
Afegir un element al final d'una llista d'items

X34562_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 rep dos paràmetres: un `Item*` que apunta a una seqüència no buida d'items encadenats, i un valor enter. La funció modifica la seqüència original afegint un nou item al final que tindrà com a valor el valor que s'ha rebut com a paràmetre.

```
// Pre: pitem apunta al primer element d'una seqüència correcta no buida d'ite
//      L'últim element de la seqüència apunta a NULL.
// Post: La funció afegeix un nou item al
//      final de la seqüència amb valor el value que li passen com a paràmetre
void push_back(Item* pitem, int value);
```

Aquí tenim un exemple de l'efecte de la funció:

```
{ pitem = [3]->[2]->[5]->[1]->[8]->NULL }
push_back(pitem, 4)
{ pitem = [3]->[2]->[5]->[1]->[8]->[4]->NULL }
```

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `Makefile`, `program.cpp`, `push_back.hpp`. Us falta crear el fitxer `push_back.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 push_back.cpp
```

Entrada

L'entrada té un nombre arbitrari de casos. Cada cas consisteix en dues línies: la primera amb una llista no buida de valors enters, i la segona amb un valor enter. 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 llista 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

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

Exemple de sortida

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

Informació del problema

Autor : PRO1

Generació : 2022-10-25 20:56:23

© *Jutge.org*, 2006–2022.

<https://jutge.org>