

```

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

using namespace std;

//Pre: 0 <= i <= j < lista.size()
bool esOrden(vector<int> const& lista, int i, int j, int& max, int& min) {

    if (i - j == 0) {

        max = lista[i]; min = lista[i];
        return true;
    }
    else if (i - j == 1) {

        if (lista[i] < lista[j]) {

            max = lista[j]; min = lista[i];
            return true;
        }
        else {

            min = lista[j]; max = lista[i];
            return false;
        }
    }
    else {

        int maxD, maxI, minD, minI;
        int m = (i + j) / 2;

        bool izq = esOrden(lista, i, m, maxI, minI);
        bool der = esOrden(lista, m + 1, j, maxD, minD);

        if (maxD > maxI) max = maxD;
        else max = maxI;

        if (minD < minI) min = minD;
        else min = minI;

        bool sol = (izq && der) && (maxD >= maxI) && (minI <= minD);

        return sol;
    }
}

//Post: sol = ((maximo(lista, i, i+j/2) <= maximo(lista, (i+j)/2 + 1, j)) &&
(minimo(lista, i, i+j/2) <= minimo(lista, (i+j)/2 + 1, j)) && esOrden(lista, i, i, i+j/2)
&& esOrden(lista, (i+j)/2 + 1, j))
//Donde:
//Maximo(v, i, j) = max w: i <= w <= j: v[w] && Minimo(v, i, j) = min u: i <= u <= j: v[u]

```

```

bool resuelveCaso() {

    // Lectura de los datos

    int num;
    cin >> num;

    if (num == 0) return false;

    vector<int> lista;

    while (num != 0) {

        lista.push_back(num);

        cin >> num;
    }

    int max, min;

    // Calculo del resultado: una funcion aparte
    bool sol = esOrden(lista, 0, lista.size() - 1, max, min);

    // Escritura del resultado
    if (sol) cout << "SI" << endl;
    else cout << "NO" << 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;
}

```