

第5章 循环结构程序设计

5.1 为什么需要循环控制

5.2 用**while**语句实现循环

5.3 用**do-while**语句实现循环

5.4 用**for** 语句实现循环

5.5 循环的嵌套

5.6 几种循环的比较

5.7 改变循环执行的状态

5.8 循环程序举例

5.1为什么需要循环控制

- 在日常生活中或是在程序所处理的问题中常常遇到需要重复处理的问题
 - ◆ 要向计算机输入全班**50**个学生的成绩
 - ◆ 分别统计全班**50**个学生的平均成绩
 - ◆ 求**30**个整数之和
 - ◆ 教师检查**30**个学生的成绩是否及格

5.1 为什么需要循环控制

- 例如：全班有**50**个学生，统计各学生三门课的平均成绩。

输入学生**1**的三门课成绩，并计算平均值后输出

```
scanf("%f,%f,%f",&s1,&s2,&s3);  
aver=(s1+s2+s3)/3;  
printf("aver=%7.2f",aver);
```

输入学生**2**的三门课成绩，并计算平均值后输出

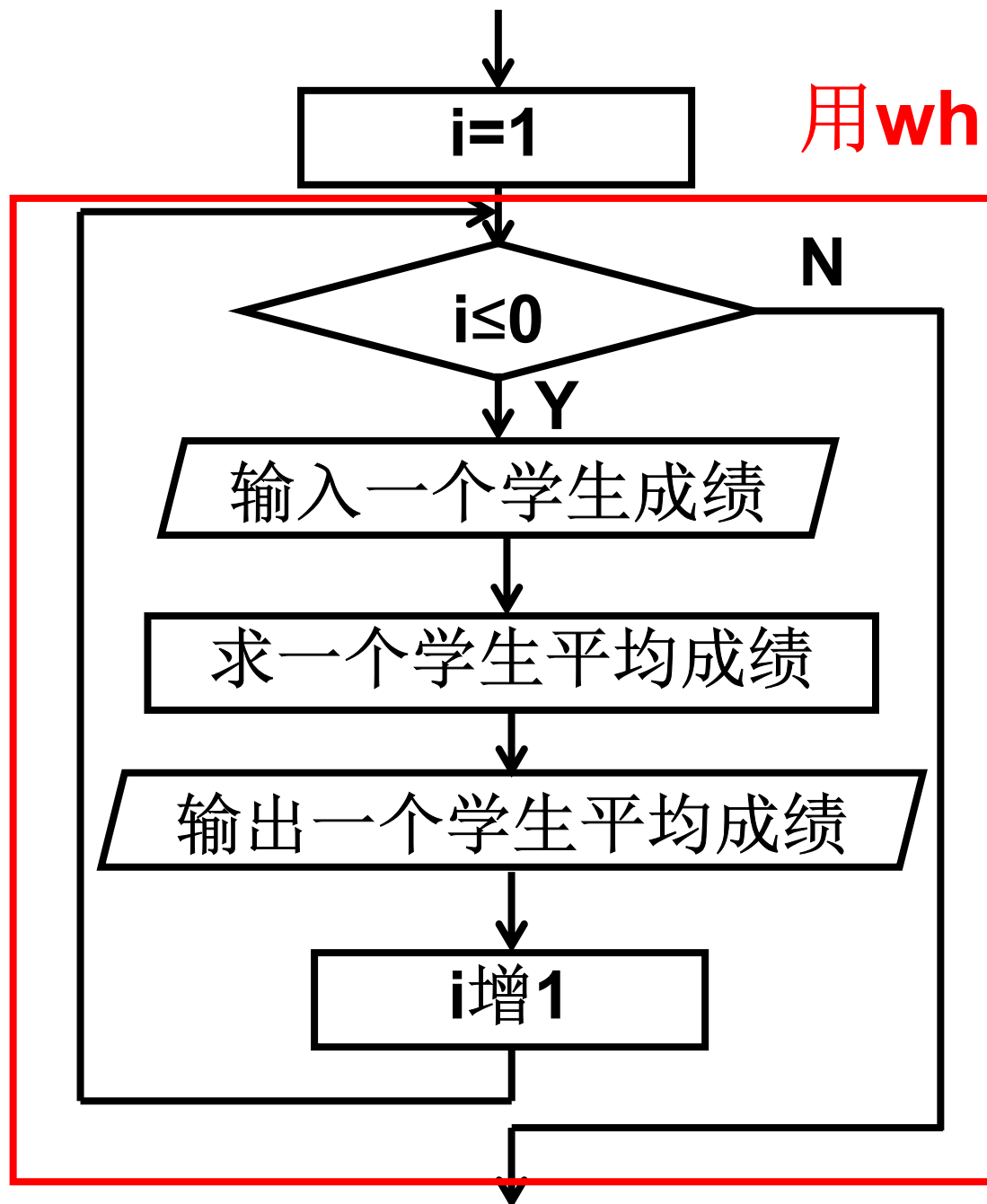
```
scanf("%f,%f,%f",&s1,&s2,&s3);  
aver=(s1+s2+s3)/3;  
printf("aver=%7.2f",aver);
```

