USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Escola Técnica Superior de Enxeñaría

Programación II
Práctica II - Gestión básica de listas y colas

1. Objetivos

En esta práctica utilizaremos los TAD lista y cola para modelar e implementar una solución a la gestión de colas y cobro de bonificaciones en un cine. En la práctica trataremos de:

- Reutilizar los TAD lista y cola, ya implementados y discutidos en clase de teoría.
- Valorar qué TAD es el más adecuado para el diseño de las diferentes tareas del programa.

2. Descripción del programa

La cadena de cines **CinePlus** desea elaborar un sistema informático para gestionar las colas de clientes y los pagos de bonificaciones al personal de las taquillas que sean ágiles despachando entradas. Cada cajero gestiona la cola de clientes de una única película y dichos clientes son atendidos por orden de llegada. Cuando un cliente entra en la cola debe introducir el número de entradas que quiere comprar para la película que gestiona esa taquilla.

El sueldo del personal de taquilla pasa ahora a componerse de una parte fija y una parte variable (bonificaciones) en función del número de entradas que vendan en su taquilla, a razón de 0,60 euros de bonificación por cada entrada. En cualquier momento el encargado de la taquilla puede comprobar las bonificaciones que tiene disponibles en ese momento y retirar una parte o la totalidad de las mismas.

3. Especificaciones básicas del problema

- El programa es iniciado por la persona encargada de una taquilla concreta.
- La cola de clientes de la taquilla debe gestionarse con el TAD cola.
- La lista de las bonificaciones que va obteniendo el personal con cada cliente debe gestionarse con el TAD lista.

4. Descripción del programa a desarrollar

La cola de espera podrá inicializarse directamente desde línea de comandos (usa los argumentos de main()) con un conjunto de números enteros que se corresponderán con el número de entradas que lleva cada cliente que ya está en la cola de esa taquilla.

El programa presentará un menú con las siguientes opciones:

(a) Ponerse a la cola: en esta opción el cliente debe anotar el número de entradas que desea, que será introducido en la cola. Deberá comprobarse que el número de entradas es válido (mayor que 0), para lo que es útil escribir la función `int comprobarNumeroEntradas(TIPOELEMENTOCOLA n)`. Al finalizar la inserción, debe imprimirse el valor del primer elemento de la cola, para lo que debéis escribir la función `void imprimirPrimeroCola(TCOLA colaEspera)`.

(b) Atender cliente: el taquillero atiende automáticamente al cliente que lleva más tiempo en la cola (el primero de la cola). La bonificación generada por ese cliente se añade al final de la lista de bonificaciones del taquillero. Al finalizar, el programa debe imprimir el valor del primer elemento de la cola (mediante la función `imprimirPrimeroCola()`) y la lista completa de bonificaciones del taquillero, para lo que tendréis que escribir la función `void imprimirListaBonificaciones(TLISTA listaBonificaciones)`.

(c) Cobrar bonificaciones: se permitirá al taquillero cobrar parte de las bonificaciones acumuladas hasta el momento. El taquillero debe introducir la cantidad a retirar (positiva, para lo que debéis implementar la función `int comprobarValorARecaudar(TIPOELEMENTOLISTA v)`). A continuación debe llamarse a la función `void recaudarBonificaciones(TLISTA *listaBonificaciones, TIPOELEMENTOLISTA dineroARecaudar)`, que permite al taquillero cobrar la bonificación indicada en la variable `dineroARecaudar` si el saldo es suficiente, y actualiza la lista de bonificaciones. La verificación de que la cantidad total de su lista es suficiente se debe hacer utilizando la función `TIPOELEMENTOLISTA totalBonificaciones(TLISTA listaBonificaciones)`). Si la cantidad está disponible, las bonificaciones se van recaudando por orden: primero se recaudan las bonificaciones de menor cantidad y luego las más grandes. Si al ir recaudando las bonificaciones la suma del total se pasa de la cantidad solicitada, debe modificarse la última bonificación (que provocó que nos pasáramos) para permitir cobrar la cantidad exacta. Por último, desde el programa principal debe imprimirse la lista de bonificaciones actualizada mediante el procedimiento ya escrito `imprimirListaBonificaciones()`.

A continuación tenéis un ejemplo de cómo se seleccionan las bonificaciones de la lista para efectuar la recaudación dentro de la función `recaudarBonificaciones()`. Si la lista de bonificaciones es [10 20 15 100.8 40] y el taquillero quiere retirar 55 euros de bonificaciones, deberían realizarse los siguientes pasos:

- Retirar el mínimo de la lista (10). La bonificación acumulada será de 10 euros y la lista actualizada será [20 15 100.8 40].
- Retirar el mínimo de la lista (15). La bonificación acumulada será de 25 euros y la lista actualizada será [20 100.8 40]

- Retirar el mínimo de la lista (20). La bonificación acumulada será de 45 euros y la lista actualizada será [100.8 40]
- Retirar el mínimo de la lista (40). Como la suma acumulada (85) sobrepasa la cantidad solicitada, modificamos esa bonificación retirando la cantidad necesaria (10) y dejando el resto (30), modificando por tanto esa bonificación en la lista de bonificaciones. Después de realizar estas operaciones, la bonificación acumulada será por tanto de 55 euros y la lista actualizada será [100.8 30].

(d) Obtener estadísticas: se mostrará por pantalla el número de clientes que ha atendido el taquillero y el número de clientes que están esperando a ser atendidos, el total de bonificaciones que se han recaudado previamente y el total de bonificaciones que tiene disponible para recaudar. Para recopilar alguno de estos valores tendréis que añadir el parámetro `TIPOELEMENTOLISTA *totalBonificacionesRecaudadas` a la función `recaudarBonificaciones()`, así como definir las variables `bonificacionesDisponibles`, `clientesAtendidos` y `clientesEnEspera`, que debéis mantener a lo largo del programa.

(e) Salir: se pedirá confirmación antes de abandonar el programa y se liberará toda la memoria reservada, para lo que tendréis que escribir las funciones `void liberarListaBonificaciones(TLISTA *listaBonificaciones)`, que libera la memoria de la lista, y `void liberarColaEspera(TCOLA *colaEspera)`, que libera la memoria de la cola. Debéis tener en cuenta que, como el TAD cola no tiene implementado un procedimiento propio para destruir la cola, tendréis que recorrerla eliminando todos los elementos antes de liberar la variable.

5. Uso de TAD

El programa deberá implementar obligatoriamente la lista y la cola indicadas con los TADs correspondientes, cuya descripción y funcionalidades se han explicado en clase de teoría.

Antes de iniciar el desarrollo de la práctica debéis repasar detalladamente las diapositivas y notas de teoría de los temas correspondientes y de forma especial los ejemplos, para recordar el funcionamiento del TAD.

Para la lista de bonificaciones se utilizará el TAD lista implementado en los ficheros adjuntos (listas.c y .h). Para la cola de espera se utilizará el TAD cola implementado en los ficheros adjuntos (cola.c y .h).

NO PODRÁ REALIZARSE NINGUNA MODIFICACIÓN SOBRE LOS FICHEROS DE LOS TAD PROPORCIONADOS (ni en los .c, ni en los .h).

6. Temporización recomendada

FASE 1: PREPARACIÓN DEL PROYECTO Y MENÚS

En primer lugar, debéis crear un proyecto en Netbeans (o en otro IDE) con el nombre Cineplus (por ejemplo) y decirle que sí queréis crear main (fijaos que a la derecha esté seleccionado C, no C++). Netbeans crea su propio makefile, pero podéis sobreescribirlo y funcionará igualmente la compilación (si queréis que funcione la ejecución, el archivo de salida debe estar en la carpeta ./dist/Debug/Cygwin-Windows, lo que debéis indicar al definir la variable OUTPUT o EJECUTABLE del makefile) desde el IDE. Podéis abrir el archivo Makefile de Netbeans desde el propio IDE, es cómodo hacerlo así y no editarlo por fuera. Una vez que Netbeans crea la carpeta de proyecto, **debéis crear dentro la carpeta TAD**, donde copiaréis los archivos que os pasamos en el campus virtual con la práctica y que están dentro de TAD.zip: listas.h, listas.c, cola.h y cola.c.

