	UNIVERSIDAD TECNOLÓGICA NACIONAL ***Facultad Regional Tucumán Departamento*** **SISTEMAS** ***Cátedra: Algoritmo y Estructura de Datos*** ***Ciclo 2019 / 2020***	**TRABAJO PRACTICO** **Archivo de Texto**

EJERCICIOS RESUELTO

DE ARCHIVO DE TEXTO

EJEMPLO 01

OBJETIVO: *Manejo de **Archivos texto**. Ingreso de Datos, Listados, copiar datos del archivo a un vector. Ordenar el vector y transferir los datos ordenados al archivo.*

ENUNCIADO:

Se desea guardar en un ***archivo de texto***, los el apellido y nombre de personas de las personas que forman parte de la empresa.

SE PIDE:

1- Haciendo uso de una función sin tipo ingresar el apellido y nombre de **N personas**. Teniendo en cuenta que:
   - Cada persona debe ocupar una línea diferente en el archivo.

2- Haciendo uso de una función sin tipo **Realiza el listado** de los datos guardados en el archivo de texto.

3- Haciendo uso de una función sin tipo **Ordena** el archivo de texto según la longitud del apellido y nombre de cada persona de menor tamaño a mayor tamaño**. Para ello debes pasar los datos a un vector y luego ordenar el vector, teniendo en cuenta que se ordena por su tamaño o longitud.**

4- **Realiza el listado** de los nombres ordenados por su tamaño. Y luego transferir los datos ordenados al archivo. Luego usar el block de nota o algún editor de texto para ver el archivo.

CODIFICACIÓN

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/*----- Declarar un tipo de dato nuevo -----*/
typedef char nombreDePersona[50];   //cadena de 50 caracteres.


/*-----     Declaración Prototipos de Funciones -----*/
void cargarNombres(int N);
void listarArchivo(void);
void ordenarArchivo(nombreDePersona vectorOrdenadoPorTamanio[100], int N);
void listarGuardarOrdenado(nombreDePersona vectorOrdenadoPorTamanio[100], int N);
```

```
/*--------------------        Bloque Principal  -------------------------------
Tipo de Función    : Sin Tipo.
Devuelve           : Ningún Valor.
Parámetros         : Ninguno.
Descripción        : Bloque principal, comienza a ejecutarse el programa.
--------------------------------------------------------------------------------*/
main()
{
   int cantidadDeNombres=0;
   nombreDePersona vectorOrdenadoPorTamanio[100];
   printf("Cuantos Nombres Desea Ingresar: ");
   scanf("%d", &cantidadDeNombres);

   cargarNombres(cantidadDeNombres);
   listarArchivo();
   ordenarArchivo(vectorOrdenadoPorTamanio, cantidadDeNombres);
   listarGuardarOrdenado(vectorOrdenadoPorTamanio, cantidadDeNombres);
}


/*************************************************************************
PUNTO 01:
Tipo de Función    : Sin Tipo.
Devuelve           : Ningún    Valor.
Parámetros         : Entero    indicando la      cantidad de datos a Guardar.
Descripción        : Ingresar apellido y   nombre de N personas.
*************************************************************************/
void cargarNombres(int N)
{
   nombreDePersona apellidoNombre;

   FILE *fichero;

   fichero = fopen("empleado.txt", "w");
   printf("Ingrese los nombres a Registrar\n");
   for(int i=0; i<N; i++){
       printf("\n\t Apellido y Nombre %d : ",i+1);
       _flushall();
       gets(apellidoNombre);
       fprintf(fichero, "%s\n", apellidoNombre);
   }
   fclose(fichero);
   printf("\nF i n  de  la  C a r g a      \n");
   system("PAUSE");
   system("CLS");
}     //Fin de la función cargarNombres()


/*****************************************************************
Tipo de Función: Sin Tipo.
Devuelve    : Ningún Valor.
Parámetros  : Ninguno.
Descripción : Realiza el listado de los datos (apellidos y nombres)
                    guardados en el archivo de texto
*****************************************************************/
void listarArchivo(void)
{
   char apellidoNombre[50];

