两只青蛙在网上相识了，它们聊得很开心，于是觉得很有必要见一面。它们很高兴地发现它们住在同一条纬度线上，于是它们约定各自朝西跳，直到碰面为止。可是它们出发之前忘记了一件很重要的事情，既没有问清楚对方的特征，也没有约定见面的具体位置。不过青蛙们都是很乐观的，它们觉得只要一直朝着某个方向跳下去，总能碰到对方的。但是除非这两只青蛙在同一时间跳到同一点上，不然是永远都不可能碰面的。为了帮助这两只乐观的青蛙，你被要求写一个程序来判断这两只青蛙是否能够碰面，会在什么时候碰面。   
我们把这两只青蛙分别叫做青蛙A和青蛙B，并且规定纬度线上东经0度处为原点，由东往西为正方向，单位长度1米，这样我们就得到了一条首尾相接的数轴。设青蛙A的出发点坐标是x，青蛙B的出发点坐标是y。青蛙A一次能跳m米，青蛙B一次能跳n米，两只青蛙跳一次所花费的时间相同。纬度线总长L米。现在要你求出它们跳了几次以后才会碰面。

Input

输入只包括一行5个整数x，y，m，n，L，其中x≠y < 2000000000，0 < m、n < 2000000000，0 < L < 2100000000。

Output

输出碰面所需要的跳跃次数，如果永远不可能碰面则输出一行"Impossible"

Sample Input

1 2 3 4 5

Sample Output

4

注意exgcd处理的a,b都是正整数

所以要先判断一下m,n的大小，m<n要交换一下，但注意交换m,n的同时x,y也要换

#include <iostream>

#include <cstdio>

#include <cmath>

using namespace std;

typedef long long ll;

long long exgcd(long long a,long long b,long long &x,long long &y)// 求a关于模b的逆元,x即为所求逆元 x,y你在传入的时候随便给，它们的值由函数来算

{

if (b==0)

{

x=1;y=0;

return a;

}

long long ans=exgcd(b,a%b,x,y);//执行完这一步之后，x、y已经变了，不是函数原来传入的xy了

long long tmp=x;

x=y;

y=tmp-a/b\*y;

return ans;

}

int main()

{

ll x,y,m,n,l,a,c,r,x1,gcd;

cin>>x>>y>>m>>n>>l;

if(m<n)

swap(m,n),swap(x,y);

a=m-n;

c=y-x;

gcd=exgcd(a,l,x,y);

x1=x\*c/gcd;

r=l/gcd;

if(c%gcd!=0)

{

printf("Impossible\n");

return 0;

}

else

printf("%lld\n",(x1%r+r)%r);

return 0;

}