Sumar valors en items a posició parell X37308_ca

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

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

Exercici
Implementeu una funció **RECURSIVA** que, donat un `Item*` que apunta a una seqüència d'items encadenats, retorna la suma dels valors dels items a posició parell. Sobre-entenem que el primer item 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'items 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 la suma dels valors guardats en els items a posició parell en
int sumOfValuesEvenPosition(Item *pitem);
```

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

```
sumOfValues([3]->[2]->[5]->[1]->[8]->NULL) = 16
```

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `Makefile`, `program.cpp`, `sumOfValuesEvenPosition.hpp`. Us falta crear el fitxer `sumOfValuesEvenPosition.cpp` amb els corresponents `includes` 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 sumOfValuesEvenPosition.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 una línia amb la corresponent suma dels valors de la llista a posició parell. 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

```
16
11
26
12
7
26
4
23
6
12
3
0
20
12
27
17
5
7
27
8
```

Informació del problema

Autoria: PRO1

Generació: 2026-01-25T21:10:16.362Z