**CNS Exp 3**

**Shashwat Tripathi  
D15A 64**

**Batch C**

**Aim:** Write a program in Java or Python to perform Cryptanalysis or decoding Playfair Cipher.

**Theory:**

The Playfair Cipher is a symmetric encryption technique that was designed to improve upon the security limitations of the simple Caesar cipher and other substitution ciphers. It employs a polygraphic substitution method, which means that blocks of letters are encrypted at a time. The key feature of the Playfair Cipher is the use of a 5x5 matrix (also called a key square) containing a mixed alphabet, where each letter appears only once. The matrix is usually generated using a keyword provided by the user.

### Encryption:

* Key Square Generation: Generate a key square (matrix) from a given keyword. This matrix will be used as the reference for encrypting and decrypting messages. The matrix consists of a unique combination of letters from the keyword and the remaining letters of the alphabet (excluding 'J' or replacing 'J' with 'I').
* Formatting the Message: The plaintext message is prepared for encryption. It is usually converted to uppercase, and non-alphabetic characters are removed. If there are repeating letters, they are separated by a filler, often 'X'. If the message length is odd, an 'X' might be added at the end to make it even.
* Letter Pairing: The formatted message is divided into pairs of letters.
* Encryption Process: For each letter pair:
  + If the letters are in the same row of the key square, each letter is replaced with the letter to its right (circularly).
  + If the letters are in the same column of the key square, each letter is replaced with the letter below it (circularly).
  + If the letters are neither in the same row nor in the same column, they form the vertices of a rectangle. Replace each letter with the letter located at the opposite corner of the rectangle formed by the two letters.
* Encrypted Message: The resulting encrypted pairs of letters make up the encrypted message.

### Decryption:

* Key Square Generation: Same as in encryption, generate the key square from the provided keyword.
* Decryption Process: For each letter pair in the encrypted message:
  + If the letters are in the same row, replace each letter with the letter to its left (circularly).
  + If the letters are in the same column, replace each letter with the letter above it (circularly).
  + If the letters form a rectangle, replace each letter with the letter located at the opposite corner of the rectangle.
* Decrypted Message: The resulting decrypted pairs of letters make up the decrypted message.

**Code:**

import java.util.HashMap;

public class PlayfairCryptanalysis {

public static char[][] createPlayfairMatrix(String key) {

key = key.replace("J", "I"); // Replace J with I

key = key.toUpperCase();

key = key.replaceAll("[^A-Z]", ""); // Remove non-alphabetic characters

key = key + "ABCDEFGHIKLMNOPQRSTUVWXYZ";

char[][] matrix = new char[5][5];

int k = 0;

HashMap<Character, Boolean> seen = new HashMap<>();

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

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

char c = key.charAt(k);

if (!seen.containsKey(c)) {

matrix[i][j] = c;

seen.put(c, true);

} else {

j--;

}

k++;

}

}

return matrix;

}

public static String decryptPlayfair(String ciphertext, String key) {

char[][] matrix = createPlayfairMatrix(key);

ciphertext = ciphertext.replace("J", "I"); // Replace J with I

ciphertext = ciphertext.toUpperCase();

ciphertext = ciphertext.replaceAll("[^A-Z]", ""); // Remove non-alphabetic characters

StringBuilder plaintext = new StringBuilder();

for (int i = 0; i < ciphertext.length(); i += 2) {

char c1 = ciphertext.charAt(i);

char c2 = ciphertext.charAt(i + 1);

int[] pos1 = findPosition(matrix, c1);

int[] pos2 = findPosition(matrix, c2);

if (pos1[0] == pos2[0]) { // Same row

plaintext.append(matrix[pos1[0]][(pos1[1] + 4) % 5]);

plaintext.append(matrix[pos2[0]][(pos2[1] + 4) % 5]);

} else if (pos1[1] == pos2[1]) { // Same column

plaintext.append(matrix[(pos1[0] + 4) % 5][pos1[1]]);

plaintext.append(matrix[(pos2[0] + 4) % 5][pos2[1]]);

} else { // Forming a rectangle

plaintext.append(matrix[pos1[0]][pos2[1]]);

plaintext.append(matrix[pos2[0]][pos1[1]]);

}

}

return plaintext.toString();

}

public static int[] findPosition(char[][] matrix, char letter) {

int[] position = new int[2];

