
Esborrar primer parell de llista dinàmica

X83920_ca

PRELIMINARS

En aquest exercici considerem la classe Llista que permet guardar seqüències ordenades d'enters positius amb una llista doblement encadenada, sense fantasma, no circular i amb punt d'interès.

Els nodes de la classe Llista estan doblement encadenats amb punters al següent (seg) i a l'anterior (ant). Una llista té quatre atributs: la longitud i tres punters a nodes, un pel primer element (primer_node), un per l'últim (ultim_node) i un altre per l'element actual (act), on tenim situat el punt d'interès de la llista.

EXERCICI

Cal implementar un mètode que elimina de la llista el primer valor parell que hi hagi, si n'hi ha algun, i manté el punt d'interès, excepte si el punt d'interès era l'element esborrat, cas en el qual deixarà d'haver-hi punt d'interès. Aquesta és la capcelera:

```
void esborrar_primer_parell();
/* Pre: El p.i. es una llista L */
/* Post: El p.i. s'ha modificat eliminant el seu primer valor parell.
        Si no hi havia cap parell a la llista, llavors no s'ha esborrat res.
        El punt d'interès no canvia excepte si l'element esborrat era el
        punt d'interès, cas en el qual el punt d'interès passa a ser NULL */
```

Aquí hi ha un exemple de llista abans i després d'executar el seu mètode esborrar_primer_parell.

```
1 5 2 (8) 3 6
=>
1 5 (8) 3 6
```

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: Makefile, program.cpp, llista.hpp, llista.cpp. Us falta crear el fitxer llista_esborra_primer_parell.cpp i implementar-hi la funció anterior. Quan pugeu la vostra solució al jutge, només cal que pugeu un tar construït així:

```
tar cf solution.tar llista_esborra_primer_parell.cpp
```

Entrada

L'entrada té un nombre arbitrari de casos. Cada cas consisteix en una llista amb punt d'interès: el nombre de vegades que cal avançar el punt d'interès respecte el primer element, el nombre d'enters positius de la llista i els enters positius que la formen. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquestes entrades. Només cal que implementeu el mètode abans esmentat.

Sortida

Per a cada cas, es mostrarà la llista original i la llista modificada (per l'acció d'esborrar el primer parell) dos cops, recorrent-la del primer a l'últim i recorrent-la de l'últim al primer. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquestes sortides. Només cal que implementeu la funció abans esmentada.

Exemple d'entrada 1

```
0 3
5 7 3
8 8
5 1 7 1 5 8 8 5
9 10
5 9 5 9 7 8 9 3 9 7
0 4
7 3 2 5
3 6
1 9 3 3 7 9
2 2
9 9
1 8
5 2 7 8 9 5 2 8
1 9
9 1 9 5 5 5 2 4 6
3 8
9 1 3 5 3 9 5 4
3 8
3 9 2 5 7 1 2 1
```

Exemple de sortida 1

```
[ (5) , 7, 3 ] >

[ (5) , 7, 3 ] >
[ 3, 7, (5) ] <

[ 5, 1, 7, 1, 5, 8, 8, 5 ] >

[ 5, 1, 7, 1, 5, 8, 5 ] >
[ 5, 8, 5, 1, 7, 1, 5 ] <

[ 5, 9, 5, 9, 7, 8, 9, 3, 9, (7) ] >

[ 5, 9, 5, 9, 7, 9, 3, 9, (7) ] >
[ (7) , 9, 3, 9, 7, 9, 5, 9, 5 ] <

[ (7) , 3, 2, 5 ] >

[ (7) , 3, 5 ] >
[ 5, 3, (7) ] <

[ 1, 9, 3, (3) , 7, 9 ] >

[ 1, 9, 3, (3) , 7, 9 ] >
[ 9, 7, (3) , 3, 9, 1 ] <

[ 9, 9 ] >

[ 9, 9 ] >
[ 9, 9 ] <

[ 5, (2) , 7, 8, 9, 5, 2, 8 ] >

[ 5, 7, 8, 9, 5, 2, 8 ] >
[ 8, 2, 5, 9, 8, 7, 5 ] <

[ 9, (1) , 9, 5, 5, 5, 2, 4, 6 ] >

[ 9, (1) , 9, 5, 5, 5, 4, 6 ] >
[ 6, 4, 5, 5, 5, 9, (1) , 9 ] <

[ 9, 1, 3, (5) , 3, 9, 5, 4 ] >

[ 9, 1, 3, (5) , 3, 9, 5 ] >
[ 5, 9, 3, (5) , 3, 1, 9 ] <

[ 3, 9, 2, (5) , 7, 1, 2, 1 ] >

[ 3, 9, (5) , 7, 1, 2, 1 ] >
[ 1, 2, 1, 7, (5) , 9, 3 ] <
```

Informació del problema

Autoria: PRO1

Generació: 2026-01-25T21:30:19.172Z

© *Jutge.org*, 2006–2026.

<https://judge.org>