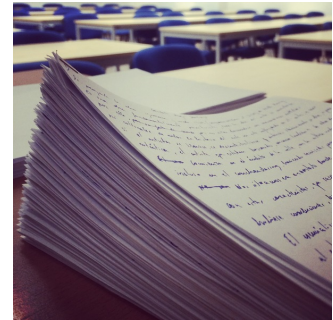


## Me pilló el toro

La evaluación continua se le ha ido de las manos al profesor. Les pide a los alumnos que no lo dejen todo para el final pero él no predica con el ejemplo. Ahora tiene todos los ejercicios que los alumnos han ido entregando durante todo el año en una pila de folios y le toca revisarlos. Los ejercicios o están bien (y entonces puntúan positivamente) o están mal (y entonces restan).

Al final del día quiere tener imprimida una lista con los nombres de todos los alumnos ordenados alfabéticamente y su puntuación en la evaluación continua (resultado de sumar todos los ejercicios que tienen bien menos los que tienen mal). Si un alumno tiene un 0 como balance, no debería aparecer en la lista.

¿Puedes ayudar al profesor?



### Entrada

La entrada consistirá en distintos casos de prueba, cada uno de ellos ocupando un número variable de líneas.

Cada caso de prueba representa los ejercicios de una de las asignaturas del profesor. Cada asignatura comienza con el número de ejercicios que el profesor tiene que corregir ( $1 \leq n \leq 100.000$ ). A continuación aparecen  $2 \cdot n$  líneas con las descripciones de cada ejercicio: una línea con el nombre del autor (hasta 100 caracteres), y otra línea indicando si está bien (CORRECTO) o mal (INCORRECTO).

Los casos de prueba terminarán cuando el número de ejercicios sea cero.

### Salida

Para cada caso de prueba se escribirá la lista de alumnos evaluados por orden alfabético. La evaluación de un alumno consiste en contar el número de ejercicios correctos menos el de ejercicios incorrectos. Un alumno se considera evaluado cuando el resultado anterior es distinto de cero.

Para escribir la lista se utilizará una línea por alumno evaluado. Cada línea tendrá el nombre (y quizá apellidos) del alumno seguido de una coma (,) un espacio y el resultado de la evaluación. Tras la lista, aparecerá una línea con tres guiones (---).

### Entrada de ejemplo

```
2
Laureano
INCORRECTO
Aldonza
CORRECTO
3
Aldonza
CORRECTO
Aldonza
INCORRECTO
Laureano
CORRECTO
1
Aldonza Lorenzo
CORRECTO
0
```

## Salida de ejemplo

```
Aldonza, 1
Laureano, -1
---
Laureano, 1
---
Aldonza Lorenzo, 1
---
```

**Autor:** Marco Antonio Gómez Martín.

# 29

## Capítulos repetidos

Andrés es un forofo de las series de televisión, se las ve todas y todos los capítulos que se emiten. Lo que no le gusta es cuando una cadena elige una serie y empieza a emitir sus capítulos de forma desordenada (no siguiendo la cronología de las temporadas) y repitiendo capítulos sin ningún criterio. ¡Hay veces que vuelve a ver el mismo capítulo dos días después!

Para cada una de sus series favoritas, ha estado apuntando los números de los capítulos que se emiten cada día, y ahora se pregunta cuál es, para cada serie, el periodo de días más largo en el que se emitieron capítulos todos diferentes entre sí. ¿Le ayudas a averiguarlo?



### Entrada

La entrada comienza con el número de series que Andrés ha estado controlando. Para cada serie, aparece en una primera línea el número de capítulos que Andrés ha apuntado (nunca más de 100.000), seguida de otra línea con los números (entre 1 y  $10^9$ ) de los capítulos emitidos, separados por espacios.

### Salida

Para cada serie, el programa deberá escribir el número máximo de días consecutivos en los que no se emitió ningún capítulo repetido (es decir, que todos los capítulos emitidos en ese periodo son diferentes entre sí).

