14. Write a C program for one-time pad version of the Vigenère cipher. In this scheme, the key is a stream of random numbers between 0 and 26. For example, if the key is 3 19 5 . . . , then the first letter of plaintext is encrypted with a shift of 3 letters, the second with a shift of 19 letters, the third with a shift of 5 letters, and so on.

a. Encrypt the plaintext send more money with the key stream 9 0 1 7 23 15 21 14 11 11 2 8 9

b. Using the ciphertext produced in part (a), find a key so that the cipher text decrypts to the plaintext cash not needed.

#include <string.h>

#include <ctype.h>

#define MAX\_LEN 100

int charToNum(char c) {

return tolower(c) - 'a';

}

char numToChar(int n) {

return 'a' + n;

}

void encryptOTP(char plaintext[], int key[], char ciphertext[]) {

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

if (plaintext[i] == ' ') {

ciphertext[i] = ' ';

} else {

int p = charToNum(plaintext[i]);

int c = (p + key[i]) % 26;

ciphertext[i] = numToChar(c);

}

}

ciphertext[strlen(plaintext)] = '\0';

}

void decryptOTP(char ciphertext[], int key[], char plaintext[]) {

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

if (ciphertext[i] == ' ') {

plaintext[i] = ' ';

} else {

int c = charToNum(ciphertext[i]);

int p = (c - key[i] + 26) % 26;

plaintext[i] = numToChar(p);

}

}

plaintext[strlen(ciphertext)] = '\0';

}

void recoverKey(char plaintext[], char ciphertext[], int key[]) {

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

if (plaintext[i] == ' ') {

key[i] = -1;

} else {

int p = charToNum(plaintext[i]);

int c = charToNum(ciphertext[i]);

key[i] = (c - p + 26) % 26;

}

}

}

void printKey(int key[], int length) {

for (int i = 0; i < length; i++) {

if (key[i] == -1)

printf(" ");

else

printf("%2d ", key[i]);

}

printf("\n");

}

int main() {

char plaintext1[] = "send more money";

int key1[] = {9, 0, 1, 7, 23, 15, 21, 14, 11, 11, 2, 8, 9};

char ciphertext1[MAX\_LEN];

printf("=== Part A: Encryption ===\n");

encryptOTP(plaintext1, key1, ciphertext1);

printf("Plaintext : %s\n", plaintext1);

printf("Key : ");

printKey(key1, strlen(plaintext1));

printf("Ciphertext: %s\n\n", ciphertext1);

printf("=== Part B: Decryption and Key Recovery ===\n");

char ciphertext2[] = "bvnz bhed fxxmqz";

char plaintext2[] = "cash not needed";

int recoveredKey[MAX\_LEN];

recoverKey(plaintext2, ciphertext2, recoveredKey);

printf("Plaintext : %s\n", plaintext2);

printf("Ciphertext: %s\n", ciphertext2);

printf("Recovered Key: ");

printKey(recoveredKey, strlen(plaintext2));

return 0;

}

OUTPUT:

