



北京大学
PEKING UNIVERSITY

信息科学技术学院

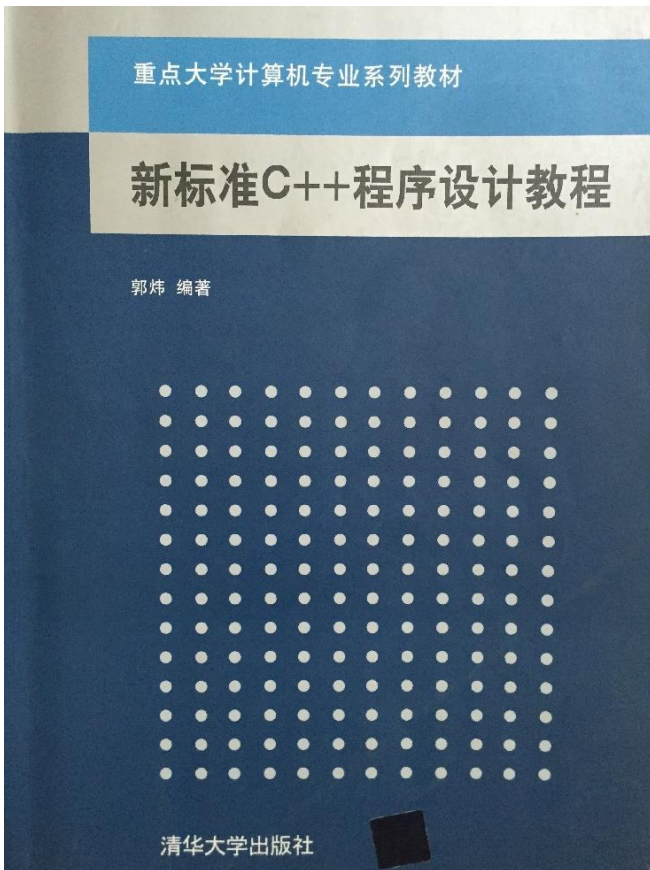
指定教材:

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>





break语句和continue语句

break语句

- 可以出现在循环体中(for、while、do...while循环均可), 其作用是跳出循环。

```
int n = 0;
while(true) {
    if( n > 100)
        break;
    ++n;
}
cout << n ;    ==> 101
```

break语句

- 可以出现在循环体中(for、while、do...while循环均可), 其作用是跳出循环。

```
int n = 0;
while(true) {
    if( n > 100)
        break;
    ++n;
}
cout << n ;    ==> 101
```

- 在多重循环的情况下, break语句只能跳出直接包含它的那一重循环

break语句

例题:如果两个不同的正整数，他们的和是他们的积的因子，就称这两个数为兄弟数，小的称为弟数，大的称为兄数。先后输入正整数 n 和 m ($n < m$)，请在 n 至 m 这 $m-n+1$ 个数中，找出一对兄弟数。如果找不到，就输出“No Solution.”。如果能找到，就找出和最小的那一对；如果有多对兄弟数和相同且都是最小，就找出弟数最小的那一对。

break语句

例题:如果两个不同的正整数，他们的和是他们的积的因子，就称这两个数为兄弟数，小的称为弟数，大的称为兄数。先后输入正整数 n 和 m ($n < m$)，请在 n 至 m 这 $m-n+1$ 个数中，找出一对兄弟数。如果找不到，就输出“No Solution.”。如果能找到，就找出和最小的那一对；如果有多对兄弟数和相同且都是最小，就找出弟数最小的那一对。

思路:枚举每一对不同的数，看看是不是兄弟数。用两个变量记录当前已经找到的最佳兄弟数，如果发现更佳的，就重新记录。

```

#include <iostream>
using namespace std;
int main() {
    int n,m;
    cin >> n >> m;
    int a = m + 1, b = m + 1; //a,b记录已经找到的最佳兄弟数，a是弟数，b是兄数
    for( int i = n; i < m ; ++i ) { //取弟数，共m-n种取法
        if( i > (a + b)/2 + 1)
            break; // 跳出外重循环
        for( int j = i + 1; j <= m; ++j ) { //取兄数
            if( i + j > a + b ) break; // 跳出内重循环
            if( i * j % (i + j) == 0 ) { //发现兄弟数
                if( i + j < a + b) { //发现和更小的兄弟数
                    a = i; b = j ; //更新已找到的最佳兄弟数
                }
                else if( i + j == a + b && i < a)
                    //发现和相同但弟数更小的兄弟数
                    a = i; b = j; //更新已找到的最佳兄弟数
            }
        }
    }
}

```

```
if( a == m + 1 ) //没找到兄弟数
    cout << "No solution.";
else
    cout << a << "," << b ;
return 0;
}
```


continue语句

- 可以出现在循环体中(for、while、do...while循环均可)，其作用是立即结束本次循环，并回到循环开头判断是否要进行下一次循环。

```
for( int i = 1;i <= 10 ;++i ) {  
    if( i % 2 )  
        continue; //导致不执行后面的语句，回到循环开头  
    cout << i << ",";  
}  
=> 2,4,6,8,10,
```

continue语句

- 可以出现在循环体中(for、while、do...while循环均可)，其作用是立即结束本次循环，并回到循环开头判断是否要进行下一次循环。

```
for( int i = 1;i <= 10 ;++i ) {  
    if( i % 2 )  
        continue; //导致不执行后面的语句，回到循环开头  
    cout << i << ", ";  
}  
=> 2,4,6,8,10,
```

- 在多重循环的情况下，continue只对直接包含它的那重循环起作用。