Elimina majors suma anteriors en una cua no circular X57859_ca

Donada la classe *cua* que permet encuar elements en una estructura simplement encadenada no circular en memòria dinàmica, cal implementar el mètode

void *elimina_majors_suma_anteriors* ();

que modifica la cua eliminant els elements que són majors que la suma dels inserits abans en la cua original (o sigui, cal eliminar els elements x que són majors que la suma dels que estan entre el front i l'anterior de x). Pots veure exemples en els jocs de prova públics.
Cal enviar a jutge.org la següent especificació de la classe *cua* 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 cua del p.i.

```
#include <iostream>
#include <vector>
using namespace std;
typedef unsigned int nat;

template <typename T>
class cua { // Cua no circular en memòria dinàmica
  public:
    cua();
    // Construeix una cua buida.

    ~cua();
    // Destrueix el p.i.

    cua(const vector<int> &v);
    // Crea cua 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 elimina_majors_suma_anteriors();
    // Pre: c = C
    // Post: El resultat és C on s'han eliminat els elements que són majors
    // que la suma dels inserits abans que ells en la cua original C.

  private:
    struct node {
      T info;
      node* seg;
    };
```

```
    node* _pri; // Apunta al primer element de la cua
    node* _ult; // Apunta al darrer element de la cua
    nat _mida;

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

// Aquí va la implementació del mètode públic elimina_majors_suma_anteriors i 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 *elimina_majors_suma_anteriors* (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 *cua* i un programa principal que llegeix una cua, després crida el mètode *elimina_majors_suma_anteriors* i finalment mostra el contingut de la cua resultant.

Entrada

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

Sortida

Es mostra el contingut de la cua després d'eliminar els majors que la suma dels anteriors: el nombre d'elements de la cua seguit d'un espai i dels elements de la cua entre claudàtors i separats per espais.

Observació

Només cal enviar l'especificació de la classe *cua*, la implementació del mètode *elimina_majors_suma_anteriors* i el seu cost en funció del nombre d'elements *n* de la cua 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	Exemple de sortida 1
	`0 []`
Exemple d'entrada 2	**Exemple de sortida 2**
`2`	`0 []`
Exemple d'entrada 3	**Exemple de sortida 3**
`-2`	`1 [-2]`
Exemple d'entrada 4	**Exemple de sortida 4**
`2 5 1 9`	`1 [1]`
Exemple d'entrada 5	**Exemple de sortida 5**
`-2 5 2 4`	`3 [-2 2 4]`

Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T17:04:29.906Z