

```

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

using namespace std;

struct tSol {
    int acumulados;
    int suma;
};

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

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

        tSol solucion;
        tSol izq = ac(arbol.left());
        tSol der = ac(arbol.right());

        solucion.suma = izq.suma + der.suma;

        solucion.acumulados = izq.acumulados + der.acumulados;

        if (solucion.suma == arbol.root()) {

            solucion.acumulados++;
        }

        solucion.suma += arbol.root();

        return solucion;
    }
}

void resuelveCaso() {
    bintree <int > arbol = leerArbol(-1);
    tSol sol = ac(arbol);
    cout << sol.acumulados << 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;
}
```