

DAYSJ JARA ARISACA

CIFRADO POR FILAS

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

void mostrary(char M[][50], int f, int c) {
    for (int i = 0; i < f; ++i) {
        for (int j = 0; j < c; ++j) {
            cout << M[i][j] << " ";
        }
        cout << endl;
    }
}

string cifrar(string mensaje, int f) {
    string cifrado = "";
    int longitud = mensaje.length();
    int c = (longitud + f - 1) / f;
    char M[f][50];

    int in = 0;
    for (int j = 0; j < c; ++j) {
        for (int i = 0; i < f; ++i) {
            if (in < longitud)
                M[i][j] = mensaje[in++];
            else
                M[i][j] = ' ';
        }
    }

    cout << "Matriz:" << endl;
    mostrary(M, f, c);

    for (int i = 0; i < f; ++i) {
        for (int j = 0; j < c; ++j) {
            cifrado += M[i][j];
        }
    }

    return cifrado;
}

int main() {
    int f;
```

```

    cout << "cantidad de filas: ";
    cin >> f;

    ifstream archivo("C:\\Users\\User\\Desktop\\archivos
c++\\transposicion.txt");
    if (!archivo.is_open()) {
        cout << "Error " << endl;
        return 1;
    }

    string mensaje;
    getline(archivo, mensaje);
    archivo.close();

    string cifrado = cifrar(mensaje, f);
    cout << "cifrado: " << cifrado << endl;

    return 0;
}

```

```

cantidad de filas: 3
Matriz:
t n o c n r l
r s s i p f a
a p i o o i s
cifrado: tnocnrlrssipfaapioois

```

CIFRADO POR COLUMNAS

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

void mostrar(char M[][50], int filas, int columnas) {
    for (int i = 0; i < filas; ++i) {
        for (int j = 0; j < columnas; ++j) {
            cout << M[i][j] << " ";
        }
        cout << endl;
    }
}

string cifrarPorColumnas(string mensaje, int columnas) {
    int longitud = mensaje.length();
    int filas = (longitud + columnas - 1) / columnas;
    char M[filas][50];

    int indice = 0;
    for (int i = 0; i < filas; ++i) {
        for (int j = 0; j < columnas; ++j) {
            if (indice < longitud)
                M[i][j] = mensaje[indice++];
            else
                M[i][j] = ' ';
        }
    }

    cout << "Matriz:" << endl;
    mostrar(M, filas, columnas);

    string cifrado;
    for (int j = 0; j < columnas; ++j) {
        for (int i = 0; i < filas; ++i) {
            cifrado += M[i][j];
        }
    }

    return cifrado;
}

int main() {
    int columnas;
    cout << "numero de columnas: ";
    cin >> columnas;
```

```

        cin.ignore();

        ifstream archivo("C:\\Users\\User\\Desktop\\archivos
c++\\transposicion2.txt");
        if (!archivo.is_open()) {
            cout << "Error " << endl;
            return 1;
        }

        string mensaje;
        getline(archivo, mensaje);
        archivo.close();

        string cifrado = cifrarPorColumnas(mensaje, columnas);
        cout << "Cifrado: " << cifrado << endl;

        return 0;
    }

```

```

numero de columnas: 5
Matriz:
t r a n s
p o s i c
i o n p o
r c o l u
m n a s
Cifrado: tpirmroocnasnoaniplsscou

```

CIFRADO RAIL FENCE

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

void mostrarMatriz(char M[][50], int f, int c) {
    for (int i = 0; i < f; ++i) {
        for (int j = 0; j < c; ++j) {
            if (M[i][j] == '\0')
                cout << "- ";
            else
                cout << M[i][j] << " ";
        }
        cout << endl;
    }
}

string cifrar(string x, int f) {
    string cifrado = "";
    int l = x.length();
    char M[f][50] = {0};

    int fila = 0, columna = 0;
    bool hacia_abajo = false;

    for (int i = 0; i < l; ++i) {
        if (fila == 0 || fila == f - 1)
            hacia_abajo = !hacia_abajo;

        if (x[i] != ' ') {
            M[fila][columna] = x[i];
            fila += (hacia_abajo) ? 1 : -1;
            columna++;
        } else {
            M[fila][columna] = '-';
            columna++;
        }
    }

    cout << "Matriz Rail Fence:" << endl;
    mostrarMatriz(M, f, columna);
    cout << endl;

    for (int i = 0; i < f; ++i) {
        for (int j = 0; j < columna; ++j) {
            if (M[i][j] != 0)
```

```

        cifrado += M[i][j];
    }
}

return cifrado;
}

int main() {
    int f;
    cout << "Cantidad de filas: ";
    cin >> f;

    cin.ignore();

    ifstream archivo("C:\\Users\\User\\Desktop\\archivos
c++\\transposicion.txt");
    if (!archivo.is_open()) {
        cout << "Error al abrir el archivo." << endl;
        return 1;
    }

    string x;
    getline(archivo, x);
    archivo.close();

    string cifrado = cifrar(x, f);
    cout << "Texto cifrado: " << cifrado << endl;

    return 0;
}

```

```

Cantidad de filas: 3
Matriz Rail Fence:
t - - - s - - - i - - - n - - - f - - - s
- r - n - p - s - c - o - p - r - i - a -
- - a - - - o - - - i - - - o - - - l - -

Texto cifrado: tsinfsrnpscopriaaio1

```

