



Departamento de Lenguajes
y Ciencias de la Computación

LABORATORIO DE PROGRAMACIÓN

Examen Ordinario de Septiembre

1º Gestión Curso 2000/2010

10/09/2010 a las 9:30

APELLIDOS _____ NOMBRE _____

DNI _____ ORDENADOR _____ GRUPO (A/B) _____

La Universidad de Málaga debido a los cambios necesarios para la implantación de los nuevos grados nos pide que desarrollemos una aplicación para manejar la oferta de asignaturas para el nuevo curso en los siguientes centros (que serán manejados como enumerado): *informatica, telecomunicaciones, industriales, ciencias*. Para el manejo de la información se tendrá un tipo TUMA (que será implementado por el módulo MUMA) y que será un array cuyo tipo índice es el enumerado que indica el centro y cuyo tipo base será un TLista (implementado dentro del módulo MLista), que será una lista de asignaturas que estará ordenada por el código de la asignatura (que es un número único en toda la universidad). De cada asignatura almacenaremos la siguiente información: código de la asignatura (natural), titulación (cadena), curso (natural) y número de alumnos matriculados (natural). El número máximo de alumnos matriculados en una asignatura es de 65. Se pide realizar una aplicación que presente las siguientes opciones de menú:

```
Nombre: (Apellidos, Nombre)          Curso: 1º
Especialidad: Gestión                 Grupo: A/B
Puesto: número de ordenador          Fecha: 10/09/2010
```

ASIGNATURAS UMA

=====

- A. Insertar Asignatura.
- B. Listar Asignaturas.
- C. Cargar Asignaturas desde Fichero.
- D. Salvar Asignaturas a Fichero.
- E. Buscar Asignatura.
- F. Matricular alumno.
- G. Gestión automática de matriculación.
- X. Salir del Programa

Introduzca Opción:

- A. **Insertar Asignatura.** Se pedirá desde teclado el nombre del centro y los datos de la asignatura (el número de matriculados será 0) y se insertará en la estructura. Si ya existiera una asignatura en dicho centro con el mismo código o no hubiera memoria se informaría del error (No será necesario comprobar que no existe el código en otros centros).

Ejemplo:

```
Centro: Informatica
Código: 121
Titulación: Graduado en Informatica
Curso: 1
Operación completada Satisfactoriamente.
```

- B. **Listar Asignaturas.** Se mostrará por pantalla un listado de todas las asignaturas agrupadas por centro.

Ejemplo:

```
Listado de Asignaturas por Centro
=====
Centro: informatica
Código: 111
Titulación: Graduado en Informatica
Curso: 1
Número de alumnos matriculados: 55
Código: 121
Titulación: Graduado en Informatica
Curso: 1
Número de alumnos matriculados: 0
```

Código: 131
Titulación: Graduado en Informatica
Curso: 3
Número de alumnos matriculados: 15
Código: 141
Titulación: Graduado en Informatica
Curso: 4
Número de alumnos matriculados: 10
Código: 142
Titulación: Graduado en Informatica
Curso: 4
Número de alumnos matriculados: 13

Centro: telecomunicaciones
Código: 211
Titulación: Graduado en Telecomunicaciones
Curso: 1
Número de alumnos matriculados: 25
Código: 222
Titulación: Graduado en Telecomunicaciones
Curso: 2
Número de alumnos matriculados: 15

Centro: industriales

Centro: ciencias
Código: 411
Titulación: Graduado en Biologia
Curso: 1
Número de alumnos matriculados: 50

- C. Cargar Asignaturas desde Fichero.** Se pedirá el nombre de un fichero de texto y insertará en la estructura toda la información existente con el formato adjunto. Se deberá informar de cualquier error relativo al manejo de ficheros o de memoria. En caso de que ya esté en memoria una asignatura con el mismo código que el leído se ignorará el error de asignatura repetida. Formato del Fichero:

```
<CENTRO><ESPACIO><NUMERO ASIGNATURAS><ENTER>  
<CODIGO><ESPACIO><CURSO><ESPACIO><MATRICULADOS><ESPACIO><TITULACION><ENTER>  
. . . tantas líneas como diga <NUMERO ASIGNATURAS>  
<CODIGO><ESPACIO><CURSO><ESPACIO><MATRICULADOS><ESPACIO><TITULACION><ENTER>
```

