

## 本章导读

本章向读者介绍C语言三种基本流程控制结构：**顺序**、**选择**、**循环**中的语句形式，从实例分析着手加深对C语言程序开发设计过程的感性认识，强化培养编程思路，掌握顺序和选择结构程序设计的方法。

## 本章主要知识点

- (1) 程序的三种基本结构
- (2) if 语句
- (3) switch 语句
- (4) 选择结构的嵌套

# 第三章 顺序和选择结构程序设计

## 3.1 程序的三种基本结构

## 3.2 语句概述

## 3.3 顺序结构程序举例

## 3.4 选择结构——if 语句

## 3.5 选择结构——switch 语句

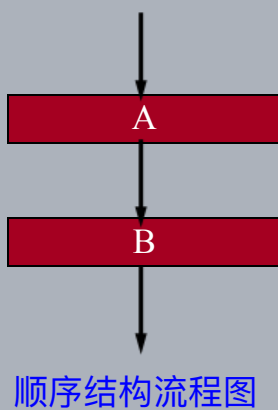
## 3.1 程序的三种基本结构

C语言提供三种结构化语句来控制程序的执行流程：

- 顺序结构**：一种按书写顺序执行的结构。
- 选择结构**：根据运行时的情况自动选择要执行的语句。
- 循环结构**：根据情况自动重复执行有关语句。

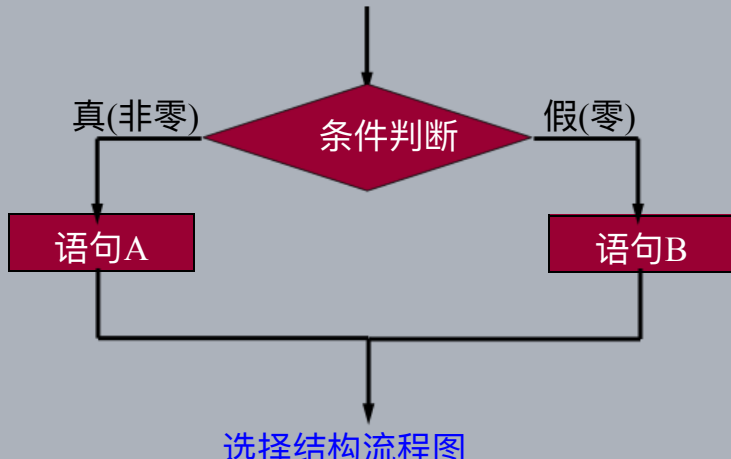
### 3.1.1 顺序结构

顺序结构即按照语句书写顺序执行的程序结构。



### 3.1.2 选择结构

选择结构又称为分支结构，是指有条件地选择要执行程序段的语句结构。

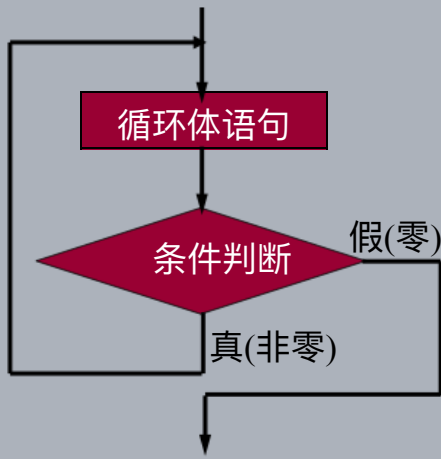
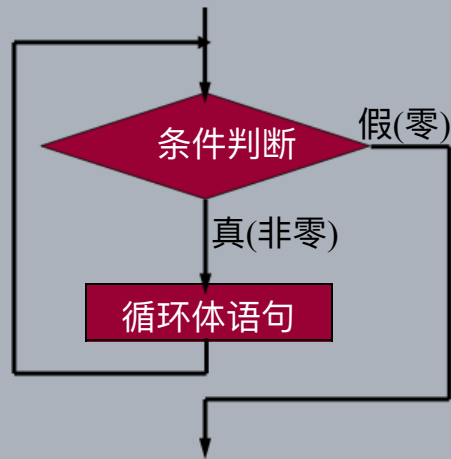


### 3.1.3 循环结构

循环结构又称为重复结构，是指根据条件重复执行循环体的语句结构。

一、当型循环

二、直到型循环



## 3.2 语句概述

表达式语句 (表达式; )

