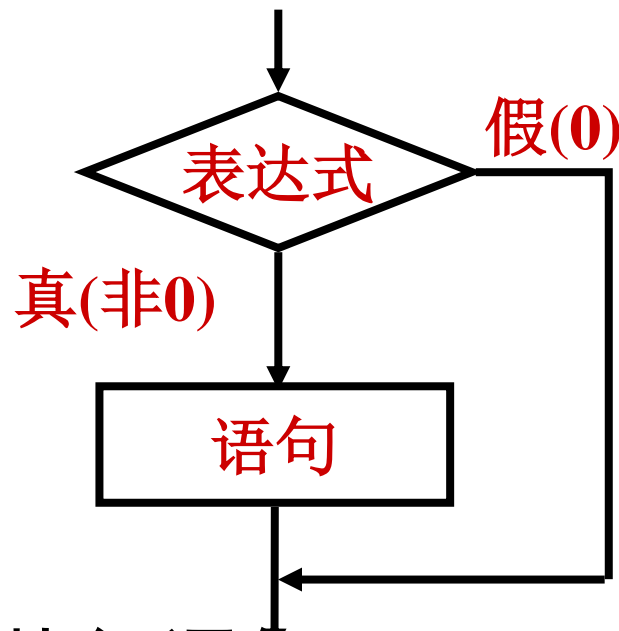
短路计算

- 对**&&**、**||**（左结合）
 - 如果根据左操作数就能推出表达式结果，就不计算右操作数。
- 例子1：
考试双证：准考证**&&**学生证
int b, x = 3, y = 5;
b = x > y && x++ == y--;
- 结果：b=0, x = 3, y = 5
- 例子2：
int b, x = 9, y = 5;
b = x > y || x++ == y--;
 - 结果：b=1, x = 9, y = 5

短路副作用：影响其他运算符副作用

5.2 if语句

- if 语句最简单格式：
 - **if (表达式) 语句**
 - 表达：如果.....就.....
 - 三八节，如果（女性）放假
- 计算（表达式）的值，非零执行语句，Eg
 - **if (line_num == MAX_LINES) //注意==而非=**
 - **line_num = 0;**
- 判定变量是否在某数值范围内，如[0 , n)
 - **if (i >= 0 && i < n) ...//wrong 0<=i<n**
- 相反情况判定：
 - **if (i < 0 || i >= n) ...**



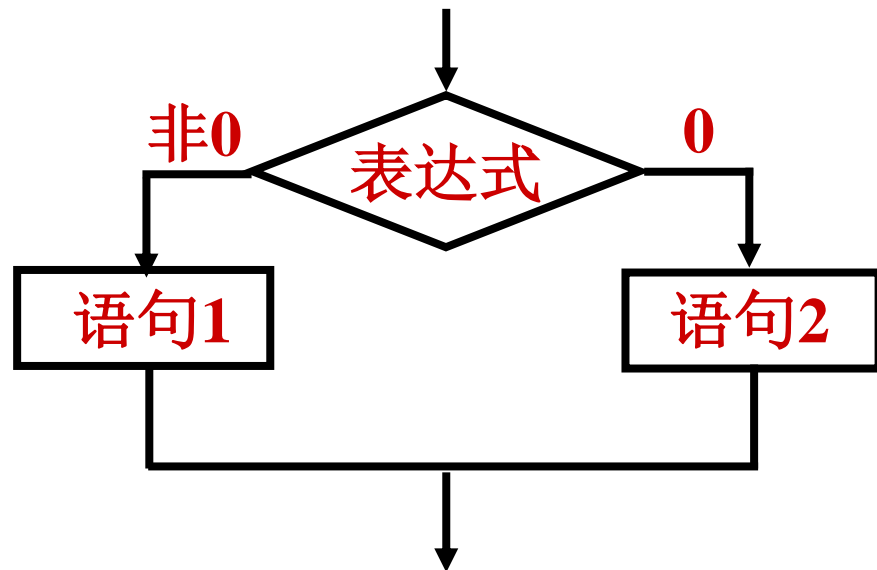
5.2.1 复合语句(compound statement)

- 复合语句：把多条语句组合到一起，形成单条语句。
 - 完成一个任务必须执行多条语句，要么都执行要么都不执行
- if语句处理复杂功能（多条语句），必须构造复合语句，格式：
 {多条语句}
- 例子：
 if (line_num == MAX_LINES)
 {
 line_num = 0;
 page_num++;
 }
 - 注意：{}外没有“;”
- 也出现在循环和其他需要多条语句的地方

5.2.2 else子句

- 格式：
if (表达式)
 语句1
else
 语句2
表达：如果……就……
 否则……

- 例子：
if (i > j)
 max = i;
else
 max = j;



