中国人民大学
RENMIN UNIVERSITY OF CHINA

C程序设计

第4讲 控制语句

余力

buaayuli@ruc.edu.cn

内容提要

4.1 枚举逻辑思维

4.2 分支控制语句

4.3 循环控制语句

4.4 多重循环语句

中国人民大学
RENMIN UNIVERSITY OF CHINA

1. 枚举逻辑思维

先从一个例子说起……

- 某班级有四位同学中的一位做了好事，不留名，表扬信来了之后，班主任问这四位是谁做的好事。

  A说：不是我。

  B说：是C。

  C说：是D。

  D说：他胡说。

  已知三个人说的是真话，一个人说的是假话。现在要根据这些信息，找出做了好事的人。

- 题目如何求解？如何计算机求解？

问题1：如何建模四个人所说的话

char thisman=' '； // 定义字符变量并初始化为空

说话人	说的话	写成关系表达式
A	"不是我"	thisman!='A'
B	"是C"	thisman=='C'
C	"是D"	thisman=='D'
D	"他胡说"	thisman!='D'

问题2：如何确定可能的答案

枚举!

状态	赋值表达式
1	thisman='A'
2	thisman='B'
3	thisman='C'
4	thisman='D'

- 所谓枚举：按照者四种假定逐一地去测试四个人的话有几句是真话，

  如果不满足三句为真，就否定掉这一假定，换下一个状态再试。

（1）假定让thisman='A'代入四句话中

说话人	说的话	关系表达式	值
A	thisman!='A' ;	'A'!='A'	0
B	thisman=='C' ;	'A'=='C'	0
C	thisman=='D' ;	'A'=='D'	0
D	thisman!='D';	'A' !='D'	1

只有一个说的是真的，因此显然不是 ‘A’ 做的好事。

（2）假定让thisman='B'代入四句话中

说话人	说的话	关系表达式	值
A	thisman!=‘A’;	‘B’!=‘A’	1
B	thisman==‘C’;	‘B’==‘C’	0
C	thisman==‘D’;	‘B’==‘D’	0
D	thisman!=‘D’;	‘B’!=‘D’	1

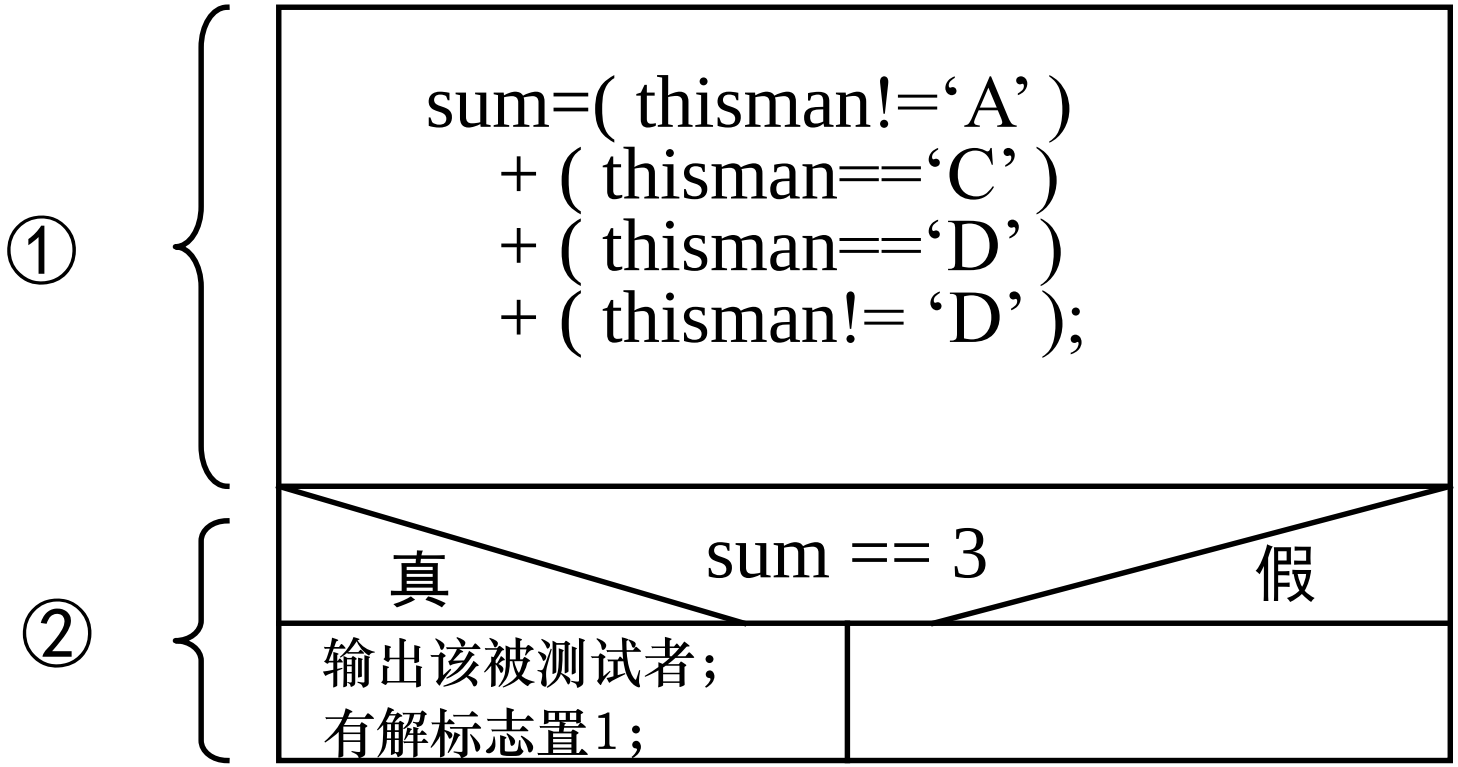
四个关系表达式的值的和为2，显然不是 ‘B’ 做的好事。

（3）假定让thisman='C'代入四句话中

说话人	说的话	关系表达式	值
A	thisman!=‘A’;	‘C’!=‘A’	1
B	thisman==‘C’;	‘C’==‘C’	1
C	thisman==‘D’;	‘C’==‘D’	0
D	thisman!=‘D’;	‘C’!=‘D’	1

四个关系表达式的值的和为3，就是 ‘C’ 做的好事。

逐个试！


①
sum=(thisman!='A')
+ (thisman=='C')
+ (thisman=='D')
+ (thisman!= 'D');
②
sum == 3
真
假
输出该被测试者;
有解标志置1;

总结：枚举思路

- 枚举思想
  - 完备考虑所有可能性(体现在代码中)
  - 逐一处理 (使用循环语句)
- 自然语言描述-->数学语言描述
  - 以实现其"可计算性"
  - 关系表达式与逻辑表达式
- 数学语言表达-->程序语言表达
  - 满足其"可行性",变成可执行的程序

#224 求区间*（1* 枚举搜索）
#217 空间位置搜索*（枚举）
#424 算数比赛（枚举）
#141 矩阵搜索【多重循环**】
#300 火车票（枚举、条件）
#131 火柴棒等式*【枚举、难+】
#650 砝码组合
#494 选择奖品（循环、枚举）
#147 教室排课*【枚举 3*】
#595 address*（4*）
#492 加法表达式（枚举、统计排序）
#192 哥德巴赫猜想*（枚举、循环 3*）
#183 一元三次方程求解（枚举3*）
#182 鹿死谁手*【3*枚举】

中国人民大学
RENMIN UNIVERSITY OF CHINA

2. 分支控制语句

如何判断三角形

```
int main() {
    int a, b, c;
    double p, area;
    scanf("%d%d%d", &a, &b, &c);
    if (a + b <= c || a + c <= b || b + c <= a)
        printf("不能构成三角形，请重新输入！ ");
    else {
        p = (a + b + c) / 2.0;
        area = sqrt(p * (p - a) * (p - b) * (p - c));
        printf("%.2f\n", area);
    }
    return 0;
}
```

```
int judge(int a,int b,int c)
{
    int t;
    if(a>b)
    {
        t=a;
        a=b;
        b=t;
    }
    if(b>c)
    {
        t=b;
        b=c;
        c=t;
    }

    if(a+b>c){ return 1; }
    else{ return 0; }
}
```