简单语句

函数调用语句 (函数名 (参数表) ;

空语句 (;)

复合语句 ({ 一条或多条语句 })

结构化语句

条件语句( if语句, switch语句)

while语句

循环语句 do\_while语句

for语句

流程控制语句

break语句

条件转向语句 continue语句

非结构化语句

return语句

无条件转向语句 (goto语句)

### 3.2.1 简单语句

#### 一、表达式语句

表达式的后面加一个分号就构成了一个语句，例如：sum=a+b;。事实上，C语言中有使用价值的表达式语句主要有3种：

- (1) 赋值语句。例如：sum=a+b;
- (2) 自增、自减运算符构成的表达式语句。例如：i++;
- (3) 逗号表达式语句。例：x=1, y=2;

#### 二、函数调用语句

由一个函数调用加上一个分号组成一个语句，例如：scanf("%d%d",&a,&b);  
printf("a=%d,b=%d\n",a,b);

### 3.2.2 空语句

空语句就是一个分号。

格式：

;

**作用：**什么都不做。一般和后面章节所讲循环语句结合使用起到延时作用。

**注意：**虽然空语句什么都不做，也不能在程序中随便添加，在某些位置添加空语句可能会引起程序的逻辑错误。

### 3.2.3 复合语句

用一对大括号 {} 包含起来的一条或者若干条语句就是复合语句。

格式：{ [数据的说明部分]  
    执行语句部分  
}

**注意：**

- (1) 复合语句在语法上相当于一条语句；
- (2) 复合语句内的“数据的说明部分”定义的变量属于局部变量，只能在复合语句内使用，超过大括号将不能再使用。

### 3.2.4 流程控制语句

流程控制语句包括结构化语句和非结构化语句。

用在选择结构中的语句有：

if语句、switch语句、break语句

用在循环结构中的语句有：

while语句、do-while语句、for语句、break语句、continue语句。

return语句的作用是函数返回；goto语句是无条件跳转语句（不提倡使用）。

## 3.3 顺序结构编程举例

#### 键盘输入函数 scanf() 的使用：

调用 scanf 函数可以从键盘格式化输入各种数据。

函数原型：scanf(“格式控制”, [地址项列表]);

例如：scanf(“%d%d”, &a, &b);

**说明：**

- (1) 格式控制部分由两类符号组成：普通字符（原模原样输入）、格式说明符（以%号开头）。注意没有转义字符，这一点需和 printf 区别开。
- (2) “格式控制”部分中必须有以 % 开头的格式说明符，有多少个格式说明符，后面就有多少个地址项与之对应，注意地址项中每个变量前面有取地址符

## 3.3 顺序结构编程举例

续上：

(3) 格式说明符的使用方法与printf函数一样：

int 整型 —— %d

char 字符型 —— %c

float 单精度型 —— %f

double 双精度型 —— %lf

字符串 —— %s

(4) 当“格式控制”部分中有“普通字符”时，必须原样输入，否则会导致程序运行错误。例如要从键盘输入10和20两个整数分别给变量a、b，scanf的用法为：

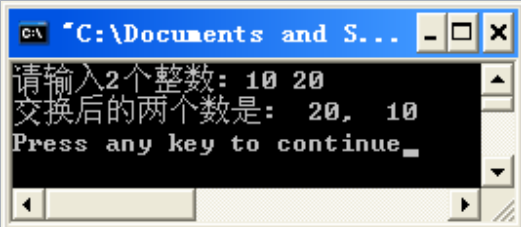
```
scanf(“%d%d”, &a, &b);
```

```
scanf(“%d, %d”, &a, &b);
```

## 3.3 顺序结构编程举例

例1：从键盘输入两个整数a、b，将它们交换，例如a为3，b为5，交换后a为5，b为3。

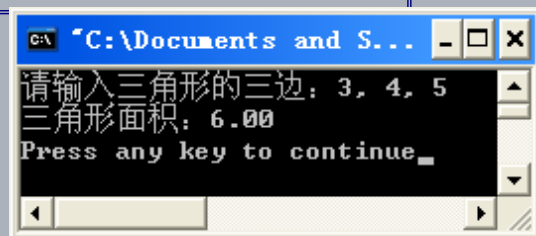
```
#include <stdio.h>
main()
{
    int a, b, t; // t 是一个
    printf("请输入2个整数: ");
    scanf("%d%d", &a, &b);
    t=a;  a=b;  b=t; //首尾相接的三条语句
    printf("交换后的两个数是: %d, %d\n", a, b);
}
```



