

Link list

Part 1 : Class

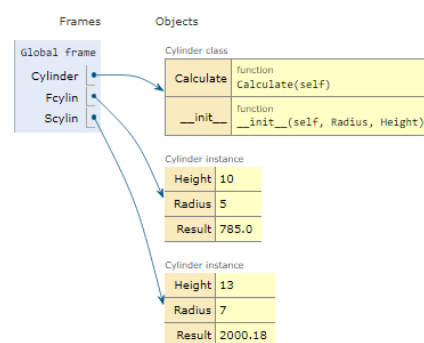
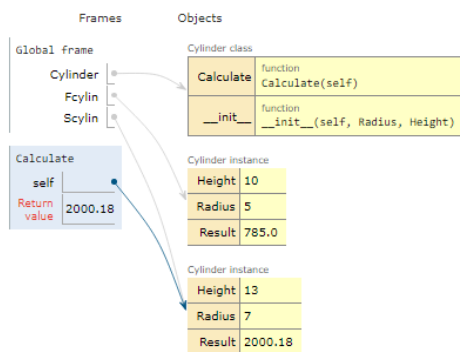
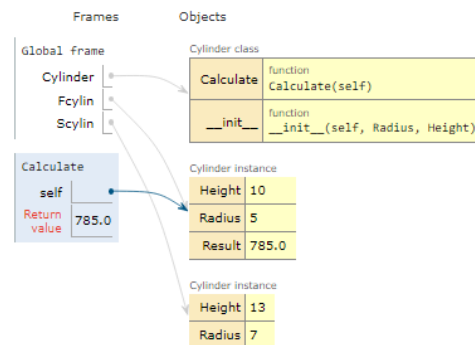
Exercise 1 : Create two cylinders by using class

Code:

```
1 class Cylinder():
2     def __init__(self, Radius, Height):
3         self.Radius = Radius
4         self.Height = Height
5     def Calculate(self):
6         self.Result = 3.14*(self.Radius*self.Radius)*self.Height
7         return self.Result
8
9 Fcylin = Cylinder(5,10)
10 Scylin = Cylinder(7,13)
11
12 print("-----First Cylinder-----")
13 print("Radius : ", Fcylin.Radius)
14 print("Height : ", Fcylin.Height)
15 print("Result : ", Fcylin.Calculate())
16 print("-----Second Cylinder-----")
17 print("Radius : ", Scylin.Radius)
18 print("Height : ", Scylin.Height)
19 print("Result : ", Scylin.Calculate())
20 print("-----")
21
```

Result :

```
-----First Cylinder-----
Radius : 5
Height : 10
Result : 785.0
-----Second Cylinder-----
Radius : 7
Height : 13
Result : 2000.1800000000003
-----
```



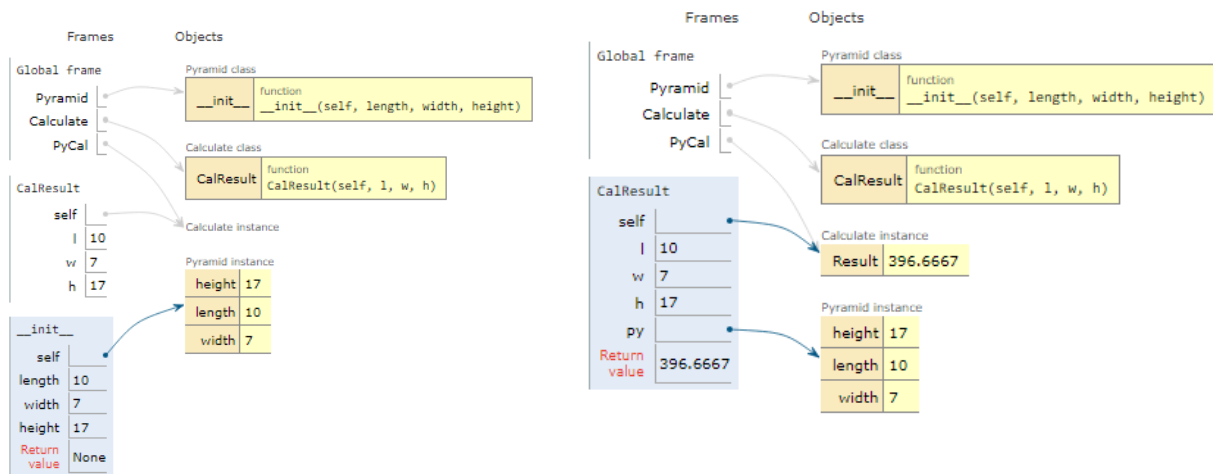
Exercise 2 : Create a pyramid by using class with linked parametre method