三角形面积

```
int main()
{
        int a, b, c;
        double s, q, sum;
        scanf("%d %d %d", &a, &b, &c);
        while(a<=0||b<=0||c<=0||a+b<=c||a+c<=b||b+c<=a)
        {
                printf("构成不了三角形，请重新输入：");
                scanf("%d %d %d", &a, &b, &c);
         }
         q = (a + b + c) / 2.0;
         sum = q * (q - a) * (q - b) * (q - c);
         s = sqrt(sum);
        printf("%.2lf", s);
        return 0;
}
```

Chp04_三角形面积.cpp

关系表达式

- 用关系运算符将两个表达式连接起来的式子，称关系表达式

运算符	说明
>	大于
>=	大于等于
<	小于
<=	小于等于
==	等于
!=	不等于

优先级相同（高）

优先级相同（低）

逻辑运算

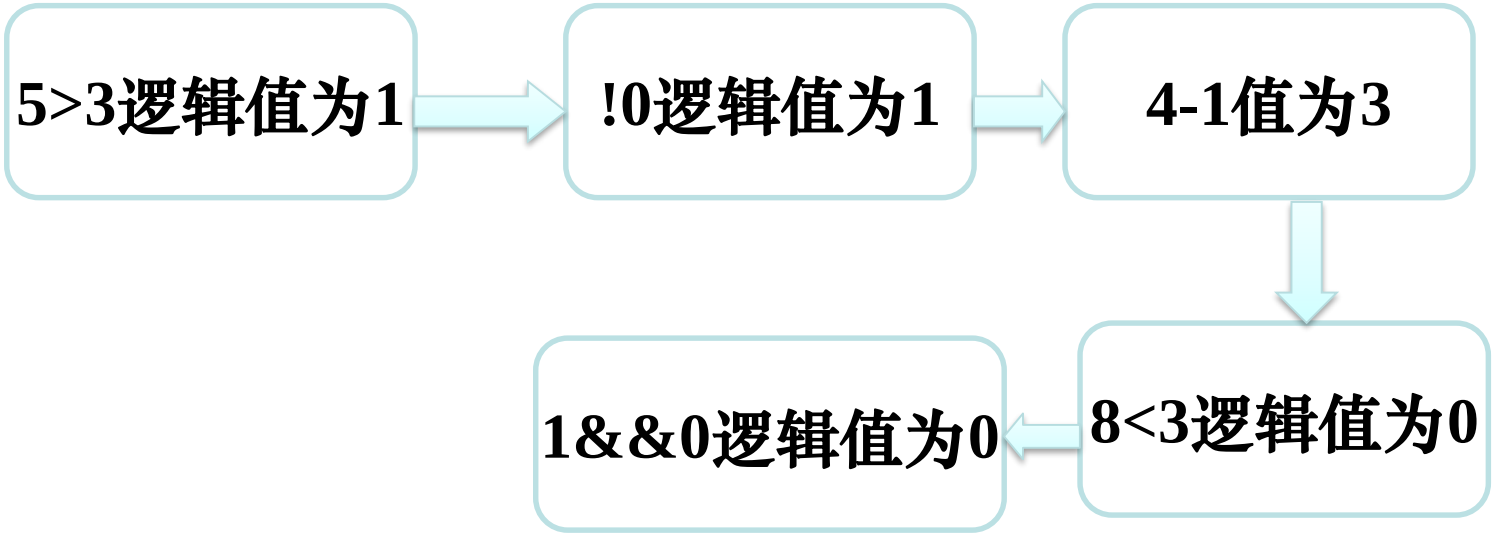
- 逻辑运算符如下：

运算符	说明
!	逻辑非（NOT）
&&	逻辑与（AND）
||	逻辑或（OR）

- !为单目运算符，&&和||为双目运算符，结合方向为从右至左。

逻辑表达式

- 用逻辑运算符将关系表达式或逻辑量连接起来的式子
- 求解： 5>3&&8<4-!0 的值



表达式值为0

条件运算符

- 条件表达式格式：
  - 表达式1 ？表达式2 : 表达式3
  - 例如：max = (a>b)? a : b ;

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

= (3>2)? 3 : 2

思考题（1）

- 1. 判断下面逻辑表达式的值
  - 100 > 3 && 'a'>'c'
  - 100 > 3 || 'a'>'c'
  - !(100>3)

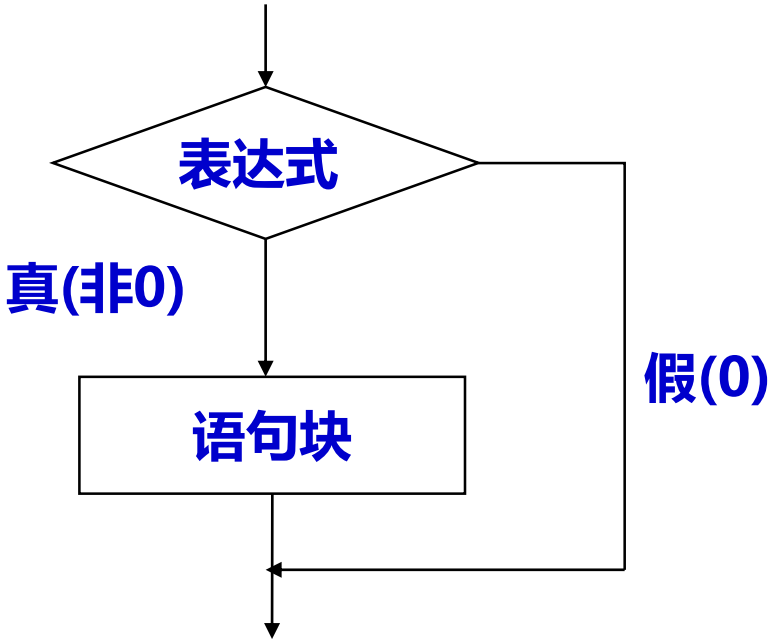
if (a>b) 与 if (“a>b”)

- 2. 写出下述条件的逻辑表达式
  - number is equal to or greater than 90 but smaller than 100.
  - ch is not a q or a k character.
  - number is not between 1 and 9.

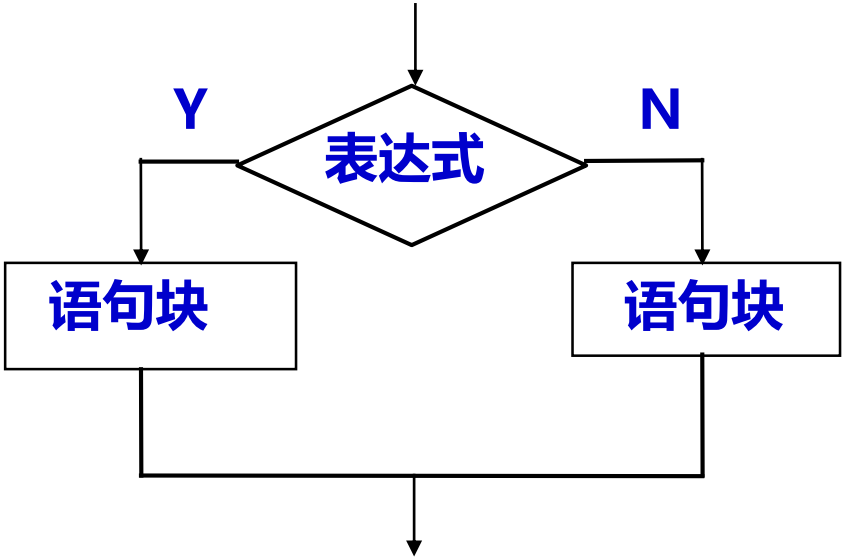
思考题 (2)

- 判断下面表达式的值
  - (5 > 2)+2
  - 3+4>2 && 3<2
  - x>=y || y>x
  - d = 5+(6>2)
  - 'X'>'T' ? 10 : 5
  - x>y ? y>x : x>y

