Practical 1

Q1) Demonstrate Caesar Cipher.

Ans:

caesar\_cipher.java

/\*

caesar\_cipher.java

Author: Jagrut Gala

Date: 01-07-2021

Objective: Demonstrate Caesar Cipher.

\*/

import java.io.\*;

class caesar\_cipher {

caesar\_cipher() {}

String encrypt(String str){

String cipher\_text= "";

str= str.toLowerCase();

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

if(str.charAt(i)== ('x')){

cipher\_text+= "a";

} else if(str.charAt(i)== ('y')){

cipher\_text+= "b";

} else if(str.charAt(i)== ('z')){

cipher\_text+= "c";

} else {

char ch= str.charAt(i);

cipher\_text+= (char) (ch+ 3);

}

}

return(cipher\_text);

}

String decrypt(String str){ // yes

String plain\_text= "";

str= str.toLowerCase();

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

if(str.charAt(i)== ('a')){

plain\_text+= "x";

} else if(str.charAt(i)== ('b')){

plain\_text+= "y";

} else if(str.charAt(i)== ('c')){

plain\_text+= "z";

} else {

char ch= str.charAt(i);

plain\_text+= (char) (ch- 3);

}

}

return(plain\_text);

}

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

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

caesar\_cipher cc= new caesar\_cipher();

System.out.println("\*\*\*Caesar Encryption\*\*\*");

System.out.print("Enter Your Text: ");

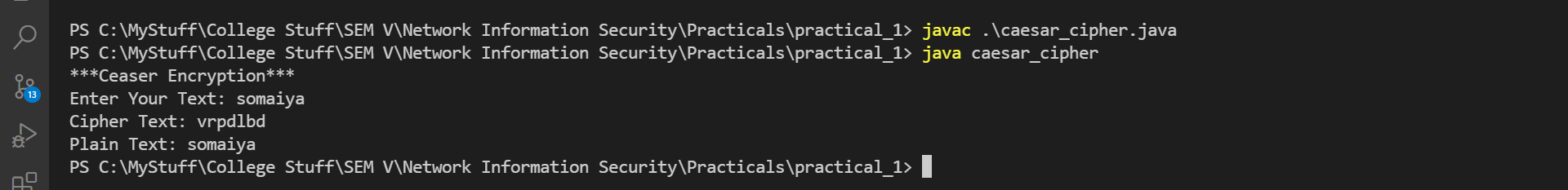
String text= br.readLine();

System.out.println("Cipher Text: "+ cc.encrypt(text));

System.out.println("Plain Text: "+ cc.decrypt(cc.encrypt(text)));

}

}



Practical 2

Q1) Demonstrate Rail Fence Cipher.

Ans:

railfence\_cipher.java

/\*

railfence\_cipher.java

Author: Jagrut Gala

Date: 08-07-2021

Practical: 2

Objective: Demonstrate Rail Fence Cipher.

\*/

import java.io.\*;

public class railfence\_cipher {

railfence\_cipher() {}

String encrypt(String text) {

String str1= "";

String str2= "";

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

if(i% 2== 0) {

str1+= text.charAt(i);

} else {

str2+= text.charAt(i);

}

}

text= str1+ str2;

return text;

}

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

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

railfence\_cipher rc= new railfence\_cipher();

System.out.println("\*\*\*Railfence Encryption\*\*\*");

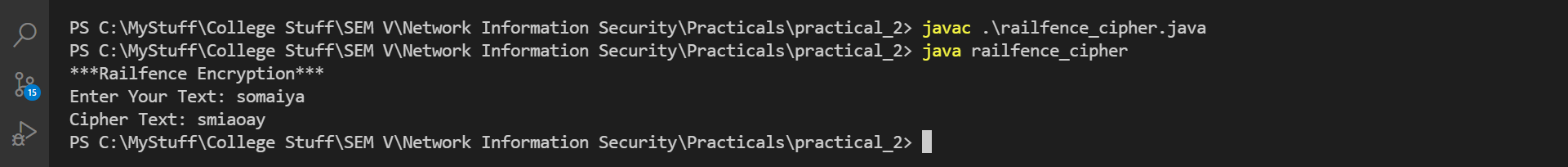
System.out.print("Enter Your Text: ");

String text= br.readLine();

System.out.println("Cipher Text: "+ rc.encrypt(text));

}

}



Practical 3

Q1) Demonstrate Mono Alphabetic Cipher.

