



## Práctica 6 - Listas

1. Generar una lista simplemente enlazada con el contenido de un arreglo de N enteros desordenados:
  - a) invirtiendo el orden de sus elementos (no recorrer el arreglo comenzando por el último elemento)
  - b) manteniendo el orden de sus elementos.
  - c) de forma ordenada
2. Dada una lista simplemente enlazada de caracteres, escribir una función o “procedimiento” (*void*) que:
  - a) muestre su contenido.
  - b) devuelva la cantidad de nodos que poseen vocales.
  - c) indique si la lista está ordenada.
  - d) invierta sus elementos. No se pueden generar nuevos nodos.
  - e) elimine el elemento ubicado en una posición P (entero) pasada como parámetro.
3. Escribir un programa que lea desde un archivo un texto que contiene una cantidad desconocida de palabras (una en cada línea) y genere una lista ordenada de todas las palabras existentes en el texto con su frecuencia de aparición.
4. Corregir los errores en el siguiente programa, el cual incorpora en una lista ordenada de números los elementos de un arreglo de N números ordenados, respetando el orden de la lista.

```
#define N 2
...
int main(){
    int i, arr[N] = {4,8,9,60};
    TLISTA lista=NULL, pult=NULL, act, ant, nuevo;
    cargarLista(&lista);
    printf("\nLista Original\n");
    muestra(&lista);

    act = lista;   ant = NULL;
    for(i=0; i<=N; i++) {
        while(act!=NULL && arr[i] <= act->n) {
            act = ant;   act = act->sig;
        }
        *nuevo = (TLISTA)malloc(sizeof(NODO));
        nuevo->n = arr[i-1];
        if(ant == NULL) {
            nuevo->sig = lista; *lista = &nuevo;
        } else {
            nuevo->sig = ant; ant->sig = &nuevo;
        }
        act = nuevo->sig;
    }
    printf("\nLista Resultado\n");
    muestra(&lista);
    return 0;
}
```

5. Dadas dos listas simplemente enlazadas de enteros L1 y L2:
  - a) verificar si son iguales. Considerar que están ordenadas.
  - b) añadir L2 al final de L1, en el caso de que el primer elemento de L2 sea mayor al último de L1. En otro caso, añadir L1 al final de L2.
  - c) eliminar de L1 los nodos cuyas posiciones se encuentran en L2, la cual está ordenada.



6. A partir de una lista ordenada de enteros, generar dos listas de salida: una conteniendo los valores impares ordenados descendentemente, y la segunda con los valores pares ordenados de manera ascendente. No se deberá crear nuevos nodos, sino utilizar los de la lista original generando nuevos enlaces.
7. Una lista contiene en cada nodo una cola de números enteros. Desarrollar un programa que genere una pila con los elementos máximos de cada cola de la lista. (Utilizar TDA pila dinámica y TDA cola circular).
8. Se tiene una lista con equipos de fútbol (nombre y puntaje en el torneo), la cual está ordenada descendentemente por puntaje. Cada equipo cuenta con una sublista de jugadores (nombre y edad).
  - a) Listar para todos los clubes los jugadores cuyo nombre comienza con la letra K (dato). Al finalizar mostrar cantidad de jugadores listados y el club que tiene más jugadores que cumplan el criterio solicitado.
  - b) Listar los clubes que tienen un mínimo de X (dato) puntos junto con todos sus jugadores, mostrando edad promedio de cada equipo.
9. Simular un gestor de archivos simplificado mediante listas de carpetas que contengan sublistas de carpetas y archivos. Implementar el listado del contenido de una carpeta ingresada como dato. No es necesario listar el contenido de las subcarpetas de dicha carpeta.
10. Desarrollar un programa que gestione las reservas de pasajes para los vuelos de una aerolínea. Para ello se cuenta con una lista de destinos con la siguiente estructura:
  - Destino
  - Sublista ordenada de Vuelos (a dicho destino)
    - Código de vuelo
    - Fecha y horario de salida
    - Capacidad de pasajeros
    - Sublista de pasajes vendidos (para dicho vuelo)
      - Documento del pasajero

Tanto para la reserva como para su cancelación se solicita: destino, código de vuelo y documento del pasajero. El programa debe además permitir agregar destinos y vuelos. Cada vuelo tiene una fecha y horario de salida y capacidad de pasajeros. El programa debe también listar el estado (cantidad de reservas, lugares disponibles, etc.) de todos los vuelos de un destino solicitado al operador.

