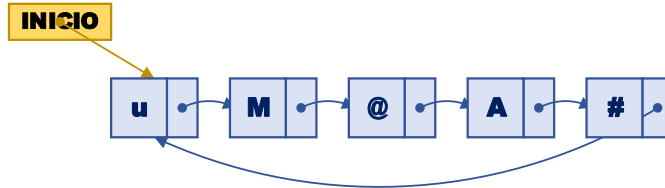


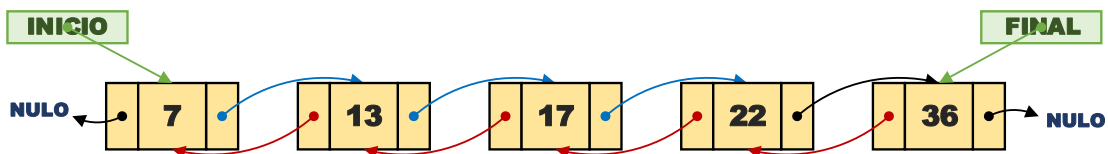
Tema: Listas Enlazadas, TDA Simple, TDA Pila

Apellido y Nombre: DNI:.....

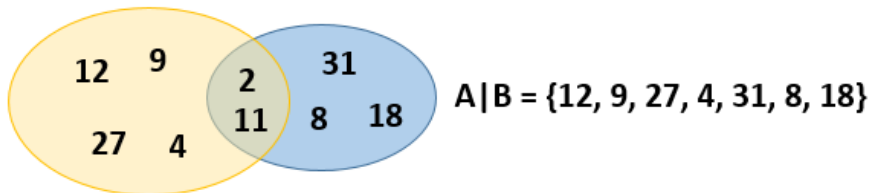
1. Dada la siguiente lista simple circular realice lo siguiente:



- Consigne la declaración de tipos y variables de la estructura.
 - Diseñe un procedimiento/función que permita *agregar al final* de la lista un nuevo nodo siempre y cuando éste no contenga un valor repetido. Considere que los nodos no agregados debe liberarse.
2. Dada una lista doble de enteros realice lo siguiente:



- Consigne la declaración de tipos y variables de la estructura, considerando que debe registrarse la cantidad de valores pares y cantidad de valores impares almacenados.
 - Diseñe un procedimiento/función que permita agregar, **en orden**, elementos a la lista realizando el recorrido de ésta **desde el último nodo hacia el primero**.
3. Un conjunto se define como una colección homogénea de elementos (no repetidos) donde el orden de éstos no es importante. Sabiendo esto, defina el *TDA conjunto de enteros* (implementado con **arreglos**) y desarrolle la operación **diferencia simétrica**. Dados los conjuntos A y B, la diferencia simétrica es el conjunto de elementos no comunes entre A y B.



NOTA: considere que dispone de las operaciones de conjunto *crear_vacío*, *agregar_elemento* y *pertenencia*, debiendo sólo invocarlas en el módulo propuesto.

- Considerando que únicamente se dispone de un arreglo (10 elementos) para implementar el *TDA pila* y suponiendo que la posición central de éste se utilizará como indicador de la misma, defina el TDA pila y desarrolle las operaciones *iniciar_pila*, *pila_llena* y *agregar_pila*.
- Utilizando el TDA pila y sus operaciones básicas diseñe un **módulo** que verifique la parentización de una cadena compuesta exclusivamente por paréntesis y llaves de apertura y cierre. Tenga en cuenta que el algoritmo debe verificar que cada símbolo de apertura cuente con su correspondiente símbolo de cierre.

Por ejemplo, la cadena `{ } () { () () }` está correctamente parentizada, mientras la cadena `{ } () { () () }` es incorrecta.