如何表示分支结构




```
if （表达式）
    语句 1;
```




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

IF语句的嵌套和级联

if (条件1) <语句1>

else if (条件2) <语句2>

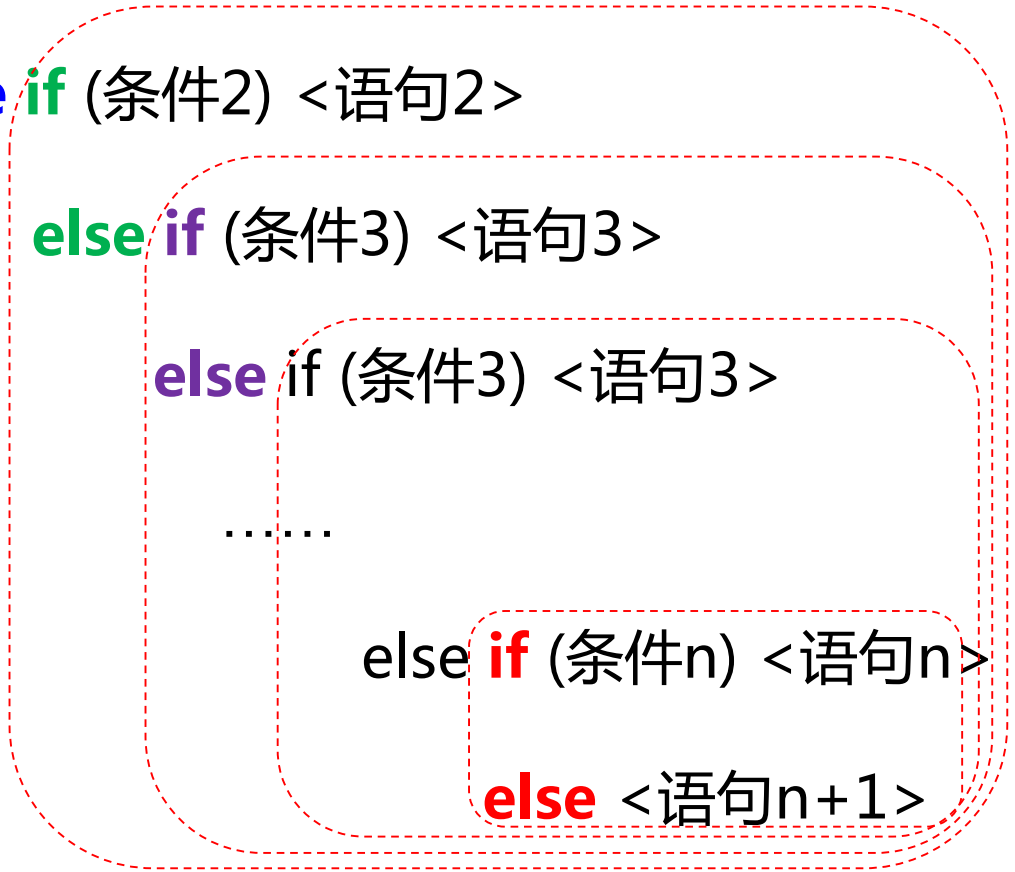
else if (条件3) <语句3>

else if (条件4) <语句4>

……

else if (条件n) <语句n>

else <语句n+1>



复合语句！

语句块

多条语句捆绑成一条语句！

条件满足后，要做多件事

```
int main() {
    int a, b, c;
    double p, area;
    scanf("%d%d%d", &a, &b, &c);
    if (a + b <= c || a + c <= b || b + c <= a)
        printf("不能构成三角形，请重新输入！ ");
    else {
        p = (a + b + c) / 2.0;
        area = sqrt(p * (p - a) * (p - b) * (p - c));
        printf("%.2f\n", area);
    }
    return 0;
}
```

```
if(a>b)
{
    t=a;
    a=b;
    b=t;
}
if(b>c)
{
    t=b;
    b=c;
    c=t;
}
```

```
int main() {
    int type, money;
    printf("数字1表示甜粽子，否则就是咸粽子\n");
    printf("请输入粽子口味和可支付金额：");
    scanf("%d,%d", &type, &money);
    if (type == 1) {
        if (money >= 5 && money < 10)
            printf("您可以吃到五元的甜粽子\n");
        else if (money >= 10)
            printf("您可以吃到十元的甜粽子\n");
        else
            printf("您不可以吃到甜粽子\n");
    } else {
        if (money >= 4 && money < 12)
            printf("您可以吃到四元的咸粽子\n");
        else if (money >= 12)
            printf("您可以吃到十二元的咸粽子\n");
        else
            printf("您不可以吃到咸粽子\n");
    }
    return 0;
}
```

Chp04_IF嵌套.cpp

二元一次方程

```
#include <stdio.h>
#include <math.h>
int main()
{
  double a,b,c,disc,x1,x2,realpart, imagpart;
  scanf("%lf,%lf,%lf",&a,&b,&c);
  printf("The equation ");
  if( fabs(a)<=1e-6 )
      printf("is not a quadratic\n");
```

scanf 输入double型，必须用%lf

printf 输出double型就用%f

实型不能用**if (a==0)**

```
else
  {disc=b*b-4*a*c;
   if(fabs(disc)<=1e-6)
       printf("has two equal roots:%8.4f\n", -b/(2*a));
   else
       if(disc>1e-6)
           {x1=(-b+sqrt(disc))/(2*a);
            x2=(-b-sqrt(disc))/(2*a);
            printf("has distinct real roots:%8.4f and %8.4f\n",x1,x2);
           }
        else
```

不能用if (disc==0)

Chp04_二次方程.cpp

```
        { realpart=-b/(2*a);
          imagpart=sqrt(-disc)/(2*a);
          printf(" has complex roots:\n");
          printf("%8.4f+%8.4fi\n", realpart, imagpart );
          printf("%8.4f-%8.4fi\n", realpart, imagpart );
        }
    }
  return 0;
}
```

```
1,2,1
The equation has two equal roots: -1.0000
```

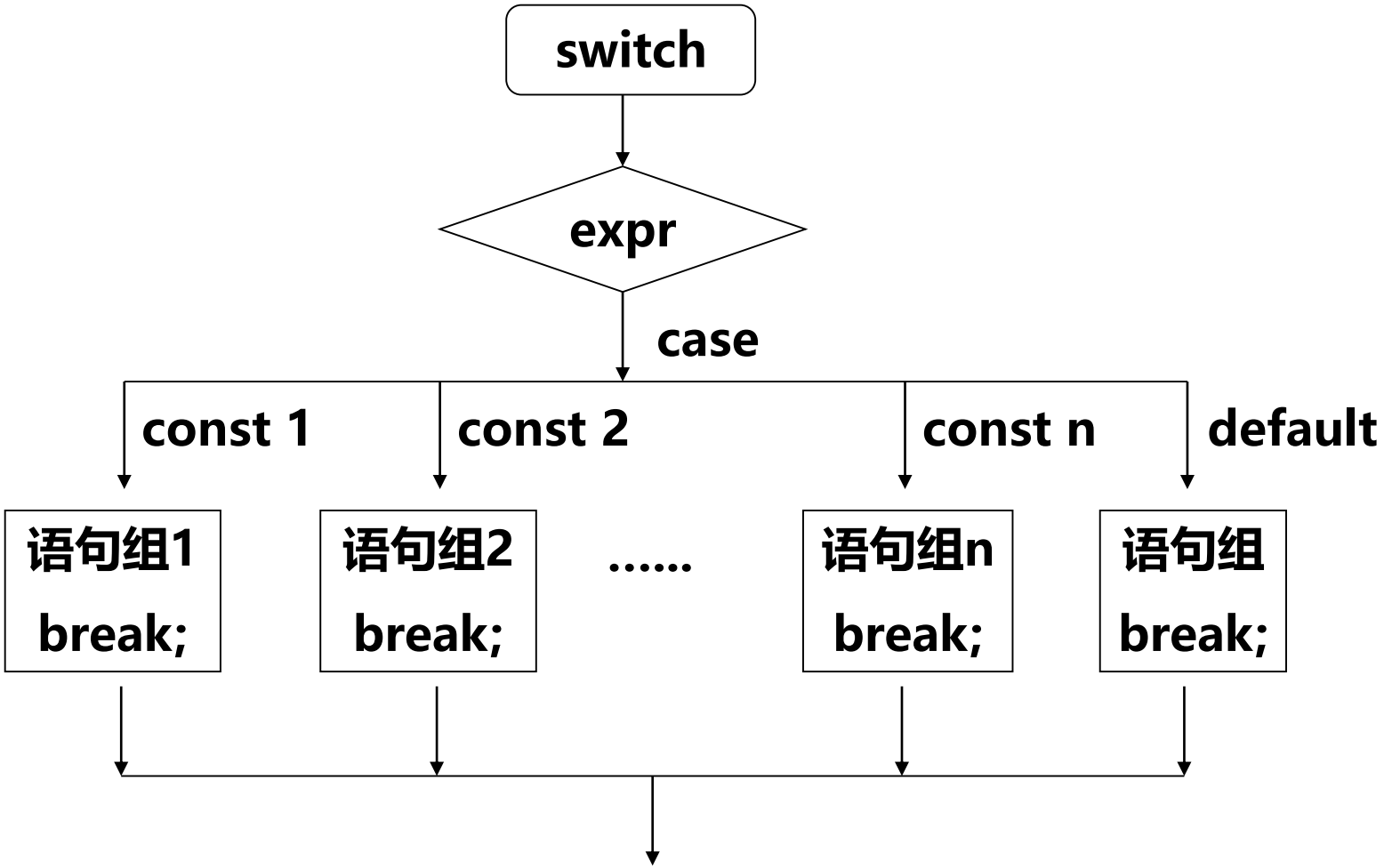
switch语句

- 格式：

```
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;
 }
```

- 执行过程：
  - 求出"表达式"的值，并执行与其相匹"常量"对应的语句。
  - 跟随case的语句将顺序执行，直到遇到break语句或case块结束；
  - 如果没有与之相匹的"常量"就执行default块，若default不存在，就退出switch语句。

switch结构：


switch
expr
case
const 1
const 2
const n
default
语句组1
break;
语句组2
break;
……
语句组n
break;
语句组
break;

运费问题

例 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
$250 \le s < 500$	2%折扣
$500 \le s < 1000$	5%折扣
$1000 \le s < 2000$	8%折扣
$2000 \le s < 3000$	10%折扣
$3000 \le s$	15%折扣

- 解题思路：
  - 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
  - 总运费f的计算公式为f=p×w×s×(1-d)
- 折扣的变化规律：
  - 折扣的"变化点"都是250的倍数
  - 在横轴上加一种坐标c，c的值为s/250
  - c代表250的倍数
  - 当c<1时，表示s<250，无折扣
  - 1≤c<2时，表示250≤s<500，折扣d=2%
  - 2≤c<4时，d=5%；4≤c<8时，d=8%；

    8≤c<12时，d=10%；c≥12时，d=15%

```
#include <stdio.h>
int main()
{
   int c,s;
   float p,w,d,f;
   scanf("%f,%f,%d", &p, &w, &s );
   if(s>=3000)  c=12;
   else         c=s/250;

   f = p*w*s*(1-d/100);
   printf("freight=%10.2f\n", f );
   return 0;
}
```

```
switch(c)
 { case 0:   d=0; break;
   case 1:   d=2; break;
   case 2:
   case 3:   d=5; break;
   case 4:
   case 5:
   case 6:
   case 7:   d=8; break;
   case 8:  case 9:  case 10:
   case 11:  d=10; break;
   case 12:  d=15; break;
 }
```

Chp04_Switch运费.cpp

switch结构的应用

接入热线	一级菜单	二级菜单
10086	1. 话费、套餐使用情况、归属地及积分查询	1. 话费查询
	(自动播报话费余额)	2. 套餐及上网流量查询
		3. 归属地查询
		4. 账单查询
		5. 积分查询
		6. 查询其他手机信息
	2. 最新优惠活动	
	3. 停复机、手机上网及密码服务	1. 停复机
		2. 手机上网流量查询
		3. 手机上网套餐办理
		4. 密码修改
		5. 密码重置
		6. 他机办理
	4. 增值业务查询与退订	
	5. 无线宽带、家庭业务	1. 无线宽带
		2. 家庭业务
	8. 集团业务	1. 话费、套餐等信息查询
		2. 故障申告
		3. 业务咨询和办理
		0. 人工服务
	0. 人工服务	

- 请使用switch结构，写一个10086电话语音服务的按键信息
- 接收用户按键，打印出提示语句

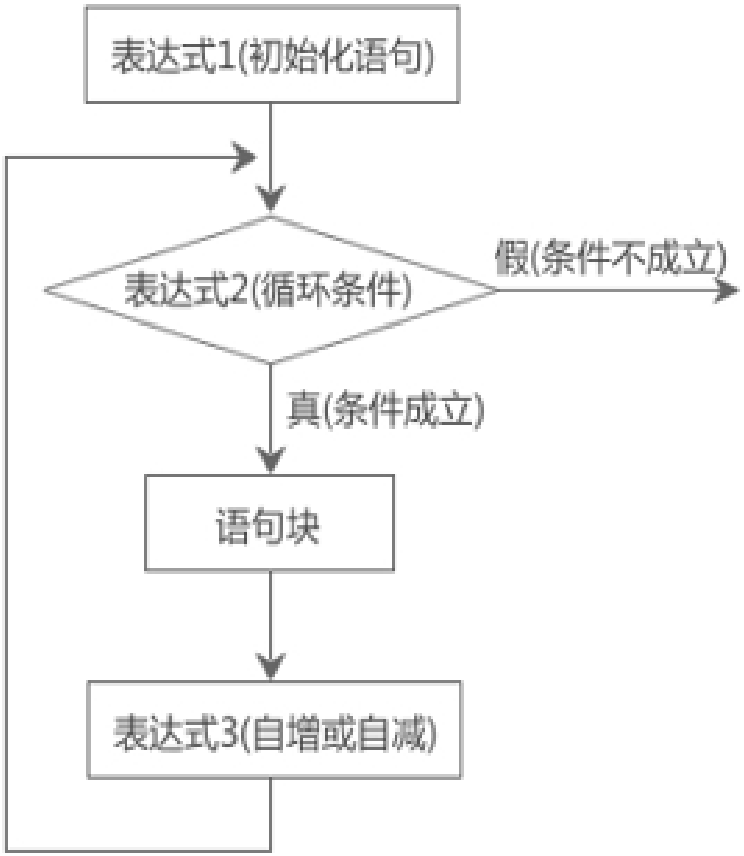
中国人民大学
RENMIN UNIVERSITY OF CHINA

3. 循环控制语句

for循环: 计数型循环

for(表达式1；表达式2；表达式3)
{
　循环体（语句组）
}

```
int main(){
int i, sum=0;
for(i=1,sum=0; i<=100; i++)
   sum+=i;
printf("%d\n",sum);
return 0;
}
```



Chp04_For循环求和.cpp

思考

题目：谁做了好事

逐个试!

怎么试?

```
sum =( thisman!='A' )
      + ( thisman=='C' )
      + ( thisman=='D' )
      + ( thisman !='D' );
```

```
#include <stdio.h>
int main()       {
int k=0,sum=0,find=0,count=0;
char thisman=' ';

for(k=0; k<4; k++) {
   thisman ='A'+k;
   sum =( thisman!='A' )
         + ( thisman=='C' )
         + ( thisman=='D' )
         + ( thisman !='D' );
   if (sum==3) {
     printf( "做好事者为%c\n", thisman );
     find=1;count++;}
  }

  if (find!=1)   printf( "Can't found!\n");
  return 0;
}
```

Chp04_谁做了好事.cpp

思考题

```
#include <stdio.h>
int main(void) {
  int secs;
  for (secs = 10;   ; secs--)
        { printf("%d seconds!\n", secs);
          if  !(--secs > 0) break;}
  printf("We have ignition!\n");
  return 0;
}
```

运行结果?

Chp04_倒计时.cpp

思考1：如果做到两秒两秒倒计时?

