

---

## Arbre general. Són els elements del recorregut en preordre especular una seqüència estrictament creixent?

X10400\_ca

---

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

```
bool preordre_especular_creixent ();
```

que determina si els elements del recorregut en preordre especular de l'arbre (o sigui, un recorregut en preordre on els fills es visiten de dreta a esquerra) són una seqüència d'enters estrictament creixent. Considerem que una seqüència buida es estrictament creixent.

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

```
#include <iostream>
#include <cstdlib>
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; després de fer b.afegir\_fill(a), a no és un arbre vàlid.

```
void afegir_darrer_fill (Arbre<T> &a);
```

```
static const int ArbreInvalid = 400;
```

```
//Comprova si els elements del recorregut en preordre especular de l'arbre
//(o sigui, un recorregut en preordre on els fills es visiten de dreta a
//esquerra) són una seqüència d'enters estrictament creixent.
```

```
bool preordre_especular_creixent ();
```

**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
};

// Aquí va la implementació del mètode preordre_especular_creixent

```

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 i crida el mètode *preordre\_especular\_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).

### Sortida

Una línia amb el text "NO es estrictament creixent" si es dona el cas que els elements del recorregut en preordre especular de l'arbre (o sigui, un recorregut en preordre on els fills es visiten de dreta a esquerra) no són una seqüència d'enters estrictament creixent. La sortida serà, "SI es estrictament creixent" altrament.

### Observació

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

#### Exemple d'entrada sample-1

```

-7 3
 8 0
 4 2
  3 1
   0 1
    6 0
  -5 0
 2 4
  9 0
  1 0
  8 0
  5 0

```

#### Exemple d'entrada sample-2

```

-7 3
 38 0
 18 2
  26 1
   32 1
    36 0

```

#### Exemple de sortida sample-1

```

NO es estrictament creixent

```

```

21 0
 2 4
 12 0
 10 0
  8 0
  5 0

```

### Exemple de sortida sample-2

SI es estrictament creixent

### Exemple d'entrada sample-3

```
-7 3
 38 0
 18 2
 26 1
 32 1
 36 0
 21 0
 2 4
 10 0
 10 0
 8 0
 5 0
```

### Exemple d'entrada sample-4

```
7 0
```

### Exemple d'entrada sample-5

```
7 1
 8 0
```

### Exemple d'entrada sample-6

```
7 1
 7 0
```

## Informació del problema

Autor : Ignasi Gómez-Sebastià  
Generació : 2021-03-18 17:15:04

© *Jutge.org*, 2006–2021.  
<https://jutge.org>

### Exemple de sortida sample-3

NO es estrictament creixent

### Exemple de sortida sample-4

SI es estrictament creixent

### Exemple de sortida sample-5

SI es estrictament creixent

### Exemple de sortida sample-6

NO es estrictament creixent