
Fusió dels elements de dues cues ordenades no circulars X31110_ca

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

```
void fusiona(const cua<T> &c2);  
// Pre: Les cues del p.i. i c2 estan ordenades de menor a major  
// Post: Al p.i. se li han afegit els elements de c2 ordenadament. c2 no es modifica
```

que, a partir de dues cues ordenades de menor a major, fusiona els elements de les dues ordenadament deixant el resultat al paràmetre implícit, sense modificar la cua c2. Pots veure exemples en els jocs de prova públics.

Cal enviar a jutge.org la següent especificació de la classe *cua* 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 cua del p.i. i nombre d'elements *n2* de la cua c2.

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

```
template <typename T>  
class cua { // Cua no circular en memòria dinàmica  
  public:  
    cua();  
    // Construeix una cua buida.  
  
    ~cua();  
    // Destruïx el p.i.  
  
    cua(const vector<int> &v);  
    // Crea cua 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 cua<T> &c2);  
    // Pre: Les cues del p.i. i c2 estan ordenades de menor a major  
    // Post: Al p.i. se li han afegit els elements de c2 ordenadament. c2 no es modifica  
  
  private:  
    struct node {  
      T info;  
      node* seg;  
    };
```

```

node* _pri; // Apunta al primer element de la cua
node* _ult; // Apunta al darrer element de la cua
nat _mida;

// 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 *cua* i un programa principal que llegeix dues cues, després crida el mètode *fusiona* i finalment mostra el contingut de les dues cues.

Entrada

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

Sortida

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

Observació

Només cal enviar l'especificació de la classe *cua*, la implementació del mètode *fusiona* i el seu cost en funció del nombre d'elements n_1 i n_2 de les dues cues 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

Autor : Jordi Esteve

Generació : 2022-02-28 19:59:24

© *Jutge.org*, 2006–2022.

<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]