## 3.3 顺序结构编程举例

例2：输入三角形的三边长，求三角形面积。

```
#include <stdio.h>
#include <math.h>
main()
{
    double a, b, c, s, area;
    printf("请输入三角形的三边: ");
    scanf("%lf,%lf,%lf", &a, &b, &c);
    s = 1.0 / 2 * (a + b + c);
    area = sqrt(s * (s - a) * (s - b) * (s - c));
    printf("三角形面积: %.2f\n", area);
}
```



## 3.3 顺序结构编程举例

调用数学库函数时，需包含头文件 `math.h`

以下是常用数学库函数：

(1) 求次方值

`z=pow(x, y)`——求 $x$ 的 $y$ 次方值，结果赋值给 $z$

(2) 求开方值

`z=sqrt(x)`——求 $x$ 的开方值，结果赋值给 $z$

(3) 求绝对值

`z=fabs(x)`——求 $x$ 的绝对值，结果赋值给 $z$

(4) 求对数值

`z=log(x)`——求 $\ln(x)$ 的值，结果赋值给 $z$



## 3.4 选择结构——if语句

### 3.4.1 单分支选择if语句

### 3.4.2 双分支选择if语句

### 3.4.3 多分支选择if语句

### 3.4.4 选择结构的嵌套

### 3.4.5 if语句练习题

### 3.4.6 if语句编程举例

### 3.4.1 单分支选择 if 语句

if 语句有三种形式：单分支选择结构、双分支选择结构、多分支选择结构。

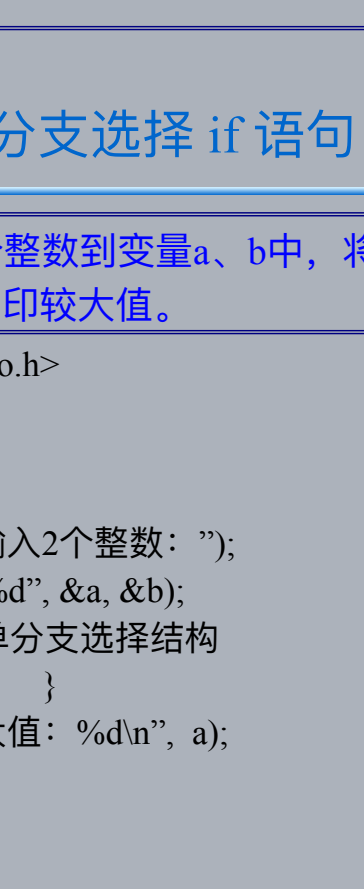
#### 一、单分支选择结构

语句形式：

if(表达式)

语句;

此处最好有缩进



### 3.4.1 单分支选择 if 语句

说明：

- (1) if(表达式) 中的“表达式”可以是任何符合C语言语法的表达式，当表达式的值为“非零”时为真；当表达式的值为“零”时为假。
- (2) if(表达式) 只能自动结合一条语句，当有多条语句时，必须用大括号括起来，构成复合语句，因为复合语句在语法上相当于一语句。
- (3) if(表达式) 的后面不能随意加分号，如果加了分号，写成 if(表达式); 则表示此时的语句体是空语句，将会引起程序运行时的逻辑错误。

### 3.4.1 单分支选择 if 语句

例3：从键盘输入两个整数到变量a、b中，将较大值存放在变量a中，并打印较大值。

```
#include <stdio.h>
main()
{
    int a, b;
    printf("请输入2个整数: ");
    scanf("%d%d", &a, &b);
    if(a < b) //单分支选择结构
    {
        a = b;
    }
    printf("较大值: %d\n", a);
}
```

### 3.4.2 双分支选择 if 语句

#### 二、双分支选择结构

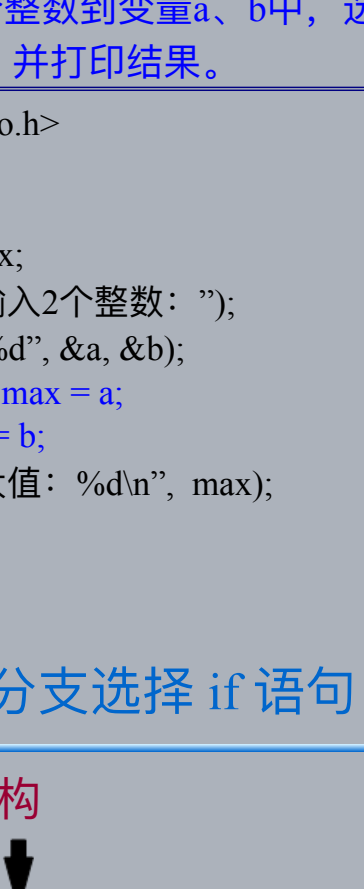
语句形式：

if(表达式)

