
Fusió dels elements de dues piles ordenades**X81447_ca**

Donada la classe *pila* que permet apilar elements en una estructura simplement encadenada en memòria dinàmica, cal implementar el mètode

```
void fusiona(const pila<T> &p2);  
// Pre: Les piles del p.i. i p2 estan ordenades de menor a major  
// Post: Al p.i. se li han afegit els elements de p2 ordenadament. p2 no es modifica
```

que, a partir de dues piles ordenades de menor a major, fusiona els elements de les dues ordenadament deixant el resultat al paràmetre implícit, sense modificar la pila *p2*. Pots veure exemples en els jocs de prova públics.

Cal enviar a jutge.org la següent especificació de la classe *pila* i la implementació del mètode dins del mateix fitxer. La resta de mètodes públics i privats ja estan implementats. Indica dins d'un comentari a la capçalera del mètode el seu cost en funció del nombre d'elements *n1* de la pila del p.i. i nombre d'elements *n2* de la pila *p2*.

```
#include <iostream>  
#include <vector>  
using namespace std;  
typedef unsigned int nat;
```

```
template <typename T>
```

```
class pila { // pila en memòria dinàmica
```

```
    public:
```

```
        pila ();  
        // Crea pila buida
```

```
        ~pila ();  
        // Destrueix el p.i.
```

```
        pila(const vector<int> &v);  
        // Crea pila amb els elements de v amb el mateix ordre.
```

```
        nat longitud() const;  
        // Retorna el nombre d'elements del p.i.
```

```
        void mostra() const;  
        // Mostra el p.i. pel canal estàndard de sortida.
```

```
        void fusiona(const pila<T> &p2);  
        // Pre: Les piles del p.i. i p2 estan ordenades de menor a major  
        // Post: Al p.i. se li han afegit els elements de p2 ordenadament. p2 no es modifica
```

```
    private:
```

```
        struct node {  
            T info;  
            node* seg;
```

```

};
node* _cim; // Apunta al cim de la pila
nat _mida; // Nombre d'elements de la pila

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

// Aquí va la implementació del mètode públic fusiona i privats addicionals

```

Degut a que jutge.org només permet l'enviament d'un fitxer amb la solució del problema, en el mateix fitxer hi ha d'haver l'especificació de la classe i la implementació del mètode *fusiona* (el que normalment estarien separats en els fitxers *.hpp* i *.cpp*).

Per testejar la solució, jutge.org ja té implementats la resta de mètodes de la classe *pila* i un programa principal que llegeix dues piles, després crida el mètode *fusiona* i finalment mostra el contingut de les dues piles.

Entrada

L'entrada conté dues línies formades per seqüències d'enters ordenades, són els elements que tindran les dues piles inicials.

Sortida

Es mostra el contingut de les dues piles després de fer la fusió. Per cada pila s'escriu el nombre d'elements de la pila seguit d'un espai i dels elements de la pila entre claudàtors i separats per espais.

Observació

Només cal enviar l'especificació de la classe *pila*, la implementació del mètode *fusiona* i el seu cost en funció del nombre d'elements n_1 i n_2 de les dues piles inicials. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat. No es poden usar estructures de dades auxiliars com per exemple vectors.

Exemple d'entrada 1

```

2 3 4 5 8
1 6 9

```

Exemple de sortida 1

```

8 [1 2 3 4 5 6 8 9]
3 [1 6 9]

```

Exemple d'entrada 2

```

-6 -2 0 3 4 8
-5

```

Exemple de sortida 2

```

7 [-6 -5 -2 0 3 4 8]
1 [-5]

```

Exemple d'entrada 3

```

-5
-6 -2 0 3 4 8

```

Exemple de sortida 3

```

7 [-6 -5 -2 0 3 4 8]
6 [-6 -2 0 3 4 8]

```

Exemple d'entrada 4

```

-6 -2 0 3 4 8

```

Exemple de sortida 4

```

6 [-6 -2 0 3 4 8]
0 []

```

Exemple d'entrada 5

-6 -2 0 3 4 8

Exemple d'entrada 6

Exemple d'entrada 7

1 2 3 4 5 8 9
1 4 6 9

Exemple d'entrada 8

-6 -2 0 3 4 8
-7

Exemple d'entrada 9

-6 -2 0 3 4 8
9

Exemple d'entrada 10

-7
-6 -2 0 3 4 8

Exemple d'entrada 11

9
-6 -2 0 3 4 8

Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T16:37:34.650Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>

Exemple de sortida 5

6 [-6 -2 0 3 4 8]
6 [-6 -2 0 3 4 8]

Exemple de sortida 6

0 []
0 []

Exemple de sortida 7

11 [1 1 2 3 4 4 5 6 8 9 9]
4 [1 4 6 9]

Exemple de sortida 8

7 [-7 -6 -2 0 3 4 8]
1 [-7]

Exemple de sortida 9

7 [-6 -2 0 3 4 8 9]
1 [9]

Exemple de sortida 10

7 [-7 -6 -2 0 3 4 8]
6 [-6 -2 0 3 4 8]

Exemple de sortida 11

7 [-6 -2 0 3 4 8 9]
6 [-6 -2 0 3 4 8]