

Hemos decidido extender la clase `Cjt_estudiants` que habéis visto en el laboratorio con dos nuevas funcionalidades.

La primera funcionalidad calcula la nota de laboratorio, la nota de teoría y la nota global de un estudiante en una asignatura a partir de sus notas en los exámenes parciales de dicha asignatura, y asigna las notas de laboratorio, teoría y global calculadas a dicho estudiante. Las notas de laboratorio, teoría y global de un estudiante pueden ser -1 , si se considera que la calificación del estudiante en ese aspecto de la asignatura debe ser NP (No Presentado), o un valor real en el rango $[0 \dots \text{Estudiant::nota_maxima}())$, que corresponde a: 1) la media de las notas de los exámenes parciales de laboratorio en el caso de la nota de laboratorio; 2) la media de las notas de los exámenes parciales de teoría en el caso de la nota de teoría; y 3) la media de la nota de laboratorio y de la nota de teoría en el caso de la nota global de la asignatura. Para realizar la evaluación global de los estudiantes de una asignatura, la clase `Cjt_estudiants` necesita representar cierta información sobre dicha asignatura, en particular, el número de exámenes parciales que se realizan en la asignatura, el subconjunto de los exámenes parciales que contribuyen a la nota de laboratorio, y el subconjunto de los exámenes parciales que contribuyen a la nota de teoría. Por este motivo, hemos añadido cinco atributos nuevos a la clase `Cjt_estudiants`: (1) `nombre_parcials` de tipo entero, que representa el número total de exámenes parciales de la asignatura; (2) `nlab` de tipo entero, que representa el número de exámenes parciales que contribuyen a la nota de laboratorio; (3) `lab` de tipo `vector<bool>`, que permite representar el subconjunto de los exámenes parciales que contribuyen a la nota de laboratorio; (4) `nteo` de tipo entero, que representa el número de exámenes parciales que contribuyen a la nota de teoría; y (5) `teoria` de tipo `vector<bool>`, que permite representar el subconjunto de los exámenes parciales que contribuyen a la nota de teoría. En esta representación del tipo `Cjt_estudiants` asumimos que el identificador del j -ésimo examen parcial de una asignatura es j . Por tanto, los vectores `lab` y `teoria` mencionados anteriormente representan dos subconjuntos disjuntos del conjunto de enteros $\{1, \dots, \text{nombre_parcials}\}$ que contienen respectivamente los identificadores de los exámenes parciales de laboratorio y de los exámenes parciales de teoría.

Concretamente, la evaluación global de un estudiante se realiza de la siguiente manera. Si un estudiante no se ha presentado a ninguno de los exámenes parciales de laboratorio, su nota de laboratorio es -1 (que es el valor elegido para representar la calificación "No Presentado" en nuestra implementación del tipo `Cjt_estudiants`). En otro caso, la nota de laboratorio es la suma de las notas del estudiante en los exámenes parciales de laboratorio a los que se ha presentado dividida por el número de exámenes parciales de laboratorio (`nlab`) de la asignatura. Del mismo modo, si un estudiante no se ha presentado a ninguno de los exámenes parciales de teoría, su nota de teoría es -1 (i.e. No Presentado). En otro caso, la nota de teoría es la suma de las notas del estudiante en los exámenes parciales de teoría a los que se ha presentado dividida por el número de exámenes parciales de teoría (`nteo`) de la asignatura. Finalmente, la nota global de un estudiante en la asignatura es -1 (i.e. No Presentado) si su nota de laboratorio es -1 o su nota de teoría es -1 . En otro caso, la nota global es simplemente la media de la nota de laboratorio del estudiante en la asignatura y de su nota de teoría en la asignatura.

