

```
// Ejercicio 2. El regreso de los enanos
```

```
#include <iostream>
```

```
#include <fstream>
```

```
#include "bintree_eda.h"
```

```
using namespace std;
```

```
struct tSol {  
    string lider;  
    int num;  
};
```

```
tSol enanos(bintree<string> const& arbol) {
```

```
    if (arbol.empty()) {  
        return{ "Ninguno", 0 };  
    }
```

```
    else if (arbol.left().empty() && arbol.right().empty()) {  
        return { arbol.root(), 1};  
    }
```

```
    else {
```

```
        tSol izq = enanos(arbol.left());  
        tSol der = enanos(arbol.right());
```

```
        tSol sol;
```

```
        if (izq.num > der.num) {  
            sol.lider = izq.lider;  
        }
```

```
        else if (der.num > izq.num) {  
            sol.lider = der.lider;  
        }
```

```
        else {  
            if (izq.lider >= der.lider) {  
                sol.lider = der.lider;  
            }  
            else {  
                sol.lider = izq.lider;  
            }  
        }
```

```
        sol.num = izq.num + der.num;
```

```
        if (arbol.root() == "Orcos") {
```

```
            if (sol.num == 1) {
```

```
                sol.lider = "Ninguno";  
                sol.num = 0;
```

```

    }
    else if(sol.num != 0){

        int restantes;

        if(sol.num % 2 == 0) restantes = sol.num / 2;
        else restantes = (sol.num / 2);

        sol.num = restantes;
    }
}

return sol;
}
}

void resuelveCaso() {

    string punto = ".";

    bintree<string> arbol = leerArbol(punto);

    tSol sol = enanos(arbol);

    cout << sol.lider;

    if (sol.num != 0) {
        cout << " " << sol.num;
    }

    cout << endl;
}

int main() {
    // ajustes para que cin extraiga directamente de un fichero
#ifdef DOMJUDGE
    std::ifstream in("datos.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf());
#endif

    int casos = 0;

    cin >> casos;

    for (int i = 0; i < casos; i++) {
        resuelveCaso();
    }

    // para dejar todo como estaba al principio
#ifdef DOMJUDGE
    std::cin.rdbuf(cinbuf);
    system("PAUSE");
#endif
}

```

```
#endif  
    return 0;  
}
```