**Parcial 1**

**Juan Felipe Vásquez Bolívar & Jackh Emmanuel Narváez Guerra**

**C.C 1001509367 & C.C 1002752230**

**Augusto Enrique Salazar Jiménez**

**Abril 2023**

**Universidad de Antioquia**

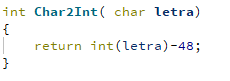
**Facultad de Ingeniería**

**Informática II**

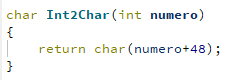
**Funciones**

Para el desarrollo de esta parte del programa y cumplir con el objetivo antes mencionado, se crean las siguientes funciones, las cuales estarán contenidas dentro de **“GestionInformacion.h”**:

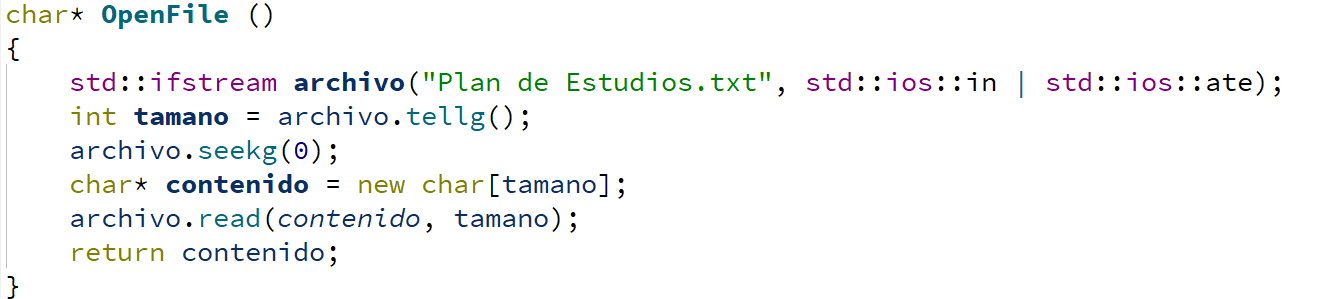
**1. Char2Int:** Función que recibe como entrada una variable tipo Char, y retorna una variable tipo Int. Esta función está diseñada para recibir algún carácter numérico y retornar el correspondiente carácter, pero como variable tipo entero, por ejemplo, si ingresamos el carácter ‘1’, retorna el número 1.



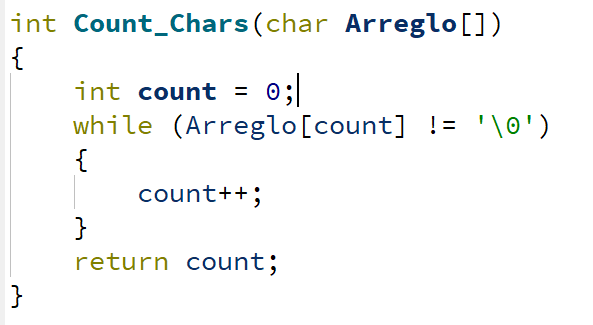
**2. Int2Char:** Función que recibe como entrada una variable tipo Int, y retorna una variable tipo Char. Esta función está diseñada para recibir algún número de un digito y retornar el correspondiente número como un carácter tipo Char, por ejemplo, si ingresamos el número 1, retorna el carácter ‘1’.



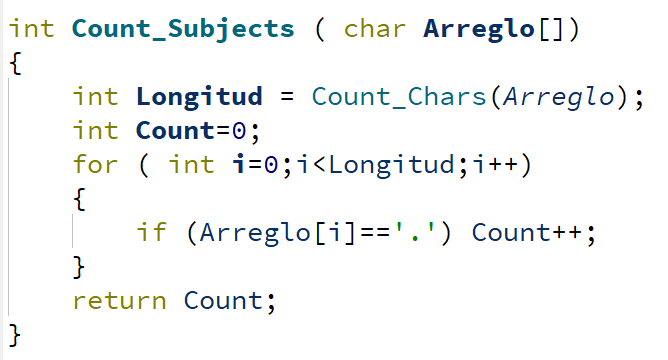
**3. Open\_File :** Función que no recibe ninguna entrada, y retorna un puntero tipo Char, que apunta a un arreglo de caracteres que contiene la información del archivo “Plan de Estudios.txt”.



**4. Count\_Chars :** Función que recibe como entrada un arreglo de caracteres, y retorna un numero entero con la cantidad de caracteres de dicho arreglo.

