

```

NECESARIO "bintree_eda.h"

#include <iostream>
#include <fstream>
#include "bintree_eda.h"

using namespace std;

template <typename T>
bool simetrico(bintree <T> const& a, bintree <T> const& b) {

    if (a.empty() && b.empty()) {
        return true;
    }
    else if ((a.empty() && !b.empty()) || (!a.empty() && b.empty())) {
        return false;
    }
    else {
        bool simetrico1 = simetrico(a.left(), b.right());
        bool simetrico2 = simetrico(a.right(), b.left());

        if (simetrico1 && simetrico2) {
            return true;
        }
        else {
            return false;
        }
    }
}

}

void resuelveCaso() {

    bintree <char > arbol = leerArbol('.');

    bool si = simetrico(arbol.left(), arbol.right());

    if (si) cout << "SI";
    else cout << "NO";
    cout << endl;
}

int main() {
    // Para la entrada por fichero.
    // Comentar para acepta el reto
#ifndef DOMJUDGE
    ifstream in("datos.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf()); //save old buf and redirect std::cin to
casos.txt
#endif
}

```

```
int numCasos;
cin >> numCasos;
for (int i = 0; i < numCasos; ++i)
    resuelveCaso();

    // Para restablecer entrada. Comentar para acepta el reto
#ifdef DOMJUDGE // para dejar todo como estaba al principio
    std::cin.rdbuf(cinbuf);
    system("PAUSE");
#endif

    return 0;
}
```