Code :

```
1 class Pyramid():
2     def __init__(self,length,width,height):
3         self.length = length
4         self.width = width
5         self.height = height
6
7 class Calculate():
8     def CalResult(self,l,w,h):
9         py = Pyramid(l,w,h)
10        self.Result = (py.length*py.width*py.height)/3
11        return self.Result
12
13 PyCal = Calculate()
14 print("volume of pyramid = ",PyCal.CalResult(10,7,17))
15
```

Result :

volume of pyramid = 396.6666666666667



Part 2 : Linked list

Exercise 3 :

Code :

```
1 class node:
2     def __init__(self, val):
3         self.prv = None
4         self.val = val
5         self.next = None
6
7 class linkedList:
8     def __init__(self):
9         self.head = None
10
11     def push(self, val):
12         newNode = node(val)
13         if self.head == None:
14             self.head = newNode
15         else:
16             poin = self.head
17             while poin.next is not None:
18                 poin = poin.next
19             poin.next = newNode
20             newNode.prv = poin
21
22     def pushStart(self, val):
23         newNode = node(val)
24         poin = self.head
25
26         if self.head == None:
27             self.head = newNode
28         else:
29             newNode.next = poin
30             self.head = newNode
31             poin.prv = newNode
32
33     def delete(self, val):
34         poin = self.head
35
36         while poin.val != val:
37             if poin.next == None:
38                 break
39             else:
40                 poin = poin.next
41         if poin.next == None and poin.val != val:
42             print("This option is not available")
43         else:
44             print("Delete number '", poin.val, "' complete")
45
46             prevNode = poin.prv
47             prevNode.next = poin.next
48
49             poin.next = None
50             poin.prv = None
51
52
53     def display(self):
54         displayVal = self.head
55         print("header -> ", end="")
56         while displayVal is not None:
57             print(displayVal.val, end="")
58             displayVal = displayVal.next
59             if displayVal is not None:
60                 print(" -> ", end="")
61             else:
62                 print("")
```

```

64 ▶ if __name__ == "__main__":
65     List = linkedList()
66
67     List.push(44)
68     List.push(36)
69     List.push(90)
70     List.push(10)
71     List.push(60)
72     List.push(99)
73
74     while True:
75         print("\n-----",
76             "\n1 for Insert",
77             "\n2 for Insert from header",
78             "\n3 for Delete",
79             "\n4 for Display",
80             "\n5 for exit",
81             "\n-----")
82         num = int(input("select : "))
83         if num == 1:
84             inPush = input("Enter the number you want to Insert : ")
85             List.push(inPush)
86         elif num == 2:
87             inPushHead = input("Enter the number you want to Insert : ")
88             List.pushStart(inPushHead)
89         elif num == 3:
90             inDelete = input("Enter the number you want to Delete : ")
91             List.delete(inDelete)
92         elif num == 4:
93             List.display()
94         elif num == 5:
95             break
96         else:
97             print("This option is not available")
98

```

Result :