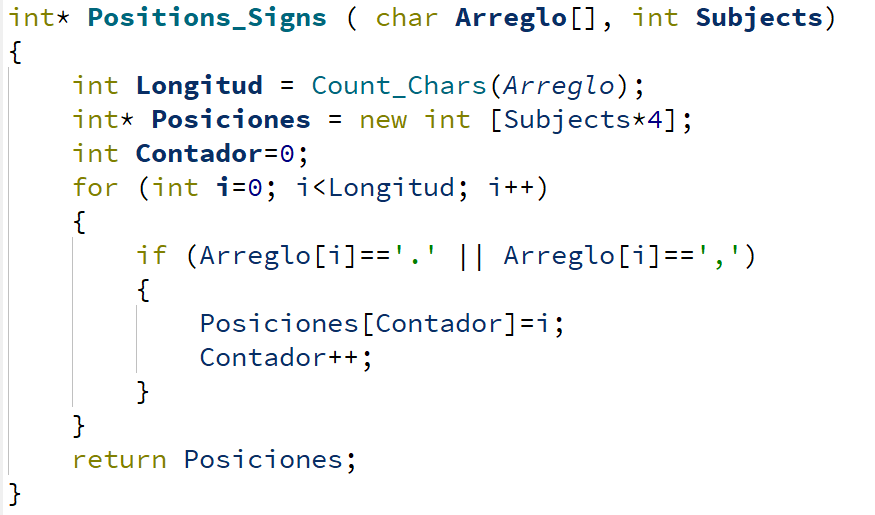


**5. Count\_Subjects :** Función que recibe como entrada un arreglo de caracteres, y retorna un numero entero con la cantidad de asignaturas registradas. Teniendo en cuenta que cada materia se encuentra separada por medio de un punto (‘.’), bastará con contar la cantidad de veces que se repite el carácter ‘.’, para determinar el número materias.



**6. Positions\_Signs:** Función que recibe como entrada un arreglo de caracteres, y un numero entero, y retorna un arreglo de enteros con la posición de los caracteres ‘,’ y ‘.’, dentro del arreglo.

El número entero corresponde a la cantidad de materias (Puntos ‘.’) que contiene el arreglo y teniendo en cuenta que en la estructura diseñada, cada materia cuenta con 4 signos ( 3 comas y 1 punto ), se crea el arreglo de enteros a retornar en la memoria dinámica.



**7. Reserve\_Matix\_Group:** Función que recibe como entrada dos arreglos de números enteros, un número entero, y retorna un puntero triple. El primer arreglo contiene la posición de los caracteres **‘,’** y **‘.’**, dentro del archivo **“Plan de Estudios.txt”** y el número entero corresponde a la cantidad de materias que contiene el anterior archivo.

Esta función realiza las reservas de memoria dinámica con base en la cantidad de caracteres que contenga cada subcategoría de cada materia, con el objetivo de realizar un uso óptimo de la memoria dinámica.

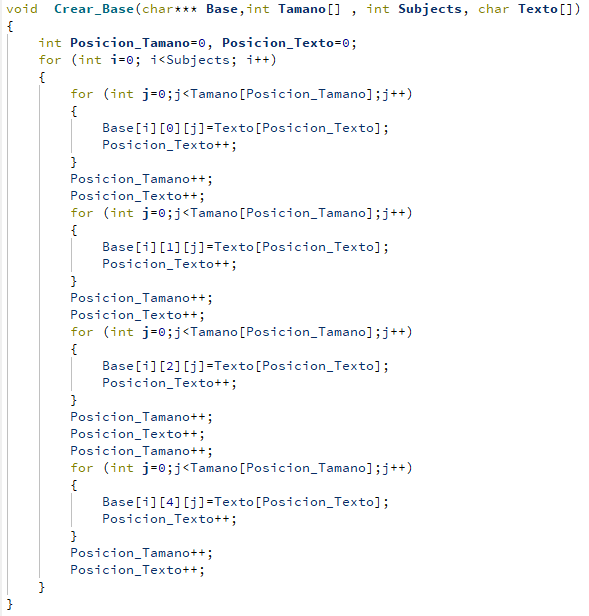
Así mismo, a medida que realiza las reservas de memorias, en el segundo arreglo de números enteros, empieza a agregar los tamaños de cada arreglo, con el objetivo de facilitar el registro de la información en el puntero triple.



**8. Crear\_Base:** Función que recibe como entrada un puntero triple, un arreglo de números enteros, un número entero y un arreglo de caracteres. El puntero triple corresponde al grupo de matrices (Punteros dobles) con su respectiva reserva de memoria, donde registraremos la información de los cursos, el arreglo de números enteros contiene la longitud de las filas de cada una de las matrices (Punteros dobles), el numero entero corresponde a la cantidad de materias, las cuales son equivalentes a la cantidad de matrices (Punteros dobles) del grupo de matrices (Puntero triple) y el arreglo de caracteres contiene la información del archivo **“Plan de Estudios.txt”.**

La función empieza a registrar la información de cada materia en su respectiva matriz (Puntero doble), ubicando en la fila (Puntero simple) 1 el nombre de la materia, en la fila 2 el código, en la fila 3 las horas con el docente, dejando la fila 4 vacía y la fila 5 con la cantidad de créditos de dicha materia.

