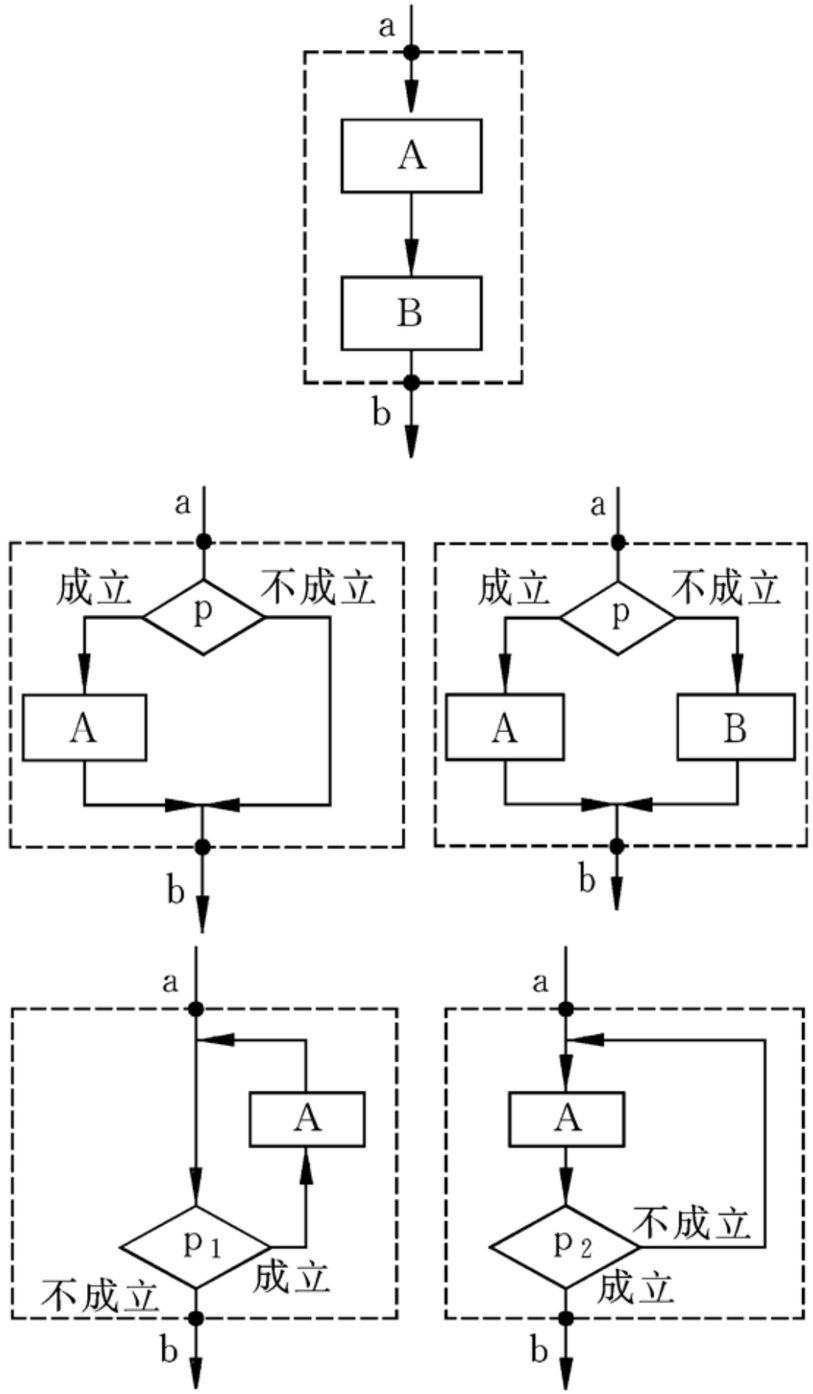
C语言程序设计

计算机科学与技术学院

选择结构程序设计

算法的基本结构

- **三种基本结构**：顺序结构、选择结构、循环结构
- **选择结构**：
  - 根据条件进行判断
  - 选择执行相应操作
  - 也称为分支控制结构
  - 一般分为单分支、双分支和多分支三种结构



选择结构

- 单分支形式

  如果(条件成立)
  那么{...}

  - 妈妈：
    去楼下的超市买 1 个锅；
    如果（有鸡蛋），
    那么 {买 20 个}

  - 程序员：

选择结构

- 单分支形式

  如果(条件成立)
  那么{...}

  - 妈妈：
    去楼下的超市买 1 个锅；
    如果（有鸡蛋），
    那么 {买 20 个}

  - 程序员：

选择结构

- 双分支形式

  如果（成绩大于等于60）

  那么 打印 "及格"

  否则 打印 "不及格"

- 多分支形式

  如果（成绩>=90)

  那么 打印 "A"

  如果（90>成绩>=80)

  那么 打印 "B"

  如果（80>成绩>=70)

  那么 打印 "C"

  如果（70>成绩>=60)

  那么 打印 "D"

  如果（成绩<60)

  那么 打印 "E"

C中的分支语句

- if-else

```
if (条件){
    …
}
else{
    …
}
```

- switch

```
switch(){
    …
}
```

条件的表示

- 例：
  如果（成绩大于等于60）
  那么 打印 "及格"
  否则 打印 "不及格"

```
if (cj >= 60) {
    printf("及格");
}
else{
    printf("不及格");
}
```

- （）中的条件：逻辑值
  - C中没有专门的逻辑数据类型
  - 0：假
  - 1：真
- 关系表达式：
  - 比较对象之间的大小关系
  - 关系运算符
  - 关系表达式的结果是逻辑值
- 逻辑表达式：
  - 多个条件同时满足或部分满足
  - 逻辑运算符
  - 逻辑表达式的结果是逻辑值

关系运算符

- 关系运算符
  - <、>、<=、>=： 大小
  - ==、!=： 相等、不相等
  - <、>、<=、>= 优先级 高于 == !=
- 关系运算的两种结果：
  - 1：真
  - 0：假
- 8 > 9 结果为：0
- 4 + 5 >= 9 结果为：1

逻辑运算符

- 逻辑运算符
  - !：逻辑求反（单目运算）
  - &&：逻辑与（双目运算）
  - ||：逻辑或（双目运算）
- 优先级：
  - ! 高于 &&
  - && 高于 ||
- 真值表

a	b	!a	!b	a&&b	a||b
真	真	假	假	真	真
真	假	假	真	假	真
假	真	真	假	假	真
假	假	真	真	假	假

关系与逻辑表达式

- 数学表达式5<x<10如何表示?
  - x < 10 && x > 5
  - （x < 10） &&（x > 5）
- 数学表达式y<=5或y>=10如何表示?
  - y >= 10 || y <= 5
  - （y >= 10） ||（y <= 5）

关系与逻辑表达式

- 1900年是闰年吗?
- 闰年的条件:
  - 能被4整除， 但不能被100整除
  - 或者
  - 能被400整除
- (year %4 == 0 && year %100 != 0) || (year % 400 == 0)

关系与逻辑表达式进阶：0、1、非0

- 逻辑运算表达式中的操作数：
  - 0 视为 “假”
  - 非0 视为 “真”
- 逻辑运算表达式的运算结果：
  - “假” 使用 0 表示
  - “真” 使用 1 表示

- 例：

```
0&&1 -> 0

9||0 -> 1
```

若：

```
char c='b';
int i = 3, j = 3;
double x = 0.0;
```

则：

```
x || i && j - 3  等价于  x || (i && (j - 3)) ->  0
'a' <= c && c<='z'   等价于  ('a' <= c ) && (c <='z')   ->    1
c - 1=='a' || c + 1 == 'z'  等价于  ((c-1)=='a')|| ((c+1)=='z')    -> 1
```

关系与逻辑表达式进阶：短路

```
!(8 > 9)                    -> 1
8 < 9 && 7 > 6              -> 1
9 > 0 || 8 > 9              -> 1
9 > 0 || 8 < 9              -> 1
0 > 1 && 2 > 3              -> 0
0 > 1 && 2 < 3              -> 0
```

- 在一个复杂的逻辑表达式，有的表达式可能没有被运算到，称为短路
- 例：

```
a=9; b=8; c=3;d=0;
x = a<b && (d=c);
z = (c=b) && a>b;
y = a<b || (c=d);
```

if语句

- 形式一：

```
if (expression)
    {statements}
```

- 形式二：

```
if (expression)
    {statement1}
else
    {statement2}
```

- 形式三：

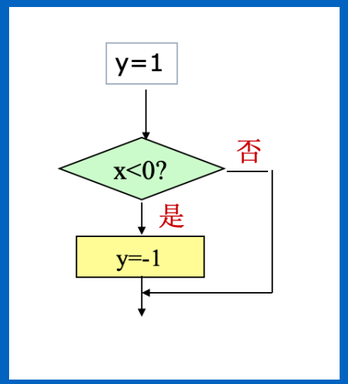
```
if (expression1)
    {statement1}
else if  (expression2)
    {statement2}
...
else
    {statementn}
```

- 若statement超过一条语句，必须用“ {} ”括起来

if语句 -单分支

例：编写程序，当输入x时，根据以下公式计算y.

