

Práctica 3: Duplicar cadenas

Descripción del problema

En este problema de práctica, deberás implementar una función que reciba una cierta cantidad **N** de cadenas de caracteres (“strings”), así como un número **K**, y realice dos tareas:

1. **Duplicate las últimas K palabras del arreglo.**
2. **Calcule las longitudes de las primeras K palabras del arreglo.**

Duplicar “pasa” genera “pasapasa”, duplicar “pepe” genera “pepepepe”, duplicar “abc” genera “abcabc”, etc.

Para esto, la modificación a las últimas **K** debe realizarse **sobre el mismo arreglo que se recibe** (es decir, hay que “pisar” las palabras).

Las longitudes de las primeras **K** deben colocarse en otro arreglo de enteros.

Se ejercitan específicamente:

- Recibir entrada en un arreglo: iterar, saber longitudes
- Strings, conocer longitudes
- Modificar “in-place” un arreglo dado
- Escribir resultados en un arreglo dado como parámetro a la función
- Funciones void

Descripción de la función

Debes implementar la función

```
duplilong(palabras : ARREGLO[N]
de CADENAS, K : ENTERO, longitudes
: ARREGLO[K] de ENTEROS)
```

Donde `palabras` será el arreglo con las **N** cadenas de caracteres que se debe procesar.

La función debe pisar el mismo `palabras` de acuerdo a lo pedido en el enunciado, y guardar en `longitudes` un arreglo con las longitudes pedidas.

Evaluador

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

- Una línea con dos números **N** y **K**, que indican la cantidad de cadenas y el valor **K** del enunciado.
- **N** líneas, cada una con una cadena del arreglo palabras.

El evaluador devuelve dos líneas: en la primera, los valores del arreglo `palabras` al terminar de ejecutar la función `duplilong`. En la segunda, los valores del arreglo `longitudes` al terminar la ejecución de la función.

Cotas

$$1 \leq K \leq N \leq 20$$

Todas las cadenas tendrán una longitud de entre 1 y 10 caracteres inclusive.

Únicamente contendrán caracteres en minúsculas, correspondientes a una de las 26 letras del alfabeto inglés.

Ejemplo

Si se invoca al evaluador con la siguiente entrada:

```
5 3
aaa
bb
c
dd
e
```

Realiza la llamada a la función con las palabras cargadas en el arreglo `palabras`, y un $K = 3$. Con una implementación correcta, el evaluador devolvería:

```
aaa bb cc dddd ee
3 2 1
```