Enunciado de la práctica 1:

GESTIÓN DE RESERVAS DE UN RESTAURANTE

El objetivo de esta práctica es simular el funcionamiento en la gestión de reservas, mesas y pedidos, durante los tres turnos de comida de un restaurante.

1. Cada una de las reservas incluye los siguientes datos:
    - Nombre del cliente.
    - Situación de la mesa: Terraza o Interior.
    - Número de personas: máximo 8 personas por mesa.
    - Hora de la reserva: 13:00, 14:00 o 15:00.
    - Preferencia de menú: vegano, sin Gluten, completo.

    Las reservas se almacenan utilizando dos colas, *cReservas* para las reservas realizadas por los clientes y *cPendientes* que irá almacenado las reservas pendientes de ser gestionadas.

1. De cada una de las mesas del restaurante se tienen los siguientes datos:
    - Número de la mesa: de 1 a 20.
    - Capacidad: 4 u 8 personas.
    - Situación: Terraza/Interior.

    Las mesas libres del restaurante se almacenan en una pila *pMesas*.

2. Para cada pedido se tienen los siguientes datos:
    - Número de la mesa (si el número de personas de la reserva es mayor de 4 se le asignará una de 8, teniendo en cuenta la situación indicada en la reserva).
    - Nombre del cliente.
    - Número de personas.
    - Preferencia de menú.
    - Situación de la mesa.
    - Finalizado.

    Los pedidos se almacenarán en otra cola llamada *cPedidos*.

Funcionamiento de la gestión de reservas, mesas y pedidos:

El programa comienza generando la cola *cReservas* con las *reserva*s realizadas por los clientes y la pila *pMesas* de mesas libres. A continuación, se simulará la gestión de las reservas de *cReservas* y creación de la *cPedidos* del restaurante, todo ello a través de un menú de opciones.

- Para cada reserva que sale de *cReservas,* se asigna una mesa, se genera el pedido y se inserta en *cPedidos.*
- Para asignar la mesa, se busca una mesa en *pMesas* con la capacidad y situación reservadas y se elimina de la misma (si la mesa que está encima de la pila no se corresponde con la reservada, se busca una más abajo dejando las mesas no asignadas como estaban situadas en la pila). Si no hay una mesa libre que pueda ser asignada, la reserva se almacena en la cola de reservas pendientes *cPendientes.*

- Por *cada dos reservas* que salen de *cReservas*, se comprobará si puede gestionarse una de *cPendientes* (ya no se tendrá en cuenta la hora de reserva), generando y almacenando el pedido correspondiente en *cPedidos*.
- Una vez finalizadas todas las reservas de una hora, y antes de comenzar con las reservas de la siguiente, se marcarán como finalizados la mitad de los pedidos de *cPedidos*, simulando que quedan libres las mesas correspondientes (que se apilarán en la pila *pMesas*).
- Una vez finalizadas las reservas de *cReservas*, se gestionan las de la *cPendientes*, si las hay.
- El programa finaliza cuando se han gestionado todas las reservas de ambas colas o no puedan gestionarse las reservas en *cPendientes*.

El **programa** mostrará un **Menú con las siguientes opciones**:

1. Generar aleatoriamente la *cReservas* antes de la apertura del restaurante (en la cola se incluirán, al menos y en este orden, cuatro reservas para las 13:00, cuatro para las 14:00 y cuatro para las 15:00, el resto de los datos en cada reserva será aleatorio).
2. Mostrar en pantalla los datos de la *cReservas.*
3. Borrar la *cReservas.*
4. Generar aleatoriamente la *pMesas* (al menos ocho serán en terraza, la capacidad será aleatoria).
5. Mostrar en pantalla los datos de la *pMesas.*
6. Borrar la *pMesas.*
7. Simular la gestión de la primera reserva de la cola *cReservas*, según se ha detallado en el funcionamiento del programa, creando el pedido del cliente si es posible e insertándolo *en cPedidos.* Una vez gestionada la reserva, se mostrarán[1] en pantalla ambas colas de reservas, la pila de mesas libres y la cola de pedidos.
8. Simular la gestión de las reservas y creación de pedidos de todas las reservas para la misma hora, al principio de *cReservas*, según se ha detallado en el funcionamiento del programa. Una vez que se gestionan todas las reservas de una hora, se mostrarán en pantalla ambas colas de reservas, la pila de mesas libres y la cola de pedidos.
9. Simular la gestión de todas las reservas del restaurante, según se ha detallado en el funcionamiento del programa, hasta que se hayan gestionado todas las reservas de ambas colas o no puedan gestionarse las reservas en *cPendientes*. Al finalizar, se mostrarán en pantalla ambas colas de reservas, la pila de mesas y la cola de pedidos.
0. Salir

Se deben implementar, **utilizando Programación Orientada a Objetos y memoria dinámica**, al menos, los TAD´s ***Pila y Cola*** (con las operaciones habituales y otras que consideréis necesarias).

OBSERVACIONES:

Aquí se está describiendo el funcionamiento básico del programa y las estructuras correspondientes, los alumnos deben tomar decisiones sobre los datos, su implementación, la interface del programa, etc, que deben ser justificadas en la documentación.

[1] Es decisión de cada grupo la forma en que se mostrará en pantalla el contenido de cada estructura, siempre que se muestre la información con claridad.

NORMAS PARA LA REALIZACIÓN Y ENTREGA DE LAS PRÁCTICAS

Las prácticas se realizarán en grupos de dos alumnos **que deberán ser los mismos para las dos prácticas** de la asignatura. Deberán **confirmarse** las personas que forman el grupo durante la clase **del día 16 de octubre** (o email al profesor).

1. La práctica se implementará en C++, utilizando CodeBlocks. Debe entregarse un fichero comprimido incluyendo **todos los ficheros fuente del proyecto** C++ y el documento descrito en el punto 4. Se subirá **un fichero por grupo** a la plataforma **antes de la fecha indicada.** El nombre del fichero será el nombreapellido1_nombreapellido1 de ambos miembros del grupo.
2. En la **defensa** de la práctica **se verificará la autoría de la práctica entregada y será calificada con APTO/NO APTO, siendo necesaria la calificación de APTO para poder ser evaluado de la práctica**.
3. La entrega de prácticas copiadas supondrá el suspenso de la asignatura en esta convocatoria para todos los alumnos implicados.
4. La documentación que se subirá en un fichero *.pdf* junto con el proyecto (tendrá un peso del 10% de la nota), deberá tener al menos los siguientes apartados:
    a. Nombre y DNI de los alumnos del grupo.
    b. Detalles y justificación de la implementación:
        b.1 Especificación concreta de la interfaz de los TAD´s implementados:
            b.1.1 TAD`s creados.
            b.1.2 Definición de las operaciones del TAD (Nombre, argumentos y retorno).
        b.2 Solución adoptada: descripción de las dificultades encontradas.
        b.3 Diseño de la relación entre las clases de los TAD implementados.
            b.3.1 Explicación de los métodos más destacados.
            b.3.2 Explicación del comportamiento del programa.
        b.4 Bibliografía.

FECHAS

La práctica 1 se entregará antes del **27 de octubre a las 18:59 horas**. Para la **defensa**, individual y obligatoria para ser calificado, es necesario traer la práctica en el portátil de uno de los alumnos o en un pen drive (en caso de no traer el portátil). La defensa se realizará durante la clase del **30 de octubre**.

EN CASO DE NO PRESENTARSE A LA DEFENSA NO SE CALIFICARÁ LA ENTREGA.