Corregid el makefile para tener en cuenta los ficheros a compilar, enumerando correctamente los archivos .c y los .h y también indicando correctamente la carpeta en la que están.

Una vez preparado el proyecto, trabajaréis en el archivo main.c. Lo primero que debéis hacer es crear las estructuras de datos que usaréis a lo largo del programa: **cola de clientes y lista de bonificaciones** (recordad inicializar a NULL los punteros al declarar las variables), y a continuación crear el menú con todas las opciones (a-e), aunque todavía no hagáis nada en cada opción. También podéis implementar las funciones de `liberarListaBonificaciones()` y `liberarColaEspera()` que se indicaron en la sección 4, operaciones que deben realizarse al salir del programa.

El proyecto debe compilar correctamente y debe poder ejecutarse desde el terminal.

FASE 2: PREPARACIÓN DE LA COLA DE CLIENTES

Como segundo paso, podéis implementar la primera opción de ponerse a la cola, recordando que debéis comprobar que el número de entradas sea válido y que debéis imprimir la cola cada vez que se modifique, por lo que será útil escribir los procedimientos `void imprimirPrimeroCola()` y `int comprobarNumeroEntradas()` que se describieron en la sección 4.

FASE 3: ATENDER CLIENTES

Si la cola NO está vacía, debéis procesar su primer elemento: leerlo, eliminarlo de la cola e imprimirlo por pantalla. Como la cola se ha modificado, debéis imprimir por pantalla el valor del nuevo primer elemento con el procedimiento ya creado imprimirPrimeroCola(). Llegados a este punto, tendréis que empezar a introducir en la lista de bonificaciones del taquillero las bonificaciones correspondientes al número de entradas que ha vendido. Tendréis que calcular la bonificación generada (es recomendable definir una constante BONIFICACION de valor 0.6) e insertar ese valor (numEntradas*BONIFICACION) AL FINAL de la lista. Debéis imprimir el valor de la bonificación generada e imprimir la lista completa de bonificaciones del taquillero escribiendo el procedimiento `void imprimirListaBonificaciones()` descrito en la sección 4.

El proyecto debe compilar correctamente y debe poder ejecutarse desde el terminal.

FASE 4: COBRAR BONIFICACIONES

Básicamente tendréis que implementar los procedimientos

`int comprobarValorARecaudar(),`

`TIPOELEMENTOLISTA totalBonificaciones(TLISTA listaBonificaciones)`

y

`void recaudarBonificaciones(TLISTA *listaBonificaciones, TIPOELEMENTOLISTA dineroARecaudar)`

explicados en la sección 4 (el tercer parámetro que aparece en esa sección es para la siguiente fase del proyecto) y en el procedimiento recaudarBonificaciones() contemplar lo que se indica en el apartado (c) de la sección 4.

El proyecto debe compilar correctamente y debe poder ejecutarse desde el terminal.

FASE 5: ESTADÍSTICAS

Tendréis que añadir variables que sumaricen todos los movimientos del taquillero: número de clientes que ha atendido, número de clientes que están esperando para ser atendidos, total de bonificaciones recaudadas y total de bonificaciones disponibles. Para ello es posible que tengáis que modificar alguna función, como `void recaudarBonificaciones(TLISTA *listaBonificaciones, TIPOELEMENTOLISTA dineroARecaudar, TIPOELEMENTOLISTA *totalBonificacionesRecaudadas)`, a la que se le añade el último parámetro como se indicó en la sección 4.

FASE 6: INICIALIZACIÓN DE LA COLA POR LÍNEA DE COMANDOS

Como se indicó al inicio del apartado 4, la cola podrá inicializarse directamente desde la línea de comandos, utilizando para ello los argumentos de main(). Los números deben ser enteros ya que corresponderán al número de entradas de cada cliente en la cola de esa taquilla. Esta parte es independiente del menú, y debe realizarse previamente al mismo.

