1. Create profile class for VIT students with the following details:

a. **Name (should contain only alphabets –No numbers and special characters)**
b. **Register Number in the format (two numbers three characters four numbers)**
c. **Age (should be greater than 18 and less than 21)**
d. **Phone number (10 numbers).**

Write appropriate Exception handling block(separate block for each type of exception-4 user defined exception) to handle invalid data entry for all data items. If the valid data is entered construct the email id and password of the student and display Email ID format : first namefour digityear @ vitstudent.ac.in and password is "uppercase of first four characters of name+last four numbers of regnumber+@"

CODE

```
package lab7;

import java.io.*;
import java.util.*;
import java.util.regex.Pattern;
import java.util.regex.Matcher;

class NameException extends Exception
{
    private String name;
    NameException(String s)
    {
        super(s);
        this.name=s;
    }
    String getName()
    {
        return this.name;
    }
}

class RegisterNumberException extends Exception
{
    private String regno;
    RegisterNumberException(String s)
    {
        super(s);
        this.regno=s;
    }
    String getRegNo()
    {
        return this.regno;
    }
}

class AgeException extends Exception
{
    private String age;
    AgeException(String s)
    {
        super(s);
        this.age=s;
    }
    String getAge()
    {
        return this.age;
    }
}

class PhoneNumberException extends Exception
{
    private String phno;
    PhoneNumberException(String s)
    {
        super(s);
        this.phno=s;
    }
    String getPhoneNumber()
    {
        return this.phno;
    }
}
class Profile{
        String name;
        String regno;
        int age;
        long phone;
        String email;
        String password;

        Profile(String name,String regno,int age,long phone,String email,String 
password)
        {
                this.name=name;
                this.regno=regno;
                this.age=age;
                this.phone=phone;
                this.email=email;
                this.password=password;
        }

        public void display()
        {
                System.out.println("Name: "+name+" Register number: "+regno+" Age: 
"+age+" Phone number: "+phone+" Email: "+email+" Password: "+password);
        }
}
```

```
public class ques1 {
      public static void main(String[] args) {
          try {
             String name;
             String regno;
             int age;
             long phone;
             String email;
             String password;

             Scanner sc=new Scanner(System.in);
                   System.out.println("Enter the student name: ");
                   name=sc.nextLine();
                   Pattern p1=Pattern.compile("[a-zA-Z]");
                   Matcher m1=p1.matcher(name);
                   boolean matchfound1=m1.find();
                   if(!matchfound1)
                   {
                         throw new NameException(name);
                   }
                   System.out.println("Enter the student registration number: ");
                   regno=sc.nextLine();
                   Pattern p2=Pattern.compile("[0-9]{2}[a-zA-Z]{3}[0-9]{4}");
                   Matcher m2=p2.matcher(regno);
                   boolean matchfound2=m2.find();
                   if(!matchfound2)
                   {
                         throw new RegisterNumberException(regno);
                   }
                   System.out.println("Enter the student age: ");
                   age=sc.nextInt();
                   if(age<18||age>21)
                   {
                         throw new AgeException(String.valueOf(age));
                   }
                   System.out.println("Enter the student phone number: ");
                   phone=sc.nextLong();
                   String ph=String.valueOf(phone);
                   if(ph.length()!=10)
                   {
                         throw new PhoneNumberException(ph);
                   }

                   String name_arr[]=name.split(" ");
                   email=name_arr[0]+"20"+regno.substring(0,2)+"@vitstudemt.ac.in";

                   password=name_arr[0].toUpperCase()+regno.substring(5)+"@";

                   Profile p=new Profile(name,regno,age,phone,email,password);

                   p.display();

          }
          catch(NameException e){
             System.out.println(e.getName()+" is invalid input!\nName should contain
only alphabets!");
             System.exit(0);
           }
           catch(RegisterNumberException e){
             System.out.println(e.getRegNo()+" is invalid input!\"Register number
should contain two numbers three characters four numbers!");
             System.exit(0);
           }
           catch(AgeException e){
             System.out.println(e.getAge()+" is invalid input!\nAge should be between
18 and 21!");
             System.exit(0);
           }
           catch(PhoneNumberException e){
             System.out.println(e.getPhoneNumber()+" is invalid input!\nPhone number
should contain 10 numbers!");
             System.exit(0);
           }
         }
}
```

2. Accept an integer array from the user and create three threads for the following
    a. Check whether the number is odd or even. Print ODD/ EVEN
    b. Check whether it is prime or not Print PRIME/ NOT PRIME
    c. Find the sum of digits of a number

Run the three threads for each element of a number print the result with the thread name in the following pattern:

1. For first element ODD/ EVEN
   PRIME/NOT PRIME
   Sum of digits
   For second element ODD/ EVEN
   PRIME/NOT PRIME
   Sum of digit
   And so on…..