Ejemplo: curso2010.txt

```
informatica 5  
111 1 55 Graduado en Informatica  
121 3 35 Graduado en Informatica  
131 3 15 Graduado en Informatica  
141 4 10 Graduado en Informatica  
142 4 13 Graduado en Informatica  
telecomunicaciones 2  
211 1 25 Graduado en Telecomunicaciones  
222 2 15 Graduado en Telecomunicaciones  
industriales 0  
ciencias 1  
411 1 50 Graduado en Biologia
```

- X. Salir del Programa.** Se pedirá confirmación de salida.

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 Asignaturas a Fichero.** Se pedirá al usuario el nombre de un fichero de texto y se almacenará en el fichero toda la información contenida en la estructura, sin que ésta se vea modificada, en el mismo formato de la opción C. Se deberá informar de cualquier error relativo al manejo de ficheros o de memoria.

- E. Buscar Asignatura.** Se pedirá el código de una asignatura y se mostrará por pantalla el centro y datos de dicha asignatura. Ejemplo:

Codigo de la Asignatura: 411
Centro: ciencias
Código: 411
Titulación: Graduado en Biologia
Curso: 1
Número de alumnos matriculados: 50

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

F. Matricular alumno. Se solicitará el código de la asignatura donde desea matricularse el alumno y se incrementará en 1 el número de alumnos matriculados en dicha asignatura. Si la asignatura no existiera o ya estuviera completa se mostrará el mensaje de error correspondiente.

Ejemplo:

Código de la Asignatura: 411

Operación completada Satisfactoriamente.

G. Gestión automática de matriculación. Se pedirá el nombre de un fichero de texto que contiene en cada línea el NIF de un alumno y el código de la asignatura en que desea matricularse, separados por espacio. Se procesará todo el fichero procediendo a las matriculaciones y se generará un fichero cuyo nombre será "Matriculacion." + el nombre del fichero de entrada en el que figurarán las matriculaciones realizadas según el formato del ejemplo:

Fichero de Entrada: info2010.txt

11111111-A 111

11111111-A 121

11111111-A 131

22222222-B 411

33333333-c 411

Fichero de Salida: Matriculacion.info2010.txt

11111111-A matriculado en el centro informatica en la asignatura 111 de la titulación Graduado en Informatica

11111111-A matriculado en el centro informatica en la asignatura 121 de la titulación Graduado en Informatica

11111111-A matriculado en el centro informatica en la asignatura 131 de la titulación Graduado en Informatica

22222222-B matriculado en el centro ciencias en la asignatura 411 de la titulación Graduado en Biología

33333333-c matriculado en el centro ciencias en la asignatura 411 de la titulación Graduado en Biología

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.

➤ Define los siguientes PROC/FUNC:

▪ MuestraError(error): Muestra el error por pantalla.

❖ MAsignatura:

➤ Define las constantes MAX_ALUM (65), ENTER ('\n') y SP (' ')

➤ Define los tipos TNatural y TAsignatura

➤ Define los siguientes PROC/FUNC:

▪ Codigo(a): Devuelve el código de la asignatura que se pasa como parámetro.

▪ Curso(a): Devuelve el curso de la asignatura que se pasa como parámetro.

▪ Titulacion(a): Devuelve la titulación de la asignatura que se pasa como parámetro.

▪ Matriculados(a): Devuelve el número de alumnos matriculados de la asignatura que se pasa como parámetro.

▪ Matricula(a,num,error): Incrementa si es posible el número de matriculados en la asignatura en tantas unidades como se le diga por parámetro.

▪ LeeAsignatura(a): Lee una asignatura desde teclado.

