L10: Cuestionarios aleatorios

Estructuras de Datos
Facultad de Informática - UCM

Para realizar este ejercicio, descarga la plantilla que se proporciona en este enunciado (icono del clip al lado del título) o a través del Campus Virtual.

La entrega de este ejercicio consiste en un único fichero `.cpp` que se subirá a *DOMjudge*. Podéis subir tantos intentos como queráis. Se tendrá en cuenta el último intento con el veredicto `CORRECT` que se haya realizado antes de la hora de entrega por parte de alguno de los miembros de la pareja.

No olvidéis poner el nombre de los componentes de la pareja en el fichero `.cpp` que entreguéis. Solo es necesario que uno de los componentes de la pareja realice la entrega.

Evaluación: La práctica se puntuará de 0 a 10 si ha obtenido el veredicto `CORRECT` por parte del juez. En caso contrario, la calificación será de 0. No se realizará evaluación por pares en esta práctica.

Manuel Blanquinegro es un amigo mío que trabaja como profesor en la Universidad Compliquense. El otro día me comentó que estaba harto de que algunos estudiantes de la asignatura *Estrechuras de datos* hiciesen trampa en los exámenes tipo test de la asignatura. Para mitigar este problema, decidió preparar varios modelos de examen, uno para cada estudiante. Todos los modelos tenían las mismas preguntas, pero en distinto orden. Como aquello no parecía suficiente, decidió elaborar un amplio catálogo de preguntas, y que cada modelo de examen consistiese en una selección aleatoria de preguntas de este catálogo. De este modo, los modelos de examen no solo diferían en el orden de las preguntas, sino también en las preguntas en sí.

Cada una de las preguntas del catálogo tiene un identificador alfanumérico sin espacios, como por ejemplo `A01`, `P23`, etc. Una pregunta puede tener distintas respuestas, de las cuales solamente una es correcta. Cada una de las posibles respuestas tiene un identificador, que también puede ser una secuencia de caracteres alfanuméricos sin espacios, como por ejemplo `a`, `b`, `1`, `Verdadero`, etc.

El profesor parecía contento con el nuevo sistema, porque el número de copias ha disminuido notablemente. No obstante, parecía preocupado con el examen de la semana pasada, porque piensa que las preguntas fueron demasiado fáciles. Para poder confirmarlo, quiere conocer cuántos estudiantes han acertado todas las preguntas que les ha tocado resolver.

Para realizar este ejercicio, has de tener en cuenta las siguientes restricciones:

1. Debes utilizar los TADs implementados en la STL de C++. Al final de este enunciado encontrarás en qué se diferencian con respecto a los vistos en clase.
2. No se puede almacenar todo el catálogo de preguntas en memoria, ya que puede ser muy grande. Por el contrario, sí esta permitido almacenar la información relativa a los estudiantes y a las respuestas de sus exámenes.

Cualquier entrega que no cumpla con alguna de estas restricciones **será calificada con 0**, independientemente del veredicto del juez.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso de prueba comienza con tres números:

- N, que es el número de estudiantes de la asignatura ($1 \leq N \leq 10^4$).
- A, que es el número de preguntas que tiene que contestar cada estudiante en su examen ($1 \leq A \leq 100$).
- P, que es el número de preguntas del catálogo ($1 \leq P \leq 10^9$).

Después vienen N líneas, una por cada estudiante. Cada línea comienza con el nombre del estudiante, y va seguida por $2A$ identificadores separados por espacios. Los identificadores van agrupados en pares, uno por cada pregunta que el estudiante tiene que responder. La primera componente de cada par contiene el identificador de la pregunta, y la segunda es el identificador de la respuesta que el estudiante ha marcado para dicha pregunta.

Por último, tenemos una línea con $2P$ identificadores, que son las preguntas del catálogo. Estos identificadores van agrupados en pares, en los que la primera componente indica el identificador de la pregunta, y la segunda componente indica el identificador de la respuesta correcta para esa pregunta.

Tanto los nombres de los estudiantes como los identificadores de preguntas y respuestas son secuencias de caracteres alfanuméricos sin espacios.

La entrada finaliza con `0 0 0`, caso que no se procesa.

Salida

Para cada caso de prueba ha de imprimirse una línea con un número que indica el número de estudiantes que han acertado todas las preguntas que les ha tocado responder.

Entrada de ejemplo

```
4 3 6
Angel A3 a B1 b A1 b
Pedro B2 c C4 a B1 b
Jorge A1 a C4 a A3 a
Adrian C4 c A1 b A3 a
A1 b B1 b A3 a B2 c C3 c C4 a
2 1 1
Gerardo AX Verdadero
Diana AX Verdadero
AX Falso
0 0 0
```

Salida de ejemplo

```
2
0
```

Notas sobre la implementación

Ten en cuenta lo siguiente a la hora de utilizar los diccionarios de la STL:

- En las librerías de la STL se introduce el método `contains()` a partir del estándar C++20. El compilador del juez aún no soporta este estándar. En su lugar, puedes utilizar el método `count()`, que devuelve `1` si la clave está en el diccionario, o `0` en caso contrario.
- A la hora de iterar sobre un diccionario, ten en cuenta que las entradas son objetos de tipo `std::pair<K, V>` con dos atributos: `first` y `second`. Puedes utilizar la sintaxis de C++17 para obtener ambas componentes a la vez:

  ```
  for (auto [k, v] : diccionario) {
    // ...
  }
  ```

Lo explicado aquí **no implica** que deba utilizarse necesariamente la función `count()` o que deba recorrerse un diccionario para resolver el ejercicio.