思考2：如何输出小写字母表?

课后练习：尝试for的表达式不同形式

数列求和

- **1 + 1/2 + 1/4 + 1/8 + 1/16 +....**

```
int main(void) {
  int n;   double sum, x;    int limit;
  printf("Enter number of terms you want: ");
  scanf("%d", &limit);
  sum = 0;
  for (x = 1, n = 1; n <= limit; n ++, x *= 2.0) {
        sum += 1.0 / x;
        printf("sum = %f when terms = %d.\n", sum, n);
  }
  return 0;
}
```

Chp04_数列求和.cpp

思考

如何实现

输入多个数字

直到按Q键

然后求其和

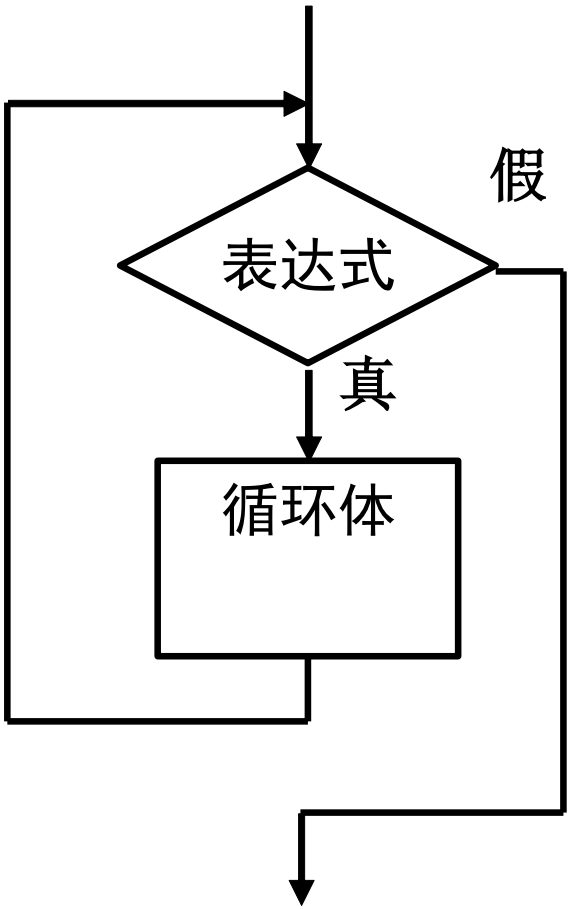
输入多个数字求和

```
int main(void) {
    long num;
    long sum = 0L;
    int status;

    printf("请输入一个整数(q to quit): ");
    status = scanf("%ld", &num);

    while (status == 1) {
        sum = sum + num;
        printf("num=%d\n", num);
        printf("请输入一个整数(q to quit): ");
        status = scanf("%ld", &num);
    }
    printf("输入数据的和是  %ld.\n", sum);
    return 0;
}
```

+Chp04_While输入求和.cpp

While循环

```
while ( 表达式 )
{
    语句块；(循环体)
}
```



思考：画出summing程序的流程图和N-S图

如何终止循环？

- 下面语句会产生什么结果？

```
index = 1;
while (index < 5)
    printf("Good morning!\n");
```

- 怎么终止循环？
  - while(index++ < 5)
  - break

死循环

求两个整数的最小公倍数

- 分析：
  - 假定有x ,y 且 x > y，设最小公倍数为 z，则：
    1) z 一定会 >= x
    2) z = k x， k= 1, 2, …
    3) z 一定会被 y 整除

```
#include <stdio.h>
int main(){
    int x=0, y=0, gbs=0, w=0;  // 整型变量
    {printf("请输入两个整数，用空格隔开："); // 提示信息}
    {scanf("%d", &x);                  // 键盘输入整数 x}
    scanf("%d", &y);                   // 键盘输入整数 y
    if ( x < y ) {                     // 让 x 表示两者中的大数
        w = x;   x = y;   y = w;       }
    max=x, min=y;
    gbs = max;                         // 将一个大数赋给gbs
    while ( gbs % min != 0 )// 当gbs不能被y整除时
        gbs = gbs + max;
    printf("最小公倍数为%d", gbs);         // 输出最小公倍数
    return 0;
}
```

Chp04_最公倍数.cpp

For与While的关系

```
for(i=0; (c=getchar())!='\n'; i+=c)
 ;

while((c=getchar())!='\n')
        printf("%c",  c+2);

for(      ; (c=getchar())!='\n';     )
        printf("%c",  c);
```

字符串处理

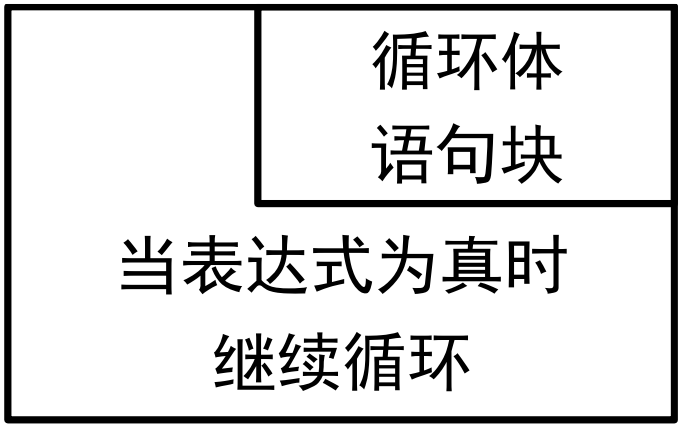
```
int main() {
    char ch;
    int count = 0;
    int a = 0, e = 0, i = 0, o = 0, u = 0;
    while ((ch = getchar()) != '\n') {
        if (ch == 'a' || ch == 'A')
            a ++;
        else if (ch == 'e' || ch == 'E')
            e ++;
        else if (ch == 'i' || ch == 'I')
            i ++;
        else if (ch == 'o' || ch == 'O')
            o ++;
        else if (ch == 'u' || ch == 'U')
            u ++;
        count ++;
    }
    printf("%.1f%%,%.1f%%,%.1f%%,%.1f%%,%.1f%%",
           (float) a * 100 / count,
           (float) e * 100 / count,
           (float) i * 100 / count,
           (float) o * 100 / count,
           (float) u * 100 / count
          );
    return 0;
}
```

