**Find the good sets!**

You are given array a consisting of n distinct integers. A set s of numbers is called good if you can rearrange the elements in such a way that each element divides the next element in the order, i.e. si | si + 1, such that i < |s|, where |s| denotes size of set |s|.

Find out number of distinct good sets that you can create using the values of the array. You can use one value only once in a particular set; ie. a set cannot have duplicate values. Two sets are said to be different from each other if there exists an element in one set, which does not exist in the other.

As the answer could be large, print your answer modulo 10^9 + 7.

#include<iostream>

#include<vector>

using namespace std;

int f[1000000];

//sieve for factors

void sieve(int n) {

for(int i=0;i<n+1;i++)

f[i]=0;

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

if(f[i]==0){

for(int j=i;j<=n;j+=i){

f[j]++;

}

}

}}

int main() {

int t;

cin>>t;

vector<pair<int,int>> m;

int \*g=new int[t];

for(int i=0;i<t;i++) {

int a,b,n;

cin>>a>>b>>n;

pair<int,int> p;

p.first=a;

p.second=b;

g[i]=n; //storing n in array

m.push\_back(p); //storing a and b in vector

}

int max=m[0].second; //finding max b of array

for(int i=1;i<t;i++) {

if(m[i].second>max)

max=m[i].second;

}

sieve(max);

for(int i=0;i<t;i++) {

pair<int,int> p=m[i];

int d=p.first;

int e=p.second;

int count=0;

for(int j=d;j<=e;j++) {

if(f[j]==g[i])

count++;

}

cout<<count<<endl;

}

return 0;

}