MOVING HYPHENS TO FRONT

```
def mov_to_frt(s):
    if s is None:
        return None
    hyp=""
    oth=""
    for char in s:
        if char =="-":
            hyp+=char
        else:
            oth+=char
    return hyp+oth

a="hello-hi-sir"
b=mov_to_frt(a)
print(b)
```

```
--hellohisir
```

LINKED LIST (DEL BY VAL AND REVERSE)

```
class MyLL:
    class Node:
        def __init__(self,data):
            self.data=data
            self.next=None
    def __init__(self):
        self.head=None
    def insb(self,data):
        nd=MyLL.Node(data)
        nd.next=self.head
        self.head=nd
    def insed(self,data):
        nd1=MyLL.Node(data)
        if self.head==None:
            self.head==nd1
            return
        current=self.head
        while current.next:
            current=current.next
        current.next=nd1
    def insaft(self,data,prev):
        nd3=MyLL.Node(data)
        nd3.next=pre.next
        prev.next=nd3
    def prt(self):
        current=self.head
        while current:
```

```
            print(current.data,end='-->')
            current=current.next
    def srch(self,vl):
        current=self.head
        while current:
            if current.data==val:
                return True
            current=current.next
        return False
    def delfrst(self):
        self.head=self.head.next
    def deled(self):
        if self.head==0:
            return
        current=self.head
        while current.next.next:
            current=current.next
        current.next=None
    def byval(self,x):
        if self.head==x:
            self.head==self.head.next
        current=self.head
        while current.next:
            if current.next.data!=x:
                current=current.next
            else:
                break
        if current.next:
            current.next=current.next.next
        else:
            return 'not found'
    def rev(self):
        prv=None
        current=self.head
        while current:
            n=current.next
            current.next=prv
            prv=current
            current=n
        self.head=prv

l=MyLL()

l.insb(40)
l.insb(45)
l.insb(46)
l.insb(47)
l.insb(48)
l.prt()
```

```
48-->47-->46-->45-->40-->

l.byval(46)

l.prt()

48-->47-->45-->40-->

l.rev()
l.prt()

40-->45-->47-->48-->
```