UNIVERSIDAD DE MÁLAGA

Departamento de Lenguajes y Ciencias de la Computación

LABORATORIO DE PROGRAMACIÓN

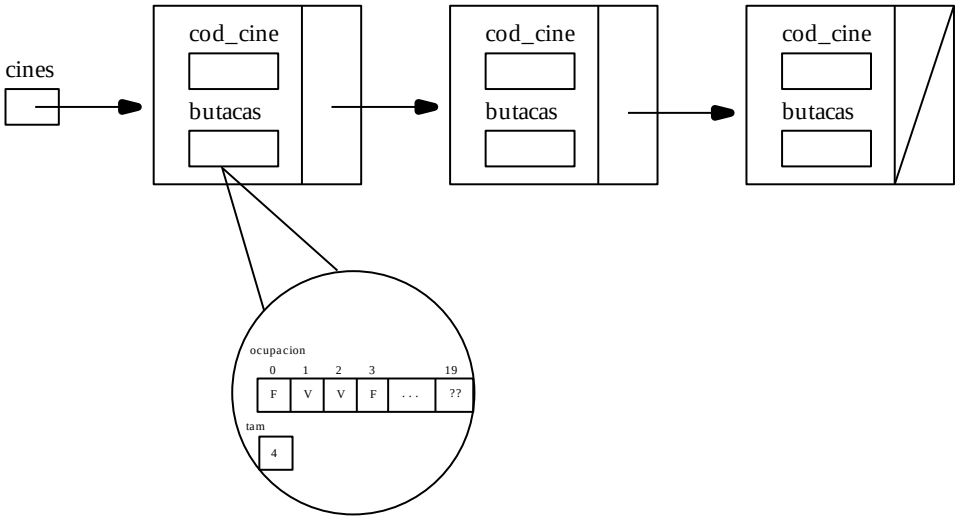
E.T.S.I. Informática- 1º SISTEMAS. 4-SEPTIEMBRE-2008

APELLIDOS_____NOMBRE_____

DNI _____ ORDENADOR_____GRUPO (A/B/C)_____

Se desea implementar un programa para gestionar la compra de tickets en una cadena de cines. Para ello, se ha diseñado el programa adjunto `cines.cpp`, que ofrece un menú con las opciones básicas, y que deberá completarse con las operaciones necesarias según se explica más adelante.

La estructura de datos que almacena la información (ver figura adjunta) vendrá dada por una **lista enlazada <u>ordenada</u>** por el código del cine, donde cada nodo almacena en el campo `cod_cine`[1] el código del cine que representa, y en el campo `butacas`, la información de ocupación de sus butacas[2]. El campo `butacas` se define mediante el tipo `TButacas`, un registro con dos campos, donde el campo `tam` almacena el número de butacas del cine (nunca superior a 20) y el campo `ocupacion` es un array en el que cada posición "i" representa el estado de ocupación de la butaca "i" en el cine.



Se pide, desarrollar los módulos `MListaCines` y `MButacas`, necesarios para completar el módulo de programa adjunto `cines.cpp`, teniendo en cuenta que:

- El módulo de programa suministrado `cines.cpp` tendrá que modificarse apropiadamente para implementar las distintas opciones del menú.
- El módulo `MButacas` define el tipo `TButacas` y las siguientes operaciones:
  - `InicializarButacas(but,num)`. Inicializa la estructura `but`, a un estado en el que hay `num` butacas, todas ellas vacías. Si `num` es mayor que 20 (máximo número de butacas por cine), entonces la estructura se inicializará con exactamente 20 butacas.
  - `LeerButacasFich(but,fich)`. Inicializa la estructura `but` con la información del estado de ocupación de un cine (el número de butacas y su estado de ocupación), leído del fichero `fich`. Nótese que `fich` no es el nombre de un fichero, sino el manejador del fichero del que se lee. Se asume que el formato del fichero proporcionado es correcto. El registro en el fichero tendrá el siguiente formato: `num␣ocupación`, donde `num` es el número de butacas[1] y `ocupacion` es una secuencia de `num` caracteres donde cada carácter puede ser 'O' o 'V'. Por ejemplo, el registro `10␣VOOOOOOOOV` representa las butacas de un cine con 10 butacas, donde la primera y la última están vacías y el resto ocupadas.