   FILE *fichero;
   fichero = fopen("empleado.txt", "r");

   fgets(apellidoNombre, 50, fichero);

   printf("\nI N I C I O   D E L   L I S T A D O\n");
```

FILE: Crea una variable puntero a fichero (archivo)

fopen; asocia la variable a un archivo en disco, el cual puede ser abierto como nuevo, solo lectura o modificación según el caso. El tipo de la máscara o formato establece la forma de apertura del archivo.

fprintf: Permite transferir el contenido de una variable, en este caso un texto, a un archivo. El "\n" permite colocar detrás de la cadena ingresada un salto de línea.

fclose: Cierra el archivo asociado a la variable fichero.

fgets: Permite leer una cantidad de caracteres de un archivo en disco y guardar en una variable o un vector de caracteres.
En este caso lee 50 caracteres del archivo, asociado a la variable fichero, y lo guarda en apellidoNombre.

```
   while (!feof(fichero))
   {
       puts(apellidoNombre);
       fgets(apellidoNombre, 50, fichero);
   }
   fclose(fichero);
   printf("\nF I N DEL   L I S T A D O.....\n");
   system("PAUSE");
   system("CLS");
}  //Fin de la función listarArchivo().

/**********************************************************************
Tipo de Función    : Sin Tipo.
Devuelve           : Ningún Valor.
Parámetros         : Entero, indica la     cantidad de datos guardados.
Descripción        : Ordena    el archivo  de texto, según la longitud del
                           apellido y nombre de cada persona.
**********************************************************************/
void ordenarArchivo(nombreDePersona vectorOrdenadoPorTamanio[100], int N)
{
   nombreDePersona cadAux;
   int i=0;

   FILE *fichero;
   fichero = fopen("empleado.txt", "r");

   fgets(vectorOrdenadoPorTamanio[i], 50, fichero);
   while (!feof(fichero)){
       i++;
       fgets(vectorOrdenadoPorTamanio[i], 50, fichero);
   }

   fclose(fichero);

   for(int j=0; j<N; j++) // Ordena el vector.
   {
      for(i=0; i<N-1; i++)
      {
        if(strlen(vectorOrdenadoPorTamanio[i])>strlen(vectorOrdenadoPorTamanio[i+1]) )
        {
             strcpy(cadAux, vectorOrdenadoPorTamanio[i+1]);
             strcpy(vectorOrdenadoPorTamanio[i+1], vectorOrdenadoPorTamanio[i]);
             strcpy(vectorOrdenadoPorTamanio[i], cadAux);
        }          }
    } //fin del ordenamiento del vector.

       fclose(fichero);
       printf("\n S E   T E R M I N O   D E   O R D E N A R\n");
       printf("==========================================\n");
       system("PAUSE");
       system("CLS");
}  //Fin de la función ordenarArchivo().


/**********************************************************************
PUNTO 04:
Tipo de Función    : Sin Tipo.
Devuelve           : Ningún    Valor.
Parámetros         : Vector con los datos ordenados y cantidad de nombres.
Descripción        : Lista los elementos del vector y luego los guarda en el archivo.
**********************************************************************/
void listarGuardarOrdenado(nombreDePersona vectorOrdenadoPorTamanio[100], int N)
{
   FILE *fichero;
   fichero = fopen("empleado.txt", "w");
   system("CLS");
   printf("\n\n\nLISTADO DE MENOR A MAYOR LOGITUD\n\n");
```

Ciclo que se repite hasta llegar al final del archivo empleado.

Muestra y lee otros 50 caracteres que son guardado en la variable apellidoNombre.

Ciclo para transferir los datos almacenados o guardados en un archivo hacia un vector.

foef: función que devuelve verdadero cuando se llegó al final del archivo o falso cuando no se está en el final del archivo.

Bloque que ordena los nombres del vector de menor a mayor teniendo en cuenta el largo de cada nombre

```
   for(int i=0; i<N; i++)
   {
      printf("%s", vectorOrdenadoPorTamanio[i]);
      fprintf(fichero, "%s", vectorOrdenadoPorTamanio[i]);
   }
   fclose(fichero);
   system("PAUSE");
   system("CLS");
}
```

EJEMPLO 02

OBJETIVO: *Manejo de **Archivos texto**. Ingreso de Datos, eliminación de la última frase ingresada.*

ENUNCIADO:

Crea un programa que vaya leyendo las frases que el usuario teclea y las guarde en un fichero de texto llamado "registroDeUsuario.txt". Terminará cuando la frase introducida sea "fin" (esa frase no deberá guardarse en el fichero).

CODIFICACIÓN

