

---

## Arbre general. Calcula arbre compta graus.

X84340\_ca

---

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

```
void arbre_compta_graus ();
```

que modifica el contingut dels nodes per tal de guardar a cada node el seu grau.

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;
```

```
// Modifica el contingut dels nodes per tal de guardar a cada node el seu grau
```

```
void arbre_compta_graus ();
```

**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 arbre_compta_graus

```

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 *arbre\_compta\_graus* 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 *arbre\_compta\_graus*. 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

```

2
 1
   1
    3
     0
     0
     0
  2
   1
    5
     1
     0
     0
     3
     0
     0
     0
     0
     0
     2
     0
     0
  0

```

### Exemple d'entrada 2

7 0

### Exemple d'entrada 3

7 1  
8 0

### Exemple de sortida 2

0

### Exemple de sortida 3

1  
0

### Informació del problema

Autor : Jordi Esteve

Generació : 2020-11-05 00:01:43

© *Jutge.org*, 2006–2020.

<https://jutge.org>