注意缩进对齐

程序练习3——标准体重计算

根据身高计算标准体重

标准体重（男） $= (\text{身高cm} - 100) \times 0.9 (\text{kg})$

标准体重（女） $= (\text{身高cm} - 100) \times 0.9 (\text{kg}) - 2.5 (\text{kg})$

- 男女有别：

男：m (M) ， 女：f (F)

男：1， 女：0

`scanf("%c".....) ;`

程序练习——标准体重计算

```
/*计算标准体重*/
```

```
#包含 stdio.h
```

```
int main()
```

```
{
```

```
    声明变量;          char sex;//'m'或 'f'
```

```
    提示输入性别;
```

```
    读入性别;
```

```
    提示输入身高;
```

```
    读入身高;
```

```
    如果是男性
```

```
        标准体重 = (身高-100) * 0.9f;
```

```
    否则//女性
```

```
        标准体重 = (身高-100) * 0.9f - 2.5;
```

```
    显示标准体重;
```

```
    程序返回;
```

```
}
```


程序练习——标准体重计算

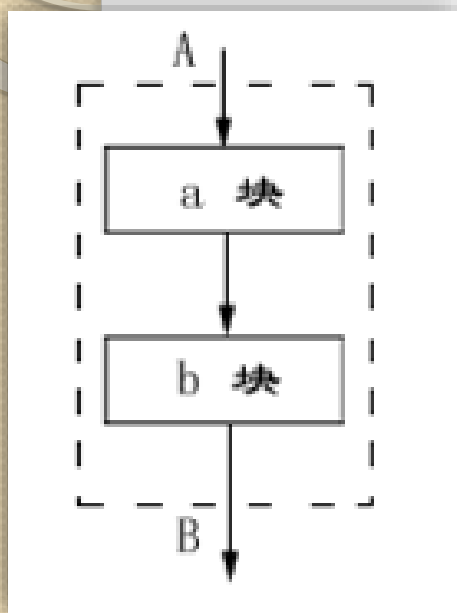
```
#include <stdio.h>
#define FACTOR 0.9f
int main(void){
    int height;
    float weight, stdwt;
    char sex;
```

- **程序改进：**
 - **输入身高、体重，计算与标准体重的差值，并提示**

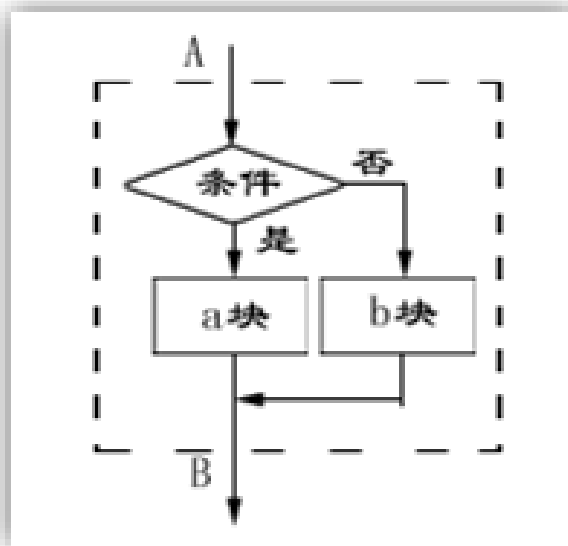
```
    if (sex == 'm')
        stdwt = (height - 100) * FACTOR;
    else
        stdwt = (height - 100) * FACTOR - 2.5;
    printf("你的标准体重应是%.1fkg\n.", stdwt);
    return 0;
}
```


