Program playfair cipher in c

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#define SIZE 5

void prepareKeyMatrix(const char \*key, char keyMatrix[SIZE][SIZE]) {

int i = 0, j = 0, k = 0;

int visited[26] = {0};

while (k < strlen(key) || j < SIZE) {

if (k < strlen(key) && key[k] != 'j' && !visited[key[k] - 'a']) {

keyMatrix[i][j++] = key[k] == 'j' ? 'i' : key[k];

visited[key[k++] - 'a'] = 1;

} else if (j < SIZE && !visited[k + 'a']) {

keyMatrix[i][j++] = k + 'a';

visited[k++] = 1;

} else {

j = 0;

i++;

}

}

}

void findPosition(const char keyMatrix[SIZE][SIZE], char ch, int \*row, int \*col) {

if (ch == 'j') ch = 'i';

for (\*row = 0; \*row < SIZE; (\*row)++)

for (\*col = 0; \*col < SIZE; (\*col)++)

if (keyMatrix[\*row][\*col] == ch) return;

}

void encryptPlayfair(const char keyMatrix[SIZE][SIZE], char ch1, char ch2) {

int row1, col1, row2, col2;

findPosition(keyMatrix, ch1, &row1, &col1);

findPosition(keyMatrix, ch2, &row2, &col2);

if (row1 == row2) col1 = (col1 + 1) % SIZE, col2 = (col2 + 1) % SIZE;

else if (col1 == col2) row1 = (row1 + 1) % SIZE, row2 = (row2 + 1) % SIZE;

else col1 ^= col2 ^= col1 ^= col2;

printf("%c%c%c%c", keyMatrix[row1][col1], keyMatrix[row2][col2], keyMatrix[row2][col2], keyMatrix[row1][col1]);

}

int main() {

char keyMatrix[SIZE][SIZE], key[26], plaintext[100];

int i = 0;

printf("Enter the key: ");

fgets(key, sizeof(key), stdin);

key[strcspn(key, "\n")] = '\0';

prepareKeyMatrix(key, keyMatrix);

printf("Enter the plaintext: ");

fgets(plaintext, sizeof(plaintext), stdin);

plaintext[strcspn(plaintext, "\n")] = '\0';

while (i < strlen(plaintext)) {

char ch1 = tolower(plaintext[i++]), ch2 = (i < strlen(plaintext)) ? tolower(plaintext[i++]) : 'x';

encryptPlayfair(keyMatrix, ch1, ch2);

}

return 0;

}

Output

