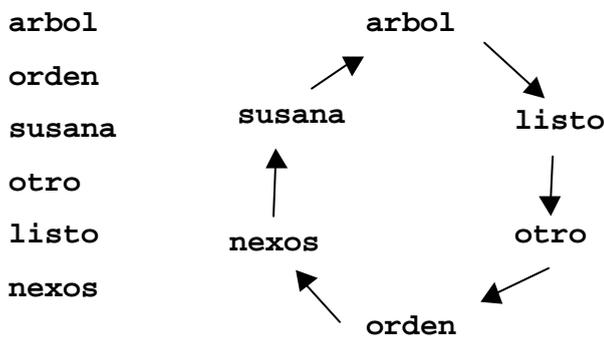


**Descripción del problema**

Dada una lista de palabras, se pide encontrar un ordenamiento circular de las mismas en el cual una palabra B puede suceder a otra A si la primer letra de B es igual a la última letra de A. El ordenamiento debe ser circular (toda palabra debe tener una que la antecede y una que la sucede) y completo (debe incluir a todas las palabras de la lista).

Como ejemplo, se muestra una lista de palabras y un ordenamiento circular posible para las mismas.



Se solicita que, dada una lista de palabras, se informe un ordenamiento circular de las mismas. En caso de no existir un ordenamiento se debe informar "No existe ordenamiento".

**Restricciones**

la cantidad **N** de palabras de la lista  
 $0 < N \leq 9000$   
 el largo de una palabra  $\leq 6$

**Datos de entrada**

Se recibe un archivo **PALABRAS . IN**, ubicado en el directorio actual, que contiene

- En la primera línea, la cantidad **N** de palabras de la lista
- En las **N** líneas siguientes, las palabras que componen la lista de a una palabra por línea.

**Datos de salida**

Se debe grabar en el directorio actual un archivo **PALABRAS . OUT**

- En caso de que no exista un ordenamiento circular, una línea con la frase "No existe ordenamiento"
- En caso de que exista un ordenamiento circular, **N** líneas con las palabras ordenadas, de a una por línea.

**Ejemplo**

Para el caso de la figura, el archivo **PALABRAS . IN** puede contener:

```

6
arbol
orden
susana
otro
listo
nexos
  
```

En cuyo caso el archivo **PALABRAS . OUT** podría contener:

```

susana
arbol
listo
otro
orden
nexos
  
```