Rail Fence Transposition:

#include <stdio.h>

#include <string.h>

void encrypt(char \*text, int key) {

char res[100]; int len = strlen(text), idx = 0;

for (int r = 0; r < key; r++) {

int j = r, down = 1;

while (j < len) {

res[idx++] = text[j];

if (r == 0 || r == key - 1)

j += 2 \* (key - 1);

else {

j += down ? 2 \* (key - r - 1) : 2 \* r;

down = !down;

}

}

}

res[idx] = '\0';

strcpy(text, res);

}

void decrypt(char \*text, int key) {

int len = strlen(text), pos = 0;

char res[100], mark[100] = {0};

for (int r = 0; r < key; r++) {

int j = r, down = 1;

while (j < len) {

mark[j] = 1;

j += (r == 0 || r == key - 1) ? 2 \* (key - 1) :

(down ? 2 \* (key - r - 1) : 2 \* r);

down = !down;

}

}

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

if (mark[i]) res[i] = text[pos++];

int r = 0, dir = 1;

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

text[i] = res[r];

r += dir;

if (r == 0 || r == key - 1) dir = -dir;

}

text[len] = '\0';

}

int main() {

char text[100]; int key, ch;

printf("Enter text: "); scanf("%s", text);

printf("Enter key (rails): "); scanf("%d", &key);

printf("1.Encrypt 2.Decrypt: "); scanf("%d", &ch);

if (ch == 1) encrypt(text, key);

else if (ch == 2) decrypt(text, key);

else return printf("Invalid choice\n"), 1;

printf("Result: %s\n", text);

return 0;

}