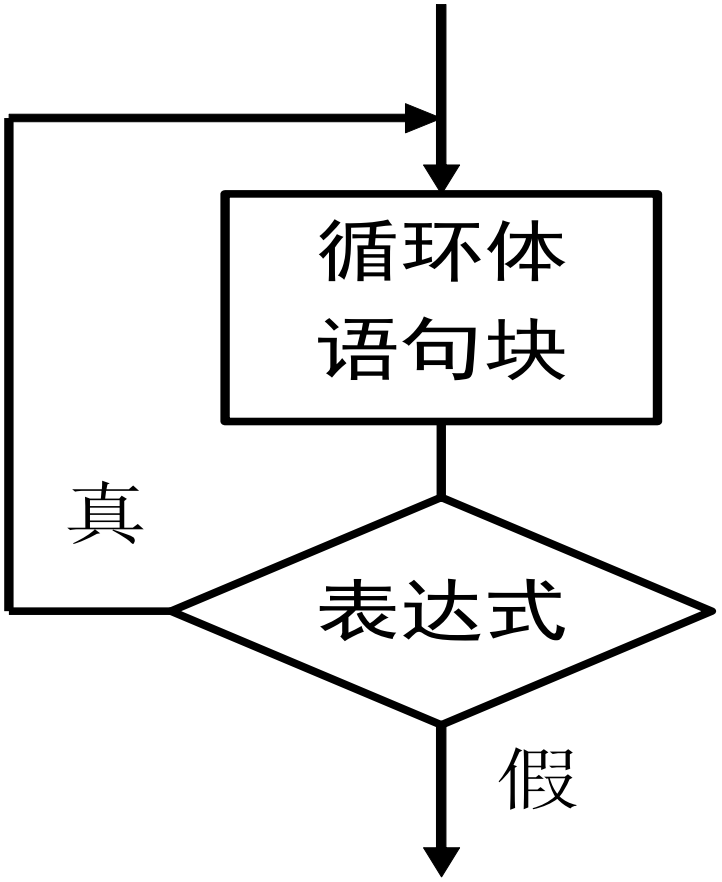
统计元音字母频率

Chp04_统计元音字母数.cpp

do ... while

```
do {
        循环体语句块;
} while ( 表达式 )
```





直到表达式为假时才退出循环，所以循环体至少执行一次。

- **猜数字：**
  - 让用户不断地猜一个数字，直到猜对为止

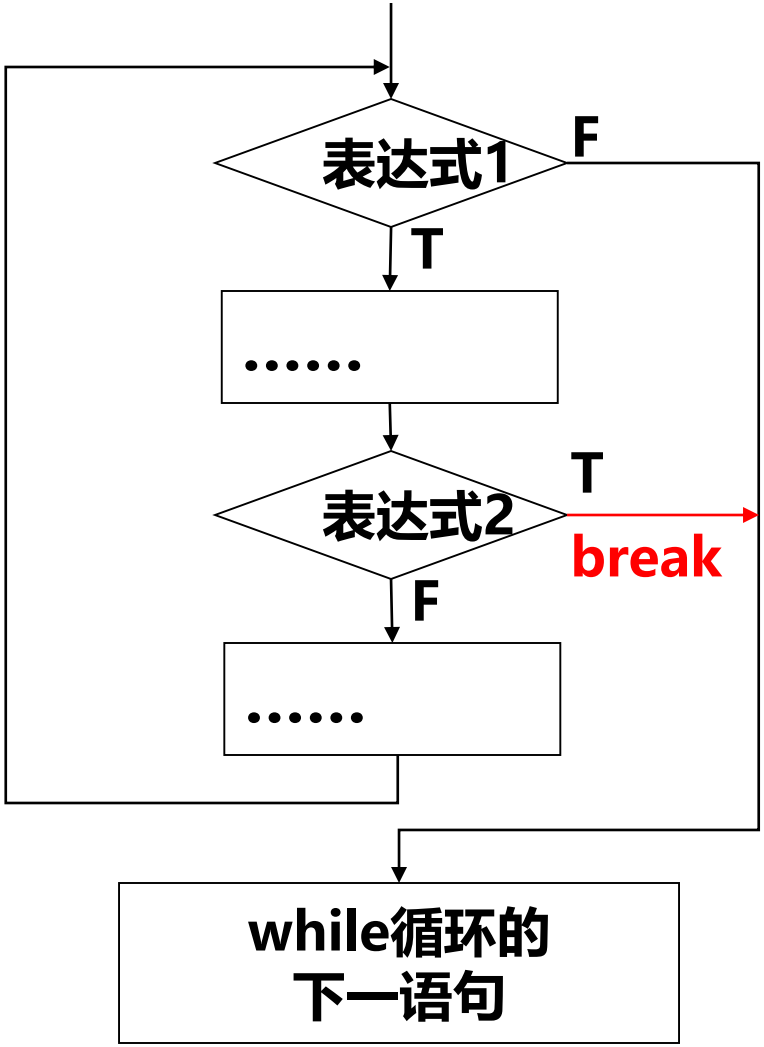
```
int num = 31, guess = 0;
do {
  printf ("Guess a number: ");
  scanf ("%d", &guess);
} while (guess != num);


printf ("Guess a number: ");
scanf ("%d", &guess);
while (guess != num) ;
  printf ("Guess a number: ");
  scanf ("%d", &guess);
```

break语句

- 格式：
  - break;
- 功能：
  - 结束最近的循环（do、for和while）或switch语句，把流程从该循环的内部跳到循环外部
- 思考：
  - 什么是最近的循环语句？

break语句

```
while (表达式1)
{
    ……
    if(表达式2)
        break;
    ……
}
```



捐款

```
#include <stdio.h>
#define SUM 100000
int main()
{  float amount, aver, total;   int i;

    for (i=1,total=0;i<=1000;i++)
        { printf("please enter amount:");
          scanf("%f",&amount);
          total= total+amount;
          //if (total>=SUM) break;
        }

   aver=total / i ;
   printf("num=%d\naver=%10.2f\n", i, aver);
   return 0;
}
```

捐款达到到一定数额停止捐款！

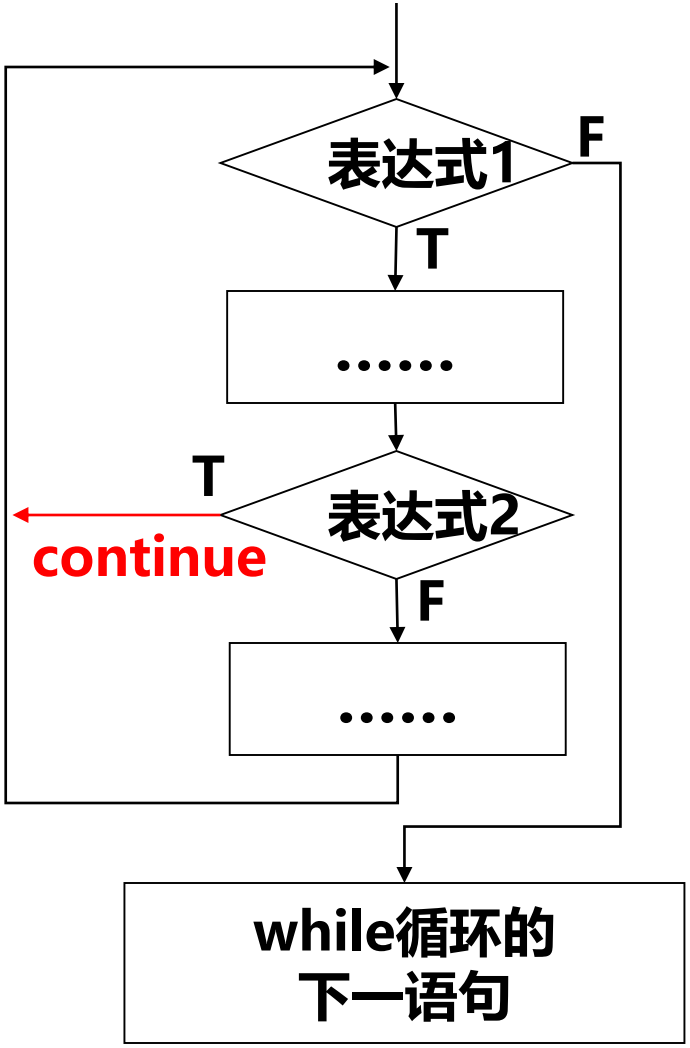
Chp04_捐款.cpp

continue语句

- 格式：continue;
- 功能：把控制转到最近的循环语句头,包括do、for和while
- 说明：
  - 中止本次循环，即不执行continue后的循环体中的语句。
  - continue与break的区别是：continue只结束本次循环，而不终止整个循环语句的执行；而break则是结束整个循环，不再进行条件判断。
  - continue不用在for、do和while以外的其它语句中

continue示意

```
while (表达式1)
{
    ……
    if(表达式2)
        continue;
    ……
}
```



示例

- 猜数字游戏
  - 编写一个猜数字的游戏，由编程人员设定一个100以内的奇数，让用户来猜，每次反馈猜的大了还是小了，直到猜对停止。如果用户输入了偶数，提醒用户输入奇数。

```
int main () {
  int num = 31, guess = 0;
  while (1) {
    printf ("Guess an odd number in [1,99]\n");
    scanf ("%d", &guess);
    if (guess % 2 == 0) {
        printf ("Please guess an ODD number\n");
        continue; }
    if (guess > num) printf ("Too big\n");
    else if (guess < num) printf ("Too small\n");
          else {
                printf ("Bingo! The number is %d", num);
                break;
              }
  }
  return 0;
}
```

