
Creació d'una llista doblement encadenada, circular i amb fantasma a partir d'un vector X32967_ca

Donada la classe *Llista* que permet guardar seqüències d'enters amb una llista doblement encadenada, circular i amb fantasma, cal implementar el mètode constructor

```
Llista (const vector <int> &v)
```

que crea una llista amb els elements del vector *v* amb el mateix ordre.

Cal enviar a jutge.org només la implementació del mètode constructor. La classe *Llista* té la següent especificació:

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

class Llista {
    // Llista doblement encadenada, circular i amb fantasma.
private:
    struct node {
        int info; // Informació del node
        node *seg; // Punter al següent element
        node *ant; // Punter a l'anterior element
    };
    node *_prim; // Punter a l'element fantasma
    nat _long; // Nombre d'elements

public:
    Llista ();
    // Pre: True
    // Post: El p.i. és una llista buida.

    Llista (const vector <int> &v);
    // Pre: True
    // Post: El p.i. conté els elements de v amb el mateix ordre.

    ~Llista ();
    // Post: Destruïx els elements del p.i.

    nat longitud () const;
    // Pre: True
    // Post: Retorna el nombre d'elements del p.i.

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

```
void mostra_invertida () const;
// Pre: True
// Post: Mostra el p.i. en ordre invers pel canal estàndard de sortida.
};
```

Per testejar la solució, jutge.org ja té implementats la resta de mètodes de la classe *Llista* i un programa principal que processa línies d'enters amb els que crear vectors i posteriorment llistes.

Entrada

L'entrada conté diverses línies formades per seqüències d'enters. Cadascuna d'elles són els elements que tindrà el vector amb el que es construirà cada llista.

Sortida

Per a cada línia d'entrada, escriu una línia amb el resultat: El nombre d'elements de la llista seguit d'un espai, els elements de la llista entre claudàtors i separats per espais i finalment aquests mateixos elements però amb ordre invers, també entre claudàtors i separats per espais.

Observació

Cal enviar la solució (el fitxer *solution.cpp*) comprimida en un fitxer *.tar*:

```
tar cvf solution.tar solution.cpp
```

Només cal enviar la implementació del mètode constructor a partir de vector. Seguiu estrictament la definició de la classe de l'enunciat.

Exemple d'entrada

```
3 -6 8 0 4 -2
5

9 7
```

Exemple de sortida

```
6 [3 -6 8 0 4 -2] [-2 4 0 8 -6 3]
1 [5] [5]
0 [] []
2 [9 7] [7 9]
```

Informació del problema

Autor : Jordi Esteve

Generació : 2020-10-08 23:47:44

© Jutge.org, 2006–2020.

<https://jutge.org>