

APELLIDOS _____ NOMBRE _____

DNI _____ ORDENADOR _____ GRUPO (A/B/C) _____

La clínica veterinaria “El Arca de Noe” tiene una residencia para cada una de las especies de animales que trata: Pez, Perro, Gato, Conejo (enumerado). De cada animal desea almacenar la siguiente información: su raza, el nombre del animal, el nombre del dueño (todos cadenas de caracteres) y el número del chip identificativo del animal (entero). Como se muestra en la figura adjunta, para almacenar toda la información usaremos un array cuyo tipo índice será la especie de animal (que podrá manejarse como un enumerado o como su ordinal) y cuyo tipo base será una lista enlazada de animales que estará ordenada por el número de chip (número que es diferente para cada animal). Se pide implementar un programa llamado `arca.cpp` que presente el siguiente menú:

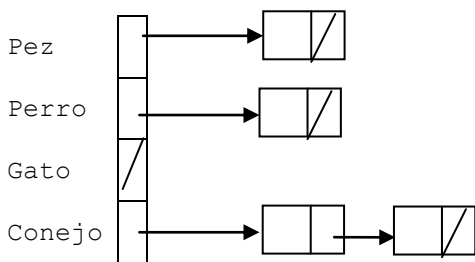


Figura. Esquema de la Estructura

```
Nombre: (Apellidos, Nombre)           Curso: 1º
Especialidad: Sistemas                 Grupo: A/B/C
Puesto: número de ordenador           Fecha: 03/09/2009
```

El Arca de Noe

=====

- A. Insertar Animal.
- B. Cargar Animales desde Fichero de Texto.
- C. Mostrar Todos los Animales.
- D. Salvar Todos los Animales a Fichero de Texto.
- E. Eliminar Animal.
- F. Buscar Animal por Nombre.
- G. Mostrar Clínica Ordenada Por Chip.
- X. Salir del Programa

Introduzca Opción:

De las opciones del menú anterior, las **OPCIONES BÁSICAS** que deben realizar **TODOS LOS ALUMNOS** son:

- A. Insertar Animal.** Se pedirá desde teclado la especie (que deberá leerse como una cadena) y todos los datos del animal y se insertará en la clínica. Si ya existiera en la estructura un animal con el mismo chip que el introducido o no hubiera memoria se informaría del error.
- B. Cargar Animales desde Fichero de Texto.** Se pedirá al usuario el nombre de un fichero de texto y se cargará toda la información en memoria. Se deberá informar de cualquier error relativo al manejo de ficheros o de memoria. Se mantendrán los datos que existan en memoria previos a la carga del fichero, añadiendo los nuevos datos a los ya existentes. En caso de que ya esté en memoria un animal con el mismo chip que el leído desde fichero se ignorará el error de animal repetido y no se insertará.
Formato recomendado del Fichero:
<ESPECIE>#<NÚMERO DE ANIMALES><ENTER>
<RAZA>#<NOMBRE:ANIMAL>#<NUMERO_CHIP><ESPACIO><NOMBRE_DUEÑO><ENTER>
.....
<RAZA>#<NOMBRE:ANIMAL>#<NUMERO_CHIP><ESPACIO><NOMBRE_DUEÑO><ENTER>
- C. Mostrar Todos los Animales.** Se mostrará por pantalla un listado de todos los animales agrupados por especie y ordenados por número de chip dentro de cada especie.
- X. Salir del Programa.** Se pedirá confirmación de salida.

Ejemplo Fichero: Animales.txt

```
Pez#1
Siam#Jaimito#123 Juan
Perro#3
Mastin#Thor#121 Pepe
Cocker#Chapi#201 Alberto
Yorkshire#Nuka#223 Laura
Gato#2
Persa#Juana#321 Conchi
Angora#Daisy#12 Maria
Conejo#0
```

Los **ALUMNOS QUE NO TENGAN SUPERADO EL TRABAJO EN CLASE** (tienen una nota acumulada < 2), además tendrán que implementar las siguientes opciones del menú principal:

- D. Salvar Todos los Animales a Fichero de Texto.** Se pedirá el nombre de un fichero de texto y se salvará toda la información con el mismo formato del apartado B. Si el fichero no existe se creará y si ya existiera se borrará su contenido.
- E. Eliminar Animal.** Se pedirá al usuario el número del chip del animal y se borrará de la estructura. Si el animal no existiera se informará del error.

Una vez completadas las operaciones básicas pedidas, PARA SACAR MÁS NOTA habrá que implementar el resto de opciones del menú:

F. Buscar Animal por Nombre. Se pedirá al usuario el nombre de un animal y se mostrará por pantalla la especie y toda la información del animal. Si hubiera más de un animal con el mismo nombre se mostrará el primero que se encuentre. Si no existiera ningún animal con dicho nombre se informará del error.