语句1;

else

语句2;



### 3.4.2 双分支选择 if 语句

说明：

- (1) if(表达式) 或者else都只能自动结合一条语句，当有多条语句时，必须用大括号括起来，构成复合语句。例如：

```
if(a > b)
{
    a++;
    b++;
}
else
{
    a--;
    b--;
}
```

### 3.4.2 双分支选择 if 语句

说明：

- (2) else必须有对应的if与之配套使用，即if和else应成对出现。因此在双分支选择结构中，if(表达式)的后面一定不能加分号，如果写成

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

这将构成语法错误，因为if和else中间有两条语句。

### 3.4.2 双分支选择 if 语句

例4：从键盘输入两个整数到变量a、b中，选出较大值存放在变量max中，并打印结果。

```
#include <stdio.h>
main()
{
    int a, b, max;
    printf("请输入2个整数: ");
    scanf("%d%d", &a, &b);
    if(a >= b) max = a;
    else max = b;
    printf("较大值: %d\n", max);
}
```

### 3.4.3 多分支选择 if 语句

#### 三、多分支选择结构

语句形式：

if(表达式1)

语句1;

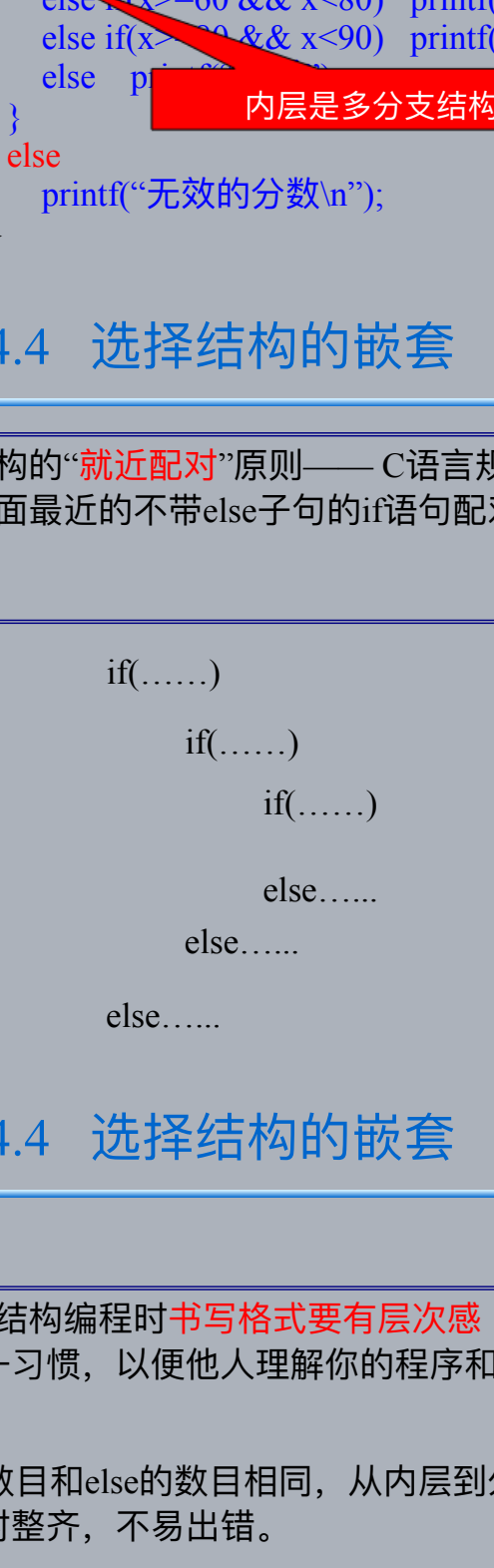
else if(表达式2)

语句2;

.....

else

语句n;