Ans:

monoalphabetic\_cipher.java

/\*

monoalphabetic\_cipher.java

Author: Jagrut Gala

Date: 15-07-2021

Practical: 3

Objective: Demonstrate Mono Alphabetic Cipher.

\*/

import java.io.\*;

public class monoalphabetic\_cipher {

char[] plain\_char = {'a', 'b', 'e', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'};

char[] cipher\_char = { 'Q', 'w', 'E', 'R', 'T', 'Y', 'U', 'I', 'O', 'P', 'A', 'S', 'D', 'F', 'G', 'H', 'J', 'K', 'L', 'Z', 'X', 'C', 'V', 'B', 'N', 'M'};

monoalphabetic\_cipher() {}

String encrypt(String text) {

String cipher\_text= "";

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

for (int j = 0; j < plain\_char.length; j++) {

if(text.charAt(i)== plain\_char[j]){

cipher\_text+= cipher\_char[j];

}

}

}

return(cipher\_text);

}

String decrypt(String text) {

String plain\_text= "";

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

for (int j = 0; j < cipher\_char.length; j++) {

if(text.charAt(i)== cipher\_char[j]){

plain\_text+= plain\_char[j];

}

}

}

return(plain\_text);

}

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

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

monoalphabetic\_cipher mc= new monoalphabetic\_cipher();

System.out.println("\*\*\*Monoalphabetic Encryption\*\*\*");

System.out.print("Enter Your Text: ");

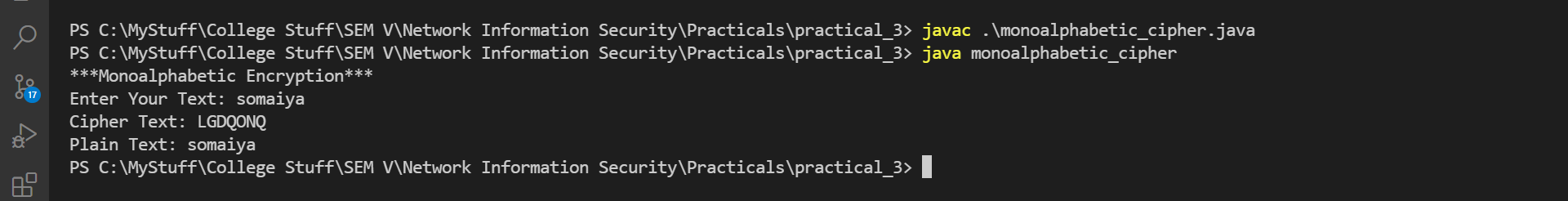
String text= br.readLine();

System.out.println("Cipher Text: "+ mc.encrypt(text));

System.out.println("Plain Text: "+ mc.decrypt(mc.encrypt(text)));

}

}



Practical 4

Q1) Demonstrate Vernam Cipher.

Ans:

vernam\_cipher.java

/\*

vernam\_cipher.java

Author: Jagrut Gala

Date: 22-07-2021

Practical: 4

Objective: Demonstrate Vernam Cipher.

\*/

import java.io.\*;

public class vernam\_cipher {

char[] alpha\_arr= new char[26];

vernam\_cipher() {

for(int i=0; i<this.alpha\_arr.length; i++){

this.alpha\_arr[i]= (char) ('A'+ i);

System.out.println(i+ ", " + this.alpha\_arr[i]);

}

}

char[] getKeyArray(String key, int len) {

char[] key\_arr= new char[len];

for(int i=0; i< key\_arr.length; i++) {

System.out.println(i% key.length());

key\_arr[i]= key.charAt(i% key.length());

}

return key\_arr;

}

String encrypt(String text, String key) {

char[] text\_arr= text.toCharArray();

char[] key\_arr= this.getKeyArray(key, text\_arr.length);

int[] num\_arr= new int[text\_arr.length];

for(int i=0; i< num\_arr.length; i++) {

num\_arr[i]= 0;

