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

using namespace std;
const unsigned int MAXP = 100000;

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

//Pre: 2 <= nPers <= longitud(dalt)
pair<bool, bool> sonDalton(int dalt[], int nPers) {

    //INV: 0 < n <= nPers && (sol = P.t. i: 0 < i < n: (dalt[i - 1] < dalt[i]) || sol2 =
P.t. u: 0 < u < n: (dalt[u - 1] > dalt[u]))
    //B: n < nPers
    //Cota: n

    bool sol1 = true;
    bool sol2 = true;

    int n = 1;

    while (n < nPers && (sol1 || sol2)) {

        if (dalt[n - 1] >= dalt[n]) {
            sol1 = false;
        }

        if (dalt[n - 1] <= dalt[n]) {
            sol2 = false;
        }

        n++;
    }

    return {sol1, sol2};
}
// Post: sol1 = P.t. w: 0 < w < nPers: (dalt[w - 1] < dalt[w])  || sol2 = P.t. j: 0 < j <
nPers: (dalt[j - 1] > dalt[j])

bool resuelveCaso() {

    // Lectura de los datos

    int nPers;

    cin >> nPers;

    if (nPers == 0) return false;

    int dalt[MAXP];
    for (int i = 0; i < nPers; i++) {
        cin >> dalt[i];
    }

    pair<bool, bool> son = sonDalton(dalt, nPers);

    if (son.first || son.second) cout << "DALTON" << endl;
    else cout << "DESCONOCIDOS" << endl;

    return true;
}


int main() {

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

    // Resolvemos
    while (resuelveCaso()) {}

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

    return 0;
}
```