### 3.4.3 多分支选择 if 语句

说明：

- (1) 多分支选择结构的“表达式”都是在 if 后面，不能在else后面，即只能有“if(表达式)”的形式，或者“else if(表达式)”的形式，不能有“else(表达式)”的形式。

- (2) 对于最后一个分支  
如果需要判断条件 就写成 else if(表达式n) 语句n;  
如果不需要判断条件 就写成 else 语句n;

### 3.4.3 多分支选择 if 语句

例5：编程计算分段函数。

```
#include <stdio.h>
main()
{
    float x, y;
    printf("请输入 x 的值: ");
    scanf("%f", &x);
    if(x <= 1) y = x + 5;
    else if(x > 1 && x <= 10) y = 2 * x;
    else if(x > 10) y = 3 / (x - 10);
    printf("y = %f\n", y);
}
```

$$y = \begin{cases} x+5 & (x \leq 1) \\ 2x & (1 < x \leq 10) \\ \frac{3}{x-10} & (x > 10) \end{cases}$$

注意：不能写成 else if(1 < x <= 10)

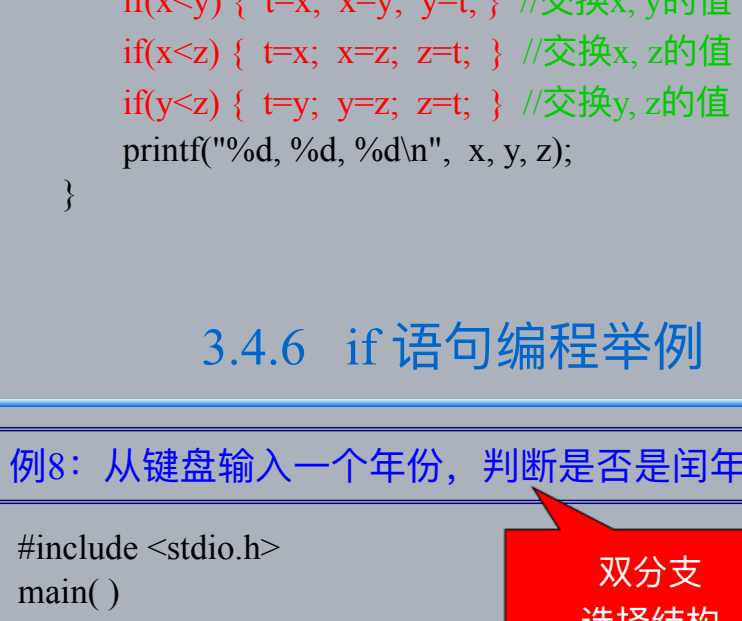
### 3.4.4 选择结构的嵌套

在一个if语句中又包含另一个if语句，从而构成了if语句的嵌套结构。  
内嵌的if语句既可以嵌套在if子句中，也可以嵌套在else子句中。

if语句的嵌套形式不是刻意去追求的，而是在解决问题过程中随着解决问题的需要而采用的。

### 3.4.4 选择结构的嵌套

例如以下嵌套结构：



### 3.4.4 选择结构的嵌套

例6：从键盘输入一个分数，打印该分数对应的级别。0~59分之间属于“不及格”；60~79分之间属于“中等”；80~89分之间属于“良好”；90~100分之间属于“优秀”。

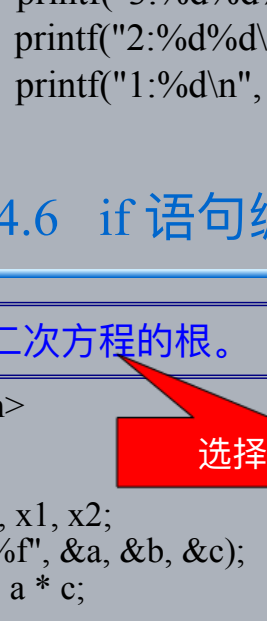
```
#include <stdio.h>
main()
{
    int x;
    printf("请输入一个分数: ");
    scanf("%d", &x);
    if(x >= 0 && x <= 100)
    {
        if(x >= 0 && x < 60) printf("不及格");
        else if(x >= 60 && x < 80) printf("中等");
        else if(x >= 80 && x < 90) printf("良好");
        else if(x >= 90 && x <= 100) printf("优秀");
    }
    else printf("无效的分数\n");
}
```

