```python
class Node:
    def __init__(self,data):
        self.data=data
        self.ref=None
node1=Node(10)

class linkedlist:
    def __init__(self):
        self.head=None
    def print_ll(self):
        if self.head is None:
            print("LL is empty")
        else:
            n=self.head
            while n is not None:
                print(n.data)
                n=n.ref

    #insertion at beg
    def add_beg(self,data):
        newnode=Node(data)
        newnode.ref=self.head
        self.head=newnode

    #insertion at end
    def add_end(self,data):
        newnode=Node(data)
        if self.head is None:
            self.head=newnode
        else:
            n=self.head
            while n.ref is not None:
                n=n.ref
            n.ref=newnode

    #in between
    def add_after(self,data,x):
        n=self.head
        while n is not None:
            if x==n.data:
                break
            n=n.ref
        if n is None:
            print("Node is not present")
        else:
            newnode=Node(data)
            newnode.ref=n.ref
            n.ref=newnode

    #deletion at beginning
    def delete_beg(self):
        if self.head is None:
            print("LL is empty so we cant delete nodes")
        else:
            self.head=self.head.ref

    #deletion at end
    def del_end(self):
        if self.head is None:
            print("LL is empty so we cant delete nodes")
        else:
            n=self.head
            while n.ref.ref is not None:
                n=n.ref
            n.ref=None

ll1=linkedlist()
ll1.add_end(1000)
ll1.add_beg(10)
ll1.add_beg(20)
ll1.add_after(1234,20)
ll1.delete_beg()
ll1.del_end()
ll1.print_ll()

    1234
    10
```