

EX - 1 : CaesarCipher.java

```
import java.util.Scanner;

class CaesarCipher {
    static char[] enc(String msg,int key){
        char[] crypt=new char[msg.length()];
        for(int i=0;i<msg.length();i++)
        {
            if (msg.charAt(i)==' '){
                crypt[i]=' ';
                continue;
            }
            if(msg.charAt(i)+key>122)
                crypt[i]=(char)(96+key%26);
            else
                crypt[i]=(char)(msg.charAt(i)+key);
        }
        return crypt;
    }
    public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);
        String words[] = {"hey","hello","hi"};
        System.out.println("Enter the message (lower Case, without spaces): ");
        String msg=scan.nextLine();
        char[] crypt=new char[msg.length()];

        System.out.println("Enter the key value (displacement): ");
        int key=scan.nextInt();
        System.out.println("ENCRYPTED :");
        char[] encrypted = enc(msg,key);
        System.out.println(encrypted);

        System.out.println("\nDECRYPTED : ");
        char[] decrypted = enc(new String(encrypted),-1* key);
        System.out.println(decrypted);

        Scanner scan2=new Scanner(System.in);
        System.out.println("Enter the encrypted string");
        String msg2=scan2.nextLine();
        String message_parts[] = msg2.split(" ");
        int final_key = 0;
        boolean flag = false;
        for (int i=0;i<message_parts.length;i++){
            for (int j=1;j<27;j++){
                String dec_word = new String(enc(message_parts[i],-1*j));
                for (int k=0;k<words.length;k++){
                    if (dec_word.matches(words[k])){
                        System.out.println("Key Matched :"+j);
                        final_key = j;
                        flag = true;
                    }
                }
                if (flag)
                    break;
            }
            if (flag)
                break;
        }
        if (flag){
            String decrypted_message = new String(enc(msg2,-1*final_key));
            System.out.println("DECRYPTED :"+decrypted_message);
        }
    }
}
```

SAMPLE I/O:

Enter the message (lower Case, without spaces):

hello world

Enter the key value (displacement):

3

ENCRYPTED :

khoor zruog

DECRYPTED :

