

```

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

using namespace std;

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

//Pre: 0 <= i <= j <= N - 1 && forall u: 0 <= u < longitud(lista) - 1: lista[u] < lista[u
+ 1]
int busqueda(vector<int> const& lista, int i, int j, int const& b) {

    int m = (i + j) / 2;

    if (lista[m] == (b + m)) {
        return m;
    }
    else if (i == j) {
        return -1;
    }
    else {
        if (lista[m] > (b + m)) {
            return busqueda(lista, i, m, b);
        }
        else {
            return busqueda(lista, m + 1, j, b);
        }
    }
}

//Post: {(forone u: 0 ≤ u < num: v[u] = b + u && sol = u) || (forall z: 0 ≤ z < num : b !=
v[z] && sol = -1)}

void resuelveCaso() {

    // Lectura de los datos

    int num, bingo;
    cin >> num >> bingo;

    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 = busqueda(lista, 0, lista.size() - 1, bingo);

    // Escritura del resultado
    if (sol != -1)
        cout << lista[sol] << endl;
    else
        cout << "NO" << 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;
}

```