### Entrada de ejemplo

```
3
6
1 2 3 1 2 3
4
2 2 2 2
5
6 4 5 6 7
```

### Salida de ejemplo

```
3
1
4
```

**Autor:** Alberto Verdejo.

# 30

## Actualización de un diccionario

En este problema un diccionario es una colección de pares clave-valor, donde las claves son cadenas de letras en minúscula y los valores enteros no negativos. El diccionario acaba de ser actualizado. Dado el diccionario antiguo y el nuevo, ¿puedes encontrar cuáles han sido los cambios?



### Entrada

La entrada comienza con el número de casos de prueba que vendrán a continuación. Cada caso consiste en dos líneas, la primera con el diccionario antiguo y la segunda con el diccionario nuevo. Cada diccionario es una secuencia (posiblemente vacía) de pares clave-valor, donde cada clave aparecerá solamente una vez, pero los pares aparecen en cualquier orden.

### Salida

Para cada diccionario actualizado se escribirán los cambios de la siguiente manera. Primero, si se han añadido claves, se escribirá una línea con un '+' seguido de las nuevas claves, en orden creciente. Segundo, si se han eliminado claves, se escribirá una línea con un '-' seguido de las claves que han desaparecido, en orden creciente. Por último, si hay claves cuyo valor ha cambiado, se escribirá una línea con un '\*' seguido de las claves que han sido modificadas, en orden creciente. Si no ha habido cambios, se escribirá 'Sin cambios'.

Tras procesar cada caso se escribirá una línea más con ----.

### Entrada de ejemplo

```
3
a 3 b 4 c 5 f 6
a 3 c 2 d 7 e 9
abc 123 xyz 987
xyz 987 abc 123

a 1 b 2
```

### Salida de ejemplo

```
+ d e
- b f
* c
----
Sin cambios
----
+ a b
----
```

**Autor:** Alberto Verdejo.

# 31

## Referencias cruzadas

Dado un texto organizado por líneas, el *problema de las referencias cruzadas* consiste en producir un listado de palabras ordenado alfabéticamente, donde cada palabra del texto va acompañada de una lista de referencias que contiene los números de todas las líneas del texto en las que aparece la palabra en cuestión, ordenados de menor a mayor.

### Entrada

La entrada consta de diversos casos de prueba. Cada caso comienza con el número  $N$  de líneas que forman el texto (entre 1 y 10.000). A continuación, aparecen  $N$  líneas con las palabras (entre 1 y 30) del texto, donde se han omitido los signos de puntuación o tildes. Las palabras tienen como mucho 10 letras. La entrada termina cuando  $N$  es 0.

### Salida

Para cada caso de prueba se escribirán las referencias cruzadas, donde solamente se tendrán en cuenta palabras con más de 2 letras y no se tendrán en cuenta las mayúsculas. En las listas de números de línea no habrá repeticiones (aunque la palabra aparezca varias veces en la misma línea). Tras procesar cada caso se escribirá una línea más con ----.

### Entrada de ejemplo

```
4
Guerra tenia una parra y Parra tenia una perra
la perra de Parra rompio la parra de Guerra y Guerra aporreo con la porra a la perra
Si la perra de Parra no hubiera roto la parra de Guerra
este no hubiera aporreado con la porra a la perra de Parra
0
```

### Salida de ejemplo

```
aporreado 4
aporreo 2
con 2 4
este 4
guerra 1 2 3
hubiera 3 4
parra 1 2 3 4
perra 1 2 3 4
porra 2 4
rompio 2
roto 3
tenia 1
una 1
----
```

**Autor:** Alberto Verdejo.

# 32

## Eligiendo deporte

La directora del instituto ha decidido que ya es hora de tomar medidas contra el sedentarismo de sus estudiantes, por lo que ha decidido ponerles a practicar deportes. Al comienzo del curso se han colocado una serie de hojas en el tablón de la entrada, cada una con el nombre de un deporte (en letras mayúsculas). Los alumnos deben apuntar su identificador personal (una cadena de letras minúsculas y números, que comienza con una letra) en la hoja del deporte que quieren practicar, solamente en una.



