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)));
    }
}
```

```
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_1> javac .\caesar_cipher.java
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_1> java caesar_cipher
***Ceaser Encryption***
Enter Your Text: somaiya
Cipher Text: vrpdlbd
Plain Text: somaiya
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_1>
```

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));
    }
}
```

```
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_2> javac .\railfence_cipher.java
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_2> java railfence_cipher
***Railfence Encryption***
Enter Your Text: somaiya
Cipher Text: smiaoay
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_2>
```

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)));
    }
}
```

```
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_3> javac .\monoalphabetic_cipher.java
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_3> java monoalphabetic_cipher
***Monoalphabetic Encryption***
Enter Your Text: somaiya
Cipher Text: LGDQONQ
Plain Text: somaiya
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_3>
```

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)));
    }
}
```

```
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_4> javac .\vernam_cipher.java
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_4> java vernam_cipher
***Vernam Encryption***
Enter Your Text: SOMAIYA
Enter Your Key: COLLEGE
Cipher Text: UCXLMEE
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_4>
```

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);
    }
}
```

```
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_5> javac .\columnar_cipher.java
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_5> java columnar_cipher
Enter a String: somaiya
Enter Number of Rows: 3
Enter Number of Columns: 4
Enter your order of columns: 4 3 1 2

Encryption: a  ma si oy
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_5>
```

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);
        }
    }
}
```

```
PROBLEMS 7   OUTPUT   DEBUG CONSOLE   TERMINAL                                   powershell

PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_6> javac diffie_hellman_exchange.java
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_6> java diffie_hellman_exchange
Enter a Prime Number: 29
Enter a Primitive root of 29: 13
Enter a Number(key for a): 4
Enter a Number(key for b): 9
xa: 16
yb: 16
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_6>
```

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());
    }
}
```

```
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_7> javac .\rsa.java
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_7> java rsa
Prime Number P is: -1887537371
Prime Number Q is: 1631880479
n = p * q is: -1089821061
Public Key is (-1089821061,7)
Private Key is (-1089821061,-2079032889)
Encrypted Text is: -1388900736
Decrypted Text is: 42
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_7>
```

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);
        }
    }
}
```

```
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_8> javac .\des.java
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_8> java des
***DES Encryption***
Enter Your Text: somaiya
Entered String: somaiya
Encrypted Text: 3vX9FjiKfbk=
Decrypted Text: somaiya
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_8>
```

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);
        }
    }
}
```

```
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_9> javac .\aes.java
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_9> java aes
***AES Encryption***
Enter Your Text: college
Entered String: college
Encrypted Text: v/U9f1jeLYK9gNOgu1bxoQ==
Decrypted Text: college
PS C:\MyStuff\College Stuff\SEM V\Network Information Security\Practicals\practical_9>
```