

# Facultad de Informática – Universidad Complutense de Madrid

## Fundamentos de la programación I – Grupos XXX

### Examen de la Convocatoria Extraordinaria – 2020-2021

Tiempo disponible: 2 horas y 30 minutos

Se quiere construir un programa en C++ que gestione las reservas de un restaurante durante un turno de comidas. El restaurante puede disponer de un máximo de 20 mesas, en terraza o en interior, cada una con una capacidad de 4, 6 u 8 comensales.

Los datos sobre las mesas del restaurante se encuentran en el archivo `mesas.txt`. Cada línea corresponde con una mesa y registra la capacidad de la mesa y una letra mayúscula que indica si está en la terraza (T) o en el interior (I). La última línea contiene -1 como capacidad (centinela).

En otro archivo, con nombre `reservas.txt`, tenemos las reservas que recibirá el restaurante. Cada línea empieza por una R (si se trata de una reserva) o una C (si se trata de una cancelación), y las iniciales del cliente (cadena de una sola palabra). Si es una reserva, seguirán el número de comensales y una T o una I (terraza o interior). La última línea del archivo tiene una X (centinela). Ejemplos de archivos:

`mesas.txt`

```
4 T
6 I
4 I
8 T
6 T
4 T
4 I
...
4 T
6 T
-1
```

`reservas.txt`

```
R KNZ 3 I
R HJC 6 I
R YEO 3 T
C YEO
R WGJ 1 I
R LKG 5 T
R ZQV 3 T
C HJC
R FBT 8 T
...
X
```

El programa deberá, primero, cargar las mesas desde `mesas.txt`. A continuación leerá de `reservas.txt` cada reserva o cancelación:

- Si es una reserva, buscará una mesa adecuada en la terraza o en el interior, según se indique, y que sea adecuada para el número de comensales (de 1 a 4 comensales, en mesa de 4; 5 o 6 comensales en mesa de 6; 7 u 8 comensales en mesa de 8; y si son más de 8 comensales no se aceptará la reserva). Si hay

mesa disponible, se actualizarán sus datos en la lista de mesas (está reservada, cliente que la reservó y comensales).

- Si es una cancelación, se deberá localizar la mesa en la lista de mesas para eliminar los datos de la reserva en la lista de mesas.

El programa informará del resultado de procesar cada reserva o cancelación como se muestra en el ejemplo de ejecución. Además al finalizar el procesado de reservas.txt mostrará el estado final de las mesas (ver al final).

Empieza tu código con las declaraciones adecuadas para representar cada mesa (tMesa) (ubicación, capacidad, reservada y cliente que la reservó), la lista de mesas (tListaMesas), cada reserva o cancelación (tReserva) (tipo, cliente, comensales, ubicación y su estado (se ha podido realizar o no, o se ha cancelado)).  
**[1,5 puntos]**

El programa principal comenzará creando la lista de mesas a partir del archivo mesas.txt. Después, procesará el archivo reservas.txt, y se actualizará, convenientemente, la lista de mesas. El programa terminará mostrando el estado de las mesas. Asume que no hay errores en los archivos.

Implementa, al menos, los siguientes subprogramas:

- **[1 punto]** cargarMesas(): carga la información del archivo mesas.txt en la lista de mesas. Devuelve true si se pudo cargar llevar a cabo o false en otro caso. Si en el archivo hay más mesas que el máximo permitido, se ignorarán las adicionales y se devuelve true.
- **[0,5 puntos]** buscarCliente(): dados la lista de mesas y el acrónimo de un cliente, devuelve la posición en la lista de la mesa con la reserva de ese cliente o -1 si no hay ninguna reserva de ese cliente.
- **[0,5 puntos]** buscarMesa(): dados la lista de mesas, el número de comensales a sentar y la zona deseada (T o I), devuelve la posición en la lista de la primera mesa encontrada que es adecuada para una reserva con esos comensales en esa zona o -1 si no hay ninguna adecuada.
- **[3 puntos]** reservar(): dadas la lista de mesas y una reserva, busca una mesa adecuada y si se encuentra, la reserva. Además, mostrará si se ha podido reservar o si ha sido imposible (ver el ejemplo de ejecución).
- **[1 punto]** cancelar(): dadas la lista de mesas y una cancelación, intenta localizar la mesa reservada por el cliente y si se encuentra, cancela la reserva. Además, mostrará si se ha podido cancelar o si no se ha encontrado (ver el ejemplo de ejecución).

