5-1. 一只猴子摘了N个桃子，第一天吃了一半又多吃了一个，第二天又吃了余下的一半又多吃了一个，到第十天的时候发现还有一个。

用倒推法计算，其基本思想是：从最后一天起逐次推出前一天的桃子数，直到推出第1天吃桃子前的全部桃子数为止。

根据题意，猴子某天吃桃子前的桃子数是前一天剩下桃子数的一半再减1，即：

$Peach_n=(Peach_{n-1})/2-1$ 或 $Peach_{n-1}=(Peach_n+1)*2$

这样就可以从第n天的桃子数倒推出第n-1天的桃子数（都是指该天未吃桃子前的数），从第n-1天的桃子数倒推出第n-2天的桃子数……一直可递推求出第1天的桃子数(最初值)。

反复的递推过程可以用循环结构实现。

位权

$$222.2=2\times10^{2}+2\times10^{1}+2\times10^{0}+2\times10^{-1}$$

位权

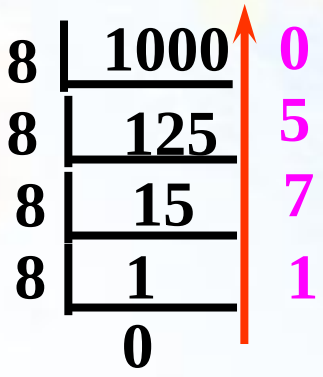
- 一个数字在某个固定位置所代表的值称为位权。
- 各进位制中位权的值恰好是基数的若干次幂。
- 以小数点为界，整数自右向左分别为0次方、1次方、2次方、…n-1次方，小数自左向右分别为-1次方、-2次方、-3次方、…-m次方。

实验5-2：十进制转八进制

$(1000)_{10}=(1750)_8$

除基数取余数，直到商为0

余数从下到上排列



把余数“当成”十进制数处理

位权10^0（1）

位权10^1

位权10^{i-1}

可设变量result=0×10^0+5×10^1+7×10^2+1×10^3 =1750

```
int n,r,w=1;  double result=0;
输入十进制数n
n==0
    r=n%8;      //短除法，除基取余
    result=result+r*w;    //位权公式
    w=w*10;     //位权
    n=n/8;      //每次处理过程中得到的商
输出result;
```

十进制转八进制算法NS流程图

int n,r,w=1;　double result=0;	
输入十进制数n	
n==0	r=n%8;　//短除法，除基取余
	result=result+r*w;　//位权公式
	w=w*10;　//位权
	n=n/8;　//每次处理过程中得到的商
输出result;	