温故而知新——程序流程

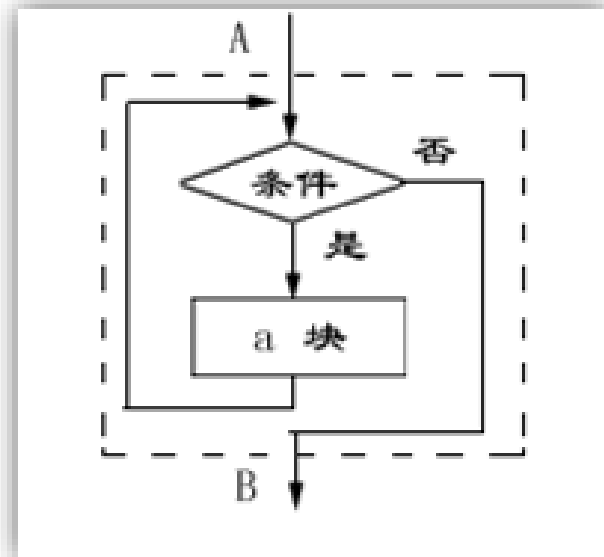
- 程序解决问题的思路、或步骤



顺序结构



选择结构



循环结构

如何控制程序流程？

1. 条件：逻辑表达式

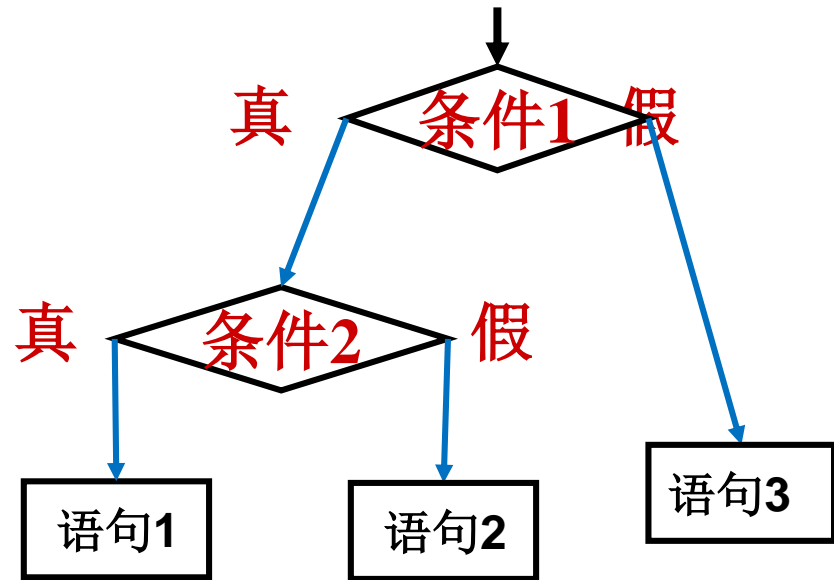
2. 流程控制语句：控制语句（逻辑表达式）操作；

温故而知新——逻辑表达式

- 逻辑运算符将关系表达式或逻辑量连接起来的有意义的式子。
- 运算符
 - 关系运算符
 - $<$, $>$, $<=$, $>=$
 - 判等运算符
 - $==$, $!=$
 - 逻辑运算符
 - $\&\&$ （并且）， \parallel （或者）， $!$ （非）
- 结果
 - 假：0
 - 真：1，非0

if语句的嵌套

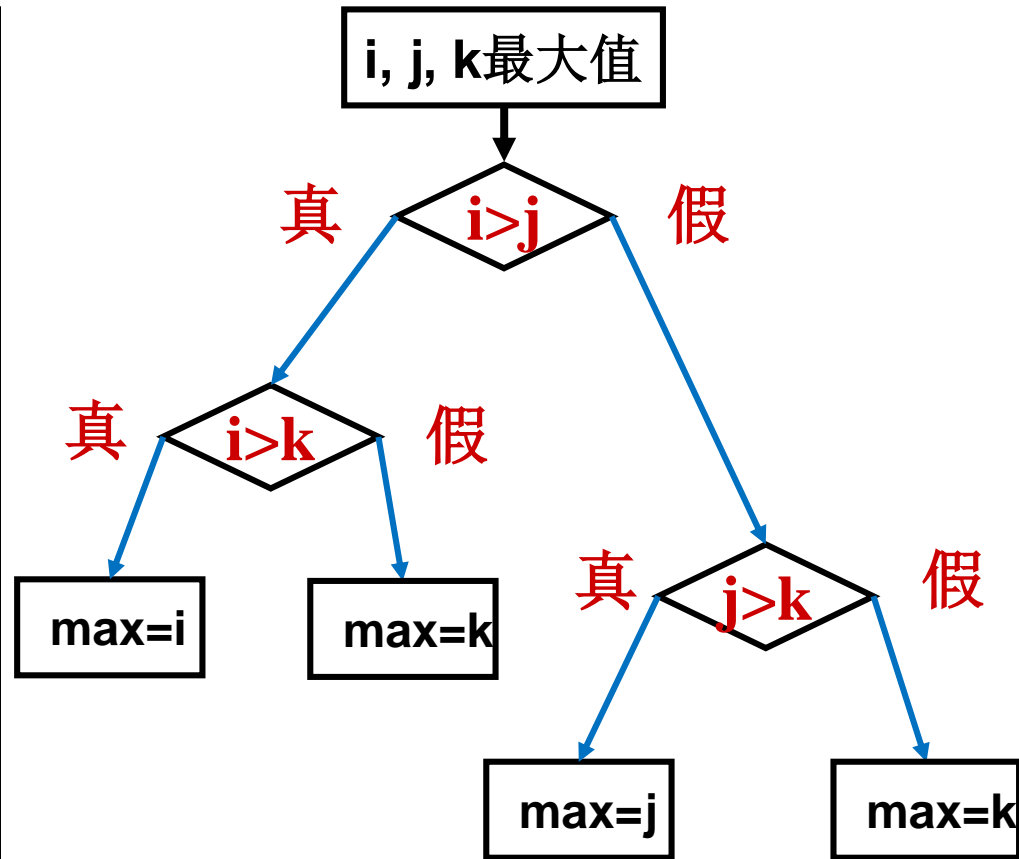
- 前面的if
 - 一个条件：两个分支（选择）
- 现实中可能多种（层）选择
- 嵌套的选择eg.
 - 动物园（age≤12）
 - 儿童
 - height≤150 free
 - height>150 half
 - 成人
 - 老年
 - 非老年
 - 持证



if语句的嵌套

- 求 i、j 和 k 中的最大值

```
if (i > j)
    if (i > k)
        max = i;
    else
        max = k;
else // i <= j
    if (j > k)
        max = j;
    else
        max = k;
```



if语句的嵌套

- 求i、j和k中最大值

```
if (i > j)
    if (i > k)
        max = i;
    else
        max = k;
else
    if (j > k)
        max = j;
    else
        max = k;
```

