

```

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

using namespace std;

//Pre: 0 <= N <= 1000
int mejorP(vector<int> const& lista, int N) {

    //INV: max p, q: 0 <= p <= q < n && (forall u, j: p <= u < j <= q: v[u] >= v[j]): q -
p + 1
    //Cota: N - n

    if (N != 0) {
        int n = 1;

        int r = 1;

        int s = 1;

        while (n < N) {

            if (lista[n - 1] >= lista[n]) s++;
            else {
                r = max(r, s);
                s = 1;
            }

            n++;
        }

        r = max(r, s);

        return r;
    }
    else {
        return 0;
    }
}

//Post: max p, q: 0 <= p <= q < N && (forall u, j: p <= u < j <= q: v[u] >= v[j]): q - p +
1

void resuelveCaso() {

    // Lectura de los datos

    int num;
    cin >> num;

    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 = mejorP(lista, num);

    // Escritura del resultado

    cout << sol << endl;
}

int main() {

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

    // Resolvemos
    int numCasos;
    std::cin >> numCasos;
    for (int i = 0; i < numCasos; ++i)
        resuelveCaso();

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

    return 0;
}

```