

```

    for (row2 = 0; row2 < 5; row2++) {
        for (col2 = 0; col2 < 5; col2++) {
            if (matrix[row2][col2] == toupper(plaintext[i + 1])) {
                break;
            }
        }
        if (col2 < 5) {
            break;
        }
    }

    if (row1 == row2) {
        ciphertext[k++] = matrix[row1][(col1 + 1) % 5];
        ciphertext[k++] = matrix[row2][(col2 + 1) % 5];
    } else if (col1 == col2) {
        ciphertext[k++] = matrix[(row1 + 1) % 5][col1];
        ciphertext[k++] = matrix[(row2 + 1) % 5][col2];
    } else {
        ciphertext[k++] = matrix[row1][col2];
        ciphertext[k++] = matrix[row2][col1];
    }
}
ciphertext[k] = '\0';
}

void playfair_decipher(char *ciphertext, char *key, char *plaintext) {
    char matrix[5][5];
    int i, k, row1, col1, row2, col2;
    int ciphertext_len = strlen(ciphertext);

    create_playfair_matrix(key, matrix);

    for (i = 0, k = 0; i < ciphertext_len; i += 2) {
        for (row1 = 0; row1 < 5; row1++) {
            for (col1 = 0; col1 < 5; col1++) {
                if (matrix[row1][col1] == toupper(ciphertext[i])) {
                    break;
                }
            }
        }
        if (col1 < 5) {
            break;
        }
    }
    for (row2 = 0; row2 < 5; row2++) {
        for (col2 = 0; col2 < 5; col2++) {
            if (matrix[row2][col2] == toupper(ciphertext[i + 1])) {
                break;
            }
        }
    }

```