



质数

【问题描述】

输出 100~200 中所有的质数。

【运行结果】

101	103	107	109	113	127	131	137	139
149	151	157	163	167	173	179	181	191
193	197	199						

【分析】

对 100~200 之间的每一个整数进行判断，若它是质数，则输出。

对于任意整数 i ，根据质数定义，从 2 开始到 \sqrt{i} ，找 i 的因数，若找到一个因数，则 i 不是质数。

【参考程序】

```
1. #include<iostream>
2. #include<cmath>
3. using namespace std;
4. int main()
5. {
6.     for (int i=100; i<=200; i++)
7.     {
8.         //是否找到因数
9.         bool found = false;
10.        //floor 为取整函数，在 cmath 库中
11.        int n = floor(sqrt(i));
12.        for(int j=2; j<=n; j++)
13.        {
14.            //找到因数，跳出循环
15.            if(i%j==0)
16.            {
17.                found = true;
18.                break;
19.            }
20.        }
21.        //没有找到说明是质数
22.        if (!found)
```



```
23.     {  
24.         cout << i << " ";  
25.     }  
26. }  
27.     return 0;  
28. }
```



分解质因数

【问题描述】

把一个合数分解成若干个质因数乘积的形式叫做分解质因数。输入一个正整数 n ，将 n 分解成质因数乘积的形式。

【输入样例】

36

【输出样例】

36=2*2*3*3

【分析】

将任意的 n 分解为质因数的乘积：从 2 开始试除，能整除就输出 2，再对商继续试除，直到不再含有因子 2；然后用下一个数反复试除；一直到商为 1，停止操作。

试除因数的递增是一层循环，每一个因数的反复试除是一层循环。因此需要两层循环。

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int n;
6.     cin >> n;
7.     cout << n << "=";
8.     int i = 2;
9.     do
10.    {
11.        //n 能被 i 整除，就重复做除法操作
12.        while(n % i == 0)
13.        {
14.            cout << i;
15.            n /= i;
16.            //最后一项后面不用输出*
17.            if(n != 1)
18.            {
19.                cout << "*";
```



```
20.         }  
21.     }  
22.         //下一个可能的因数，如果不是素数，则 n%i 一定不是 0  
23.         i ++;  
24.     } while(n!=1);    //n 没有除尽，就重复操作  
25.     return 0;  
26. }
```