}

for(int i=0; i< text\_arr.length; i++) {

for(int j=0; j< this.alpha\_arr.length; j++) {

if(text\_arr[i] == alpha\_arr[j]) {

num\_arr[i]+= j;

// num\_arr[i]= (text\_arr[i] + key\_arr[i% key\_arr.length])% alpha\_arr.length;

}

if(key\_arr[i] == alpha\_arr[j]){

num\_arr[i]+= j;

}

}

num\_arr[i]%= this.alpha\_arr.length;

}

for(int i=0; i< num\_arr.length; i++) {

text\_arr[i]= this.alpha\_arr[num\_arr[i]];

}

text= new String(text\_arr);

return text;

}

String decrypt(String text, String key) { // no

return text;

}

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

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

vernam\_cipher vc= new vernam\_cipher();

System.out.println("\*\*\*Vernam Encryption\*\*\*");

System.out.print("Enter Your Text: ");

String text= br.readLine();

System.out.print("Enter Your Key: ");

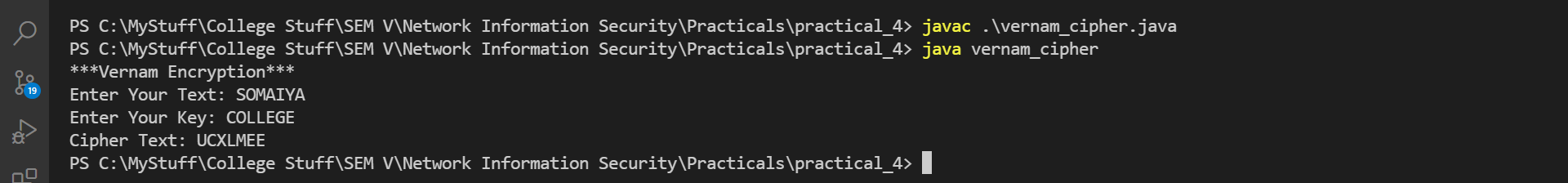
String key= rb.readLine();

System.out.println("Cipher Text: "+ vc.encrypt(text));

System.out.println("Plain Text: "+ vc.decrypt(vc.encrypt(text)));

}

}



Practical 5

Q1) Demonstrate Columnar Cipher.

Ans:

columnar\_cipher.java

/\*

columnar\_cipher.java

Author: Jagrut Gala

Date: 29-07-2021

Practical: 5

Objective: Demonstrate Columnar Cipher.

\*/

import java.io.\*;

public class columnar\_cipher {

columnar\_cipher() {}

String encrypt(String text, int row, int col, int[] choice) {

int count= 0;

char[][] cipher\_matrix= new char[row][col];

for(int i= 0; i< row; i++) { // columnar creation

for(int j= 0; j< col; j++) {

if(count>= text.length()) {

cipher\_matrix[i][j]= ' ';

} else {

cipher\_matrix[i][j]= text.charAt(count);

}

count++;

System.out.print(cipher\_matrix[i][j]);

}

}

System.out.println("");

String cipher\_text= "";

for(int i= 0; i< col; i++) { // columnar encrption

int k= choice[i];

for(int j= 0; j< row; j++) {

cipher\_text+= cipher\_matrix[j][k];

}

}

return cipher\_text;

}

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

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

columnar\_cipher cc= new columnar\_cipher();

System.out.print("Enter a String: ");

String text= br.readLine();

System.out.print("Enter Number of Rows: ");

int row\_num = Integer.parseInt(br.readLine());

System.out.print("Enter Number of Columns: ");

int col\_num= Integer.parseInt(br.readLine());

if(row\_num\* col\_num < text.length()) {

throw new Exception("Insufficent Area for Text");

}

System.out.print("Enter your order of columns: ");

String[] order= br.readLine().trim().split(" ");

if(order.length != col\_num) {

throw new Exception("Invalid order of Colmuns given");

}

int[] choice= new int[col\_num];

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

choice[i]= Integer.parseInt(order[i])- 1;

