



Laboratorio 8 - Archivos

Programas de ejemplos

Obtener los archivos fuentes proveídos por el profesor y realizar las siguientes pruebas con cada código fuente.

Lectura y escritura de archivos

1. **1_abrir_cerrar_archivo.c:**
 - a. El programa intenta abrir un archivo. Si no lo logra, imprime un mensaje pero el programa continua con la ejecución.
 - b. Verificar el comportamiento del programa.
 - c. Realizar una modificación para evitar que el programa continúe con la ejecución si no encuentra el archivo.
2. **2_1_lectura_archivo.c:**
 - a. El programa abre un archivo e imprime el contenido realizando una lectura caracter a carácter usando la función *fgetc*.
 - b. La impresión difiere un poco del contenido del archivo. Realizar los cambios necesarios para que la impresión sea realizada como el contenido se ve en el archivo.
3. **2_2_lectura_archivo.c:**
 - a. El programa abre un archivo e imprime el contenido realizando una lectura línea por línea usando la función *fgets*.
 - b. ¿Se tiene en cuenta la cantidad de caracteres por línea?
4. **2_3_lectura_archivo.c:**
 - a. El programa abre un archivo e imprime el contenido realizando una lectura línea por línea usando la función *fgets*.
 - b. Analizar el comportamiento del programa.
5. **2_4_lectura_archivo.c:**
 - a. El programa abre un archivo e imprime el contenido realizando una lectura línea por línea usando la función *fscanf*.
 - b. Analizar el programa y determinar la diferencia con los programas anteriores.
6. **3_1_escritura_archivo.c:**
 - a. El programa ingresa al archivo todos los caracteres leídos desde el teclado utilizando la función *fputc*.
 - b. Analizar el funcionamiento del programa.
7. **3_2_escritura_archivo.c:**
 - a. El programa ingresa una cadena de caracteres al archivo utilizando la función *fputs*.
 - b. Analizar el funcionamiento del programa.
8. **3_3_escritura_archivo.c:**



- a. El programa ingresa diferentes tipos de datos al archivo utilizando la función *fprintf*.
- b. Analizar el comportamiento del programa.

Partir un archivo en n partes

1. **partir_archivo.c:**
 - a. El programa toma un archivo y lo parte en n archivos.
 - b. Analizar el comportamiento del programa.

Ejercicios de nivel inicial

1. Escribir un programa que lea caracteres de un archivo llamado *alumnos.txt* si existe. Si el archivo existe, el programa deberá contar la cantidad de líneas que tiene el archivo e imprimirla en pantalla.
2. Escribir un programa que pida al usuario el nombre del archivo a leer. El programa deberá imprimir en pantalla la cantidad de vocales que tiene el archivo. Tener en cuenta las vocales con acento.
3. El archivo *ventas.txt* contiene las ventas de un periodo de tiempo de un comercio. Leer los montos de las ventas almacenados en el archivo e imprimir el total. Los valores de las ventas están separados por espacios en blanco. Los valores de las ventas no contienen puntos.
4. Escribir un programa que copie los datos de un archivo a otro archivo. Se deben solicitar los nombres de los archivos de entrada y de salida.

Ejercicios de nivel intermedio

1. (Usado en Lenguaje C) Escribir un programa que declare la siguiente estructura

```
struct alumno
{
    char nombre[25];
    char apellido[25];
    int nro_documento;
    int porcentaje_primer_parcial;
    int porcentaje_segundo_parcial;
};
```

El programa deberá leer los datos del archivo *alumnos.txt* y cargarlos en un vector de alumnos. Los datos de un alumno se encuentran en cada fila del archivo. Una vez finalizada la carga se deberá imprimir todos los datos del vector.



2. (Usado en Lenguaje C) Modificar la estructura del ejercicio anterior para agregar los siguientes datos del alumno:
- promedio_ponderado
 - porcentaje_examen_final
 - calificacion

El programa además de leer los datos del ejercicio anterior, deberá leer el porcentaje del examen final que también debe encontrarse en el archivo. El programa deberá calcular el promedio ponderado (promedio de los porcentajes de los parciales). Luego deberá calcular la calificación utilizando la siguiente fórmula: $\text{promedio ponderado} * 0.4 + \text{nota final} * 0.6$. Para obtener la calificación se deberá utilizar la siguiente escala:

Porcentaje calculado	Calificación
0-59%	1
60-69%	2
70-79%	3
80-93%	4
94-100%	5

El programa deberá generar el archivo **calificaciones.txt** con la calificación de todos los alumnos. En el archivo se debe indicar el número de documento del alumno y su calificación.

