

## C10: El Bingo de Azahara

Estructuras de Datos  
Facultad de Informática - UCM

Este ejercicio debe entregarse a través del problema C10 de *DOMjudge*. La fecha tope de entrega es el **10 de mayo de 2020**.

La entrega consiste en un único fichero `.cpp` que se subirá a *DOMjudge*. Puedes subir tantos intentos como quieras. 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 del grupo.

No olvides poner el nombre de los componentes del grupo en el fichero `.cpp`. Solo es necesario que uno de los componentes del grupo realice la entrega.

**Evaluación:** Este ejercicio se puntuará de 0 a 10. Para poder obtener una calificación superior a 0 es necesario obtener un veredicto `CORRECT`.

El Bingo es un juego de azar en el que cada jugador dispone de una o varias tarjetas, llamadas *cartones*. Cada cartón tiene impresa una serie aleatoria de números. Por otro lado, un locutor extrae sucesivamente varias bolas numeradas de un bombo, anunciando los números extraídos. Cada vez que el locutor anuncia un número, los jugadores lo tachan de sus cartones, en caso de tenerlo. El primer jugador que consigue tachar todos los números de alguno de sus cartones grita "*¡Bingo!*" y gana el juego.

Azahara Fortunata ha decidido montar un salón con una variante de este juego, llamada *Bingo Injusto*. En esta variante, cada jugador dispone de un único cartón, pero la cantidad de números que contiene el cartón es aleatoria. De este modo, puede haber jugadores que tengan más números que tachar que otros. Otra consecuencia de esto es que puede haber varios jugadores que completen su cartón y terminen gritando *¡Bingo!* simultáneamente. En ese caso todos ellos ganan la partida.

Como Azahara no dispone de mucho presupuesto, ha decidido ella misma tomar el papel de locutora del juego. El problema es que le aburre tener que comprobar cuáles de los cartones son los premiados, y ha decidido hacer un programa que realice esta ingrata tarea por ella. ¿Puedes ayudarla?

**Importante:** En este ejercicio tienes que utilizar las clases de la STL que implementan el TAD Diccionario y el TAD Conjunto. El compilador que está instalado en el servidor de *DOMjudge* no utiliza el estándar C++20, por lo que estas clases no soportan el método `contains()` visto en los videos de teoría. Utiliza, en su lugar, el método `count()`. De este modo, si `c` es un objeto de la clase `set` o `unordered_set`, la expresión `c.count(x)` devuelve 1 si `x` pertenece al conjunto, o 0 en caso contrario. Similarmente ocurre con los diccionarios.

## Entrada

La entrada consta de una serie de casos de prueba, cada uno de ellos describiendo una partida de Bingo. Cada caso de prueba comienza con el número  $N$  de jugadores participantes ( $1 \leq N \leq 2000$ ). A continuación aparecen  $N$  líneas, cada una describiendo el cartón de cada jugador. Cada una de estas líneas comienza con el nombre del jugador y va seguida por la secuencia de números contenidos en el cartón de dicho jugador. Esta secuencia finaliza con el número 0, que no forma parte del cartón.

Con respecto a los nombres de los jugadores, algunos de estos prefieren identificarse mediante un apodo o un número, por lo que el nombre de un jugador puede estar formado por letras y/o números. Eso sí: cada nombre está formado por una única palabra sin espacios o saltos de línea en su interior.

Tras la descripción de los cartones de los participantes, aparece una secuencia de números, que son los que va extrayendo Azahara del bombo hasta que algún/algunos de los jugadores grita *iBingo!*. Suponemos que no hay ningún jugador despistado que se olvide de tachar un número de su cartón, o que se olvide gritar *iBingo!* cuando consigue tachar todos los números de su cartón.

La entrada finaliza con una partida de 0 jugadores, que no se procesa.

## Salida

Para cada caso de prueba se escribirá una línea con los nombres de los jugadores que han ganado la partida. La lista de ganadores ha de estar ordenada de manera ascendente según la relación de orden definida por el operador < de la clase `string`. Si una partida tiene varios ganadores, sus nombres deben estar separados por un espacio.

## Entrada de ejemplo

```
2
Diana 30 21 45 23 47 32 0
Martin 20 21 31 46 78 23 0
20 21 45 23 47 32 46 30
3
Felipe 15 9 21 0
Ricardo 10 2 9 0
Gerardo 7 9 0
15 4 21 6 10 2 7 9
0
```

## Salida de ejemplo

```
Diana
Felipe Gerardo Ricardo
```