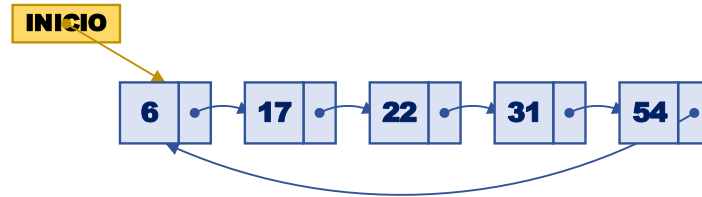
NOTA: No debe implementar las operaciones básicas de pila, sólo invocarlas.



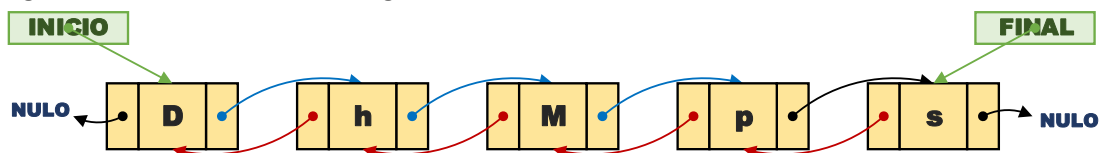
Tema: Listas Enlazadas, TDA Simple, TDA Pila

Apellido y Nombre: DNI:.....

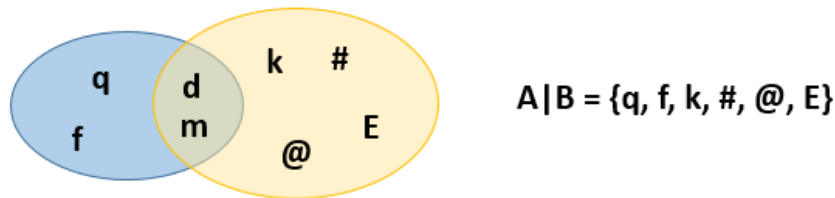
1. Dada la siguiente lista simple circular realice lo siguiente:



- Consigne la declaración de tipos y variables de la estructura.
 - Diseñe un procedimiento/función que permita *agregar al final* de la lista un nuevo nodo siempre y cuando éste contenga un valor mayor al último almacenado. Considere que los nodos no agregados debe liberarse.
2. Dada la siguiente lista doble realice lo siguiente:



- Consigne la declaración de tipos y variables de la estructura, considerando que debe registrarse la cantidad de mayúsculas y cantidad de minúsculas almacenadas.
 - Diseñe un procedimiento/función que permita quitar un valor específico la lista realizando el recorrido de ésta **desde el último nodo hacia el primero**.
3. Un conjunto se define como una colección homogénea de elementos (no repetidos) donde el orden de éstos no es importante. Sabiendo esto, defina el *TDA conjunto de caracteres* (implementado con **arreglos**) y desarrolle la operación **diferencia simétrica**. Dados los conjuntos *A* y *B*, la diferencia simétrica es el conjunto de elementos no comunes entre *A* y *B*.



NOTA: considere que dispone de las operaciones de conjunto *crear_vacio*, *agregar_elemento* y *pertenencia*, debiendo sólo invocarlas en el módulo propuesto.

4. Considerando que únicamente se dispone de un arreglo (12 elementos) para implementar el *TDA pila* y suponiendo que la posición central de éste se utilizará como indicador de la misma, defina el TDA pila y desarrolle las operaciones *iniciar_pila*, *pila_vacia* y *quitar_pila*.
5. Utilizando el TDA pila y sus operaciones básicas diseñe un **módulo** que verifique la parentización de una cadena compuesta exclusivamente por corchetes y paréntesis angulares de apertura y cierre. Tenga en cuenta que el algoritmo debe verificar que cada símbolo de apertura cuente con su correspondiente símbolo de cierre.

Por ejemplo, la cadena [] < > [< > < >] está correctamente parentizada, mientras la cadena [> <] [< > < >] es incorrecta.

NOTA: No debe implementar las operaciones básicas de pila, sólo invocarlas.