外层是双分支结构

内层是多分支结构

### 3.4.4 选择结构的嵌套

重点：嵌套结构的“就近配对”原则——C语言规定，else子句总是和前面最近的不带else子句的if语句配对，与书写格式无关。



### 3.4.4 选择结构的嵌套

说明：

- (1) 注意嵌套结构编程时书写格式要有层次感（好的程序员应该养成这一习惯，以便他人理解你的程序和自己将来修改程序）。
- (2) 如果if的数目和else的数目相同，从内层到外层一一对应，书写格式对齐，不易出错。
- (3) 如果if与else的数目不一致，为实现编程者的意图，必要的时候必须加“{ }”，来强制确定配对关系。否则，就不能实现编程者的真正意图。

### 3.4.5 if语句练习题

练习：

(1) 以下程序段

```
int a,b,c;
a=10;b=50;c=30;
if(a>b) a=b;b=c;c=a;
printf("a=%d b=%d c=%d\n",a,b,c);
```

程序的运行结果是

- A) a=10 b=50 c=10    B) a=10 b=50 c=30  
C) a=10 b=30 c=10    D) a=50 b=30 c=50

【解析】考查if语句只能自动结合一条语句，若有多条语句要结合，则应加“{ }”。本题的“if(a>b)”只能自动结合一条语句“a=b;”

### 3.4.5 if语句练习题

(2) 有以下程序

```
#include <stdio.h>
main()
{
    int x=1, y=2, z=3;
    if(x>y)
        if(y<z) printf("%d", ++z);
    else printf("%d", ++y);
    printf("%d\n", x++);
}
```

程序的运行结果是

- A) 1 2 3    B) 1 3 2    C) 1 2 4    D) 1 3 4

【解析】考查if语句嵌套结构的“就近配对”原则，本题的外层是单分支选择结构，内层是双分支选择结构。

### 3.4.5 if语句练习题

(3) 有以下程序

```
main()
{
    int a=0, b=0, c=0, d=0;
    if(a==1) b=1; c=2;
    else d=3;
    printf("%d,%d,%d,%d\n", a, b, c, d);
}
```

程序的运行结果是

- A) 0,1,2,0    B) 0,0,0,3    C) 1,1,2,0    D) 编译有错

【解析】同时考查“if语句只能自动结合一条语句”和“就近配对”原则。本题的“if(a==1)”只能自动结合“b=1;”，语句“c=2;”就隔在了if和else中间，于是else没有与之配对的if。

### 3.4.6 if语句编程举例

例7：从键盘输入任意三个整数到变量x、y、z中，并按从大到小的顺序将三个数依次放在x、y、z中。

```
#include <stdio.h>
main()
{
    int x, y, z, t;
    scanf("%d%d%d", &x, &y, &z);
    if(x<y) { t=x; x=y; y=t; } //交换x, y的值
    if(x<z) { t=x; x=z; z=t; } //交换x, z的值
    if(y<z) { t=y; y=z; z=t; } //交换y, z的值
    printf("%d, %d, %d\n", x, y, z);
}
```

单分支选择结构

### 3.4.6 if语句编程举例

例8：从键盘输入一个年份，判断是否是闰年。

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

双分支选择结构

### 3.4.6 if语句编程举例

例9：从键盘输入一个不多于4位的正整数，求出它是几位数，并将各位上的数字逆序打印。

```
#include <stdio.h>
main()
{
    int x, ge, shi, bai, qian;
    scanf("%d", &x);
    qian = x / 1000;          bai = x % 1000 / 100;
    shi = x % 100 / 10;       ge = x % 10;
    if(qian!=0) printf("4:%d%d%d%d\n", ge, shi, bai, qian);
    else if(bai!=0) printf("3:%d%d%d\n", ge, shi, bai);
    else if(shi!=0) printf("2:%d%d\n", ge, shi);
    else if(ge!=0) printf("1:%d\n", ge);
}
```

多分支选择结构

### 3.4.6 if语句编程举例

例10：求一元二次方程的根。

```
#include <stdio.h>
main()
{
    float a, b, c, t, x1, x2;
    scanf("%f%f%f", &a, &b, &c);
    t = b * b - 4 * a * c;
    if(t < 0)
        printf("方程没有实根\n");
    else
    {
        x1 = (-b + sqrt(t)) / (2 * a);
        x2 = (-b - sqrt(t)) / (2 * a);
        printf("方程有两个相同的实根: ");
        printf("x1 = x2 = %.2f\n", x1);
    }
    else
    {
        x1 = (-b + sqrt(t)) / (2 * a);
        x2 = (-b - sqrt(t)) / (2 * a);
        printf("方程有两个不同的实根: ");
        printf("x1 = %.2f, x2 = %.2f\n", x1, x2);
    }
}
```

