

APELLIDOS _____ NOMBRE _____

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

Se quiere realizar una aplicación que permita gestionar las obras existentes en un museo. Se van a manejar obras que pueden ser de los siguientes tipos: Pintura, Escultura, Arqueología, Otros (enumerado). De cada obra se almacenará la siguiente información: referencia, título y autor (cadenas de caracteres), año de creación (entero) y valoración (real). Para almacenar toda la información usaremos un array cuyo tipo índice será el tipo de obra (que podrá manejarse como un enumerado o como su ordinal) y cuyo tipo base será una lista enlazada de obras ordenada por el año de creación de la obra. Se pide implementar un programa llamado museo.cpp que presente el siguiente menú:

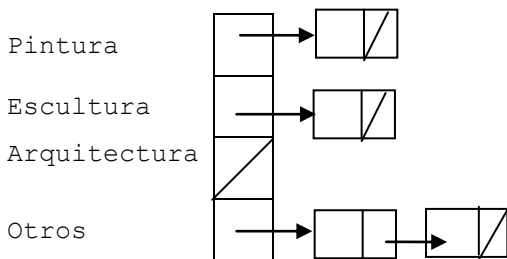


Figura. Esquema de la Estructura

```

Nombre: (Apellidos, Nombre)           Curso: 1º
Especialidad: Sistemas                 Grupo: A/B/C
Puesto: número de ordenador           Fecha: 02/09/2010

Gestion Museo
=====
A. Insertar Obra.
B. Cargar Obras desde Fichero de Texto.
C. Mostrar Catalogo Museo.
D. Eliminar Obra.
E. Salvar Obras a Fichero de Texto.
F. Mostrar Obras de un mismo Autor.
G. Mostrar Promedio Valoración Obras.
H. Mostrar Museo Ordenado por Año.
X. Salir del Programa
Introduzca Opción:
  
```

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

A. Insertar Obra. Se introducirán desde teclado todos los datos de la obra y se insertarán en el museo. Si el tipo (que se leerá como una cadena) no es válido, el programa volverá a solicitarlo. No es necesario controlar la existencia de referencias duplicadas. En caso de no haber memoria suficiente se informará del error.

B. Cargar Obras 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 **borrarán** los datos que existan en memoria previos a la carga del fichero.

Formato del Fichero:

```

<TIPOOBRA>#<NÚMERO DE OBRAS><ENTER>
<REF>#<TITULO OBRA><ENTER>
<AUTOR>#<AÑO><SP><VALORACION><ENTER>
. . .
<REF>#<TITULO OBRA><ENTER>
<AUTOR>#<AÑO><SP><VALORACION><ENTER>
  
```

C. Mostrar Catalogo Museo. Se mostrará por pantalla un listado de todas las obras agrupadas por tipo de obra y ordenadas por año de creación de cada obra.

D. Eliminar Obra. Se pedirá al usuario la referencia de una obra a eliminar del museo. En caso de no existir ninguna obra con dicha referencia se informará de ello.

X. Salir del programa. Se pedirá confirmación de salida.

Ejemplo Fichero: obras.txt

```

Pintura#4
P1#Guernica
Picasso#1937 2500
V1#Las Meninas
Velazquez#1656 3000
G1#Girasoles
Van Gogh#1888 4000
M1#El juicio final
Miguel Angel#1541 2000
Escultura#1
M2#David
Miguel Angel#1555 3000
  
```

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:

E. Salvar Obras 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 descartará su contenido.

F. Mostrar Obras de un mismo Autor. Se pedirá al usuario el nombre de un autor y a continuación se mostrarán todas las obras de dicho autor. En caso de no existir el autor se mostrará un listado vacío.

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

G. Mostrar Promedio Valoración Obras. Al seleccionar esta opción se calculará la media de valoración de todas las obras del museo independientemente de su tipo y se mostrará el valor obtenido. En caso de no haber obras, se informará de ello.

