



Primer parcial – 15 de octubre de 2015

Nombre y Apellido:

Calificación:

Ej. 1 (4 ptos.): Desarrollar un subprograma recursivo que elimine todos los datos de una pila que sean iguales a un valor recibido por parámetro. La pila debe mantener su orden original y el subprograma debe retornar la cantidad de elementos eliminados.

- Implementar con TDA pila estática definiendo el tipo y desarrollando los operadores faltantes *poneP* y *vaciaP* del archivo provisto por la cátedra

Ej. 2 (6 ptos.): Un restaurante almacena en una lista simplemente enlazada los mozos que tienen mesas pendientes de cerrar con la siguiente estructura:

- Id Mozo (Cadena de 5) (no ordenada)
- Sublista de mesas pendientes de cerrar
 - Nro de Mesa (ordenada)
 - Cantidad de comensales

Implementar subprogramas para:

- cerrar una mesa, solicitando mozo y número de mesa; esto implica la eliminación de la misma de la sublista. Considerar la eliminación del mozo de la lista si quedase sin mesas pendientes de cierre.
- Al finalizar, mostrar un listado con el siguiente diseño:

Listado de Mozos		
Id Mozo	Mesas sin cerrar	Mesas con un solo comensal
xxxxx	99	99
xxxxx	99	99

[OPCIONAL] Desarrollar un programa principal que cargue la lista ingresando las mesas pendientes de cerrar de forma ordenada ascendente (desde archivo de texto o teclado) e invoque a los subprogramas desarrollados, además debe mostrar la lista antes y después de estas invocaciones.

Nota: Puede usar el archivo de texto provisto por la cátedra. La estructura del archivo es la siguiente (un mozo por línea, cada dato separado por un espacio):

idMozo CantidadMesas Nro.Mesa1 Comensal1 Nro.Mesa2 Comensal2

Serán considerados al calificar este examen la eficiencia de las soluciones y del uso de las características del lenguaje C y de la programación estructurada.

Para aprobar es necesario obtener al menos 5 puntos, y al menos el 50% del puntaje de cada ejercicio.