Chp04_猜数字.cpp

更多循环

#185 迭代法求平方根

给定一个正整数a，用迭代法求出a的平方根并输出迭代次数。迭代公式如下：

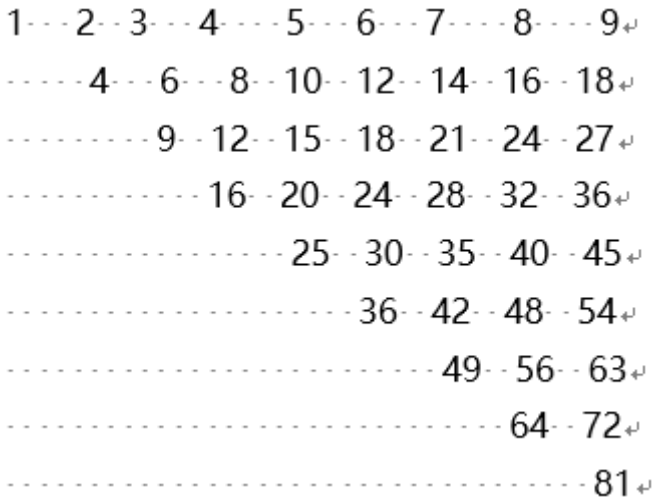
$$X[n+1]=1/2(X[n]+a/X[n])$$

要求前后两次求出的x的差绝对值小于10^-5，迭代初值取a/2。

#255 矩阵转置

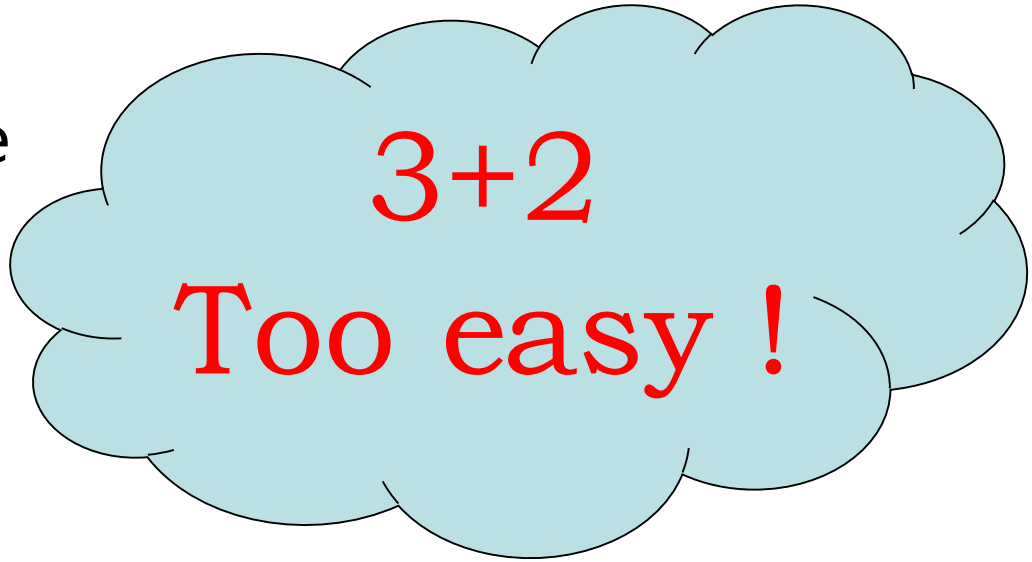
给定一个n×m矩阵相乘，求它的转置

#116 数字三角



循环语句总结

- 循环语句设计关键：
  - 循环条件
  - 循环体
- For, While，Do...while
- Break和Continue



中国人民大学
RENMIN UNIVERSITY OF CHINA

4. 多重循环语句

案例：找出作案人

- 某地刑侦队对涉及六个嫌疑人的一桩疑案进行案情分析：
  - A、B至少有一人作案；
  - A、D 不可能是同案犯；
  - A、E、F三人中至少有两人参与作案；
  - B、C或同时作案,或与本案无关；
  - C、D 中有且仅有一人作案；
  - 若 D 没参与，则 E 也不可能参与。
- 试编写程序将作案人找出来。

解题思路

- 核心思路

枚举法

- 比上例复杂的地方
  - "一位做了好事" VS 人人皆可犯案
  - "是"、"不是" VS "同案犯"、"有且仅有"、"若...则..."
- 总共有2^6 = 64种可能性
- 如何设计变量？
  - 设计6个01变量：1表示参与；0表示没参与

变量设计

- 六个人每个人都有作案或不作案两种可能,因此有64 (2^6) 种组合,从这些组合中挑出符合条件的作案者。

- 定义6个整型变量,分别表示六个人A - F
- 如何枚举每个人的可能性?
  - 取值 0 表示没有参与作案;
  - 取值 1 表示参与作案

案情建模

- 基于定义的变量，如何做案情分析？
  - A、B至少有一人作案;
  - A、D 不可能是同案犯;
  - A、E、F三人中至少有两人参与作案;
  - B、C或同时作案,或与本案无关;
  - C、D 中有且仅有一人作案;
  - 如果 D 没有参与作案,则 E 也不可能参与。

用逻辑表达式建模

对案情逐一建模 (1)

- 已知：变量A和B分别表示A和B作案
  - 怎么表示"A、B至少有一人作案"？
  - 表达式：**CC1 = A || B**
- 已知：变量A和D分别表示A和D作案
  - 怎么表示"A、D不可能是同案犯"？
  - 考虑相反事件：A和D是同案犯 → A && D
  - 表达式：**CC2 = !(A && D)**

对案情逐一建模 (2)

- 考虑三个变量A、E、F
  - 怎么表示"A、E、F三人中至少有两人参与作案"？
  - 可能有几种情况发生
    - 两个人A和E同时参与作案 → A && E
    - 两个人A和F同时参与作案 → A && F
    - 两个人E和F同时参与作案 → E && F
  - 表达式：**CC3 = (A&&E) || (A&&F) || (E&&F)**

对案情逐一建模 (3)

- 考虑两个变量B、C
  - 怎么表示"B、C或同时作案或同时与本案无关"？
  - 可能有几种情况发生
    - 两个人B和C同时参与作案 → B && C
    - 两个人A和F与本案无关 → !B && !C
  - 表达式：**CC4 = (B&&C) || (!B&&!C)**

对案情逐一建模 (4)

- 考虑两个变量C、D
  - 怎么表示"C、D中有且仅有一人作案"？
  - 可能有几种情况发生
    - 只有C作案，D没作案 → C && !D
    - 只有D作案，C没作案→ !C && D
  - 表达式：**CC5 = (C&&!D) || (!C&&D)**

对案情逐一建模 (5)

- 考虑两个变量D、E
  - 怎么表示"如果D没有参与作案，则E也不可能参与作案"？
  - 可能有几种情况发生
    - 如果D没有参与作案，E也没作案 → !D && !E
    - 如果D参与作案，则E既作案也没作案→ D
  - 表达式：**CC6 = (!D&&!E) || (D)**

案情建模

- A，B至少有一人作案 —— A||B
- A，E，F三人中至少有两人参与作案 —— (A&&E)||(A&&F)||(E&&F)
- A，D不可能是同案犯 —— !(A&&D)
- B，C或同时作案，或与本案无关 —— (B&&C)||(!B&&!C)
- C，D中有且仅有一人作案 —— (C&!D)||(!C&&D)
- 如果D没有参与作案，则E也不可能参与作案 ——(!D&&!E) || (D)