3. El archivo *puntajes.txt* contiene los puntajes obtenidos por los alumnos en el examen de programación. Los alumnos que obtuvieron un puntaje mayor o igual a 60, superaron la materia. Los alumnos que obtuvieron un puntaje menor a 60 deberán rendir un examen recuperatorio. Leer el archivo y separar los puntajes en dos archivos. Por un lado el archivo *pasaron.txt* contendrá los puntajes de los alumnos que pasaron. Por el otro lado, el archivo *no_pasaron.txt* contendrá los puntajes de los alumnos que no pasaron.
4. Complementar el ejercicio anterior para generar un tercer archivo llamado *resumen.txt* que deberá tener la siguiente información:

****RESUMEN DE ALUMNOS QUE PASARON****

Cantidad de alumnos que pasaron: xx

Promedio de puntajes de alumnos que pasaron: xx

Puntaje mayor: xx

****RESUMEN DE ALUMNOS QUE NO PASARON****

Cantidad de alumnos que no pasaron: xx

Promedio de puntajes de alumnos que no pasaron: xx



Puntaje menor: xx

5. Se realizó una encuesta para conocer al candidato presidencial con mayor intención de votos. Existen tres candidatos. El archivo *encuesta.txt* contiene las mediciones realizadas. Cada línea del archivo contiene el número del candidato elegido por la persona encuestada (1, 2 o 3). Escribir programa que lea el archivo e imprima en pantalla el siguiente resumen:

**** Intenciones de voto ****

Intenciones de voto candidato 1: xx (yy%)

Intenciones de voto candidato 2: xx (yy%)

Intenciones de voto candidato 3: xx (yy%)

6. Una empresa de distribución de agua almacena sus lecturas mensuales de los medidores de agua de los clientes en un archivo con el siguiente formato:

numero_cliente, lectura_anterior, lectura_actual

Todos los datos son numéricos. Tanto la lectura anterior como la actual están medidas en m3.

Escribir un programa que permita calcular la cantidad de m3 consumidos por el cliente en un mes y permita determinar la cantidad adeudada por cada cliente. El programa deberá realizar los cálculos y generar un archivo con el siguiente formato:

numero_cliente, deuda

El costo del agua es de 10.000 por m3

7. La municipalidad de Asunción cuenta con un archivo que contiene la deuda de los clientes de diferentes años en concepto de impuesto inmobiliario. Los datos dentro del archivo tienen el siguiente formato:

id_cliente, año, deuda

Escribir un programa que genere un archivo que contenga todos los clientes con la deuda total.

8. El centro de salud de la ciudad de San Bernardino cuenta con datos históricos de las consultas realizadas en dicho centro de los últimos 3 años. El director de dicho centro está necesitando determinar el porcentaje de consultas de niños y adultos con el fin de optimizar la compra de medicamentos en los próximos años. Los datos dentro del archivo tienen el siguiente formato:

año_consulta, tipo_paciente, requirió_medamento



donde

año_consulta: un numérico que indica el año en el cual se realizó la consulta.

Tipo_paciente: N, niños; A, adultos.

Requirió medicamento: S, sí; N, no.

Escribir un programa que genere un informe con el siguiente formato:

**** INFORME DE CONSULTAS MÉDICAS ****

Cantidad consulta adultos: xx

Cantidad consulta niños: xx

Año con más consultas: xxxx

Ejercicios de nivel avanzado

1. Una empresa que vende ropas para bebés almacena en archivos los rankings de los productos vendidos. Las ventas son almacenadas por año y agrupadas por productos para niños y niñas. Por ejemplo, el archivo *ranking_ventas_2017.txt* contiene los siguientes datos en sus primeras líneas:

1	gorritos	24564	pijama	18547
2	chaqueta	20478	medias	17524
3	pijama	15874	guantes	15748

En el ranking 1, para los varones estuvo el gorrito con 24564 ventas y el pijama para las niñas con 18547 ventas. Es decir, a la izquierda se encuentra el producto de los niños y la derecha el de las niñas en una posición del ranking.

Escribir un programa que pida al usuario el año (existen tres archivos, uno por año), el sexo (niño o niña) y el nombre del producto. El programa deberá indicar el ranking del producto en ese año.

2. Modificar el programa anterior para que pida al usuario el nombre del producto. El programa deberá indicar la posición del ranking de dicho producto en cada año sin discriminar si el producto pertenece a niños o niñas.
3. Una empresa de venta de calzados tiene sus registros de ventas de un año en un archivo. Cada registro dentro del archivo tiene el siguiente formato:

numero_vendedor, mes, total_visitas, total_ventas

donde

numero_vendedor: es un número que va de 1 a 10.

mes: es un número que va de 1 a 12



total_visitas: es la cantidad de visitas que realizó el vendedor en un mes determinado.

total_ventas: es el total de ventas que logró el vendedor en un mes determinado.

El programa deberá generar el siguiente informe:

*** Resumen Anual ***

Total de ventas en el año: xx

Total de visitas en el año: xx

Porcentaje de ventas respecto a visitas: xx%

Vendedor con más ventas: x

Vendedor con menos ventas: x

Mes con más ventas: x

Mes con menos ventas: x

4. Una empresa tiene almacenado los contactos de sus clientes en tres archivos diferentes. Los tres archivos tienen la misma estructura:

Nombre/Apellido/Número de teléfono

Escribir un programa que lea los tres archivos y genere un archivo consolidado sin repetir los clientes.