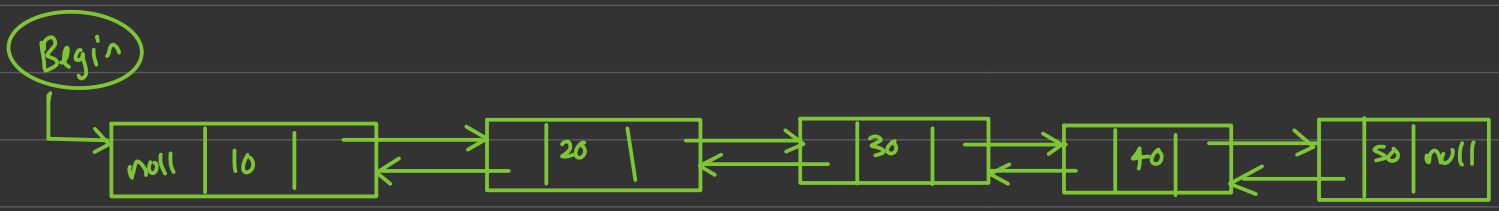
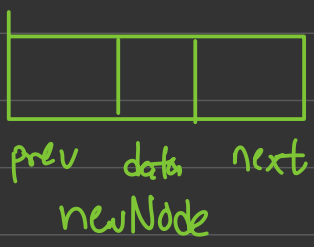
17 June 2023

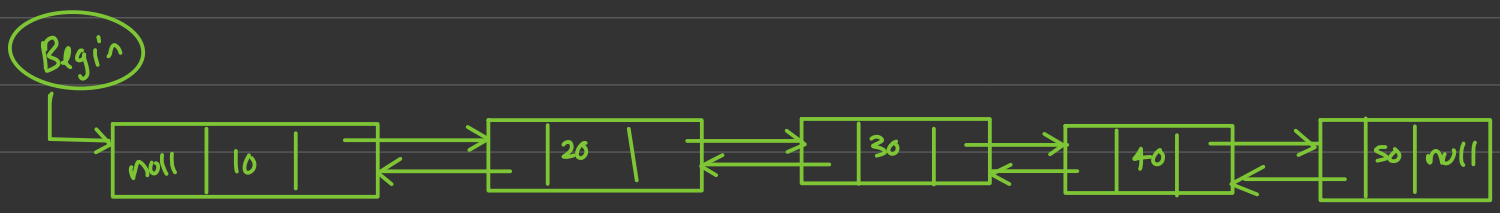
https://my.newtonschool.co/playground/code/eqkl0wizy38p



1. newNode.data = 5



2. If Begin = NULL // if LL is empty
   newNode.next = null
   newNode.prev = null
   Begin = newNode



3. If newNode.info < Begin.info
   newNode.next = Begin
   newNode.prev = null
   Begin.prev = newNode
   Begin = newNode

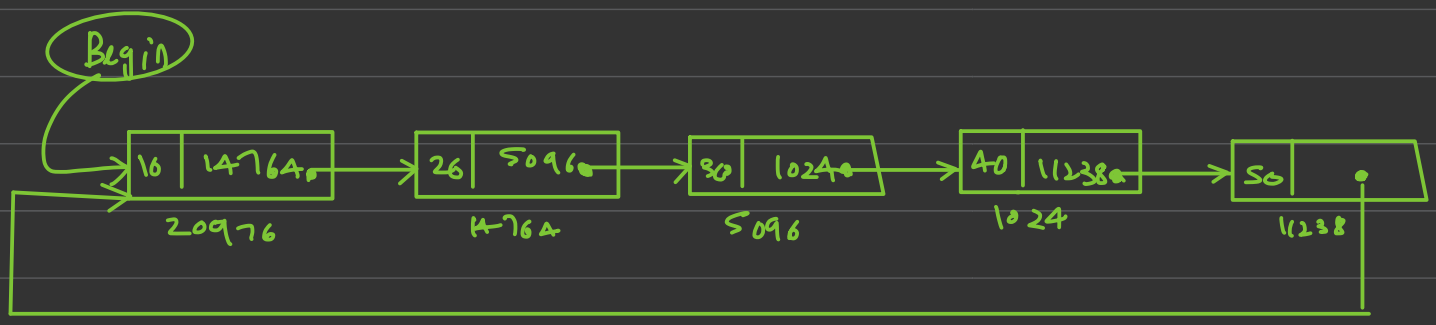
4. Set p = Begin
   if ((newNode.info >= p.info) && (newNode.info < p.next.info) &&
   (p.next != null))
   newNode.next = p.next

```
newNode.prev = p.prev
p.next = newNode
newNode.next.prev = newNode
```

else if `p.next = null`

```
newNode.prev = p
newNode.next = null
p.next = newNode
```