[1] Número natural

[2] Las butacas se numeran empezando en 0.

- `ComprarButacas(but,num,ok)`. Intenta comprar `num` butacas en la estructura `but`. Si es posible (había al menos `num` butacas vacías, no necesariamente de forma consecutivas), devuelve `true` en el parámetro `ok`, y las pone a ocupadas (no importa cuales se pongan a ocupada, siempre que se pongan exactamente `num`). En caso de no haber butacas suficientes, devuelve `false` en `ok`.
- `MostrarOcupacion(but)`. Muestra en pantalla una línea con el estado de ocupación de las butacas `but`. Las butacas deben aparecer en orden, apareciendo por cada una, 'O' si está ocupada y 'V' si está vacía. Por ejemplo, si el cine tiene 10 butacas, estando las dos primeras vacias y el resto ocupadas, deberá mostrar: VVOOOOOOOO

- El módulo `MListaCines` define el tipo `TCine`, que representa un cine, el tipo `TError`, un enumerado que permitirá informar sobre los errores que se producen en el módulo, y el tipo `TLista`, que representa a una lista de cines. Definirá además las operaciones básicas para trabajar con una lista de cines, que serán:
  - `CrearLista(L)`. Esta operación crea en `L` una lista de cines vacía.
  - `ListaVacia(L)`. Esta operación devuelve `true` si la lista de cines `L` está vacía, y `false` en caso contrario.
  - `LeerCinesFichero(L,nom_fich,error)`. Esta operación crea en `L` una lista de cines con la información almacenada en el fichero de texto `nom_fich`. Si la lista de entrada contenía alguna información, ésta se destruirá antes de comenzar la lectura desde el fichero. Devolverá un error en caso de que no pueda abrirse el fichero.

    El fichero estará formado por líneas de texto donde cada línea contiene los datos de un cine, dados por el código del cine y la información sobre sus butacas según el formato descrito anteriormente en la operación `LeerButacasFich`:

    `código_cine□información_butacas®`

    Por ejemplo, si hay dos cines, uno con código 101, y 10 butacas (todas ellas ocupadas salvo la primera y la última), y otro con código 120, y 7 butacas (todas ellas vacías menos la última), su información se almacenaría de la siguiente forma:

    101□10□VOOOOOOOOV®
    120□7□VVVVVVO®

  - `MostrarCines(L)`. Muestra por pantalla la información de todos los cines (código y estado de ocupacion de sus butacas) contenidos en `L`.
  - `InsertarCine(L,cod,but,error)`. Añade a la lista de cines `L` un nuevo nodo (**ordenado por código**), cuyo código es `cod`, y `but` son el número de butacas. Devolverá un error si el cine ya existe.
  - `BorrarCine(L,cod,error)`. Elimina de la lista de cines `L` aquel cuyo código es `cod`. Devolverá un error si el cine no existe.
  - `ComprarButacasCine(L,cod,num,error)`. Selecciona de la lista de cines aquel cuyo código es cod. Devolverá un error si el cine no existe. Si existe, intenta comprar `num` butadas en dicho cine. Devolverá un error si no hay un número de butacas suficiente.
  - `DestruirCines(L)`. Libera todos los recursos ocupados por la lista de cines `L`.