Esta funcionalidad se implementa en la clase `Cjt_estudiants` mediante el método privado

```

void avaluacio_global_iesim(int i);
/* Pre: 1 <= i <= mida()
   El estudiante i-ésimo tiene notas asignadas para todos los exámenes
   parciales de la asignatura. La nota de cada examen parcial puede
   ser -1, si el estudiante no se ha presentado a dicho examen, o una
   nota válida en el rango [0...Estudiant::nota_maxima()]. */
/* Post: El estudiante i-ésimo pasa a tener asignadas sus notas de
   laboratorio, teoría y global en la asignatura, que pueden ser -1,
   si se considera que su calificación debe ser NP (No Presentado),
   o una nota válida en el rango [0...Estudiant::nota_maxima()]
   que corresponde a: 1) la suma de las notas del estudiante en los
   exámenes parciales de laboratorio a los cuales se ha presentado
   dividida por el número de exámenes parciales de laboratorio de
   la asignatura (nlab) en el caso de la nota de laboratorio; 2) la
   suma de las notas del estudiante en los exámenes parciales de
   teoría a los cuales se ha presentado dividida por el número de
   exámenes parciales de teoría de la asignatura (nteo) en el caso
   de la nota de teoría; y 3) la media de la nota de laboratorio y
   de la nota de teoría en el caso de la nota global de la asignatura. */

```

Por ejemplo, si c es un objeto de la clase `Cjt_estudiants` que representa los estudiantes de una asignatura con 5 exámenes parciales de los cuales 3 exámenes parciales son de laboratorio y 2 son de teoría, los identificadores de los exámenes parciales de laboratorio son $\{1, 3, 4\}$, los identificadores de los exámenes parciales de teoría son $\{2, 5\}$, y las notas en los exámenes parciales del quinto estudiante de c son $\{-1, 9, -1, -1, 8\}$, después de la llamada `c.avaluacio_global_iesim(5)`, la nota de laboratorio del quinto estudiante de c debe ser -1 (i.e. No Presentado), su nota de teoría debe ser 8.5 y su nota global -1 (i.e. No Presentado). Del mismo modo, si las notas en los exámenes parciales del cuarto estudiante de c son $\{-1, 9.75, 9.75, 5, -1\}$, después de la llamada `c.avaluacio_global_iesim(4)`, la nota de laboratorio del cuarto estudiante de c debe ser 4.92, su nota de teoría 4.88 y su nota global 4.9.

La segunda funcionalidad añadida a la clase `Cjt_estudiants` calcula y escribe en el canal estándar de salida el subconjunto de exámenes parciales a los que no se ha presentado alguno de los estudiantes que ha aprobado la asignatura, es decir, algún estudiante cuya nota global en la asignatura es mayor o igual que 5. Esta funcionalidad se implementa en la clase `Cjt_estudiants` mediante el método público

```

void parcials_no_presentats_aprovats() const;
/* Pre: Todos los estudiantes del parámetro implícito tienen notas
   asignadas para todos los exámenes parciales y también tienen
   asignada su nota global en la asignatura. La nota global y la
   nota de cada examen parcial puede ser -1, si el estudiante tiene
   la calificación global No Presentado o si no se ha presentado al
   examen parcial, o una nota válida en el rango
   [0...Estudiant::nota_maxima()]. */
/* Post: En el canal de salida estándar se han escrito los
   identificadores de los exámenes parciales a los cuales no se
   ha presentado algún estudiante aprobado en la asignatura, es
   decir, algún estudiante que tiene una nota global en la
   asignatura mayor o igual que 5. Los identificadores de
   estos exámenes parciales están ordenados crecientemente. */

```

Por ejemplo, si c es un objeto de la clase `Cjt_estudiants` que representa los estudiantes de una asignatura con 5 exámenes parciales en el que hay exactamente 4 estudiantes $\{e_{i_1}, \dots, e_{i_4}\}$ con nota global mayor o igual que 5, y sabemos que de esos 4 estudiantes e_{i_3} no se ha presentado ni al primer ni al quinto exámenes parciales, y e_{i_4} no se ha presentado ni al primer ni al

cuarto exámenes parciales, después de la llamada `c.parcials_no_presentats_aprovats()` se escribirá en la pantalla el subconjunto $\{1,4,5\}$, ya que al menos un estudiante aprobado no se ha presentado a estos exámenes parciales.

