

Programación y Algoritmos

Parte 1: Lenguaje C

Tarea #6

Fecha de entrega: 1/10/2021.

1. Crear un archivo de estructuras con la siguiente información:

```
struct escuela {
    char grupo; // A, B, C, D, E, F
    char turno; // Matutino o Vespertino
}
struct estudiante {
    char *nombre;
    char *calif_promedio; // Formato USA {A+ > A > A- > B+ > B > B- > C}
    int edad;
    struct escuela *E;
};
```

Leer el archivo y desplegar un menú de opciones que se realicen de manera indefinida:

- 1) Imprimir archivo
- 2) Ordenar por nombre:
- 3) Ordenar por Edad;
- 4) Ordenar por Promedio
- 5) Número de estudiantes por grupo
- 6) Número de estudiantes por turno
- 7) Baja estudiante
- 8) Alta estudiante
- 9) Invertir archivo

Desplegar el resultado de cada opción seleccionada en pantalla dentro de la función main(), no dentro de las función de ordenamiento (esto es, la función main() debe conocer los cambios que se hicieron en cada una de las funciones).

NOTA 1: El archivo debe de darse como línea de comando (ejem: ./ejecutable archivo), usando los argumentos de main(argc,*argv[]).

NOTA 2: Cada vez que mande llamar a una función del sistema, verificar si existe error mediante la variable errno.

NOTA 3: Usar solo una función para ordenar independientemente del tipo de dato; use un apuntador a función como argumento para la comparación de datos en la función ordenar.

NOTA 4: Las opciones 7-9 deben modificar el archivo original. AL invertir el archivo, considere que está limitado de memoria principal. Se evaluará la eficiencia de su código en cuanto a uso de memoria con respecto al peor de los casos, esto es leer todo el archivo en un arreglo grande y escribir el arreglo al revés. Puede hacer uso cualquier función de archivos, se haya o no visto en clase (fseek, rewind, ftell, etc.). De igual forma, al dar de baja a un estudiante trate de hacer programas eficientes en cuanto a memoria y/o tiempo de ejecución.