

# Ejercicios del Tema 4

## Cadenas de caracteres

NOTA: DEBE SABER RESOLVER ESTOS EJERCICIOS SIN LA LIBRERIA `string.h`

### 1. Espacios por guiones

Escribe un programa que cambie los espacios en blanco por un guión bajo `_` en la frase siguiente:

```
char texto[] = //Usamos \ para escribir varias lineas
"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras euismod\
orci ac porttitor finibus. Curabitur sed tincidunt est, nec mollis\
lorem. Vestibulum facilisis dolor sit amet faucibus ultrices. Sed\
pharetra vel erat et ornare. Duis eu lorem non leo dictum\
egestas. Integer diam arcu, volutpat ut elit vitae, finibus molestie\
risus. Vivamus sagittis commodo risus vel finibus. Vestibulum\
vehicula tortor ut ante tincidunt, non tincidunt turpis sodales. Nam\
orci diam, pulvinar in ante a, dignissim pulvinar magna. Mauris massa\
tortor, fermentum pretium lobortis sed, luctus vitae\
tortor. Suspendisse sagittis augue sit amet risus molestie, sed\
dapibus enim vulputate. Sed tempus risus vel dolor ornare, eget\
imperdiet ligula aliquam. Mauris ac auctor lacus. Quisque suscipit\
interdum rutrum. Sed placerat sit amet urna in vulputate. Nulla\
facilisis mi nisi, vel pulvinar odio auctor posuere.";
```

### 2. Letras al revés

Escribe un programa que imprima al revés una palabra o frase introducida por teclado. (ejemplo: casa roja se convierte en (ajor asac).

### 3. Palabras al revés

Escribe un programa que imprima al revés una frase introducida por teclado. Las letras de cada palabra estarán en orden correcto, pero las palabras deben quedar en orden contrario (ejemplo: Escuela Técnica de Ingenieros se convierte en (Ingenieros de Técnica Escuela).

### 4. Gira palabras

Escribe un programa que procese la frase del ejercicio 1. Como resultado, cada palabra de la frase debe de escribirla al revés.

### 5. Sustituir `string.h`

Realice sus propios programas para suplir a las funciones: `strlen()`, `strcpy()`, `strcat()`, `strcmp()`.

## 6. Cuenta letras

Escribe un programa que analice la frase del ejercicio anterior contando en número de veces que aparece cada vocal. Al final del programa, se debe pintar en el terminal, para cada letra, una línea que contiene el mismo número de asteriscos que veces aparece esa vocal.

Por ejemplo, si el texto fuera:

"Euforia es una palabra con las 5 vocales"

El resultado podría ser:

A: \*\*\*\*\*

E: \*\*\*

I: \*

O: \*\*\*

U: \*\*

## 7. Separa palabras

Escribe un programa que pida al usuario introducir una frase de varias palabras. Pida una nueva frase si ha introducido una sola palabra. Después, imprima cada palabra en una nueva línea.

## 8. Convertir letras

Escribe un programa que después de introducir una palabra convierta alternativamente las letras a mayúsculas y minúsculas y viceversa.

## 9. Buscar nombre

Construya un programa que almacene en un vector los nombres de un grupo de n personas y que, a continuación, busque la posición del vector en la que se almacenó el nombre de una persona que el usuario ingresa. Si dicho nombre no está en el vector, el programa deberá informarlo.

## 10. Mover letras en palabra

Introduzca una palabra por teclado. Realice un programa que permita llevar la última letra al inicio. Repítalo hasta recomponer nuevamente la palabra ingresada. (ejemplo: casa -> acas -> saca -> asac -> casa).

## 11. Encriptar texto

En este ejercicio debes realizar un programa que encripte una cadena de caracteres empleando un número entero como clave de encriptación. En este programa el contenido de la cadena será tu correo electrónico de la UPM, y la clave se determinará a partir de tu número de matrícula. Todos los valores deben estar asignados dentro del código (no uses scanf o gets).

Debes calcular la suma de las cifras de tu número de matrícula. Si tu número de matrícula es impar, el programa debe asignar un signo negativo al resultado. Por ejemplo, si tu número de matrícula es 12345 el resultado de este paso debe ser -15, y si es 43210 el resultado es 10. Éste valor será la clave de encriptación.

Además, define una cadena de caracteres y asígnale tu correo electrónico de la UPM, crea otra cadena para encriptar el correo. El programa recorre la cadena de caracteres sumando a cada carácter la clave de encriptación (con su signo) salvo a los puntos y a la @, que serán copiados sin cambios.

Por ejemplo, si tu correo es «nombre.apellidos@upm.es» y el número de matrícula es 12345, el correo encriptado es `_ScV.RaV]]ZU'@fa^.Vd`.

Para finalizar, el programa debe mostrar en pantalla el correo original y el correo encriptado.