

小马加编信息学教案(八)

C++循环结构(二)

- 一. 课程内容
- 二. 知识讲解
 - 1. for 语句的格式、功能
 - * 1.1 for 语句的格式
 - * 1.2 for 语句的含义
 - * 1.3 for 语句的代码流程图
 - * 1.4 for 语句的代码功能
 - 2. for 语句的应用实例
 - 3. 三种循环语句的综合应用
- 三. 经典例题
- 四. 提高巩固

一. 课程内容

1. for 语句的格式、功能
2. for 语句的应用
3. 三种循环语句的综合应用

二. 知识讲解

1. for 语句的格式、功能

1.1 for 语句的格式

for 语句的格式如下：

```
for ( 表达式 1; 表达式 2; 表达式 3) {  
    循环体  
}
```

1.2 for 语句的含义

它的**运行过程**为：

- (1) 先执行“表达式1”。
- (2) 再执行“表达式2”，如果它的值为真（非0），则执行循环体，否则结束循环。
- (3) 执行完循环体后再执行“表达式3”。
- (4) 重复执行步骤 (2) 和 (3)，直到“表达式2”的值为假，就结束循环。

上面的步骤中，(2) 和 (3) 是一次循环，会重复执行，for 语句的主要作用就是不断执行步骤 (2) 和 (3)。

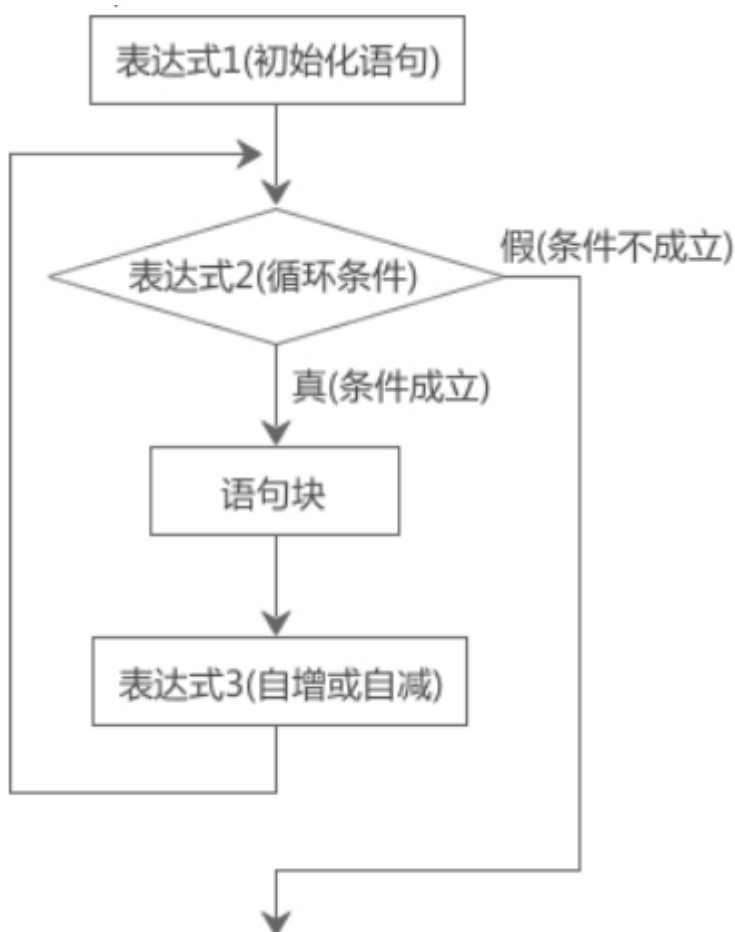
“表达式1”**仅在第一次循环时执行，以后都不会再执行**，可以认为这是一个初始化语句。

“表达式2”一般是一个关系表达式，决定了是否还要继续下次循环，称为“**循环条件**”。

“表达式3”很多情况下是一个带有自增或自减操作的表达式，**以使循环条件逐渐变得“不成立”**。

- for 循环中可以在初始化语句中定义变量,但在该语句中定义的变量只能在循环体内使用而不能超出循环体使用。

1.3 for 语句的代码流程图



1.4 for 语句的代码功能

- 如果重复执行的操作（循环体）次数是固定的、已知的，则一般使用 for 语句。

2. for 语句的应用实例

```
#include <bits/stdc++.h>
int main() {
    int sum=0;
    for(int i = 1; i <= 100; i++){
        sum += i;
    }
    cout << sum << endl;
    return 0;
}
```

