

```

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

using namespace std;

template <class T >
void minimo(const bintree<T> arbol, T& min) {

    if (!arbol.empty()) {

        if (arbol.root() < min) min = arbol.root();

        minimo(arbol.left(), min);
        minimo(arbol.right(), min);
    }
}

```

```

bool resuelveCaso() {

    string clase;
    cin >> clase;

    if (!cin) return false;

    if (clase == "N") {

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

        minimo(arbol, min);

        cout << min << endl;
    }
    else if (clase == "P") {

        string a = "#";
        bintree<string> arbol = leerArbol(a);
        string min = arbol.root();

        minimo(arbol, min);

        cout << min << endl;
    }

    return true;
}

```

```

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

    while( 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;
}
```