



第一个 do-while 循环程序

这是一个 do-while 循环程序，你能猜测出来它的输出结果是什么吗？

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int a = 0;
6.     cout << a << " before loop" << endl;
7.     do
8.     {
9.         cout << a << endl;
10.        a = a + 1;
11.    }while (a < 10);
12.    cout << a << " after loop" << endl;
13.    return 0;
14. }
```

下面就是程序的输出结果，在循环开始前输出的 a 值是 0，循环中输出从 0 到 9，循环后输出 a 值是 10。

```
0 before loop
0
1
2
3
4
5
6
7
8
9
10 after loop
```

下面我们来正式介绍 C++ 的标准 do-while 循环结构。



do-while 循环语法结构及执行过程

C++标准 do-while 循环的语法结构如下所示：

```
1. do
2. {
3.     .....
4. } while (条件表达式);
```

其中循环体部分由多个语句构成，由一对花括号括起来，构成一个语句块的形式。在书写时，循环体的语句要相对于 do 缩进。

当循环体内的语句只有一条时，可以简化为如下形式：

```
1. do
2.     语句;
3. while (条件表达式);
```

这是一段最简单的 do-while 循环程序代码：

```
1. i = 0;
2. do
3. {
4.     i++;
5. } while (i < 10);
```

可以简化书写为：

```
1. i = 0;
2. do
3.     i++;
4. while (i < 10);
```

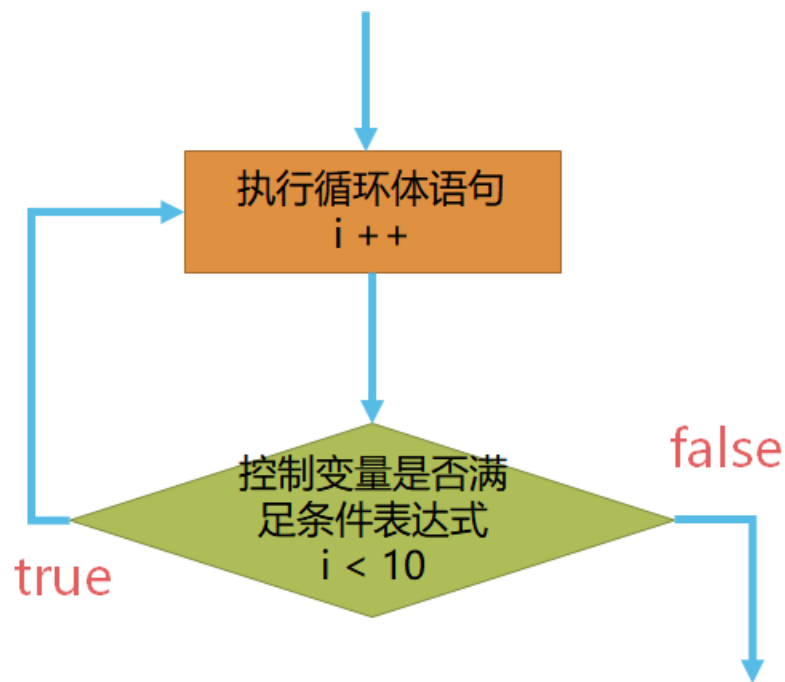


图1. do-while 循环执行过程

C++的 do-while 循环结构具体执行过程如下所述：

- (1) 执行一遍循环体。
- (2) 求出作为循环条件的“条件表达式”的值
- (3) 若为逻辑值真则自动转向第 1 步，否则结束 do-while 循环的执行过程，继续执行其后面的语句。



乘积的尾数

【问题描述】

求 2018 个 2018 的乘积的末两位数是多少？

【输出结果】

24

【分析】

积的个位与十位数只与被乘数与乘数的个位与十位数字有关，所以本题相当于求 2018 个 18 相乘，而且本次的乘积是下一次相乘的被乘数，因此也只需取末两位参与运算就可以了。

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int a = 1;
6.     int t = 0;
7.     do
8.     {
9.         t++;
10.        a = (a*18)%100;
11.    }while (t < 2018);
12.    cout << a << endl;
13.    return 0;
14. }
```



体操队

【问题描述】

校体操队到操场集合，排成每行 2 人，最后多出 1 人；排成每行 3 人，也多出 1 人；分别按每行排 4、5、6 人，都多出 1 人；当排成每行 7 人时，正好不多。求校体操队至少多少人？

【运行结果】

301

【分析】

设校体操队为 x 人，根据题意 x 应是 7 的倍数，因此 x 的初值为 7，以后用 $(x+=7)$ 改变 x 值。

为了控制循环，用逻辑变量 found 为真 (true) 使循环结束。

如果所有条件都满足，found 的值就会为真。

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     bool found = false;
6.     int x = 0;
7.     do
8.     {
9.         x += 7;
10.        if( (x%2==1) && (x%3==1) && (x%4==1) && (x%5==1) && (x%6==1) )
11.        {
12.            found = true;
13.        }
14.    }while (!found);
15.    cout << x;
16.    return 0;
17. }
```