



Práctica 6 - Listas

1. Dada una lista simplemente enlazada de caracteres, escribir una función *void* que:
 - a) muestre su contenido.
 - b) devuelva la cantidad de nodos que poseen vocales.
 - c) indique si la lista está ordenada.
 - d) elimine el elemento ubicado en una posición P (entero) pasada como parámetro.
2. Generar una lista simplemente enlazada a partir del contenido de un arreglo de N enteros no ordenados:
 - a) invirtiendo el orden de sus elementos (no se permite recorrer el arreglo comenzando por el último elemento)
 - b) manteniendo el orden de sus elementos.
 - c) de forma ordenada
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
...
void 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);
}
```

5. 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).



6. Se tiene una lista con equipos de fútbol (nombre y puntaje en el torneo), la cual está ordenada descendientemente por puntaje (los equipos con el mismo puntaje están ordenados alfabéticamente). Cada equipo cuenta con una sublista de jugadores (nombre y edad, estado ([A]ctivo, [S]uspendido)).
 - 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.
 - c) Verificar si el equipo E (dato) tiene P (dato) puntos.
 - d) Eliminar los jugadores que se encuentren en estado inactivo en todos los equipos.
7. 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 disponibles para ser prestados de cada autor:
 - Autor
 - Sublista de libros de dicho autor
 - Título
 - Autor
 - 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 (mover el nodo de la sublista del autor al final de la sublista de libros prestados para el socio) y devoluciones (mover nodo del libro prestado de dicha sublista a la sublista del autor de acuerdo al orden indicado). Dicha lista posee la siguiente estructura:

- Socio
 - Sublista de libros prestados
 - Título
 - Autor
 - Año de edición
8. Rehacer el ejercicio 1 desarrollado para listas simplemente enlazadas, modificando su implementación como:
 - a) lista doblemente enlazada.
 - b) lista circular
 9. Copiar el contenido completo de una lista simplemente enlazada en una lista doblemente enlazada y en una lista circular, ambas inicialmente vacías. Ambas copias deben realizarse en la misma función. En ambos casos se debe mantener el orden de la lista original, siendo el último elemento de la lista simple el último de la circular. La lista original no debe ser modificada.
 10. Dada una lista doblemente enlazada de caracteres que representa una palabra, verificar si es palíndroma.



- 11.** 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;
        }
    }
}
```

- 12.** Una lista doble contiene en cada nodo un rango de números enteros [A,B] y una sublista de enteros. Generar una lista simple paralela a la lista doble donde cada nodo contenga el promedio de los valores de la sublista y la cantidad de elementos de la misma que no pertenecen al rango [A,B].