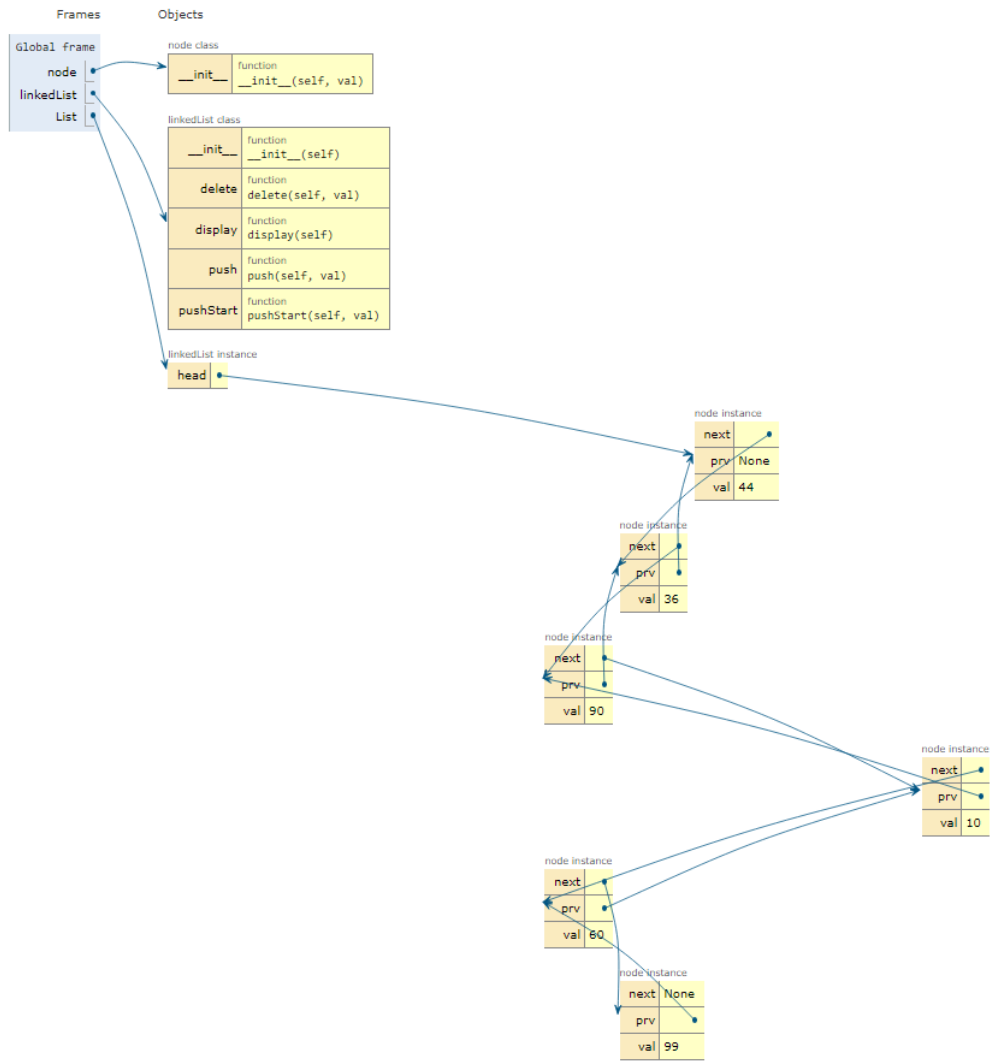
Ex 3.1 : Create linked list as below

```

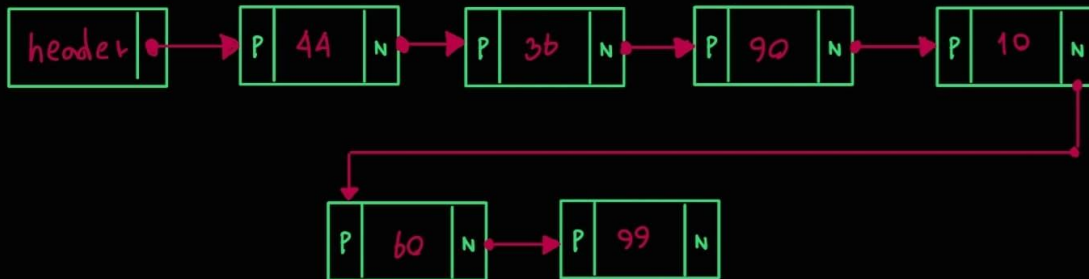
-----
1 for Insert
2 for Insert from header
3 for Delete
4 for Display
5 for exit
-----

select : 4
header -> 44 -> 36 -> 90 -> 10 -> 60 -> 99