hello world

Enter the encrypted string

khoor zruog

Key Matched : 3

DECRYPTED :hello world

EX - 1 : PlayFair.java

```
import java.io.*;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
class PlayFair{
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        int k=0, keylen = 0, i = 0, j = 0;
        char ch;
        String key;
        char[][] mat=new char[5][5];
        System.out.println("Enter key: ");
        key = s.nextLine();
        keylen = key.length();
        Map<Character, Integer> map = new HashMap<>();
        i = j = k = 0;
        for(k = 0; k < keylen; k++){
            ch = key.charAt(k);
            if(!map.containsKey(ch)){
                map.put(ch, 1);
                mat[i][j++]=ch;
                if(j==5)
                {
                    j=0;
                    i++;
                }
            }
        }
        int newi = i, newj = j;
        ch = 'A';
        for(ch = 'A'; ch <= 'Z'; ch++){
            if(!map.containsKey(ch)){
                if(ch == 'I' || ch == 'J')
                    if(map.containsKey('I') || map.containsKey('J'))
                        continue;
                map.put(ch, 1);

                if(newj == 5){
                    if(newi == 4)
                        break;
                    newi ++;
                    newj = 0;
                }
                mat[newi][newj++]=ch;
            }
        }
        for(i=0;i<5;i++){
            for(j=0;j<5;j++){
                System.out.print(mat[i][j] + " ");
            }
            System.out.println();
        }

        System.out.println("Enter message to encrypt: ");
        String message = s.nextLine(), cipherText = "";
        int msgLen = message.length(), row1, col1, row2, col2, row, col;
        char ch1, ch2;
        boolean flag1, flag2;
        for(i = 0; i < msgLen; i++){
            ch1 = message.charAt(i++);
            if(i < msgLen)
                ch2 = message.charAt(i);
            else
                ch2 = 'X';
            if(ch1 == ch2 || (ch1 == 'I' && ch2 == 'J') || (ch1 == 'J' && ch2 ==
'I')){
                ch2 = 'X';
                i--;
            }
        }
    }
}
```

```

        flag1 = flag2 = false;
        row1 = col1 = row2 = col2 = -1;
        for(row = 0; row < 5; row++){
            for(col = 0; col < 5; col++){
                if(flag1 && flag2)
                    break;
                if(mat[row][col] == ch1 || (ch1 == 'I' && mat[row][col]
== 'J')) || (ch1 == 'J' && mat[row][col] == 'I')){
                    row1 = row;
                    col1 = col;
                    flag1 = true;
                }
                else if(mat[row][col] == ch2 || (ch2 == 'I' && mat[row]
[col] == 'J')) || (ch2 == 'J' && mat[row][col] == 'I')){
                    row2 = row;
                    col2 = col;
                    flag2 = true;
                }
            }
            if(flag1 && flag2)
                break;
        }
        if(row1 == row2){
            cipherText += (char)mat[row1][(col1+1)%5] + "" + (char)mat[row2]
[(col2+1)%5];
        }
        else if(col1 == col2){
            cipherText += (char)mat[(row1+1)%5][col1] + "" +
(char)mat[(row2+1)%5][col2];
        }
        else{
            cipherText += (char)mat[row1][col2] + "" + (char)mat[row2]
[col1];
        }
    }
    System.out.println("cipherText = "+cipherText);

    int cipherLen = cipherText.length();
    String decipheredText = "";
    for(i = 0; i < cipherLen; i++){
        ch1 = cipherText.charAt(i++);
        ch2 = cipherText.charAt(i);
        flag1 = flag2 = false;
        row1 = col1 = row2 = col2 = -1;
        for(row = 0; row < 5; row++){
            for(col = 0; col < 5; col++){
                if(flag1 && flag2)
                    break;
                if(mat[row][col] == ch1 || (ch1 == 'I' && mat[row][col]
== 'J')) || (ch1 == 'J' && mat[row][col] == 'I')){
                    row1 = row;
                    col1 = col;
                    flag1 = true;
                }
                else if(mat[row][col] == ch2 || (ch2 == 'I' && mat[row]
[col] == 'J')) || (ch2 == 'J' && mat[row][col] == 'I')){
                    row2 = row;
                    col2 = col;
                    flag2 = true;
                }
            }
            if(flag1 && flag2)
                break;
        }
        if(row1 == row2){
            if(col1 - 1 < 0)
                col1 = 5;
            if(col2 - 1 < 0)
                col2 = 5;

```

```

        decipheredText += (char)mat[row1][(col1-1)] + ""+
(char)mat[row2][(col2-1)];
    }
    else if(col1 == col2){
        if(row1 - 1 < 0)
            row1 = 5;
        if(row2 - 1 < 0)
            row2 = 5;
        decipheredText += (char)mat[(row1-1)][col1] + ""+
(char)mat[(row2-1)][col2];
    }
    else{
        decipheredText += (char)mat[row1][col2] + ""+(char)mat[row2]
[col1];
    }
}
System.out.println("decipheredText = "+decipheredText);
}
}

```

SAMPLE I/O:

Enter key:

HELLOWORLD

H E L O W

R D A B C

F G I K M

N P Q S T

U V X Y Z

Enter message to encrypt:

MARYHADALITTLELAMB

cipherText = ICBULRABAQQZQWLOCIAY

decipheredText = MARYHADALITXTLELAMBX

EX2 : HillCipher.java

```
import java.io.*;
import java.util.*;
import java.lang.Math.*;

public class HillCipher {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.println("\nHILL CIPHER\n");
        System.out.println("1. Encryption\n2. Decryption\n3. Exit");
        int choice = -1;
        Methods method= new Methods();

        while(choice != 3) {
            System.out.print("\nEnter an option : ");
            choice = scanner.nextInt();
            switch(choice) {
                case 1:
                    method.Encrypt();
                    break;
                case 2:
                    method.Decrypt();
                    break;
                case 3:
                    break;
            }
        }
    }

    class Methods {

        int GCD(int m,int n){
            if(m==0)
                return n;
            return GCD(n%m,m);
        }

        boolean Invertible(int[][] A) {
            int det = 0;
            for(int i=0;i<3;i++) {
                int a = 1;
                int b = (i+1)%3;
                int partial = (A[a][b] * A[(a+1)%3][(b+1)%3]);
                partial -= (A[a][(b+1)%3] * A[(a+1)%3][b]);
                partial *= A[0][i];
                det += partial;
            }

            if(det == 0){
                System.out.println("The given key matrix is not Invertible");
                return false;
            }

            // Have to find d^-1
            // d^-1 does not exist if gcd(d,26)<> 1
            // In that case find a different key
            if(GCD(det,26) != 1){
                System.out.println("The inverse key does not exist for the given key
matrix");
                return false;
            }
            System.out.println("The given key matrix is Invertible");
            return true;
        }

        int[][] Inverse(int[][] A) {
```

```

int det = 0;
for(int i=0;i<3;i++) {
    int a = 1;
    int b = (i+1)%3;
    int partial = (A[a][b] * A[(a+1)%3][(b+1)%3]);
    partial -= (A[a][(b+1)%3] * A[(a+1)%3][b]);
    partial *= A[0][i];
    det += partial;
}

//Find determinant modulo 26
while(det<0 || det>25){
    if(det<0)
        det+=26;
    else det%=26;
}

//Find inverse determinant
int inverseDet=0;
for(int i=1;i<=25;i++) {
    if((det*i)%26 == 1){
        inverseDet = i;
        break;
    }
}

//transpose
for(int i=0;i<3;i++){
    for(int j=0;j<i;j++){
        int temp = A[i][j];
        A[i][j] = A[j][i];
        A[j][i] = temp;
    }
}

int[][] inverseMatrix = new int[3][3];

for(int i=0;i<3;i++) {
    int minorDet = 0;
    for(int j=0;j<3;j++) {
        int a = (i+1)%3;
        int b = (j+1)%3;
        minorDet = (A[a][b] * A[(a+1)%3][(b+1)%3]);
        minorDet -= (A[a][(b+1)%3] * A[(a+1)%3][b]);
        minorDet*=inverseDet; // d^-1 * Adj(A)
        inverseMatrix[i][j] = minorDet;
    }
}
return inverseMatrix;
}

int[] MatrixMultiply(int A[],int B[][]){
    int sum[] = new int[3];
    for(int i=0;i<3;i++) {
        sum[i] = 0;
        for(int j=0;j<3;j++){
            sum[i] += (A[j]*B[j][i]);
            sum[i] = sum[i]%26;
        }
    }
    return sum;
}

public void Encrypt() {
    Scanner scanner = new Scanner(System.in);
    System.out.println("\nENCRYPTION");
    System.out.print("\nEnter the plain text : ");
    String plainText = scanner.next();
    System.out.println("Enter the key matrix : ");
    int key[][] = new int[3][3];
    for(int i=0;i<3;i++)

```



```

        for(int j=0;j<3;j++)
            key[i][j] = scanner.nextInt();

    if(!Invertible(key))
        return;

    int len = plainText.length();
    String cipherText = "";
    for(int i=0;i<len;) {
        int[] pair = new int[3];
        for(int j=0;j<3;j++) {
            if(i<len)
                pair[j] = plainText.charAt(i++) - 65;
            else pair[j] = 0;
        }

        pair = MatrixMultiply(pair,key);
        for(int j=0;j<3;j++) {
            cipherText +=(char)(pair[j] + 65);
        }
    }

    System.out.println("\nThe cipher text is : " + cipherText);
}

public void Decrypt() {

    Scanner scanner = new Scanner(System.in);
    System.out.println("\nDECRYPTION");
    System.out.print("\nEnter the cipher text : ");
    String cipherText = scanner.next();
    System.out.println("Enter the key matrix : ");
    int key[][] = new int[3][3];
    for(int i=0;i<3;i++)
        for(int j=0;j<3;j++)
            key[i][j] = scanner.nextInt();

    if(!Invertible(key))
        return;

    int[][] inverseKey = new int[3][3];

    inverseKey = Inverse(key);

    int len = cipherText.length();
    String plainText = "";
    for(int i=0;i<len;) {
        int[] pair = new int[3];
        for(int j=0;j<3;j++) {
            if(i<len)
                pair[j] = cipherText.charAt(i++) - 65;
            else pair[j] = 0;
        }

        pair = MatrixMultiply(pair,inverseKey);
        for(int j=0;j<3;j++) {
            if(pair[j] >=0)
                plainText +=(char)(pair[j] + 65);
            else plainText += (char)(65 + pair[j] + 26);
        }
    }

    System.out.println("\nThe plain text is : " + plainText);
}
}

```

SAMPLE I/O:

HILL CIPHER

1. Encryption
2. Decryption
3. Exit

Enter an option : 1

ENCRYPTION

Enter the plain text : HELLOWORLD

Enter the key matrix :