G. Mostrar Clínica Ordenada Por Chip. Se mostrará por pantalla un listado de todos los animales (indicando su especie) ordenados por número de chip, independientemente de la especie a la que pertenezcan.

Módulos a Implementar: (LA DEFINICIÓN DE LOS MÓDULOS NO SE PODRÁ MODIFICAR)

- MError: Manejo de errores.
 - Define el tipo TError como un enumerado con los posibles errores generados.
 - MuestraError: Muestra el error por pantalla.
- MAnimal: Manejo del registro TAnimal.
 - Define las constantes SP (espacio), SEP (#); y ENTER.
 - Define el tipo TAnimal como un registro con los campos raza, nombre, chip y amo.
 - LeerAnimal: Lee un animal (raza, nombre, chip y amo) desde teclado.
 - EscribirAnimal: Escribe un animal (raza, nombre, chip y amo) por pantalla.
 - LeerAnimalFicheroTXT: Lee un animal (raza, nombre, chip y amo) desde un fichero de texto cuyo descriptor se pasa como parámetro.
 - EscribirAnimalFicheroTXT: Escribe un animal (raza, nombre, chip y amo) en un fichero de texto cuyo descriptor se pasa como parámetro. (Sólo para quien tenga que hacer la opción D)
- MListaAnimales: Manejo de una lista de animales.
 - Define los tipos TListaAnimales y TNode para representar una lista enlazada de animales.
 - CrearLista: Crea una lista de animales vacía.
 - DestruirLista: Destruye una lista de animales.
 - InsertarLista: Inserta un animal en la lista ordenado por su número de chip. No podrá haber chip repetidos y deberá informarse de cualquier error producido.
 - SacarAnimalLista: Saca de la lista el animal que ocupa la posición cuyo índice se pasa como parámetro. Se considera que el primer animal almacenado estará en la posición 0. Caso de pedirse una posición donde no haya animales almacenados se informará del error.
 - LongitudLista: Devuelve el número de animales almacenados en la lista.
 - ListaVacía: Nos dice si una lista de animales está vacía.
- MClinica: Manejo del array de tipo índice especies de tipo base una lista de animales.
 - Define la constante MAX_ESPECIES.
 - Define los tipos Tespecie (enumerado de valores Pez, Perro, Gato, Conejo) y TClinica (array de tipo índice especies de tipo base una lista de animales).
 - LeerEspecie: Lee un especie desde teclado.
 - EscribirEspecie: Escribe una especie por pantalla.
 - CrearClinica: Crea un TClinica vacío.
 - DestruirClinica: Destruye una TClinica liberando toda la memoria.
 - InsertarAnimalClinica: Inserta un animal en la clínica, dadas la información de la especie y del animal. Se informará de cualquier error producido.
 - MostrarClinica: Mostrará por pantalla la información de todos los animales de la clínica agrupados por especies y ordenado por número de chip.
 - CargarClinicaFichero: Cargará en la clínica toda la información que hay en el fichero cuyo nombre se facilita. Se informará de cualquier error producido.
 - SalvarClinicaFichero: Almacenará toda la información de la clínica en el fichero cuyo nombre se facilita. Se informará de cualquier error producido. (Sólo para quien tenga que hacer la opción D).
 - EliminarAnimalClinica: Eliminará de la clínica el animal cuyo chip se pasa como parámetro. Se informará de cualquier error producido. (Sólo para quien tenga que hacer la opción E)
 - BuscarAnimalNombre: Devolverá la información del animal y su especie dado su nombre. Se informará de cualquier error producido. (Sólo para quien quiera sacar más nota)
 - MostrarClinicaOrdenadaPorChip: Mostrará por pantalla la información de todos los animales de la clínica (incluyendo su especie) ordenada por número de chip. (Sólo para quien quiera sacar más nota)

NOTAS

1. Todo PROGRAMA QUE NO COMPILE o tenga efectos laterales se considerará SUSPENSO.
2. Añadir procedimientos o funciones a la definición de un módulo y el uso de detalles de implementación de la lista fuera del módulo de implementación de la misma (MlistaAnimales.cpp) será CAUSA de SUSPENSO, es decir, en el programa principal ni en el módulo Mclínica no se podrá hacer cosas tipo: l= NULL, ptr=ptr->sig, etc.
3. Se recomienda y valora el tratamiento de errores y la buena descomposición del programa principal en procedimientos y funciones, así como el uso de procedimientos y funciones auxiliares dentro de la implementación de los módulos cuando estas sean necesarias.