# Visible Lattice Points（spoj7001+初探莫比乌斯）gcd（a，b，c）=1 经典

2015年08月17日 22:00:16 [寻找星空的孩子](https://me.csdn.net/u010579068" \t "https://blog.csdn.net/u010579068/article/details/_blank) 阅读数：913更多

所属专栏： [ACM--数论专题](https://blog.csdn.net/column/details/acm-theory-of-number.html" \t "https://blog.csdn.net/u010579068/article/details/_blank)

 版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/u010579068/article/details/47733123

## **VLATTICE - Visible Lattice Points**

*no tags*

Consider a N\*N\*N lattice. One corner is at (0,0,0) and the opposite one is at (N,N,N). How many lattice points are visible from corner at (0,0,0) ? A point X is visible from point Y iff no other lattice point lies on the segment joining X and Y.   
   
Input :   
The first line contains the number of test cases T. The next T lines contain an interger N   
   
Output :   
Output T lines, one corresponding to each test case.   
   
Sample Input :   
3   
1   
2   
5   
   
Sample Output :   
7   
19   
175   
   
Constraints :   
T <= 50   
1 <= N <= 1000000

****参考学长博客**[>>芷水<<](http://www.cnblogs.com/tom987690183/p/3933335.html" \t "https://blog.csdn.net/u010579068/article/details/_blank)**

题意：GCD(a,b,c)=1, 0<=a,b,c<=N ;  
  
莫比乌斯反演，十分的巧妙。  
  
GCD(a,b)的题十分经典。这题扩展到GCD(a,b,c)加了一维，但是思想却是相同的。  
  
设f(d) = GCD(a,b,c) = d的种类数 ；   
  
F(n) 为GCD(a,b,c) = d 的倍数的种类数, n%a == 0 n%b==0 n%c==0。  
  
即 ：F(d) = (N/d)\*(N/d)\*(N/d);//N中是d的倍数的个数，然后组合  
  
则f(d) = sigma( mu[n/d]\*F(n), d|n )  
  
由于d = 1 所以f(1) = sigma( mu[n]\*F(n) ) = sigma( mu[n]\*(N/n)\*(N/n)\*(N/n) );  
  
由于0能够取到，所以对于a,b,c 要讨论一个为0 ，两个为0的情况 (3种).其实又要在开头加两个0的情况；中间算三

个的时候，把其中一个变成0，就是一个0的情况了。

****初探莫比乌斯。还有很多不是很懂。跟进中。。。****

****发现莫比乌斯真的很巧妙呢。。。要多加练习。嗯！****

****转载请注明出处：**[寻找&星空の孩子](http://blog.csdn.net/u010579068" \t "https://blog.csdn.net/u010579068/article/details/_blank)**

题目链接：[http://www.spoj.com/problems/VLATTICE/](http://www.spoj.com/problems/VLATTICE/" \t "https://blog.csdn.net/u010579068/article/details/_blank)   对了：用c++4.3.2交过的

（ps：这里的账号居然我都注册不了，还得借学长的。这算是歧视吗。。。〒\_〒）

#include<cstdio>

#include<cstring>

#include<algorithm>

using namespace std;

const int MAXN = 1e6+10;

typedef long long LL;

LL F[MAXN],f[MAXN];

LL pri[MAXN],pri\_num;

LL mu[MAXN];*//莫比乌斯函数值*

int vis[MAXN];

void mobius(int N) *//筛法求莫比乌斯函数*

{

pri\_num = 0;*//素数个数*

memset(vis, 0, sizeof(vis));

vis[1] = mu[1] = 1;

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

{

if(!vis[i])

{

pri[pri\_num++] = i;

mu[i] = -1;

}

for(int j=0; j<pri\_num && i\*pri[j]<N ;j++)

{

vis[i\*pri[j]]=1;*//标记非素数*

*//eg:i=3,i%2,mu[3\*2]=-mu[3]=1;----;i=6,i%5,mu[6\*5]=-mu[6]=-1;*

if(i%pri[j])mu[i\*pri[j]] = -mu[i];

else

{

mu[i\*pri[j]] = 0;

break;

}

}

}

}

int main()

{

int T,n;

scanf("%d",&T);

mobius(1000005);

while(T--)

{

scanf("%d",&n);

LL ans = 3;*//(0,0,1)(0,1,0)(1,0,0)三个特例*

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

ans+=mu[i]\*(n/i)\*(n/i)\*((n/i)+3);*//+3因为有个0的也符合条件,二维的比三维的少一维-三个平面。别忘了还有x,y,z三个轴*

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

}

return 0;

}