Doubly Linked List

1] Doubly Linked List In Python :

```
class Node:
    def __init__(self,data):
        self.data=data
        self.prev=None
        self.next=None

def prindll(head):
    curr=head
    while curr!=None:
        print(curr.data,end=" ")
        curr=curr.next
    print()


head=Node(10)
temp1=Node(20)
temp2=Node(30)

head.next=temp1
temp1.prev=head

temp1.next=temp2
temp2.prev=temp1

prindll(head)
```

OUTPUT :

10 20 30

2] Insert at the beginning of doubly linked list :

```
class Node:
    def __init__(self,data):
        self.data=data
        self.prev=None
        self.next=None


def insertBegDDL(head,x):
    temp=Node(x)
    if head!=None:
        head.prev=temp
    temp.next=head
    return temp

def printDLL(head):
    curr=head
    while curr!=None:
        print(curr.data,end=" ")
        curr=curr.next
    print()

head=Node(10)
temp1=Node(20)
temp2=Node(30)

head.next=temp1
temp1.prev=head

temp1.next=temp2
temp2.prev=temp1

printDLL(head)

head=insertBegDDL(head,40)

printDLL(head)
```

OUTPUT :

10 20 30

40 10 20 30

3] Inset at the end of DDL :

```
class Node:
    def __init__(self, data):
        self.data=data
        self.prev=None
        self.next=None


def insertEnd(head,x):
    temp=Node(x)

    if head==Node:
        return temp

    curr=head
    while curr.next!=None:
        curr=curr.next

    curr.next=temp
    temp.prev=curr

    return head

def printDll(head):
    curr=head
    while curr!=None:
        print(curr.data,end=" ")
        curr=curr.next
    print()


head=Node(10)
temp1=Node(20)
temp2=Node(30)

head.next=temp1
temp1.prev=head

temp1.next=temp2
temp2.prev=temp1

printDll(head)

head=insertEnd(head,40)
```

```
printDll(head)
```

OUTPUT :

10 20 30

10 20 30 40

4] delete head of doubly linked list :

```
class Node:
    def __init__(self,data):
        self.data=data
        self.prev=None
        self.next=None

def deleteHead(head):
    if head==None or head.next==None:
        return None

    else:
        head=head.next
        head.prev=None

        return head

def printDll(head):
    curr=head
    while curr!=None:
        print(curr.data,end=" ")
        curr=curr.next
    print()

head=Node(10)
temp1=Node(20)
temp2=Node(30)

head.next=temp1
temp1.prev=head

temp1.next=temp2
temp2.prev=temp1

printDll(head)

head=deleteHead(head)

printDll(head)
```

OUTPUT :

10 20 30

20 30

5] Delete last Node of DLL in python :

```
class Node:
    def __init__(self,data):
        self.data=data
        self.prev=None
        self.next=None

def deleteLast(head):
    if head==None or head.next==None:
        return None

    curr = head
    while curr.next.next != None:
        curr = curr.next

    curr.next = None
    return head

def printDll(head):
    curr=head
    while curr!=None:
        print(curr.data,end=" ")
        curr=curr.next

    print()

head=Node(10)
temp1=Node(20)
temp2=Node(30)

head.next=temp1
temp1.prev=head

temp1.next=temp2
temp2.prev=temp1

printDll(head)

head=deleteLast(head)

printDll(head)
```

OUTPUT : 10 20 30

10 20

6] Reverse a doubly linked list in python :

```
class Node:
    def __init__(self,data):
        self.data=data
        self.prev=None
        self.next=None

def reverseDll(head):
    if head==None or head.next==None:
        return head

    curr=head
    while curr!=None:
        prev=curr
        curr.next,curr.prev=curr.prev,curr.next
        curr=curr.prev

    return prev

def printDll(head):
    curr=head
    while curr!=None:
        print(curr.data,end=" ")
        curr=curr.next
    print()

head=Node(10)
temp1=Node(20)
temp2=Node(30)

head.next=temp1
temp1.prev=head

temp1.next=temp2
temp2.prev=temp1

printDll(head)

head=reverseDll(head)

printDll(head)
```

OUTPUT : 10 20 30

30 20 10