

```
NECESARIO "bintree_eda.h"
```

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

```
using namespace std;
```

```
struct tSol {
    int elem;
    int prof;
};
```

```
bool esPrimo(int numero) {

    // Casos especiales

    if (numero == 0 || numero == 1 || numero == 4) return false;

    for (int x = 2; x < numero / 2; x++) {

        if (numero % x == 0) return false;
    }

    // Si no se pudo dividir por ninguno de los de arriba, s♦ es primo
    return true;
}
```

```
template <typename T>
tSol barrera(bintree <T> const& a) {

    if (a.empty() || esPrimo(a.root())) return { -1, 0 };
    else {

        tSol izq = barrera(a.left());
        tSol der = barrera(a.right());

        if (a.root() % 7 == 0) {

            return { a.root(), 1 };
        }
        else if (izq.elem != -1 && der.elem != -1) {

            if (izq.prof <= der.prof) return { izq.elem, izq.prof + 1 };
            else return { der.elem, der.prof + 1 };
        }
        else if (izq.elem != -1 && der.elem == -1) {

            return { izq.elem, izq.prof + 1 };
        }
        else if (izq.elem == -1 && der.elem != -1) {
```

```

        return { der.elem, der.prof + 1 };
    }
    else {

        return { -1, 0 };
    }

}

}

void resuelveCaso() {

    bintree <int > arbol = leerArbol(-1);

    tSol solucion = barrera(arbol);

    if (solucion.elem == -1) cout << "NO HAY";
    else cout << solucion.elem << " " << solucion.prof;

    cout << endl;
}

int main() {
    // Para la entrada por fichero.
    // Comentar para acepta el reto
#ifdef 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;
}

```