PROJECT REPORT

R-S-A ALGORITHM

Submitted to: Sir Umer Ramzan

Submitted By:

Hamza Aslam

Project Name:

RSA Rivest-Shamir-Adleman Algorithm

Project Scope:

RSA (Rivest–Shamir–Adleman) is an algorithm used by modern computers to encrypt and decrypt messages. It is an asymmetric cryptographic algorithm. It is particularly useful for sending information over an insecure network such as the internet. RSA algorithm is safe and secure for transmitting confidential data. Cracking RSA algorithm is very difficult as it involves complex mathematics. Sharing public key to users is easy.

Platform:

Eclipse.

Language:

JAVA

Implementation:

Code 1:

```
Scanner sc= new Scanner(System.in);

System.out.print("Enter the size of array greater than 1: ");
int size = sc.nextInt();
while(size <=1){
   System.out.print("Enter the size of array greater than 1: ");
   size = sc.nextInt();
}//while
```

Input of array size for prime numbers which should be greater than 1.

Code 2:

```
int[] array = new int[size];
for(int i = 0; i < size; i++){
System.out.print("Enter prime number for index [" + i + "] :  ");
int number = sc.nextInt();
if(isPrime(number)) {
array[i] = number;
}//if

while(!(isPrime(number))){
System.out.print(number + " is not a prime number \nEnter Prime number at index [ " + i + " ]:  ");
number= sc.nextInt();
if(isPrime(number)) {
array[i] = number;
}//if
}//while
}//for
```

Fill Array with prime numbers and check that the number should be prime number still checking the number should be prime.

Code 3:

```
public static  boolean isPrime(int num){
        if(num <= 1){
            return false;
        }
        for(int i=2;i<=num/2;i++){
            if((num%i)==0)
                return  false;
        }
        return true;
    }//isPrime
```

This is method for checking the input number is prime or not. If it is it return True else False.

Code 4:

```
  System.out.println("_____");
      System.out.print("Step1: Prime Numbers{ ");
      for(int i = 0; i < size; i++){
      System.out.print(array[i] + ",");
      }//for
      System.out.println("}");
// Step 1 Select largest prime numbers P and Q.
      int p = array[0];
      int q = array[1];
         if(p<q){
            int temp=p;
            p=q;
            q=temp;
         }//if

      for(int i=2; i<size; i++){
         if (array[i] > p){
            q = p;
            p = array[i];
         }else if (array[i] > q && array[i] != p){
            q = array[i];
         }//else-if
      }//for
      System.out.println("P = "+ p);
      System.out.println("Q = "+ q);
      System.out.println("_____");
```

Selection of two largest prime numbers from the array and store in p and q , Print these two prime numbers.

Code 5:

```
//Step 2 n = p*q

         int n = p * q;
         System.out.println("Step 2 : The Value of n =" + n);
      System.out.println("_____");
```

Perform step 2 of RSA Algorithm the multiplication of two largest prime numbers

Code 6:

```
//Step 3 Euler's Totiont

        int eulerTotiont = (p - 1) * (q - 1);
        System.out.println("Step 3: Euler Totiont = " + eulerTotiont);
    System.out.println("_____");
```

Calculation of euler totient by the fornula. And store then print the value.

Code 7:

```
        int e;
//Step 4 choose value of e. eg. 1<e<eulerTotiont(n) & gcd (eulerTotiont(n),e) = 1
        for (e = 2; e < eulerTotiont; e++){
            if (gcd(e, eulerTotiont) == 1){
                break;
            }//if
        }//for
        System.out.println("Step 4: Value of e = " + e);
    System.out.println("_____");
```

Calculate the value of e through a loop.

Code 8:

```
    public static int gcd(int e, int eulerTotiont){
        if (e == 0){
            return eulerTotiont;
        }else{
            return gcd(eulerTotiont % e, e);
        }}//gcd-recursive Method
```

The recursive method for checking the gcd value of e and euler totient. And return the value.

