//7 이하 -> 7\*6\*5\*4\*3\*2\*1 => 7자리수 각각 소수인지 판별 ->1000000 각각 30000

//+ dp로 소수 여러개 7자리 최대값은 9999999 100000000까지

// import java.util\*;

import java.util.\*;

import java.io.\*;

import java.lang.\*;

class Solution {

static char[] str\_arr;

static Set<Integer> list = new TreeSet();

static int str\_len;

public int solution(String numbers) {

int answer = 0;

str\_len = numbers.length();

str\_arr = new char[str\_len];

boolean[] visited= new boolean[str\_len];

boolean[] prime = new boolean[10000000];

//배열에 문자1개씩 넣기

for(int i = 0; i<str\_len; i++){

str\_arr[i]=(numbers.charAt(i));

}

// System.out.println(Integer.parseInt("00001234"));

//stringbuilder로 붙이기

// StringBuilder sb = new StringBuilder();

char[] result = new char[str\_len];

prime[1]=true;

for(int i = 2; i<=(int)Math.sqrt(prime.length); i++){

if(!prime[i]){

for(int j=2; i\*j<prime.length; j++){

prime[i\*j]=true;

}

}

}

for(int i = 1; i<=str\_len ; i++){

permutation(i,0,result,visited);

}

System.out.println(list.size());

for(Integer integer : list){

// System.out.println(integer);

// // System.out.println("integer");

if(!prime[integer]){

answer++;

}

}

return answer;

}

static void permutation(int r, int cur, char[] result, boolean[] visited){

if(r==cur){

StringBuilder sb = new StringBuilder();

for(int i=0; i<result.length; i++){

if(sb.length()==0 && result[i]=='0'){

continue;

}

if(result[i]!=0){

sb.append(result[i]);

}

// System.out.println("result: " + sb);

// System.out.println("len: " + sb.length());

}

if(sb.length()!=0){

String str2 = sb.toString();

list.add(Integer.parseInt(str2));

}

//

// System.out.println(list.size());

// System.out.println("sb : " + sb.length());

// if(sb.length() == 0){

// String str2 = sb.toString();

// System.out.println(str2);

// if(!str2.equals(str2)){

// list.add(Integer.parseInt(str2));

// System.out.println(list.size());

// }

// }

}else{

for(int i = 0; i<str\_len ; i++){

if(!visited[i]){

visited[i]=true;

result[cur] = (str\_arr[i]);

permutation(r,cur+1,result,visited);

visited[i]=false;

}

}

}

}//함수 끝

}

//순열 디버그 --------------- 성공

// for(int i=0; i<result.length; i++){

// System.out.print(result[i]+", ");

// }

// System.out.println();

//--------------