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




```c
#include <stdio.h>
int main (){
    int x, y;
    scanf(“%d”,&x);

    y=1;
    if(x < 0)
        y = -1;

    printf(“%d”,y);

    return 0;
}
```

易错点：条件中的 == 与 =

- == 是关系运算符

```
int a = 0, b = 0;
if (a == 1)
    b = a;
printf("%d,%d", a, b);

//输出:0,0
```

- = 是赋值运算符

```
int a = 0, b = 0;
if (a = 1) //逻辑错误
    b = a;
printf("%d,%d", a, b);

//输出:1,1
```

易错点：分支中的{ }

例：输入两个实数，按代数值由小到大的次序输出这两个数。

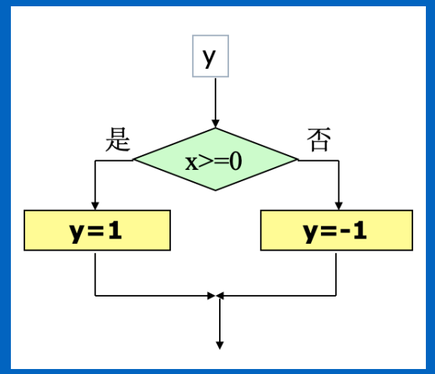
```c
#include <stdio.h>
int main (){
    float a,b,t;
    scanf("%f,%f",&a,&b);
    if(a > b){  //不要漏掉{}!
        t = a;
        a = b;
        b = t;
    }

    printf("%5.2f,%5.2f",a,b);

    return 0;
}
```

```c
#include <stdio.h>
int main (){
    float a,b,t;
    scanf("%f,%f",&a,&b);
    if(a > b)   //执行过程不同!
        t = a;

    a = b;
    b = t;


    printf("%5.2f,%5.2f",a,b);

    return 0;
}
```

if语句 -双分支

例：编写程序，当输入x时，根据以下公式计算y.

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




```
#include <stdio.h>
int main (){
    int x, y;
    scanf(“%d”,&x);

    if(x >= 0)
        y = 1;
    else
        y = -1;

    printf(“%d”,y);

    return 0;
}
```

if语句 -双分支

例：输入两个实数，按代数值由小到大的次序输出这两个数。

```c
#include <stdio.h>
int main (){
    float a,b,t;
    scanf("%f,%f",&a,&b);
    if(a > b){
        t = a;
        a = b;
        b = t;
    }
    printf("%5.2f,%5.2f",a,b);

    return 0;
}
```

```c
#include <stdio.h>
int main (){
    float a,b;
    scanf("%f,%f",&a,&b);

    if(a > b)
        printf("%5.2f,%5.2f",b,a);
    else
        printf("%5.2f,%5.2f",a,b);

    return 0;
}
```

条件运算符

- d1 ? d2 : d3
- 三目运算符
  - 如果d1的值为1，整个表达式的值就是表达式d2的值；
  - 如果d1的值为0，整个表达式的值就是d3的值

- 例：

```
int a=3, b=4, c;
c=(a > b ? a : b); // c=4
```

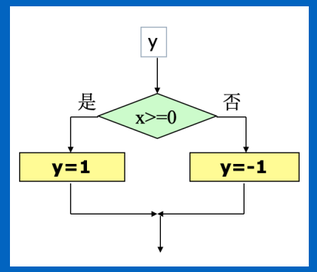
- 相当于精简的 if-else 语句

```
int a=3, b=4, c;
if (a > b)
    c = a;
else
    c = b;
```

if语句 -双分支

例：编写程序，当输入x时，根据以下公式计算y.

