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

#define MOD 26


int mod(int a, int m) {
    int res = a % m;
    return res < 0 ? res + m : res;
}


int mod_inverse(int a, int m) {
    a = mod(a, m);
    for (int x = 1; x < m; x++) {
        if ((a * x) % m == 1) return x;
    }
    return -1;
}


void encrypt(char *message, int key_matrix[2][2], char *encrypted) {
    int vector[2];
    for (int i = 0; i < 2; i++) {
        vector[i] = message[i] - 'A';
    }

    for (int i = 0; i < 2; i++) {
        encrypted[i] = mod(key_matrix[i][0] * vector[0] + key_matrix[i][1] * vector[1], MOD) + 'A';
    }
    encrypted[2] = '\0';
}


void decrypt(char *cipher, int key_matrix[2][2], char *decrypted) {
    int det = mod(key_matrix[0][0]*key_matrix[1][1] - key_matrix[0][1]*key_matrix[1][0], MOD);
    int det_inv = mod_inverse(det, MOD);

    if (det_inv == -1) {
        printf("Key matrix not invertible modulo 26.\n");
        return;
    }

    int inv_matrix[2][2];
    inv_matrix[0][0] = mod( key_matrix[1][1] * det_inv, MOD);
    inv_matrix[0][1] = mod(-key_matrix[0][1] * det_inv, MOD);
    inv_matrix[1][0] = mod(-key_matrix[1][0] * det_inv, MOD);
```