- 增加花括号，增强可读性

```
if (i > j)
{
    if (i > k)
        max = i;
    else
        max = k;
}
else
{
    if (j > k)
        max = j;
    else
        max = k;
}
```

5.2.3 级联式if语句

格式：

if (表达式1)语句1

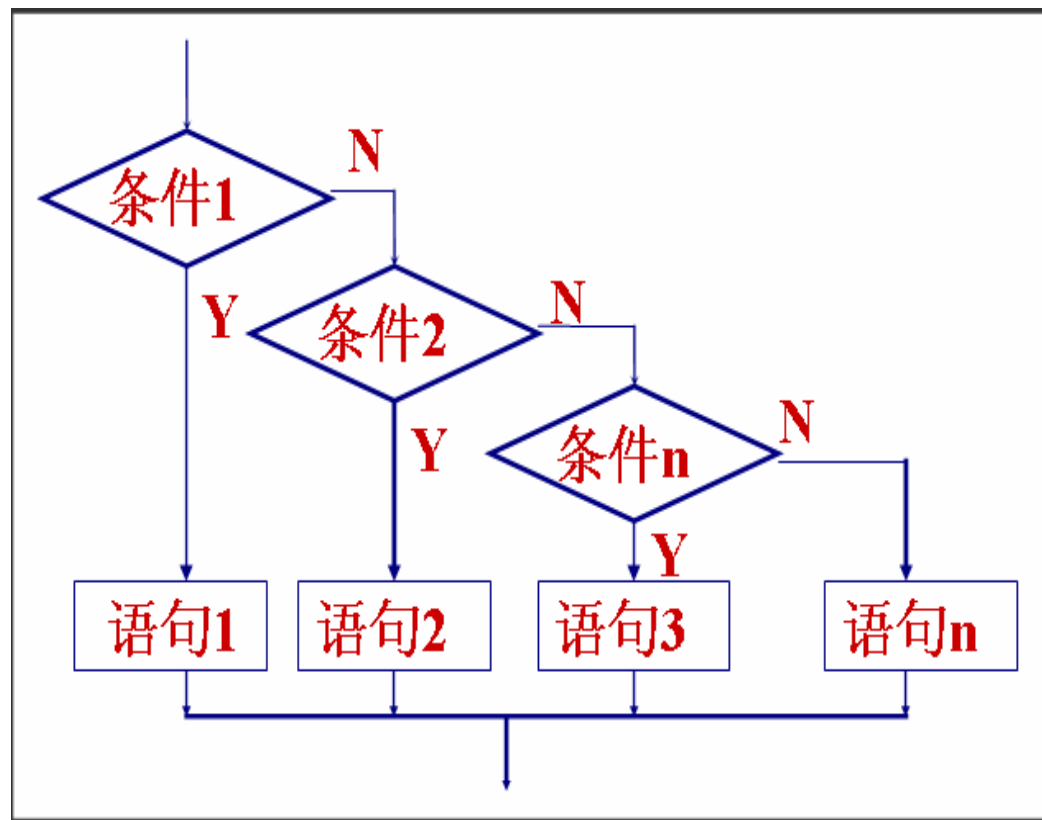
else if(表达式2)语句2

else if(表达式3)语句3

.....

else if(表达式m)语句m

else 语句n



水晶鞋选灰姑娘

级联式if语句示例

- 判数（数大于0，等于0，小于0）

```
if (n < 0)
    printf("n is less than 0\n");
else if (n == 0)
    printf("n is equal to 0\n");
else
    printf("n is greater than 0\n");
```

- 动物园售票程序：

```
if(age<12)
    printf("free for children ");
else if(y>60)
    printf("free for old people ");
else if(soldier_cert)
    printf("free for soldier ");
else
    ticket = 10;
```

例子：计算股票经纪人的佣金

- 股票经纪人的佣金计算表：

交易额范围	佣金费用
低于\$2,500	\$30 + 1.7%
\$2,500–\$6,250	\$56 + 0.66%
\$6,250–\$20,000	\$76 + 0.34%
\$20,000–\$50,000	\$100 + 0.22%
\$50,000–\$500,000	\$155 + 0.11%
超过\$500,000	\$255 + 0.09%

