过程

数论分块的过程大概如下：考虑和式

$\sum_{i=1}^{n} f(i)\left\lfloor\frac{n}{i}\right\rfloor$

那么由于我们可以知道 $\left\lfloor\frac{n}{i}\right\rfloor$ 的值成一个块状分布（就是同样的值都聚集在连续的块中），那么就可以用数论分块加速计算，降低时间复杂度。

利用上述结论，我们先求出 $f(i)$ 的 **前缀和**（记作 $s(i)=\sum_{j=1}^{i} f(j)$），然后每次以 $[l,r]=[l,\left\lfloor\frac{n}{\lfloor\frac{n}{i}\rfloor}\right\rfloor]$ 为一块，分块求出贡献累加到结果中即可。

伪代码如下：

$$
\begin{array}{l}
\text{Calculate } s(i)\text{, the prefix sum of } f(i). \\
l \leftarrow 1 \\
r \leftarrow 0 \\
result \leftarrow 0 \\
\textbf{while } l \le n \textbf{ do :} \\
\qquad r \leftarrow \left\lfloor \dfrac{n}{\lfloor n/l \rfloor} \right\rfloor \\
\qquad result \leftarrow result + [s(r) - s(l-1)] \times \left\lfloor \dfrac{n}{l} \right\rfloor \\
\qquad l \leftarrow r + 1 \\
\textbf{end while}
\end{array}
$$

最终得到的 $result$ 即为所求的和式。

Figure 1: 整除分块.png

```cpp
#include<bits/stdc++.h>
using namespace std;
#define ll long long
#define inf 100000000000000
const int N=2e6+7;
//题目，在 1<=a<b<=n 的条件下，求 gcd(a,b)的和

//这里用到了欧拉函数
//欧拉函数也可用欧拉筛求出
int main()
{
    void solve();
    ios::sync_with_stdio(0);cin.tie(0);cout.tie(0);
    solve();
    return 0;
}
bool isprime[N];
vector<ll> p;
ll phi[N]={0,1};//边界条件
```

```
void eular(int n)
{
    for(int i=2;i<=n;i++)
    {
        if(!isprime[i])
        {
            p.push_back(i);
            phi[i]=i-1;
        }
        for(auto re:p)
        {
            if(i*re>n)break;
            isprime[i*re]=1;
            if(i%re==0){phi[i*re]=phi[i]*re;break;}
            phi[i*re]=phi[i]*(re-1);
        }
    }
    //到此，欧拉函数就求出来了
    for(int i=1;i<=n;i++)phi[i]+=phi[i-1];  //此处在求函数的前缀和，用于整除分块
}


ll cal(ll n,ll m)
{
    ll l=1,r=0,ans=0;
    while(l<=n)
    {
        r=min((n/(n/l)),(m/(m/l)));
        ans+=(phi[r]-phi[l-1])*(n/l)*(m/l);
        l=r+1;
    }
    return ans;
}
void solve()
{
    ll n;
    cin>>n;
    eular(n);
    ll ans=(cal(n,n)-n*(n+1)/2)/2;
    cout<<ans<<"\n";
}
```