1 2 3

1 2 3

1 2 3

The given key matrix is not Invertible

Enter an option : 1

ENCRYPTION

Enter the plain text : HELLOWORLD

Enter the key matrix :

6 24 1

13 16 10

20 17 15

The given key matrix is Invertible

The cipher text is : CDEMENFPLSUD

Enter an option : 2

DECRYPTION

Enter the cipher text : CDEMENFPLSUD

Enter the key matrix :

6 24 1

13 16 10

20 17 15

The given key matrix is Invertible

The plain text is : HELLOWORLDAA

Enter an option : 3

EX2 - Vigenere.java

```
import java.io.*;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
class Vigenere{
    public static char findCharacter(char array[], char ch){
        int i = 0;
        for(i = 0; i < 26; i ++){
            if(array[i] == ch)
                return (char)(65 + i);
        }
        return 0;
    }
    public static void main(String[] args){
        String input, key;
        Scanner sc = new Scanner(System.in);
        int inputLen = 0, i = 0, j = 0, keyLen = 0, k = 0;
        System.out.println("Enter input message: ");
        input = sc.nextLine();
        inputLen = input.length();

        System.out.println("Enter key: ");
        key = sc.nextLine();
        keyLen = key.length();

        k = keyLen; i = 0;
        while(k < inputLen){
            key += key.charAt(i%keyLen);
            i++;
            k++;
        }
        System.out.println("Key repeated to form: "+key);
        char VigenereMatrix[][] = new char[26][26];
        for(i = 0; i < 26; i ++){
            for(j = 0; j < 26; j ++){
                VigenereMatrix[i][j] = (char)(65 + ((i+j)%26));
                System.out.print(VigenereMatrix[i][j] + " ");
            }
            System.out.println();
        }

        String encryptedMessage = "";
        for(i = 0; i < inputLen; i ++){
            encryptedMessage += VigenereMatrix[input.charAt(i) - 'A']
[key.charAt(i) - 'A'];
        }
        System.out.println("Encrypted Message = "+encryptedMessage);

        String decryptedMessage = "";
        for(i = 0; i < inputLen; i ++){
            decryptedMessage += findCharacter(VigenereMatrix[key.charAt(i) -
'A'], encryptedMessage.charAt(i));
        }
        System.out.println("Decrypted Message = "+decryptedMessage);

    }
}
```

SAMPLE I/0

Enter input message:

HELLOWORLD

Enter key:

APPLE

Key repeated to form: APPLEAPPLE

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
B C D E F G H I J K L M N O P Q R S T U V W X Y Z A
C D E F G H I J K L M N O P Q R S T U V W X Y Z A B
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
E F G H I J K L M N O P Q R S T U V W X Y Z A B C D
F G H I J K L M N O P Q R S T U V W X Y Z A B C D E
G H I J K L M N O P Q R S T U V W X Y Z A B C D E F
H I J K L M N O P Q R S T U V W X Y Z A B C D E F G
I J K L M N O P Q R S T U V W X Y Z A B C D E F G H
J K L M N O P Q R S T U V W X Y Z A B C D E F G H I
K L M N O P Q R S T U V W X Y Z A B C D E F G H I J
L M N O P Q R S T U V W X Y Z A B C D E F G H I J K
M N O P Q R S T U V W X Y Z A B C D E F G H I J K L
N O P Q R S T U V W X Y Z A B C D E F G H I J K L M
O P Q R S T U V W X Y Z A B C D E F G H I J K L M N
P Q R S T U V W X Y Z A B C D E F G H I J K L M N O
Q R S T U V W X Y Z A B C D E F G H I J K L M N O P
R S T U V W X Y Z A B C D E F G H I J K L M N O P Q
S T U V W X Y Z A B C D E F G H I J K L M N O P Q R
T U V W X Y Z A B C D E F G H I J K L M N O P Q R S
U V W X Y Z A B C D E F G H I J K L M N O P Q R S T
V W X Y Z A B C D E F G H I J K L M N O P Q R S T U
W X Y Z A B C D E F G H I J K L M N O P Q R S T U V
X Y Z A B C D E F G H I J K L M N O P Q R S T U V W
Y Z A B C D E F G H I J K L M N O P Q R S T U V W X
Z A B C D E F G H I J K L M N O P Q R S T U V W X Y

