import java.util.Scanner;

import java.util.ArrayList;

import java.util.Iterator;

//example of java synchronized method

class Table{

// private static ArrayList<Integer> resary,cntary= new ArrayList<>();

// private static int[] arr = new int[2];

static int maxval=Integer.MIN\_VALUE,resval=0;

public synchronized void printTable(int a,int b){//synchronized method

int cnt,max=Integer.MIN\_VALUE,result=0;

for(int i=a;i<=b;i++)

{

cnt=0;

for(int j=2;j<i;j++)

{

if(i%j==0)

{

// arr.add(j);

cnt++;

}

}

if(cnt>max)

{

// res.clear();

result = i;

max = cnt;

// Iterator it = arr.iterator();

// while(it.hasNext())

// {

// res.add((Integer)it.next());

// }

}

// arr.clear();

}

if(maxval<=max)

{

maxval=max;

resval=result;

}

// Iterator rt = res.iterator();

// System.out.printf("The number that has maximum number of divisors from %d to %d is : %d",a,b,result);

// System.out.println();

// System.out.println("Count = "+max);

// System.out.println("The divisors are...");

// while(rt.hasNext())

// {

// System.out.print(rt.next()+" ");

// }

}

public int[] getFinalResult()

{

// int max = Integer.MIN\_VALUE,temp,index=0,i=0;

// Iterator it = cntary.iterator();

// while(it.hasNext())

// {

// Integer iv = (Integer)it.next();

// temp = iv.intValue();

// if(max<temp)

// {

// max = temp;

// index=i;

// }

// i++;

// }

return new int[]{maxval,resval};

}

}

class MyThread extends Thread{

Table t;

int a,b;

MyThread(Table t,int a,int b){

this.t=t;

this.a=a;

this.b=b;

}

public void run(){

t.printTable(a,b);

}

// public void callConstructor(Table t,int a,int b)

// {

// this(t,a,b);

// }

}

public class TestSynchronization2{

public static void main(String args[]) throws Exception{

Scanner sc = new Scanner(System.in);

int value, kvalue;

int[] result = new int[2];

Table obj = new Table();//only one object

System.out.println("Enter the value: ");

value = sc.nextInt();

MyThread[] t = new MyThread[10];

//1000

kvalue = value/10; //100

int j=1,k=kvalue;

for(int i=0;i<10;i++)

{

t[i] = new MyThread(obj,j,k);

t[i].start();

t[i].join();

j+=kvalue;//1 101

k+=kvalue;//100 200

}

result = obj.getFinalResult();

System.out.printf("Result: %d Count: %d \n",result[1],result[0]);

}

}