

```

#include <iostream>
#include <fstream>
#include <vector>
#include <string>

using namespace std;

void inicializarMatrizVigenere(vector<vector<char>>& matriz) {
    matriz.resize(26, vector<char>(26));
    char letra = 'A';
    for (int i = 0; i < 26; ++i) {
        for (int j = 0; j < 26; ++j) {
            matriz[i][j] = letra;
            ++letra;
            if (letra > 'Z') {
                letra = 'A';
            }
        }
        ++letra;
    }
}

void imprimirMatrizVigenere(const vector<vector<char>>& matriz) {
    cout << "Matriz Vigenere:" << endl;
    for (int i = 0; i < 26; ++i) {
        for (int j = 0; j < 26; ++j) {
            cout << matriz[i][j] << " ";
        }
        cout << endl;
    }
}

string cifrarMensajeVigenere(const string& mensajeOriginal, const string&
clave) {
    string mensajeCifrado = mensajeOriginal;

    for (size_t i = 0, j = 0; i < mensajeOriginal.length(); ++i) {
        if (isalpha(mensajeOriginal[i])) {
            char base = isupper(mensajeOriginal[i]) ? 'A' : 'a';
            mensajeCifrado[i] = (mensajeOriginal[i] - base + clave[j %
clave.length()] - 'A') % 26 + base;
            ++j;
        }
    }

    return mensajeCifrado;
}

```

```

string descifrarMensajeVigenere(const string& mensajeCifrado, const
string& clave) {
    string mensajeDescifrado = mensajeCifrado;

    for (size_t i = 0, j = 0; i < mensajeCifrado.length(); ++i) {
        if (isalpha(mensajeCifrado[i])) {
            char base = isupper(mensajeCifrado[i]) ? 'A' : 'a';
            mensajeDescifrado[i] = (mensajeCifrado[i] - clave[j %
clave.length()] + 26) % 26 + base;
            ++j;
        }
    }

    return mensajeDescifrado;
}

int main() {

    ifstream archivoEntrada("C:\\Users\\User\\Desktop\\archivos
c++\\cifradodevigenere.txt");
    if (!archivoEntrada) {
        cerr << "Error " << endl;
        return 1;
    }

    string mensajeOriginal;
    getline(archivoEntrada, mensajeOriginal);
    archivoEntrada.close();

    string clave;
    cout << "clave: ";
    cin >> clave;

    vector<vector<char>> matrizVigenere;
    inicializarMatrizVigenere(matrizVigenere);
    imprimirMatrizVigenere(matrizVigenere);

    string mensajeCifrado = cifrarMensajeVigenere(mensajeOriginal,
clave);
    cout << "\\ncifrado: " << mensajeCifrado << endl;

    string mensajeDescifrado = descifrarMensajeVigenere(mensajeCifrado,
clave);
    cout << "\\ndescifrado: " << mensajeDescifrado << endl;

    return 0;
}

```

clave: CIELO

Matriz Vigenere:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	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	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

cifrado: JMVXCUW

descifrado: HERMOSO