12. a. Write a C program to Encrypt the message “meet me at the usual place at ten rather than eight oclock” using the Hill cipher with the key. 9 4 5 7

a. Show your calculations and the result.

b. Show the calculations for the corresponding decryption of the ciphertext to recover the original plaintext.

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#define MOD 26

void matrixMultiply(int key[2][2], int pair[2], int result[2]) {

result[0] = (key[0][0]\*pair[0] + key[0][1]\*pair[1]) % MOD;

result[1] = (key[1][0]\*pair[0] + key[1][1]\*pair[1]) % MOD;

}

int modInverse(int a) {

a %= MOD;

for (int x = 1; x < MOD; x++) {

if ((a \* x) % MOD == 1) return x;

}

return -1;

}

void inverseKey(int key[2][2], int invKey[2][2]) {

int det = (key[0][0]\*key[1][1] - key[0][1]\*key[1][0]) % MOD;

if (det < 0) det += MOD;

int invDet = modInverse(det);

if (invDet == -1) {

printf("Key matrix is not invertible.\n");

return;

}

invKey[0][0] = ( key[1][1] \* invDet) % MOD;

invKey[0][1] = (-key[0][1] \* invDet) % MOD;

invKey[1][0] = (-key[1][0] \* invDet) % MOD;

invKey[1][1] = ( key[0][0] \* invDet) % MOD;

for (int i = 0; i < 2; i++)

for (int j = 0; j < 2; j++)

if (invKey[i][j] < 0) invKey[i][j] += MOD;

}

void encryptHill(char \*message, int key[2][2]) {

int len = strlen(message);

printf("\nEncrypted Text:\n");

for (int i = 0; i < len; i += 2) {

int pair[2] = { message[i] - 'A', message[i+1] - 'A' };

int res[2];

matrixMultiply(key, pair, res);

printf("%c%c", res[0] + 'A', res[1] + 'A');

}

printf("\n");

}

void decryptHill(char \*cipher, int key[2][2]) {

int len = strlen(cipher);

int invKey[2][2];

inverseKey(key, invKey);

printf("\nDecrypted Text:\n");

for (int i = 0; i < len; i += 2) {

int pair[2] = { cipher[i] - 'A', cipher[i+1] - 'A' };

int res[2];

matrixMultiply(invKey, pair, res);

printf("%c%c", res[0] + 'A', res[1] + 'A');

}

printf("\n");

}

void preprocess(char \*input, char \*output) {

int k = 0;

for (int i = 0; input[i]; i++) {

if (isalpha(input[i])) {

output[k++] = toupper(input[i] == 'J' ? 'I' : input[i]);

}

}

if (k % 2 != 0) output[k++] = 'X';

output[k] = '\0';

}

int main() {

char text[] = "meet me at the usual place at ten rather than eight oclock";

char clean[500];

int key[2][2] = {{9, 4}, {5, 7}};

preprocess(text, clean);

printf("Cleaned Message: %s\n", clean);

encryptHill(clean, key);

char ciphertext[] = "KCLUBGUBDKXIJAFKXZQLNDWSJAGRLJCKYUVCDPVQGVQMLYHUG";

decryptHill(ciphertext, key);

return 0;

}

OUTPUT:

