



北京大学  
PEKING UNIVERSITY

信息科学技术学院

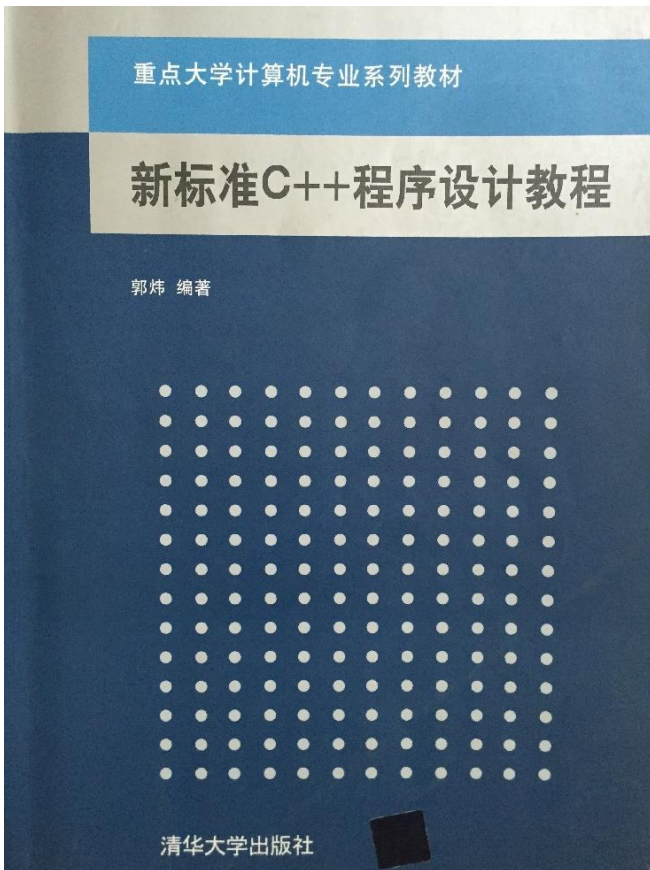
指定教材:

# 程序设计与算法(一)

李文新 郭炜

主讲教师互动微博:

<http://weibo.com/guoweiofpku>





## while循环和do...while 循环

# while循环

并非到达指定次数，而是满足某条件时即停止循环，则适合用**while**语句来实现循环

```
while(表达式) {  
    语句组  
}
```

- 1) 判断“表达式”是否为真，如果不为真，则转4)
- 2) 执行“语句组”
- 3) 转1)
- 4) **while**语句结束，继续执行**while**语句后面的语句。

# while循环

并非到达指定次数，而是满足某条件时即停止循环，则适合用**while**语句来实现循环

```
while(表达式) {  
    语句组  
}
```

```
while(true) {  
    语句组  
} //死循环，可以用break跳出
```

- 1) 判断“表达式”是否为真，如果不为真，则转4)
- 2) 执行“语句组”
- 3) 转1)
- 4) while语句结束，从while语句后面的语句继续执行。

# while循环

**例题:**输入若干个(至少1个) 不超过100的正整数, 输出其中的最大值、最小值以及所有数的和。输入的最后一个数是0, 标志着输入结束。

```
#include <iostream>
using namespace std;
int main()
{
    int sum = 0, maxN = 0, minN = 200, n;
    cin >> n;
    while( n ) {
        if( n > maxN)        maxN = n;
        if( n < minN)        minN = n;
        sum += n;
        cin >> n;
    }
    cout << maxN << " " << minN << " " << sum;
    return 0;
}
```

5 8 2 7 0 ✓

8 2 22

12 43 11 98 47 34 0 ✓

98 11 245

1 0 ✓

1 1 1

# while循环

**例题:**用牛顿迭代法求输入的数的平方根。

欲求 $a$ 的平方根，首先猜测一个值 $x_1=a/2$ （也可以是随便什么其他值）作为其平方根，然后根据下面的迭代公式算出 $x_2$ ，再将 $x_2$ 代入公式右边算出 $x_3$ ……直到连续两次算出的 $x_n$ 和 $x_{n+1}$ 的差的绝对值小于某个值 $\varepsilon$ ，即认为找到了足够精确的平方根。这个 $\varepsilon$ 值取得越小，计算出来的平方根就越精确。

$$\text{迭代公式: } x_{n+1} = (x_n + a / x_n) / 2$$

```
#include <iostream>
using namespace std;
double EPS = 0.001; //用以控制计算精度
int main()
{
    double a;
    cin >> a ; //输入a,要求a的平方根
    if( a >= 0) {
        double x = a/2, lastX = x + 1 + EPS; //确保能够进行至少一次迭代
        while( x - lastX > EPS || lastX - x > EPS){ //只要精度未达要求,
                                                    //就继续迭代

            lastX = x;
            x = (x + a/x)/2;
        }
        cout << x;
    }
    else
        cout << "It can't be nagitive.";
    return 0;
}
```

```

#include <iostream>
using namespace std;
double EPS = 0.001; //用以控制计算精度
int main()
{
    double a;
    cin >> a ; //输入a,要求a的平方根
    if( a >= 0) {
        double x = a/2, lastX = x + 1 + EPS; //确保能够进行至少一次迭代
        while( x - lastX > EPS || lastX - x > EPS){ //只要精度未达要求,
                                                    //就继续迭代

            lastX = x;
            x = (x + a/x)/2;
        }
        cout << x;
    }
    else
        cout << "It can't be nagitive.";
    return 0;
}

```

输入 2时,	
EPS取值	输出结果
1	1.5
0.1	1.41667
0.01	1.41422
0.001	1.41421
0.0001	1.41421



# do...while循环

如果希望循环至少要执行一次，  
就可以使用**do...while**语句

```
do {  
    语句组  
} while (表达式);
```

每执行一次循环后，都要判断“表达式”的值是否为真，如果真就继续循环，如果为假，就停止循环。

## do...while循环

输出1到10000以内所有2的整数次幂:

```
int n = 1;
do {
    cout << n << endl;
    n *= 2;
} while( n < 10000);
```