要对**50**个学生进行相同操作 重复**50**次

- 大多数的应用程序都会包含循环结构
- 循环结构和顺序结构、选择结构是结构化程序设计的**三种基本结构**，它们是各种复杂程序的基本构造单元

5.2用while语句实现循环

- 全班有**50**个学生，统计各学生三门课的平均成绩。



用**while**循环结构实现

```
while( $i \leq 50$ )  
{  
    scanf..... ;  
    aver=..... ;  
    printf..... ;  
    i++;  
}
```

while语句的一般形式如下：

while (表达式) 语句

循环体

while语句的一般形式如下：

while (**表达式**) 语句

循环条件表达式

“真”时执行循环体语句
“假”时不执行

while循环的特点是：

先判断条件表达式，后执行循环体语句

例**5.1**求 **$1+2+3+\dots+100$** ，即 $\sum_{n=1}^{100} n$

➤解题思路：

- ◆这是累加问题，需要先后将**100**个数相加
- ◆要重复**100**次加法运算，可用循环实现
- ◆后一个数是前一个数加**1**而得
- ◆加完上一个数*i*后，使*i*加**1**可得到下一个数

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int i=1,sum=0; 不能少
```

```
    while (i<=100)
```

```
    { sum=sum+i;
```

```
      i++;
```

```
    }
```

复合语句

```
    printf("sum=%d\n",sum);
```

```
    return 0;
```

```
}
```

```
#include <stdio.h>
int main()
{
    int i=1,sum=0;
    while (i<=100)
    { sum=sum+i;
      i++;    不能丢，否则循环永不结束
    }
    printf("sum=%d\n",sum);
    return 0;
}
```

A terminal window with a black background and white text showing the output of the program: sum=5050.

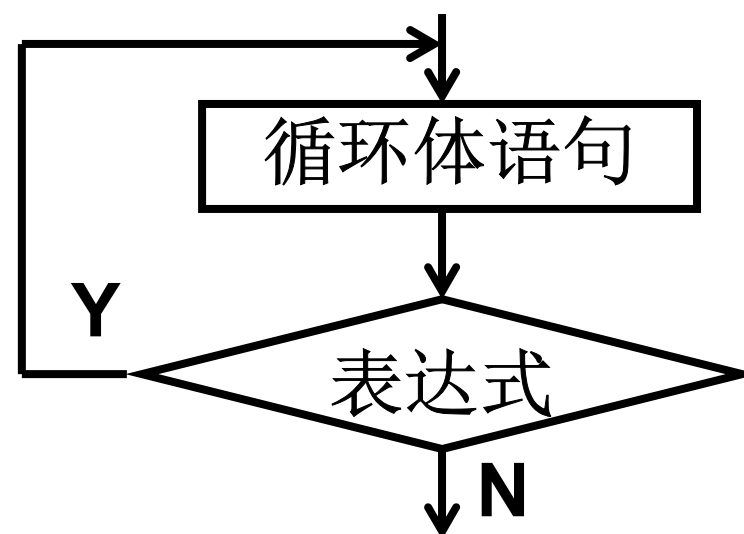
sum=5050

5.3用do---while语句实现循环

➤ **do---while**语句的特点：先无条件地执行循环体，然后判断循环条件是否成立

➤ **do---while**语句的一般形式为：

do
 语句
while (表达式);



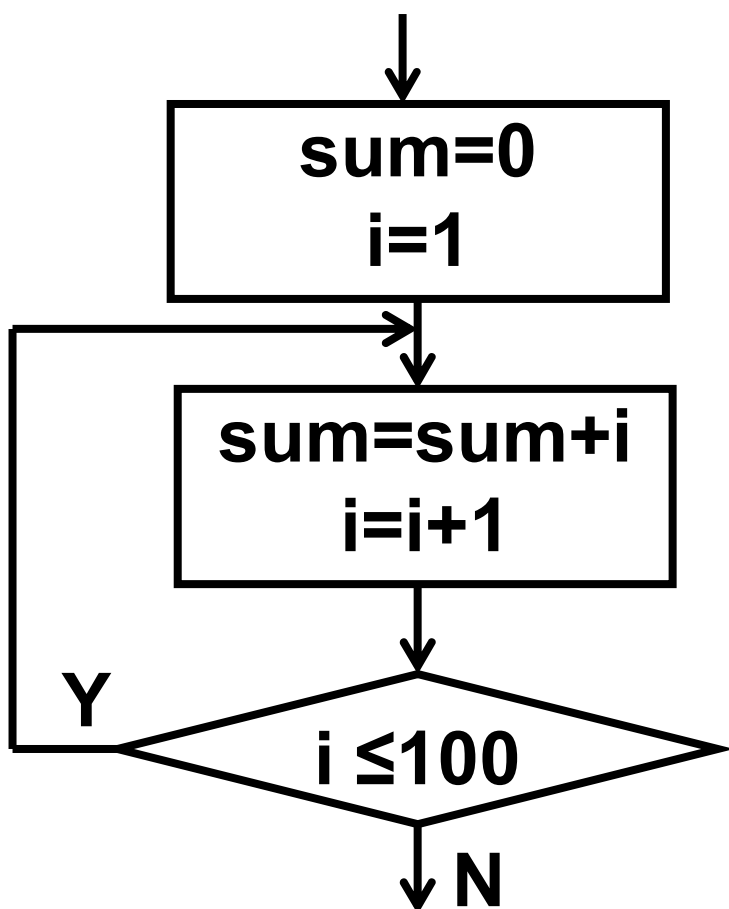
5.3用do---while语句实现循环

例5.2 用do...while语句求：

$$1+2+3+\dots+100, \text{ 即 } \sum_{n=1}^{100} n$$


5.3用do---while语句实现循环

➤ 解题思路:



```
i=1;  sum=0;  
do  
{  
    sum=sum+i;  
    i++;  
}while(i<=100);
```

```
#include <stdio.h>
int main()
{ int i=1,sum=0;
  do
  {
    sum=sum+i;
    i++;
  }while(i<=100);
  printf("sum=%d\n",sum);
  return 0;
}
```

A screenshot of a terminal window with a black background and white text. The text displayed is "sum=5050".

例5.3 while和do---while循环的比较。

in 当**while**后面的表达式的第一次的值为“真”
p 时，两种循环得到的结果相同；否则不相同

```
scanf("%d",&i);  
while(i<=10)  
{  
    sum=sum+i;  
    i++;  
}  
printf("sum=%d\n",sum);
```

```
scanf("%d",&i);  
do  
{  
    sum=sum+i;  
    i++;  
}while(i<=10);  
printf("sum=%d\n",sum);
```

