**IT8761 – Security Laboratory**

**Reshma Ramesh Babu**

**312217104129**

**Aim:** To implement the substitution techniques: Caesar Cipher and Playfair Cipher.

**Caesar Cipher:**

**Code:**

public class caesar{

//encryption

public static String encrypt(String plaintext, int k)

{

StringBuilder result= new StringBuilder("");

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

{

if (Character.isUpperCase(plaintext.charAt(i)))

{

char ch = (char) (((int) plaintext.charAt(i) + k - 65) % 26

+ 65);

result.append(ch);

}

else

{

char ch = (char) (((int) plaintext.charAt(i) + k - 97) % 26

+ 97);

result.append(ch);

}

}

String op = result.toString();

return op;

}

//decryption

public static String decrypt(String cipher, int k)

{

StringBuilder result= new StringBuilder("");

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

{

if (Character.isUpperCase(cipher.charAt(i)))

{

char ch = (char) (((int) cipher.charAt(i) +(26-

k) - 65) % 26 + 65);

result.append(ch);

}

else

{

char ch = (char) (((int) cipher.charAt(i) + (26-

k) - 97) % 26 + 97);

result.append(ch);

}

}

String op = result.toString();

return op;

}

public static void main (String[] args)

{

//Read plaintext

String plaintext = null;

int shiftkey,c ;

System.out.println("Enter Plain Text:");

plaintext = System.console().readLine();

System.out.println("Enter shift key:");

shiftkey= Integer.parseInt(System.console().readLine());

do

{

System.out.println("MENU");

System.out.println("1. Encrypt");

System.out.println("2. Decrypt");

System.out.println("3.Exit");

System.out.println("Enter choice:");

c = Integer.parseInt(System.console().readLine());

String output=encrypt(plaintext,shiftkey);

if(c==1)

{

System.out.println("PlainText encypted:"+ output);

}

else if(c==2)

{

System.out.println("Cipher Decrypted:"+decrypt(output,shiftkey));

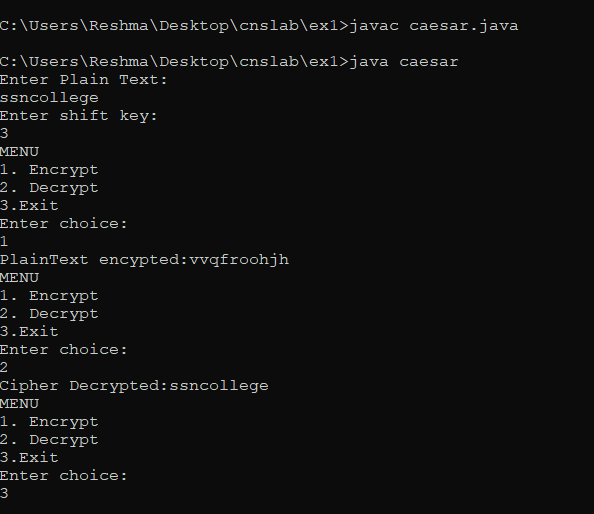
}

}while(c!=3);

}

}

**OUTPUT:**



**Playfair Cipher:**

**Code:**

import java.util.\*;

class AlphabetChecker{

String alphabets="ABCDEFGHIJKLMNOPQRSTUVWXYZ";

boolean checker(char c)

{

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

{

if(alphabets.charAt(i)==c)

return true;

}

return false;

}

}

class PlayFairAlgo{

AlphabetChecker b=new AlphabetChecker();

char keyMatrix[][]=new char[5][5];

boolean repeat(char c)

{

if(!b.checker(c))

{

return true;

}

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

{

for(int j=0;j < keyMatrix[i].length;j++)

{

if(keyMatrix[i][j]==c || c=='J')

return true;

}

}

return false;

}

void insertKey(String key)

{

key=key.toUpperCase();

key=key.replaceAll("J", "I");

key=key.replaceAll(" ", "");

int a=0,b=0;

for(int k=0;k < key.length();k++)

{

if(!repeat(key.charAt(k)))

{

keyMatrix[a][b++]=key.charAt(k);

if(b>4)

{

b=0;

a++;

}

}

}

char p='A';

while(a < 5)

{

while(b < 5)

{

if(!repeat(p))

{

keyMatrix[a][b++]=p;

