Ex. No : 1	Mono-alphabetic Substitution Cipher

AIM:
Breaking the Mono-alphabetic Substitution Cipher using Frequency analysis method.

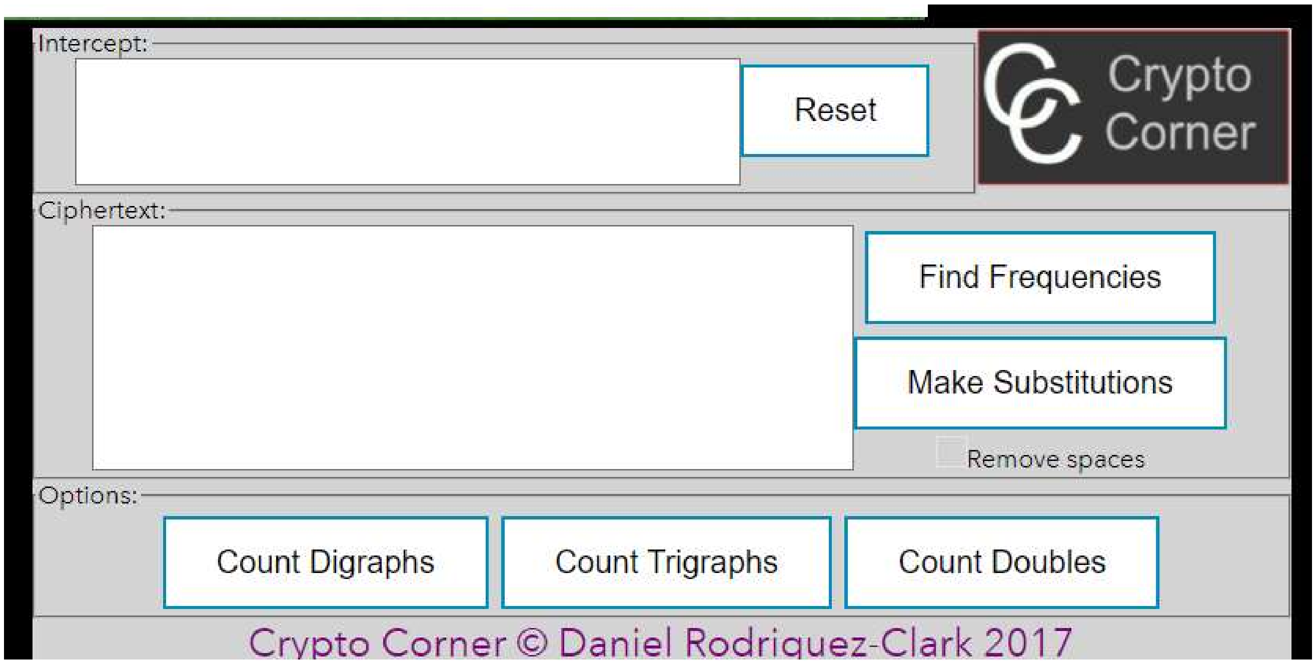
Theory:

The Method

The methodology behind frequency analysis relies on the fact that in any language, each letter has its own *personality*. The most obvious trait that letters have is the frequency with which they appear in a language. Clearly in English the letter "Z" appears far less frequently than, say, "A". In times gone by, if you wanted to find out the frequencies of letters within a language, you had to find a large piece of text and count each frequency. Now, however, we have computers that can do the hard work for us. But in fact, we don't even need to do this step, as for most languages there are databases of the letter frequencies, which have been calculated by looking at millions of texts, and are thus very highly accurate.
From these databases we find that "E" is the most common letter in English, appearing about 12% of the time (that is just over one in ten letters is an "E"). The next most common letter is "T" at 9%. The full frequency list is given by the graph below.

We can use this information to help us break a code given by a Monoalphabetic Substitution Cipher. This works because, if "e" has been encrypted to "X", then every "X" was an "e". Hence, the most common letter in the ciphertext should be "X".
Thus, if we intercept a message, and the most common letter is "P", we can guess that "P" was used to encrypt "e", and thus replace all the "P"'s with "e". Of course, not every text has exactly the same frequency, and as seen above, "t" and "a" have high frequencies too, so it could be that "P" was one of those. However, it is unlikely to be "z" as this is rare in the English Language. By repeating this process we can make good progress in breaking a message.