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

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

if (matrix[i][j] == letter) {

position[0] = i;

position[1] = j;

return position;

}

}

}

return null;

}

public static void main(String[] args) {

String ciphertext = "VFYUGEDW";

String key = "QWERTY";

String plaintext = decryptPlayfair(ciphertext, key);

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

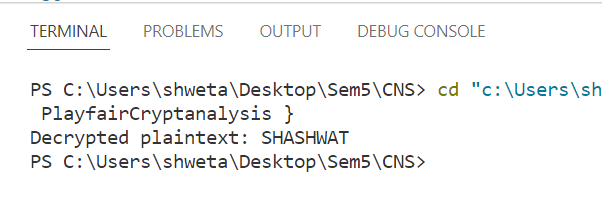
}

}

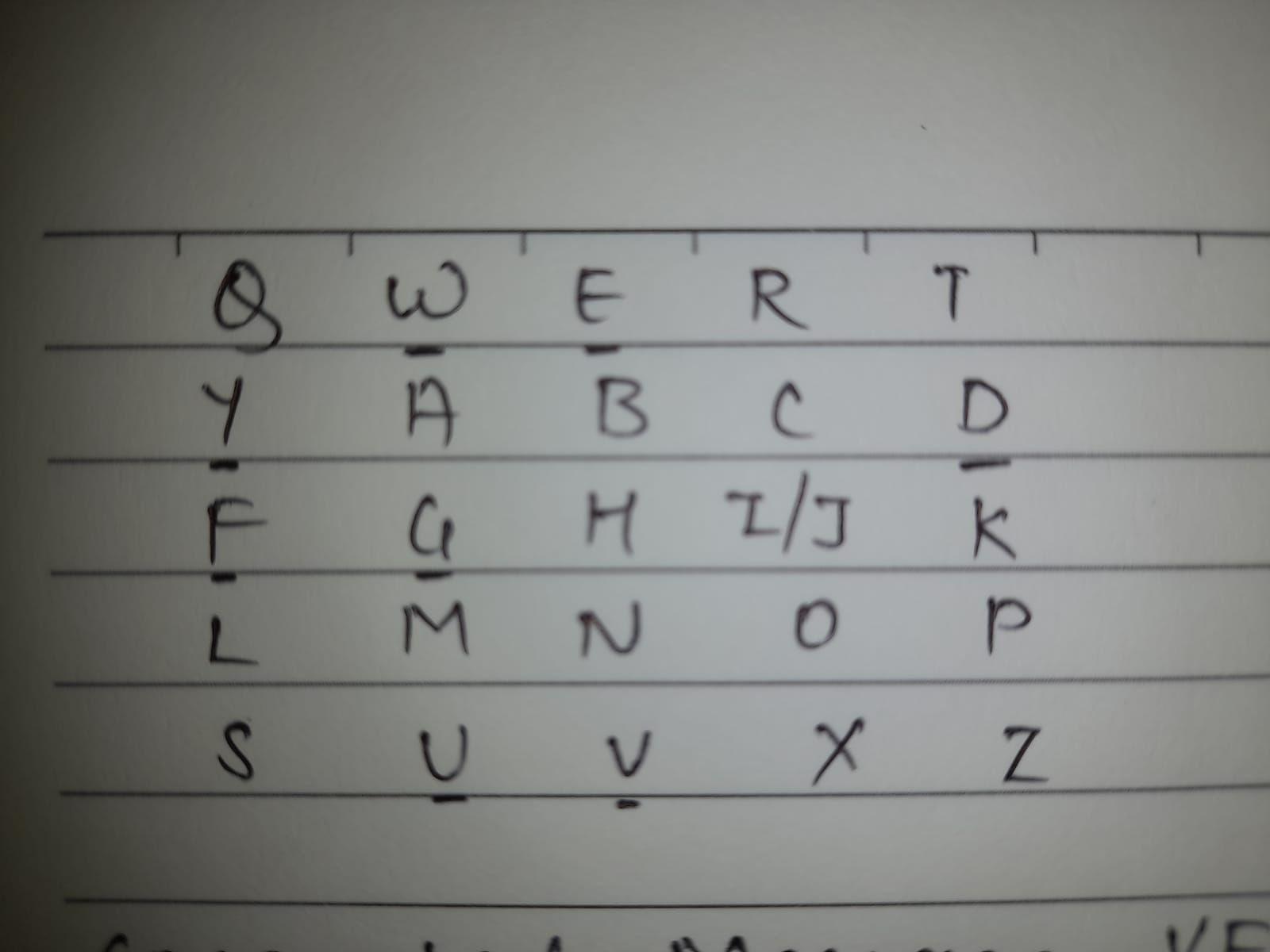
**Output:**

Cipher Text Used: VFYUGEDW  
Key Used: QWERTY