2. For all elements print ODD/EVEN
   For all elements print PRIME/NOT PRIME
   For all elements Sum of digits.

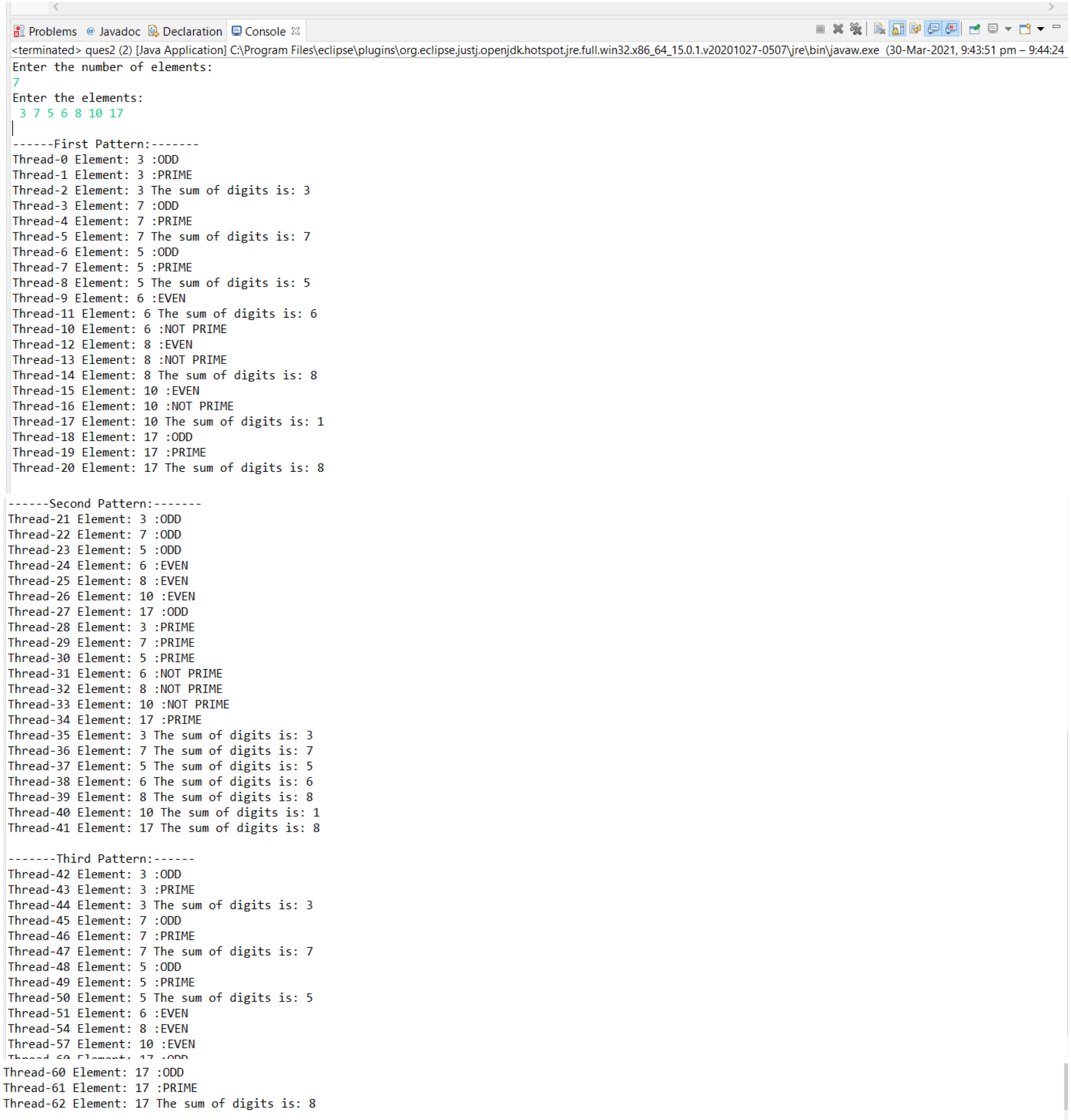
3. For each element if the number is ODD then check for PRIME . IF it is prime then find the sum of digits.
   If it is even print " EVEN" then continue
   IF it is odd and Not prime then print "NOT PRIME" then continue.