```

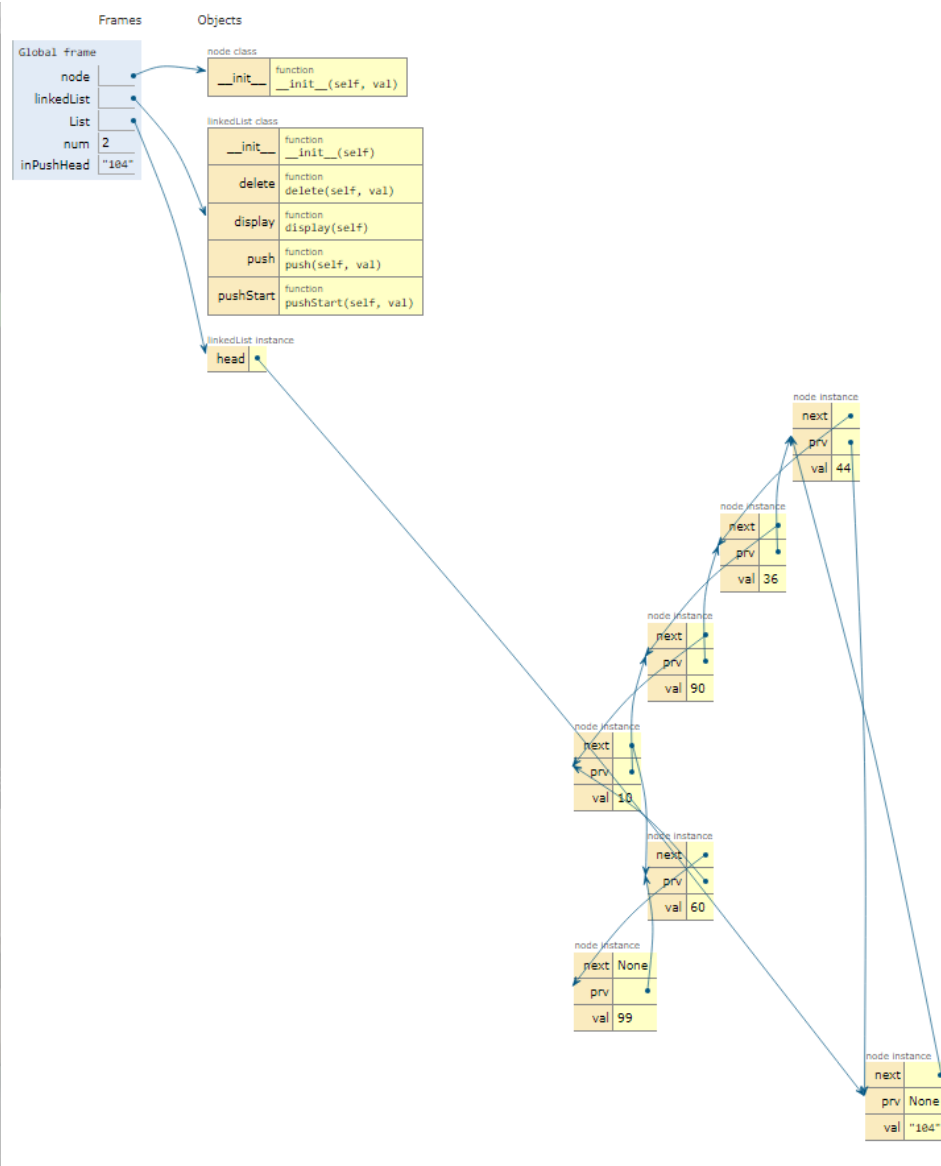


Create linked list



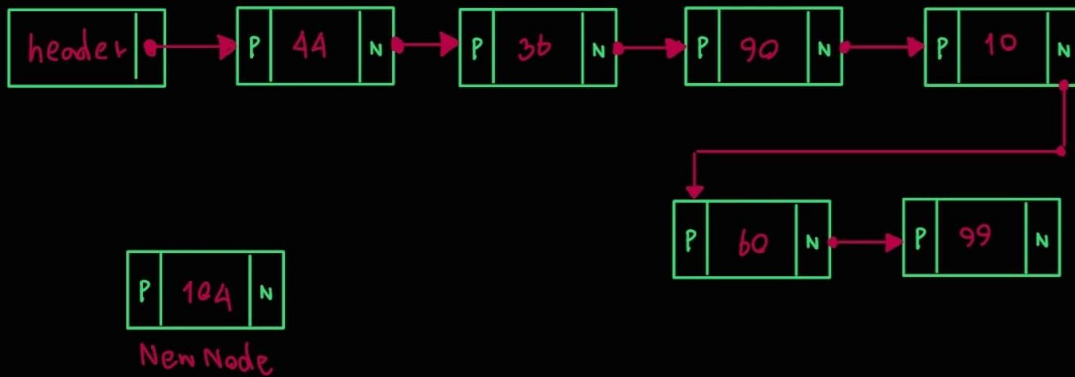
Ex 3.2 : Insert 104 from header

```
-----  
1 for Insert  
2 for Insert from header  
3 for Delete  
4 for Display  
5 for exit  
-----  
select : 2  
Enter the number you want to Insert : 104  
  
select : 4  
header -> 104 -> 44 -> 36 -> 90 -> 10 -> 60 -> 99
```