- **[1 punto]** `procesarReservas()`: Dada la lista de mesas, lee las solicitudes del archivo `reservas.txt` e intenta realizarlas mostrando su resultado por pantalla y actualizando el estado de la lista de mesas.
- **[1 punto]** `mostrar()`: Dada la lista de mesas, muestra una el estado de cada una de ellas (*ver el ejemplo de ejecución*).

*Recuerda que en el código que desarrolles se valorarán la corrección del diseño, la eficiencia del código, la ausencia de código repetido, la correcta utilización de los esquemas vistos en clase, etc. Recuerda que no se permite el uso de variables globales, ni de instrucciones de salto salvo `return` en las funciones.*

## Ejemplo de ejecución

```
Reservar en interior, KNZ, 3 pax : Confirmada mesa 3
Reservar en interior, HJC, 6 pax : Confirmada mesa 2
Reservar en interior, FPF, 2 pax : Confirmada mesa 7
Reservar en terraza, XZR, 7 pax : Confirmada mesa 4
Reservar en terraza, EQQ, 9 pax : Imposible
Reservar en interior, JYQ, 8 pax : Confirmada mesa 8
Reservar en interior, TER, 7 pax : Imposible
Reservar en terraza, YEO, 3 pax : Confirmada mesa 1
Cancelar mesa de YEO : Cancelada mesa 1
Reservar en interior, WGJ, 1 pax : Confirmada mesa 10
Reservar en terraza, LKG, 5 pax : Confirmada mesa 5
Reservar en terraza, CXN, 12 pax : Imposible
Reservar en terraza, ACK, 8 pax : Confirmada mesa 13
Reservar en terraza, YPJ, 1 pax : Confirmada mesa 1
Reservar en terraza, YJR, 6 pax : Confirmada mesa 9
Cancelar mesa de ACK : Cancelada mesa 13
Cancelar mesa de CXN : Reserva no encontrada
Reservar en terraza, SWP, 6 pax : Confirmada mesa 16
Reservar en interior, YMU, 5 pax : Imposible
Cancelar mesa de BCD : Reserva no encontrada
Reservar en interior, OKS, 2 pax : Confirmada mesa 11
Reservar en terraza, ZQV, 3 pax : Confirmada mesa 6
Cancelar mesa de HJC : Cancelada mesa 2
Reservar en terraza, FBT, 8 pax : Confirmada mesa 13
Reservar en interior, CKI, 6 pax : Confirmada mesa 2
Reservar en interior, VIN, 2 pax : Imposible
```

Lista de mesas...

```
Mesa 1 (T4) : Reservada a YPJ (1)
Mesa 2 (I6) : Reservada a CKI (6)
Mesa 3 (I4) : Reservada a KNZ (3)
Mesa 4 (T8) : Reservada a XZR (7)
Mesa 5 (T6) : Reservada a LKG (5)
Mesa 6 (T4) : Reservada a ZQV (3)
Mesa 7 (I4) : Reservada a FPF (2)
Mesa 8 (I8) : Reservada a JYQ (8)
Mesa 9 (T6) : Reservada a YJR (6)
Mesa 10 (I4) : Reservada a WGJ (1)
Mesa 11 (I4) : Reservada a OKS (2)
Mesa 12 (T4) : No reservada
Mesa 13 (T8) : Reservada a FBT (8)
Mesa 14 (T4) : No reservada
Mesa 15 (T4) : No reservada
Mesa 16 (T6) : Reservada a SWP (6)
```

