

```

#include <iostream>
#include <fstream>
#include <string>
#include "stack_eda.h"
using namespace std;

bool resuelveCaso() {

    stack<char>pila;
    int i = 0;
    string cadena;

    getline(cin, cadena);

    if (!cin) return false;

    while (i < cadena.size()) {
        if (cadena[i] == '(' || cadena[i] == '[' || cadena[i] == '{') {
            pila.push(cadena[i]);
        }
        else {
            if (cadena[i] == ')') {
                if (!pila.empty() && pila.top() == '(') {
                    pila.pop();
                }
                else {
                    pila.push(cadena[i]);
                }
            }
            else if (cadena[i] == ']') {
                if (!pila.empty() && pila.top() == '[') {
                    pila.pop();
                }
                else {
                    pila.push(cadena[i]);
                }
            }
            else if (cadena[i] == '}') {
                if (!pila.empty() && pila.top() == '{') {
                    pila.pop();
                }
                else {
                    pila.push(cadena[i]);
                }
            }
        }
        i++;
    }

    if (pila.empty()) {
        cout << "SI" << endl;
    }
}

```

```

    }
    else {
        cout << "NO" << 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;
}

```