



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>

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

循环例题选讲

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;
    cin >> mx;
    while(cin >> n) {
        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;
}
```

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;
} //此解法做了不必要的尝试, 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;
}
```