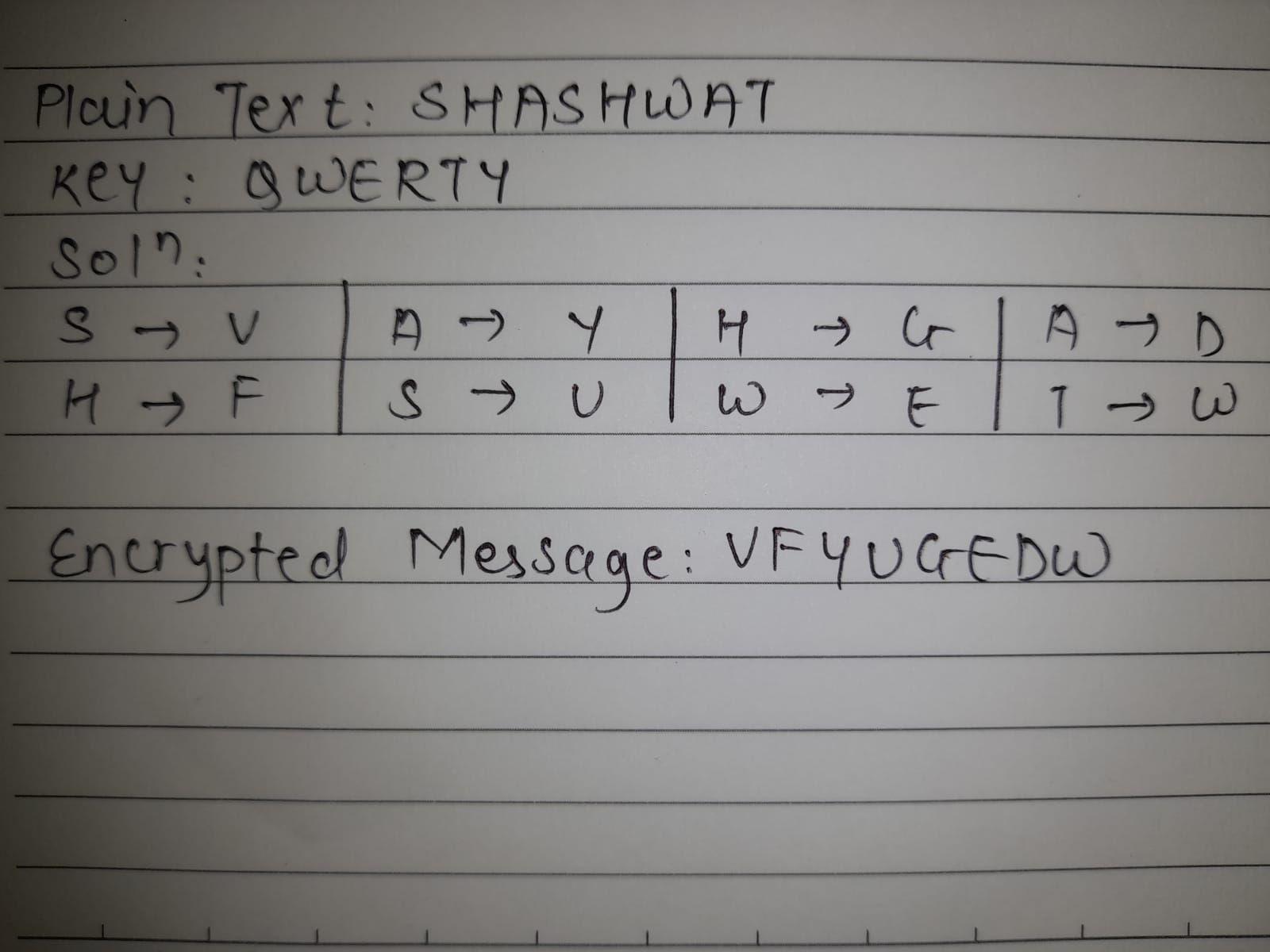
Decrypted Plain Text: **shashwat**

****

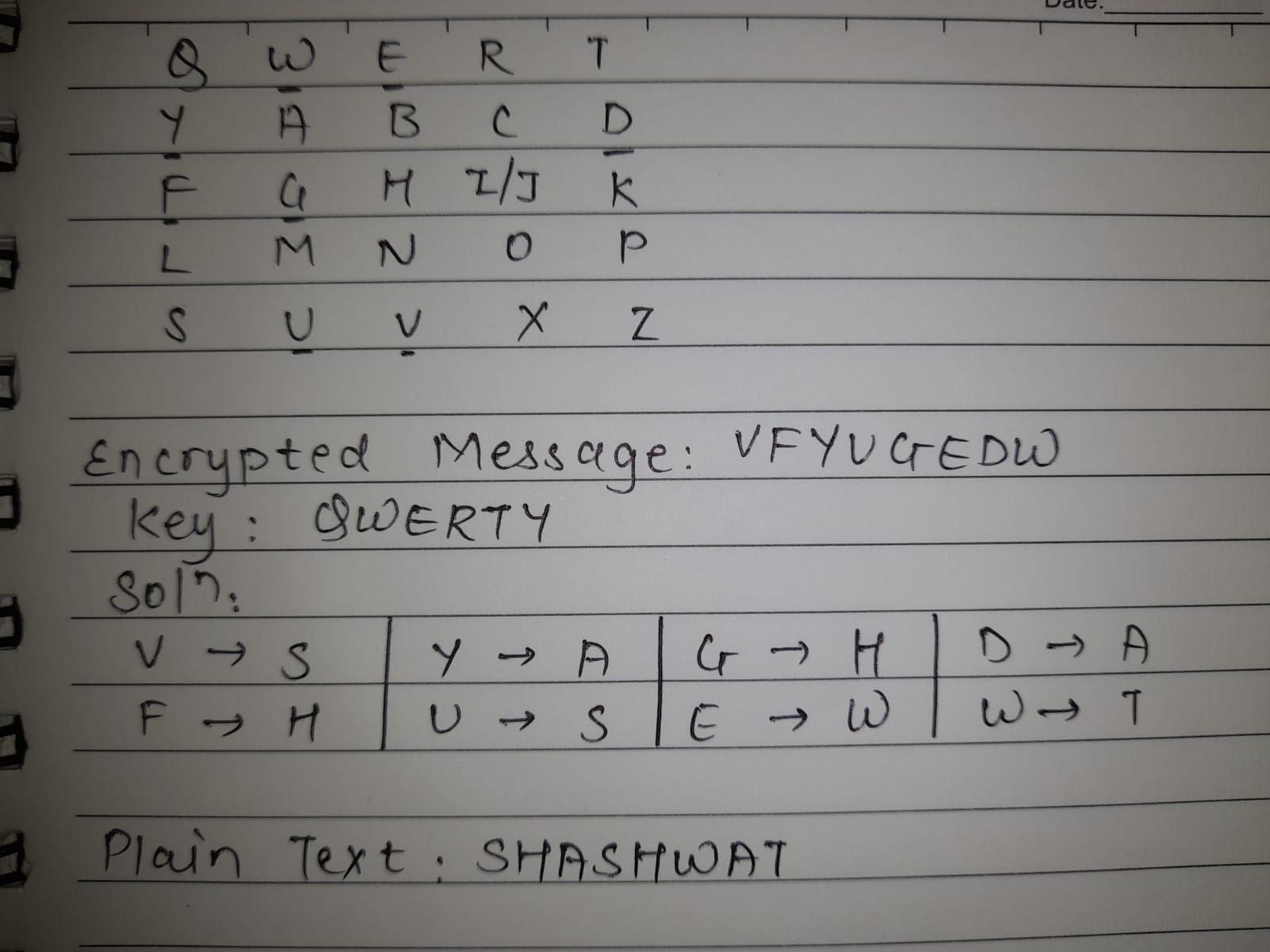
**Solving:**

****

Encryption

****

Decryption

****

**Conclusion:** Therefore, we have understood the encryption and Decryption methods of Playfair Cipher.