

## Procesador de textos

Contribución de Agustín Santiago Gutiérrez

### Descripción del problema

Un reconocido procesador de textos incorpora varias funciones convenientes para el procesamiento automático de las palabras que lo componen. Por ejemplo: corrección ortográfica, manejo automático de mayúsculas y minúsculas, sugerencias de sinónimos, etc.

Una de sus funciones le permite corregir automáticamente textos en los cuales existe un error gramatical por utilizar sustantivos en singular donde deberían haber estado en plural. Por ejemplo, si el usuario ingresa una oración como “Los auto de la ciudad eran todos rojos”, el programa le sugiere automáticamente utilizar la palabra “autos” en lugar del singular “auto”.

Con la intención de crear una versión competidora de este exitoso programa, debes implementar una función que dada una lista de  $N$  palabras - que serán todas sustantivos en singular- las transforme a plural automáticamente.

Las reglas completas de formación de plurales en castellano son realmente complejas ya que según la Real Academia Española, por ejemplo, existen ¡17 reglas! que dependen incluso de la acentuación y combinaciones de letras que tenga la palabra y, a veces, dan más de un plural válido posible. Para esta versión del programa sin embargo **nos conformaremos con utilizar las siguientes 4 reglas** que funcionan correctamente en la gran mayoría de los casos:

1. Si el sustantivo **termina en vocal** (a, e, í, o, u) se agrega **s**.
2. Si el sustantivo **termina en s o x**, queda **igual** en plural.
3. Si el sustantivo **termina en z**, se cambia la z por **ces**.
4. Si el sustantivo **termina en otra consonante** se agrega **es**.

La primera regla determina, por ejemplo, los siguientes cambios: pato → patos, lancha → lanchas, iglú → iglús.

La segunda determina que se habla tanto de “el paréntesis” como de “los paréntesis”, “la crisis” y “las crisis”, “el fénix” y “los fénix”, “el martes” y “los martes”.

La tercera corresponde a los cambios pez → peces, y voz → voces.

Finalmente, la cuarta regla es la que corresponde a pared → paredes, limón → limones, rey → reyes. Notar que nunca usaremos esta regla con las palabras que terminan en s o en x dado que para esas aplicamos la regla 2.

Para crear la función, debes asumir que **no existen palabras con plurales irregulares** (como bloc → bloques, espray → espraís, quórum → quórums, chef → chef) sino que, por el contrario, todas se forman utilizando únicamente las reglas anteriores.

Todas las palabras contendrán letras minúsculas (26 letras del alfabeto inglés, sin “tilde” ni “ñ”) y las generadas por la función deben estar también en minúsculas. También se deberá calcular cuántas veces se aplica cada regla al procesar el listado de palabras.

## Descripción de la función

Debes implementar la función

```
pluralizador(N : ENTERO;  
palabras : ARREGLO[N] de PALABRAS;  
cantidadesPorRegla:ARREGLO[4]de ENTEROS)
```

que devuelva en `palabras` las palabras pluralizadas y en `cantidadesPorRegla` la cantidad de veces que se aplica cada regla.

Se recibe en `N` la cantidad de palabras, y en `palabras` un arreglo con las `N` palabras.

Notar que es el mismo parámetro `palabras` en el que se reciben las palabras, el que se debe sobrescribir con las palabras pluralizadas.

El arreglo `cantidadesPorRegla` que se recibe ya tendrá tamaño 4, y únicamente será necesario escribir sus valores.

## Evaluador

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

- Una línea con la cantidad de palabras `N`
- `P` líneas con las palabras a pluralizar

El evaluador devuelve la salida con el siguiente formato:

- `P` líneas con las palabras pluralizadas
- Una línea con 4 números que indican la cantidad de veces que se aplicaron las reglas de 1,2,3 y 4 respectivamente.

## Puntaje

Se obtiene el 10% del puntaje por cada cantidad correcta (10% por contar bien la cantidad de palabras con regla 1, 10% por regla 2, etc), y el 60% restante por convertir correctamente las palabras a plural.

## Cotas

$$1 \leq N \leq 1000$$

Todas las palabras tienen una longitud de entre 1 y 20 letras inclusive.

Solamente contienen letras minúsculas de la a hasta la z, sin ñ ni tildes.

## Ejemplo

Si se recibe `N = 3`,

`palabras[0] = casa`

`palabras[1] = perro`

`palabras[2] = limon`

se debería sobrescribir `palabras` de manera que pase a tener los valores

`palabras[0] = casas`

`palabras[1] = perros`

`palabras[2] = limones`

y además se debería escribir

2 en `cantidadesPorRegla[0]`

0 en `cantidadesPorRegla[1]`

0 en `cantidadesPorRegla[2]`

1 en `cantidadesPorRegla[3]`

ya que se aplicó la regla uno en 2 casos (casa y perro) y la regla cuatro en un caso (limon).

## Subtareas

En casos de prueba por un total de 20 puntos, se recibe una única palabra en el listado, es decir, `N = 1`