
Inserir sumes de prefixos en una llista (Pro2)

X80181_ca

Heu d'implementar una funció que rep una llista d'enters $[x_0, x_1, x_2, \dots, x_{n-2}, x_{n-1}]$ com a paràmetre per referència. La funció haurà d'inserir, després de cada element, la suma de la llista des del principi fins aquell element, és a dir, la funció retorna la llista:

$$[x_0, x_0, x_1, x_0 + x_1, x_2, x_0 + x_1 + x_2, \dots, x_{n-2}, x_0 + \dots + x_{n-2}, x_{n-1}, x_0 + \dots + x_{n-1}]$$

Important: Heu de garantir que els elements que la llista contenia inicialment queden inalterats i ocupant les posicions parells (indexant des de 0). En particular, la funció no els pot eliminar i tornar a afegir després.

Aquesta és la capçalera:

```
// Pre: Sigui [x0,x1,x2,...,x{n-1}] el valor inicial de l.
// Post: El valor de l és [x0, x0, x1, x0+x1, x2, x0+x1+x2, ..., x{n-1}, x0+...
//      A més a més, els elements inicials de la llista han persistit i
//      no han canviat de valor, i ocupen les posicions parells (indexant des
void insertSumsPrefixes(list<int> &l);
```

Aquí tenim un exemple de comportament de la funció:

```
insertSumsPrefixes(L = [2,3,1]) => L = [2,2,3,5,1,6]
```

Fixeu-vos que l'enunciat d'aquest exercici ja ofereix uns fitxers que haureu d'utilitzar per a compilar: `main.cc`, `insertSumsPrefixes.hh`. Us falta crear el fitxer `insertSumsPrefixes.cc` amb els corresponents `includes` i implementar-hi la funció anterior. Només cal que pugueu `insertSumsPrefixes.cc` al jutge.

Entrada

L'entrada té un nombre arbitrari de casos. Cada cas consisteix en una llista d'enters en una línia. 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 el corresponent resultat de la funció. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquesta sortida. Només cal que implementeu la funció abans esmentada.

Exemple d'entrada

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

```
5 6 9 3 7 4 5
5 4 7 4 4
```

Exemple de sortida

```
6 6 7 13 5 18 3 21 5 26 6 32
9 9 1 10 2 12 7 19 0 19
3 3 6 9 0 9 6 15 2 17 6 23
8 8 7 15 9 24 2 26 0 26 2 28 3 31 7 38 5
```

```
2 2 8 10 9 19 7 26 3 29
1 1 2 3 9 12 3 15 1 16 9 25 4 29 7 36 8 44
5 5 0 5 3 8 6 14 1 15 0 15 6 21
2 2 0 2 6 8 1 9 5 14 5 19 4 23 7 30
5 5 6 11 9 20 3 23 7 30 4 34 5 39
4 3 9 4 9 7 16 4 20 4 24
```

Observació

La vostra funció i subfuncions que creeu han de treballar només amb llistes. Avaluació sobre 10 punts:

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

Entenem com a solució ràpida una que és correcta, de cost lineal i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics.

Una solució que altera, o elimina els elements originals de la primera llista i els torna a afegir més tard rebrà un 0.

Informació del problema

Autor : PRO2

Generació : 2024-03-29 00:00:33

© *Jutge.org*, 2006–2024.

<https://jutge.org>