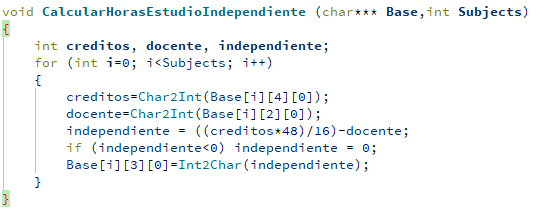
Esta función no retorna nada, ya que realiza modificaciones directas en el puntero triple.



**9. CalcularHorasEstudioIndependiente:** Función que recibe como entrada un puntero triple y un numero entero. El puntero triple contiene la información de cada uno de los cursos ubicándolos en matrices (Punteros dobles) de 5 filas (Punteros simples), y el numero entero contiene la cantidad de matrices que tiene el puntero triple.

Esta función avanza por cada matriz y extrae de la fila 3 y 5, las horas con el docente y los créditos, respectivamente, y utilizando funciones ya mencionadas, convierte sus valores a números enteros y calcula las horas de estudio independiente, para finalmente convertir este valor a carácter y registrarlo en la fila 4 de cada matriz.

Esta función no retorna nada, ya que realiza modificaciones directas en el puntero triple.



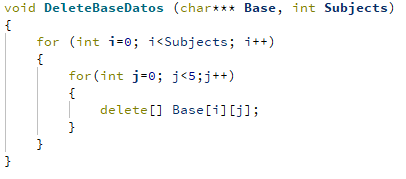
**10. PrintBaseDatos:** Función que recibe como entrada un puntero triple, un arreglo de números enteros y un número entero. El puntero triple contiene la información de los cursos registrados en matrices (Puntero dobles), el arreglo de números enteros contiene la longitud de las filas (Punteros simples) de cada una de las matrices y el número entero corresponde a la cantidad de materias.

Esta función imprime en pantalla la información de cada uno de los cursos registrados en **“Plan de Estudios.txt”,** incluyendo las horas de estudio independiente.

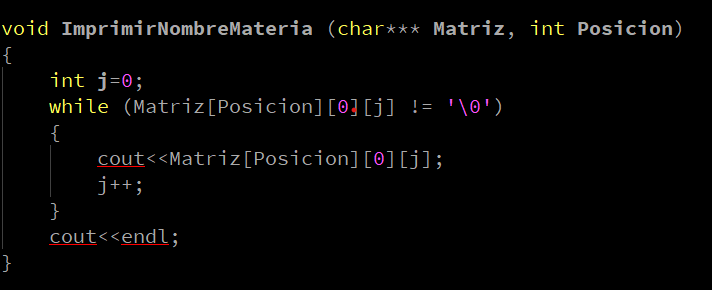


**11. PrintBaseDatos:** Función que recibe como entrada un puntero triple y un número entero. El puntero triple contiene la información de los cursos registrados en matrices (Puntero dobles) y el número entero corresponde a la cantidad de materias.

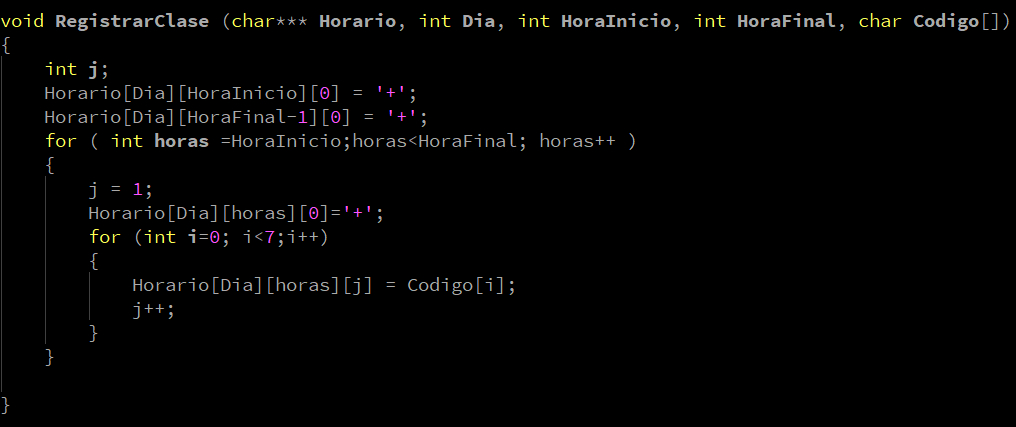
Esta función libera la memoria dinámica utilizada por el grupo de matrices (Puntero Triple), por lo tanto, no retorna nada.



