大意：给定正整数N，求出

for(i=1;i<N;i++)  
     for(j=i+1;j<=N;j++)  
        G+=gcd(i,j);

G的值。

假设a、b(a<b)互质，那么gcd(a,b)=1，这样当i循环到a、j循环到b时就会向结果中+1，而i循环到2\*a、j循环到2\*b时就会向结果中+2（gcd(2\*a,2\*b)=2）...循环到k\*a和k\*b时就会向结果中+k。**这样实际上引起结果变化的根源就在于各对互质的数**，当i、j循环到他们自身或者自身的倍数时就会引起结果的改变，那么我们不妨先将每对互质的数对结果的贡献值算出来，最后将各对互质的数对结果的贡献累加起来就可以了。

    假设和b互质的数有n个，也就是n对(?,b)（?和b互质），那么在i、j循环到?、b时结果会增加n，循环到(2\*?,2\*b)时结果就会增加2\*n...当i、j循环到k\*?、k\*b时结果就会增加k\*n。那么我们不妨用a[i]记录各种k、b在满足k\*b=i时会增加多少结果，那么最后我们要输出的就是a[2]+a[3]+...+a[N]。

    至于找和b互质的数，就是计算b的欧拉函数的值，然后暴力循环k，并修改对应的a[k\*b]即可，整体的复杂度是O(N\*logN)的。

分析：老样子，举例看规律

http://latex.codecogs.com/gif.latex?%5C%5C%20gcd(x,3)=1%20%20%5Cquad%20there%20~are%5Cquad%20%5Cphi(3)=2%5Cquad%20pairs     n=3   贡献的公约数：1\*2

http://latex.codecogs.com/gif.latex?%5C%5C%20gcd(2x,6)=2%20%20%5Cquad%20there%20~are%5Cquad%20%5Cphi(3)=2%5Cquad%20pairs    n=6   贡献的公约数：2\*2

http://latex.codecogs.com/gif.latex?%5C%5C%20gcd(3x,9)=3%20%20%5Cquad%20there%20~are%5Cquad%20%5Cphi(3)=2%5Cquad%20pairs     n=9   贡献的公约数：3\*2

所以我们可以这样想，对N素因子分解，然后用得到的一个个素因子去计算所有n的公约数贡献值。

比如上面，p=3.

n=3,  ans+=phi(3)\*1;

n=3\*2, ans+=phi(3)\*2

n=3\*3, ans+=phi(3)\*3.

设一个数组f[]，f(i\*j)+=phi(i)\*j.

最后累加。

#include<cstdio>

#include<cstring>

#include<iostream>

#include<algorithm>

#include<map>

#include<vector>

#include<queue>

#include<stack>

#include<set>

#include<cmath>

#define ll long long

const int maxn=5e6;

#define mod 1000000007

using namespace std;

int phi[maxn]; //欧拉函数

bool is\_prime[maxn]; //初始化

int prime[maxn]; //保存素数

ll f[maxn];

void Euler(int n)

{

int i, j, k;

k = 0;

for(i = 2; i < n; i++)

{

if(is\_prime[i] == false)

{

prime[k++] = i;

phi[i] = i-1;

}

for(j = 0; j<k && i\*prime[j]<n; j++)

{

is\_prime[ i\*prime[j] ] = true;

if(i%prime[j] == 0)

{

phi[ i\*prime[j] ] = phi[i] \* prime[j];

break;

}

else

{

phi[ i\*prime[j] ] = phi[i] \* (prime[j]-1);

}

}

}

}

int main()

{

Euler(maxn);

int n;

for(int i=2;i<maxn;i++)

for(int j=1;j\*i<maxn;j++)

f[j\*i]+=j\*phi[i];

while(cin>>n&&n)

{

ll ans=0;

for(int i=1;i<=n;i++)

ans+=f[i];

printf("%lld\n",ans);

}

return 0;

}