

	<b>Facultad de Ingeniería y Ciencias Agrarias</b>	<b>INFORMÁTICA GENERAL</b>
		<b>Practica N° 7</b>
	Archivos	

**Ej. 1:** Dado un archivo cargado con números enteros (*uno por cada línea del archivo*). Desarrollar la función **informe** que reciba por parámetro el nombre del archivo y retorne una lista con los siguientes cuatro valores (valor máximo, valor mínimo, valor promedio y total de líneas de archivo).

Desde el programa principal invocar a la función **informe** y luego imprimir la lista que retorna dicha función.

**Ej. 2:** Se tiene un archivo CSV con valores enteros (cada línea del archivo puede tener uno o más valores separado por una coma en la misma línea)

Desarrollar una función **agregarMedia** que reciba como parámetro el nombre del archivo y agregue debajo de cada línea del archivo el promedio de los valores que contiene dicha línea.

Ayuda: Para realizar el proceso, cargar inicialmente todo el archivo como una lista y luego reescribir el archivo con lo solicitado.

**Ej. 3:** Se tiene un archivo (de texto) que contiene un texto muy extenso, el cual corresponde a un libro de cuentos. Desarrollar una función **frecuenciaPalabra** que reciba el nombre del archivo y genere un segundo archivo (llamado *frecuencias.csv con formato CSV*) que contenga un listado de las palabras que aparecen en el texto y la cantidad de veces que aparece dicha palabra. Es decir una línea con el par (palabra, cantidad), donde la palabra no deberá repetirse.

Ayuda: Puede utilizar una colección para procesar la información.

Desde el programa principal invocar a la función **frecuenciaPalabra**, pasándole por parámetro el nombre del archivo, luego imprimir en pantalla el contenido del archivo *frecuencias.csv* que genere la función

**Ej. 4:** Desarrollar una función **cabecera** que recibe cuatro parámetros: nombre de archivo (**arch**), cantidad de palabras (**cant**), longitud de palabra mínima (**pmin**), y longitud de palabra máxima (**pmax**). La función deberá leer el archivo **arch** pasado por parámetro e imprimir por pantalla las primeras **cant** palabras que tengan longitud comprendida entre el valor mínimo **pmin** y el valor máxima **pmax** pasados por parámetro.

Desde el programa principal invocar a la función **cabecera** pasándole los parámetros requeridos.

**Ej. 5:** Desarrollar la función **cabecera2**, que será una modificación de la función anterior **cabecera**, para que el listado de palabras que se obtenga se vuelque en un nuevo archivo de texto llamado *“resultado.csv”*. Este archivo debe contener una palabra por línea y las mismas no deben estar repetidas.

	<b>Facultad de Ingeniería y Ciencias Agrarias</b>	<b>INFORMÁTICA GENERAL</b>
		<b>Practica N° 7</b>
	Archivos	

Desde el programa principal invocar a la función **cabecera2** pasándole los parámetros requeridos y luego imprimir el contenido archivo resultado.csv generado por la función.

- Ej. 6:** Desarrollar la función **generadora** (*generadora de texto*) que recibe tres parámetros:
- ori:** Parámetro (string) que contiene el nombre de un archivo de texto, con un listado de palabras (una palabra por línea). Ej: "origen.txt"
  - dest:** Parámetro (string) que contiene el nombre de un archivo de texto, donde se deberá escribir lo procesado por la función. Ej: "destino.txt"
  - cant:** Parámetro (entero) que representa una cantidad.

La función deberá escribir en **dest** una cantidad **cant** de palabras extraídas en forma aleatoria del archivo **ori**. Las palabras no deben repetirse.

Desde el programa principal solicitar al usuario el ingreso de los parámetros, luego invocar a la función **generadora** pasándole los parámetros correspondientes. Por último mostrar en pantalla el archivo de destino generado.

