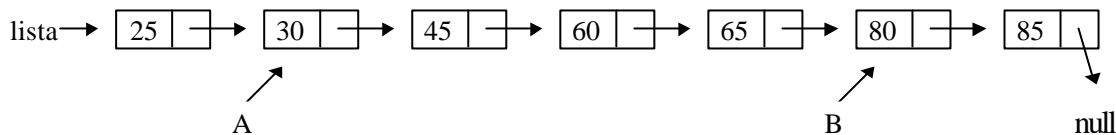


**Trabajo Práctico Nº 07**

**Tema: La Clase Lista**

Dada la lista enlazada dinámica:



1. Dar los valores de las siguientes expresiones:

|                                   |  |
|-----------------------------------|--|
| A.getInfo()                       |  |
| B.getSig().getInfo()              |  |
| lista.getSig().getSig().getInfo() |  |

2. Verdadero o falso?

|                            |  |
|----------------------------|--|
| lista.getSig() == A        |  |
| A.getSig().getInfo() == 60 |  |
| B.getSig() == null         |  |

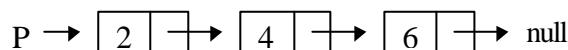
3. Escribir la sentencia para cada una de las siguientes tareas:

|   |  |
|---|--|
| Hacer que lista apunte al nodo que contiene 45. |  |
| Hacer que B apunte al último nodo de la lista.  |  |
| Hacer que lista apunte a una lista vacía.       |  |

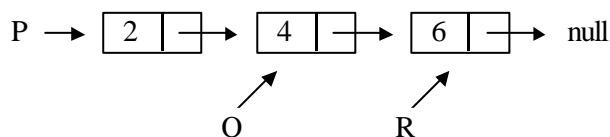
4. Utilizar las siguientes declaraciones para los ejercicios descritos a continuación:  
Nodo P, Q, R;

Mostrar en el diagrama lo que harían las sentencias:

- a) P = P.getSig();
- b) Q = P;
- c) R = P.getSig();

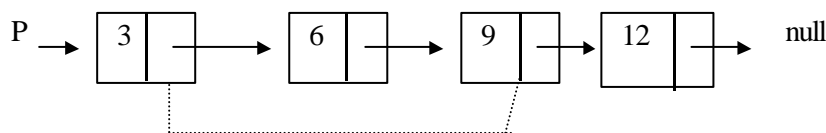


- d) P.setInfo(Q.getInfo());
- e) P.setInfo(Q.getSig().getInfo());
- f) R.setSig(P);

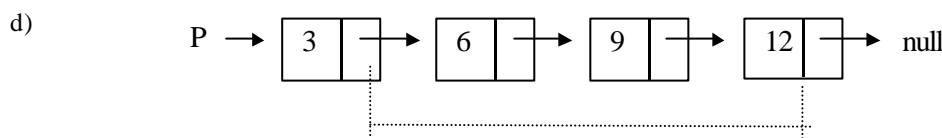
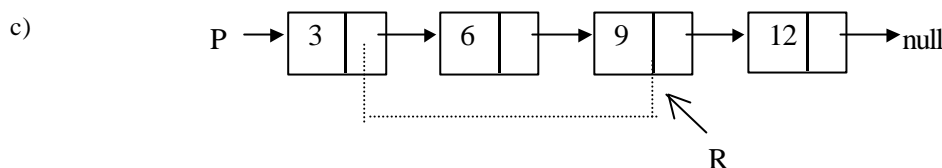
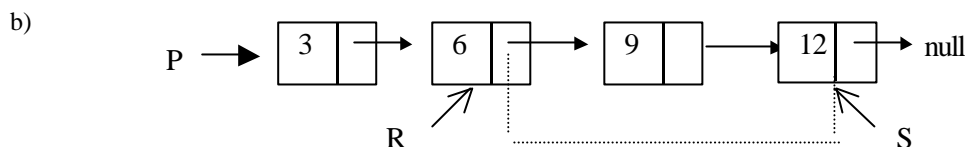


