
Arbre binari. Calcula l'arbre suma de dos arbres**Z96088_ca**

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

```
void suma_2_arbres(const Abin<T> &a);
```

que modifica el contingut de l'arbre del p.i. per tal de guardar un arbre que sigui la suma dels dos arbres donats, l'arbre del p.i. i l'arbre a. La suma de dos arbres binaris és un arbre binari on cada node conté la suma dels nodes de la mateixa posició dels dos arbres a sumar; si un dels nodes a sumar és buit es considera que el seu valor és zero.

Cal enviar a jutge.org la següent especificació de la classe *Abin* i la implementació del mètode dins del mateix fitxer. Indica dins d'un comentari a la capçalera del mètode el seu cost en funció del nombre d'elements *n* de l'arbre.

```
#include <cstdlib>
```

```
#include <iostream>
```

```
using namespace std;
```

```
template <typename T>
```

```
class Abin {
```

```
    public:
```

```
        Abin(): _arrel (NULL) {};
```

```
        // Pre: cert
```

```
        // Post: El resultat és un arbre sense cap element.
```

```
        Abin(Abin<T> &ae, const T &x, Abin<T> &ad);
```

```
        // Pre: cert
```

```
        // Post: El resultat és un arbre amb un element i dos subarbres.
```

```
        // Destructora
```

```
        ~Abin();
```

```
        // Assignació
```

```
        Abin<T>& operator=(const Abin<T>& a);
```

```
        // No està disponible el constructor per còpia
```

```
        Abin(const Abin<T> &a) = delete;
```

```
        // operador « d'escriptura
```

```
template <class U> friend std::ostream& operator<<(std::ostream&, const Abin<U> &a);
```

```
        // operador » de lectura
```

```
template <class U> friend std::istream& operator>>(std::istream&, Abin<U> &a);
```

```
void suma_2_arbres(const Abin<T> &a);
```

```
    // Pre: A1 = arbre del p.i, A2 = a
```

```
    // Post: Modifica l'arbre del p.i. per tal de guardar un arbre que sigui
```

```
    // la suma dels arbres A1 i A2. L'arbre a no es modifica.
```

```

private:
    struct node {
        node* f_esq;
        node* f_dret;
        T info;
    };
    node* _arrel;
    static void esborra_nodes(node* m);
    // Pre: m apunta a un node d'un arbre binari o a nullptr si és buit.
    // Post: Esborra tots els nodes de l'arbre apuntat per m.
    static node* copia_nodes(node* m);
    // Pre: m apunta a un node d'un arbre binari o a nullptr si és buit.
    // Post: Retorna un punter que apunta a l'arrel d'un arbre còpia de l'apuntat per m.
    static void print_nodes(node* m, ostream &os, string d1);
    // Pre: m apunta a un node d'un arbre binari o a nullptr si és buit.
    // Post: Imprimeix per canal de sortida os el contingut de l'arbre apuntat per m afegint
    a cada línia el prefix d1.

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

// Aquí va la implementació del mètode públic i dels mètodes privats addicionals

```

Per testejar la solució, jutge.org ja té implementats la resta de mètodes de la classe *Abin* i un programa principal que llegeix dos arbres binaris i després crida el mètode *suma_2_arbres*.

Entrada

L'entrada consisteix en la descripció de dos arbres binaris d'enters (per cada arbre es proporciona el seu recorregut en preordre, en el qual inclou les fulles marcades amb un -1). Per exemple, l'arbre (mira el PDF de l'enunciat) es descriuria amb

```
3 0 7 -1 4 -1 -1 2 -1 -1 5 4 -1 -1 7 6 -1 1 -1 -1 -1
```

Sortida

El contingut dels dos arbres binaris abans i després de cridar el mètode *suma_2_arbres*.

Observació

Només cal enviar la classe requerida i la implementació del mètode *suma_2_arbres* amb el seu cost en funció del nombre d'elements *n* de l'arbre. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat.

Exemple d'entrada 1

```
-1
-1
```

Exemple de sortida 1

```
.
.
```

•

Exemple de sortida 2

[3]

__.

__.

.

[3]

__.

__.

•

Exemple de sortida 3

•

[3]

__.

__.

[3]

_.