▪ EscribeAsignatura(a): Escribe una asignatura por pantalla.

▪ LeeAsignaturaFichero(fich,a): Lee una asignatura desde un fichero de texto cuyo manejador se le pasa como parámetro.

▪ EscribeAsignaturaFichero(fich,a): Escribe una asignatura en un fichero de texto cuyo manejador se le pasa como parámetro.

❖ MLista:

➤ Define los tipos TLista y TNode

➤ Define los siguientes PROC/FUNC:

▪ CrearLista(): Crea una Lista Vacía.

▪ ListaVacía(l): Comprueba si una lista está vacía.

▪ ListaLlena(l): Comprueba si una lista está llena.

▪ InsertarLista(l,a,error): Inserta una asignatura en la lista.

▪ MostrarLista(l): Muestra una lista por pantalla.

▪ SacarPrimeroLista(l,a,error): Saca y devuelve la primera asignatura de la lista si no está vacía.

▪ EliminarLista(l,código,error): Elimina (si existe) la asignatura cuyo código se le pasa como parámetro.

▪ DestruirLista(l): Libera toda la memoria reservada por la lista.

❖ MUMA

- Define la constante MAX_CENTROS (4)
- Define los tipos TCentro y TUMA
- Define los siguientes PROC/FUNC:
 - Centro_a_Cadena(c): Devuelve la cadena que representa a un centro.
 - Cadena_a_Centro(s, c, ok): Convierte, si es posible, una cadena en el centro al que representa.
 - SUC(c): Devuelve el sucesor del centro que se le pasa como parámetro.
 - CrearUMA(tabla): Inicializa las listas de un array TUMA.
 - DestruirUMA(tabla): Destruye las listas de un array TUMA.
 - InsertarAsignaturaUMA(tabla, c, a, error): Inserta en un array TUMA una asignatura dado su centro.
 - ListarAsignaturasUMA(tabla): Muestra por pantalla todas las asignaturas de un array TUMA.
 - CargarAsignaturasUMAFichero(tabla, nomFich, error): Carga en la estructura todas las asignaturas que se encuentran en el fichero cuyo nombre se pasa como parámetro.
 - SalvarAsignaturasUMAFichero(tabla, nomFich, error): Salva todas las asignaturas que se encuentran en la estructura en el fichero cuyo nombre se pasa como parámetro.
 - BuscarAsignaturaUMA(tabla, codigo, c, a, error): Busca una asignatura en la estructura por su código y devuelve el centro y los datos de la asignatura.
 - MatricularAsignaturaUMA(tabla, codigo, error): Intenta la matriculación en la asignatura cuyo código se pasa como parámetro.
 - GestionAutomaticaMatriculacion(tabla, nomFich): Realiza la matriculación automática de los alumnos que se encuentran en el fichero que se pasa como parámetro.

NOTAS

1. El directorio de trabajo será C : \LPGSEP10
2. OPCIONES A, B y C (mínimo para aprobar para quien tenga superada la nota de clase) 3 puntos.
3. OPCIONES D, y E (sólo deben hacerla quien no tenga superada la nota de clase) 2 puntos
4. OPCIONES F, G y calidad de la implementación (opciones para subir nota) 5 puntos
5. **El uso de detalles de implementación fuera del módulo de implementación del mismo será CAUSA de SUSPENSO**, es decir, en el programa principal o en el módulo MUMA no se podrá hacer cosas tipo: l= NULL, ptr=ptr->sig, etc.
6. **Añadir procedimientos o funciones a la definición de un módulo será CAUSA de SUSPENSO**
7. Todo **PROGRAMA QUE NO COMPILE** o tenga **efectos laterales** se considerará **SUSPENSO**.
8. No se distinguirá entre mayúsculas y minúsculas.
9. 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 éstas sean necesarias.
10. Recordad que todo fichero de cabecera debe llevar su pragma #ifndef #endif . Por ejemplo:

```
#ifndef _MError_h_
#define _MError_h_
    . . .
#endif
```