Como responsable de la sección deportiva, tu trabajo ahora consiste en contar cuántos alumnos se han apuntado a cada uno de los deportes propuestos. Algunos alumnos despistados han apuntado su nombre dos veces en la misma hoja. Eso no importa, pero deben ser contados solamente una vez. Otros en cambio han apuntado su nombre en más de una hoja. Estos no deben ser tenidos en cuenta en esta fase, para ninguno de los deportes. Ya se le ocurrirá a la directora un deporte para ellos.

### Entrada

La entrada consiste en varios casos de prueba. Cada caso de prueba contiene la información de una o más hojas, y termina con una línea con la palabra `_FIN_`. Cada hoja comienza con el nombre de un deporte en mayúsculas en una línea, seguida de los identificadores de los alumnos que se apuntaron en esa hoja (posiblemente ninguno), cada uno en una línea.

### Salida

Para cada caso de prueba se escribirá la demanda que ha tenido cada deporte, escribiendo el nombre del deporte y los alumnos que se han apuntado a practicarlo. La lista debe aparecer ordenada de mayor a menor demanda. Si dos o más deportes están empatados en número de alumnos, aparecerán en orden alfabético. Tras la estadística de cada caso de prueba se escribirá una línea con `***`.

### Entrada de ejemplo

```
HOCKEY
peter92
benzu
j97lee
RUGBY
filipo78
peter92
PETANCA
BUCEO
besugo42
besugo42
_FIN_
```

### Salida de ejemplo

```
HOCKEY 2
BUCEO 1
RUGBY 1
PETANCA 0
***
```

**Autor:** Alberto Verdejo.

## Ranking de un concurso de programación

Tu tarea consiste en producir la clasificación final de un concurso de programación conociendo los envíos de soluciones a los distintos problemas que han realizado los equipos participantes. Para la clasificación se tienen en cuenta las siguientes reglas:

- El criterio principal para ordenar a los equipos es el número de problemas resueltos. Cuantos más problemas se resuelvan, mejor será el puesto en la clasificación.
- A igualdad de problemas resueltos, los equipos que hayan tardado menos tiempo se clasifican primero.
- El tiempo utilizado por un equipo es igual a la suma del tiempo necesario para resolver cada problema resuelto correctamente.
- El tiempo necesario para resolver un problema es la suma del tiempo transcurrido desde el comienzo del concurso hasta el primer envío correcto a ese problema, más 20 minutos de penalización por cada envío (a ese problema) incorrecto anterior. Los envíos posteriores al primer envío correcto se ignoran.
- Los problemas que finalmente no se resuelven no penalizan.
- Si dos equipos resuelven el mismo número de problemas en el mismo tiempo, serán ordenados por su nombre de menor a mayor.

### Entrada

La entrada consta de una serie de casos de prueba. En la primera línea aparece el número de casos que vendrán a continuación.

Cada caso consiste en una serie de descripciones de envíos, cada uno en una línea. Cada envío está formado por el nombre del equipo (una cadena de caracteres sin espacios), el nombre del problema (otra cadena de caracteres sin espacios), los minutos transcurridos desde el comienzo del concurso, y el veredicto del juez (AC, WA, TLE, etc). Un envío se considera correcto solamente si el veredicto es AC.

Los envíos de cada caso están ordenados de menor a mayor número de minutos. El último envío está seguido de una línea con la palabra FIN.

### Salida

Para cada caso de prueba se escribirá la clasificación final. Esta contendrá una línea por cada equipo que haya realizado algún envío, donde aparecerá el nombre del equipo, el número de problemas resueltos y el tiempo empleado.

Los equipos aparecen ordenados según las reglas de clasificación, comenzado por el mejor clasificado. La clasificación de cada caso estará seguida por una línea con “----”.

### Entrada de ejemplo

```
1
LosIntrepidos PequalsNP 8 TLE
EquipoA ElSolitario 30 AC
JavaForever ElSolitario 30 RTE
LosIntrepidos PequalsNP 100 AC
LosIntrepidos Sumar2y2 101 AC
Lentorros Sumar2y2 200 AC
LosIntrepidos ElSolitario 210 WA
B_team Horarios 220 WA
FIN
```

### Salida de ejemplo

```
LosIntrepidos 2 221
EquipoA 1 30
Lentorros 1 200
B_team 0 0
JavaForever 0 0
----
```

**Autor:** Alberto Verdejo.



## Problema número 109

# Liga de pádel

Tiempo máximo: 4,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=109>

Los organizadores de las ligas de pádel de Hill Valley no conocen los ordenadores, de manera que siguen anotando los resultados de cada enfrentamiento en un cuaderno, algo increíble teniendo en cuenta que las ligas que manejan pueden tener hasta 2000 parejas distintas.

Al final de la temporada, terminan teniendo tanto lío, que no saben qué pareja es la ganadora de cada categoría. Por si eso fuera poco, durante el invierno, bien debido a las inclemencias meteorológicas o a lesiones de los participantes, algunos de los partidos de cada jornada no se disputan. El problema es que los jugadores no lo avisan, por lo que los organizadores no apuntan nada en el cuaderno. Afortunadamente, se sabe que todas las parejas han llegado a jugar algún partido.

Haz un programa que ayude a aclarar la situación al final de la temporada.

## Entrada

Como entrada, recibirá el nombre de la categoría, seguido de todos los resultados anotados sobre ella. Un resultado se compondrá del nombre de la pareja que juega “en casa”, el número de sets que ha ganado, seguido del nombre de la pareja visitante, y el número de sets ganados, separados por espacio. Tanto los nombres de las categorías como de las parejas estarán compuestos de una única palabra de un máximo de 16 letras.

Cada categoría acabará con la palabra **FIN**.

Por su parte, la entrada terminará con una categoría de nombre **FIN**.

## Salida

La salida del programa indicará, para cada categoría, el nombre de la pareja ganadora. En caso de empate, se mostrará **EMPATE**. Por cada victoria, la pareja se llevará 2 puntos, por cada derrota se llevará 1, y la no asistencia no sumará ningún punto. Recuerda que en pádel no hay posibilidad de que un partido acabe empatado.

Además de la pareja ganadora (si la hay), también indicará el *número* de partidos no jugados al final de la liga. Ten en cuenta que las ligas tienen ida y vuelta.

## Entrada de ejemplo

```
Junior
Buenisimos 3 Malisimos 0
Buenillos 2 Malillos 1
Buenillos 3 Malisimos 0
Buenisimos 3 Malillos 0
Buenisimos 2 Buenillos 1
Malisimos 0 Buenisimos 3
Malillos 1 Buenillos 2
Malisimos 0 Buenillos 3
Malillos 0 Buenisimos 3
Buenillos 1 Buenisimos 2
FIN
Senior
Abuelos 3 Abueletes 0
Abueletes 2 Abuelos 1
FIN
FIN
```

## Salida de ejemplo

Buenisimos 2 EMPATE 0
--------------------------

**Autores:** Pedro Pablo Gómez Martín, Marco Antonio Gómez Martín y Patricia Díaz García.

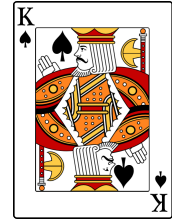
# Abdicación de un Rey

Tiempo máximo: 1,000-3,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=214>

Cuando un rey abdica, su primogénito hereda el trono y debe recibir, en su coronación, un número que lo identificará para la posteridad. La numeración es importante porque, de otro modo, sería difícil diferenciar a reyes con el mismo nombre de una misma dinastía al compartir también apellido.

El resultado es que ante la abdicación de un rey, toca revisar los libros de historia para averiguar su número. ¿Eres capaz de hacerlo tú?



