EX.NO: 1.b) ROLL.NO: 210701275

DATE:

Implement the Playfair Cipher technique

AIM:

To implement playfair cipher technique on the user input message.

ALGORITHM:

1. Initialize the Playfair key matrix based on the provided key, handling duplicates and 'J' substitution.
2. Preprocess the plaintext, removing non-alphabetic characters, converting to uppercase, and adding 'X' between consecutive identical characters.
3. Implement a method to retrieve the row and column positions of characters within the key matrix.
4. Encrypt the plaintext by iterating through character pairs, applying Playfair Cipher rules based on character positions, and constructing the ciphertext.
5. Accept user input for the key and plaintext, instantiate the Playfair Cipher, encrypt the plaintext, and output the ciphertext.

PROGRAM:

```
import java.util.*;

class PlayfairCipher {
   private char[][] keyMatrix;

   public PlayfairCipher(String key) {
      key = key.replaceAll("[Jj]", "I").toUpperCase();
      Set<Character> uniqueChars = new LinkedHashSet<>();
      for (char c : key.toCharArray()) {
         if (!Character.isLetter(c)) continue;
         uniqueChars.add(c);
      }
      StringBuilder keyBuilder = new StringBuilder();
      for (char c : uniqueChars) {
         keyBuilder.append(c);
      }
      String cleanKey = keyBuilder.toString();
      String alphabet = "ABCDEFGHIKLMNOPQRSTUVWXYZ";
      for (char c : cleanKey.toCharArray()) {
         alphabet = alphabet.replace(Character.toString(c), "");
      }
      cleanKey += alphabet;
      keyMatrix = new char[5][5];
      int row = 0, col = 0;
```

```
        for (char c : cleanKey.toCharArray()) {
            keyMatrix[row][col] = c;
            col++;
            if (col == 5) {
                col = 0;
                row++;
            }
        }
    }

    private String formatPlainText(String plainText) {
        plainText = plainText.replaceAll("[^A-Za-z]", "").toUpperCase();
        StringBuilder formattedText = new StringBuilder();
        for (int i = 0; i < plainText.length(); i++) {
            formattedText.append(plainText.charAt(i));
            if (i + 1 < plainText.length() && plainText.charAt(i) == plainText.charAt(i + 1)) {
                formattedText.append('X');
            }
        }
        if (formattedText.length() % 2 != 0) {
            formattedText.append('X');
        }
        return formattedText.toString();
    }

    private int[] getCharPos(char c) {
        int[] pos = new int[2];
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 5; j++) {
                if (keyMatrix[i][j] == c) {
                    pos[0] = i;
                    pos[1] = j;
                    return pos;
                }
            }
        }
        return pos;
    }

    public String encrypt(String plainText) {
        StringBuilder cipherText = new StringBuilder();
        plainText = formatPlainText(plainText);
        for (int i = 0; i < plainText.length(); i += 2) {
            char char1 = plainText.charAt(i);
            char char2 = plainText.charAt(i + 1);
            int[] pos1 = getCharPos(char1);
            int[] pos2 = getCharPos(char2);
            int row1 = pos1[0], col1 = pos1[1];
            int row2 = pos2[0], col2 = pos2[1];
            if (row1 == row2) {
```

```
                col1 = (col1 + 1) % 5;
                col2 = (col2 + 1) % 5;
            } else if (col1 == col2) {
                row1 = (row1 + 1) % 5;
                row2 = (row2 + 1) % 5;
            } else {
                int temp = col1;
                col1 = col2;
                col2 = temp;
            }
            cipherText.append(keyMatrix[row1][col1]);
            cipherText.append(keyMatrix[row2][col2]);
        }
        return cipherText.toString();
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the key: ");
        String key = scanner.nextLine();
        System.out.print("Enter the plaintext: ");
        String plainText = scanner.nextLine();
        PlayfairCipher cipher = new PlayfairCipher(key);
        String encryptedText = cipher.encrypt(plainText);
        System.out.println("Encrypted text: " + encryptedText);
    }
}
```

OUTPUT:

```
Enter the key: computer
Enter the plain text:communicate
Encrypted text : OMRMPCSGPTER
```

RESULT:

Thus the playfair cipher technique has been successfully compiled and executed .