CC1+CC2+…+CC6==6

思路 ？

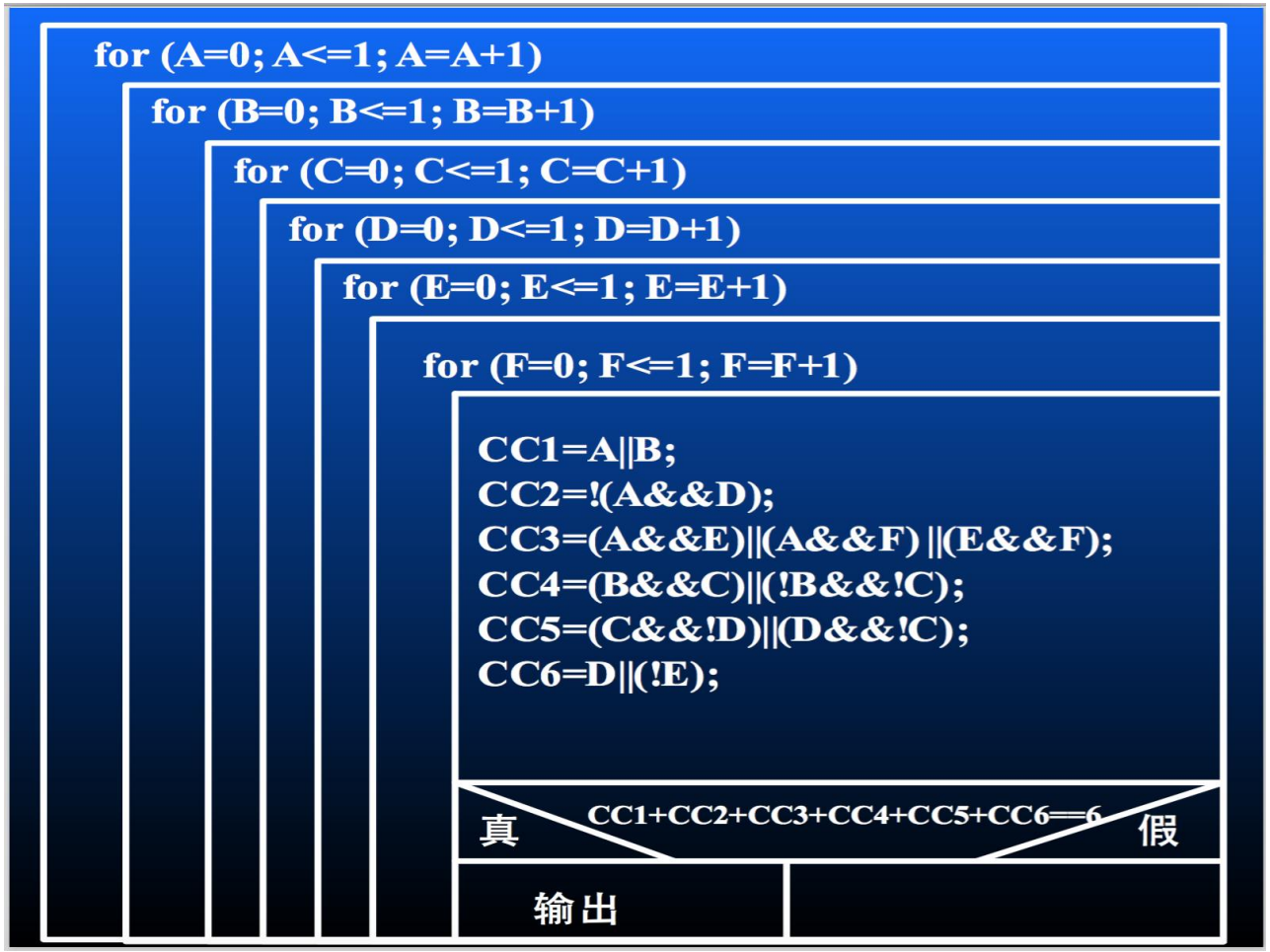
如何进行枚举呢?

- 方法1：沿用上例的单循环语句
  - for(k=0; k<4; k++)
- 方法2：

多重循环!

什么是多重循环？

- 编一个程序输出所有的排列数
  - 000000
  - 000001
  - 000010
  - 000011
  - ……
  - 111111



```cpp
int main () {
    cout << "例：找出作案人" << endl;
    for (int A = 0; A < 2; A++) {
        for (int B = 0; B < 2; B++) {
            for (int C = 0; C < 2; C++) {
                for (int D = 0; D < 2; D++) {
                    for (int E = 0; E < 2; E++) {
                        for (int F = 0; F < 2; F++) {
                            int CC1 = A || B;
                            int CC2 = !(A && D);
                            int CC3 = (A && E) || (A && F) || (E && F);
                            int CC4 = (B && C) || (!B && !C);
                            int CC5 = (C && !D) || (!C && D);
                            int CC6 = (!D && !E) || (D);
                            if (CC1 + CC2 + CC3 +
                                CC4 + CC5 + CC6 == 6) {
                                cout << A << B << C
                                    << D << E << F << endl;
                            }
                        }
                    }
                }
            }
        }
    }
    return 0;
}
```

Chp04_多人作案.cpp

循环的嵌套

```
int main(void) {
  const int ROWS = 6;
  const int CHARS = 6;
  int row;
  char ch;
  for (row = 0; row < ROWS; row++) {
      for (ch = ('A' + row); ch < ('A' + CHARS); ch++)
          printf("%c", ch);
      printf("\n"); }
  return 0;
}
```

Chp04_字母三角形.cpp

思考题

- 假设有8间教室，容量分别为C1 – C8。现考虑四个班级，人数分别为N1 – N4。如果班级人数小于等于教室容量，就可以把教室分配给该班级，共有多少种可行的分配方案

```
int classes[4];
printf("Input capacities of the 8 rooms (separated by space): ");
scanf("%d %d %d %d %d %d %d %d", &rooms[0], &rooms[1], &rooms[2], &rooms[3],
      &rooms[4], &rooms[5], &rooms[6], &rooms[7]);
printf("Input numbers of students in the 4 classes (separated by space): ");
scanf("%d %d %d %d", &classes[0], &classes[1], &classes[2], &classes[3]);
```

Chp04_分配教室.cpp

```
int solution = 0;
for (int i = 0; i < 8; i ++)  // assignment of the 1st class
    for (int j = 0; j < 8; j ++)
        for (int k = 0; k < 8; k ++)
            for (int l = 0; l < 8; l ++)
                if (classes[0] <= rooms[i] && classes[1] <= rooms[j] && classes[2] <= rooms[k] && classes[3] <= rooms[l] && (i != j)
                        && k != i && k != j && l != i && l != j && l != k) {
                    printf("Feasible assignment for the 4 classes: %d %d %d %d\n", i, j, k, l);
                    solution++;
                }

if (solution == 0)
    printf("No feasible assignment!\n");
else
    printf("Number of solutions =%d\n", solution);
```

思考题

```
for (int row = 0; row < line_num; row ++) {
        //print the spaces in the left
        for (int i = 0; i < line_num - row - 1; i ++)
            printf(" ");
        // print the letters in the middle
        for (int i = 0; i < row + 1; i ++)
            printf("%c", 'A' + i);
        for (int i = row - 1; i >= 0; i --)
            printf("%c", 'A' + i);
        printf("\n");
}
```

```
A
ABA
ABCBA
ABCDCBA
ABCDEDCBA
```

Chp04_字母金字塔.cpp

迭代

- 迭代：不断用新值取代变量旧值，由旧值递推出新值
- 例 兔子繁殖问题

  有一对兔子，从出生后第3个月起每个月都生一对兔子，小兔子长到第三个月后每个月又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少对？

  Fibonacci数列:

  $$fib_1 = fib_2 = 1$$

  $$fib_n = fib_{n-1} + fib_{n-2} \qquad (n \geq 3)$$

```
int main () {
    int n = 0;
    int fib, fib1, fib2;
    fib = fib1 = fib2 = 0;
    printf ("Enter the month num: ");
    scanf ("%d", &n);
    for (int i = 0; i < n; i ++) {
        if (i == 0 || i == 1)
            fib = fib1 = fib2 = 1;
        else {
            fib = fib1 + fib2;
            fib2 = fib1;
            fib1 = fib;
        }
    }
    printf ("%d", fib);
}
```

Chp04_ Fibonacci数列.cpp

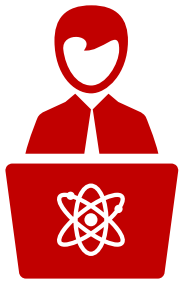
迭代

- 进制转换
  - 读入一串二进制数，将它转换成对应的10进制数

```
int main() {
    int num = 0;
    char ch;
    printf ("Enter a binary number: ");
    while ((ch = getchar()) != '\n') {
        num = num * 2 + (ch - '0');
    }
    printf ("%d", num);

}
```

Chp04_进制转换.cpp

中国人民大学
RENMIN UNIVERSITY OF CHINA



谢谢大家！