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

using namespace std;
const unsigned int MAXP = 10000;

const int MIN_TEMP = -50;
const int MAX_TEMP = 60;

// Pre: 0 <= p < nNums
bool bienDiv(vector<int> nums, int nNums, int p) {

    //INVmaximo: 0 <= j <= p && max w: 0 <= w <= j: nums[w]
    //B: j <= p
    //Cota: j

    int j = 0;
    int mx = nums[0];

    while (j <= p) {

        mx = max(mx, nums[j]);
        j++;
    }

    //INVprinc: (sol = forall i: p < i < n: (nums[i] > mx)) && p < n <= nNums
    //B: n < nNums
    //Cota: n

    bool sol = true;

    int n = p + 1;

    while (n < nNums) {

        if (nums[n] <= mx) sol = false;

        n++;
    }

    return sol;
}
// Post: sol = forall i: p < i < nNums: (i > maximo(nums, p))
// Donde maximo = max w: 0 <= w <= p: w

void resuelveCaso() {

    // Lectura de los datos

    int nNums, p;
    cin >> nNums >> p;
```

```
    vector<int> nums(nNums);
    for (int i = 0; i < nNums; i++)
    {
        cin >> nums[i];
    }

    // Calculo del resultado: una funcion aparte
    bool sol = bienDiv(nums, nNums, p);

    // Escritura del resultado

    if (sol) cout << "SI" << endl;
    else cout << "NO" << endl;
}


int main() {

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

    unsigned int numCasos;
    cin >> numCasos;
    // Resolvemos
    while (numCasos--) {
        resuelveCaso();
    }

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

    return 0;
}
```