八进制转十进制

$(1750)_8 = 1\times8^3+7\times8^2+5\times8^1+0\times8^0=(1000)_{10}$

位权法：各数位上的数字乘以其权值(基数的幂次)再累加

从右往左：

第一位(个位)：n%10

第二位(十位)：n/10%10

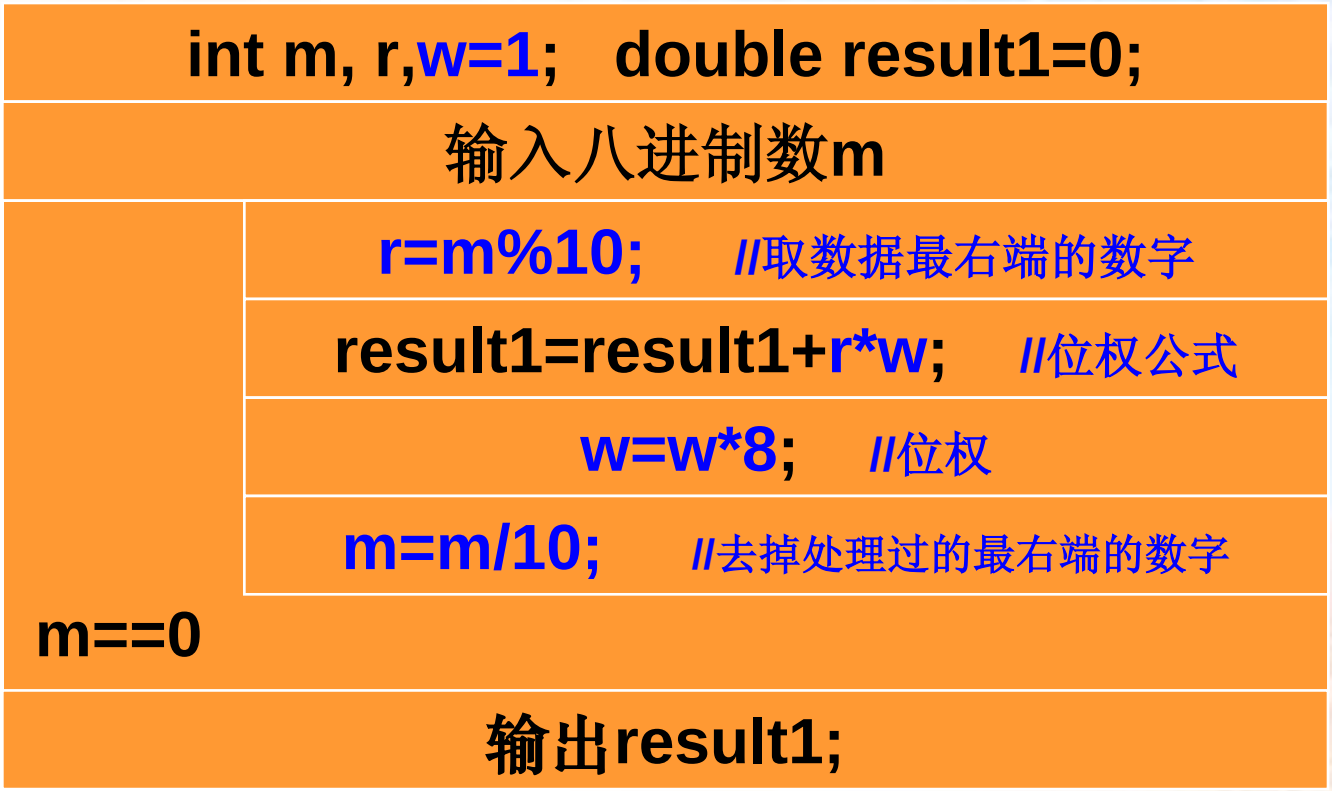
第三位(百位)：n/100%10

……

```
do
{
 num=n%10;
①利用位权公式计算累加和
②求下一个数据位数字的位权
 n=n/10;
}while(n!=0);
```

```
int m, r,w=1;  double result1=0;
输入八进制数m
m==0
    r=m%10;      //取数据最右端的数字
    result1=result1+r*w;    //位权公式
    w=w*8;    //位权
    m=m/10;     //去掉处理过的最右端的数字
输出result1;
```

