

Tarea 6 - Programación y algoritmos  
Giovanni Gamaliel López Padilla

## Índice

<b>1. Archivo de entrada</b>	<b>2</b>
1.1. Lectura de datos . . . . .	2
<b>2. Menú</b>	<b>2</b>
<b>3. Imprimir en un archivo</b>	<b>3</b>
<b>4. Ordenar información</b>	<b>3</b>
4.1. Ordenar por nombre . . . . .	3
4.2. Ordenar por edad . . . . .	3
4.3. Ordenar por promedio . . . . .	3
<b>5. Contabilización</b>	<b>4</b>
5.1. Contabilización por grupo . . . . .	4
5.2. Contabilización por turno . . . . .	4
<b>6. Actualización de estudiantes</b>	<b>4</b>
6.1. Baja de estudiantes . . . . .	4
6.2. Alta de estudiantes . . . . .	4
<b>7. Inversión de los datos</b>	<b>4</b>

## 1. Archivo de entrada

El archivo que contiene los datos iniciales del programa se dará al momento de iniciarse el programa. Este archivo debe contener como cabecera los siguientes títulos: *Calificacion*, *Edad*, *Grupo*, *Turno* y *Nombre*.

Los datos que contiene el archivo inicialmente se muestran en la tabla 1.

Calificacion	Edad	Grupo	Turno	Nombre
A+	9	C	V	Valeria Quirarte
A-	12	A	V	Valentina Quirarte
B	10	D	M	Luis Aldama
B-	14	A	V	Marco Lopez
A	15	A	V	Mayra Medellin
C	13	E	M	Pedro Arturo

**Tabla 1:** Datos iniciales del archivo `data.txt`.

### 1.1. Lectura de datos

Al inicio del programa se obtiene el número de personas (filas) que contiene el archivo. Esta cantidad es obtenida en la función `obtain_size`. Con esta cantidad es iniciado el arreglo de estructuras `students`. Al tener ya reservada la memoria para los datos de la tabla 1, se realiza la lectura de datos. Este mecanismo es realizado en la función `obtain_information`.

## 2. Menú

Al tener los datos almacenados dentro del arreglo, se inicia el menú. El menú contiene las siguientes opciones:

- Imprimir un archivo
- Ordenar por nombbre
- Ordenar por edad
- Ordenar por promedio
- Numero de estudiantes por grupo
- Numero de estudiantes por turno
- Baja de un estudiante
- Alta de un estudiante
- Invertir estudiante
- Salir

Estas opciones se ejecutan con la función `menu`. La función lee un número entero entre 0-9 para elegir las opciones posibles, ejecuta la función de la opción y al termino regresa a la lectura de la nueva opción. La única manera de salirse del menú es seleccionando la opción *Salir*.

### 3. Imprimir en un archivo

Esta opción es habilitada una vez se hayan leído los datos base. Al seleccionar esta opción por medio del menú ejecuta la función `print_file`. La cual realizará la impresión en el archivo dado siguiendo el formato descrito en la sección 1.

Al inicio de la función el archivo es cerrado, para ser abierto en modo de escritura. Al término de la función, el archivo será cerrado y abierto nuevamente en modo de lectura. Esto para asegurar los cambios dentro del archivo si es que el programa llega a cerrarse inesperadamente.

### 4. Ordenar información

EL menú contiene descritos diferentes modos de ordenar la información de los estudiantes. Los criterios creados para el ordenamiento son los siguientes:

- Ordenar por nombre
- Ordenar por edad
- Ordenar por promedio

Los criterios antes mencionados siguen el mismo algoritmo de ordenamiento. EL algoritmo de ordenamiento planteado es el `quick_sort`, el cual había sido desarrollado en tareas anteriores. Esta función fue reimplementada para recibir como argumento una función el cual le indica que objeto (conjunto de caracteres o números) es mayor con respecto a otro. Los algoritmos de cada modo de ordenamiento están contenidos en el archivo `sort.h`.

#### 4.1. Ordenar por nombre

La comparación de los nombres es realizada por orden alfabético. Si un nombre es prefijo del otro, este se pondrá ocupará un lugar menor. Por ejemplo Luisa > Luis. Esta función está contenida en `comparison_names`. El programa supone que los nombres están bien escritos, esto es que el inicio de los nombres son mayúsculas.

#### 4.2. Ordenar por edad

La comparación de las edades es realizada en orden ascendente. Al ser datos de tipo entero su comparación es más sencilla. Esta función está contenida en `comparison_ages`.

#### 4.3. Ordenar por promedio

La comparación de las calificaciones es realizada de modo descendente. Esto siguiendo el orden de USA, esto es, A+ > A > A- > B+ > B > B- > C. Esta función es implementada en `comparison_grades`. El algoritmo es semejante al ordenamiento de nombres, la diferencia radica en que máximo tendremos dos caracteres en cada elemento. Entonces se realizará el siguiente algoritmo:

```
1  int compare = grade1[0] - grade2[0];
2  if (compare == 0)
3  {
4      compare = compare_sign(grade1[1], grade2[1]);
```

```
5         if (compare != 0)
6             return compare;
7         compare = compare_sign(grade2[1], grade1[1]);
8         if (compare != 0)
9             return compare;
10        compare = grade1[1] - grade2[1];
11        return compare;
12    }
13    return compare;
```

En la línea 2, se comprueba que no sea la misma letra inicial, si no lo es, entonces devolverá la diferencia al algoritmo de [quick\\_sort](#). Si es igual, entonces recibirá si se trata de un signo +, - o en blanco. Esta función de auxilia de [compare\\_sign](#) para realizar llevar el orden de preferencia en los signos. Si el programa llega a la línea 11, esto quiere decir que las calificaciones que esta comparando tienen signo.

## 5. Contabilización

### 5.1. Contabilización por grupo

### 5.2. Contabilización por turno

## 6. Actualización de estudiantes

### 6.1. Baja de estudiantes

### 6.2. Alta de estudiantes

## 7. Inversión de los datos