

模型部分：

在求最大公约数时用辗转相除法，

在求最小公倍数时利用性质

$$<\text{最大公约数} \times \text{最小公倍数} = \text{两数之积}>$$

来进行计算。代码如下（GCD 为最大公约数，LCM 为最小公倍数）：

```
void SAMPLE::GETGCD()
{
    int a = input1;
    int b = input2;
    while (1)
    {
        a %= b;
        if (a == 0) break;
        swap(a, b);
    }
    cout << input1 << "和" << input2 << "的最大公约数为" << b << '。' << endl;
    GCD = b;
}

void SAMPLE::GETLCM()
{
    int k = input1 * input2;
    cout << input1 << "和" << input2 << "的最小公倍数为" << k / GCD << '。' << endl;
}
```

验证部分：

采用手动输入测试样例之后以编程手段辅助的方法进行验证，原程序

部分结果如下：

Microsoft Visual Studio 调试控制

```
请输入两个整数：
25 125
25和125的最大公约数为25。
计算所用时间是1毫秒。
25和125的最小公倍数为125。
计算所用时间是0毫秒。
```

Microsoft Visual Studio 调试控制台

```
请输入两个整数：
3456 2500
3456和2500的最大公约数为4。
计算所用时间是0毫秒。
3456和2500的最小公倍数为2160000。
计算所用时间是1毫秒。
```

Microsoft Visual Studio 调试控制台

```
请输入两个整数：
1234 1314
1234和1314的最大公约数为2。
计算所用时间是1毫秒。
1234和1314的最小公倍数为810738。
计算所用时间是1毫秒。
```

Microsoft Visual Studio 调试控制台

```
请输入两个整数：
135 19
135和19的最大公约数为1。
计算所用时间是1毫秒。
135和19的最小公倍数为2565。
计算所用时间是1毫秒。
```

因为最小公倍数由最大公约数得到，因此只验证最大公约数，
验证程序的源代码如下：

```

#include<bits/stdc++.h>
using namespace std;

int main( )
{
    int a,b,maxi=0;
    cin>>a>>b;
    for(int i = 1;i <= min(a,b); i++)
    {
        if( a%i == 0 && b%i == 0)//同时为两个数的因数
            maxi=i;//取最大公因数
    }
    cout<<maxi;
    return 0;
}

```

对于原程序的四个样例，验证程序结果如下，与原程序一致：

E:\cpp\FOP\oj.exe

```

25 125
25
-----
Process exited after 13.
请按任意键继续. . .

```

E:\cpp\FOP\oj.exe

```

3456 2500
4
-----
Process exited after
请按任意键继续. . .

```

E:\cpp\FOP\oj.exe

```

1234 1314
2
-----
Process exited after
请按任意键继续. . .

```

E:\cpp\FOP\oj.exe

135 19

1

Process exited after
请按任意键继续. . .