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




```c
#include <stdio.h>
int main (){
    int x, y;
    scanf("%d",&x);

    printf("y=%d", x >= 0 ? 1 : -1);

    return 0;
}
```

if语句 -多分支

- 例：计算物流公司的运费，可以根据数量打折。

$$discount = \begin{cases} 1.0 & (quantity < 400) \\ 0.9 & (600 > quantity \geq 400) \\ 0.8 & (800 > quantity \geq 600) \\ 0.7 & (1000 > quantity \geq 800) \\ 0.6 & (quantity \geq 1000) \end{cases}$$

```
float quantity ;
if (quantity < 400) discount = 1.0;
else if(quantity < 600) discount = 0.9;  //  (quantity < 600) && ( quantity >= 400)
else if(quantity < 800) discount = 0.8;  //  (quantity < 800) && ( quantity >= 600)
else if(quantity < 1000) discount = 0.7;  //  (quantity < 1000) && ( quantity >= 800)
else discount = 0.6; //  (quantity >= 1000)
```

if-else进阶1：实际使用中，可使用等价形式

- 判别某个变量为 非0

```
if( a != 0 )
    printf("OK");
```

等价于:

```
if( a )
    printf("OK");
```

等价于:

```
if( !a == 0 )
    printf("OK");
```

- 判别某个变量为 0

```
if (a == 0)
    printf("OK");
```

等价于:

```
if( !a )
    printf("OK");
```

if-else进阶2：嵌套形式

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

- 分多支：

```
if (x>0) y= 1;
else if(x==0) y=0;
else y=-1;
```

- 嵌套：

```
if (x>0){
    y=1;
}
else{
    if (x==0){
        y=0;
    }
    else{
        y=-1;
    }
}
```

- 不致混淆时，{ }可缺省简化：

```
if (x>0)
    y=1;
else
    if (x==0)
        y=0;
    else
        y=-1;
```

if-else进阶2：嵌套形式

- 关键字 if 与 关键字 else 的匹配原则
  - { }缺省时，else子句总是与其之前最近的未配对的同层级if匹配
  - { }括起复合语句，在其内、外的语句分别属于不同的层级

- "我"认为的代码：

```
int a = 0, b = 0, c = 1;
if (a == b)
    if(a == c)
        printf("a = b = c");
else
    printf("a != b");
```

- "C"认为的代码：

```
int a = 0, b = 0, c = 1;
if (a == b){
    if(a == c)
        printf("a = b = c");
    else
        printf("a != b");
}
```

- 修正：

```
int a = 0, b = 0, c = 1;
if (a == b){
    if(a == c)
        printf("a = b = c");
}
else{
    printf("a != b");
}
```

if-else进阶2：嵌套形式

- 例：输入两数并判断其大小关系。

```
#include <stdio.h>
int main (){
    int x,y;
    printf("Enter integer x,y:");
    scanf("%d,%d",&x,&y);

    if(x!=y)
        if(x>y)  printf("X>Y");
        else     printf("X<Y");
    else
        printf("X=Y ");

    return 0;
}
```

- 运行：

```
Enter integer x,y:12,23
X<Y
Enter  integer  x,y:12,6
X>Y
Enter integer x,y:12,12
X=Y
```

if-else进阶2：嵌套形式

- 例：

  编写程序，输入x，根据以下函数输出y的值：

$$y=\begin{cases} x & (x<1) \\ 2x-1 & (1\le x<10) \\ 3x-11 & (x\ge 10) \end{cases}$$

- 简化：

```c
#include <stdio.h>
int main (){
    float x,y;
    scanf("%f",&x);

    if(x<1)
        y=x;
    else
        if (x<10 )
            y=2*x-1;
        else
            y=3*x-11;

    printf("x=%7.2f,y=%7.2f",x,y);

    return 0;
}
```

- 完整：

```c
#include <stdio.h>
int main (){
    float x,y;
    scanf("%f",&x);

    if(x<1){
        y=x;
    }
    else{
        if (x<10 ){
            y=2*x-1;
        }
        else{
            y=3*x-11;
        }
    }
    printf("x=%7.2f,y=%7.2f",x,y);

    return 0;
}
```