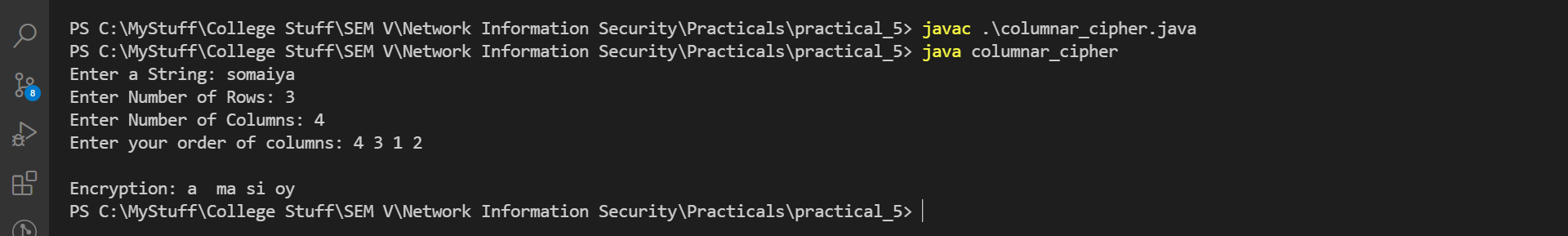
}

String cipher\_text=cc.encrypt(text, row\_num, col\_num, choice);

System.out.println("Encryption: "+ cipher\_text);

}

}



Practical 6

Q1) Demonstrate diffie\_hellman\_exchange

Ans:

diffie\_hellman\_exchange.java

/\*

diffie\_hellman\_exchange.java

Author: Jagrut Gala

Date: 12-08-2021

Practical: 6

Objective: Demonstrate diffie\_hellman\_exchange

Input:

\*/

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.math.BigInteger;

public class diffie\_hellman\_exchange { // just key generation

static BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

BigInteger prime1;

BigInteger prime2;

static BigInteger getBigIntegerNum(String msg) {

BigInteger num= new BigInteger("0");

try {

System.out.print(msg+ ": ");

num= new BigInteger(br.readLine());

} catch(Exception err) {

System.out.println(err);

}

return num;

}

static boolean isPrime(BigInteger num){

if(num.isProbablePrime(10)) {

return true;

} else {

return false;

}

}

public static void main(String[] args){

BigInteger a, b, x, y, p, g, xa, yb;

p= getBigIntegerNum("Enter a Prime Number");

while(!isPrime(p)) {

System.out.println("Not Prime");

p= getBigIntegerNum("Enter a Prime Number");

}

g= getBigIntegerNum("Enter a Primitive root of "+ p.toString(10));

a= getBigIntegerNum("Enter a Number(key for a)");

x= g.modPow(a, p);

b= getBigIntegerNum("Enter a Number(key for b)");

y= g.modPow(b, p);

xa= y.modPow(a, p);

System.out.println("xa: "+ xa);

yb= x.modPow(b, p);

System.out.println("yb: "+ yb);

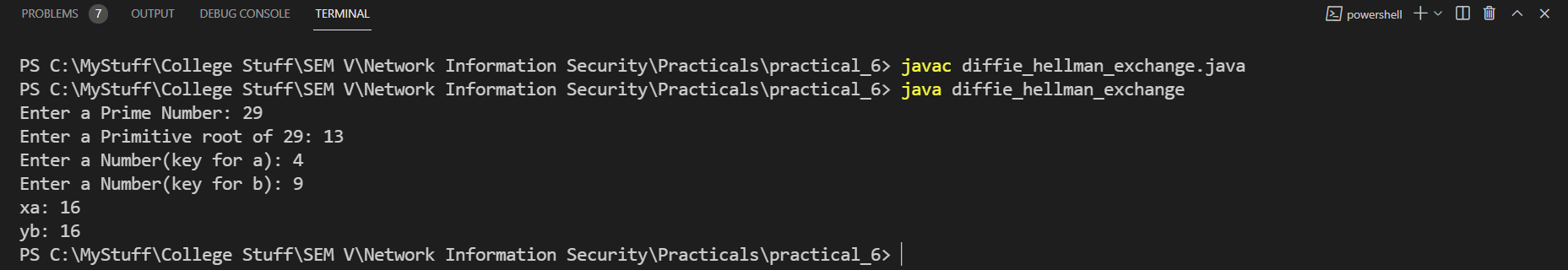
if(xa == yb) {

System.out.println("Keys are Symmetric: "+ xa);

