

---

## Control PRO2 - Torn 2 (primavera 2015)

X45504\_ca

---

Hem decidit estendre la classe `ListaPalabras` que heu vist a la sessió 5 de laboratori (sense el mètode `max_freq`) amb dues noves funcionalitats:

- modificar la llista eliminant (si és possible) una de les aparicions d'una paraula donada en la llista
- obtenir la freqüència mitjana de les paraules amb una longitud determinada.

Hem optat per afegir dues noves operacions públiques amb la següent especificació:

```
double frec_media_long(int l) const;
/* Pre: 0 < l <= Palabra::longitud_maxima() */
/* Post: el resultado es la frecuencia media de las palabras de
longitud l en el parámetro implícito (cero si no hay ninguna) */

void borrar_palabra(const Palabra & p);
/* Pre: si p pertenece al p.i., tiene frecuencia f */
/* Post:
    si p no pertenece al p.i., no cambia nada;
    si f = 1, p ya no pertenece al p.i.;
    si f > 1, la frecuencia de p se ha decrementado en 1 */
```

Això ha suposat afegir atributs nous a la classe. Tenint això en compte, només heu d'implementar eficientment dues operacions públiques: `borrar_palabra` i

```
void anadir_palabra(const Palabra &p);
/* Pre: el número de palabras en el parámetro implícito es menor que la
longitud máxima o p ya está en él; p no es la palabra vacía */
/* Post: si p estaba en el parámetro implícito original, su frecuencia se ha
incrementado en 1; si no, p pasa a estar en él, con frecuencia 1 */
```

Per implementar aquestes operacions heu d'usar una operació auxiliar privada que també heu d'implementar:

```
void buscar_palabra_posicion(const Palabra &p, bool &b, int &i) const;
/* Pre: cierto */
/* Post: b indica si p está en el parámetro implícito;
    si b, entonces i es la posición en la que está p; si no, i es la
longitud del parámetro implícito */
```

## Observació

Heu de lliurar un fitxer `solucio.cc` amb una implementació eficient de les operacions `anadir_palabra`, `borrar_palabra` i `buscar_palabra_posicion` que ha de tenir el següent format:

```
#include "ListaPalabras.hh"

void ListaPalabras::anadir_palabra(const Palabra &p)
{
    ... // codi de la implementació
}

void ListaPalabras::borrar_palabra(const Palabra &p)
```

```

{
    ... // codi de la implementació
}

void ListaPalabras::buscar_palabra_posicion(const Palabra &p, bool &b, int &i) const
{
    ... // codi de la implementació
}

```

Copieu aquesta plantilla en el vostre `solucio.cc` i completeu-la. El vostre `solucio.cc` no pot contenir la implementació d'altres operacions de la classe.

A l'apartat *Public files* del Jutge us proveïm amb material addicional comprimit en un fitxer `.tar`. Podeu descomprimir aquest fitxer amb la comanda

```
tar -xvf nom_fitxer.tar
```

Aquest material addicional consisteix en els següents fitxers:

- `ListaPalabras.hh`: l'especificació Pre/Post de totes les operacions públiques i privades d'aquesta nova versió de la classe `ListaPalabras`, així como la definició dels camps privats. Fixeu-vos que hi ha dos atributs nous que són vectors d'enters:
  - `npar_long`, que conté el nombre de paraules diferents de la llista de cada longitud
  - `suma_frec_long`, que conté la suma de freqüències de les paraules de la llista de cada longitud.

Aquest dos atributs ens han permès implementar amb la màxima eficiència l'operació `frec_media_long` i cal actualitzar-los en les operacions modificadores de la classe. Fixeu-vos també en l'invariant de l'atribut `paraules`. **És molt important que la implementació de les operacions que us hem encarregat tingui en compte i preservi l'invariant de la representació de la classe `ListaPalabras`.**

- `ListaPalabras.cc`: la implementació de totes de les operacions de la nova versió de la classe `ListaPalabras` tret de les operacions `anadir_palabra`, `borrar_palabra` i `buscar_palabra_posicion`.
- `Palabra.hh`: l'especificació de la classe `Palabra` i la definició dels seus atributs.
- `Palabra.cc`: la implementació de la classe `Palabra`.
- `pro2.cc`: un programa principal que podeu fer servir per provar les operacions públiques d'aquesta versió de la classe `ListaPalabras`.
- `llegeixme.txt`: instruccions per a generar l'executable del programa `pro2` i provar-lo.

Valorarem positivament que la solució no contingui instruccions (especialment bucles o crides a operacions costoses) ni objectes (especialment vectors o llistes de paraules) innecessaris. No es pot emprar cap estructura de dades que no hagi aparegut a les sessions 1-5 de laboratori.

Quan feu els enviaments el Jutge us indicarà quants jocs de proves passeu i de quin tipus (public o privat). El joc de proves anomenat `public` és el que teniu al fitxer `llegeixme.txt`. El `privat1` és molt semblant al `public`. Els `privat2` i `privat3` proven situacions especials.

## **Informació del problema**

Autoria: Professors de PRO2

Generació: 2026-01-25T16:20:13.969Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>