

---

## Quants aprovats?

S66165\_ca

---

Disposem d'una llista d'estudiants amb el seu DNI i la seva nota, ordenada per nota de **major a menor**. La llista té almenys un estudiant aprovat ( $\text{nota} \geq 5.0$ ) i un suspès ( $\text{nota} < 5.0$ ). Volem saber quants estudiants han aprovat.

Disposem d'una tupla `Estudiant` amb la declaració:

```
#include <vector>
#include <string>
using namespace std;

struct Estudiant {
    string dni;
    double nota;
};

/**
 * @brief Retorna el nombre d'estudiants que han aprovat ( $\text{nota} \geq 5.0$ ).
 *
 * @pre El vector `E` està ordenat de major a menor nota.
 *      El vector té almenys un estudiant aprovat i un suspès.
 */
int quants_aprovats(const vector<Estudiant>& E);
```

Fes una funció `quants_aprovats` amb la declaració de dalt.

Per exemple, si tenim la llista d'estudiants:

```
12345678A 9.5
23456789B 7.2
34567890C 5.0
45678901D 4.9
56789012E 3.1
```

La funció hauria de retornar 3, ja que hi ha 3 estudiants amb  $\text{nota} \geq 5.0$ .

## Observació

Aquest problema té com a centre d'interès l'*eficiència*. Feu servir el millor algorisme possible a la funció i considereu que rebreu un vector de grans dimensions.

**IMPORTANT:** Quan envieu aquest problema, cal posar en el fitxer `.cc` només: *a*) els `includes`, *b*) la declaració de la tupla `Estudiant` i *c*) la definició de la funció `quants_aprovats`. Si modifiqueu el `main.cc` per poder provar en el vostre ordinador, podeu esborrar (o comentar amb `/* i */`) la funció `main i llegir_estudiants`.

## Entrada

L'entrada són diversos casos de llistes d'estudiants. De la lectura ja se n'encarrega el programa principal. Cada cas comença amb un enter  $n$  ( $n \geq 2$ ) seguit de  $n$  línies amb el DNI i

la nota de cada estudiant. La llista està ordenada per nota de **major a menor** i té almenys un estudiant aprovat i un suspès.

## Sortida

La sortida és el nombre d'aprovat per a cada llista. De l'escriptura també se n'encarrega el programa principal.

### Exemple d'entrada 1

```
5
12345678A 9.5
23456789B 7.2
34567890C 5.0
45678901D 4.9
56789012E 3.1

4
11111111A 10.0
22222222B 8.5
33333333C 6.0
44444444D 4.5

3
55555555A 5.0
66666666B 3.2
77777777C 2.1

2
88888888A 9.0
99999999B 4.0
```

### Exemple de sortida 1

```
3
3
1
1
```

## Informació del problema

Autoria: Pau Fernández

Generació: 2026-02-03T07:29:36.729Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>