5. Escribir una sentencia para efectuar el cambio indicado en la línea de puntos

a)



**Trabajo Práctico Nº 07**



6. Implementar los siguientes métodos en la clase Lista y probar su correcto funcionamiento (los elementos son enteros y se insertan en forma ordenada)

- Escribir un método que busque un elemento en la lista y retorne verdadero o falso según corresponda.
- Escribir un método que busque un elemento en la lista y retorne un puntero al nodo, en caso verdadero o null en caso.
- Escribir un método que retorne la cantidad elementos de la lista.
- Escribir un método que retorne la sumatoria de los elementos de la lista.
- Escribir métodos que devuelvan el mayor y el menor valor, sin realizar comparaciones entre los valores.
- Escribir un método que muestre la lista en forma inversa (usar una pila)
- Escribir un método que muestre la lista en forma inversa (usar recursividad)
- Escribir un método que invierta la lista.
- Escribir un método que devuelva una copia de la lista.

7. Escribir la clase ListaDesordenada, con los métodos insertarPrimero (inserta siempre al principio) e insertarUltimo (inserta siempre al final) y los métodos restantes.(suprimir, mostrar, buscar, etc.)

8. Implementar los siguientes métodos en la clase ListaDesordenada y probar su correcto funcionamiento:

- Escribir métodos que devuelvan el mayor y el menor valor.
- Dada las formas de inserción es posible que la lista contenga elementos duplicados, escribir un método que elimine los duplicados.
- Escribir un método que intercambie dos elementos adyacentes de una lista enlazada, ajustando solo los punteros, no los datos.
- Escribir un método que reciba como parámetro otra lista y la concatene.
- Escribir un método que reciba como parámetro otra lista y la mezcle.

9. Realizar a la clase Lista (ordenada) el siguiente comportamiento:

- además del puntero al primer nodo, la clase tiene como atributo un puntero que es movable;
- actual: retorna la información apuntada por el puntero movable
- siguiente: mueve el puntero movable al nodo siguiente y retorna la información

### Trabajo Práctico Nº 07

- primero: mueve el puntero movable al primer nodo y retorna la información
- último: mueve el puntero movable al último nodo y retorna la información

10. Utilizando la implementación de la lista del ejercicio 9, escribir los métodos que generen listas a partir de las siguientes operaciones:

- a)  $L1 \cup L2$
- b)  $L1 \cap L2$
- c)  $L1 - L2$
- d)  $(L1 \cup L2) - L3$

11. Dadas dos listas enlazadas L1 y L2, que contienen enteros y están ordenadas, escribir un método Visualizar(...) que muestre los elementos de L1 que están en las posiciones dadas por L2. Por ejemplo, si  $L2 = 1, 3, 4, 6$  se mostrará los nodos primeros, tercero, cuarto y sexto de L1. (No hay elementos repetidos en ninguna de las listas, pero L2 pueden hacer referencia a nodos que no existan en L1).

12. Escribir un programa que genere dos listas Pol1 y Pol2, las cuales almacenan polinomios. Luego generar un tercer lista SUMA, que contenga la suma de Pol1 y Pol2. Mostrar las tres listas.

13. Escribir un programa tipo menú, que permita generar una lista, insertar elementos, eliminar elementos y recorrer la lista. Podrán almacenarse elementos repetidos, pero para no desperdiciar espacio en memoria, se agrega un atributo al nodo:

|      |      |     |
|------|------|-----|
| info | cont | sig |
|------|------|-----|

En el atributo info, se almacena el elemento. El atributo cont se incrementa en uno cada vez que se inserta una nueva ocurrencia de elemento y de la misma manera se decrementa cada vez que se realiza una eliminación del elemento. Cuando, cont queda en 0 se elimina el nodo de la lista.