Circular Linked List



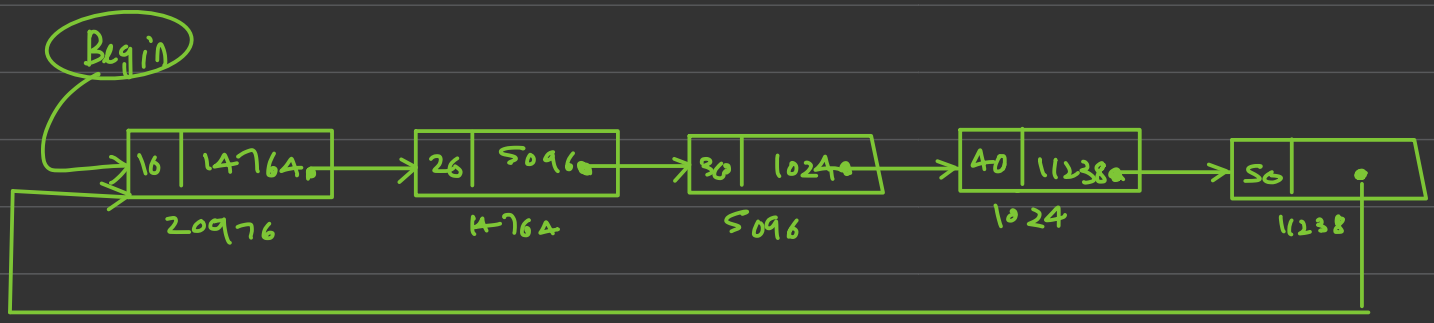
Traversal:

1. If Begin = null
   print "CLL is empty"
2. Print Begin.info
3. Set p = Begin.next

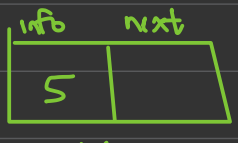
   while p ≠ Begin
   print p info
   p = p.next

18 June 2023

Insertion in CLL: Begining (at start)



1. Create new
   new.info = data

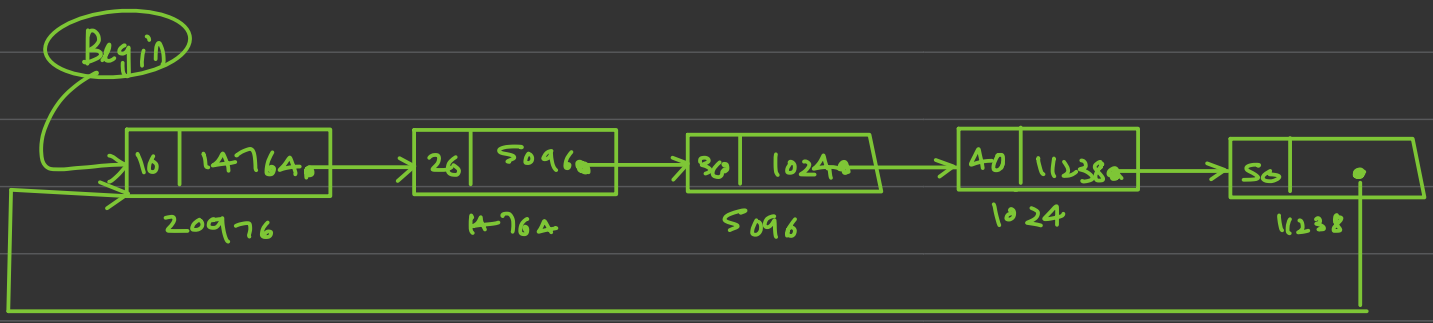


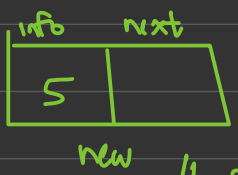
2. If Begin = null // If CLL is empty
   new.next = new
   Begin = new

3. Set p = Begin
   while p.next != Begin
   p = p.next

4. new.next = Begin
   p.next = new
   Begin = new

Insertion in CLL: Ending





1. Create new
   new.info = data
2. If Begin = null // If CLL is empty
   new.next = new
   Begin = new
3. Set p = Begin

   while p.next ≠ Begin
   p = p.next

4. new.next = Begin
   p.next = new

X