In [*]:
```
class Node:
    def __init__(self,data):
        print("Node created",data)
        self.data=data
        self.next=None
class S_L_List:
    def __init__(self):
        self.head=None
        self.ctr=0
    def insert_beginning(self,data):
        node=Node(data)
        if self.head==None:
            self.head=node
            self.ctr+=1
        else:
            node.next=self.head
            self.head=node
            self.ctr+=1
        print("Node inserted",data)
        return
    def insert_middle(self,pos,data):
        if pos==0:
            self.insert_beginning(data)
        elif pos==self.ctr+1:
            self.insert_end(data)
        else:
            node=Node(data)
            temp=self.head
            i=0
            while(i<pos-1):
                temp=temp.next
                i+=1
            node.next=temp.next
            temp.next=node
            self.ctr+=1
            print("node inserted",data)
        return
    def insert_end(self,data):
        node=Node(data)
        node.next=None
        if self.head==None:
            self.head=node
```

```
            self.ctr+=1
            return
        temp=self.head
        while(temp.next is not None):
            temp=temp.next
        temp.next=node
        self.ctr+=1
        print("Node inserted",data)
        return
    def delete_beginning(self):
        if self.head==None:
            print("no nodes exist")
        elif self.ctr==1:
            print("Node deleted",self.head.data)
            self.head=None
            self.ctr-=1
        else:
            print("Node deleted",self.head.data)
            self.head=self.head.next
            self.ctr-=1
        return
    def delete_middle(self,pos):
        if self.head==None:
            print("No nodes exist")
        elif pos==0:
            self.delete_beginning()
        elif pos==self.ctr:
            self.delete_end()
        else:
            temp=self.head
            prev=temp
            i=0
            while(i<pos):
                prev=temp
                temp=temp.next
                i+=1
            prev.next=temp.next
            print("node delted",temp.data)
            temp.next=None
            self.ctr-=1
        return
    def delete_end(self):
        if self.ctr==0:
```

```
            print("No nodes present")
        elif self.ctr==1:
            self.ctr=0
            print("Node deleted",self.head.data)
            self.head=None
        else:
            temp=self.head
            prev=self.head
            while(temp.next is not None):
                prev=temp
                temp=temp.next
            print("Node deleted",temp.data)
            prev.next=temp.next
            self.ctr-=1
        return
    def traverse_forward(self):
        if self.head==None:
            print("No nodes exist")
        print("traversal forward")
        temp=self.head
        while(temp is not None):
            print(temp.data)
            temp=temp.next
    def menu(self):
        print("1.insert at beginning")
        print("2.insert at middle")
        print("3.insert at end")
        print("4.delete at beginning")
        print("5.delete at middle")
        print("6.delete at end")
        print("7.traversal forward")
        print("8.count number of nodes")
        print("9.Exit")
        ch=eval(input("Enter choice:"))
        return ch
print("********* SINGLE LINKED LIST *********")
l=S_L_List()
while True:
    ch=l.menu()
    if ch==1:
        data=eval(input("enter data:"))
        l.insert_beginning(data)
    elif ch==2:
```

```
        data=eval(input("enter data:"))
        pos=eval(input("enter the position"))
        l.insert_middle(pos,data)
    elif ch==3:
        data=eval(input("enter data:"))
        l.insert_end(data)
    elif ch==4:
        l.delete_beginning()
    elif ch==5:
        pos=eval(input("enter position"))
        l.delete_middle(pos)
    elif ch==6:
        l.delete_end()
    elif ch==7:
        l.traverse_forward()
    elif ch==8:
        print("Number of nodes",l.ctr)
    elif ch==9:
        print("exit")
        break
    else:
        print("Invalid Choice")
```

```
********* SINGLE LINKED LIST *********
1.insert at beginning
2.insert at middle
3.insert at end
4.delete at beginning
5.delete at middle
6.delete at end
7.traversal forward
8.count number of nodes
9.Exit

Enter choice:
```