Code 9:

```
int d = 0;
   int i;
for (i = 0; i <= 9; i++){
    int x = 1 + (i * eulerTotiont);
    if (x % e == 0){
        d = x / e;
        break;
    }//if
}//for
System.out.println("Step 5: Value of d = " + d);
```

Calculate the value of d through the loop.

Code 10:

```
System.out.println("Public Key[" + e + "," + n +"]");
System.out.println("Private Key[" + d + "," + n +"]");

System.out.println("_____");
```

The printing of two value p and q;

Code 11:

```
System.out.print("Enter Message =");
String letter= sc.next();
int [] position= findPosition(letter);
String text = "";
for(int j = 0 ;j< position.length; j++){
text = text + position[j];
}
```

Input of users message that to be encrypt

Code 12:

```
public static int[] findPosition(String inputLetter){
int [] position = new int [inputLetter.length()];
for(int i =0; i<inputLetter.length(); i++){
char text_charcter= Character.toLowerCase(inputLetter.charAt(i));
int Value= (int)text_charcter;
position[i]= (int) (Value-96)-1;
}
return position;
}//findPosition
```

This method should return the numeric value of each character of message

Code 13:

```
System.out.println("_______Encryption Cipher_______");

   long cipher = (long) ((Math.pow(num,e)) % n);
   System.out.println("Encrypted Text = " + cipher);
System.out.println("_____________________________________");
```

The encryption of the message through the the formula

Code 14:

```
System.out.println("_______ Decryption Message_______");

   long plain = (long) (Math.pow(cipher, d) % n);
   System.out.println("Decrypted Text = " + plain);
System.out.println("_____________________________________");
```

The Decryption of the cypher text and print.

Test Cases:

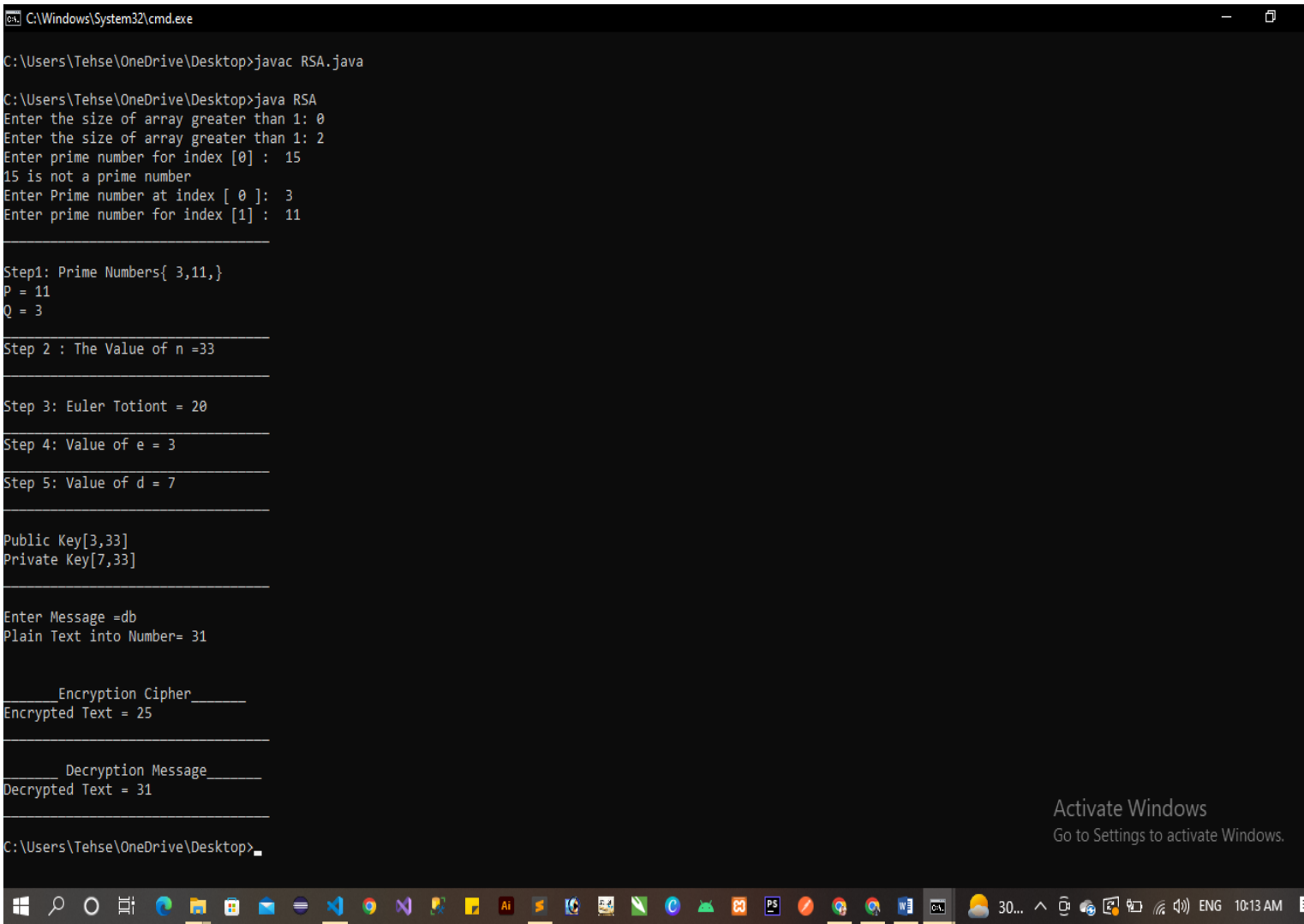
RSA Algorithm
Input Values and code responses:
Array size > 1 = 0 //Error
Array size > 1 = 2 //Array created with size 2