CODE

```
import java.util.*;
class evenodd extends Thread{
private int number;
public int flag;
evenodd(int num){
number=num;
}
public void run(){
if(number%2==0){
System.out.println(this.getName()+" Element: "+number+" :EVEN");
}
```

```
else{
System.out.println(this.getName()+" Element: "+number+" :ODD");
flag=1;
}}}
class prime extends Thread{
private int number;
public int flag;
prime(int num){
number=num;
}
public void run(){
int ct=0,i;
for(i=1;i<=number;i++){
if(number%i==0){
ct++;
}}if(ct==2){
System.out.println(this.getName()+" Element: "+number+" :PRIME");
flag=1;
}
else{
System.out.println(this.getName()+" Element: "+number+" :NOT PRIME");
}}}
class sum extends Thread{
private int number;
sum(int num){
number=num;
}
public void run(){
int sum=0;
int n;
n=number;
while(n>0){
sum+=n%10;
n=n/10;
}
System.out.println(this.getName()+" Element: "+number+ " The sum of digits is: 
"+sum);
}}
public class ques2 {
public static void main(String[] args){
Scanner in=new Scanner(System.in);
System.out.println("Enter the number of elements:");
int n=in.nextInt();
int i;
int arr[]=new int[n];System.out.println("Enter the elements:");
for(i=0;i<n;i++){
arr[i]=in.nextInt();}
System.out.println();
System.out.println("------First Pattern:-------");
for(i=0;i<n;i++){
Thread T1=new evenodd(arr[i]);
Thread T2=new prime(arr[i]);
Thread T3=new sum(arr[i]);
try{
 T1.start();
 T2.start();
 T3.start();
 T1.join();
 T2.join();
 T3.join();
 }
catch(InterruptedException e)
{System.out.println("ERROR: "+e);}}
System.out.println();
System.out.println("------Second Pattern:-------");
for(i=0;i<n;i++){
Thread T1=new evenodd(arr[i]);
T1.start();
try{
T1.join();}
catch(InterruptedException e){System.out.println("ERROR: "+e);}}
for(i=0;i<n;i++){
Thread T2=new prime(arr[i]);
T2.start();
try{T2.join(); }
catch(InterruptedException e){System.out.println("ERROR: "+e);}}
for(i=0;i<n;i++){
Thread T3=new sum(arr[i]);
T3.start();
try{
T3.join();}
catch(InterruptedException e){System.out.println("ERROR: "+e);}}
System.out.println();
System.out.println("-------Third Pattern:------");
for(i=0;i<n;i++){
evenodd T1=new evenodd(arr[i]);
prime T2=new prime(arr[i]);
Thread T3=new sum(arr[i]);
T1.start();
try{
T1.join();}
catch(InterruptedException e){ System.out.println("ERROR: "+e);}
if(T1.flag==1){
T2.start();
try{
T2.join();}
catch(InterruptedException e){ System.out.println("ERROR: "+e);}
if(T2.flag==1){
T3.start();
try{
T3.join();}
          catch(InterruptedException e){ System.out.println("ERROR: "+e);}}}}}}
```

OUTPUT



```
<terminated> ques2 (2) [Java Application] C:\Program Files\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.1.v20201027-0507\jre\bin\javaw.exe  (30-Mar-2021, 9:43:51 pm – 9:44:24
Enter the number of elements:
7
Enter the elements:
 3 7 5 6 8 10 17

------First Pattern:-------
Thread-0 Element: 3 :ODD
Thread-1 Element: 3 :PRIME
Thread-2 Element: 3 The sum of digits is: 3
Thread-3 Element: 7 :ODD
Thread-4 Element: 7 :PRIME
Thread-5 Element: 7 The sum of digits is: 7
Thread-6 Element: 5 :ODD
Thread-7 Element: 5 :PRIME
Thread-8 Element: 5 The sum of digits is: 5
Thread-9 Element: 6 :EVEN
Thread-11 Element: 6 The sum of digits is: 6
Thread-10 Element: 6 :NOT PRIME
Thread-12 Element: 8 :EVEN
Thread-13 Element: 8 :NOT PRIME
Thread-14 Element: 8 The sum of digits is: 8
Thread-15 Element: 10 :EVEN
Thread-16 Element: 10 :NOT PRIME
Thread-17 Element: 10 The sum of digits is: 1
Thread-18 Element: 17 :ODD
Thread-19 Element: 17 :PRIME
Thread-20 Element: 17 The sum of digits is: 8

------Second Pattern:-------
Thread-21 Element: 3 :ODD
Thread-22 Element: 7 :ODD
Thread-23 Element: 5 :ODD
Thread-24 Element: 6 :EVEN
Thread-25 Element: 8 :EVEN
Thread-26 Element: 10 :EVEN
Thread-27 Element: 17 :ODD
Thread-28 Element: 3 :PRIME
Thread-29 Element: 7 :PRIME
Thread-30 Element: 5 :PRIME
Thread-31 Element: 6 :NOT PRIME
Thread-32 Element: 8 :NOT PRIME
Thread-33 Element: 10 :NOT PRIME
Thread-34 Element: 17 :PRIME
Thread-35 Element: 3 The sum of digits is: 3
Thread-36 Element: 7 The sum of digits is: 7
Thread-37 Element: 5 The sum of digits is: 5
Thread-38 Element: 6 The sum of digits is: 6
Thread-39 Element: 8 The sum of digits is: 8
Thread-40 Element: 10 The sum of digits is: 1
Thread-41 Element: 17 The sum of digits is: 8

-------Third Pattern:------
Thread-42 Element: 3 :ODD
Thread-43 Element: 3 :PRIME
Thread-44 Element: 3 The sum of digits is: 3
Thread-45 Element: 7 :ODD
Thread-46 Element: 7 :PRIME
Thread-47 Element: 7 The sum of digits is: 7
Thread-48 Element: 5 :ODD
Thread-49 Element: 5 :PRIME
Thread-50 Element: 5 The sum of digits is: 5
Thread-51 Element: 6 :EVEN
Thread-54 Element: 8 :EVEN
Thread-57 Element: 10 :EVEN
Thread-60 Element: 17 :ODD
Thread-61 Element: 17 :PRIME
Thread-62 Element: 17 The sum of digits is: 8
```