
Copiar subseqüència posicions parells

X18717_ca

Preliminars

En aquest exercici treballarem sobre la següent estructura de dades, que ens serveix per a mantenir una seqüència de valors dins de ítems encadenats mitjançant punters.

```
struct Item {
    int value;
    Item* next;
};
```

Exercici

Implementeu una funció **RECURSIVA** que, donat un `Item*` que apunta a una seqüència d'ítems encadenats, retorna un altre `Item*` que apunta a una nova seqüència que és una còpia de la subseqüència d'elements a posició parell en la original. En altres paraules, la nova seqüència d'ítems no comparteix memòria amb la original, però la seva corresponent seqüència de valors és la subseqüència de valors a posició parell original. Sobre-entenem que el primer ítem accessible està a posició 0, el seu `next` a posició 1, el següent `next` a posició 2, i així successivament.

```
// Pre:  pitem apunta al primer element d'una seqüència correcta d'ítems encade
//       L'últim element de la seqüència apunta a NULL. El propi pitem podria s
//       cas en el qual no hi hauria elements a la seqüència.
// Post: Retorna un Item* que representa una seqüència d'ítems nous tals que la
//       corresponent seqüència de valors és una còpia de la subseqüència de va
//       a posició parell de la seqüència original.
//       La seqüència de valors original no ha canviat.
Item* copyEvenPos(Item* pitem);
```

Aquí tenim un exemple de paràmetres entrada i sortida de la funció:

```
copy([3]->[2]->[5]->[1]->[8]->NULL) = [3]->[5]->[8]->NULL
```

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

```
tar cf solution.tar copyEvenPos.cpp
```

Entrada

L'entrada té un nombre arbitrari de casos. Cada cas consisteix en una línia amb una llista de valors enters. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquestes entrades. Només cal que implementeu la funció abans esmentada.

Sortida

Per a cada cas, la sortida conté dues línies, la primera amb la mateixa llista de valors original, i la segona amb la llista còpia de valors a posició parell resultant. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquestes dades. Només cal que implementeu la funció abans esmentada.

Exemple d'entrada 1

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

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

Exemple de sortida 1

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

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

Informació del problema

Autoria: PRO1

Generació: 2026-01-25T21:02:03.514Z

© Jutge.org, 2006–2026.

<https://jutge.org>