_.

[3]

__.

__.

Exemple de sortida 4

$$\begin{array}{l} [3] \\ \backslash _ . \\ \backslash _ [2] \\ \quad \backslash _ . \\ \quad \backslash _ . \end{array}$$

[3]
 __.
 __.

$$\begin{array}{l} [6] \\ \backslash _ \\ \backslash _ [2] \\ \quad \backslash _ \\ \quad \backslash _ \end{array}$$

[3]

_.

_.

Exemple de sortida 5

[3]
 __.
 __.

```

[3]
 \_.
 \__[2]
      \_.
      \_.

[6]
 \_.
 \__[2]

```

Exemple d’entrada 6

```

3 2 -1 -1 -1
3 2 -1 -1 -4 -1 -1

```

Exemple d’entrada 7

```

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

```

```

 \_.
 \_.

[3]
 \_.
 \__[2]
      \_.
      \_.

```

Exemple de sortida 6

```

[3]
 \_.
 \__[2]
      \_.
      \_.

[3]
 \__[-4]
 |   \_.
 |   \_.
 \__[2]
      \_.
      \_.

[6]
 \__[-4]
 |   \_.
 |   \_.
 \__[4]
      \_.
      \_.

[3]
 \__[-4]
 |   \_.
 |   \_.
 \__[2]
      \_.
      \_.

```

Exemple de sortida 7

```

[-3]
 \__[-4]
 |   \_.
 |   \_.
 \__[-2]
      \_.
      \_.

[7]
 \__[8]
 |   \__[4]
 |   |   \__[3]
 |   |   |   \_.
 |   |   |   \_.
 |   |   \__[6]
 |   |   \_.
 |   |   \_.

```

```

|   \__[9]
|   \___.
|   \___.
\__[5]
   \___.
   \___.

```

```

[4]
\__[4]
|   \__[4]
|   |   \__[3]
|   |   |   \___.
|   |   |   \___.
|   |   |   \__[6]
|   |   |   \___.
|   |   |   \___.
|   |   \__[9]
|   |   \___.
|   |   \___.

```

```

\__[3]
   \___.
   \___.

[7]
\__[8]
|   \__[4]
|   |   \__[3]
|   |   |   \___.
|   |   |   \___.
|   |   |   \__[6]
|   |   |   \___.
|   |   |   \___.
|   |   \__[9]
|   |   \___.
|   |   \___.
\__[5]
   \___.
   \___.

```

Exemple d'entrada 8

```

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

```

Exemple de sortida 8

```

[7]
1\_-1[8] -1
|   \__[4]
|   |   \__[3]
|   |   |   \___.
|   |   |   \___.
|   |   |   \__[6]
|   |   |   \___.
|   |   |   \___.
|   |   \__[9]
|   |   \___.
|   |   \___.
\__[5]
   \___.
   \___.

[3]
\__[5]
|   \__[7]
|   |   \___.
|   |   \__[6]
|   |   |   \__[1]
|   |   |   |   \___.
|   |   |   |   \___.
|   |   |   \___.
|   |   \__[4]
|   |   \___.
|   |   \___.
\__[0]
   \__[2]
   |   \___.
   |   \___.
   \__[7]
       \__[4]
       |   \___.
       |   \___.
       \___.

```

[10]

```

\__[13]
|  \__[11]
|  |  \__[3]
|  |  |  \__.
|  |  |  \__.
|  |  \__[12]
|  |  \__[1]
|  |  |  \__.
|  |  |  \__.
|  |  \__.
|  \__[13]
|  \__.
|  \__.
\__[5]
  \__[2]
  |  \__.
  |  \__.
  \__[7]
    \__[4]
    |  \__.
    |  \__.
    \__.

```

```

[3]
  \__[5]
  |  \__[7]
  |  |  \__.
  |  |  \__[6]
  |  |  \__[1]
  |  |  |  \__.
  |  |  |  \__.
  |  |  \__.
  |  \__[4]
  |  \__.
  |  \__.
\__[0]
  \__[2]
  |  \__.
  |  \__.
  \__[7]
    \__[4]
    |  \__.
    |  \__.
    \__.

```

Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T20:03:49.892Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>