Encrypted Message = HTAWSWDGWH

Decrypted Message = HELLOWORLD

EX3 - RailFence.java

```
import static java.lang.Math.abs;
import java.util.Scanner;

class RailFence {
    public static void main(String[] args) {
        int key = 0; int i = 0; int j = 0; int len = 0;

        Scanner reader = new Scanner(System.in);
        Scanner keyread = new Scanner(System.in);

        System.out.println("\nEnter the key: ");
        key = keyread.nextInt();

        System.out.println("\nEnter the plain text: ");
        String message = reader.nextLine();
        len = message.length();
        System.out.println(len);
        System.out.println(key);
        char matrix [][] = new char [key][len];
        for (i=0; i<len; i++){
            for (j=0; j<key; j++){
                matrix[j][i]='*';
            }
        }
        char[] a = message.toCharArray();
        int ind;
        int tkey = key - 1;
        for (i=0; i<len; i++){
            ind = tkey - Math.abs(tkey-i%(2*tkey));
            matrix[ind][i]=a[i];
        }
        for (i=0; i<key; i++){
            for (j=0; j<len; j++){
                System.out.print(matrix[i][j]);
                System.out.print("\n");
            }
        }
        System.out.println("ENCRYPTED :");
        for (i=0; i<key; i++){
            for (j=0; j<len; j++){
                System.out.print(matrix[i][j]);
            }
        }
        System.out.println("\n\nDECRYPTION :");
        System.out.println("\nEnter the key for decryption: ");
        int dec_key = keyread.nextInt();
        tkey = dec_key - 1;
        for (i=0; i<len; i++){
            ind = tkey - Math.abs(tkey-i%(2*tkey));
            System.out.print(matrix[ind][i]);
        }
        System.out.println();
    }
}
```

SAMPLE I/0

Enter the key:

4

Enter the plain text:

hello world

11

4

h*****w*****

*e*** *o***

l*o*r*d

l**l*

ENCRYPTED :

h*****w*****e*** *o*****l*o***r*d***l*****l*

DECRYPTION :

Enter the key for decryption:

4

hello world

EX3 - RowColCipher.java

```
import java.io.*;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
class RowColCipher{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int n = 0, i = 0, j = 0, k = 0, inputLen = 0;
        String inputMsg = "";
        char ch = ' ';
        System.out.println("Enter the message: ");
        inputMsg = sc.nextLine();
        System.out.println("Enter the number of columns: ");
        n = sc.nextInt();
        inputLen = inputMsg.length();

        /*****ENCRYPTION*****/
        char Matrix[][] = new char[100][n];
        i = j = k = 0;
        for(i = 0; k < inputLen; i++){
            for(j = 0; j < n; j++){
                while(k < inputLen && (ch = inputMsg.charAt(k)) == ' '){
                    k++;
                }
                if(k < inputLen){
                    Matrix[i][j] = ch;
                    System.out.print(Matrix[i][j] + " ");
                }
                else{
                    while(j < n){
                        Matrix[i][j++] = 'x';
                        System.out.print(Matrix[i][j-1] + " ");
                    }
                }
                k++;
            }
            System.out.println();
        }

        int m = i;

        Map<Integer, Integer> keyMap = new HashMap<>();
        String cipherText = "";
        int temp, key[] = new int[n];
        System.out.println("Enter key: ");
        for(i = 0; i < n; i++){
            temp = sc.nextInt();
            key[i] = temp-1;
            keyMap.put(temp-1, i);
        }
        for(i = 0; i < n; i++){
            for(j = 0; j < m; j++){
                cipherText += Matrix[j][keyMap.get(i)];
            }
        }
        System.out.println("Cipher Text = "+cipherText);

        /*****DECRYPTION*****/
        char decipherMatrix[][] = new char[n][m];
        k = 0;
        for(i = 0; i < n; i++){
            for(j = 0; j < m; j++){
                decipherMatrix[i][j] = cipherText.charAt(k++);
                System.out.print(decipherMatrix[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

```
String decipheredText = "";
int row = 0, col = 0;
for(col = 0; col < m; col++){
    for(row = 0; row < n; row++){
        decipheredText += decipherMatrix[key[row]][col];
    }
}

System.out.println("decipheredText = "+decipheredText);
}
```


SAMPLE I/O

```
Enter the message:
hello world
Enter the number of columns:
4
h e l l
o w o r
l d x x
Enter key:
2
1
3
4
Cipher Text = ewdholloxlrx
e w d
h o l
l o x
l r x
decipheredText = helloworldxx
```