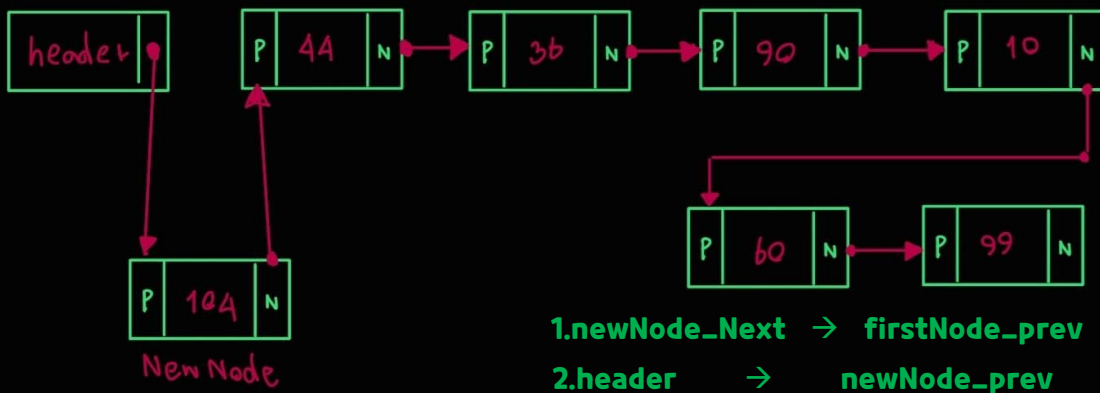
1

Insert 104 from header



2

Insert 104 from header