选择结构的嵌套

### 3.4.6 if语句编程举例

续上

```
if(t == 0)
{
    x1 = (-b + sqrt(t)) / (2 * a);
    printf("方程有两个相同的实根: ");
    printf("x1 = x2 = %.2f\n", x1);
}
else
{
    x1 = (-b + sqrt(t)) / (2 * a);
    x2 = (-b - sqrt(t)) / (2 * a);
    printf("方程有两个不同的实根: ");
    printf("x1 = %.2f, x2 = %.2f\n", x1, x2);
}
```

### 3.5 选择结构——switch语句

switch 语句的一般形式是：

```
switch(表达式)
```

```
{
```

```
    case 常量表达式1: [语句序列1;] [break;]
```

```
    case 常量表达式2: [语句序列2;] [break;]
```

```
    .....
```

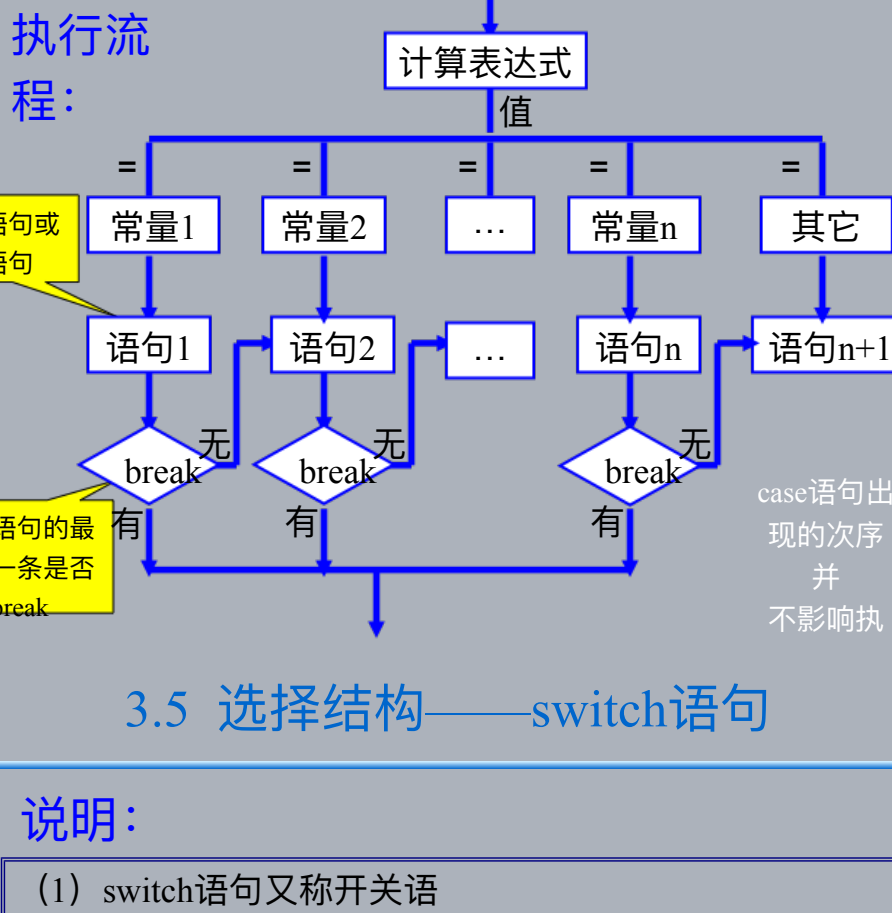
```
    case 常量表达式n: [语句序列n;] [break;]
```

```
    [ default: 语句序列n+1; ]
```

```
}
```



