```
NECESARIO queue_eda.h y stack_eda.h

#include <iostream>
#include <fstream>
#include <string>
#include "queue_eda.h"
#include "stack_eda.h"

using namespace std;



bool vocal(char c) {
    return tolower(c) == 'a' || tolower(c) == 'e' || tolower(c) == 'i' || tolower(c) ==
'o' || tolower(c) == 'u';
}

bool resuelveCaso() {


    string mensaje;

    getline(cin, mensaje);

    if (!cin) return false;


    //primera fase

    int i = 0;

    queue<char> dec1;
    stack<char> pilaF;


    for (char c : mensaje) {

        if (i % 2 == 0) {

            dec1.push(c);
        }
        else {

            pilaF.push(c);
        }

        i++;
    }

    while (!pilaF.empty()) {

        dec1.push(pilaF.top());
        pilaF.pop();
    }
```

```
    //segunda fase

    string frase;
    stack<char> cons;

    while (!dec1.empty()) {

        if (vocal(dec1.front())) {

            if (!cons.empty()) {
                while (!cons.empty()) {

                    frase.push_back(cons.top());
                    cons.pop();
                }
            }
            frase.push_back(dec1.front());
        }
        else {

            cons.push(dec1.front());
        }

        dec1.pop();
    }

    if (!cons.empty()) {
        while (!cons.empty()) {
            frase.push_back(cons.top());
            cons.pop();
        }
    }

    cout << frase << endl;

    return true;
}


int main() {
# ifndef DOMJUDGE
    std::ifstream in("datos.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf()); // save old buf and redirect std :: cin to
casos .txt
# endif


    while (resuelveCaso()) {}

# ifndef DOMJUDGE // para dejar todo como estaba al principio
    std::cin.rdbuf(cinbuf);
    system(" PAUSE ");
# endif
```

```
    return 0;
}
```