# **PRACTICAL -8**

**Aim:** Write a program to implement RSA asymmetric (public key and private key)-Encryption.

public class RSA {

public static boolean prime(int n) {

for (int i = 2; i < n / 2; i++) {

if (n % i == 0) {

return true;

}

}

return false;

}

public static String Encryption(int plainNumber, int p, int q) {

int n = p \* q;

int fiy = (p - 1) \* (q - 1);

int e = 0;

for (int i = 2; i < fiy; i++) {

if (!prime(i)) {

if (fiy % i != 0) {

e = i;

break;

}

}

}

int ans = (int) (((double) Math.pow(plainNumber, e)) % n);

return String.valueOf(ans);

}

public static String Decryption(int ct, int p, int q) {

int n = p \* q;

int fiy = (p - 1) \* (q - 1);

int e = 1;

int k = 1;

int j = 1;

for (int i = 2; i < fiy; i++) {

if (!prime(i)) {

if (fiy % i != 0) {

e = i;

break;

}

}

}

while (true) {

if ((j \* fiy + 1) % e == 0) {

k = j;

break;

}

j++;

}

int temp = ((k \* fiy + 1) / e);

double ans = (Math.pow(ct, temp)) % n;

return String.valueOf(ans);

}

public static void main(String[] args) {

System.out.println("Plain Text : " + 3);

System.out.println("Encryption : " + Encryption(3, 5, 7));

System.out.println("Decryption : " + Decryption(33, 5, 7));

}

}

**Output** (Screenshot):

Signature of Faculty:\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: