Program decryption in playfair cipher in c

#include <stdio.h>

#include <string.h>

#define SIZE 5

void initializeKeyTable(const char \*keyword, char keyTable[SIZE][SIZE]) {

char alphabet[26] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'};

const char \*keyPtr = keyword;

int alphabetIndex = 0;

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

for (int j = 0; j < SIZE; j++) {

if (\*keyPtr == '\0') {

while (strchr(keyTable[0], alphabet[alphabetIndex]) != NULL) {

alphabetIndex++;

}

keyTable[i][j] = alphabet[alphabetIndex];

alphabetIndex++;

} else {

if (strchr(keyTable[0], \*keyPtr) == NULL) {

keyTable[i][j] = \*keyPtr;

} else {

j--;

}

keyPtr++;

}

}

}

}

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

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

for (int j = 0; j < SIZE; j++) {

if (keyTable[i][j] == ch) {

\*row = i;

\*col = j;

return;

}

}

}

}

void decryptPlayfair(const char \*ciphertext, const char keyTable[SIZE][SIZE], char \*plaintext) {

int len = strlen(ciphertext);

char first, second;

int row1, col1, row2, col2;

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

first = ciphertext[i];

second = ciphertext[i + 1];

findPositions(keyTable, first, &row1, &col1);

findPositions(keyTable, second, &row2, &col2);

if (row1 == row2) {

col1 = (col1 - 1 + SIZE) % SIZE;

col2 = (col2 - 1 + SIZE) % SIZE;

} else if (col1 == col2) {

row1 = (row1 - 1 + SIZE) % SIZE;

row2 = (row2 - 1 + SIZE) % SIZE;

} else {

int temp = col1;

col1 = col2;

col2 = temp;

}

plaintext[i] = keyTable[row1][col1];

plaintext[i + 1] = keyTable[row2][col2];

}

plaintext[len] = '\0';

}

int main() {

const char \*ciphertext = "KXJEY UREBE ZWEHE WRYTU HEYFS KREHE GOYFI WTTTU OLKSY CAJPO BOTEI ZONTX BYBNT GONEY CUZWR GDSON SXBOU YWRHE BAAHY USEDQ";

char keyTable[SIZE][SIZE];

char plaintext[200];

initializeKeyTable("KEYWORD", keyTable);

decryptPlayfair(ciphertext, keyTable, plaintext);

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

printf("Plaintext: %4s\n", plaintext);

return 0;

}

output