- 最低收益39美元。

broker.c

要求用户输入交易额，然后显示佣金数额：

Enter value of trade: 30000

Commission: \$166.00

```
/* Calculates a broker's commission */
```

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

```
int main(void)
```

```
{
```

```
    float commission, value;
```

```
    printf("Enter value of trade: ");
```

```
    scanf("%f", &value);
```

```
if (value < 2500.00f)
    commission = 30.00f + .017f * value;
else if (value < 6250.00f) // 2500.00f <= value
    commission = 56.00f + .0066f * value;
else if (value < 20000.00f)
    commission = 76.00f + .0034f * value;
else if (value < 50000.00f)
    commission = 100.00f + .0022f * value;
else if (value < 500000.00f)
    commission = 155.00f + .0011f * value;
else
    commission = 255.00f + .0009f * value;

if (commission < 39.00f)
    commission = 39.00f;
printf("Commission: $%.2f\n", commission);
return 0; }
```

5.2.4 悬空else问题

- 例子： x/y

```
if (y != 0) {  
    if (x != 0)  
        result = x / y; }//加括号强制配对（内层if语  
else                               句结束）  
    printf("Error: y is equal to 0\n");
```
- else和谁匹配？
 - 从缩进看，编程者想else和外层if匹配，
 - 实际是和内层匹配的。else与前面最近的if（未配对）配对

```
if (y != 0)  
    if (x != 0)  
        result = x / y;  
else  
    printf("Error: y is equal to 0\n");
```

if语句怎么写

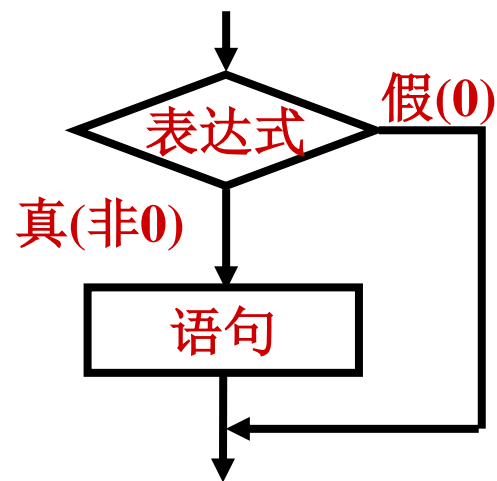
- If语句怎么写，怎么嵌套，怎么级联
- 先用自然语句把逻辑表达出来
- 再翻译成if语句

三八节放假安排

如果你是女性，则放假

if (你是女性) 放假

```
char sex; // 'F'表示女性
scanf("%c", &sex);
if(sex== 'F' ) printf("have a
holiday\n");
```



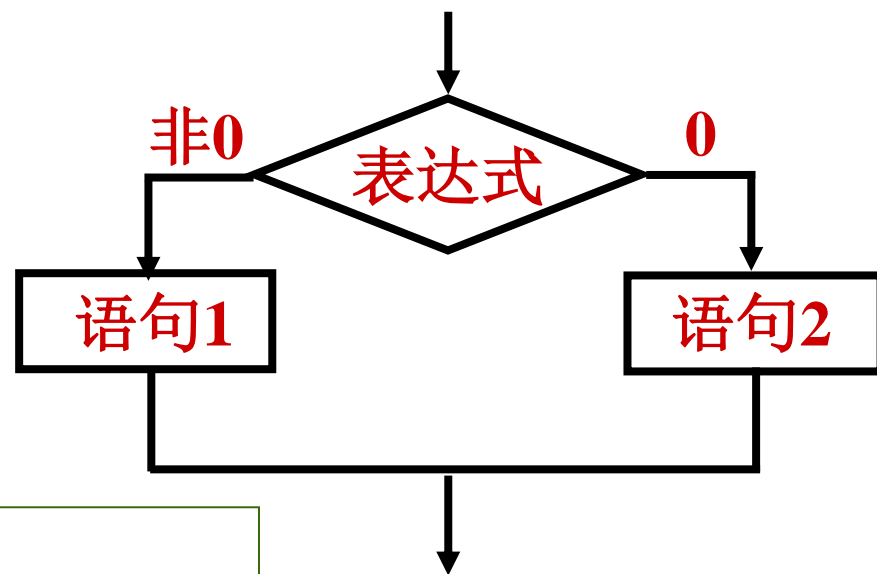
简单if语句

游泳池更衣指示

如果你是女性，则往右
否则[男性]，则往左

```
if (你是女性) turn right;  
else turn left;
```

```
char sex; // 'F'表示女性  
scanf("%c", &sex);  
if(sex== 'F' ) printf("turn right\n");  
else printf("turn left\n");
```



完整if语句

动物园售票

12岁及以下儿童，身高150及以下免票，150以上半价，60岁及以上半价，其他全价

如果你是12岁及以下儿童，并且如果你身高150及以下则免票，否则半票

否则，如果你是60岁以上老人，则半价
否则[你是成人] 全票

```
if(你是12岁及以下儿童)
    if(身高150及以下) free
    else half price;
else if(你是老人) half price
else full price
```

```
int age, height;
if(age <= 12)
    if (height <= 150) printf("free\n");
    else printf("half price\n");
else if(age >= 60) printf("half price\n");
else printf("full price\n")
```

嵌套if语句

- 两个年龄节点 12岁、60岁将人群分为三类
- 在12岁儿童段，身高150cm将儿童分为两类

if语句注意事项

- 若if或else分支下多条语句，必须复合

```
if (age <= 12)
```

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

```
    if (height <= 150) p = 0;
```

```
    else p = PRICE/2;
```

```
else if (age >= 60) p = PRICE/2;
```

```
[Error] D:\my Docs\C-Free\Temp\未命名1.cpp:13: error: expected `;' before "else"
```

```
[Error] D:\my Docs\C-Free\Temp\未命名1.cpp:14: error: expected primary-expression before "else"
```

```
[Error] D:\my Docs\C-Free\Temp\未命名1.cpp:14: error: expected `;' before "else"
```

```
构建中止 未命名1: 6 个错误, 0 个警告
```

if语句的if或else子句（分支）如果有多条语句必须复合，否则编译器认为第一条语句后该子句就结束了。

5.2.5 条件表达式

- 根据条件产生值（简化这类if语句）
 - **if (条件) v=e1;**
 - **else v=e2;**
- 条件运算符：三元运算符
 - **由符号? 和符号: 组成。**
- 格式：表达式 1 ? 表达式 2 : 表达式 3
- 还原if
 - **if(表达式1) 表达式2**
 - **else 表达式3**
- 结果：
 - **计算表达式1的值：**
 - **如果不为0（真），计算表达式2的值，为最后结果。**
 - **如果为0，计算表达式3的值，为最后结果。**

例子

```
int i, j, k;
```

```
i = 1;
```

```
j = 2;
```

```
k = i > j ? i : j;
```

```
/* k is now 2 ,
```

- 意义：为 i，j 中的最大值*/

```
k = (i >= 0 ? i : 0) + j;
```

```
/* k is now 3 */
```

- 条件运算符优先级
 - 只比赋值运算符高，比其他的都低。

例子

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

- 替换为：

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

- 可以应用于return语句：

```
return i > j ? i : j;
```

- 可以应用于printf语句：

```
printf("%d\n", i > j ? i : j);
```

5.2.6 C89中的布尔值

- C89没有定义布尔值类型。
- 可以定义一个int变量，赋值为0或1：

```
int flag;
```

```
flag = 0; ...
```

```
flag = 1;
```

- 为了程序更便于理解，可以定义宏：

```
#define TRUE 1
```

```
#define FALSE 0
```

例子

- 使用上页的宏：
flag = FALSE; ...
flag = TRUE;
- 判定flag为真
if (flag == TRUE) ...
或（等价）：**if (flag) ...**
- 判定flag为假
if (flag == FALSE) ...
或：**if (!flag) ...**

flag	flag==TRUE
1	1
0	0

定义类型的宏

- 定义一个int型的宏：
 - **#define BOOL int**
- 声明布尔型时就用BOOL代替int
 - **BOOL flag;**
- 预编译后
 - **int flag;**
- 宏定义实质：另起名字
 - **eg. #define PI 3.14**
- 作用：三便
 - 便于书写
 - 便于修改
 - 便于理解

5.2.7 C99中的布尔值

- C99中提供了 `_Bool` 类型：
 - **`_Bool flag;`**
- `_Bool` 是一个整形值（无符号整形），只能赋值为 0或1。
- 往 `_Bool` 变量中存储非零值会使值变为 1：
 - **`flag = 5; /* flag is assigned 1 */`**