- Ej. 7:** Desarrollar un programa que muestre por pantalla el siguiente menú:

*Ej: de presentación de menú en pantalla*

1. AGREGAR REGISTRO
2. ELIMINAR REGISTRO
3. BUSCAR REGISTRO
4. ORDENAR ARCHIVO POR
5. MOSTRAR ARCHIVO
6. SALIR

Ingrese el valor de la opción:

El programa debe permitir agregar, eliminar, buscar, ordenar archivo y mostrar archivo. El archivo **persona.csv** es CSV, con registros de personas compuesto por tres campos separados por coma que contiene (**nombre**, **apellido**, **dni**). Cada línea del archivo contendrá un registro único por persona cuya clave (*campo no repetido*) es el **dni**.

Para cada una de las opciones del menú se deberá invocar a una función que resuelva lo solicitado. A continuación el detalle de las características de cada opción del menú:

**AGREGAR REGISTRO:** Agrega un registro nuevo (**nombre**, **apellido**, **dni**) al archivo **persona.csv**. Permite el ingreso de datos al usuario desde el teclado. No se debe permitir agregar **dni** repetidos.

	<b>Facultad de Ingeniería y Ciencias Agrarias</b>	<b>INFORMÁTICA GENERAL</b>
		<b>Practica N° 7</b>
	Archivos	

**ELIMINAR REGISTRO:** Eliminar un registro existente del archivo **persona.csv** a partir de un **dni** el cual deberá ser solicitado al usuario.

**BUSCAR REGISTRO:** Debe permitir búsquedas por **dni** o **apellido**. Se solicitará el ingreso al usuario del **dni** o **apellido** a buscar. Se deberá imprimir en pantalla los valores de los registros que coincidan con la búsqueda especificada.

**ORDENAR ARCHIVO POR:** Modifica el archivo **persona.csv** dejándolo ordenado por el número de campo ingresado. Se solicitará el ingreso de un número entre 1 y 3 el cual indicará por cual campo deberá ser ordenado.

**MOSTRAR ARCHIVO:** Imprime el archivo **persona.csv** en pantalla.

**SALIR:** Sale del programa.

*Ayuda:* Es recomendable pensar a cada opción del menú como un problema en sí mismo, por tanto es aconsejable realizar una función con los parámetros necesarios y retorno (si necesitara) para resolver cada caso.

**Ej. 8:** Una consultora cuenta con información de sus bases de datos, que son los siguientes tres archivos (que no tienen ningún criterio de ordenamiento):

Archivo	Archivo Estructura de datos	Ejemplo de contenido
provincias.txt (archivo CSV)	ID_provincia(entero) nombre(cadena de caracteres) ID_pais	1,Buenos Aires,1 8,Jujuy,1 3,Rio Grande,2 9,San Diego,2
localidades.txt (archivo CSV)	ID_localidad (entero) nombre (cadena de caracteres) ID_provincia (entero) superficie (entero) poblacion (entero)	1,Adolfo Alsina,1,14222,12 12, San Salvador,8,1234,32 81,La Paz,9,887,999 53,Lujan,1,1234,10 71,Ezeiza,1,220,101
países.txt (archivo CSV)	ID_pais (entero) nombre (cadena de caracteres) ID_idioma (entero)	1, Argentina, 1 2, Brasil, 5

	<b>Facultad de Ingeniería y Ciencias Agrarias</b>	<b>INFORMÁTICA GENERAL</b>
		<b>Practica N° 7</b>
	Archivos	

Los nombre subrayados significa que son campos clave: Es decir que su contenido dentro del archivo es único (no se repite)

- 8.1. Realizar la función **poblacion(ID\_provincia)** que reciba como parámetro el ID de la provincia e imprime por consola la cantidad habitantes para dicha provincia, indicando además el nombre de la provincia.

Ejemplo de salida: Para **poblacion(1)**, la salida debería ser -

Buenos Aires: 123 Habitantes

- 8.2. Realizar una función **localidadMaxima()** que imprime por consola la localidad con mayor cantidad de habitantes indicando, cantidad de habitantes, nombre de la localidad, nombre de la provincia y nombre del país.

Ejemplo de salida:

Población: 999  
Localidad: La Paz  
Provincia: San Diego  
Pais: Brasil