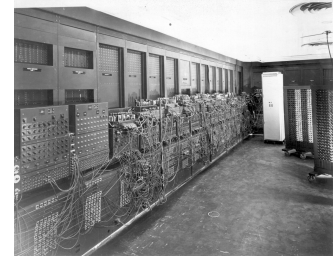


Jugando con el ENIAC

Una de las cosas curiosas de viajar al pasado es que tienes al alcance de la mano cosas que en tu tiempo son parte importante de la historia.

De tus lecturas sobre la historia de la informática sabías que el ENIAC fue un ordenador electrónico digital revolucionario en cuya construcción participó, entre otros, von Neumann. Se dice que cuando en 1955 se apagó había hecho más cálculos matemáticos que los realizados por toda la humanidad anteriormente.

Y mira tú por donde, ahora estás en 1955, hace pocos meses de ese apagado, el ENIAC ha sido donado al pueblo de Hill Valley y está en una nave propiedad del ayuntamiento.



La tentación es muy fuerte, así que quieres colarte con tus padres en esa nave para ver a *la criaturita* y que les pique aún más el gusanillo por la informática.

Lamentablemente la nave se encuentra custodiada por un policía que hace la ronda por las inmediaciones. Has pasado una noche entera registrando cada cuánto tiempo pasa el vigilante por delante de la puerta por la que te quieres colar y estás seguro de que hay un ciclo (de duración incierta). Si descubrieras ese ciclo sabrías de cuánto tiempo dispones la próxima vez que el guardia pase por la puerta...

Entrada

La entrada estará compuesta por distintos casos de prueba, cada uno en dos líneas.

La primera línea contiene un número n con el número de veces que has visto al guardia pasar por delante de la puerta (hasta 1000). La segunda tiene n números indicando el tiempo que ha transcurrido desde que le viste por última vez.

La entrada termina con una línea con un 0.

Salida

Por cada caso de prueba debes indicar el *tamaño del ciclo*, o lo que es lo mismo el número de veces que el guardia pasa por delante de la puerta antes de volver a repetir el recorrido completo.

Se garantiza que en la entrada el ciclo se repite *completo* al menos dos veces. Ten en cuenta, eso sí, que el último ciclo puede no ser completo.

Entrada de ejemplo

```
4
1 2 1 2
5
1 2 1 2 1
4
1 1 1 1
0
```

Salida de ejemplo

```
2
2
1
```

Notas

Ten en cuenta que:

- Para que el ejercicio sea evaluado debe ser primero aceptado por el juez.
- Debes tener al menos las dos funciones siguientes:

1. `bool esCiclico(int v[], int n, int longitud)`; que devuelve `true` si el vector de entrada presenta ciclos de la longitud indicada. Debes indicar precondition, postcondición, complejidad e invariante del bucle.
 2. `int longitudCiclo(int v[], int n)` que, utilizando la anterior, devuelve la longitud del ciclo (si hay varios, el ciclo más pequeño). Para esta función basta con indicar la complejidad.
- NO se admitirán soluciones que modifiquen los datos de entrada ni que requieran memoria auxiliar (haciendo copias del vector).

Nota

Este ejercicio debe verse en el contexto de la asignatura de Estructura de Datos y Algoritmos (EDA), FDI-UCM 2018/2019 (prof. Marco Antonio Gómez Martín). Por tanto *no* vale cualquier solución, sino sólo aquellas que utilicen los conceptos de EDA. Es muy posible que se den aclaraciones adicionales en clase a este respecto.