

Teclado predictivo

Contribución de Guillermo García

Descripción del problema

La app OIA-2023 para teléfono móvil tiene un muy simple editor predictivo para sus mensajes, que sigue las siguientes reglas:

- No predice la primera ni la segunda palabra del mensaje.
- Para el resto de las palabras del mensaje, predice la palabra más usada a partir de las dos anteriores; esta predicción está guardada en una base de datos del editor.

La base de datos del editor consiste en una lista de N pares de palabras, junto a la siguiente palabra que se predice para el texto cuando las últimas dos palabras escritas hasta el momento coinciden con las del par.

Un ejemplo de base de datos del editor podría ser:

- las uvas \rightarrow verdes
- las flores \rightarrow rosas
- flores las \rightarrow venden
- uvas rojas \rightarrow sabrosas
- uvas verdes \rightarrow acidas
- verdes acidas \rightarrow son

En este ejemplo, si el usuario ya escribió “las uvas verdes”, el editor predictivo le sugiere “acidas” como siguiente palabra, de acuerdo con la regla

uvas verdes \rightarrow acidas, y el usuario podrá decidir entre obedecer la sugerencia o escribir otra palabra manualmente.

El usuario de la app escribe hasta 3 palabras manualmente (la primera, la segunda, y otra más, no se sabe cuál).

Dada la base de datos del editor, debes implementar una función que determine si un mensaje puede ser generado con el editor predictivo de OIA-2023, sabiendo que el usuario escribe hasta **3** palabras (incluyendo las primeras dos) no sugeridas por el editor predictivo.

Descripción de la función

Debes implementar la función `predictivo(r1,r2,r3,texto)`. Sus parámetros son:

- $r1,r2,r3$: Arreglos de N cadenas de caracteres. La i -ésima regla de la base de datos es $r1[i] r2[i] \rightarrow r3[i]$, es decir, si la anteúltima palabra escrita fue $r1[i]$ y la última fue $r2[i]$, el editor predictivo sugiere $r3[i]$ como siguiente palabra.
- `texto`: Cadena de caracteres con el texto del mensaje.

La función debe retornar **1** si es posible generar el texto con el editor predictivo de OIA-2023 de manera tal que el usuario escriba hasta **3** palabras no sugeridas por el editor predictivo, y **0** en caso contrario.

Evaluador

El evaluador local lee de la entrada estándar con el siguiente formato:

- Una línea con la cadena `texto`
- El entero N
- N ternas de cadenas $r1[i], r2[i], r3[i]$

El evaluador local escribe a la salida estándar la respuesta retornada por la función.

Restricciones

- $1 \leq N \leq 100$
- La longitud de cada una de las cadenas $r1[i]$, $r2[i]$, $r3[i]$ está entre **1** y **10** inclusive
- La longitud de la cadena `texto` está entre **1** y **100** inclusive
- Las cadenas $r1[i]$, $r2[i]$, $r3[i]$ solamente contienen letras minúsculas del alfabeto inglés
- La cadena `texto` solamente contiene letras minúsculas del alfabeto inglés o espacios. Además no hay dos espacios consecutivos, ni espacio al comienzo ni al final de la cadena.
- La cadena `texto` contiene al menos **2** espacios (contiene al menos **3** palabras)
- No existen índices i, j distintos entre sí tales que $r1[i] = r1[j]$ y $r2[i] = r2[j]$ (es decir, para el mismo par de palabras hay como máximo una sugerencia).

Ejemplos

Si se invoca al evaluador con:

```
las uvas verdes acidas son mejores
6
las uvas verdes
las flores rosas
flores las venden
uvas rojas sabrosas
uvas verdes acidas
verdes acidas son
```

Para un programa correcto, la salida será:

1

Si en cambio la entrada es:

```
unas uvas rojas baratas
4
las uvas verdes
las flores rosas
flores las venden
uvas rojas sabrosas
```

Para un programa correcto, la salida será:

0

Si es:

```
las uvas rojas son berretas
6
las uvas verdes
las flores rosas
flores las venden
uvas rojas sabrosas
uvas verdes acidas
verdes acidas son
```

la salida será:

0

Y si es:

```
las flores rosas
6
las uvas verdes
las flores rosas
flores las venden
uvas rojas sabrosas
uvas verdes acidas
verdes acidas son
```

la salida será:

1

Subtareas

1. $N = 1$, `texto` contiene exactamente **4** palabras y todas las palabras en `texto` y en la regla tienen longitud **1** (10 puntos)
2. $N = 1$ y todas las palabras en `texto` y en la regla tienen longitud **1** (16 puntos)
3. $N = 1$ (27 puntos)
4. Todas las palabras en `texto` y en las reglas tienen longitud **1** (26 puntos)
5. Sin más restricción (21 puntos)