Arbre general. El pare es sempre més gran? X11926_ca

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

```
bool es_pare_gran ();
```

que determina si el valor de cada node es més gran que els valors dels nodes que són els seus fills. Els nodes fulla (els de grau 0) no es comproven.
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 preodre, cada element en una nova línia i
  // precedit per espais segons el nivell on està situat.
  void preordre () const;

  static const int ArbreInvalid = 400;

 //determina si el valor de cada node es més gran que els valors dels nodes que son els
 //seus fills. Els nodes fulla (els de grau 0) no es comproven.
  bool es_pare_gran ();

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 es_pare_gran
```

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 mètode *es_pare_gran*.

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

Una línia amb el text "NO es pare mes gran" indicant que el valor d'algun node no es més gran que el dels seus fills o "SI es pare mes gran" altrament.

Observació

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

Exemple d'entrada sample-1

```
106 2
  26 1
    16 1
      6 3
        1 0
        2 0
        -8 0
  36 2
    26 1
      16 5
        6 1
          -3 0
        -12 0
        6 3
          2 0
          -19 0
          4 0
        6 0
        6 2
          1 0
        -7 0
  6 0
```

Exemple de sortida sample-1

```
SI es pare mes gran
```

Exemple d'entrada sample-2

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

Exemple de sortida sample-2

```
NO es pare mes gran
```

Exemple d'entrada sample-3

```
7 0
```

Exemple de sortida sample-3

```
SI es pare mes gran
```

Exemple d'entrada sample-4

```
7 1
  8 0
```

Exemple de sortida sample-4

```
NO es pare mes gran
```

Exemple d'entrada sample-5

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

Exemple de sortida sample-5

```
SI es pare mes gran
```

Informació del problema

Autor : Ignasi Gómez-Sebastià
Generació : 2021-03-18 16:45:41