

8. Single linked list

A) Insertion:

(i) At Beginning:

Algorithm:

Step 1 → If head = Null go to step 7

Step 2 → Set new node = head

Step 3 → Set head = head → next

Step 4 → Set new node → data = d

Step 5 → new node → next = head

Step 6 → head = new node

Step 7 → exit

Process:

→ For inserting at beginning we are first going to break a node with data 'd' & its next as none.

→ Before process starts the value of head will be none

→ Now we are going to create linked list for elements 6, 5, 7, 3, 1.

→ we are first going to pass 6 to insert begin, A new node will be created

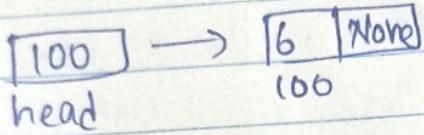
[6 | None]

100

Since this is first node, head will be None



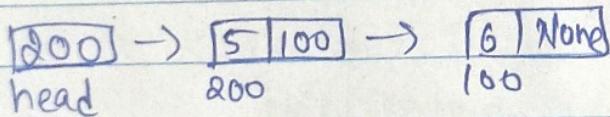
Now we are assigning new node.next to self.head change to new node for which it store the address of new node.



Now value 5 will be passed for which 5 None
will be created.

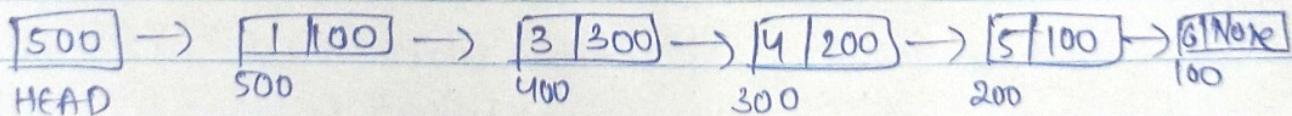
Again newnode.next = self.head \Rightarrow 5 100
200

self.head = New node \Rightarrow 200



Similarly for remaining elements 4,3,1 by following above process

The final output list is



The Linked list is created.

Output-

Enter N: 5

Enter number: 6

Enter number: 5

Enter number: 4

Enter number: 3

Enter number: 1

1 → 3 → 4 → 5 → 6

Code-

Class Node:

```
def __init__(self,d):  
    self.data=d  
    self.next=None
```

Class LL:

```
def __init__(self):  
    self.head=None  
def InsertBegin(self,d):  
    New node=Node(d)  
    New node.next=self.head  
    self.head=New node  
def print(self):  
    temp=self.head  
    while temp is not None:  
        print(temp.data,end="→")  
        temp=temp.next  
n=int(input("Enter N:"))  
L1=LL()  
for i in range(0,n):  
    d=int(input("Enter Number:"))  
    L1.InsertBegin(d)  
L1.print()
```

At End:

Algorithm -

```
def Insert End(self, d):
    New node ← Node(d)
    if self.head is None:
        self.head ← New node
    else:
        temp ← self.head
        while temp.next ≠ None:
            temp ← temp.next
        temp.next ← New node
```

Process -

In this process we are going to create a linked list by inserting new node at end of list

Initialize head as none

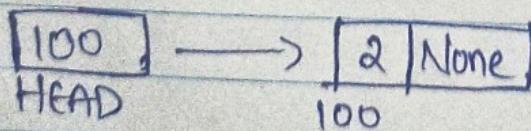
None
head

Now create list with elements: 2, 3, 6, 9

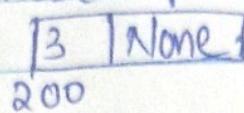
Create a new node with data '2' by keeping next as none:

2	None
100	

Now self.head ← New node where in head it gets address of new node '2' i.e '100'



- In this similar way we will insert all elements.
- Create new node of '3'



- By traversing through list with some temporary variable find last element

- Now temp.next ← New node where the new node's address will be inserted to last node's next

Hence:

100	→	2 100	→	3 None
head	100	200		300

- By following above steps for remaining elements the total output is:

100	→	2 200	→	3 300	→	6 400	→	9 None
head	100	200		300		400		

Hence linked list is formed.