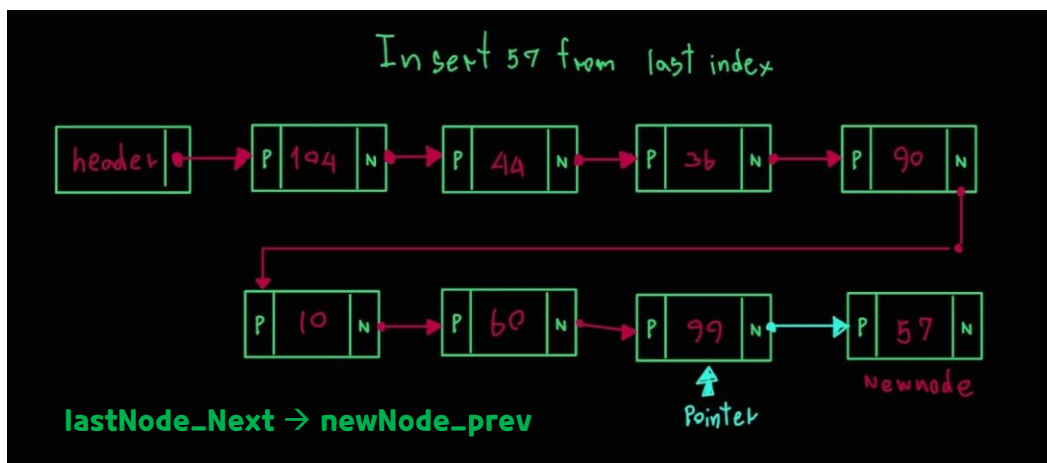
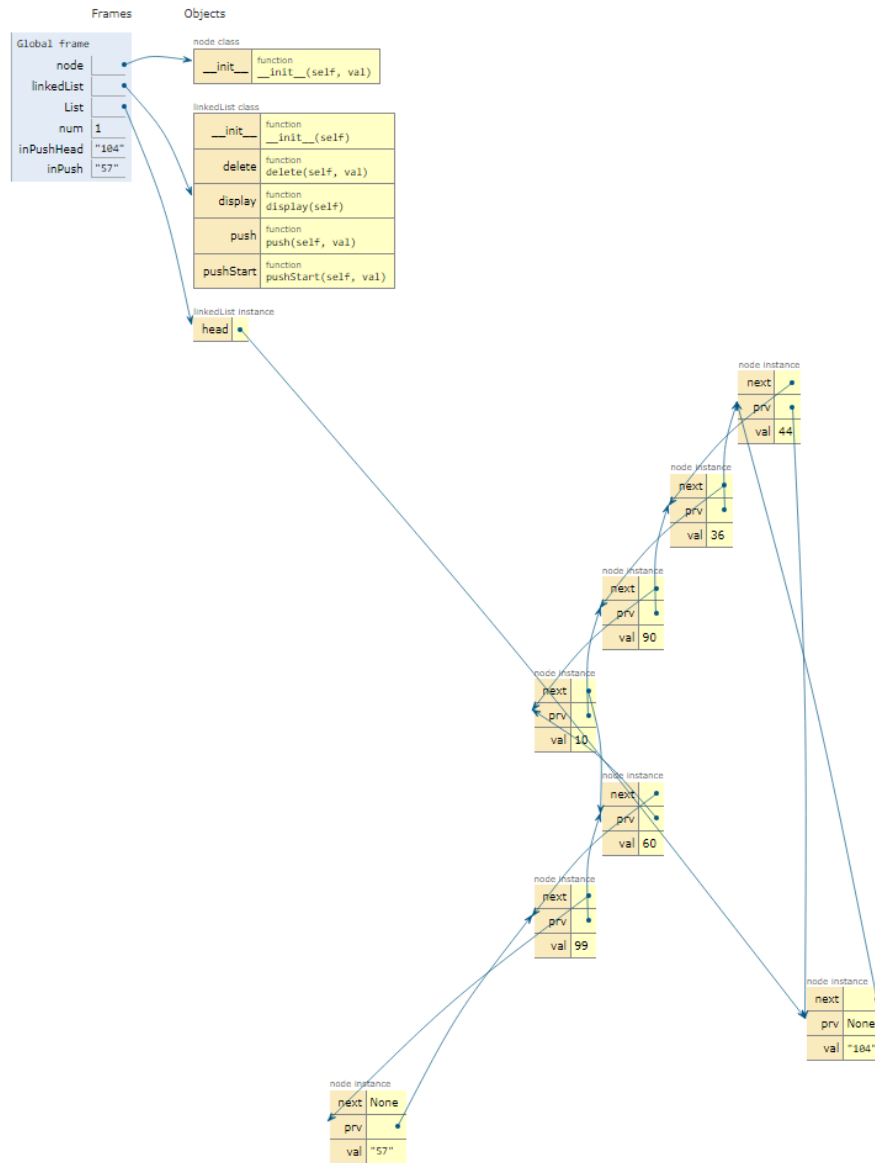