### 3.5 选择结构——switch语句



### 3.5 选择结构——switch语句

#### 说明：

- (1) switch语句又称开关语句。switch、case、break、default都是关键字。
- (2) “switch(表达式)”的后面不能加分号。并且表达式的值必须是整型数据或字符型数据，不能是实型数据。
- (3) “case 常量表达式:”中的表达式不能是变量或者变量表达式，并且该表达式的值应是整型数值或字符型数值。常量表达式后面要加冒号。
- (4) 各个“常量表达式”的值必须互不相同，以实现不同的分支选择。
- (5) 每个“语句序列”后面的break语句可以有也可以缺省。如果有break语句，则执行完该case分支就跳出switch语句；如果缺省了break语句，则执行完该case分支将继续执行下一个case分支。
- (6) 允许某个或某几个case后面缺省“语句序列”，这样就形成多个case分支共用一组语句序列的情况。
- (7) default分支的书写位置不受限制。

### 3.5 选择结构——switch语句

**例11：**假设用0, 1, 2, …, 6 分别表示星期日、星期一……星期六。现输入一个数字，输出对应的星期几的英文单词。如果输入3，输出“Wednesday”

```
main()
{
    int n;
    scanf("%d", &n);
    switch(n)
    {
        case 0: printf("Sunday\n");
        case 1: printf("Monday\n");
        case 2: printf("Tuesday\n");
        case 3: printf("Wednesday\n");
        case 4: printf("Thursday\n");
        case 5: printf("Friday\n");
        case 6: printf("Saturday\n");
        default: printf("Error");
    }
}
```

**运行结果：**

```
3
Wednesday
Thursday
Friday
Saturday
Error
```

### 3.5 选择结构——switch语句

**在每个case表达式后添加break，程序修改如右侧所示：**

```
main()
{
    int n;
    scanf("%d", &n);
    switch(n)
    {
        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("Error");
    }
}
```

**运行结果：**

```
3
Wednesday
```

在switch语句中使用break语句才能真正起到多分支的效果

### 3.5 选择结构——switch语句

**练习：**

(1) 若a、b、c、y、均是整型变量，正确的switch语句是

A) switch(a+b);  
{ case 1:y=a+b; break;  
case 0:y=a-b; break;  
}

B) switch(a\*b)  
{ case 3:  
case 1:y=a+b;break;  
case 3:y=b-a;break;  
}

C) switch a  
{ case c1 :y=a-b; break;  
case c2: x=a\*d; break;  
default:x=a+b;  
}

D) switch(a-b)  
{ default:y=a\*b;break;  
case 3:  
case 4:x=a+b;break;  
case 10:y=a-b;break;  
}

【解析】考查switch语句中某个case分支缺省break语句的情况。程序执行过程为：

表达式 a%3 的值为1

执行case 1 □ m=1

表达式 b%2 的值为1

执行default □ m=2

执行case 0 □ m=3

### 3.5 选择结构——switch语句

**例12：**从键盘输入一个年份和一个月份，打印该月的天数。（假定输入的年份是有效数值）

```
#include <stdio.h>
main()
{
    int year, month;
    printf("请输入一个年份和一个月份：");
    scanf("%d%d", &year, &month);
    switch(month)
    {
        case 4:
        case 6:
        case 9:
        case 11: printf("30天\n"); break;
        case 12: printf("31天\n"); break;
        case 2:
            if(year%400==0 || (year%4==0 & year%100!=0))
                printf("29天\n");
            else
                printf("28天\n");
            break;
        default: printf("输入的月份无效.\n");
    }
}
```

所有30天的月份共用一条打印语句

所有31天的月份共用一条打印语句

对2月份的处理

输入错误月份的处理

### 3.5 选择结构——switch语句

**实验作业解答**

输入三个数，求这三个数的最大值

方法一：

```
#include <stdio.h>
main()
{
    int x, y, z, t;
    scanf("%d%d%d", &x, &y, &z);
    if(x<y) { t=x; x=y; y=t; } //交换x, y的值
    if(x<z) { t=x; x=z; z=t; } //交换x, z的值
    printf("最大数为: %d\n", x);
}
```

方法二：

```
#include <stdio.h>
main()
{
    int a,b,c;
    printf("请输入三个整数：");
    scanf("%d%d%d",&a,&b,&c);
    (a>b)?(a>c,printf("其最大值为%d:",a)):
    (b>c?printf("其最大值为%d:",b):printf("其最大值为%d:",c));
}
```

### 实验六

商品优惠消费，优惠比例f如右表：	优惠比例 (f)	优惠条件
	0	X<100(元)
	0.05	100<=x<500
	0.1	500<=x<2000
	0.15	2000<=x

已知商品价格x元，求优惠后实际应付多少钱