}

p++;

}

b=0;

a++;

}

System.out.println("-Key Matrix-");

for(int i=0;i < 5;i++)

{

System.out.println();

for(int j=0;j < 5;j++)

{

System.out.print("\t"+keyMatrix[i][j]);

}

}

}

int rowPos(char c)

{

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

{

for(int j=0;j < keyMatrix[i].length;j++)

{

if(keyMatrix[i][j]==c)

return i;

}

}

return -1;

}

int columnPos(char c)

{

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

{

for(int j=0;j < keyMatrix[i].length;j++)

{

if(keyMatrix[i][j]==c)

return j;

}

}

return -1;

}

String encryptChar(String plain)

{

plain=plain.toUpperCase();

char a=plain.charAt(0),b=plain.charAt(1);

String cipherChar="";

int r1,c1,r2,c2;

r1=rowPos(a);

c1=columnPos(a);

r2=rowPos(b);

c2=columnPos(b);

if(c1==c2)

{

++r1;

++r2;

if(r1>4)

r1=0;

if(r2>4)

r2=0;

cipherChar+=keyMatrix[r1][c2];

cipherChar+=keyMatrix[r2][c1];

}

else if(r1==r2)

{

++c1;

++c2;

if(c1>4)

c1=0;

if(c2>4)

c2=0;

cipherChar+=keyMatrix[r1][c1];

cipherChar+=keyMatrix[r2][c2];

}

else{

cipherChar+=keyMatrix[r1][c2];

cipherChar+=keyMatrix[r2][c1];

}

return cipherChar;

}

String Encrypt(String plainText,String key)

{

insertKey(key);

String cipherText="";

plainText=plainText.replaceAll("j", "i");

plainText=plainText.replaceAll(" ", "");

plainText=plainText.toUpperCase();

int len=plainText.length();

for(int i=0;i < len-1;i=i+2)

{

cipherText+=encryptChar(plainText.substring(i,i+2));

}

return cipherText;

}

String decryptChar(String cipher)

{

cipher=cipher.toUpperCase();

char a=cipher.charAt(0),b=cipher.charAt(1);

String plainChar="";

int r1,c1,r2,c2;

r1=rowPos(a);

c1=columnPos(a);

r2=rowPos(b);

c2=columnPos(b);

if(c1==c2)

{

--r1;

--r2;

if(r1 < 0)

r1=4;

if(r2 < 0)

r2=4;

plainChar+=keyMatrix[r1][c2];

plainChar+=keyMatrix[r2][c1];

}

else if(r1==r2)

{

--c1;

--c2;

if(c1 < 0)

c1=4;

if(c2 < 0)

c2=4;

plainChar+=keyMatrix[r1][c1];

plainChar+=keyMatrix[r2][c2];

}

else{

plainChar+=keyMatrix[r1][c2];

plainChar+=keyMatrix[r2][c1];

}

return plainChar;

}

String Decrypt(String cipherText,String key)

{

String plainText="";

cipherText=cipherText.replaceAll("j", "i");

cipherText=cipherText.replaceAll(" ", "");

cipherText=cipherText.toUpperCase();

int len=cipherText.length();

for(int i=0;i < len-1;i=i+2)

{

plainText+=decryptChar(cipherText.substring(i,i+2));

}

return plainText;

}

}

class playfair{

public static void main(String args[])

{

PlayFairAlgo p=new PlayFairAlgo();

Scanner scn=new Scanner(System.in);

String key,cipherText,plainText,encryptedText;

int c;

System.out.println("Enter plaintext:");

plainText=scn.nextLine();

if(plainText.length() %2!=0)

plainText=plainText + "Z";

System.out.println("Enter Key:");

key=scn.nextLine();

cipherText=p.Encrypt(plainText,key);

encryptedText=p.Decrypt(cipherText, key);

System.out.println();

do

{

System.out.println("MENU");

System.out.println("1.Encrypt");

System.out.println("2.Decrypt");

System.out.println("3.Exit");

System.out.println("Enter choice:");

c = Integer.parseInt(scn.nextLine());

System.out.println();

if(c==1)

{

System.out.println("Encrypted:"+cipherText);

}

else if (c==2)

{

System.out.println("Decrypted:"+encryptedText);

}

}while(c!=3);

}

}

**Output:**