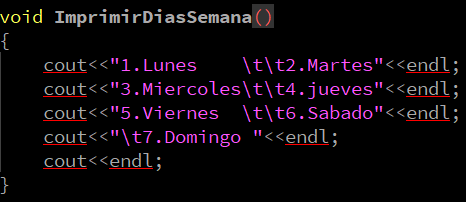
12. esta función se encarga de imprimir el nombre de la materia correspondiente a la posición "Posicion" en la matriz "Matriz". Para lograr esto, utiliza un bucle "while" que recorre los caracteres del nombre de la materia en la matriz y los imprime en la consola. Al final, se agrega un salto de línea para separar la impresión de nombres de materia.



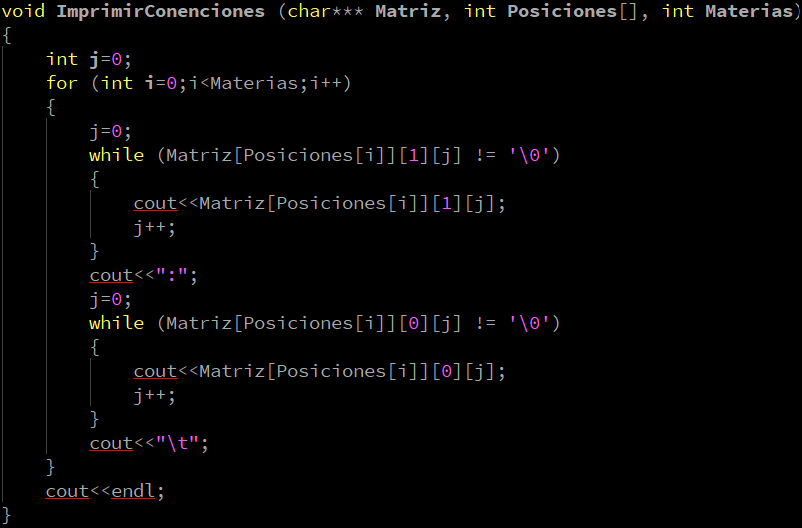
13. se utiliza para registrar una clase en un horario. Toma como entrada una matriz de cadenas de caracteres tridimensional "Horario", un entero "Dia" que representa el día en el que se va a registrar la clase, dos enteros "HoraInicio" y "HoraFinal" que representan las horas de inicio y fin de la clase respectivamente, y un arreglo de caracteres "Codigo" que representa el código de la materia de la clase.



14. esta función se encarga de imprimir los días de la semana con su número correspondiente en la consola. Utiliza la función "cout" para imprimir el nombre y número de cada día de la semana en líneas separadas.



15. la función "ImprimirConvenciones" recibe como entrada una matriz tridimensional que contiene información sobre los horarios de las materias y un arreglo que contiene la posición de las materias en la matriz. Luego, la función recorre el arreglo, imprimiendo el nombre de cada materia seguido de su horario en la consola. Finalmente, la función imprime un salto de línea.

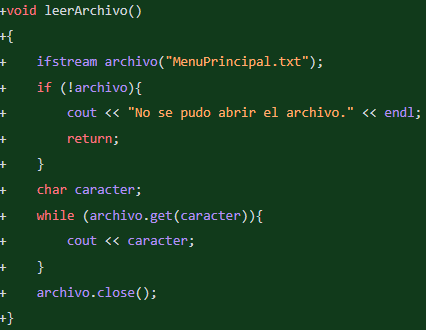


16. la función "ImportarCreditos" recibe como entrada una matriz tridimensional que contiene información sobre las materias y la posición de la materia de la cual se desea obtener el número de créditos. La función utiliza la función "Char2Int" para convertir el carácter que representa el número de créditos en la submatriz correspondiente a la materia en la "Matriz" en un número entero, y finalmente devuelve este número entero como resultado de la función.

Texto

Descripción generada automáticamente

17. "leerArchivo" que abre un archivo de texto llamado "MenuPrincipal.txt" y lee su contenido caracter por caracter, imprimiendo cada uno en la consola. Si no se puede abrir el archivo, muestra un mensaje de error en la consola.



18. la función "contarCaracteres" cuenta la cantidad de caracteres en un string utilizando un ciclo while que recorre el arreglo de caracteres hasta encontrar el carácter nulo que indica el final del string.

Texto

Descripción generada automáticamente

19. La función convertir\_a\_mayuscula toma como argumento un caracter en minúscula y devuelve el mismo caracter en mayúscula. La función se implementa verificando si el caracter en minúscula está dentro del rango ASCII correspondiente a las letras minúsculas (que va de 'a' a 'z'). Si ese es el caso, se resta 32 al valor ASCII del caracter, lo que resulta en la conversión a su respectiva letra en mayúscula. Si el caracter no está en el rango de letras minúsculas, la función simplemente devuelve el mismo caracter sin cambios.

