
Cuántos aprobados?

S66165_es

Disponemos de una lista de estudiantes con su DNI y su nota, ordenada por nota de **mayor a menor**. La lista tiene al menos un estudiante aprobado (nota ≥ 5.0) y uno suspendido (nota < 5.0). Queremos saber cuántos estudiantes han aprobado.

Disponemos de una tupla `Estudiant` con la declaración:

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

struct Estudiant {
    string dni;
    double nota;
};

/**
 * @brief Devuelve el número de estudiantes que han aprobado (nota >= 5.0).
 *
 * @pre El vector `E` está ordenado de mayor a menor nota.
 *      El vector tiene al menos un estudiante aprobado y uno suspendido.
 */
int quants_aprovats(const vector<Estudiant>& E);
```

Haz una función `quants_aprovats` con la declaración de arriba.

Por ejemplo, si tenemos la lista de estudiantes:

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

La función debería devolver 3, ya que hay 3 estudiantes con nota ≥ 5.0 .

Observación

Este problema tiene como centro de interés la *eficiencia*. Usad el mejor algoritmo posible en la función y considerad que recibiréis un vector de grandes dimensiones.

IMPORTANTE: Al enviar este problema, hay que poner en el fichero `.cc` solo: *a)* los includes, *b)* la declaración de la tupla `Estudiant` y *c)* la definición de la función `quants_aprovats`. Si modificáis el `main.cc` para poder probar en vuestro ordenador, podéis borrar (o comentar con `/* y */`) la función `main` y `llegir_estudiants`.

Entrada

La entrada son varios casos de listas de estudiantes. De la lectura ya se encarga el programa principal. Cada caso empieza con un entero n ($n \geq 2$) seguido de n líneas con el DNI y la

nota de cada estudiante. La lista está ordenada por nota de **mayor a menor** y tiene al menos un estudiante aprobado y uno suspendido.

Salida

La salida es el número de aprobados para cada lista. De la escritura también se encarga el programa principal.

Ejemplo de 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
```

Ejemplo de salida 1

```
3
3
1
1
```

Información del problema

Autoría: Pau Fernández

Traducción: Pau Fernández

Generación: 2026-02-03T07:29:41.610Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>