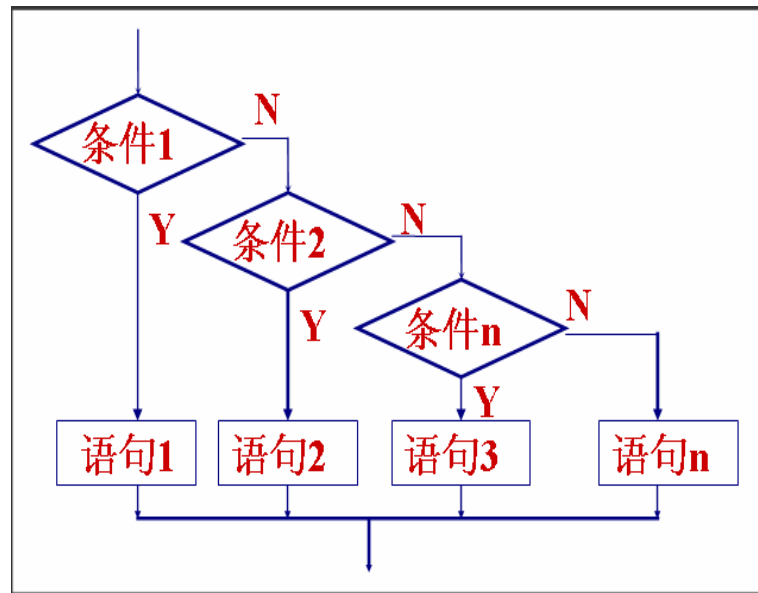
C99中的布尔值

- C99提供了一个新头文件 `<stdbool.h>`,
 - 提供了**bool**宏,
 - `#define bool _Bool`
 - 提供了 **true**和**false**两个宏表示1和0
- 如果`#include <stdbool.h>`, 就可以:
 - **`bool flag; /* same as _Bool flag; */`**
 - **`flag = false; ...`**
 - **`flag = true;`**

5.3 switch语句

- 把表达式和一系列值比较，可以用级联式if语句：

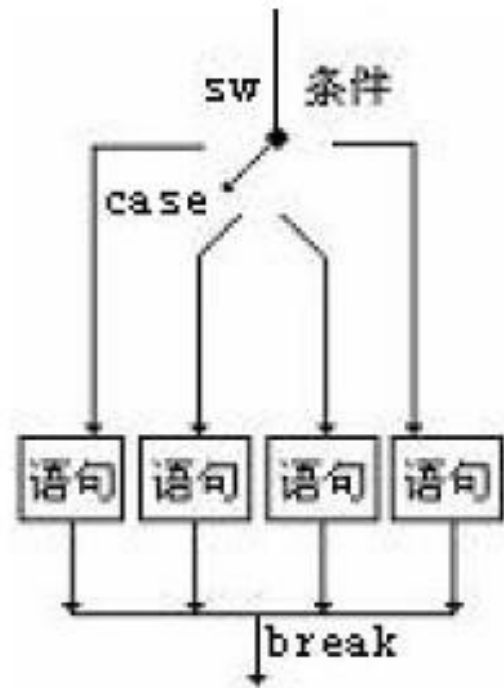
```
if (grade == 4)
    printf("Excellent");
else if (grade == 3)
    printf("Good");
else if (grade == 2)
    printf("Average");
else if (grade == 1)
    printf("Poor");
else if (grade == 0)
    printf("Failing");
else
    printf("Illegal grade");
```



水晶鞋挑灰姑娘——挨个儿试

switch语句

```
switch (grade) {  
    case 4:  printf("Excellent");  
             break;  
    case 3:  printf("Good");  
             break;  
    case 2:  printf("Average");  
             break;  
    case 1:  printf("Poor");  
             break;  
    case 0:  printf("Failing");  
             break;  
    default: printf("Illegal grade");  
}
```



(图 4.4.1)

弄清鞋码，对号入座

switch语句

- 比级联式if语句更容易阅读，执行速度也快。格式：


```
switch (号式) { //控制表达式  
    case 常座1式1: 语句1 //分支标号  
    case 常座2式2: 语句2  
    ...  
    case 常座n式n: 语句n  
    default 座n+1  
}
```

switch语句

- （座，鞋码）分支标号： case 常量表达式
 - 不能包含变量和函数调用。
 - 整数
- （号，脚码）控制表达式：
 - 整型（字符可当整数处理）
 - 不能用浮点数和字符串
- 语句：
 - 每个分支标号后可以跟任意数量的语句，
 - 不需要花括号，case即括号
 - 最后通常是break语句：结束switch
 - 分支结束“结账走人”

switch语句

- 多个分支共用一组语句:

```
switch (grade) {  
    case 4:  
    case 3:  如果加break;?  
    case 2:  
    case 1:    printf("Passing");  
              break;  
    case 0:    printf("Failing");  
              break;  
    default:   printf("Illegal grade");  
              break;  
}
```

轮滑：多码一鞋

switch语句

- 为了节省空间，可把多个分支放一行。

```
switch (grade) {  
    case 4: case 3: case 2: case 1:  
        printf("Passing");  
        break;  
    case 0: printf("Failing");  
        break;  
    default: printf("Illegal grade");  
        break;  
}
```

- 不要求必须有default，如果没有，控制表达式的值和任何分支都不匹配，就会传给switch后面的语句。

布尔问题

- 布尔运算boolean: 数字符号化的逻辑推演法, 很多计算机语言中就是**逻辑运算**, 结果为布尔值。
 - 逻辑判断, 结果: 正确(真)、错误(假)
 - Eg, 考试, 计算题int, 选择题char, 填空/问答题char[N], 判断题bool型
- C89没有定义布尔值类型, 定义int变量, 赋值为0或1。

int flag = 0; // 旗帜、标示、标记

flag = age <= 12 || age >= 60 || soldier_card;

if(flag) puts("free");

