

---

## Duplicació dels parells i eliminació dels senars d'una llista doblement encadenada X50741\_ca

---

Donada la classe *Llista* que permet guardar seqüències d'enters amb una llista doblement encadenada, sense fantasma i no circular, cal implementar el mètode

```
void duplica_parells_elimina_senars ()
```

que duplica els elements parells i elimina els elements senars del paràmetre implícit.

Cal enviar a jutge.org només la implementació del mètode *duplica\_parells\_elimina\_senars*.

La classe *Llista* té la següent especificació:

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

class Llista {
// Llista doblement encadenada, sense fantasma i no circular.
private:
    struct node {
        int info; // Informació del node
        node *seg; // Punter al següent element
        node *ant; // Punter a l'anterior element
    };
    node *_prim; // Punter al primer element
    node *_ult; // Punter a l'últim element
    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.

void duplica_parells_elimina_senars ();
// Pre: True
// Post: S'han duplicat els elements parells i
// s'han eliminat els elements senars del p.i.
// Exemple: [2 3 0 5 -2] => [2 2 0 0 -2 -2]
};

```

Per testejar la solució, [jutge.org](http://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 crea llistes i després crida el mètode *duplica\_parells\_elimina\_senars*.

## Entrada

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

## Sortida

Per a cada línia d'entrada, escriu una línia amb el resultat després d'haver duplicat els elements parells i eliminat els elements senars de la llista: 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 *duplica\_parells\_elimina\_senars*. Segueix estrictament la definició de la classe de l'enunciat.

### Exemple d'entrada 1

```
2 3 0 5 -2
```

### Exemple d'entrada 2

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

### Exemple d'entrada 3

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

### Exemple de sortida 1

```
6 [2 2 0 0 -2 -2] [-2 -2 0 0 2 2]
```

### Exemple de sortida 2

```
12 [2 2 -6 -6 4 4 0 0 8 8 2 2] [2 2 8 8 0 0 4 4 -6 -6 2 2]
10 [2 2 -6 -6 0 0 8 8 2 2] [2 2 8 8 0 0 -6 -6 2 2]
```

### Exemple de sortida 3

```
10 [6 6 8 8 0 0 4 4 2 2] [2 2 4 4 0 0 8 8 6 6]
6 [8 8 0 0 4 4] [4 4 0 0 8 8]
0 [] []
```

#### Exemple d'entrada 4

#### Exemple d'entrada 5

```
0
3
-3
2
```

#### Exemple d'entrada 6

```
0 0
-3 -5
8 3
3 8
```

#### Exemple de sortida 4

```
0 [] []
```

#### Exemple de sortida 5

```
2 [0 0] [0 0]
0 [] []
0 [] []
2 [2 2] [2 2]
```

#### Exemple de sortida 6

```
4 [0 0 0 0] [0 0 0 0]
0 [] []
2 [8 8] [8 8]
2 [8 8] [8 8]
```

### Informació del problema

Autor : Jordi Esteve

Generació : 2021-10-12 20:58:42

© *Jutge.org*, 2006–2021.

<https://jutge.org>