

```

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

using namespace std;

// Pre :  $0 \leq a \leq b \leq \text{longitud}(v) - 1$ 
void particion(vector<int>& v, int a, int b, int& p, int& veces) {

    // Pre :  $0 \leq a \leq b \leq \text{longitud}(v) - 1$ 

    int i, j;
    int aux;
    i = a + 1;
    j = b;
    while (i <= j) {
        if ((v[i] > v[a]) && (v[j] < v[a])) {
            aux = v[i]; v[i] = v[j]; v[j] = aux; veces++;
            i = i + 1; j = j - 1;
        }
        else {
            if (v[i] <= v[a])
                i = i + 1;
            if (v[j] >= v[a])
                j = j - 1;
        }
    }
    p = j;
    aux = v[a]; v[a] = v[p]; v[p] = aux; veces++;
}

// Post :  $0 \leq a \leq p \leq b \leq \text{longitud}(v) - 1 \wedge$ 
//  $\forall x : a \leq x \leq p - 1 : v[x] \leq v[p] \wedge$ 
//  $\forall y : p + 1 \leq y \leq b : v[y] \geq v[p]$ 

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

// Pre :  $0 \leq a \leq \text{longitud}(v) \text{ \&\& } -1 \leq b \leq \text{longitud}(v) - 1 \text{ \&\& } a \leq b+1$ 
int quickSort(vector<int> v, int a, int b) {

    int p;
    int veces = 0;
    int sol1 = 0;
    int sol2 = 0;

```

```

    if (a <= b) {
        particion(v, a, b, p, veces);
        sol1 = quickSort(v, a, p - 1);
        sol2 = quickSort(v, p + 1, b);
    }

    veces += sol1 + sol2;

    return veces;
}
// Post:

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 = quickSort(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;
}
```