```
#include <stdio.h>
#include <string.h>

main()
{
   FILE* ptFichero;
   char fin[]="fin";  //Frase con la que debe finalizar el ingreso. y Además no se debe guardar
   char frase[60];  //Supondremos que las frases ingresadas no tendrán un tamaño mayor a 60.

   ptFichero = fopen("registroDeUsuario.txt", "w");
   printf(" PROGRAMA para ESCRIBIR FRASES.\nCuando quiera salir,"
          "escriba la palabra fin.\n\n");
   do
   {
      puts("\nEscriba una FRASE:\n(o fin). \n");
      gets(frase);
      if (strcmp(frase, fin) == 0)
         break;            //Termina el proceso de ingreso evitando que se guarde la palabra "fin"
      fprintf(ptFichero, "%s\n", frase);
   }while (strcmp(frase, fin) != 0);

   fclose(ptFichero);
}
```

DESAFIO

Teniendo en cuenta lo visto en el ejercicio 01 realizar una función que reciba la variable punto a registro y realizar el listado del archivo mostrando las frases ingresadas.

EJEMPLO 03

OBJETIVO: *Manejo de **Archivos texto**. Leer un archivo ya creado cuyo nombre está en una variable de tipo string – Mostrar una a una las letras.*

ENUNCIADO:

Se tiene un archivo (creado con el bock de nota) de nombre Promedio.txt donde esta registrados los nombres de un grupo de alumnos y el promedio de obtenido.

Ejemplo:

Bustos Juan Carlos 8
Algañaraz Mariano 7

Solo se pusieron dos como ejemplo del contenido del archivo, pero pueden ser muchos más.

CODIFICACIÓN

```
#include <stdio.h>
#include <stdlib.h>

main()
{
  system("color 4f");      //ponemos color rojo al fondo y letras blancas.
  FILE *archivo;
  char unaLetra;

  char miTxt[] = "promedio.txt";

  archivo = fopen(miTxt,"r");

  if (archivo == NULL)
   {
      printf("Error de apertura del archivo. \n\n");
   }else{
     printf("El contenido de %s es:\n\n", miTxt);

     while (feof(archivo) == 0)
     {
        unaLetra = fgetc(archivo);      //Lee un character y le asigna a la variable unaLetra
        printf("%c",unaLetra);          //Muestra el contenido de la variable unaLetra
     }
   }

   fclose(archivo);

   printf("\n\nFin del archivo.\n\n");
   system("pause");
}
```

EJEMPLO 04

OBJETIVO: *Manejo de archivo guardado en disco, contando cuantas palabras, blanco, otros,.*

ENUNCIADO:

Dado un archivo de texto almacenado en disco, determinar cuantas palabras tiene el archivo, cuantos caracteres tiene el archivo, cantidad de espacios en blancos, cuantas líneas tiene,

CODIFICACIÓN

```
#include <stdio.h>
#include <stdlib.h>

main()
{
   char c;
   int contCaracteres, contLineas, contEspacios, contPalabras;    //Declaro los contadores
   int e=0;
   contCaracteres=contLineas=contEspacios=contPalabras=0;   //Inicializo los contadores con cero
   FILE *ft;

   ft=fopen("texto.txt","r"); //abre un archivo ya existente llamado texto.txt en la misma carpeta del programa, solo lec tura.
```

El archivo debe ser creado con bloc de nota e ingresar varias líneas de texto para probar el programa.

```
   while((c=fgetc(ft))!=EOF)
   {
       putchar(c);
       contCaracteres++;                //cuenta caracteres
       if(c==' ')
         contEspacios++;                //cuenta espacios
       if(c=='\n')
         contLineas++;                  //cuenta lineas
       if(c==' ' || c=='\n')
         contPalabras++;                //cuenta palabras
   }

   fclose(ft);

   printf("\n");
   printf(" cantidad de caracteres = %d\n",contCaracteres);
   printf(" cantidad de espacios = %d\n",contEspacios);
   printf(" cantidad de lineas = %d\n",contLineas);
   printf(" cantidad de palabras = %d\n",contPalabras);

   printf("   Fin de programa ... Enter");
   getchar();
}
```