

# Problema 3

Implementar un algoritmo que resuelva el siguiente problema:

```
method problema3 (a : array<int>, p : int) returns (b : bool)
requires a != null && a.Length > 0
requires 0 <= p < a.Length
ensures b == forall u, w :: 0 <= u <= p < w < a.Length ==> a[u] < a[w]
```

*Requisitos de implementación.*

- El orden de complejidad del algoritmo debe ser lineal respecto al número de elementos del vector.
- Debe utilizarse la clase `vector` para almacenar los datos y llamar a la función `resolver`

## Entrada

La entrada comienza con un valor entero que indica el número de casos de prueba. Cada caso de prueba consta de dos líneas. La primera contiene el número de elementos del vector y el valor de  $p$ . La segunda línea contiene los valores del vector.

## Salida

Para cada caso de prueba se escribe en una línea SI si el resultado de la función es cierto y NO si es falso.

## Entrada de ejemplo

```
4
15 5
5 7 3 4 7 3 8 9 8 10 8 9 8 9 10
4 0
4 7 5 6
4 3
4 3 6 2
5 2
5 3 4 6 4
```

## Salida de ejemplo

```
SI
SI
SI
NO
```

**Autor:** Isabel Pita.