

```

NECESARIO "bintree_eda.h"

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

using namespace std;

struct tSol {
    int diametro;
    int alturaMax;
};

template <class T >
tSol diametro(bintree<T> const& arbol) {

    if (arbol.empty()) return { 0, 0 };
    else {

        tSol izq = diametro(arbol.left());
        tSol der = diametro(arbol.right());

        tSol solucion;

        solucion.alturaMax = max(izq.alturaMax, der.alturaMax) + 1;
        int di = max(izq.diametro, der.diametro);

        solucion.diametro = max(di, izq.alturaMax + der.alturaMax + 1);

        return solucion;
    }
}

void resuelveCaso() {

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

    tSol solucion = diametro(arbol);

    cout << solucion.diametro << 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;
}
```