

IMPORTANTE:

- **Crear un proyecto con su Nombre y Apellido.**
- **Realizar todas las funciones que se indican.**
- **Añadir comentarios a su código identificando cada inciso. MUY IMPORTANTE!!!!**

Nos han encargado organizar un viejo archivo de registros médicos de un Hospital, con el fin de agilizar las búsquedas y modernizar el sistema. El archivo consta de la siguiente estructura de datos:

```
typedef struct{
    int idRegistro;
    int idPaciente;
    int idEspecialidad;
    char nombrePaciente[30];
    char apellidoPaciente[30];
    char diagnostico[100];
    char fechaAtencion[11]; /// formato: AAAA-mm-DD
    char especialidadMedica[30];
    char nombreDoctor[30];
}stRegistroMedico;
```

Archivo entregado: *registroMedico.dat* (origen de datos)

Debemos diseñar una estructura compuesta que se adapte de la mejor manera a fin de organizar la información, teniendo en cuenta que **agruparemos cada una de las atenciones médicas efectuadas a los pacientes, clasificadas por especialidad médica**. La cantidad de especialidades médicas se desconoce, pero se sabe que como máximo, habrá 10. Asimismo, el sistema debería permitir agregar nuevas especialidades si fuera necesario.

En todo momento recuerde trabajar de forma tal que cada estructura administre sus datos de acuerdo a su responsabilidad.

Utilizando el archivo que se le entregó como origen de datos, luego de diseñar una **Estructura Compuesta que considere**, implementando las funciones necesarias para administrarla, **deberá cargar la misma a fin de organizar la información en el sistema**.

Se nos pide:

1- Diseñar una estructura compuesta que se adapte a la necesidad, donde podamos almacenar todas las atenciones realizadas, organizadas según la especialidad médica. Además, deberá desarrollar las funciones que permitan inicializar **la estructura principal** y **la estructura secundaria** “*en vacío*” según corresponda al tipo de dato creado.

En un apartado de su examen, escriba un comentario donde explique brevemente el porqué de la estructura compuesta elegida.

2- Utilizando el archivo que se le entregó (“registroMedico.dat”), hacer una función que permita pasar todos los elementos desde el origen de datos (archivo) a la estructura compuesta según corresponda, invocando la/las funciones necesarias para lograrlo. Tenga en cuenta que para hacer esto deberá:

- Buscar la especialidad médica.
- Agregar una nueva especialidad médica si ésta no existe.
- Agregar el registro leído del archivo (previo procesarlo) en la estructura secundaria, teniendo en cuenta la especialidad médica a la que corresponde.
 - Para el **manejo y administración** de la **estructura secundaria**, deberá **crear todas las funciones que necesite para lograr agregar información a la misma**.

UTN FRMDP - TUP - Laboratorio 2
Segundo Parcial - 18 de noviembre de 2022

3- Realizar las funciones necesarias para **mostrar todos los elementos de la estructura compuesta, permitiendo visualizar cada especialidad médica con sus atenciones realizadas**. Tenga en cuenta la **modularización** y la **responsabilidad** de cada estructura.

4- Realizar las funciones necesarias **para efectuar el alta de nuevas atenciones al sistema**. Esta función debe interactuar con el usuario, solicitando la información para generar **un nuevo registro en la Estructura Compuesta**.

- Cargará datos mientras el usuario así lo requiera.
- Si la especialidad médica no existe, deberá agregarla.

5- Se necesita averiguar si un paciente, del cual conocemos su nombre y apellido, fue atendido en una especialidad determinada.

- Realice las funciones que considere necesarias.
- Las funciones pueden ser iterativas o recursivas.
- El subprograma retornará **1 si lo encuentra y 0 si no**.

6- Hacer una función (o varias) que determine qué especialidad médica tuvo más atenciones.

- Las funciones pueden ser iterativas o recursivas.
- El subprograma informará por pantalla la especialidad y su cantidad de atenciones.

7- Hacer una función (o varias) que realice la persistencia de las atenciones médicas de una especialidad médica en particular.

- La función recibe por parámetro el nombre de la especialidad médica que se desea guardar en un archivo, más los datos que considere necesarios.
- El archivo saliente debe ser del tipo **stRegistroMedico**.

8- Hacer una función main() que invoque a los módulos anteriores y demuestre el funcionamiento del programa.

- Para hacer esto, cree las variables que considere necesarias e invoque las funciones (de forma directa o indirecta) como corresponde en cada caso.
- Muestre los resultados cada vez que sea necesario.
- Si lo considera, cree un menú de opciones para ejecutar cada función o subprograma.
- A fin de identificar cada inciso, comente su código indicando a qué apartado corresponde, por ejemplo:
// Apartado 3.a

El desarrollo de la función main es de carácter obligatorio y sumará puntos extras en caso de necesitarlos.

Apartado	1	2	3	4	5	6	7	8	
								main	compila
Puntaje	10	40	10	10	10	10	10	Si/No	Si/No

Tabla de puntuación:

Obtenido	10	20	30	40	50	60	70	80	90	100
Nota	1	2	3	4	5	6	7	8	9	10
Condición	Desaprobado					Aprobado				