

CODE:

```
#include <bits/stdc++.h>

using namespace std;

string encryptRailFence(string text, int key) {

    char rail[key][(text.length())];

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

        for (int j = 0; j < text.length(); j++)

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

    bool dir_down = false;

    int row = 0, col = 0;

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

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

            dir_down = !dir_down;

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

        dir_down ? row++ : row--;

    }

    string result;

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

        for (int j = 0; j < text.length(); j++)

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

                result.push_back(rail[i][j]);

    return result;

}
```

```

string decryptRailFence(string cipher, int key) {
    char rail[key][cipher.length()];

    for (int i = 0; i < key; i++)
        for (int j = 0; j < cipher.length(); j++)
            rail[i][j] = '\n';

    bool dir_down;

    int row = 0, col = 0;

    for (int i = 0; i < cipher.length(); i++){
        if (row == 0)
            dir_down = true;

        if (row == key - 1)
            dir_down = false;

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

        dir_down ? row++ : row--;
    }

    int index = 0;

    for (int i = 0; i < key; i++)
        for (int j = 0; j < cipher.length(); j++)
            if (rail[i][j] == '*' && index < cipher.length())
                rail[i][j] = cipher[index++];

    string result;

    row = 0, col = 0;

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

```

```

        if (row == 0)

            dir_down = true;

        if (row == key - 1)

            dir_down = false;

        if (rail[row][col] != '*')

            result.push_back(rail[row][col++]);

        dir_down ? row++ : row--;

    }

    return result;

}

int main(){

    string og = "attack at once";

    cout<<"Original Message: "<<og<<endl;

    string encrypt = encryptRailFence(og, 2);

    cout<<"Encrypted Message: "<<encrypt<<endl;

    cout <<"Decrypted Message: "<<decryptRailFence(encrypt, 2) << endl;

    return 0;

}

```

OUTPUT:

Original Message: attack at once

Encrypted Message: atc toctaka ne

Decrypted Message: attack at once