• 代码解释

- 执行到 `for` 语句时，先给 i 赋初值1，判断 $i \leq 100$ 是否成立；因为此时 $i = 1$ ， $i \leq 100$ 成立，所以执行循环体。循环体执行结束后（ sum 的值为1），再计算 $i++$ 。
- 第二次循环时， i 的值为 2， $i \leq 100$ 成立，继续执行循环体。循环体执行结束后（ sum 的值为3），再计算 $i++$ 。
- 重复执行步骤 (2)，直到第101次循环，此时 i 的值为101， $i \leq 100$ 不成立，所以结束循环。

- 3个表达式可以同时省略，`for(; ;)` 相当于 `while(1)`

3. 三种循环语句的综合应用

从本质上来说，三种循环都是可以互相替代的
但我们实际应用中，应该**选取最适合我们的循环语句来实现程序**

对于 `while` 和 `do-while` 循环，**循环体中应包括使循环趋于结束的语句。**

对于 `while` 和 `do-while` 循环，循环变量的初始化操作应该在 `while` 和 `do-while` 语句之前完成，而 `for` 循环可以在内部实现循环变量的初始化。

`for` 循环是最常用的循环，它的功能强大，一般都可以代替其他循环

三. 经典例题

1. 循环第一次

小马加编学员们今天刚学会了使用 `for` 循环！于是决定练练手。

输入格式

第一行一个正整数 n ,接下来一行有 n 个数

输出格式

一行一个整数，即为这 n 个数的和。

样例输入	样例输出
5 3 5 -2 -4 1	3
9 3 -2 -1 -5 5 0 4 -4 0	0

2. 乘方计算

输入正整数 a, n 求 a^n 的值 即 n 个 a 相乘的值

输入格式

一行两个正整数 a, n

输出格式

一行一个整数，即为 a^n 的值

样例输入	样例输出
2 5	32
5 4	625

数据范围

$$a \leq 5, n \leq 10$$

3.

$S_n = 1 + 1/2 + 1/3 + \dots + 1/n$, 即 $S_n = \sum_{i=1}^n \frac{1}{i}$

显然对于任意一个整数 K ，当 n 足够大的时候， $S_n > K$

则对于给定的 K ，求出最小的 n ，使得 $S_n > K$

输入格式

一行一个正整数 K

输出格式

一行一个整数，即满足条件的最小的 n

样例输入	样例输出
1	2
2	4

四. 提高巩固

1. 分组求和

读入 n 个正整数，要求对这 n 个数中的奇数和偶数分别求和

输入格式：
第 1 行一个正整数 n 。
第 2~ $n + 1$ 行，每行一个正整数。

输出格式：
共有两行，每行包含一个整数
第一行为所有奇数之和，第二行为所有偶数之和。

样例输入	样例输出
3 2 6 1	1 8
5 1 4 3 5 2	9 6

数据范围
 $1 \leq n \leq 5000$
每个数都在 1 ~ 20000 之间

2. 整数的个数

给定 k 个正整数，每个数都是大于或等于 1，且小于或等于 10。
编程统计其中 1、5 和 10 出现的次数。

输入格式：
第一行一个正整数 k ；
第二行包含 k 个正整数，每两个正整数之间用一个空格隔开。

输出格式：
第一行为 1 出现的次数，第二行为 5 出现的次数，第三行为 10 出现的次数。

样例输入	样例输出
5 1 5 8 10 5	1 2 1

3. 斐波那契数列

斐波那契数列是指这样的数列：

第一个数和第二个数都为 1，接下来每个数都等于前面两个数之和。

编程输入一个正整数 k ，输出斐波那契数列第 k 个数。

输入格式：

一行一个正整数 k

输出格式：

一行一个正整数，表示斐波那契数列第 k 个数的大小。

样例输入	样例输出
5	5
19	4181

数据范围

$1 \leq k \leq 46$