G —— 解题报告

题意:本题题意很裸,求模线性方程组在(l, r)区间内的解.

思路:应用中国剩余定理即可.注意本题中,给出的模数不一定两两互质,所以要进行处理,采用合并方程的方法即可.

输入的数据中可能会出现模得结果比模数大的情况,如 x = 5 (mod 4),注意特判.

而输出(l, r)区间内的解的时候,注意解数超过 100 时,只输出前 100 个,但是输出解的个数时还是要输出所有解的个数,不能输出 100.

数据范围很大,虽然实际计算结果不会超过 long long,但是根据实现方法不同,运算过程中还是可能超过 long long 的,所以直接 java 搞起吧,python 不知为什么一直提示我 Non-zero Exit Code …

```
import java.io.*;
import java.util.*;
import java.math.*;

public class Main
{
    static BigInteger l, r;
    public static void main(String[] args)
    {
        Scanner cin = new Scanner(new BufferedInputStream(System.in));
        crt gao = new crt();
        BigInteger b[] = new BigInteger[110];
        BigInteger m[] = new BigInteger[110];
        int num;
        while(cin.hasNext())
        {
            boolean flag=false;
            num = cin.nextInt();
            for(int i = 0; i < num; i++)
                m[i] = cin.nextBigInteger();
            for(int i = 0; i < num; i++)
            {
                b[i] = cin.nextBigInteger();
                if(b[i].compareTo(m[i]) >= 0)
                    flag=true;
            }
            l = cin.nextBigInteger();
            r = cin.nextBigInteger();
            if(flag == true)
            {
                System.out.println("0");
                continue;
            }
            gao.solve(b, m, num);
        }
    }
    public static class crt
    {
        BigInteger x, y;
        BigInteger exgcd(BigInteger a, BigInteger b)
        {
            if(b.equals(BigInteger.ZERO))
            {
                x = BigInteger.ONE;
                y = BigInteger.ZERO;
                return a;
            }
            BigInteger d = exgcd(b, a.mod(b)), tp = x;
            x = y;
            y = tp.subtract(a.divide(b).multiply(y));
            return d;
        }
        void solve(BigInteger b[], BigInteger m[], int num)
        {
            int i;
            boolean flag=false;
            BigInteger m1 = m[0], m2, b1 = b[0], b2, bb, d, t, k;
            for(i = 1; i < num; i++)
            {
                m2 = m[i];
                b2 = b[i];
                bb = b2.subtract(b1);
                d = exgcd(m1, m2);
                if(false == bb.mod(d).equals(BigInteger.ZERO))
                {
                    flag = true;
                    break;
                }
                k = bb.divide(d).multiply(x);
                t = m2.divide(d);
                if(t.compareTo(BigInteger.ZERO) < 0)
                    t = t.negate();
                k = (k.mod(t).add(t)).mod(t);
                b1 = b1.add(m1.multiply(k));
                m1 = m1.divide(d).multiply(m2);
            }
            if(flag)
```

```
			{
				System.out.println("0");
				return;
			}
			if(b1.compareTo(l) < 0)
			{
				BigInteger tmp = l.subtract(b1).subtract(BigInteger.ONE);
				b1 = b1.add((tmp.divide(m1).add(BigInteger.ONE)).multiply(m1));
			}
			if(b1.compareTo(r) > 0)
			{
				System.out.println("0");
				return;
			}
			BigInteger cnt = ((r.subtract(b1)).divide(m1)).add(BigInteger.ONE);
			System.out.println(cnt);
			System.out.print(b1);
			int kk = 1;
			for(b1 = b1.add(m1); b1.compareTo(l) >= 0 && b1.compareTo(r) <= 0 && kk < 100;
b1 = b1.add(m1), kk++)
				System.out.print(' '+b1.toString());
			System.out.println();
		}
	}
}
```