9.3-循环结构（for语句）

循环结构：根据一定的条件控制一段程序重复执行若干次

构成有效循环的两个要素：循环条件（什么时候结束）、循环体（要重复执行什么）

三种循环语句：for循环、while循环和do-while循环

**语法：**

for (表达式1;表达式2;表达式3)

{循环体}

表达式1：（声明、）初始化变量【说一下在不同地方声明的区别】；只在循环开始执行1次。这个变量可以和控制循环的变量无关。

表达式2：循环条件，一般是具有逻辑值的表达式，判定是否继续循环；

在每次执行循环体前执行，如为真继续执行循环体

表达式3：循环变量调整，使得最终结束循环；也可以与控制循环的变量无关，

但要保证循环能够正常结束

执行完循环体后进行

循环体：即反复执行的语句或语句块(复合语句)

**执行过程：**

(1)求解表达式1

(2)求解表达式2，若值为真(非0)，执行循环体体，然后执行第(3)步；

若为假(0)，则结束循环，转至第(5)步

(3)求解表达式E3

(4)转回步骤(2)继续执行

(5)循环结束，执行for语句下面的一个语句

代码示例1：

求n!的值。（n<=10）

int i;

int n=1,m;

cin>>m;

for(i=1;i<=m;i++)

{

n\*=i;

cout<<"i="<<i<<" n="<<n<<endl;

}

cout<<"i="<<i<<" n="<<n<<endl;

三个表达式可以省略的情况：

表达式1可以省略，也就是将初始化的工作放到for循环之前完成；但注意分号不能省略，例如：for(;i<=m;i++)

表达式2也可以省略，但没有意义，因为循环条件为空，永远判定为真值，循环无终止。

表达式3也可以省略，但仍需保证循环正常结束，一般将表达式3写在循环体的最后一句。例如：

int i;

int n=1,m;

cin>>m;

for(i=1;i<=m;)

{

n\*=i;

cout<<"i="<<i<<" n="<<n<<endl;

i++；

}

cout<<"i="<<i<<" n="<<n<<endl;

同样地，也可以同时省略多个，例如：

int i=1;

int n=1,m;

cin>>m;

for(;i<=m;) //while(i<=m)

{

n\*=i;

cout<<"i="<<i<<" n="<<n<<endl;

i++;

}

cout<<"i="<<i<<" n="<<n<<endl;

同样可以实现求阶乘的运算。

省略了“表达式1”和“表达式3”的情况下，可以用while和do-wihle语句代替

示例代码2：

输入一个整数n，之后按顺序输出斐波那契数列的前n项，并求出前n项的和。

int i;

int n;

long long f1=0,f2=1,f3;

long long sum;

cin>>n;

if(n==1)

{

cout<<"斐波那契数列的第一项为:"<<f1<<endl;

cout<<"斐波那契数列前一项的和为:"<<f1<<endl;

}

if(n==2)

{

cout<<"斐波那契数列的前两项为:"<<f1<<" "<<f2<<endl;

cout<<"斐波那契数列前一项的和为:"<<f1+f2<<endl;

}

if(n>2)

{

cout<<"斐波那契数列的前"<<n<<"项为:"<<f1<<" "<<f2<<" ";

sum=f1+f2;

for(i=3;i<=n;i++)

{

f3=f1+f2; //具体分析一下

f1=f2;

f2=f3;

cout<<f3<<" ";

sum+=f3;

}

cout<<endl;

cout<<"斐波那契数列前"<<n<<"项的和为:"<<sum<<endl;

}