

---

## Arbre general. Són els elements del nivell $n$ -èssim una seqüència estrictament creixent? X80854\_ca

---

Donada la classe *Arbre* que permet gestionar arbres generals usant memòria dinàmica, cal implementar el mètode

```
bool nivell_n_creixent (int nivell_cercat );
```

que determina si els elements del *nivell\_cercat*, llegits d'esquerra a dreta, són una seqüència d'enters estrictament creixent. Considerem que una seqüència buida és estrictament creixent.

Cal enviar a jutge.org la següent especificació de la classe *Arbre* i la implementació del mètode dins del mateix fitxer.

```
#include <cstdlib>
#include <iostream>
using namespace std;
typedef unsigned int nat;

template <typename T>
class Arbre {

public:
    // Construeix un Arbre format per un únic node que conté a x.
    Arbre(const T &x);

    // Tres grans.
    Arbre(const Arbre<T> &a);
    Arbre& operator=(const Arbre<T> &a);
    ~Arbre() throw();

    // Col·loca l'arbre donat com a darrer fill de l'arrel de l'arbre sobre el que s'aplica el
    // mètode i l'arbre a queda invalidat.
    void afegir_darrer_fill (Arbre<T> &a);

    static const int ArbreInvalid = 400;

    // Comprova si els valors dels elements de nivell i llegits d'esquerra a dreta
    // formen una seqüència estrictament creixent
    bool nivell_n_creixent (int nivell_cercat );

private:
    Arbre(): _arrel (NULL) {};
    struct node {
        T info;
        node* primf;
        node* seggerm;
    };
    node* _arrel ;
```

```
static node* copia_arbre (node* p);
static void destrueix_arbre (node* p) throw();
```

```
// Aquí va l'especificació dels mètodes privats addicionals
};
```

```
static const int minimum_int_ESIN = -2147483648;
```

```
// Aquí va la implementació del mètode nivell_n_creixent i dels mètodes privats addi-
cionals
```

Per testejar la solució, jutge.org ja té implementats la resta de mètodes de la classe *Arbre* i un programa principal que llegeix un arbre general, el nivell cercat i crida el mètode *nivell\_n\_creixent*.

## Entrada

L'entrada consisteix en la descripció d'un arbre general d'enters (el seu recorregut en preordre, en el qual al valor de cada node li segueix el seu nombre de fills). A continuació segueix un enter que representa el nivell cercat.

## Sortida

Una línia amb el text "NO és estrictament creixent al nivell X" si es dona el cas que els elements del nivell X, llegits d'esquerra a dreta, no són una seqüència d'enters estrictament creixent. La sortida serà, "SI és estrictament creixent al nivell X" altrament.

## Observació

Només cal enviar la classe requerida i la implementació del mètode *nivell\_n\_creixent*. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat.

### Exemple d'entrada 1

```
7 0
0
1
```

### Exemple de sortida 1

```
SI és estrictament creixent al nivell 0
SI és estrictament creixent al nivell 1
```

### Exemple d'entrada 2

```
7 2
 5 0
 3 0
0
1
2
```

### Exemple de sortida 2

```
SI és estrictament creixent al nivell 0
NO és estrictament creixent al nivell 1
SI és estrictament creixent al nivell 2
```

### Exemple d'entrada 3

```
-7 3
 8 0
 4 2
 3 1
```

```
0 1
 6 0
-5 0
2 4
 9 0
```