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

int key[3][3];
int invKey[3][3];

int determinant(int m[3][3]) {
    int det = m[0][0]*(m[1][1]*m[2][2] - m[1][2]*m[2][1])
            - m[0][1]*(m[1][0]*m[2][2] - m[1][2]*m[2][0])
            + m[0][2]*(m[1][0]*m[2][1] - m[1][1]*m[2][0]);
    det %= 26;
    if (det < 0) det += 26;
    return det;
}

int modInverse(int a) {
    a %= 26;
    for (int x = 1; x < 26; x++)
        if ((a * x) % 26 == 1)
            return x;
    return -1;
}

void inverseMatrix(int m[3][3], int inv[3][3]) {
    int det = determinant(m);
    int invDet = modInverse(det);
    if (invDet == -1) {
        printf("Key matrix is not invertible mod 26!\n");
        return;
    }

    int adj[3][3];
    adj[0][0] =  (m[1][1]*m[2][2] - m[1][2]*m[2][1]);
    adj[0][1] = -(m[0][1]*m[2][2] - m[0][2]*m[2][1]);
    adj[0][2] =  (m[0][1]*m[1][2] - m[0][2]*m[1][1]);
    adj[1][0] = -(m[1][0]*m[2][2] - m[1][2]*m[2][0]);
    adj[1][1] =  (m[0][0]*m[2][2] - m[0][2]*m[2][0]);
    adj[1][2] = -(m[0][0]*m[1][2] - m[0][2]*m[1][0]);
    adj[2][0] =  (m[1][0]*m[2][1] - m[1][1]*m[2][0]);
    adj[2][1] = -(m[0][0]*m[2][1] - m[0][1]*m[2][0]);
    adj[2][2] =  (m[0][0]*m[1][1] - m[0][1]*m[1][0]);

    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++) {
            inv[i][j] = adj[j][i] * invDet;
            inv[i][j] %= 26;
            if (inv[i][j] < 0) inv[i][j] += 26;
        }
}

void hillEncrypt(char text[]) {
    int len = strlen(text);
    for (int i = 0; i < len; i += 3) {
        int vector[3], result[3];
        for (int j = 0; j < 3; j++)
            vector[j] = text[i + j] - 'A';

        for (int j = 0; j < 3; j++) {
            result[j] = 0;
            for (int k = 0; k < 3; k++)
                result[j] += key[j][k] * vector[k];
            result[j] %= 26;
            text[i + j] = result[j] + 'A';
        }
    }
}
```

```
void hillDecrypt(char text[]) {
    int len = strlen(text);
    for (int i = 0; i < len; i += 3) {
        int vector[3], result[3];
        for (int j = 0; j < 3; j++)
            vector[j] = text[i + j] - 'A';

        for (int j = 0; j < 3; j++) {
            result[j] = 0;
            for (int k = 0; k < 3; k++)
                result[j] += invKey[j][k] * vector[k];
            result[j] %= 26;
            text[i + j] = result[j] + 'A';
        }
    }
}

int main() {
    char text[100];
    printf("Enter 3x3 key matrix:\n");
    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            scanf("%d", &key[i][j]);

    inverseMatrix(key, invKey);

    printf("Enter text (multiple of 3 letters, uppercase A-Z): ");
    scanf("%s", text);

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

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

    return 0;
}
```