



最大公约数

【问题描述】

求两个正整数 m 、 n 的最大公约数。

【输入样例】

```
30 54
```

【输出样例】

```
6
```

【方法 1】

任意两个正整数 m 和 n 的最大公约数，最大是 $\min(m, n)$ ，最小是 1。

用 gcd 表示最大公约数，从 $\min(m, n)$ 开始判断，若 $\text{gcd} > 1$ 并且没有同时整除 m 和 n ，那么就 $\text{gcd} - 1$ ，重复判断是否整除。

【方法 1】参考程序

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int m, n;
6.     cin >> m >> n;
7.     //注意三目运算符的写法
8.     int gcd = m>n ? n : m;
9.     while ( gcd>1 && (m%gcd!=0 || n%gcd!=0) )
10.    {
11.        //每次减 1 寻找最大公约数
12.        gcd--;
13.    }
14.    //输出最大公约数
15.    cout << gcd << endl;
16.    return 0;
17. }
```

【方法 2】



求两个正整数的最大公约数可以采用辗转相除法，即欧几里德算法。对于任意正整 m 和 n ，用 m ， n ， r 分别表示被除数、除数、余数，那么 m 和 n 的最大公约数等于 n 和 r 的最大公约数。

以下是辗转相除法的算法：

- 1) 求 m 除以 n 的余数 r ;
- 2) 当 $r \neq 0$ ，执行第 3 步；若 $r = 0$ ，则 n 为最大公约数，算法结束。
- 3) 将 n 的值赋给 m ，将 r 的值赋给 n ；再求 m 除以 n 的余数 r 。
- 4) 转到第 2 步。

【方法 2】参考程序

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int m, n;
6.     cin >> m >> n;
7.     int r = m % n;
8.     while (r != 0)
9.     {
10.        m = n;
11.        n = r;
12.        r = m % n;
13.    }
14.    cout << n << endl;
15.    return 0;
16. }
```



最大公约数

【问题描述】

用 do-while 实现：求两个正整数m、n 的最大公约数。

【完善代码】

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int m, n, r;
6.     cin >> m >> n;
7.     //辗转相除法
8.     do
9.     {
10.         r = m % n;
11.         m = ____;
12.         n = ____;
13.     } while ( ____ );
14.     cout << "The greatest common divisor is: " << ____;
15.     return 0;
16. }
```

【答案】

第 1 行: n

第 12 行: r

第 13 行: $r > 0$

第 14 行: m



最小公倍数

【问题描述】

求两个正整数 m 、 n 的最小公倍数。

【输入样例】

30 54

【输出样例】

90

【算法分析】

公式法：两个数的乘积等于这两个数的最大公约数与最小公倍数的积。

最小公倍数=两个数的乘积÷两个数的最大公约数

$$\text{lcm}(x, y) = \frac{x \times y}{\text{gcd}(x, y)}$$

【参考代码】

```
#include<iostream>
using namespace std;
int main()
{
    int m, n;
    cin >> m >> n;
    int p = m * n;
    //求最大公约数
    int r = m % n;
    while (r != 0)
    {
        m = n;
        n = r;
        r = m % n;
    }
    //输出最小公倍数
    cout << p / n;
    return 0;
}
```



}