if-else进阶2：嵌套形式

- 例：根据输入日期判断是星期几，运用基姆拉尔森计算公式。
  - 基姆拉尔森计算公式：W=(d+2m+3(m+1)/5+y+y/4-y/100+y/400+1)%7
  - 公式中d表示日期中的日数，m表示月份数，y表示年数
  - 注意：在公式中，把一月和二月看成是上一年的十三月和十四月，例：如果是2004-1-10则换算成：2003-13-10来代入公式计算

```c
#include <stdio.h>
int main (){
    float x,y;
    scanf("%f",&x);

    //存放到访日期年、月、日
    int year=0,month=0,day=0;
    //存放基姆拉尔森计算公式计算结果
    int iweek;
    printf("\n请输入到访日期\n格式为年 月 日: 2017 1 1:");
    scanf("%d %d %d",&year,&month,&day);
    //闰年判断，如果是闰年二月份天数加1
    if(month==1||month==2) {
        month+=12;
        year--;
    }

    iweek=(day+2*month+3*(month+1)/5+year+year/4-year/100+year/400)%7;

    if(iweek=0)
        printf("星期一\n");
    else if(iweek==1)
       printf("星期二\n");
    else if(iweek==2)
       printf("星期三\n");
    else if(iweek==3)
       printf("星期四\n");
    else if(iweek==4)
       printf("星期五\n");
    else if(iweek==5)
       printf("星期六\n");
    else
       printf("星期日\n");

    return 0;
}
```

switch()语句

- 一般形式

```
switch(表达式){
    case   常量1:
                                语句组 1;
    case   常量2:
                                语句组 2;
    ……
    case   常量n:
                                语句组 n;
    [default:
                                语句组 n+1];
}
```

- 常用于从多个选项中选择一个
- 基于单个变量或者一个表达式的值进行选择
- 常量1、常量2、…、常量n是整型表达式，且值必须互不相同
- 通常使用int型数字和字符组成的表达式
- 语句组由若干语句组成，也可为空
- case后可包含多个可执行语句，且不必加{}

switch()语句

```
if(iweek=0)
   printf("星期一\n");
else if(iweek==1)
   printf("星期二\n");
else if(iweek==2)
   printf("星期三\n");
else if(iweek==3)
   printf("星期四\n");
else if(iweek==4)
   printf("星期五\n");
else if(iweek==5)
   printf("星期六\n");
else
   printf("星期日\n");
```

```
switch(iweek){
    case 0: printf("星期一\n"); break;
    case 1: printf("星期二\n"); break;
    case 2: printf("星期三\n"); break;
    case 3: printf("星期四\n"); break;
    case 4: printf("星期五\n"); break;
    case 5: printf("星期六\n"); break;
    case 6: printf("星期日\n"); break;
}
```

switch()语句

- 例：输入一个百分制成绩，输出相应的成绩

  等级：

  如果（成绩>=90)

  那么 打印 "A"

  如果（90>成绩>=80)

  那么 打印 "B"

  如果（80>成绩>=70)

  那么 打印 "C"

  如果（70>成绩>=60)

  那么 打印 "D"

  如果（成绩<60)

  那么 打印 "E"

```c
#include <stdio.h>
int main (){
    float score;
    sacnf(“%f”,&score);
    switch( (int)(score/10) ) {
        case 10:
        case 9: printf(“A\n”);break;
        case 8: printf(“B\n“); break;
        case 7: printf(“C\n“); break;
        case 6: printf(“D\n“); break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0: printf(“E\n“); break;
        default:printf(“error\n“);
    }
    return 0;
}
```

switch()语句

- 表达式的值依次与常量1、常量2、…、常量n比较，当发现某两个相等时便转去执行其后的语句组;
- 执行语句组时，逐条执行，如遇到break语句时，则结束switch语句;
- case子句的本质是地址标号：若已执行的语句组不含break语句，则自动进入一下条case子句对应的语句组继续执行（不管表达式的值是否与常量n的值相等）
- 开关语句的右花括号具有退出该开关语句的作用
- default语句可以省略，也可放在开关语句花括号内的任何位置

```
#include <stdio.h>
int main (){
    float score;
    sacnf(“%f”,&score);
    switch( (int)(score/10) ) {
        case 10:
        case 9: printf(“A\n”);break;
        case 8: printf(“B\n“); break;
        case 7: printf(“C\n“); break;
        case 6: printf(“D\n“); break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0: printf(“E\n“); break;
        default:printf(“error\n“);
    }
    return 0;
}
```

switch()语句

- 例：输入字母A~E表示的成绩等级，输出相应的百分制分数范围。
  'A'表示90--100
  'B'表示 80--89
  'C'表示70--79
  'D'表示60--69
  'E'表示60以下

```
#include <stdio.h>
int main (){
    char grade;
    sacnf(“%c”,&grade);
    switch( grade) {
        case 'A': printf("90---100");break;
        case 'B': printf("80---90");break;
        case 'C': printf("70---79");break;
        case 'D': printf("60---69");break;
        case 'E': printf("<60"); break;
        default : printf(“error“);
    }
    return 0;
}
```