9.2-do-while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：

do{ 循环体 }

while(循环条件表达式)

本质：**先无条件地执行循环体一次**，然后判定循环条件表达式的结果：判定结果为true（非0值），就继续重复执行循环体中的所有语句；直到循环条件为false（0）时，跳出循环

执行过程：

(1)执行循环体语句

(2)求解循环条件表达式 若值为真(非0)，转至第(1)步；

若值为假(0)，则结束循环

(3)执行do…while语句下面的一个语句

代码示例1：

求n!的值。（n<=10）

long long m=1;

int i,n=1;

cin>>i;

do

{ m\*=n;

n++;

}

while(n<=i);

/\*

m\*=n;

n++;

if(n<=i)

{ m\*=n;

n++;

}

重复执行了i-1次（当i=0时，执行了1次）

\*/

cout<<m<<endl;

cout<<n<<endl;

long long m=1;

int i,n=1;

cin>>i;

while(n<=i)

{ m\*=n;

n++;

}

cout<<m<<endl;

cout<<n<<endl;

代码示例2：

给出一个正整数 n(n≤100)，然后对这个数字一直进行下面的操作：如果这个数字是奇数，那么将其乘 3 再加 1，否则除以 2。经过若干次循环后，最终都会回到 1。经过验证很大的数字（7\*10^11）都可以按照这样的方式比变成 1，所以被称为“冰雹猜想”。例如当 n 是 20，变化的过程是 [20, 10, 5, 16, 8, 4, 2, 1]。

根据给定的数字，验证这个猜想，并输出整个变化序列。

int m;

cin>>m;

cout<<m<<" ";

while(m!=1)

{

if(m%2) //m%2为非0值，说明m是奇数

{

m=m\*3+1;

}

else{ m/=2; }

cout<<m<<" ";

}

cout<<endl;

int m;

cin>>m;

cout<<m<<" "; //输入m=1时有问题

do {

if(m%2) //m%2为非0值，说明m是奇数

{

m=m\*3+1;

}

else{ m/=2; }

cout<<m<<" ";

}while(m!=1);

cout<<endl;

总结：

while循环结构可以和do-while结构相互转换

在一般情况下，用while语句和用do-while语句处理同一问题时，若二者的循环体是一样的，它们的结果也一样

但是如果while后面的表达式一开始就为假(0值)时，两种循环的结果可能是不同的（因为无论while后的条件表达式结果如何，do后面的语句必然执行一次）