



PRÁCTICA 3: Recursividad

Objetivo: Diseñar e implementar subprogramas recursivos. Aprender a realizar trazas de los mismos y comprender cómo se ejecutan (*fases en la ejecución de un subprograma recursivo y utilización del Stack*).

En esta ocasión, vamos a crear un proyecto de programación dividido en varios ficheros según el proceso explicado en la práctica anterior.

- Nombre del proyecto de programación: **cadenasRecursivo**
- Debe incluir los siguientes ficheros:
 - **cadenas.h** (archivo de cabecera)
 - **cadenas.c** (contendrá la implementación de las funciones implementadas)
 - **main.c** (programa principal)

Enunciado: (*léalo completo ANTES de afrontar la resolución de la práctica*)

Vamos a trabajar con dos cadenas de caracteres:

```
char V1[] = "murcielago";  
char V2[] = "ordenador";
```

El programa mostrará en pantalla, de forma repetitiva, el siguiente menú de opciones:

MENÚ DEL PROGRAMA

- 1. Contar vocales en V1 (recursivo)
- 2. Eliminar todas las consonantes de V2 (recursivo)
- 3. Intercalar letras de V1 y V2 en un nuevo string V3 (recursivo)
- 4. Duplicar cada letra de V1 (recursivo)
- 5. Convertir mayúsculas de V1 y V2 a minúsculas (versión iterativa)
- 6. Convertir mayúsculas de V1 y V2 a minúsculas (versión recursiva)
- 7. Ordenar alfabéticamente la cadena V1+V2 (versión recursiva)
- 0. Salir

La salida en pantalla que se obtendrá al ejecutar cada una de las opciones del menú se muestra a continuación:

- Opción 1: (*subprograma recursivo contarVocales()*)

Este subprograma recursivo recorrerá el string V1 y contará cuántas vocales contiene. Se debe considerar vocales tanto en minúscula como en mayúscula (a, e, i, o, u, A, E, I, O, U).

```
V1: murcielago  
Número de vocales en V1: 5
```



- Opción 2: (*subprograma recursivo **eliminarConsonantes()***)

Este subprograma recursivo que reciba la cadena V2 y devuelva una nueva cadena que solo contenga las **vocales**, eliminando todas las consonantes.

```
V2 original: ordenador  
V2 sin consonantes: oeao
```

- Opción 3: (*subprograma recursivo **intercalarCadenas ()***)

Crea un nuevo string V3 intercalando los caracteres de V1 y V2. Si una cadena es más larga que la otra, los caracteres restantes se añaden al final.

```
Cadena intercalada V3: mourrdceinealdaogro
```

- Opción 4: (*subprograma recursivo **duplicarLetras()***)

Crear un subprograma recursivo que reciba la cadena V1 y devuelva una nueva cadena donde cada letra esté duplicada.

```
V1: murcielago  
V1 duplicado: mmuurrciieellaagoo
```

- Opción 5: (*subprograma iterativo **aMinusculas_Iterativo()***)

Convierte todas las mayúsculas de V1 y V2 a minúsculas, modificando los valores en las cadenas originales.

```
char V1[] = "MurcIeLago";          ** suponiendo que estas fuesen las  
char V2[] = "ORdenADOr";          ** cadenas originales
```

```
V1 en minúsculas: murcielago  
V2 en minúsculas: ordenador
```

- Opción 6: (*subprograma recursivo **aMinusculas_Rekursivo()***)

Realiza la misma tarea que la opción anterior pero, en esta ocasión, mediante un subprograma recursivo. La salida en pantalla será similar a la indicada anteriormente.

- Opción 7: (*subprograma recursivo **ordenarCadena()***)

Este subprograma concatenará V1 y V2, y luego ordenará alfabéticamente todos sus caracteres usando un algoritmo recursivo (como quicksort o mergesort adaptado a char[]).

```
Cadena combinada ordenada: aacddeegilmnooorrru
```

Muy importante:

- Las cadenas de caracteres V1, V2 se mostrarán desde el main cuando sea necesario
- El subprograma recursivo/iterativo indicado en cada caso debe realizar únicamente la tarea considerada *contar el n° de vocales, transformar cadena en minúsculas, etc...*
- Los resultados concretos (*n° de vocales, etc.*) se mostrarán desde el main después de que haya finalizado la llamada al subprograma correspondiente.
- Cuide el estilo de programación: comentarios en cada subprograma, etc.



DOCUMENTACIÓN A ENTREGAR

Será preciso realizar una **memoria** organizada de la siguiente forma:

- 1.- Portada: Indicando Título de la práctica, Grupo de prácticas, Autor/es, curso 1º y fecha de entrega
- 2.- La memoria debe contener lo siguiente de cada una de las opciones del menú:
 - a) Para la opción 5 del menú: análisis del problema (*describa brevemente el proceso a llevar a cabo para alcanzar la solución*)
 - b) Para cada una de las opciones del menú restantes (excepto para la opción 0):
 - Análisis del problema
 - Diseño del algoritmo recursivo => identifique caso/s base, caso/s recursivo/s (*ley de recurrencia*)
 - ¿Qué tipo/s de recursividad se implementa?
 - Analice el código fuente de la aplicación e identifique - para cada uno de los subprogramas recursivos implementados - qué operaciones se llevan a cabo en cada una de las distintas etapas que se contemplan en la ejecución de un subprograma recursivo (desplegado – caso base - plegado). Debe incluir en la memoria el código fuente del subprograma recursivo considerado y marcar lo que se ejecuta en cada etapa.
 - c) Incluya en la memoria una traza de ejecución de los subprogramas recursivos considerados en las opciones 1 y 6 del menú (considere el caso concreto que se muestra en los recuadros de ejemplo)
- 3.- Pruebas de ejecución del programa (Capturas de pantalla de ejecución de cada una de las opciones del menú)

ENTREGA DE LA PRÁCTICA

La entrega de la práctica se realizará a través del campus virtual de la asignatura teniendo en cuenta las instrucciones que allí se detallan. Deberá entregarse un fichero comprimido (.zip) que contenga la memoria completa de la práctica y el código fuente del programa implementado (consultad el campus virtual).

- Fecha de publicación de la práctica: 11 de Mayo de 2025
- Fecha de entrega de la práctica: **Domingo, 25 de Mayo de 2025, 22 h.** – Pasada esta fecha no se admitirán nuevas entregas en la convocatoria ordinaria.

EVALUACIÓN:

- * En la evaluación de la práctica se tendrá especialmente en cuenta la documentación interna, la calidad de la estructuración del programa en subprogramas y la adecuación a los contenidos teóricos estudiados en la asignatura.
- * Aunque las prácticas se presentan en parejas, la calificación es individual: Cada alumno debe responsabilizarse de su participación en la elaboración de estas.