Analysis: click this link for simulator: https://crypto.interactive-maths.com/frequency-analysis-breaking-the-code.html



Result: ……………………………………………………………………………

Conclusion: ……………………………………………………………………….

Questions:

1. What is Mon alphabetic Cipher?

   ………………………………………………………………………………
   ……………………………………………………………………………….
   ……………………………………………………………………………....

2. What is substitution Technique?

   ………………………………………………………………………………
   ……………………………………………………………………………….
   ……………………………………………………………………………....

3. What is Cipher?

   ………………………………………………………………………………
   ……………………………………………………………………………….
   ……………………………………………………………………………....

4. Define term of permutation

   ………………………………………………………………………………
   ……………………………………………………………………………….

5. What is frequency diagram?

   ………………………………………………………………………………
   ……………………………………………………………………………….
   ……………………………………………………………………………....

Ex. No : 2	Design and Implement a product cipher using Substitution ciphers.

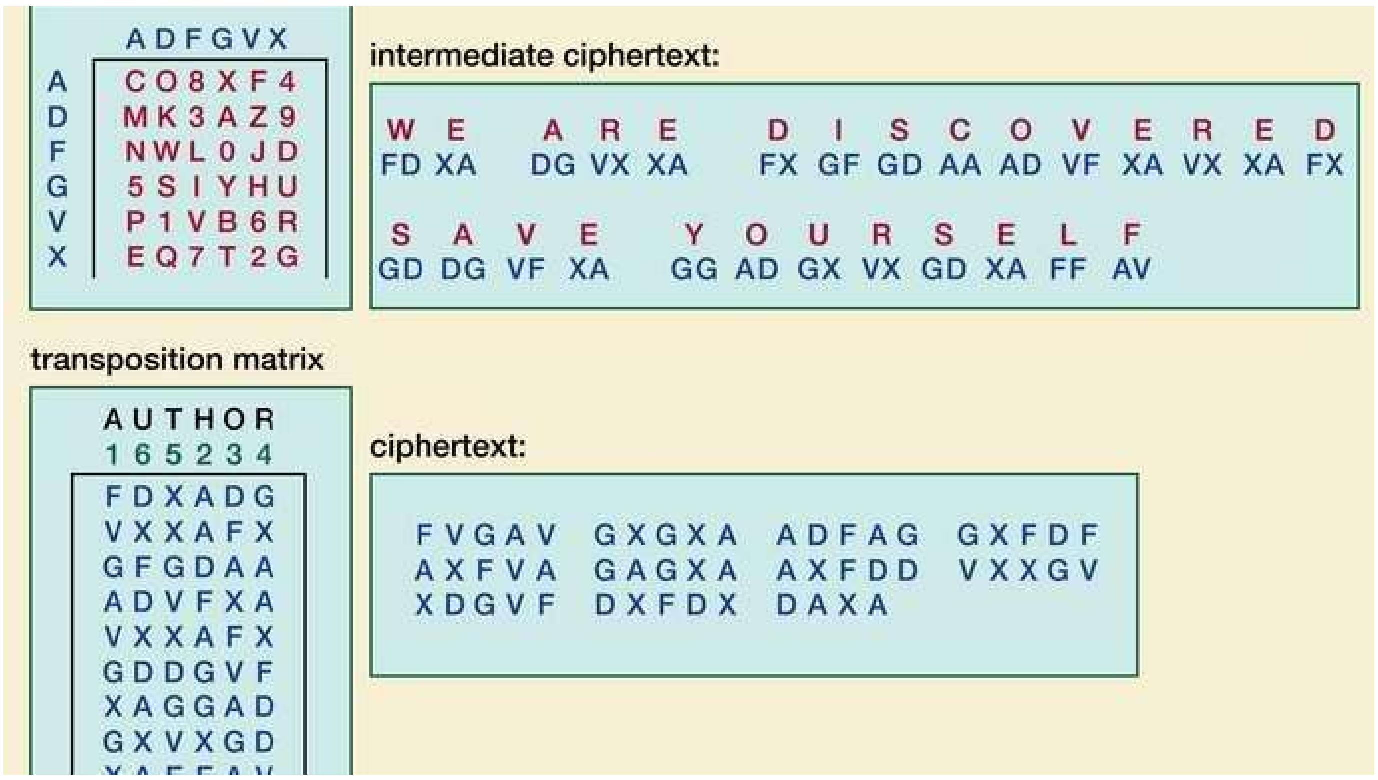
AIM:

Design and Implement a product cipher using Substitution ciphers.

Theory:

Product cipher, data encryption scheme in which the ciphertext produced by encrypting a plaintext document is subjected to further encryption. By combining two or more simple transposition ciphers or substitution ciphers, a more secure encryption may result.

transposition ciphers it was pointed out that by combining two or more simple transpositions, a more secure encryption may result. In the days of manual cryptography this was a useful device for the cryptographer, and in fact double transposition or product ciphers on key word-based rectangular matrices were widely used. There was also some use of a class of product ciphers known as fractionation systems, wherein a substitution was first made from symbols in the plaintext to multiple symbols (usually pairs, in which case the cipher is called a biliteral cipher) in the ciphertext, which was then encrypted by a final transposition, known as super encryption. One of the most famous field ciphers of all time was a fractionation system, the ADFGVX cipher employed by the German army during World War I. This system used a 6 × 6 matrix to substitution-encrypt the 26 letters and 10 digits into pairs of the symbols A, D, F, G, V, and X. The resulting biliteral cipher was then written into a rectangular array and route encrypted by reading the columns in the order indicated by a key word, as illustrated in the figure.



Program:

```
import java.util.*;

class ProductCipher {
public static void main(String args[]) {
System.out.println("Enter the input to be encrypted:");
String substitutionInput = new Scanner(System.in).nextLine();
System.out.println("Enter a number:");
int n = new Scanner(System.in).nextInt();

// Substitution encryption
StringBuffer substitutionOutput = new StringBuffer();
for(int i=0 ; i<substitutionInput.length() ; i++) {
char c = substitutionInput.charAt(i);
substitutionOutput.append((char) (c+5));
}
System.out.println("\nSubstituted text:");
System.out.println(substitutionOutput);

// Transposition encryption
String transpositionInput = substitutionOutput.toString();
int modulus;
if((modulus = transpositionInput.length()%n) != 0) {
modulus = n-modulus;
// 'modulus' is now the number of blanks/padding (X) to be appended
for( ; modulus!=0 ; modulus–) {
transpositionInput += "/";
}
}
StringBuffer transpositionOutput = new StringBuffer();
System.out.println("\nTransposition Matrix:");
for(int i=0 ; i<n ; i++) {
for(int j=0 ; j<transpositionInput.length()/n ; j++) {
char c = transpositionInput.charAt(i+(j*n));
System.out.print(c);
transpositionOutput.append(c);
}
System.out.println();
}
System.out.println("\nFinal encrypted text:");
System.out.println(transpositionOutput);

// Transposition decryption
n = transpositionOutput.length()/n;
StringBuffer transpositionPlaintext = new StringBuffer();
for(int i=0 ; i<n ; i++) {
for(int j=0 ; j<transpositionOutput.length()/n ; j++) {
```

```
char c = transpositionOutput.charAt(i+(j*n));
transpositionPlaintext.append(c);
}
}

// Substitution decryption
StringBuffer plaintext = new StringBuffer();
for(int i=0 ; i<transpositionPlaintext.length() ; i++) {
char c = transpositionPlaintext.charAt(i);
plaintext.append((char) (c-5));
}

System.out.println("\nPlaintext:");
System.out.println(plaintext);
}
}
```

Result:

……………………………………………………………

Conclusion: ………………………………………………..

