**数论不是一下就能搞定的… 先搞图论。。。**

**欧拉函数：**

对于一个正整数n，小于n且和n互质的正整数的个数，记做：φ(n)，其中φ(1)被定义为1，但是并没有任何实质的意义。

特殊性质：当n为奇数时，φ(2n)=φ(n)。  
  
**完全余数集合：**  
定义小于n且和n互质的数构成的集合为Zn，称呼这个集合为n的完全余数集合。  
  
显然，对于素数p，φ(p)= p - 1.

对于两个素数p、q，他们的乘积n = pq 满足φ(n) =(p-1)(q-1)  
  
        证明：对于质数p,q，满足φ(n) =(p-1)(q-1)  
        考虑n的完全余数集Zn = { 1,2,....,pq -1}  
        而不和n互质的集合由下面三个集合的并构成：  
        1) 能够被p整除的集合{p,2p,3p,....,(q-1)p} 共计q-1个  
        2) 能够被q整除的集合{q,2q,3q,....,(p-1)q} 共计p-1个  
        3) {0}  
        很显然，1、2集合中没有共同的元素，因此Zn中元素个数 ＝ pq - (p-1 + q- 1 + 1) = (p-1)(q-1)  
  
**欧拉定理：**对于互质的整数a和n，有**aφ(n)  ≡ 1 mod n**

{

注：

**同余符号：**

    两个整数a，b，若它们除以整数m所得的余数相等，则称a，b对于模m同余

　　记作 a ≡ b (mod m)

　　读作a同余于b模m，或读作a与b关于模m同余。

　　比如 26 ≡ 14 (mod 12)

}

        证明：  
        首先证明下面这个命题：  
        对于集合Zn={x1,x2,...,xφ(n)}，考虑集合  
        S = {ax1 mod n,ax2mod n,...,axφ(n)mod n}  
        则S = Zn  
        1) 由于a,n互质，xi也与n互质，则axi也一定于p互质，因此  
        任意xi，axi mod n 必然是Zn的一个元素  
        2) 对于Zn中两个元素xi和xj，如果xi ≠ xj  
        则axi mod n ≠ axi mod n，这个由a、p互质和消去律可以得出。  
        所以，很明显，S=Zn  
         
        既然这样，那么  
        （ax1 × ax2×...×axφ(n)）mod n  
         = （ax1 mod n × ax2mod n × ... × axφ(n)mod n）mod n  
         = （x1 × x2 × ... × xφ(n)）mod n  
         考虑上面等式左边和右边  
         左边等于(aφ(n) × （x1 × x2 × ... × xφ(n)）mod n) mod n  
         右边等于x1 × x2 × ... × xφ(n)）mod n  
         而x1 × x2 × ... × xφ(n)）mod n和p互质  
         根据消去律，可以从等式两边约去，就得到：  
          aφ(n)  ≡  1 mod n

**费马定理：**a是不能被质数p整除的正整数，则有 **ap - 1** **≡ 1 mod p**

证明这个定理非常简单，由于φ(p) = p-1，代入欧拉定理即可证明。  
同样有推论：对于不能被质数p整除的正整数a，有**ap** ≡ a mod p

**欧拉函数公式：**

( 1 ) pk 的欧拉函数

对于给定的一个素数 p ， φ(p) = p -1。则对于正整数 n = pk ，

**φ(n) = pk - pk -1**  
证明：  
小于 pk 的正整数个数为 pk - 1个，其中  
和 pk 不互质的正整数有**{p \* 1,p \* 2,...,p \* (pk - 1-1)}** 共计 **pk - 1 - 1** 个  
所以 φ(n) = pk - 1 - (pk - 1 - 1) = pk - pk - 1 。

( 2 ) p \* q 的欧拉函数

假设 p, q是两个互质的正整数，则 p \* q 的欧拉函数为

**φ(p \* q) = φ(p) \* φ(q) ， gcd(p, q) = 1 。**

证明：  
令 n = p \* q ， gcd(p,q) = 1  
根据中国余数定理，有  
Zn 和 Zp × Zq 之间存在一一映射  
**（我的想法是： a ∈ Zp ， b ∈ Zq ⇔ b \* p + a \* q ∈ Zn 。）**  
所以 n 的完全余数集合的元素个数等于集合 Zp × Zq 的元素个数。  
而后者的元素个数为 φ(p) \* φ(q) ，所以有  
φ(p \* q) = φ(p) \* φ(q) 。

