

```

#include <stdio.h>
#include <string.h>
#include <ctype.h>

char matrix[5][5];

void generateMatrix(char key[]) {
    int used[26] = {0}, i, j, k = 0;
    used['J' - 'A'] = 1;
    for (i = 0; key[i]; i++) {
        char c = toupper(key[i]);
        if (c < 'A' || c > 'Z') continue;
        if (!used[c - 'A']) {
            matrix[k / 5][k % 5] = c;
            used[c - 'A'] = 1;
            k++;
        }
    }
    for (i = 0; i < 26; i++) {
        if (!used[i]) {
            matrix[k / 5][k % 5] = 'A' + i;
            k++;
        }
    }
}

void findPosition(char c, int *row, int *col) {
    if (c == 'J') c = 'I';
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 5; j++)
            if (matrix[i][j] == c) {
                *row = i;
                *col = j;
                return;
            }
}

void playfairEncrypt(char text[]) {
    int len = strlen(text);
    for (int i = 0; i < len; i += 2) {
        int r1, c1, r2, c2;
        findPosition(toupper(text[i]), &r1, &c1);
        findPosition(toupper(text[i + 1]), &r2, &c2);

        if (r1 == r2) {
            text[i] = matrix[r1][(c1 + 1) % 5];
            text[i + 1] = matrix[r2][(c2 + 1) % 5];
        } else if (c1 == c2) {
            text[i] = matrix[(r1 + 1) % 5][c1];
            text[i + 1] = matrix[(r2 + 1) % 5][c2];
        } else {
            text[i] = matrix[r1][c2];
            text[i + 1] = matrix[r2][c1];
        }
    }
}

void playfairDecrypt(char text[]) {
    int len = strlen(text);
    for (int i = 0; i < len; i += 2) {
        int r1, c1, r2, c2;
        findPosition(toupper(text[i]), &r1, &c1);
        findPosition(toupper(text[i + 1]), &r2, &c2);

        if (r1 == r2) {
            text[i] = matrix[r1][(c1 + 4) % 5];
            text[i + 1] = matrix[r2][(c2 + 4) % 5];
        }
    }
}

```

```
    } else if (c1 == c2) {
        text[i] = matrix[(r1 + 4) % 5][c1];
        text[i + 1] = matrix[(r2 + 4) % 5][c2];
    } else {
        text[i] = matrix[r1][c2];
        text[i + 1] = matrix[r2][c1];
    }
}

int main() {
    char key[100], text[100];
    printf("Enter key: ");
    gets(key);
    generateMatrix(key);

    printf("Enter text (even length, no J): ");
    gets(text);

    playfairEncrypt(text);
    printf("Encrypted text: %s\n", text);

    playfairDecrypt(text);
    printf("Decrypted text: %s\n", text);

    return 0;
}

// OUTPUT
// Enter key: MONARCHY
// Enter text (even length, no J): CODE
// Encrypted text: HNED
// Decrypted text: CODE
```