Prime Numbers
Input1 = 15 // error not a prime number
Input 1 = 3
Input2 = 11

Display the calculated values n , e, d

Enter Text: DB
Text positions: 31

Encrypted message: 25
Decrypted message: 31



Test Cases:

RSA Algorithm
Input Values and code responses:
Array size > 1 = 3 //Array created with size 3

Prime Numbers
Input1 = 11
Input 1 = 3
Input2 = 2

Display the calculated values n , e, d= 33, 7 , 3
Enter Text: CD
Text positions: 23
Encrypted message: 23
Decrypted message: 23

```
C:\Windows\System32\cmd.exe
_____________________________________

C:\Users\Tehse\OneDrive\Desktop>javac RSA.java

C:\Users\Tehse\OneDrive\Desktop>java RSA
Enter the size of array greater than 1: 3
Enter prime number for index [0] :  11
Enter prime number for index [1] :  2
Enter prime number for index [2] :  3
_____________________________________

Step1: Prime Numbers{ 11,2,3,}
P = 11
Q = 3
_____________________________________
Step 2 : The Value of n =33
_____________________________________

Step 3: Euler Totiont = 20
_____________________________________
Step 4: Value of e = 3
_____________________________________
Step 5: Value of d = 7
_____________________________________

Public Key[3,33]
Private Key[7,33]
_____________________________________

Enter Message =cd
Plain Text into Number= 23


_______Encryption Cipher_______
Encrypted Text = 23
_____________________________________

_______ Decryption Message_______
Decrypted Text = 23
_____________________________________

C:\Users\Tehse\OneDrive\Desktop>cls
```

Test Case:
RSA Algorithm
Input Values and code responses:
Array size > 1 = 4 //Array created with size 4

Prime Numbers
Input1 = 13
Input2 = 17
Input3 = 19
Input 4 = 11

Display the calculated values n , e, d= 323, 5 , 173
Enter Text: dc
Text positions: 32
Encrypted message: 223
Decrypted message: 0 // infinity due to large value