( 3 ) 任意正整数的欧拉函数

任意一个整数 n 都可以表示为其素因子的乘积为：

**I**  
**n = ∏ piki** (I 为 n 的素因子的个数)  
**i=1**

**（注：∏是希腊字母，即π的大写形式，在数学中表示求积运算或直积运算，形式上类似于Σ，有时也用来代表圆周率值）**

根据前面两个结论，很容易得出它的欧拉函数为：   
         I                      I  
 Φ(n) = ∏  piki-1(pi-1) = n ∏ (1 - 1 / pi)  
        i=1                    i=1  
对于任意 n > 2，2 | Φ(n) ,因为必存在  pi-1 是偶数。

跟据上面的公式，可以得到关于欧拉函数的递推关系：  
假设素数p能整除n，那么  
如果p还能整除n / p, PHI(n) = PHI(n / p) \* p;  
如果p不能整除n / p, PHI(n) = PHI(n / p) \* (p - 1);  
  
下面的程序是求1到10000之间所有整数的欧拉函数：  
char mark[10000] = {0};   
int prime[1230];  
int size = 0;  
int phi[10000];  
  
int main () {  
    int i, j;  
  
    /\*筛法求素数\*/  
    for (i = 2; i < 10000; i++) {  
        if (!mark[i]) prime[size++] = i;  
  
        for (j = 0; j < size && prime[j] \* i < 10000; j++) {  
            mark[prime[j] \* i] = 1;  
            if (i % prime[j] == 0) break;  
        }  
    }  
    /\*求欧拉函数\*/  
    phi[1] = 1;  
    for (i = 2; i < 10000; i++) {  
        if (!mark[i]) {  
            phi[i] = i - 1;  
            continue;  
        }  
        for (j = 0; j < size && prime[j] \* prime[j] <= i; j++) {  
            if (i % prime[j] == 0) {  
                if (i / prime[j] % prime[j] == 0)  
                    phi[i] = prime[j] \* phi[i / prime[j]];  
                else  
                    phi[i] = (prime[j] - 1) \* phi[i / prime[j]];  
                break;  
            }  
        }  
    }  
    return 0;  
}  
  
从别人那里学到的对求欧拉函数部分的优化，使每个数的欧拉函数只由它的最小素因子求出:  
    phi[1] = 1;  
    for (i = 1; i < 10000; i++) {  
        for (j = 0; j < size && prime[j] \* i <= 10000; j++) {  
            if (i % prime[j] == 0) {  
                phi[prime[j] \* i] = prime[j] \* phi[i];  
                break;  
            }  
            else {  
                phi[prime[j] \* i] = phi[i] \* (prime[j] - 1);  
            }  
        }  
    ｝

自己写的欧拉函数单个数模板 改下N 就行了 。。

#include<stdio.h>

#include<iostream>

using namespace std;

#define N 10010

int mark[N];

\_\_int64 p[N];

int n;

void prime()

{

p[0]=1;

p[1]=2;

for(int i=4;i<N;i+=2)

mark[i]=1;

for(int i=3;i<N;i+=2)

{

if(mark[i]==1)

continue ;

p[++p[0]]=i;

for(int j=2\*i;j<N;j+=i)

mark[j]=1;

}

}

\_\_int64 oula(int n)

{

\_\_int64 sum,tmp,m;

sum=1;

m=n;

for(int i=1;p[i]<=m&&i<=p[0];i++)

{

if(m%p[i]==0)

{

tmp=1;

while(m%p[i]==0)

{

m=m/p[i];

tmp\*=p[i];

}

tmp/=p[i];

sum\*=tmp\*(p[i]-1);

}

}

if(m>1)

{

sum\*=(m-1);// 因为m是大于10000的素数 除不掉 又因为素数的特殊性质的

}

return sum;

}

int main()

{

\_\_int64 tmp;

prime();

while(scanf("%d",&n)!=EOF&&n)

{

printf("%I64d\n",oula(n));// 减一 因为1要去掉

}

return 0;

}