

Volem gestionar les notes d'un conjunt d'estudiants. Cada estudiant té un cert nombre de notes parcials, incloent la possibilitat de què algunes siguin "no presentat". La nota definitiva d'un estudiant és la mitjana ponderada de les seves notes parcials respecte a una col·lecció de pesos determinats, publicats a la Guia Docent i per tant ben coneguts. Tots els estudiants del conjunt tenen el mateix nombre de notes parcials, que coincideix amb el nombre de pesos de la col·lecció.

Més formalment, si un estudiant té n notes parcials, la seva nota definitiva (també anomenada nota ponderada) és

$$nota_def = \frac{\sum_{i=1}^n nota_parcial(i) * pes(i)}{\sum_{i=1}^n pes(i)}$$

si alguna de les notes parcials amb pes més gran que 0 és diferent de "no presentat". En cas contrari, la nota definitiva és "no presentat". Noteu que les notes parcials "no presentat" compten com a 0 a la fórmula.

Per circumstàncies extraordinàries, cada estudiant pot optar en un moment donat a que se li apliqui una col·lecció de pesos alternativa per calcular la nota definitiva. La nova opció treu pes a algunes de les notes parcials per donar més pes a la darrera nota parcial. Per això, aquesta opció s'anomena *avaluació final*.

S'han adaptat les classes `Cjt_estudiants` i `Estudiant`, que heu vist al laboratori, per donar suport a la gestió de les notes (fixeu-vos als nous camps i a les noves operacions). Noteu també que hem definit una classe nova, anomenada *Ponderacio*, que representa una col·lecció de pesos.

Aquest exercici consisteix en implementar dues operacions. La primera pertany a la classe `Estudiant` i serveix per calcular la nota ponderada d'un estudiant. Hem optat per la següent especificació:

```
double Estudiant::calcular_nota_ponderada(const Ponderacio& p1, const Ponderacio& p2){
    /* Pre: el nombre de notes parcials del p.i. és > 0 i és igual al nombre de
       pesos de p1 i de p2 */
    /* Post: el resultat és la nota ponderada del p.i. aplicant-li p1 a les
       seves notes parcials si no ha demanat avaluació final i p2 en cas contrari;
       si no té cap nota amb pes > 0, el resultat és -1.0 */
}
```

La segona, que no pertany a cap classe, s'especifica així:

```
pair<Estudiant, double> max_nota_ponderada(const Cjt_estudiants& c, const Ponderacio& p1, const Ponderacio& p2){
    /* Pre: c no és buit; el nombre de notes parcials dels elements de c és > 0
       i és igual al nombre de pesos de p1 i de p2 */
    /* Post: el primer component del resultat és l'element de c amb una nota
       ponderada més gran, on als estudiants que no han demanat avaluació final
       se'ls aplica p1 i a la resta se'ls aplica p2; en cas d'empat el resultat
       és el de DNI més petit; el segon component és la màxima nota ponderada */
}
```

Observació

Heu de lliurar un fitxer `solucio.cc` amb una implementació eficient de les dues operacions. Resultaran **d'especial interès** les noves operacions de la classe `Estudiant`. Fixeu-vos preferentment a les especificacions!

A l'apartat *Public files* del Jutge (icona del gatet) us proveim amb una plantilla pel fitxer `solucio.cc` que només cal reanomenar i completar. A més, hi trobareu material addicional, tot 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:

- `Cjt_estudiants.hh`: l'especificació de la classe `Cjt_estudiants` i la definició dels seus atributs.
- `Cjt_estudiants.cc`: la implementació dels mètodes de la classe `Cjt_estudiants`.
- `Estudiant.hh`: l'especificació de la classe `Estudiant` i la definició dels seus atributs.
- `Estudiant.cc`: la implementació dels mètodes de la classe `Estudiant`, tret de `calcular_nota_ponderacio`.
- `Ponderacio.hh`: l'especificació de la classe `Ponderacio` i la definició dels seus atributs.
- `Ponderacio.cc`: la implementació dels mètodes de la classe `Ponderacio`.
- `pro2.cc`: un programa principal que podeu fer servir per provar la solució d'aquest exercici.
- `llegeixme.txt`: instruccions per a generar i provar l'executable del programa `pro2`.
- `sampleX.inp`: entrades dels jocs de proves públics
- `sampleX.cor`: sortides dels jocs de proves públics
- `solucio.cc.txt`: plantilla per omplir la solució de l'examen

És fonamental que la solució sigui eficient en temps i espai. No es pot emprar cap estructura de dades que no hagi aparegut a les sessions 1-4 de laboratori.

Quan feu els enviaments el Jutge us indicarà quants jocs de proves passeu i de quin tipus (públic o privat).

Hi ha dos jocs de proves públics i dos de privats. El primer joc públic només prova la primera operació.

Informació del problema

Autor : Professors de PRO2

Generació : 2020-03-06 15:47:39

© *Jutge.org*, 2006–2020.

<https://jutge.org>