}

}

}



Practical 7

Q1) Demonstrate RSA

Ans:

rsa.java

/\*

rsa.java

Author: Jagrut Gala

Date: 26-08-2021

Practical: 7

Objective: Demonstarte RSA

\*/

import java.security.\*;

import java.math.\*;

public class rsa

{

public static void main(String[] args)

{

SecureRandom r;

BigInteger p,q,p1,q1,n,n1,e,d,msg,ct,pt;

int bitLength = 512;

int certinity = 100;

r = new SecureRandom();

//Step1: Generate prime number p & q

p = new BigInteger(bitLength,certinity,r);

q = new BigInteger(bitLength,certinity,r);

//Step2: n = p \* q

n = p.multiply(q);

System.out.println("Prime Number P is: " + p.intValue());

System.out.println("Prime Number Q is: " + q.intValue());

System.out.println("n = p \* q is: " + n.intValue());

//Step3: Generating Punlic Key (E)

p1 = p.subtract(new BigInteger("1"));

q1 = q.subtract(new BigInteger("1"));

n1 = p1.multiply(q1);

e = new BigInteger("2");

while (n1.gcd(e).intValue() > 1 || e.compareTo(p1) != -1)

{

e = e.add(new BigInteger("1"));

}

System.out.println("Public Key is (" + n.intValue() + "," + e.intValue() + ")");

//Step4: D = E ^ -1 mod(P-1)(Q-1)

d = e.modInverse(n1);

System.out.println("Private Key is (" + n.intValue() + "," + d.intValue() + ")");

//Step5: Encryption CT = (PT) ^ e mod n

msg = new BigInteger("42");

ct = msg.modPow(e, n);

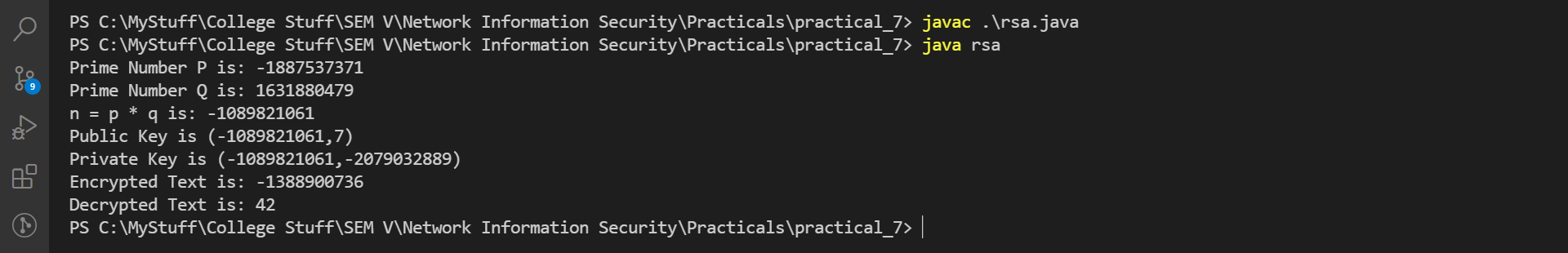
System.out.println("Encrypted Text is: " + ct.intValue());

pt = ct.modPow(d, n);

System.out.println("Decrypted Text is: " + pt.intValue());

}

}



Practical 8

Q1) Demonstrate DES.

Ans:

des.java

/\*

des.java

Author: Jagrut Gala

Date: 26-08-2021

Practical: 8

Objective: Demonstrate DES Encryption and Decryption.

\*/

import java.io.\*;

import java.util.Base64;

import javax.crypto.\*;

public class des {

Cipher encipher, decipher;

des(SecretKey key) {

try {

encipher= Cipher.getInstance("DES");

encipher.init(Cipher.ENCRYPT\_MODE, key);

decipher= Cipher.getInstance("DES");

decipher.init(Cipher.DECRYPT\_MODE, key);

} catch (Exception e) {

System.out.println(e);

}

}

String encrypt1(String plain\_text) {

String encrpypted\_text= "";

try {

byte[] utf8\_text= plain\_text.getBytes("UTF8");

byte[] enc= encipher.doFinal(utf8\_text);

encrpypted\_text= new String(Base64.getEncoder().encode(enc));

} catch (Exception e) {

System.out.println(e);