布尔问题

- 为了程序更便于理解，可以定义宏：

```
#define TRUE 1
```

```
#define FALSE 0
```

```
#define BOOL int
```

```
BOOL flag;
```

```
flag = FALSE;
```


温故而知新——条件表达式

- 条件运算符： ? :
- 三元运算符
- 格式：
 - 表达式 1 ? 表达式 2 : 表达式 3
- 结果：
 - If表达式1为真（非0）为表达式2的值，
 - 否则为表达式3的值

温故而知新——switch语句

- 比级联if更易阅读，执行速度也快。
- 格式：

```
switch (号) { //控制表达式  
    case 常量 座1: 语句1 //分支标号  
    case 常量 座2: 语句2  
    ...  
    case 常量 座n: 语句n  
    default: 语句n+1
```

要求：

号、座都为整数（包括字符）

座为常数（常量表达式，不含变量函数调用）

break的作用

- 分支处理结束，中断并跳出switch，跳到switch后
- 没有break：
 - 一个分支接一个分支，直到break或}。
- 例子：


```
switch (grade) {  
    case 4: printf("Excellent");  
    case 3: printf("Good");  
    case 2: printf("Average");  
    case 1: printf("Poor");  
    case 0: printf("Failing");  
    default: printf("Illegal grade");    }
```
- 如果grade等于3，结果为：
GoodAveragePoorFailingIllegal grade

例子：显示法定格式的日期

- 要求按下列格式显示日期：
 - **Dated this ____ day of ____, 20__.**
- 要求用户以mm/dd/yy方式录入日期，然后按上面方式显示
 - **Enter date (mm/dd/yy): 10/19/19**
 - **Dated this 19th day of October, 2019.**
- 典型“对号入座”，用switch语句实现
 - 为日期添加“th”（或“st”、“nd”、“rd”）。
 - 数字月份转换成英文表达

date.c

```
#include <stdio.h>
int main(void){
    int month, day, year;
    printf("Enter date (mm/dd/yy): ");
    scanf("%d /%d /%d", &month, &day, &year);
    printf("Dated this %d", day);
    switch (day) {
        case 1: case 21: case 31:
            printf("st"); break;
        case 2: case 22:
            printf("nd"); break;
        case 3: case 23:
            printf("rd"); break;
        default: printf("th"); break;
    }
}
```



```
printf(" day of ");
switch (month) {
    case 1:  printf("January");    break;
    case 2:  printf("February");   break;
    case 3:  printf("March");      break;
    case 4:  printf("April");      break;
    case 5:  printf("May");        break;
    case 6:  printf("June");       break;
    case 7:  printf("July");       break;
    case 8:  printf("August");     break;
    case 9:  printf("September");  break;
    case 10: printf("October");     break;
    case 11: printf("November");   break;
    case 12: printf("December");   break;
}
printf(", 20%.2d.\n", year);
return 0;
}
```

编程练习——考试评语

- 根据学生分数，输出优、良、中、及格和不及格

int s;

优秀: $90 \leq s \leq 100$;

良: $80 \leq s < 90$;

中: $70 \leq s < 80$;

及格: $60 \leq s < 70$;

不及格: $s < 60$;

- 先用if语句实现，再用switch语句实现

数学表示:

$$90 \leq s \leq 100$$



逻辑表达式:

$$s \geq 90 \ \&\& \ s \leq 100$$

if语句——算法（伪代码）

- `printf()` 提示输入分数
- `scanf(分数)` ;
- `if(分数90以上)` 优;
- `else if(分数80以上)` 良;
- `else if(分数70以上)` 中;
- `else if(分数60以上)` 及格;
- `else` 你挂了;

考试评语——if

```
#include <stdio.h>
int main()
{
    int s;
    printf("what is your score?\n");
    scanf("%d", &s);
    if(s>=90) printf("excellent");
    else if(s>=80) printf("good");
    else if(s>=70) printf("average");
    else if(s>=60) printf("pass");
    else printf("failing");
    return 0;
}
```

switch语句——算法

- scanf（分数）；
- switch（表达式） {
 - **case 标号（常量表达式）1： 优；**
 - **break；**
 - **case 标号2： 良；**
 - **break；**
 - **case 标号3： 中；**
 - **break；**
 - **case 标号4： 及格；**
 - **break；**
 - **default： 不及格；**
 - **break； }**

考试评语——switch

```
switch(s) {  
    case 100: case 99: .....case90:  
        printf("excellent");  
        break;  
    case 89: .....case 80: printf("good");  
        break;  
    case 79: .....case 70: printf("average");  
        break;  
    case 69: .....case 60: printf("pass");  
        break;  
    default: printf("failing");  
}
```

考试评语——switch

```
switch(s/10) {  
    case 10: case 9:  
        printf("excellent");  
        break;  
    case 8: printf("good");  
        break;  
    case 7: printf("average");  
        break;  
    case 6: printf("pass");  
        break;
```

注意：100分