

12 de noviembre

¡Santo cielo! Esto de estar en 1955 es increíble. El martes que viene será la histórica tormenta del 12 de noviembre de 1955 que desatará el rayo que terminó con el aún roto reloj de la torre¹.

Tú ya has oído a tus padres contar una y mil veces que la tormenta fue (¡será!) espectacular. Y todo aquello de la gran cantidad de rayos, las pequeñas inundaciones y el pueblo entero patas arriba.

Para evitar seguir escuchando a tu vuelta una y otra vez la misma historia has decidido intentar prevenir algunos de los daños poniendo sobre aviso a los afectados. Has empezado por la gente de la biblioteca y has conseguido que se lleven todos los libros a la planta superior para protegerlos de las inundaciones.

Para luego poder colocar fácilmente de vuelta todos los libros, y dado que de muchos de ellos hay varias copias, los han agrupado por el código. Así todos los libros iguales (con el mismo código) aparecen juntos. Y tras el esfuerzo de ponerlos todos en orden, ahora quieren aprovecharlo para hacer inventario y extraer algunas estadísticas. La primera es saber cuántas copias hay del libro más abundante.

Con tu CPC-6128 y un bucle eso es pan comido, así que te has ofrecido a ayudarles. Te darán la lista con todos los códigos (aparecen repetidos si hay varios libros iguales) y les dirás el número de copias del libro que más veces se repite. De hecho, ahora que lo piensas... en lugar de un bucle, puedes implementar el algoritmo usando *Divide y vencerás* para luego enseñárselo a tus padres y que aprendan un poco más sobre algoritmia.



Entrada

La entrada estará compuesta por distintos casos de prueba cada uno en una línea.

Cada caso de prueba comienza con una línea indicando el número total de ejemplares que han quedado en la biblioteca (hasta 10.000). La línea siguiente contiene la lista de códigos de todos esos ejemplares, separados por espacios. Ten en cuenta que se han agrupado los ejemplares por título, por lo que los códigos que son iguales aparecen siempre juntos.

Los códigos son palabras de hasta 7 letras, números o guiones.

La entrada terminará con un caso de prueba sin códigos que no debe procesarse.

Salida

Por cada caso de prueba se escribirá en una línea un número indicando el número de copias del libro que más veces tenemos. O dicho de otra forma, el número de veces que aparece el código más repetido.

Entrada de ejemplo

```
4
D-123 A-123 B-123 C-123
4
B-345 A-123 A-123 C-123
3
DCDC DCDC DCDC
0
```

Salida de ejemplo

```
1
2
3
```

¹Si tu supieras la importancia que tiene para tí ese rayo en un universo paralelo...

Notas

Ten en cuenta que:

- Para que el ejercicio sea evaluado debe ser primero aceptado por el juez.
- Solo se aceptarán soluciones que utilicen *Divide y vencerás*.
- El cálculo pedido debe hacerse en *una única* función recursiva.
- Para hacer una solución con *Divide y vencerás* te interesará que la función devuelva no uno, sino tres valores: el número de ejemplares del libro más repetido, pero también el número de ejemplares del primer y último libro de la lista.

De esta forma, la cabecera quedará así:

```
void numCopias(string v[], int a, int b,  
               /*out*/ int &libro,  
               /*out*/ int &primerLibro,  
               /*out*/ int &ultimoLibro);
```

Con `libro` siendo el número de copias del libro más repetido, `primerLibro` el número de copias del libro `v[a]` y `ultimoLibro` las copias de `v[b]`.

- Debes indicar (y justificar) *la complejidad* del algoritmo.

Nota

Este ejercicio debe verse en el contexto de la asignatura de Fundamentos de Algoritmia (FAL), FDI-UCM 2019/2020 (prof. Marco Antonio Gómez Martín). Por tanto *no* vale cualquier solución, sino sólo aquellas que utilicen los conceptos de FAL. Es muy posible que se den aclaraciones adicionales en clase a este respecto.