

```

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

using namespace std;

void resuelveCaso() {

    char elem;
    char vacioStr = '.';
    int vacioInt = -1;

    cin >> elem;

    if (elem == 'N') { //si es un numero

        int es;

        cin >> es;

        bintree_eda_ext <int> a = leerArbol_ext(vacioInt);

        int nivel = a.level(es);

        if (nivel == 0) cout << "NO EXISTE\n";
        else cout << nivel << endl;
    }
    else if (elem == 'C') { //si es una palabra

        char esc;

        cin >> esc;

        bintree_eda_ext <char> a = leerArbol_ext(vacioStr);

        int nivel = a.level(esc);

        if (nivel == 0) cout << "NO EXISTE\n";
        else cout << nivel << 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;
    std::cin >> numCasos;
    for (int i = 0; i < numCasos; ++i)
        resuelveCaso();

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

    return 0;
}
```