八进制转十进制算法NS流程图

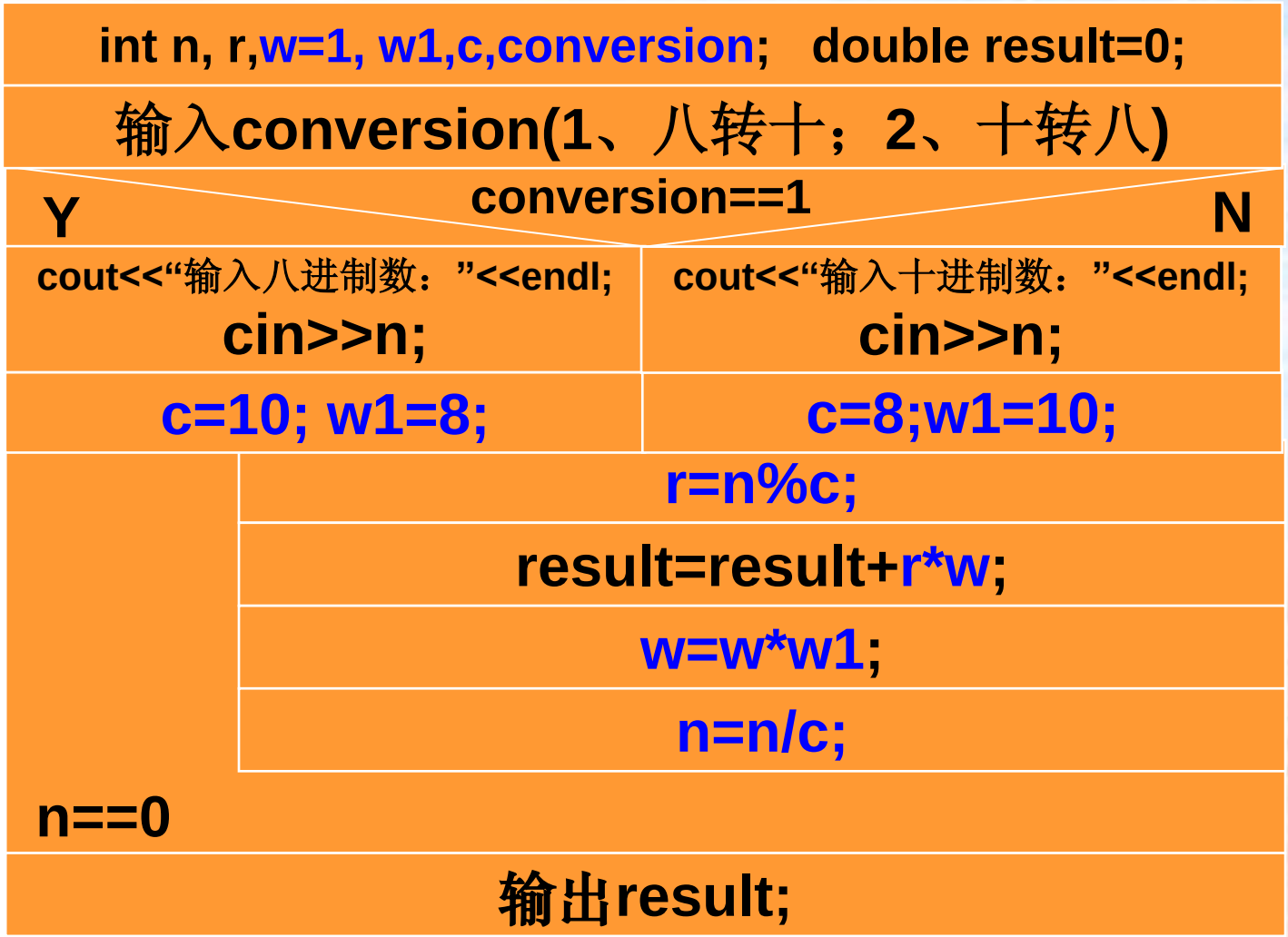


八进制↔十进制

```
int m, r,w=1;  double result1=0;
输入八进制数m
m==0
    r=m%10;      //取数据最右端的数字
    result1=result1+r*w;    //位权公式
    w=w*8;    //位权
    m=m/10;     //去掉处理过的最右端的数字
输出result1;
```

```
int n,r,w=1;  double result=0;
输入十进制数n
n==0
    r=n%8;     //短除法，除基取余
    result=result+r*w;    //位权公式
    w=w*10;    //位权
    n=n/8;     //每次处理过程中得到的商
输出result;
```

八进制↔十进制


int n, r,w=1, w1,c,conversion; double result=0;
输入conversion(1、八转十；2、十转八)
conversion==1
Y
N
cout<<“输入八进制数：”<<endl;
cin>>n;
cout<<“输入十进制数：”<<endl;
cin>>n;
c=10; w1=8;
c=8;w1=10;
r=n%c;
result=result+r*w;
w=w*w1;
n=n/c;
n==0
输出result;

实验5-3

- **问题：** 求两整数的最大公因数和最小公倍数
- **分析：** 假定两个整数分别为p和q，最大公约数应当是不超过其中较小数的一个整数。
- **实现方法：**
  - ➢ 穷举法

