
Arbre general és n -ari?

X68222_ca

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

```
bool nari(nat n) const;
```

que retorna *true* si l'arbre és n -ari (tots els nodes excepte les fulles són de grau n), *false* en cas contrari.

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

```
#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 primer 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_fill (Arbre<T> &a);
```

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

```
// Indica que si l'arbre és  $n$ -ari (tots els nodes excepte les fulles són de grau  $n$ )
```

```
bool nari(nat n) const;
```

```
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 nari
```

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 després crida varies vegades el mètode *nari*.

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 una seqüència d'enters que representen diferents valors per testejar *nari*.

Sortida

Una línia per cada element n de la seqüència d'enters d'entrada, amb el text " n -ari: NO" indicant que l'arbre no és n -ari o amb el valor " n -ari: SI" indicant que l'arbre és n -ari.

Observació

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

Exemple d'entrada 1

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

Exemple d'entrada 2

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

Exemple de sortida 1

```
0-ari: NO  
1-ari: NO  
2-ari: NO  
3-ari: SI
```

Exemple de sortida 2

```
0-ari: SI  
1-ari: SI  
2-ari: SI  
3-ari: SI
```

Exemple d'entrada 3

```
7 1
 8 0
0
1
2
3
```

Exemple de sortida 3

```
0-ari: NO
1-ari: SI
2-ari: NO
3-ari: NO
```

Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T17:42:41.600Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>