

---

## Arbre general. Elimina nodes de grau 0.

X83764\_ca

---

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

```
void elimina_grau0();
```

que elimina tots els nodes de grau 0 de l'arbre, o sigui elimina totes les fulles.

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

```
#include <cstdlib>
```

```
#include <string>
```

```
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);
```

```
// Imprimeix la informació dels nodes en preordre, cada element en una nova línia i
```

```
// precedit per espais segons el nivell on està situat.
```

```
void preordre() const;
```

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

```
// Elimina tots els nodes de grau 0 de l'arbre.
```

```
void elimina_grau0();
```

```
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();
static void preordre (node *p, string pre);

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

// Aquí va la implementació del mètode elimina_grau0

```

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 els mètodes *elimina\_grau0* i *preordre*.

## 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

El recorregut en preordre de l'arbre general resultant. Cada element en una nova línia i precedit per espais segons el nivell on està situat.

## Observació

Només cal enviar la classe requerida i la implementació del mètode *elimina\_grau0*. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat.

### Exemple d'entrada 1

```

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

```

### Exemple de sortida 1

```

-5
  9
    4
      7
        3
          0
            5
              6
                0
                  2

```

### Exemple d'entrada 2

7 0

### Exemple d'entrada 3

7 1  
8 0

### Exemple de sortida 2

### Exemple de sortida 3

7

### Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T16:45:09.859Z

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