```
i=?1  
sum=55
```

```
i=?11  
sum=0
```

```
i=?1  
sum=55
```

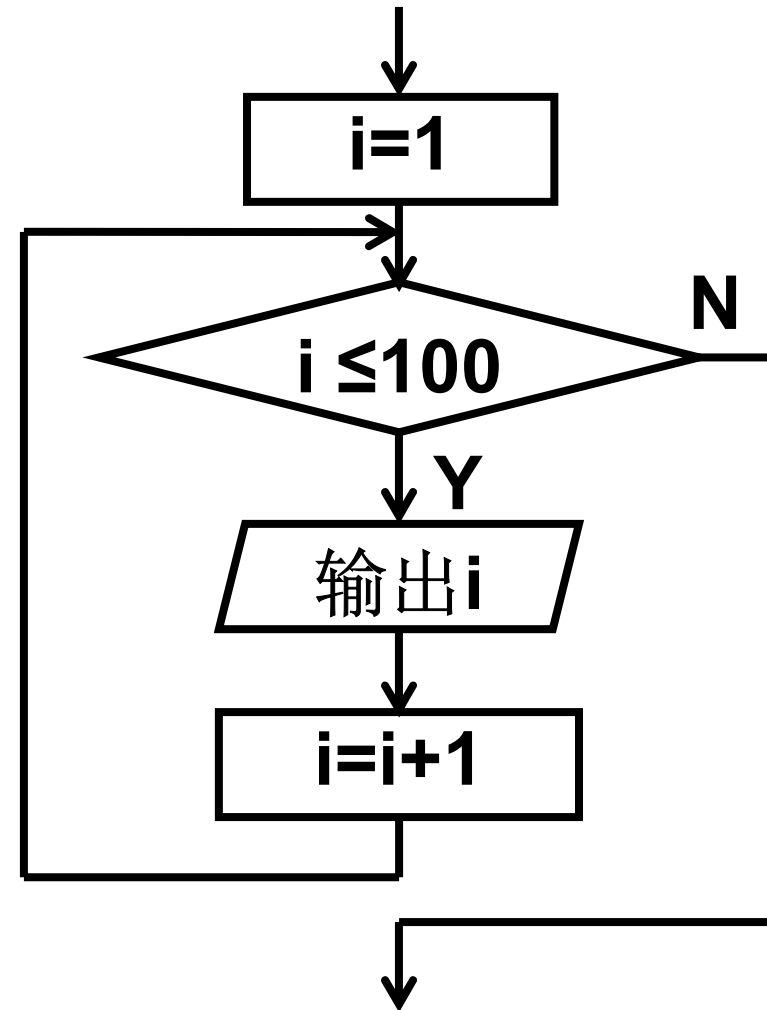
```
i=?11  
sum=11
```

5.4用for 语句实现循环

- **for**语句不仅可以用于循环次数已经确定的情况，还可以用于循环次数不确定而只给出循环结束条件的情况
- **for**语句完全可以代替**while**语句

5.4用for 语句实现循环

```
for (i=1;i<=100;i++)  
{  
    printf("%d ", i );  
}
```



5.4用for 语句实现循环

➤**for**语句的一般形式为

for(表达式1; 表达式2; 表达式3)

语句

设置初始条件，只执行一次。可以为零个、一个或多个变量设置初值执行

5.4用for 语句实现循环

➤**for**语句的一般形式为

for(表达式1; 表达式2; 表达式3)

语句

循环条件表达式，用来判定是否继续循环。在每次执行循环体前先执行此表达式，决定是否继续执行循环

5.4用for 语句实现循环

➤**for**语句的一般形式为

for(表达式1; 表达式2; 表达式3)
语句

作为循环的调整器，例如使循环变量增值，它是在执行完循环体后才进行的

5.4用for 语句实现循环

➤**for**语句的执行过程:

(1) 先求解表达式1

(2) 求解表达式2，若其值为真，执行循环体，然后执行下面第(3)步。若为假，则结束循环，转到第(5)步

(3) 求解表达式3

(4) 转回上面步骤(2)继续执行

(5) 循环结束，执行**for**语句下面的一个语句

5.4用for 语句实现循环

```
for(i=1;i<=100;i++)  
    sum=sum+i;
```

等价于

```
i=1;
```

```
while(i<=100)
```

```
{
```

```
    sum=sum+i;
```

```
    i++;
```

```
}
```

用**for**语句更简单、方便

5.4用for 语句实现循环

for(表达式1; 表达式2; 表达式3)
语句

一个或两个或三个
表达式均可以省略

5.4用for 语句实现循环

```
for (sum=0; i<=100; i++)  
    sum=sum+i;
```

与循环变量无关
合法

5.4用for 语句实现循环

```
for(sum=0,i=1 ; i<=100; i++)  
    sum=sum+i;
```

```
for(i=0,j=100 ; i<=j; i++,j-- )  
    k=i+j;
```

逗号表达式
合法

5.4用for 语句实现循环

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

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



合法

5.5循环的嵌套

- 一个循环体内又包含另一个完整的循环结构，称为循环的嵌套
- 内嵌的循环中还可以嵌套循环，这就是多层循环
- 3种循环(**while**循环、**do...while**循环和**for**循环)可以互相嵌套

5.6几种循环的比较

- (1) 一般情况下，3种循环可以互相代替
- (2) 在**while**和**do---while**循环中，循环体应包含使循环趋于结束的语句。
- (3) 用**while**和**do---while**循环时，循环变量初始化的操作应在**while**和**do---while**语句之前完成。而**for**语句可以在表达式1中实现循环变量的初始化。