NOTA: Dado que no se requieren cálculos con las fechas y horarios de salida, se sugiere almacenar dicho campo como una cadena.

11. Implementar un programa de biblioteca que administre la siguiente estructura: un arreglo de listas simples ordenadas alfabéticamente con los autores que comiencen con la misma letra, o sea que habrá un elemento del arreglo por cada letra del abecedario, aunque no existan autores que comiencen con dicha letra. Para cada autor existe una sublista, ordenada por título, con los libros de cada autor:
  - Autor
  - Sublista de libros de dicho autor
    - Título
    - Año de edición

Se debe poder agregar libros a esta estructura.

Además, gestionar una lista de socios con una sublista de libros prestados. Permitir las operaciones de registrar préstamos (agregar nodo al final de la sublista de libros prestados para



el socio) y devoluciones (eliminar nodo del libro prestado de dicha sublista). Dicha lista posee la siguiente estructura:

- Socio
- Sublista de libros prestados
  - Título
  - Año de edición

12. Dada una lista doblemente enlazada que contiene los inscriptos a un curso, en la que en cada nodo se tiene:

- Apellido (ordenada por este campo)
- Nombre
- Categoría (docente, alumno, externo)

- a) listarlos en orden alfabético, indicando el importe total recaudado. El valor del curso es un dato leído por teclado, los docentes tienen un 30% de descuento, y los alumnos un 10%.
- b) listar los inscriptos externos en orden alfabético descendente.
- c) verificar si existe una persona con apellido X (dato) dentro de la lista de inscriptos

13. A partir de una lista circular con los DNI (no están ordenados) y sexo de los votantes de una mesa:

- a) listar los votantes femeninos.
- b) contar los votantes de cada sexo.
- c) buscar un DNI, que es dato, en la lista de votantes.

14. Copiar el contenido completo de una lista simplemente enlazada en una lista doblemente enlazada y en una lista circular. En ambos casos se debe mantener el orden de la lista original. La lista original no debe ser modificada.

15. Resolver mediante subprogramas las siguientes operaciones sobre una lista doblemente enlazada:

- a) insertar un elemento en la posición media. Si la cantidad de elementos de la lista es impar insertar en la posición media más uno. Considerar la inserción en una lista vacía (único nodo)
- b) eliminar un elemento conociendo la posición (puntero) del anterior, nulo para eliminar el primer elemento.

16. Corregir los errores de la siguiente función, que elimina el nodo de una lista circular que contiene un valor pasado como parámetro (elem)

```
void lc_eliminaelem(TLISTAC *pult, TELEMENTOLC elem, int *eliminado) {
    TLISTAC ant, act = *pult->sig;
    eliminado = 0;
    if (pult = NULL) {
        do{
            ant = act; act->sig = act;
        }while ( *pult != act && elem < act.dato);
        if (elem = act->dato){
            if (*pult != (*pult)->sig)
                *pult= NULL;
            else {
                act->sig = ant->sig;
                if (act == *pult)
                    pult = *ant;
            }
            free (ant);
            eliminado = 1;
        }
    }
}
```



}

17. Resolver mediante subprogramas las siguientes operaciones sobre una lista circular:
  - a) insertar un elemento conociendo la posición (puntero) del anterior, nulo para insertar al principio de la lista.
  - b) insertar un elemento conociendo la posición (puntero) del siguiente, nulo para insertar al final de la lista.
18. Rehacer los ejercicios 2 y 3 desarrollados para listas simplemente enlazadas, modificando su implementación como:
  - a) lista doblemente enlazada.
  - b) lista circular. (excluir ej. 2 d) invertir elementos sin crear nodos)
19. A partir de la lista de equipos de fútbol con sublista de jugadores del ejercicio 8, crear una lista doblemente enlazada con los equipos que tengan algún jugador mayor de 30 años. Para cada equipo registrar cantidad de jugadores y edad promedio. La estructura de cada nodo de la lista doble es la siguiente:
  - Nombre del equipo (ordenada por este campo)
  - Cantidad de jugadores
  - Edad promedio