题意：

给出整数 a 和 b ，求区间[b, a] 内的 a 的约数对的个数，a 的约数对（比如[2, 3] 与 [3, 2] 为同一对）。

解法：

主要利用公式:

一个整数ｎ可以表示为若干素数乘积：　n = p1^a1 \* p2^a2\*…\*pm^am;

则 n 的正因数的个数可以表示为： num = (a1+1)\*(a2+1)…(am+1);

但是这个题还给了范围，是要b到a之间的约数对的个数

这样在求出总约束对个数的话还要扫一遍小于b的数，如果能整除a，就要减1

（题目要求b,a之间的约数对，只要约数对有一个不在b,a之间，这个 约数对就不在b,a之间）

最大是1e12，所以质数筛到1e6就可以了

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int maxn=1e6+10;

bool is\_prime[maxn]; //初始化，=false表示是素数

int prime[maxn]; //保存素数

int k;

void Euler(int n)

{

int i, j;

k = 0;

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

{

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

prime[k++] = i;

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

{

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

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

break;

}

}

}

ll solve(ll n)

{

ll sum(1),cnt;

for(int i=0;i<k && prime[i]<n;i++)

{

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

{

cnt=0;

while(n%prime[i]==0)//比如n=p1^a1\*p2^a2....，这里就是在算a1、a2

{

cnt++;

n/=prime[i];

}

sum\*=(cnt+1);

}

}

if(n>1)//如果n>1,说明n之前的质数没有他的因子，说明n本身是个质数，那么它出现了一次，所以乘(1+1)

sum\*=2;

return sum;

}

int main()

{

// freopen("input.txt","r",stdin);

ios::sync\_with\_stdio(false);

cin.tie(0);

Euler(maxn);

int T;

cin>>T;

ll n,m;

for(int o=1;o<=T;o++)

{

cin>>n>>m;

if(n/m<m)

{

printf("Case %d: 0\n",o);

continue;

}

ll ans=solve(n)/2;

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

if(n%i==0)ans--;

printf("Case %d: %Ild\n",o,ans);

}

return 0;

}