



程序设计与算法(一)

C语言程序设计

郭 炜

微信公众号



微博: <http://weibo.com/guoweiofpku>

学会程序和算法，走遍天下都不怕!

讲义照片均为郭炜拍摄



北京大学
PEKING UNIVERSITY

信息科学技术学院

指定教材：

《新标准C++程序设计教程》

郭炜 编著

清华大学出版社

重点大学计算机专业系列教材

新标准C++程序设计教程

郭炜 编著



清华大学出版社



北京大学
PEKING UNIVERSITY

信息科学技术学院

循环结构
之
break语句
continue语句



河北白洋淀

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。

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。



北京大学
PEKING UNIVERSITY

信息科学技术学院

OJ编程题 输入数据的处理



韩国济州岛

scanf表达式的值

- scanf(...)表达式的值为int，表示成功读入的变量个数。

```
int n,m;  
printf("%d",scanf("%d%d",&n,&m));
```

12 56 ✓

2

40 a ✓

1

a 40 ✓

0

scanf表达式的值

- scanf(...) 值为EOF(即-1)则说明输入数据已经结束

```
int n,m;  
while(scanf("%d%d",&n,&m) != EOF) { //EOF是直接能拿来用的符号常量  
    printf("%d",n+m);  
}
```

Windows系统下,不停输入两个整数再敲回车,

则不停输出它们的和,直到在单独一行输入 Ctrl+Z 然后 回车

程序结束(单独一行的Ctrl+Z表示输入结束)

Mac OS或Linux系统下是Ctrl+D或Ctrl+W

12 2 ✓
14
1 2 ✓
3
...
^Z ✓

scanf表达式的值

也可以:

```
int n,m;  
while (scanf ("%d%d", &n, &m) == 2) {  
    printf ("%d", n+m);  
}
```

Windows系统下,不停输入两个整数再敲回车,

则不停输出它们的和,直到在单独一行输入 Ctrl+Z 然后 回车

程序结束 (单独一行的Ctrl+Z表示输入结束)

Mac OS或Linux系统下是Ctrl+D或Ctrl+W

12 2 ✓
14
1 2 ✓
3
...
^Z ✓

cin表达式的值

- `cin >> m >> n ...` 表达式的值, 在成功读入所有变量时为true, 否则为false

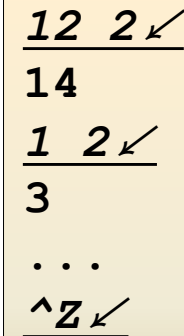
```
int n,m;  
while(cin >> n >> m ) {  
    printf("%d",n+m);  
}
```

Windows系统下, 不停输入两个整数再敲回车,

则不停输出它们的和, 直到在单独一行输入 `Ctrl+Z` 然后 回车

程序结束 (单独一行的`Ctrl+Z`表示输入结束)

Mac OS或Linux系统下是`Ctrl+D`或`Ctrl+W`



12 2 ✓
14 ✓
1 2 ✓
3 ✓
...
^Z ✓

处理无结束标记的OJ题目输入

- 输入若干个(不知道多少个)正整数，输出其中的最大值

Sample Input:

2 3 4 5 6 787 54532 12

Sample Output

54532

处理无结束标记的OJ题目输入

- 输入若干个(不知道多少个)正整数，输出其中的最大值

Sample Input:

2 3 4 5 6 787 54532 12

Sample Output

54532

```
#include <iostream>
#include <cstdio>
using namespace std;
int main() {
    int n, mx = 0;
    while (scanf("%d", &n) != EOF) {
        //或 while (scanf("%d", &n) == 1) {
            if ( n > mx )
                mx = n;
        }
    }
    printf("%d", mx);
    return 0;
}
```

本机测试的时候，
用单独一行的Ctrl+Z或trl+D
或Ctrl+W
表示输入结束

处理无结束标记的OJ题目输入

- 输入若干个(不知道多少个)正整数, 输出其中的最大值

Sample Input:

2 3 4 5 6 787 54532 12

Sample Output

54532

```
#include <iostream>
#include <cstdio>
using namespace std;
int main() {
    int n, mx = 0;
    while( cin >> n) {
        if( n > mx )
            mx = n;
    }
    printf("%d",mx);
    return 0;
}
```

本机测试的时候,
用单独一行的Ctrl+Z或trl+D
或Ctrl+W
表示输入结束



北京大学
PEKING UNIVERSITY

信息科学技术学院

循环例题选讲



韩国釜山龙头岩

例1.乘方计算

给出一个整数 a 和一个正整数 n ，求乘方 a^n 。

输入:

一行，包含两个整数 a 和 n 。 $-1000000 \leq a \leq 1000000$, $1 \leq n \leq 10000$ 。

输出:

一个整数，即乘方结果。题目保证最终结果的绝对值不超过1000000。

样例输入

2 3

样例输出

8

例1.乘方计算

```
#include <iostream>
using namespace std;
int main() {
    int a,n;
    cin >> a >> n;
    int result = a;
    for(int i = 0;i < n-1; ++i)
        result *= a;
    cout << result ;
}
```

例2. 输入若干个整数求最大值

输入若干个整数（可正可负，不超过int的表示范围），输出最大值

Sample Input

-100 -20 20 -2

Sample Output

20

例2. 输入若干个整数求最大值

```
#include <iostream>
using namespace std;
int main() {
    int n,mx;
    bool first = true; //输入的是否是第一个数
    while(cin >> n) {
        if( first ) {
            mx = n;
            first = false;
        }
        else {
            if( n > mx )
                mx = n;
        }
    }
    cout << mx << endl;
    return 0;
}
```

例3. 输入至少2个整数，求最大值和第二大值

Sample Input

1 5 6 3 4 6

Sample Output

6 6

Sample Input

1 5 6 3 4

Sample Output

6 5

例3. 输入至少2个整数，求最大值和第二大值

```
#include <iostream>
using namespace std;
int main() {
    int n,max1,max2;
    int num = 0; // 输入的是第几个数
    while(cin >> n) {
        ++num;
        if( num == 1)
            max1 = n;
        else if( num == 2) {
            if( n > max1) {
                max2 = max1;
                max1 = n;
            }
            else
                max2 = n;
        }
    }
}
```


例3. 输入至少2个整数，求最大值和第二大值

```
    else { // num > 2
        if( n >= max1) {
            max2 = max1;
            max1 = n;
        }
        else if( n > max2 )
            max2 = n;
    }
}
cout << max1 << " " << max2 << endl;
return 0;
}
```

例4. 斐波那契数列

斐波那契数列是指这样的数列: 数列的第一个和第二个数都为1, 接下来每个数都等于前面2个数之和。给出一个正整数k, 要求斐波那契数列中第k个数是多少。

输入:

输入一行, 包含一个正整数k。 ($1 \leq k \leq 46$)

输出:

输出一行, 包含一个正整数, 表示斐波那契数列中第k个数的大小

样例输入

19

样例输出

4181

例4. 斐波那契数列

```
#include <iostream>
using namespace std;
int main() {
    int a1 = 1, a2 = 1; int k;
    cin >> k;
    if( k == 1 || k == 2)
        cout << 1 << endl;
    else {
        int sum;
        for(int i = 0; i < k-2; ++i) {
            sum = a1+a2;
            a1 = a2;
            a2 = sum;
        }
        cout << a2 << endl;
    }
    return 0;
}
```

例5. 求阶乘的和

给定正整数 n ，求不大于 n 的正整数的阶乘的和
(即求 $1!+2!+3!+\dots+n!$)

输入

输入有一行，包含一个正整数 n ($1 < n < 12$) 。

输出

输出有一行：阶乘的和。

样例输入

5

样例输出

153

例5. 求阶乘的和

解法1:

```
#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    int sum = 0;
    for(int i = 1; i <= n; ++i) {
        int factorial=1; //存放i阶乘
        for(int j = 1; j <= i; ++j)
            factorial *= j; //此操作做1+2+3+...+n次
        sum += factorial;
    }
    cout << sum ;
    return 0;
}
```

例5. 求阶乘的和

更快的解法2:

```
#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    int sum = 0;
    int factorial = 1;
    for(int i = 1; i <= n; ++i) {
        factorial *= i;
        sum += factorial;
    }
    cout << sum ;
    return 0;
}
```

例6. 输入正整数 $n(n \geq 2)$, 求不大于 n 的全部质数

解法1:

```
#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    for(int i = 2; i <= n; ++i) { //每次判断i是否是质数
        int k;
        for(k = 2; k < i; ++k) {
            if( i % k == 0)
                break;
        }
        if( k == i) // k == i说明没有执行过break
            cout << i << endl;
    }
    return 0;
} //此解法做了没必要的尝试, k 大于 i的平方根后就不必再试
```

例6. 输入正整数 $n(n \geq 2)$,求不大于 n 的全部质数

改进的解法2:

```
#include <iostream>
using namespace std;

int main() {
    int n;
    cin >> n;
    cout << 2 << endl;
    for(int i = 3; i <= n; i+=2) { //每次判断i是否是质数
        int k;
        for(k = 3; k < i; k+=2) {
            if( i % k == 0)
                break;
            if( k*k > i)
                break;
        }
        if( k*k > i) cout << i << endl;
    }
    return 0;
}
```