## Entrada

El programa recibirá, por la entrada estándar, múltiples casos de prueba. Cada uno consta de una primera línea con un número indicando la cantidad de reyes de una determinada dinastía. A continuación vendrá, en otra línea, los nombres de todos sus reyes separados por espacio.

Después aparecerán dos líneas más, una con la cantidad de sucesores futuros que hay que numerar (al menos uno), y otra con sus nombres separados por espacio.

Todos los nombres estarán compuestos de una única palabra de no más de 20 letras del alfabeto inglés, y serán sensibles a mayúsculas. Además, se garantiza que en cada caso de prueba no habrá más de 20 nombres de reyes diferentes.

El último caso de prueba, que no deberá procesarse, no contendrá ningún rey en la dinastía.

## Salida

Para cada sucesor de cada caso de prueba se indicará, una línea independiente, el número que le corresponderá. Aunque normalmente se utilizan números romanos, por simplicidad se indicará el número en la notación arábiga tradicional. Después de cada caso de prueba se escribirá una línea en blanco.

## Entrada de ejemplo

```
11
Felipe Carlos Felipe Felipe Felipe Carlos Felipe Carlos Alfonso Alfonso JuanCarlos
3
Felipe Leonor Felipe
12
Carlos Isabel Carlos Jorge Jorge Jorge Jorge Guillermo Victoria Jorge Jorge Isabel
3
Carlos Guillermo Jorge
0
```

## Salida de ejemplo

```
6
1
7

3
2
7
```

**Autores:** Pedro Pablo Gómez Martín, Marco Antonio Gómez Martín y Patricia Díaz García.

# Detectando copiones

Tiempo máximo: 2,000-4,000 s Memoria máxima: 10240 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=338>

La memoria del viejo profesor de matemáticas ya no es lo que era. Hace años, cuando empezó en eso de ilustrar mentes en blanco, no sólo se sabía los nombres y apellidos de todos sus alumnos sino que además era un lince detectando copias de exámenes. Estaba tan seguro de su habilidad que mientras los alumnos intentaban resolver aquellas derivadas e integrales infernales, él se sentaba en la última fila de la clase a dormir sin preocuparse de que la información fluyera entre los estudiantes.



Su habilidad se basaba en su memoria fotográfica: cuando corregía un examen, era capaz de recordar a la perfección si había visto otro examen con exactamente las mismas respuestas o no. Y si lo encontraba, acusaba al segundo de copiar.

Ahora, con tantos años encima, su memoria fotográfica se limita a sólo unos pocos de los últimos exámenes que ha corregido, por lo que el número de copias que detecta se ha reducido drásticamente.

## Entrada

La entrada contiene distintos casos de prueba, cada uno de ellos formado por dos líneas.

En la primera línea aparecen dos números,  $N$  y  $K$ , que indican, respectivamente, el número de exámenes que tiene que corregir el viejo profesor y el número de exámenes que es capaz de recordar ( $1 \leq N \leq 1.000.000$ ,  $1 \leq K \leq 100.000$ ). Tras eso, viene una línea con  $N$  números (entre 1 y 100.000) separados por espacios que representan las respuestas de cada uno de los exámenes. Dos exámenes se consideran copiados si están representados por el mismo número.

## Salida

Para cada caso de prueba se escribirá una línea con dos números separados por un espacio. El primero indicará el número de exámenes copiados mientras que el segundo dará la cantidad de copias detectadas por el profesor, sabiendo que, en el momento de corregir un examen, éste es capaz de recordar únicamente los  $K$  exámenes inmediatamente anteriores.

## Entrada de ejemplo

```
5 1
1 2 1 2 1
5 2
1 2 1 2 1
6 2
1 2 3 1 2 1
```

## Salida de ejemplo

```
3 0
3 3
3 1
```

**Autores:** Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

**Revisores:** Ferran Borrell Micola, Cristina Gómez Alonso, Iván Cantón Sáez y Francesca Tomás Artigues.