https://programmers.co.kr/learn/courses/30/lessons/42839

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class Solution {

static String[] str\_arr;

static TreeSet<Integer> tree = new TreeSet();

static int str\_len;

public int solution(String numbers) {

int answer = 0;

str\_arr = numbers.split("");

str\_len = numbers.length();

System.out.println(str\_arr.length);

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

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

String[] result = new String[i];

permu(i,0,result,visited);

}

// System.out.print(tree.last());

boolean[] prime = new boolean[tree.last()+1];

// for(int i =2; i<=(tree.last()); i++){

for(int i =2; i<=Math.sqrt(tree.last()); i++){

prime[0]=true;

prime[1]=true;

if(!prime[i]){

for(int j =2; i\*j<=tree.last(); j++){

prime[i\*j]=true;

}

}

}

for(Integer val : tree){

// System.out.println(val);

if(!prime[val]){

System.out.println(val);

answer++;

}

}

return answer;

}

static void permu(int r, int cur, String[] result, boolean[] visited){

if(r == cur){

StringBuilder sb = new StringBuilder();

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

if(!result[i].equals("")){

sb.append(result[i]);

}

}

// System.out.println(sb);

tree.add(Integer.parseInt(sb.toString()));

}else{

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

if(!visited[i]){

visited[i]=true;

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

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

visited[i]=false;

}

}

}

}

}