main.c

```c
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define MOD 26
int charToInt(char c) {
    return toupper(c) - 'A';
}
char intToChar(int i) {
    return (i % MOD + MOD) % MOD + 'A';
}
int modInverse(int det) {
    det = (det % MOD + MOD) % MOD;
    for (int i = 1; i < MOD; i++) {
        if ((det * i) % MOD == 1)
            return i;
    }
    return -1;  // No inverse
}
void multiplyMatrix(int key[2][2], int vector[2], int result[2]) {
    for (int i = 0; i < 2; i++) {
        result[i] = 0;
        for (int j = 0; j < 2; j++) {
            result[i] += key[i][j] * vector[j];
        }
        result[i] %= MOD;
    }
}
int invertMatrix(int key[2][2], int inverse[2][2]) {
    int det = key[0][0]*key[1][1] - key[0][1]*key[1][0];
    int invDet = modInverse(det);
    if (invDet == -1) return 0;
    inverse[0][0] =  key[1][1] * invDet % MOD;
    inverse[0][1] = -key[0][1] * invDet % MOD;
    inverse[1][0] = -key[1][0] * invDet % MOD;
    inverse[1][1] =  key[0][0] * invDet % MOD;
    for (int i = 0; i < 2; i++)
        for (int j = 0; j < 2; j++)
            inverse[i][j] = (inverse[i][j] + MOD) % MOD;

    return 1;
}
```

```c
}
int main() {
    char plaintext[1000], key[100], ciphertext[1000], decrypted[1000];
    printf("Enter plaintext: ");
    fgets(plaintext, sizeof(plaintext), stdin);
    plaintext[strcspn(plaintext, "\n")] = 0;
    printf("Enter key: ");
    fgets(key, sizeof(key), stdin);
    key[strcspn(key, "\n")] = 0;
    encrypt(plaintext, key, ciphertext);
    printf("Encrypted: %s\n", ciphertext);
    decrypt(ciphertext, key, decrypted);
    printf("Decrypted: %s\n", decrypted);
    return 0;
}
```

input

```
Enter plaintext: ramana reddy
Enter key: cse
Encrypted: TSQCFE TWHFQ
Decrypted: RAMANA REDDY


...Program finished with exit code 0
Press ENTER to exit console.
```