1.newNode_Next → firstNode_prev
2.header → newNode_prev

Ex 3.3 : Insert 57 from last index (append)

```
-----
1 for Insert
2 for Insert from header
3 for Delete
4 for Display
5 for exit
-----
select : 1
Enter the number you want to Insert : 57

select : 4
header -> 104 -> 44 -> 36 -> 90 -> 10 -> 60 -> 99 -> 57
```

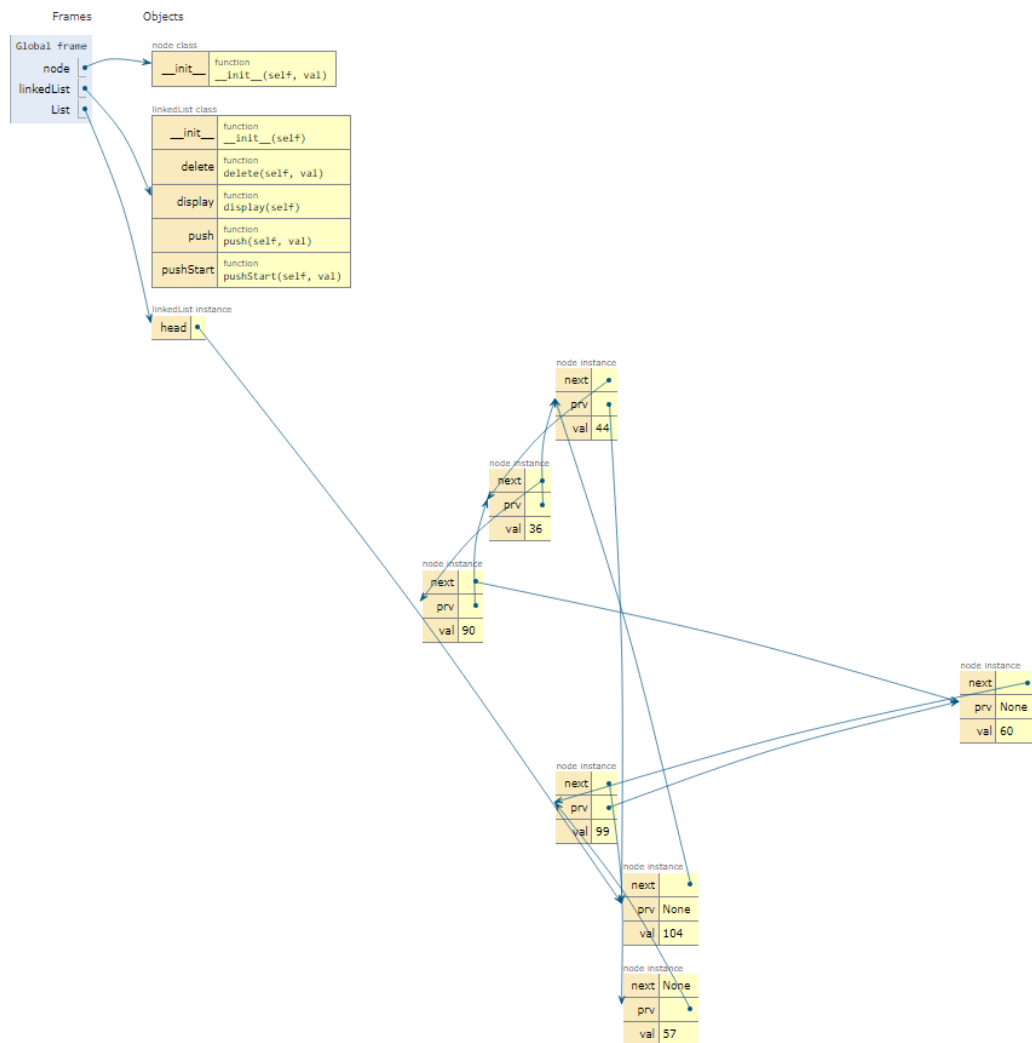


Ex 3.4 : Remove value where index = 4 (10)

```
-----  
1 for Insert  
