#include<iostream>

#include<string>

using namespace std;

string Encrypt(string text, int key)

{

if (key == 1) return text;

string result = "";

int len = text.length();

char rail[key][len];

// Fill the rail matrix with '\n'

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

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

rail[i][j] = '\n';

bool down = false;

int row = 0;

// Fill the rail matrix in a zigzag pattern

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

{

rail[row][col] = text[col];

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

down = !down;

row += down ? 1 : -1;

}

// Read the matrix row by row

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

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

if (rail[i][j] != '\n')

result += rail[i][j];

return result;

}

string Decrypt(string cipher, int key)

{

if (key == 1) return cipher;

string result = "";

int len = cipher.length();

char rail[key][len];

// Initialize rail matrix with '\n'

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

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

rail[i][j] = '\n';

// Mark the positions with '\*'

bool down = false;

int row = 0;

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

{

rail[row][col] = '\*';

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

down = !down;

row += down ? 1 : -1;

}

// Replace '\*' with characters from cipher

int index = 0;

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

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

if (rail[i][j] == '\*' && index < len)

rail[i][j] = cipher[index++];

// Read the matrix in zigzag to reconstruct the text

result = "";

row = 0;

down = false;

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

{

result += rail[row][col];

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

down = !down;

row += down ? 1 : -1;

}

return result;

}

int main()

{

string text;

cout << "Enter your text: ";

getline(cin, text);

int key;

cout << "Enter your key: ";

cin >> key;

string Encrypted = Encrypt(text, key);

cout << "Encrypted: " << Encrypted << endl;

string Decrypted = Decrypt(Encrypted, key);

cout << "Decrypted: " << Decrypted << endl;

return 0;

}