链接：[https://www.nowcoder.com/acm/contest/96/D](https://www.nowcoder.com/acm/contest/96/D" \t "_blank)  
来源：牛客网

 离散数学中有种名叫“哈斯图”的东西。

    在这题中，你们需要计算的是一些正整数在偏序关系“整除”下的哈斯图的边数。**用大白话讲**，在偏序关系“整除”下的哈斯图，就是把一个个正整数看成一个个图的节点，某些节点之间有边。连边的规则是这样的：对于任意两个正整数a和b(a<b)来说，如果b%a==0,并且不存在一个正整数c（a<c<b），使得条件b%c==0和c%a==0同时成立，那么我们就在节点a和节点b之间连一条边。

    现在问题是，给你们2个数L，R(1<=L,R<=1e6)。求由L，L+1，L+2...R这R-L+1个正整数在偏序关系“整除”下的哈斯图的边数。

    比如L=1，R=4,节点的组合有(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)。组合（1,2），（1,3），（2,4）可以连边。（1,4）因为中间存在c=2，不符合连边条件。所以当L=1，R=4的时候，这个哈斯图有3条边。

**输入描述:**

多组输入，不超过1000组数据  
每组数据一行，包含2个正整数L和R，中间由空格分开。

**输出描述:**

每组数据输出一行，包含一个整数表示哈斯图的边数。

示例1

**输入**

1 4

4 10

1 10

**输出**

3

2

11

这道题按照题意，就是想到求从a到b的每个数i的素数倍有多少个，加和就行了之前要用个素筛。非常简单，代码：(区间[a,b],p[]是素数数组)

for(i=a;i<=b/2;i++){

for(j=0;p[j]\*i<=b;j++);

ans+=j;

}

数据量1e6,on复杂度对于10组以内数据没问题，但是数据可能达到1000组，那么我们就要考虑其他方法

我还是想到了思路，考虑求[1,b]减去[1,a-1]的，1到b的很好求，用类似素筛的思路算贡献，预处理一个数组代表每个数i都是几个数的素数倍，设它是num[i]

怎么算贡献，当然是枚举每个数和素数，1的所有素数倍，2的素数倍，3的素数倍，。。。枚举每个数的素数倍，当然乘积要小于n,这个枚举的时间复杂度和哎式素筛相当，是nlog(log(n)),不要以为它是n\*n或者n\*log(n);

这算完num[]数组，求一下前缀和就是[1,i]的边数，设它是F[i]

我们要求的是[a,b]边数而不是[1,b]边数

那么减去什么呢，就有点不好做了

对于[1,10]

1: 2 3 5 7

2: 4 6 10

3 : 6 9

4: 8

5 10

共11个

而[4,10]:

1:2 3 5 7

2 4 6 10

3 6 9

4 8

5 10

只有4 8 ,5 10两个

可以找出规律是去掉

前a-1个的每一对，这里是去掉：

1: 2 3 5 7

2: 4 6 10

3 : 6 9

设pi[x]是小于等于x的素数个数

对于[a,b]去掉的个数是：

Sum(pi[b/i]),i<=a-1

这里是pi[10/1]+pi[10/2]+pi[10/3]= pi[10]+pi[5]+pi[3]

以上是数论常规套路，代码已经搞得很熟练了，思路也没多久就想出来了

但是怎么快速求Sum(pi[b/i]),i<=a-1，复杂度不还是on么？？？

这里怎么优化呢？这个代码也不难想，但我写了好久，虽然以前见过但是没总结成模板，现在看来很有必要，这个代码很容易出错

这个算法用sqrt(n)时间求出，Sum(b/i)，当然求Sum(pi[b/i])也是没问题的，

对于求到Sum(b/i),i<a,也就是前a项，也是能做到的。

在分解质因数扩展专门讲一下写个模板，思路不难，主要是代码细节处理