
Arbre general. Modifica amb màxim.

X48886_ca

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

```
void pare_maxim();
```

que modifica el contingut dels nodes per tal de guardar a cada node el valor màxim entre els valors dels nodes que son els seus fills. Els nodes fulla (els de grau 0) no es modifiquen.

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 valor màxim entre
```

```
// els valors dels nodes que son els seus fills. Els nodes fulla (els de grau 0) no es
```

```
// modifiquen.
```

```
void pare_maxim();
```

```
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 pare_maxim

```

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 d'enters i després crida els mètodes *pare_maxim* 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 *pare_maxim*. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat.

Exemple d'entrada sample-1

```

6 2
  6 1
    6 1
      6 3
        1 0
        20 0
        -8 0
  6 2
    6 1
      6 5
        6 1
          -3 0
        12 0
        6 3
          7 0
          19 0
          4 0
        6 0
        6 2
          1 0

```

```

7 0
6 0

```

Exemple de sortida sample-1

```
20
 20
  20
   20
    1
   20
  -8
19
 19
 19
```

```
-3
-3
12
19
 7
19
 4
 6
 7
 1
 7
```

6

Exemple d'entrada sample-2

```
7 0
```

Exemple de sortida sample-2

```
7
```

Exemple d'entrada sample-3

```
7 1
 8 0
```

Exemple de sortida sample-3

```
8
 8
```

Informació del problema

Autoria: Ignasi Gómez-Sebastià

Generació: 2026-01-25T16:33:25.087Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>