2 for Insert from header  
3 for Delete  
4 for Display  
5 for exit  
-----
```

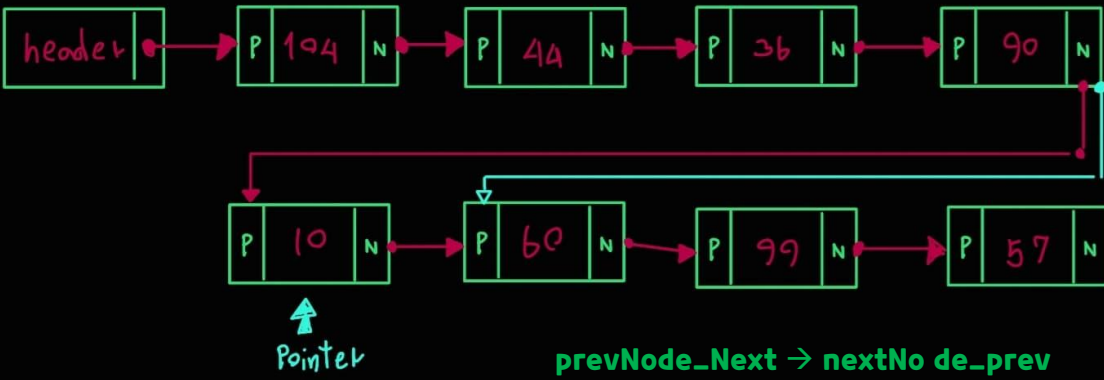
```
select : 3  
Enter the number you want to Delete : 10  
Delete number ' 10 ' complete
```

```
select : 4  
header -> 104 -> 44 -> 36 -> 90 -> 60 -> 99 -> 57
```

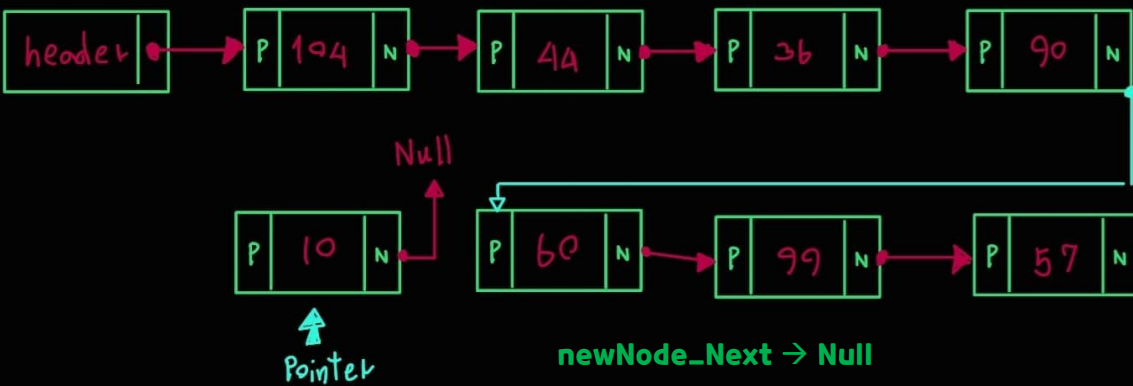


1

Remove "10" from linked list



Remove "10" from linked list



3

Remove "10" from linked list