H. Mostrar Museo ordenado por Año. Se mostrará por pantalla un listado de todas las obras del museo ordenadas por el año de su creación, independientemente del tipo de obra 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.
 - MostrarError(err): Muestra el error por pantalla.
- MObra: Manejo del registro TObra.
 - Define las constantes SP (espacio), SEP (#) y ENTER.
 - Define el tipo TObra como un registro con los campos ref, titulo, autor, anyo y valoracion.
 - LeerObra(obra): Lee una obra (referencia, título, autor, año y valoración) desde teclado.
 - EscribirObra(obra): Escribe una obra (referencia, título, autor, año y valoración) por pantalla.
 - LeerObraFicheroTXT(fich,obra): Lee una obra (referencia, título, autor, año y valoración) desde un fichero de texto cuyo descriptor se pasa como parámetro.
 - EscribirObraFicheroTXT(fich,obra): Escribe una obra (referencia, título, autor, año y valoración) en un fichero de texto cuyo descriptor se pasa como parámetro. (Sólo para quien tenga que hacer la opción E)
- MListaObras: Manejo de una lista de obras.
 - Define los tipos TListaObras y TNode para representar una lista enlazada de obras.
 - CrearLista(lista): Crea una lista de obras vacía.
 - DestruirLista(lista): Destruye una lista de obras.
 - InsertarLista(lista,obra,err): Inserta una obra en la lista ordenada por el año. Deberá informarse de cualquier error producido.
 - ObtenerObraLista(lista,pos,obra,err): Obtiene de la lista (sin borrar) la obra que ocupa la posición cuyo índice se pasa como parámetro. La primera obra estará en la posición 0. Caso de pedirse una posición no válida, se informará del error.
 - EliminarObraLista(lista,ref,err): Elimina de la lista la obra que coincida con la referencia que se pasa como parámetro. En caso de no existir una obra con dicha referencia, se informará del error.
 - LongitudLista(lista): Devuelve el número de obras almacenadas en la lista.
 - ListaVacía(lista): Indica si una lista de obras está vacía.
- MMuseo: Manejo del array de listas de obras del museo.
 - Define la constante MAX_TIPOS que indica el número de tipos de obras existentes (4).
 - Define los tipos TTipoObra (enumerado de valores Pintura, Escultura, Arqueología y Otros) y TMuseo (array de listas de obras, debe haber tantas listas como indica MAX_TIPOS).
 - LeerTipoObra(tipo,err): Lee un tipo de obra desde teclado.
 - EscribirTipoObra(tipo): Escribe un tipo de obra por pantalla.
 - CrearMuseo(museo): Crea un TMuseo vacío.
 - DestruirMuseo(museo): Destruye un TMuseo liberando toda la memoria.
 - InsertarObraMuseo(museo,tipo,obra,err): Inserta una obra en el museo, dados el tipo de obra y la obra. Se informará de cualquier error producido.
 - MostrarMuseo(museo,err): Mostrará todas las obras del museo agrupadas por tipos y ordenadas por año.
 - CargarMuseoFichero(museo,nomFich,err): Cargará en el museo toda la información que hay en el fichero cuyo nombre se facilita. Se informará de cualquier error producido.
 - EliminarObraMuseo(museo,ref,err): Eliminará del museo la obra cuya referencia se pasa como parámetro. Se informará de cualquier error producido.
 - SalvarMuseoFichero(museo,nomFich,err): Almacenará toda la información del museo en el fichero cuyo nombre se facilita. Se informará de cualquier error producido. (Sólo para quien tenga que hacer la opción E).
 - MostrarObrasAutor(museo,autor,err): Mostrará un listado de todas las obras del autor cuyo nombre se pasa como parámetro. En caso de no haber ninguna obra se mostrará un listado vacío. (Sólo para quien tenga que hacer la opción F).
 - CalcularPromedioValoracion(museo,err): Calculará la media de valoración de todas las obras del museo, retornando dicho valor. Si no hay obras se informará de ello. (Sólo para quien quiera sacar más nota)
 - MostrarMuseoOrdenadoPorAnyo(museo): Mostrará por pantalla la información de todas las obras del museo ordenadas por año de creación. (Sólo para quien quiera sacar más nota)

NOTAS

1. Para aprobar será **NECESARIO** que **TODAS** las **OPCIONES BÁSICAS** funcionen correctamente.
2. Todo **PROGRAMA QUE NO COMPILE** o tenga **efectos laterales** se considerará **SUSPENSO**.
3. **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 (MListaObras.cpp) será **CAUSA de SUSPENSO**, es decir, en el programa principal o en el módulo MMuseo no se podrá hacer cosas tipo: l= NULL, ptr=ptr->sig, etc.

4. 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.