main.c

Run

Output

Clear

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

#define SIZE 1000

// Maps characters A-Z to 0-25
int charToNum(char c) {
    return c - 'A';
}

char numToChar(int n) {
    return 'A' + (n % 26);
}

// Multiplies 2x2 matrix with 2x1 vector modulo
    26
void encryptPair(int key[2][2], int p1, int p2,
```

```
Preprocessed Plaintext:
    MEETMEATTHEUSUALPLACEATTENRATHERTHANEIGHTOCLOCKX

Encryption Steps:
ME -> 20,10 -> UK
ET -> 8,23 -> IX
ME -> 20,10 -> UK
AT -> 24,3 -> YD
TH -> 17,14 -> RO
EU -> 12,4 -> ME
SU -> 8,22 -> IW
AL -> 18,25 -> SZ
PL -> 23,22 -> XW
AC -> 8,14 -> IO
EA -> 10,20 -> KU
TT -> 13,20 -> NU
EN -> 10,7 -> KH
RA -> 23,7 -> XH
```

main.c

Run

```c
void decryptPair(int invKey[2][2], int c1, int c2, int *p1, int *p2) {
    *p1 = (invKey[0][0] * c1 + invKey[0][1] * c2) % 26;
    *p2 = (invKey[1][0] * c1 + invKey[1][1] * c2) % 26;
}

void preprocess(char *input, char *output) {
    int j = 0;
    for (int i = 0; input[i]; i++) {
        if (isalpha(input[i])) {
            output[j++] = toupper(input[i]);
        }
    }
    if (j % 2 != 0) output[j++] = 'X';  // Pad if odd length
```

Output

Clear

```
Preprocessed Plaintext:
    MEETMEATTHEUSUALPLACEATTENRATHERTHANEIGHTOCLOCKX

Encryption Steps:
ME -> 20,10 -> UK
ET -> 8,23 -> IX
ME -> 20,10 -> UK
AT -> 24,3 -> YD
TH -> 17,14 -> RO
EU -> 12,4 -> ME
SU -> 8,22 -> IW
AL -> 18,25 -> SZ
PL -> 23,22 -> XW
AC -> 8,14 -> IO
EA -> 10,20 -> KU
TT -> 13,20 -> NU
EN -> 10,7 -> KH
RA -> 23,7 -> XH
```

main.c

Run

Output

Clear

```
37
38  int main() {
39      char input[] = "meet me at the usual place
            at ten rather than eight oclock";
40      char cleaned[SIZE], encrypted[SIZE],
            decrypted[SIZE];
41      int key[2][2] = {{9, 4}, {5, 7}};
42      int invKey[2][2] = {{5, 10}, {15, 25}}; //
            Precomputed inverse of key matrix mod 26
43
44      preprocess(input, cleaned);
45      printf("Preprocessed Plaintext: %s\n",
            cleaned);
46
47      int len = strlen(cleaned);
48      int c1, c2;
49
50      printf("\nEncryption Steps:\n");
```

```
Preprocessed Plaintext:
    MEETMEATTHEUSUALPLACEATTENRATHERTHANEIGHTOCLOCKX

Encryption Steps:
ME -> 20,10 -> UK
ET -> 8,23 -> IX
ME -> 20,10 -> UK
AT -> 24,3 -> YD
TH -> 17,14 -> RO
EU -> 12,4 -> ME
SU -> 8,22 -> IW
AL -> 18,25 -> SZ
PL -> 23,22 -> XW
AC -> 8,14 -> IO
EA -> 10,20 -> KU
TT -> 13,20 -> NU
EN -> 10,7 -> KH
RA -> 23,7 -> XH
```

main.c

Run

```
59
60      printf("\nDecryption Steps:\n");
61      for (int i = 0; i < len; i += 2) {
62          decryptPair(invKey, charToNum
                (encrypted[i]), charToNum
                (encrypted[i + 1]), &c1, &c2);
63          decrypted[i] = numToChar(c1);
64          decrypted[i + 1] = numToChar(c2);
65          printf("%c%c -> %d,%d -> %c%c\n",
                encrypted[i], encrypted[i + 1], c1,
                c2, decrypted[i], decrypted[i + 1]);
66      }
67      decrypted[len] = '\0';
68      printf("\nDecrypted Text: %s\n", decrypted);
69
70      return 0;
71  }
72
```

Output

Clear

```
KH -> 16,13 -> QN
XH -> 3,0 -> DA
RO -> 17,7 -> RH
AJ -> 12,17 -> MR
RO -> 17,7 -> RH
AN -> 0,13 -> AN
QY -> 8,8 -> II
EB -> 4,7 -> EH
TL -> 23,14 -> XO
KJ -> 10,11 -> KL
EG -> 2,2 -> CC
AD -> 4,23 -> EX

Decrypted Text:
    SEKTSEUTRHWUAUCLXLYCQAFTQNDARHMRRHANIIEHXOKLCCEX


=== Code Execution Successful ===
```