
Control PRO2 - Problema 1 (Tardor 2019)

X20155_ca

Es disposa de $0 \leq k$ beques que s'atorgaran als k estudiants amb millors notes d'un conjunt c . A més, tots els estudiants becats han de tenir una nota superior o igual a un cert valor donat m , amb $0 \leq m \leq \text{Estudiant::nota_maxima}()$; si no existeixen k estudiants que satisfacin aquesta condició, llavors les beques corresponents es quedaran sense atorgar.

Per una altra part, si hi ha més d'un estudiant amb la mateixa nota, es prioritzaran per a les beques els estudiants amb DNI més gran.

S'han adaptat les classes `Cjt_estudiants` i `Estudiant`, que heu vist al laboratori, per donar suport a la gestió de les beques (fixeu-vos a les noves operacions). Noteu també que si volem llegir un conjunt, hem de proporcionar primer de tot els valors k i m .

En aquest exercici volem produir una nova operació, **que no pertany a cap classe**, per escriure els estudiants becats d'un conjunt d'estudiants en ordre creixent per DNI.

Hem optat per la següent especificació:

```
void escriure_becats(const Cjt_estudiants& c) {
    /* Pre: cert */
    /* Post: s'han escrit pel canal estàndar de sortida els estudiants
       de c amb beca en ordre ascendent per DNI */
}
```

Observació

Heu de lliurar un fitxer `solution.cc` amb una implementació eficient de l'operació `escriure_becats`. Aquest fitxer no pot accedir als elements privats de les classes. Òbviament, sí que pot fer-ne servir les operacions públiques. Resultaran **d'especial interès** les noves operacions de `Cjt_estudiants` i `Estudiant`.

A l'apartat *Public files* del Jutge (icona del gatet) us proveïm amb una plantilla pel fitxer `solution.cc` que només cal 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`: les capçaleres i l'especificació Pre/Post de les operacions públiques i privades de la classe `Cjt_estudiants`, així com la definició dels seus atributs.
- `Cjt_estudiants.o`: el fitxer **objecte** de la implementació de totes de les operacions de la nova versió 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`.
- `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`.
- `sample.inp`: entrada del joc de proves públic
- `sample.cor`: sortida del joc de proves públic

- `solution.cc`: plantilla explicada previament
- `solution.hh`: les capçaleres de les noves operacions, per fer `#include`

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 (public o privat). El joc de proves anomenat `public` s'explica al fitxer `llegeixme.txt`. El `privat1` és molt semblant al `public`. Els `privat2` i `privat3` proven situacions especials.

Informació del problema

Autor : Professors de PRO2
Generació : 2019-10-09 16:16:05

© *Jutge.org*, 2006–2019.
<https://jutge.org>