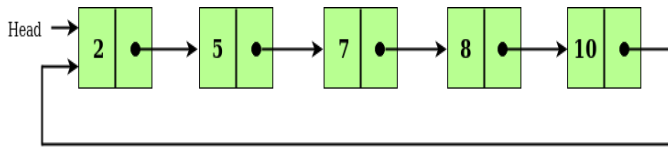


Eliminación de nodos en una Lista Ligada Circular

Considere la lista ligada que se muestra abajo:



Cada nodo queda definido por:

```
class Nodo {
    int data;
    Nodo next;
};
```

Se dará un nodo y nuestra tarea será eliminarlo de la lista.

Ejemplos:

Entrada : 2->5->7->8->10->(head node)

5

Salida : 2->7->8->10->(head node)

Entrada : 2->5->7->8->10->(head node)

7

Salida : 2->5->8->10->2(head node)

Algoritmo

Caso 1: Si la lista está vacía.

- Simplemente return.

Caso 2: Si la lista no está vacía

- Definir dos apuntes: curr inicializado a head y prev como un nodo nuevo.
- Recorrer la lista usando curr para encontrar el nodo a eliminar y antes de mover curr al siguiente nodo, poner prev = curr.
- Si el nodo es encontrado, checar si es el único nodo en la lista. Si es así, poner head = NULL y return head.
- Si la lista tiene mas de un nodo, checar si es el primer nodo. La condición es checar (curr == head). Si es cierto, entonces mover prev hasta llegar al último nodo; entonces, poner head = curr -> next y prev -> next = head.
- Si curr no es el primer nodo, checamos si es el último. La condición a checar es (curr -> next == head).
- Si curr es el último nodo. Poner prev -> next = head.
- Si el nodo a ser eliminado no es ni el primero ni el último, entonces poner prev -> next = curr -> next.
- Retornar head.