
Duplica els elements d'una pila amb sumes parcials

X25844_ca

Donada la classe *pila* que permet apilar elements en una estructura simplement encadenada en memòria dinàmica, cal implementar el mètode

```
void duplica_amb_sumes();
```

que, a partir d'una pila qualsevol, duplica els elements de la pila posant a sobre de cada element un nou element que conté la suma dels elements restants de la pila original (la suma dels elements fins arribar al fons de la pila original). Pots veure exemples en els jocs de prova públics.

Cal enviar a jutge.org la següent especificació de la classe *pila* i la implementació del mètode dins del mateix fitxer. La resta de mètodes públics i privats ja estan implementats. Indica dins d'un comentari a la capçalera del mètode el seu cost en funció del nombre d'elements n de la pila.

```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
typedef unsigned int nat;
```

```
template <typename T>
```

```
class pila { // pila en memòria dinàmica
```

```
public:
```

```
    pila ();
```

```
    // Crea pila buida
```

```
    ~pila ();
```

```
    // Destrueix el p.i.
```

```
    pila (const vector<int> &v);
```

```
    // Crea pila amb els elements de v amb el mateix ordre.
```

```
    nat longitud() const;
```

```
    // Retorna el nombre d'elements del p.i.
```

```
    void mostra() const;
```

```
    // Mostra el p.i. pel canal estàndard de sortida.
```

```
    void duplica_amb_sumes();
```

```
    // Pre: true
```

```
    // Post: S'han duplicat els elements de la pila, a sobre de cada element hi ha un
```

```
    // nou element que conté la suma dels elements restants de la pila original
```

```
private:
```

```
    struct node {
```

```
        T info;
```

```
        node* seg;
```

```

};
node* _cim; // Apunta al cim de la pila
nat _mida; // Nombre d'elements de la pila

// Aquí va l'especificació dels mètodes privats addicionals
};

// Aquí va la implementació del mètode duplica_amb_sumes i dels privats addicionals

```

Degut a que jutge.org només permet l'enviament d'un fitxer amb la solució del problema, en el mateix fitxer hi ha d'haver l'especificació de la classe i la implementació del mètode *duplica_amb_sumes* (el que normalment estarien separats en els fitxers *.hpp* i *.cpp*). Per testejar la solució, jutge.org ja té implementats la resta de mètodes de la classe *pila* i un programa principal que llegeix una pila i la mostra, després crida el mètode *duplica_amb_sumes* i finalment mostra el contingut de la pila modificada.

Entrada

L'entrada conté una línia formada per una seqüència d'enters, són els elements que tindrà la pila inicial.

Sortida

Es mostra el contingut de la pila abans i després de fer el duplicat amb sumes. De la pila s'escriu el nombre d'elements de la pila seguit d'un espai i dels elements de la pila entre claudàtors i separats per espais.

Observació

Només cal enviar l'especificació de la classe *pila*, la implementació del mètode *duplica_amb_sumes* i el seu cost en funció del nombre d'elements n de la pila inicial. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat. **No es poden usar estructures de dades auxiliars com per exemple vectors.**

Exemple d'entrada 1

0

Exemple de sortida 1

0 []
0 []

Exemple d'entrada 2

7

Exemple de sortida 2

1 [7]
2 [7 7]

Exemple d'entrada 3

2 2

Exemple de sortida 3

2 [2 2]
4 [4 2 2 2]

Exemple d'entrada 4

3 3 3

Exemple de sortida 4

3 [3 3 3]
6 [9 3 6 3 3 3]

Exemple d'entrada 5

-6 0 -2 3

Exemple d'entrada 6

1 2 3 4 5 6 7 8 9

Exemple d'entrada 7

-9 -8 -7 -6 -5 -4 -3 -2 -1

Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T14:37:44.570Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>

Exemple de sortida 5

4 [-6 0 -2 3]
8 [-5 -6 1 0 1 -2 3 3]

Exemple de sortida 6

9 [1 2 3 4 5 6 7 8 9]
18 [45 1 44 2 42 3 39 4 35 5 30 6 24 7 17 8 9 9]

Exemple de sortida 7

9 [-9 -8 -7 -6 -5 -4 -3 -2 -1]
18 [-45 -9 -36 -8 -28 -7 -21 -6 -15 -5 -10 -4 -6 -3 -3]