    ① i= p(或q)

    ② 若p，q能同时被i整除，则i即为最大公约数，结束。

    ③ 否则，i--，再回去执行②

  - ➢ 辗转相减法

    有两整数p和q：

    ① 若p > q，则p = p - q

    ② 若p < q，则q = q - p

    ③ 若p = q，则p （或q ）即为两数的最大公约数

    ④ 若p ≠ q，则再回去执行①

  - ➢ 辗转相除法

例如求9和6的最大公约数过程为：
9/6 &&6/6不能同时整除　9/5 &&6/5不能同时整除
9/4&&6/4不能同时整除　9/3&&6/3 能够同时整除
因此，3即为最大公约数

例如求27和15的最大公约数过程为：
27－15＝12(15>12)　15－12＝3(12>3)
12－3＝9(9>3)　9－3＝6(6>3)
6－3＝3(3==3)
因此，3即为最大公约数

实验5-3

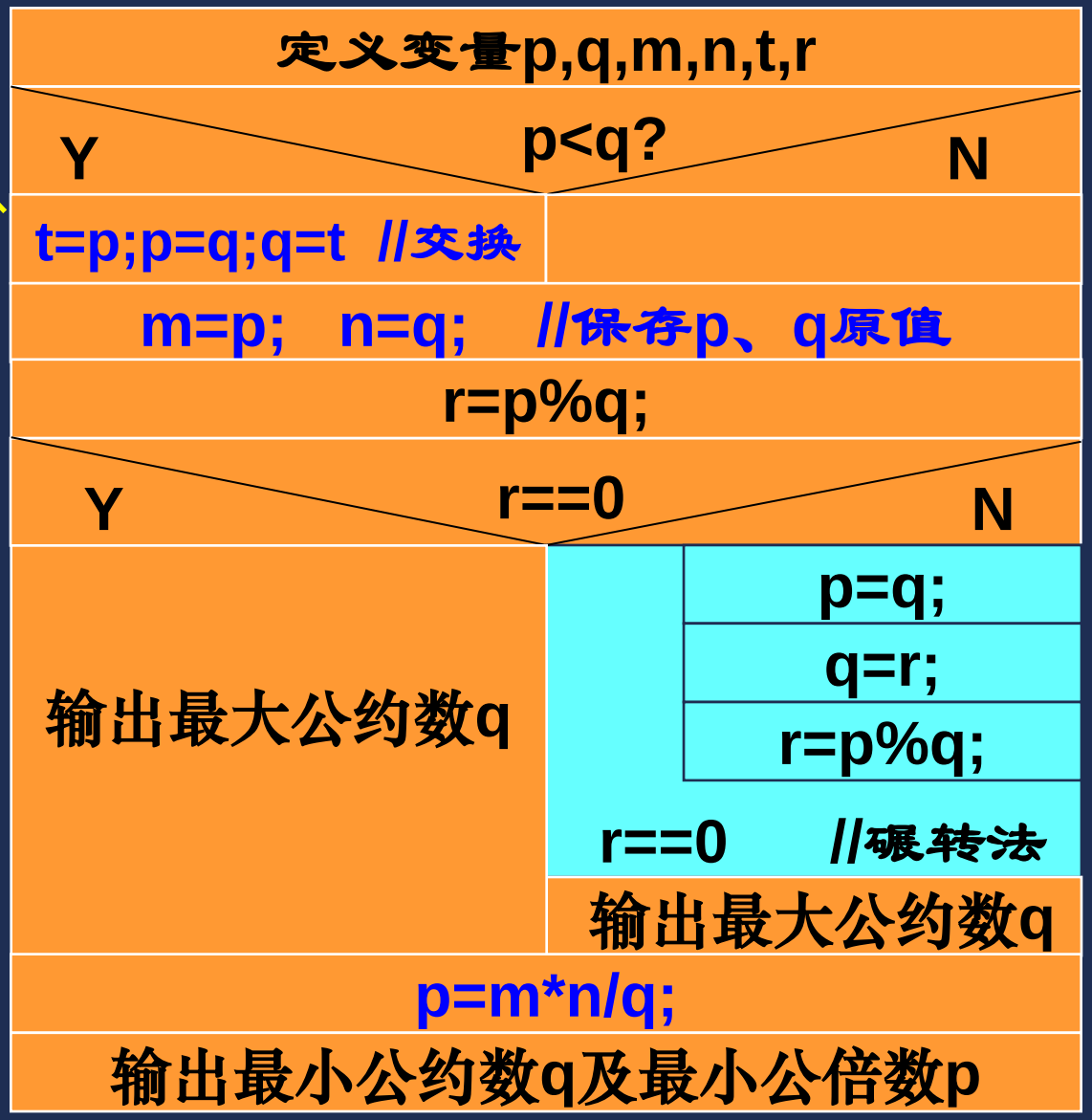
- **问题：** 求两整数的最大公因数和最小公倍数
- **辗转法：** 设p、q为输入的两个数（确保p>q，否则交换p、q的值），然后用p除以q，求出余数r，如果r==0，则当前q就是最大公约数；如果r!=0，则令p=q，q=r，重复以上相除的过程，直到r==0为止。