}

return encrpypted\_text;

}

String decrypt1(String cipher\_text) {

String decrpypted\_text= "";

try {

byte[] dec= Base64.getDecoder().decode(cipher\_text);

byte[] utf8\_text= decipher.doFinal(dec);

decrpypted\_text= new String(utf8\_text);

} catch (Exception e) {

System.out.println(e);

}

return decrpypted\_text;

}

public static void main(String[] args) {

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

System.out.println("\*\*\*DES Encryption\*\*\*");

System.out.print("Enter Your Text: ");

String text= br.readLine();

System.out.println("Entered String: "+ text);

try {

SecretKey key= KeyGenerator.getInstance("DES").generateKey();

des des\_var= new des(key);

String encrypted\_text= des\_var.encrypt1(text);

String decrypted\_text= des\_var.decrypt1(encrypted\_text);

System.out.println("Encrypted Text: "+ encrypted\_text);

System.out.println("Decrypted Text: "+ decrypted\_text);

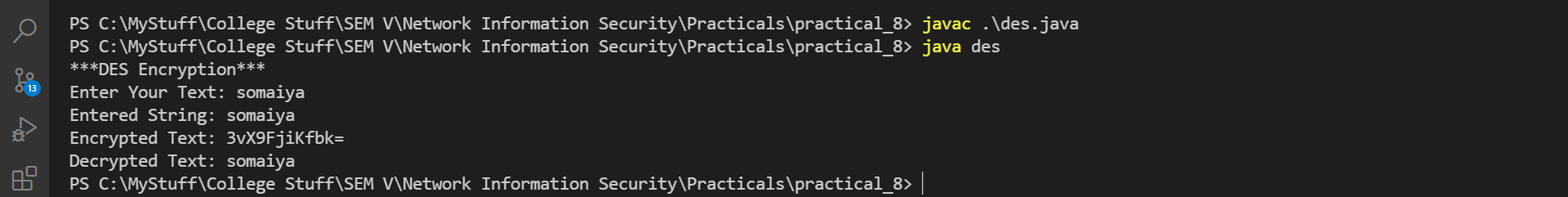
} catch(Exception e) {

System.out.println(e);

}

}

}



Practical 9

Q1) Demonstrate AES.

Ans:

aes.java

/\*

aes.java

Author: Jagrut Gala

Date: 26-08-2021

Practical: 9

Objective: Demonstrate AES Encryption and Decryption.

\*/

import java.io.\*;

import java.util.Base64;

import javax.crypto.\*;

public class aes {

Cipher encipher, decipher;

aes(SecretKey key) {

try {

encipher= Cipher.getInstance("AES");

encipher.init(Cipher.ENCRYPT\_MODE, key);

decipher= Cipher.getInstance("AES");

decipher.init(Cipher.DECRYPT\_MODE, key);

} catch (Exception e) {

System.out.println(e);

}

}

String encrypt1(String plain\_text) {

String encrpypted\_text= "";

try {

byte[] utf8\_text= plain\_text.getBytes("UTF8");

byte[] enc= encipher.doFinal(utf8\_text);

encrpypted\_text= new String(Base64.getEncoder().encode(enc));

} catch (Exception e) {

System.out.println(e);

}

return encrpypted\_text;

}

String decrypt1(String cipher\_text) {

String decrpypted\_text= "";

try {

byte[] dec= Base64.getDecoder().decode(cipher\_text);

byte[] utf8\_text= decipher.doFinal(dec);

decrpypted\_text= new String(utf8\_text);

} catch (Exception e) {

System.out.println(e);

}

return decrpypted\_text;

}

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

BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

System.out.println("\*\*\*AES Encryption\*\*\*");

System.out.print("Enter Your Text: ");

String text= br.readLine();

System.out.println("Entered String: "+ text);

try {

SecretKey key= KeyGenerator.getInstance("AES").generateKey();

aes aes\_var= new aes(key);

String encrypted\_text= aes\_var.encrypt1(text);

String decrypted\_text= aes\_var.decrypt1(encrypted\_text);

System.out.println("Encrypted Text: "+ encrypted\_text);

System.out.println("Decrypted Text: "+ decrypted\_text);

} catch(Exception e) {

System.out.println(e);

}

}

}