Para implementar estas funcionalidades hemos modificado también la representación de la clase `Estudiant` de la manera descrita en el archivo `Estudiant.hh`. En particular, representamos las notas de un estudiante en cuatro atributos nuevos: (1) `nota_lab` de tipo `double`; (2) `nota_teorica` de tipo `double`; (3) `nota_global` de tipo `double`; y (4) `notes_parcials` de tipo `vector<double>`. Concretamente, la nota del examen parcial j -ésimo del estudiante parámetro implícito es `notes_parcials[j-1]` para todo j en el rango $1 \leq j \leq \text{notes_parcials.size}()$.

Observación

Teniendo esto en cuenta debéis implementar eficientemente el método privado `avaluacio_global_iesim` y el método público `parcials_no_presentats_aprovats`. Debéis entregar un archivo `solucio.cc` con una implementación eficiente de estos dos métodos. En el caso del método público `parcials_no_presentats_aprovats` podéis completar la implementación parcial que os proponemos, o implementar este método sin utilizar la implementación parcial propuesta.

Es posible superar algunos juegos de prueba (pero no todos) **implementando únicamente el método privado** `avaluacio_global_iesim`, siempre y cuando no modifiquéis la implementación parcial del método público `parcials_no_presentats_aprovats` que os proporcionamos.

Dentro del material adicional que os proporcionamos en el apartado *Public files* del problema del juez encontraréis el archivo `plantilla.txt` con las cabeceras de ambos métodos y una implementación incompleta del método `parcials_no_presentats_aprovats`: debéis renombrar el archivo `plantilla.txt` de manera que se llame `solucio.cc`, completarlo y enviarlo al juez.

Vuestro archivo `solucio.cc` no puede contener la implementación de otros métodos de las clases `Estudiant` o `Cjt_estudiants`.

En el apartado *Public files* del juez os proporcionamos material adicional en un fichero `.tar`. Podéis extraer el contenido de este fichero con la instrucción

```
tar -xvf nom_fitxer.tar
```

Este material adicional contiene los siguientes archivos:

- `plantilla.txt`: es la plantilla del archivo `solucio.cc`; debéis renombrar este archivo de manera que se llame `solucio.cc`, completarlo y enviarlo al juez
- `Cjt_estudiants.hh`: la especificación Pre/Post de todos los métodos públicos y privados de esta nueva versión de la clase `Cjt_estudiants`, así como la definición de los atributos privados.
- `Cjt_estudiants.cc`: la implementación de todos los métodos de la clase `Cjt_estudiants`, excepto la de los métodos que os pedimos.
- `Estudiant.hh`: la especificación de la nueva versión de la clase `Estudiant` y la definición de sus atributos privados.
- `Estudiant.cc`: la implementación de todos los métodos de la clase `Estudiant`.

- `pro2.cc`: un programa principal que podéis utilizar para probar los métodos públicos de esta nueva versión de la clase `Cjt_estudiants`.
- `entrada.txt` y `sortida_correcta.txt`: archivo de entrada del juego de pruebas público, y salida correcta para dicho juego.
- `llegeixme.txt`: instrucciones para generar el ejecutable del programa `pro2.cc` y probarlo.

Valoraremos positivamente que la solución no contenga instrucciones innecesarias (especialmente bucles o llamadas a operaciones costosas), ni objetos (especialmente vectores o matrices) innecesarios, que no haga recorridos cuando debería hacer búsquedas, y que use correctamente las operaciones más adecuadas de las clases `Estudiant` y `Cjt_estudiants` siempre que sea posible. No se puede usar ninguna estructura de datos que no haya aparecido en las sesiones 1 a 4 de laboratorio.

Cuando hagáis envíos, el juez os indicará cuantos juegos de pruebas supera vuestro programa y de qué tipo (público o privado). **Tened en cuenta que es posible superar algunos juegos de prueba (pero no todos) implementando únicamente el método `avaluacio_global_iesim`, si no modificáis la implementación parcial del método `parcials_no_presentats_aprovats` que os proporcionamos.**

Información del problema

Autor : Professors de PRO2

Traductor : Professors de PRO2

Generación : 2018-03-18 14:19:58

© *Jutge.org*, 2006–2018.

<https://jutge.org>