例如求27和15的最大公约数过程为：
27÷15 余12　15÷12余3　12÷3余0
因此，3即为最大公约数

① 确保p>q；
②p%q得余数r
③若r==0，则q即为两数的最大公因数
④若r!=0，则令p=q，q=r，再回去执行②

算法NS流程图

x=x+y;
y=x-y;
x=x-y;



实验5-4：从键盘输入一个角度值x(但计算必须转换成幅度值：y=x*PI/180)，求sinx的近似值，要求截断误差小于0.0001，即累加项(通项)的值小于0.0001即停止计算。近似计算公式如下：

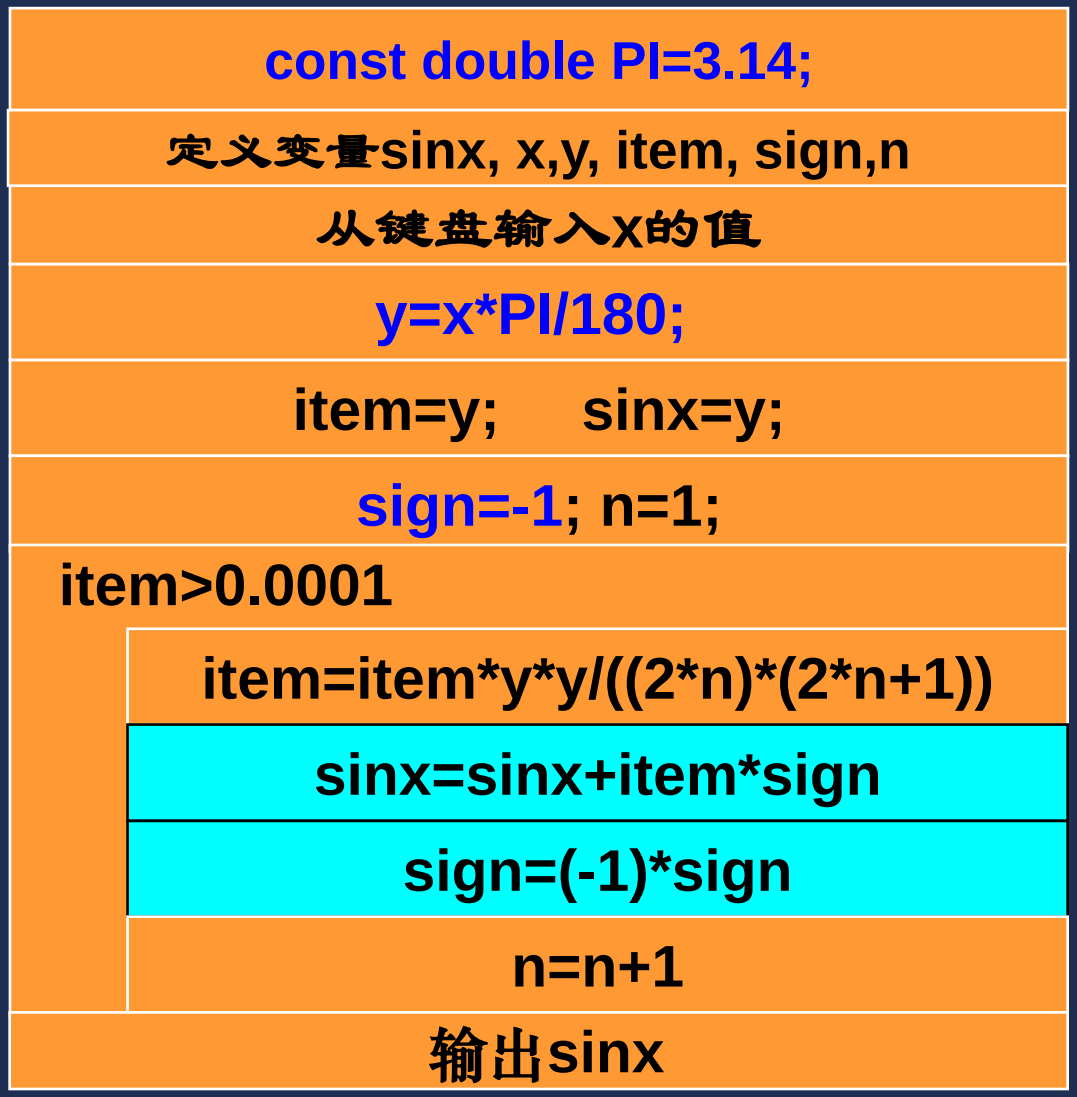
$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

