**Examen 2021 Programación IV - C**

PREAMBULO

//MUCHO TEXTO

PARTE 1 (0,5 puntos)

**1.1** Crea un nuevo módulo “grupoVacunacion” con su correspondiente fichero de cabecera y de implementación. En él, debes definir la estructura GrupoVacunacion, compuesta por dos enteros (edad mínima y edad máxima) y por una única vacuna (deberás hacer uso de la estructura Vacuna que ya se proporciona), de esta forma se sabrá qué vacuna se le asigna a cada grupo.

**1.2** Añade dentro del módulo “grupoVacunación” las siguientes funciones para poder operar:

void inicializaGrupo (<GrupoVacunacion> gv, int edadMinima, int edadMaxima, <Vacuna> v): asigna la edad mínima, la edad máxima y la vacuna al grupo de vacunación pasado como primer parámetro.

void imprimirGrupo (<GrupoVacunacion> gv): visualiza por pantalla los datos del grupo.

**1.3** Dentro del programa principal crea 3 grupos de vacunación y asígnales valores, utilizando la función inicializaGrupo (puedes basarte en el juego de ensayo que se te proporciona a continuación). Posteriormente, visualiza los datos de los grupos creados con la función imprimirGrupo.

**Juego de ensayo.** El resultado a mostrar por pantalla podría ser el siguiente:

Grupo vacunacion desde 0 hasta 45 años

Vacuna asignada: 1-Astro-Gungar

Grupo vacunacion desde 46 hasta 65 años

Vacuna asignada: 2-Antigua

Grupo vacunacion desde 66 hasta 100 años

Vacuna asignada: 3-Rasputin

PARTE 2 (0,75 puntos)

**2.1** Crea una nueva estructura dentro del módulo "grupoVacunacion” denominada Paciente. Esta estructura representa la información de un paciente y para estará compuesta por un campo DNI de tipo texto (8 números y 1 letra) y un campo entero con la edad.

**2.2** Modifica la estructura GrupoVacunacion añadiendo un array de pacientes (haz uso de la estructura anteriormente creada). Ten en cuenta que el número de pacientes que componen un grupo de vacunación no es un dato conocido a priori.

**2.3** Añade dentro del módulo “grupoVacunacion" las siguientes funciones para trabajar con los pacientes:

void asignarPacientes(<GrupoVacunacion> qv, <array-pacientes> p, int numeroPacientes): asigna el array de pacientes al grupo de vacunación (ambos pasados por parámetro).

void imprimirPacientes (<GrupoVacunacion> gv): Imprime los pacientes que tiene el grupo de vacunación pasado por parámetro.

**2.4** Dentro del programa principal crea tres arrays de pacientes (uno por cada grupo de vacunación) e introduce valores. Posteriormente, utilizando la función de asiqnarPacientes, asigna cada uno de los arrays de pacientes al correspondiente grupo de vacunación. A continuación, visualiza los datos de los grupos creados empleando la función imprimirPacientes (puedes basarte en el juego de ensayo que se te proporciona a continuación).

**Juego de ensayo.** El resultado a mostrar por pantalla podría ser el siguiente:

Grupo de vacunacion 1:

11111111A - 25 años

Grupo de vacunacion 2:

22222222B - 50 años

33333333C - 60 años

Grupo de vacunacion 3:

444444440 - 50 años

55555555E - 52 años

66666666F - 75 años

PARTE 3 (0,75 puntos)

**3.1** Añade dentro del módulo “grupoVacunacion” la siguiente función:

char\*\* obtenerDnisMalAsignados (<GrupoVacunacion> qv, <int> numDnisMalAsignados): devuelve un array de cadenas de caracteres con los DNIs de los pacientes del grupo de vacunación cuya edad no es válida (no se encuentra en el rango entre la edad mínima y la máxima del grupo al que pertenecen). Además, devuelve en el parámetro numDnisMalAsignados, el número de DNIs que llene el array que reforma la función.

**3.2** Implementa en el programa principal la siguiente función para trabajar con ficheros:

void escribirPacientes (char\*\* dnisMalAsignados, int numMalAsignados, char\* fichero): escribe en un fichero de texto los DNIs de los pacientes pasados por parámetro. En cada línea del fichero se escribiré un único DNI.

**3.3** En el programa principal llama a la función obteneroDnisMalAsignados para el tercer grupo de vacunación que has creado anteriormente. Como resultado obtendrás un array de cadena de caracteres compuesta por los DNIs de los pacientes asignados a un grupo incorrecto y el número de pacientes mal asignados a través del parámetro numDnisMalAsignados.

**3.4** En el programa principal llama a la función escribirPacientes para el array de DNIs mal asignados que has obtenido en el punto anterior. NOTA: En caso de que no hayas completado correctamente la función del punto 3.1, puedes crear el array de forma manual para completar este punto.

**3.5** Finalmente, no olvides liberar la memoria de aquellas estructuras que así lo requieran, si las hubiera.