

使用函数验证哥德巴赫猜想

本题要求实现一个判断素数的简单函数，并利用该函数验证哥德巴赫猜想：任何一个不小于6的偶数均可表示为两个奇素数之和。素数就是只能被1和自身整除的正整数。注意：1不是素数，2是素数。

函数接口定义：

```
1 bool prime( int p )
2 {
3
4 }
5 void Goldbach( int n )
6 {
7
8 }
```

其中函数 `prime` 当用户传入参数 `p` 为素数时返回 `true`，否则返回 `false`；函数 `Goldbach` 按照格式“`n=p+q`”输出 `n` 的素数分解，其中 $p \leq q$ 均为素数。又因为这样的分解不唯一（例如24可以分解为5+19，还可以分解为7+17），要求必须输出所有解中 `p` 最小的解。

裁判测试程序样例：

```
1 #include <stdio.h>
2 #include <math.h>
3
4 /* 你的代码将被嵌在这里 */
5
6 int main()
7 {
8     int m, n, i, cnt;
9
10    scanf("%d %d", &m, &n);
11    if ( prime(m) )
12    {
13        printf("%d is a prime number\n", m);
14    }
15    if ( m < 6 )
16    {
17        m = 6;
18    }
19    if ( m % 2 != 0 )
20    {
21        m++;
22    }
23    cnt = 0;
24    for( i=m; i<=n; i+=2 )
25    {
26        Goldbach(i);
27        cnt++;
28        if ( cnt%5 != 0 )
29        {
30            printf(", ");
```

```
31     }
32     else
33     {
34         printf("\n");
35     }
36 }
37
38 return 0;
39 }
```

输入样例：

```
1 89 100
```

输出样例：

```
1 89 is a prime number
2 90=7+83, 92=3+89, 94=5+89, 96=7+89, 98=19+79
3 100=3+97,
```