$$\sin(x) = \frac{x}{1!} - \frac{x^1 * x * x}{1!*2*3} + \frac{x^3 * x * x}{3!*4*5} - \frac{x^5 * x * x}{5!*6*7} + \cdots$$

分析：这个近似计算可以看作一个累加过程，关键在于累加项数的确定。item保存第n项，则推出第n+1项的方法为：

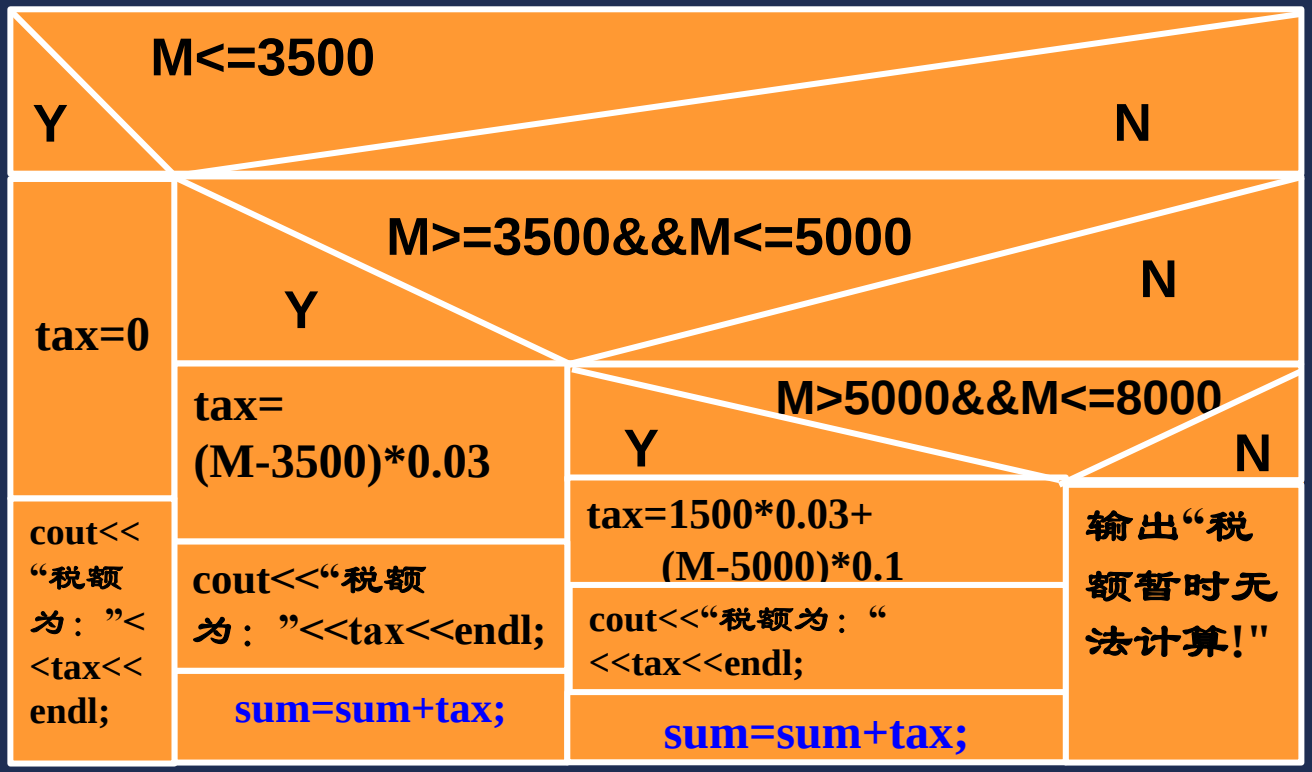
item=item*x*x/((2*n)*(2*n+1))

