题意（就是因为读错题意而wa了一次）：给一个数字n，范围在[1,2^23-1]，这个n是一系列数字的最小公倍数，这一系列数字的个数至少为2

例如12，是1和12的最小公倍数，是3和4的最小公倍数，是1,2,3,4,6,12的最小公倍数，是12和12的最小公倍数………………

那么找出一个序列，使他们的和最小，上面的例子中，他们的和分别为13,7,28,24……显然最小和为7

这个题目一开始没有头绪，写了一个暴搜程序来找答案，然后一目了然

首先假设我们知道了一系列数字a1，a2，a3……an，他们的LCM是n，那么什么时候他们是最优解呢，当他们两两互质的时候

为了方便我们以两个数来说明问题。

a和b的LCM是n，GCD是m，那么n=a/m\*b ， 它们的和就是sum=a+b;

如果m不为1（即a和b不互质），那么我们为什么不优化一下，将a变为a=a/m呢？，改变后a和b的LCM依然是n，但是他们的和显然减少了

所以我们得到最重要的一个性质，要想a1，a2，a3……an的和最小，要保证他们两两互质，只要存在不互质的两个数，就一定可以近一步优化

【思路】

        假设正整数x是c1,c2...cm的LCM，那么在什么情况下sum{ci | 1<=i<=m}最小？答案是当c1,c2...cm互素的时候，比方说只有两个数a,b. lcm(a,b)=a\*b/gcd(a,b),那么如果a,b不是互素的，也就是说gcd(a,b)!=1，那么可以让c=a/gcd(a,b)作为一个新的数，这样lcm(a,b)=c\*b，且c+b<a+b，和变得更小了. 由数论唯一分解定理，正整数x=p1^a1\*p2^a2\*p3^a3...pn^an，那么pi^ai求和即为结果，注意有几种特殊情况。

        （1）n=1，1=1\*1，应输出2

        （2）n为素数，n=n\*1，应输出n+1

        （3）n只有一个素因子即n=p1^a1，应输出 n+1

        （4）用long long，因为n=0x7fffffff是素数，输出n+1会使int溢出

#include<bits/stdc++.h>

#include <cstdio>

#include <iostream>

#include <cstring>

#include <algorithm>

#include <cmath>

#include <ctime>

using namespace std;

typedef long long ll;

const int INF=1000000007;

bool is\_prime(ll n)

{

if(n<2)return 0;

if(n==2||n==3)

return 1;

if((n&1)==0)return 0;

if(n%6!=1&&n%6!=5)

return 0;

for(ll i=5;i\*i<=n;i+=6)

{

if(n%i==0||n%(i+2)==0)

return 0;

}

return 1;

}

int main()

{

ll n;

int cas(0);

while(cin>>n &&n)

{

if(n==1)

{

printf("Case %d: %d\n",++cas,2);

continue;

}

if(is\_prime(n))

{

printf("Case %d: %lld\n",++cas,n+1);

continue;

}

//分解n

int cnt = 0;//cnt记录素因子的个数

ll ans = 0;

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

{

if (n % i == 0)

{

int tmp = 1;

++cnt;

while (n % i == 0)

{

tmp \*= i;

n /= i;

}

ans += tmp;

}

}

if (n > 1) ans += n;//最后一个素因子可能还没被分解，比如n=10时

if(cnt==1 && n==1)ans++;

printf("Case %d: %lld\n",++cas,ans);

}

return 0;

}