7. Entregables

Deberá realizarse la entrega por el Campus Virtual. Las fechas de entrega se especifican en el Campus Virtual, y las instrucciones para generar el fichero son las siguientes:

- Deberá subirse un único fichero comprimido con el nombre apellido1_apellido2 y la extensión correspondiente.
- Se incluirá ÚNICAMENTE el fichero main.c y el correspondiente makefile. Para generar el makefile se asumirá que los ficheros de los TAD necesarios estarán en una carpeta denominada TAD, que se encontrará en el mismo directorio que main.c y el makefile.
- No se incluirá NINGUNA carpeta dentro del fichero comprimido.

Cualquier ejercicio que no compile directamente con el makefile (sin opciones) en Linux/Cygwin será evaluado con la calificación de 0. Este criterio se mantendrá en el resto de prácticas de la asignatura.

8. Ejemplo de ejecución

```
./cineplus 3 5 4

--COLA DE ESPERA--
   El primer cliente quiere 3 entradas

---BIENVENIDA AL CINE PLUS---
(a) Ponerse a la cola
(b) Atender cliente
(c) Cobrar bonificaciones
(d) Calcular estadísticas
(s) Salir
--------------------
Opcion: a
Introduzca el numero de entradas: 1
-- COLA DE ESPERA --
   El primer cliente quiere 3 entradas

---BIENVENIDA AL CINE PLUS---
(a) Ponerse a la cola
(b) Atender cliente
(c) Cobrar bonificaciones
(d) Calcular estadísticas
(s) Salir
--------------------
Opcion: b
Se ha atendido a un cliente con 3 entradas, generando una bonificacion
de 1.80 euros
-- COLA DE ESPERA --
   El primer cliente quiere 5 entradas
-- LISTA DE BONIFICACIONES --
   1.80

---BIENVENIDA AL CINE PLUS---
(a) Ponerse a la cola
(b) Atender cliente
(c) Cobrar bonificaciones
(d) Calcular estadísticas
(s) Salir
--------------------
Opcion: b
Se ha atendido a un cliente con 5 entradas, generando una bonificacion
de 3.00 euros
-- COLA DE ESPERA --
   El primer cliente quiere 4 entradas
```

```
-- LISTA DE BONIFICACIONES --
   1.80 3.00

---BIENVENIDA AL CINE PLUS---
(a) Ponerse a la cola
(b) Atender cliente
(c) Cobrar bonificaciones
(d) Calculars estadísticas
(s) Salir
--------------------
Opcion: b
Se ha atendido a un cliente con 4 entradas, generando una bonificacion
de 2.40 euros
-- COLA DE ESPERA --
   El primer cliente quiere 1 entradas
-- LISTA DE BONIFICACIONES --
   1.80 3.00 2.40

---BIENVENIDA AL CINE PLUS---
(a) Ponerse a la cola
(b) Atender cliente
(c) Cobrar bonificaciones
(d) Calculars estadísticas
(s) Salir
--------------------
Opcion: b
Se ha atendido a un cliente con 1 entradas, generando una bonificacion
de 0.60 euros
-- COLA DE ESPERA --
   -Sin clientes-
-- LISTA DE BONIFICACIONES --
   1.80 3.00 2.40 0.60

---BIENVENIDA AL CINE PLUS---
(a) Ponerse a la cola
(b) Atender cliente
(c) Cobrar bonificaciones
(d) Calculars estadísticas
(s) Salir
--------------------
Opcion: c
Introduzca el valor que desea recaudar: 3.5
Se ha recaudado el dinero solicitado (3.50 euros)
-- LISTA DE BONIFICACIONES --
   3.00 1.30
```

```
---BIENVENIDA AL CINE PLUS---
(a) Ponerse a la cola
(b) Atender cliente
(c) Cobrar bonificaciones
(d) Calcular estadísticas
(s) Salir
--------------------
Opcion: d
Se han recaudado un total de 3.50 euros en bonificaciones
El total de bonificaciones disponibles es de 4.30 euros
Se han atendido 4 clientes

---BIENVENIDA AL CINE PLUS---
(a) Ponerse a la cola
(b) Atender cliente
(c) Cobrar bonificaciones
(d) Calcular estadísticas
(s) Salir
--------------------
Opcion: s
Realmente desea salir de la aplicacion? [s/n] s
Saliendo del programa
```