Question

1. What is Product Cipher?
   ……………………………………………………………
   ……………………………………………………………..
   ……………………………………………………………
2. What is Transposition techniques?
   …………………………………………………………………..
   …………………………………………………………………..
   …………………………………………………………………..
3. What is differential cryptanalysis?
   …………………………………………………………………..
   …………………………………………………………………..
   …………………………………………………………………..
4. What makes a product cipher secure
   …………………………………………………………………..
   …………………………………………………………………..
   …………………………………………………………………..
5. What are some group-theoretic properties of product ciphers?

Ex. No : 3	Playfair Cipher

AIM:

To implement a program to encrypt a plain text and decrypt a cipher text using play fair Cipher substitution technique.

Theory:

1. To encrypt a message, one would break the message into digrams (groups of 2 letters)
2. For example, "HelloWorld" becomes "HE LL OW OR LD".
3. These digrams will be substituted using the key table.
4. Since encryption requires pairs of letters, messages with an odd number of characters usually append an uncommon letter, such as "X", to complete the final digram.
5. The two letters of the digram are considered opposite corners of a rectangle in the key table. To perform the substitution, apply the following 4 rules, in order, to each pair of letters in the plaintext:

PROGRAM:

playfairCipher.java

```
import java.awt.Point;

class playfairCipher {
   private  static  char[][]  charTable;  private
   static Point[] positions;

   private  static  String  prepareText(String  s,  boolean  chgJtoI)  {  s  =
      s.toUpperCase().replaceAll("[^A-Z]", "");
      return chgJtoI ? s.replace("J", "I") : s.replace("Q", "");
   }

   private  static  void  createTbl(String  key,  boolean  chgJtoI)  {  charTable  =
      new char[5][5];
      positions = new Point[26];
      String s = prepareText(key + "ABCDEFGHIJKLMNOPQRSTUVWXYZ", chgJtoI);
      int len = s.length();
      for (int i = 0, k = 0; i < len; i++) { char c
         = s.charAt(i);
```

```
        if  (positions[c  -  'A']  ==  null)  {
            charTable[k / 5][k % 5] = c;
            positions[c - 'A'] = new Point(k % 5, k / 5); k++;
        }
    }
}

private static String codec(StringBuilder txt, int dir) { int len =
    txt.length();
    for (int i = 0; i < len; i += 2) { char a =
        txt.charAt(i);
        char b = txt.charAt(i + 1);
        int row1 = positions[a - 'A'].y; int
        row2 = positions[b - 'A'].y; int col1
        = positions[a - 'A'].x; int col2 =
        positions[b - 'A'].x; if (row1 ==
        row2) {
            col1 = (col1 + dir) % 5; col2
            = (col2 + dir) % 5;
        } else if (col1 == col2) {

            row1 = (row1 + dir) % 5; row2
            = (row2 + dir) % 5;
        } else {
            int tmp = col1;
            col1 = col2; col2
            = tmp;
        }
        txt.setCharAt(i,                    charTable[row1][col1]);
        txt.setCharAt(i + 1, charTable[row2][col2]);
    }
    return txt.toString();
}

private static String encode(String s) {
    StringBuilder sb = new StringBuilder(s); for
    (int i = 0; i < sb.length(); i += 2) {
        if (i == sb.length() - 1) {
            sb.append(sb.length() % 2 == 1 ? 'X' : "");
        } else if (sb.charAt(i) == sb.charAt(i + 1)) {
            sb.insert(i + 1, 'X');
        }
    }
    return codec(sb, 1);
}

private static String decode(String s) { return
    codec(new StringBuilder(s), 4);
}

public static void main(String[] args) throws java.lang.Exception { String key =
    "CSE";
    String txt = "Security Lab"; /* make sure string length is even */ /* change J to I */
    boolean chgJtoI = true; createTbl(key,
    chgJtoI);
    String enc = encode(prepareText(txt, chgJtoI)); System.out.println("Simulating
    Playfair Cipher\n ------------------------------------------------------------------------ ");
    System.out.println("Input Message : " + txt);
    System.out.println("Encrypted Message : " + enc);
    System.out.println("Decrypted Message : " + decode(enc));
}
}
```

RESULT:

……………………………………………….

Conclusion.

……………………………………………..

Question:

1. What is Playfair cipher
2. What is third rule
3. What is fourth rule
4. What is size of matrix
5. How to break the code balloon