

## Evaluación de Avance

APELLIDO y Nombre: \_\_\_\_\_  
Legajo: \_\_\_\_\_ DNI: \_\_\_\_\_

### Ejercicio 1 – Función QuitarUltimasVocales

**Consigna:** Implementar la función “QuitarUltimasVocales” que cumpla con la post-condición descrita.

**Criterio de evaluación:** Se evaluará el cumplimiento de los requerimientos descritos en la post-condición de la función. Se evaluará también la descomposición funcional que se realice.

**Restricción:** No se puede incluir ninguna librería de manejo de caracteres (hacer todas las funciones auxiliares que necesite).

```
/*
 * Precondición: @texto es una cadena de caracteres
 * Postcondición: Para cada palabra de @cadena se elimina la
 * última vocal de cada una de las palabras en @texto.
 * Ejemplo: para la cadena
 * "Con cada combate te haces más fuerte..."
 * cadena debe ser modificada del siguiente modo:
 * "Cn cad combat t hacs ms fuert"
 */
void QuitarUltimasVocales(char texto[]);
```

---

### Ejercicio 2 – Recursividad

**Consigna:** Implementar la función recursiva “ContarMultiplos” que cumpla con la post-condición dada.

**Criterio de evaluación:** Se evaluará el cumplimiento de los requerimientos descritos en la post-condición de la función. La implementación tiene que ser recursiva.

**Recordar que un número es múltiplo de otro si el resto de la división del número es igual a 0.**

```
/*
 * Precondición: @indiceComienzo e @indiceFin son posiciones validas
 * dentro de @numeros. @numero es un numero positivo
 * Postcondicion: Devuelve la cantidad de numeros en @numeros que son
 * multiplos de @numero que se encuentrn desde @indiceCominezo hasta
 * @indiceFin en @numeros
 */
int ContarMultiplos(int numeros[], int numero, int indiceComienzo, int
indiceFin);
```