Code :

```
import java.util.*;

public class ColumnarTranspositionCipher {
    static final String key = "HACK";
    static Map<Character, Integer> keyMap = new HashMap<>();

    static void setPermutationOrder() {
        for (int i = 0; i < key.length(); i++) {
            keyMap.put(key.charAt(i), i);
        }
    }

    static String encryptMessage(String msg) {
        int row, col;
        StringBuilder cipher = new StringBuilder();

        col = key.length();

        row = (int) Math.ceil((double) msg.length() / col);

        char[][] matrix = new char[row][col];

        for (int i = 0, k = 0; i < row; i++) {
            for (int j = 0; j < col; ) {
                if (k < msg.length()) {
                    char ch = msg.charAt(k);
                    if (Character.isLetter(ch) || ch == ' ') {
                        matrix[i][j] = ch;
                        j++;
                    }
                    k++;
                } else {

                    matrix[i][j] = '_';
                    j++;
                }
            }
        }

        for (Map.Entry<Character, Integer> entry : keyMap.entrySet()) {
            int columnIndex = entry.getValue();

            for (int i = 0; i < row; i++) {
                if (Character.isLetter(matrix[i][columnIndex]) || matrix[i][columnIndex] == ' ' ||
matrix[i][columnIndex] == '_') {
```

```
                cipher.append(matrix[i][columnIndex]);
            }
        }
    }

    return cipher.toString();
}

static String decryptMessage(String cipher) {
    int col = key.length();

    int row = (int) Math.ceil((double) cipher.length() / col);
    char[][] cipherMat = new char[row][col];

    int k = 0;
    for (int j = 0; j < col; j++) {
        for (int i = 0; i < row; i++) {
            cipherMat[i][j] = cipher.charAt(k);
            k++;
        }
    }

    int index = 0;
    for (Map.Entry<Character, Integer> entry : keyMap.entrySet()) {
        entry.setValue(index++);
    }

    char[][] decCipher = new char[row][col];
    for (int l = 0; l < key.length(); l++) {
        int columnIndex = keyMap.get(key.charAt(l));
        for (int i = 0; i < row; i++) {
            decCipher[i][l] = cipherMat[i][columnIndex];
        }
    }

    StringBuilder msg = new StringBuilder();
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            if (decCipher[i][j] != '_') {
                msg.append(decCipher[i][j]);
            }
        }
    }

    return msg.toString();
}

public static void main(String[] args) {
        String msg = "Geeks for Geeks";

        setPermutationOrder();

        String cipher = encryptMessage(msg);
        System.out.println("Encrypted Message: " + cipher);

        System.out.println("Decrypted Message: " + decryptMessage(cipher));
    }
}
```

Output :

```
Encrypted Message: e  kefGsGsrekoe_
Decrypted Message: Geeks for Geeks
```