

---

## Cerca en una llista d'estudiants amd esborrat

X25718\_ca

---

Implementeu eficientment l'operació `esborra_tots` especificada a continuació.

```
void esborra_tots ( list<Estudiant> &t, int x)
/* Pre: x>=0, t = T */
/* Post: t es com a T pero sense cap estudiant amb dni = x */
```

Heu d'enviar tres fitxers en un sol `.tar`:

- `LlistaIOEstudiant.hh` amb les funcions:

```
void LlegirLlistaEstudiant(list<Estudiant>& l);
// Pre: l és buida; el canal estandar d'entrada conté parelles
// de valors <enter, double>, acabat per un parell 0 x (qualsevol
double)
// Post: s'han afegit al final de l els estudiants llegits fins al
0 x (no inclòs)

void EscriureLlistaEstudiant(const list<Estudiant>& l);
// Pre: cert
// Post: s'han escrit al canal estandar de sortida els elements
de l
```
- `LlistaIOEstudiant.cc` amb la seva codificació.
- `solution.cc` amb l'operació de l'enunciat, amb `#include "LlistaIOEstudiant.hh"` i sense `main` ni cap altre codi

Teniu en compte que aquestes operacions han de funcionar sempre que ens arribi una seqüència d'estudiants acabada en un estudiant amb `dni = 0` i qualsevol nota. La seqüència podrà tener estudiants sense nota i amb dnis repetits.

Observeu que us donem el `Makefile` per compilar i un `main` per executar i fer proves.

### Exemple d'entrada 1

```
10 5
11 4
12 5
14 3
15 8
0 0
```

12

### Exemple d'entrada 2

```
1 10
2 10
3 -1
1 10
2 10
3 10
1 11
```

### Exemple de sortida 1

```
10 5
11 4
14 3
15 8
```

```
3 10
2 10
0 0

3
```

## Exemple de sortida 2

1 10  
2 10

1	10
2	10
1	NP
2	10

## Informació del problema

Autoria: J. Baixeries (adaptador) Borja Valles (responsable)

Generació: 2026-01-25T21:05:45.141Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>