

En todos los ejercicios donde se trabaja con archivos, asuma que se trata de archivos de texto, no binarios.

Los ejercicios se deben resolver trabajando directamente sobre el archivo, no leyendo todo el contenido y trabajando en memoria (es decir, no utilizar `file()` ni `file_get_contents()`), excepto por el ejercicio 1.

1. Dado el nombre de un archivo que contiene líneas de texto, realice procedimientos/funciones para:

- Mostrar por pantalla el contenido del archivo.
- Clonarlo con otro nombre, es decir, crear otro archivo que tenga exactamente el mismo contenido pero con nombre distinto.

Resuelva ambos problemas de dos maneras diferentes:

- Trabajando con los archivos de a una línea por vez (`fgets / fputs`).
- Trabajando con los archivos enteros (`file / file_put_contents`).

2. Dado el nombre de un archivo que contiene una lista de nombres de archivos, devolver un arreglo asociativo que contenga información sobre los archivos en la lista:

- ¿Existe?
- ¿Se puede leer? (sólo si existe)
- ¿Se puede escribir? (sólo si existe)
- Tamaño (sólo si existe)

Por ejemplo, dado un archivo con el siguiente contenido:

```
texto.txt
documento.doc
foto.jpg
```

devolver un arreglo similar al siguiente:

```
array(
  'texto.txt' => array(
    'existe' => true, 'leer' => true,
    'escribir' => true, 'tamaño' => 64
  ),
  'documento.doc' => array(
    'existe' => false
  ),
  'foto.jpg' => array(
    'existe' => true, 'leer' => true,
    'escribir' => false, 'tamaño' => 354915
  )
)
```

3. Dado el nombre de un archivo que contiene números, con un número por línea, codifique un procedimiento/función que busque un número determinado dentro del archivo y retorne el número de línea en que lo encuentra; en caso de no estar, debe retornar -1.

- Suponga que los elementos en el archivo están desordenados.
- Suponga que están ordenados. ¿Cambia en algo la solución anterior? ¿Por qué?

4. Dado un nombre de archivo, invierta su contenido, caracter por caracter.

Por ejemplo, dado un archivo que contiene:

```
| abc  
| def
```

debe quedar:

```
| fed  
| cba
```

5. Dado el nombre de un archivo que contiene números enteros, realice procedimientos para:

1. Recorrer una sola vez el archivo de manera tal que si un número es mayor que el anterior los intercambie.
2. Utilizando el procedimiento anterior, ordene el archivo por burbujeo.

6. Se tienen dos archivos; uno contiene los números de legajos de alumnos junto con sus nombres; el otro contiene números de legajo y las materias en las que se encuentran inscritos. En ambos casos, los datos de un alumno se encuentran separados por punto y coma y los datos de cada alumno se ubican en líneas separadas.

Por ejemplo, si los archivos se llaman 'alumnos.csv' y 'materias.csv':

```
alumnos.csv      | 15;Pérez, Juan  
                  | 28;González, Pedro  
  
materias.csv     | 15;Lógica  
                  | 15;Álgebra  
                  | 15;Análisis
```

1. Cree un arreglo con esos datos, que asocie el número de legajo con el nombre y las materias.

Siguiendo con el ejemplo:

```
| array(  
|   15 => array(  
|     'nombre' => 'Pérez, Juan',  
|     'materias' => array('Lógica', 'Álgebra', 'Análisis')  
|   ),  
|   28 => array(  
|     'nombre' => 'González, Pedro',  
|     'materias' => array()  
|   )  
| )
```

2. Guarde en un único archivo un arreglo con el formato del inciso anterior, de manera que cada línea contenga el número de legajo, el nombre y una materia, separados por punto y coma. En caso que un alumno no esté inscripto en ninguna materia, no lo incluya en el archivo.

Utilizando el mismo ejemplo, el archivo resultante deberá quedar:

```
| 15;'Pérez, Juan';'Lógica'  
| 15;'Pérez, Juan';'Álgebra'  
| 15;'Pérez, Juan';'Análisis'
```

7. Una función es recursiva si tiene invocaciones a sí misma. Teniendo en cuenta que $\text{factorial}(N) = N * \text{factorial}(N-1)$ para $N > 0$ y $\text{factorial}(0) = 1$, escriba una función recursiva que dado un número retorne su factorial.

Implemente también una función que retorne el mismo resultado sin emplear recursión.

8. Realice un procedimiento recursivo que verifique si un arreglo es "capicúa". Para ello considere que un arreglo a de N elementos es capicúa si $a[0] = a[N-1]$ y el "subarreglo" desde 1 a $N-2$ también lo es.

Compare la solución con la iterativa implementada en el Práctico 4.

9. Calcule en forma recursiva el Máximo Común Divisor de dos números enteros no negativos, basándose en las siguientes fórmulas matemáticas:

$$MCD(X, X) = X$$

$$X < Y \Rightarrow MCD(X, Y) = MCD(Y, X)$$

$$X > Y \Rightarrow MCD(X, Y) = MCD(X - Y, Y)$$

Compare esta solución con la iterativa implementada en el Práctico 1.

10. Resuelva el Ejercicio 9 del Práctico 5 (búsqueda en trie) en forma recursiva.

11. Plantee y codifique una estrategia recursiva que grabe el contenido de un archivo en otro, invirtiendo el orden.