

---

## Elements del nivell $n$ -èssim d'un arbre general

X70366\_ca

---

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

```
void nivell (nat i) const;
```

que escriu una línia amb els elements del nivell  $i$ -èssim, d'esquerra a dreta. Cada element ha de sortir precedit d'un espai.

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;
```

```
    // Escriu una línia amb els elements del nivell i, d'esquerra a dreta.
```

```
    // Cada element ha de sortir precedit d'un espai.
```

```
    void nivell (nat i) 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 nivell

```

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 *nivell*.

## 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 nivells.

## Sortida

Una línia per cada element *i* de la seqüència d'enters d'entrada, amb els elements de l'arbre situats en el nivell *i*, d'esquerra a dreta. Cada element surt precedit d'un espai.

## Observació

Només cal enviar la classe requerida i la implementació del mètode *nivell*. Podeu ampliar la classe amb mètodes privats. Seguiu estrictament la definició de la classe de l'enunciat.

### Exemple d'entrada 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
3
1
5
0
2

```

### Exemple d'entrada 2

```

7 0
0
1

```

### Exemple d'entrada 3

```

7 1 8 0
1
2
0

```

### Exemple de sortida 1

```

0
8 4 2

-7
3 -5 9 1 8 5

```

### Exemple de sortida 2

```

7

```

### Exemple de sortida 3

```

8
7

```

## Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T19:19:14.713Z

© *Jutge.org*, 2006–2026.  
<https://jutge.org>