LAS OPERACIONES DESCRITAS ANTERIORMENTE SON EL CONJUNTO MÍNIMO NECESARIO PARA IMPLEMENTAR LAS OPCIONES BÁSICAS DEL MENÚ PRINCIPAL:

Opciones BÁSICAS del menú principal (para todos los alumnos):

A.- Leer cines de un fichero (cuyo nombre se lee por teclado)
B.- Mostrar cines
C.- BorrarCine (cuyo código se lee por teclado)
D.- InsertarCine (cuyo código y número de butacas se leen por teclado)
E.- Comprar butacas de un cine (cuyo código y número de butacas se leen por teclado)
X.- Salir del Programa

Los ALUMNOS QUE NO TENGAN SUPERADO EL TRABAJO EN CLASE (tienen una nota acumulada < 2), además tendrán que implementar las siguientes operaciones adicionales:

- En el módulo `MListaCines`:
  - `ListadoOrdenado(L)`. Recibe la lista de cines y la muestra por pantalla, ordenada de menor a mayor por el número de butacas vacías existentes en las salas.

Esta operación es necesaria para extender el menú principal con la siguiente opción:

F- Listado Ordenado por butacas vacías

Una vez completadas las operaciones básicas pedidas, PARA SUBIR NOTA proceda a implementar las siguientes operaciones adicionales:

- En el módulo `MListaCines`:
  - `ComprarButacasPosicionCine(L,cod,num,pos,error)`. Intenta comprar `num` butacas en el cine cuyo código se pasa como parámetro, **a partir de la posición pos**. Si es posible (había al menos `num` butacas vacías) las pone a ocupadas. En caso de no haber butacas suficientes, devuelve el correspondiente error.
  - `BuscarHuecoAjustadoCine(L,cod,num,error)`. Dado un código de cine y el número de butacas pedidas devuelve la posición dentro del cine a partir de la cual aparecen al menos `num` butacas consecutivas. En caso de que haya más de un hueco que cumpla el criterio deberá devolverse la posición del hueco de tamaño más pequeño. Si hay varios de igual tamaño no importa cuál se devuelva. Devolverá un error en caso de que no haya un hueco del tamaño necesario.
- En el módulo `MButacas`:
  - `ComprarButacasPosicion(but,num,pos,ok)`. Intenta comprar `num` butacas en la estructura `but`, **a partir de la posición pos**. Si es posible (había al menos `num` butacas vacías), devuelve `true` en el parámetro `ok`, y las pone a ocupadas. En caso de no haber butacas suficientes, devuelve `false` en `ok`.
  - `BuscarHuecoAjustado(but,num,pos,ok)`. Es una operación que devuelve en `pos` la posición dentro del array de butacas a partir de la cual aparecen al menos `num` butacas consecutivas. En caso de que en la estructura `but` haya más de un hueco que cumpla el criterio deberá devolverse la posición de inicio de hueco de tamaño más próximo al número pedido. Devolverá en `ok` `true` or `false` dependiendo de si hay o no en el array de butacas `num` butacas consecutivas a partir de la posición `pos`.

Estas operaciones son necesarias para extender el menú principal con las siguientes opciones:

Opciones ADICIONALES del menú principal para subir nota (para todos los alumnos):

G.- Comprar butacas a partir de una posición dada (el código del cine, el número de butacas y la posición a partir de la cual se quieren comprar se leen desde teclado. Al terminar se muestra el estado actual del cine).

H.- BuscarHuecoAjustado (el código del cine y el número de butacas se leen desde teclado. Si se encuentra un hueco se muestra por pantalla la posición donde comienza dicho hueco)

NOTAS

1. Es obligatorio trabajar en el directorio **`C:\LPSSEPT08`**.
2. **Mínimo Obligatorio para Aprobar para quien tenga superado el trabajo en clase** (3 puntos): correcta la definición de tipos, la modularización y funcionar **CORRECTAMENTE** las opciones señaladas en la parte básica (A-E,X).
3. **Mínimo Adicional Obligatorio para Aprobar para quien NO tenga superado el trabajo en clase** (2 puntos): funcionar **CORRECTAMENTE** todas las opciones señaladas en la parte básica (A-E,X), más la opción F.