

```

#include <stdio.h>
#include <iostream>
#include <fstream>
#include <string>
#include <vector>

using namespace std;

/*COSTE:
*
*          k          n = 0
*  T(n):
*          T(n/2) + k'    n >= 1
*
*
*  O(logn)
*/

//Pre: 0 <= i <= j <= N - 1 && forall u: 0 <= u < longitud(lista) - 1: lista[u] != lista[u
+ 1]
int minimo(vector<int> const& lista, int i, int j) {

    if (i == j) {
        return lista[i];
    }
    else {

        int m = (i + j) / 2;

        if (lista[m] > lista[m + 1]) {
            return minimo(lista, m + 1, j);
        }
        else {
            return minimo(lista, i, m);
        }
    }
}

//Post: min u: 0 <= u < N: v[u]

bool resuelveCaso() {

    // Lectura de los datos

    int num;
    cin >> num;

    if (!cin) return false;

    vector<int> lista;

    int j;

    for (int i = 0; i < num; i++) {

```

```

        cin >> j;

        lista.push_back(j);
    }

    // Calculo del resultado: una funcion aparte
    int sol = minimo(lista, 0, lista.size() - 1);

    // Escritura del resultado
    cout << sol << endl;

    return true;
}

int main() {

    // Para la entrada por fichero.
#ifdef DOMJUDGE
    std::ifstream in("casos.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf());
#endif

    // Resolvemos
    while (resuelveCaso()) {}

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

    return 0;
}

```