N-S流程图


const double PI=3.14;
定义变量sinx, x,y, item, sign,n
从键盘输入x的值
y=x*PI/180;
item=y; sinx=y;
sign=-1; n=1;
item>0.0001
item=item*y*y/((2*n)*(2*n+1))
sinx=sinx+item*sign
sign=(-1)*sign
n=n+1
输出sinx

实验5-5

- 由键盘输入若干职工的工资收入（以负数结束），输出每个职工的工资收入、计算并输出每个职工的缴税额，统计并输出职工总人数和总缴税额。
- 税收计算方法：个人应税起征点为月收入3500元；不超过起征点1500元的，超过部分按3%交纳个人所得税；超过起征点1500至4500元的，超过部分按10%交纳个人所得税。(参见实验3-5)

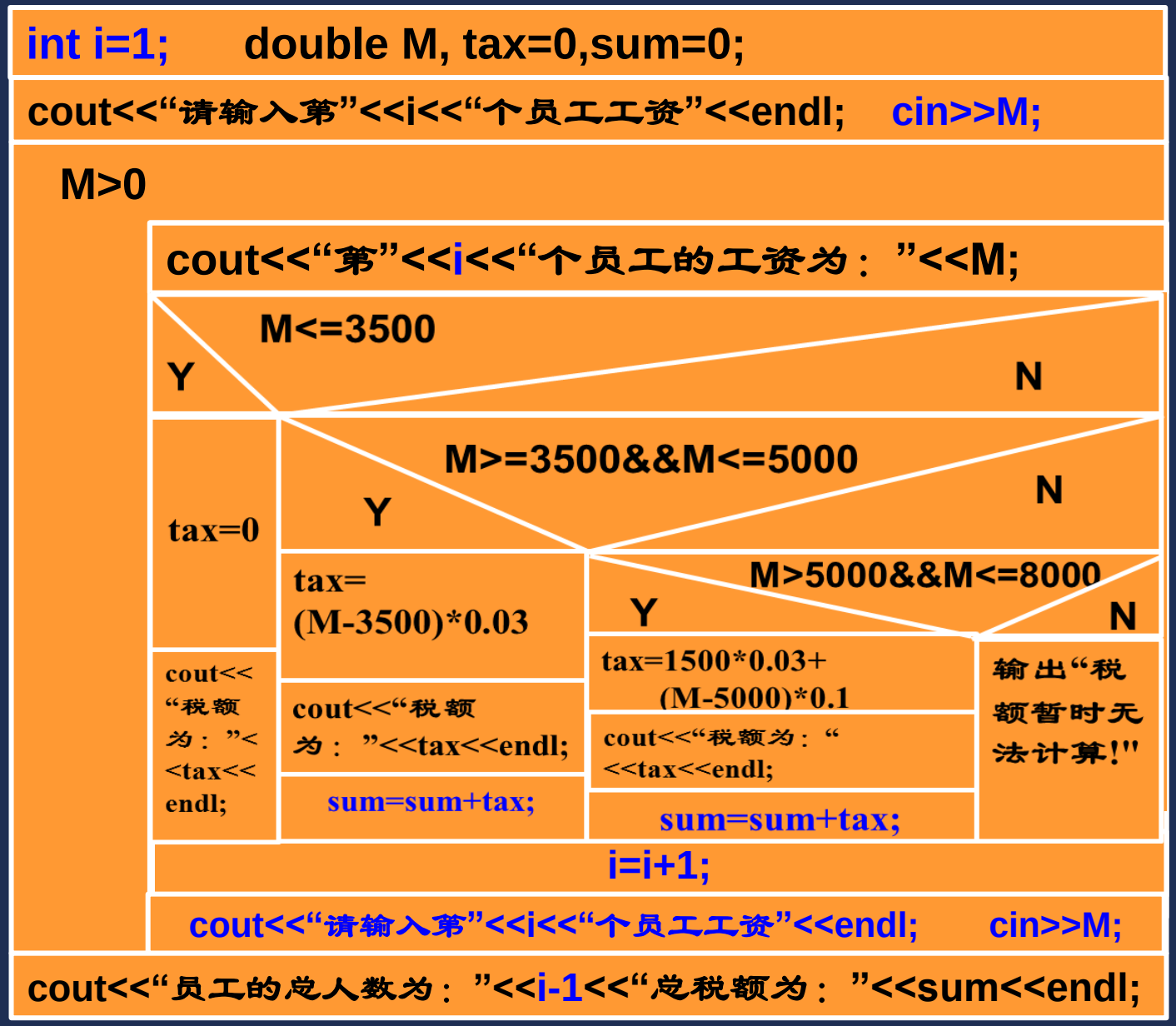


通过键盘输入的值控制循环

```
#include <iostream>
using namespace std;
void main( )
{
  int a;   //或者 char c;
  cout<<“请输入变量的值： ”<<endl;
  cin>>a;
  while(a!=0)   //  while(c!=‘N’ || c!=‘N’)
  {
    …..//任意功能代码，如猜拳、求税收等
    cout<<“是否继续:”<<endl;
    cin>>a;
  }
}
```

```
#include <iostream>
using namespace std;
void main( )
{
  int a;   //或者 char c;
  do     //至少执行一次循环体
  {
    …..//任意功能代码，如猜拳、求税收等
    cout<<“是否继续:”<<endl;
    cin>>a;
  }while(a!=0)   //  while(c!=‘N’ || c!=‘N’);
}
```

实验5-5


int i=1; double M, tax=0,sum=0;
cout<<“请输入第”<<i<<“个员工工资”<<endl; cin>>M;
M>0
cout<<“第”<<i<<“个员工的工资为：”<<M;
M<=3500
Y
N
tax=0
cout<<“税额为：”<<tax<<endl;
M>=3500&&M<=5000
Y
N
tax=(M-3500)*0.03
cout<<“税额为：”<<tax<<endl;
sum=sum+tax;
M>5000&&M<=8000
Y
N
tax=1500*0.03+(M-5000)*0.1
cout<<“税额为：“<<tax<<endl;
sum=sum+tax;
输出“税额暂时无法计算!"
i=i+1;
cout<<“请输入第”<<i<<“个员工工资”<<endl; cin>>M;
cout<<“员工的总人数为：”<<i-1<<“总税额为：”<<sum<<endl;