Code :

```
class Node:  
    def __init__(self,d):  
        self.data=d  
        self.next=None  
  
class linked list:  
    def __init__(self):  
        self.head=None  
  
    def insert_end(self,d):  
        if self.head is None:  
            self.head=New node  
        else:  
            temp=self.head  
            while temp.next is not None:  
                temp=temp.next  
            temp.next=new node  
  
    def print(self):  
        temp=self.head  
        while temp is not None:  
            print(temp.data,end="→")  
            temp=temp.next  
  
n=int(input("enter N:"))  
L1=linked list()  
  
for i in range(0,n)  
d=int(input("enter data:"))  
L1.insert_end(d)  
  
L1.print()
```

Output -

```
Enter N:4  
Enter data:8  
Enter data:3  
Enter data:6  
Enter data:9  
8→3→6→9
```

(iii) At Given position -

Algorithm -

```
def Insert(self, d, n)
    new node ← Node(d)
    i ← 1
    if self.head is none:
        print(No List)
    else if i in range (i+1, n):
        temp ← temp.next
        i ← i+1
```

New node.next ← temp

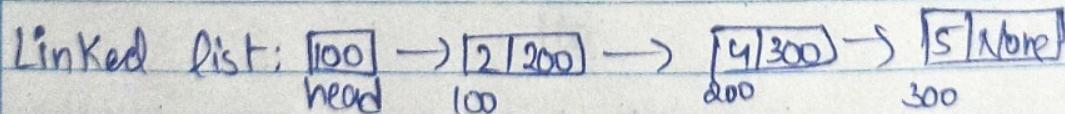
temp.i.next ← new node

If i is l: L.insert Begin(d)

If j is n: L.insert End(d)

Process -

→ In this we are inserting a node in the linked list at given position

Linked list:  head → 100 → 200 → 300 → 400 → 500 → None

Let position be at 2

→ Let the data be 'd'

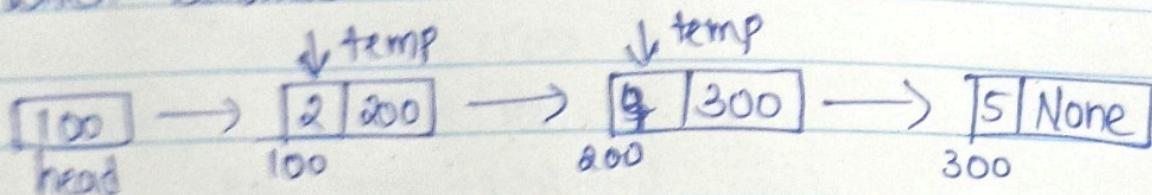
→ Create a new node with given data 'd' & next as 'None'.

d	None
110	

→ 'j' is in the range between 2 & 3 and i is 1 because we are at beginning.

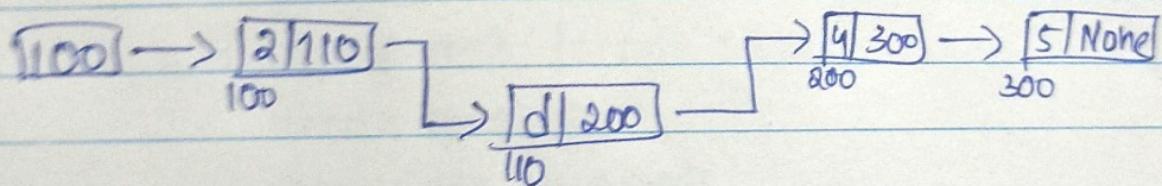
→ while loop starts since $i < j$ which is $1 < 2$
temp1 = temp
temp = temp.next
 $i = i + 1$

which becomes



Now loop breaks as condition unsatisfied

Now we have to insert the node before & after position.



For this first step is

temp.next ← new node
 $[200] \rightarrow [3\text{None}]$
100

Next non node.next ← temp

$[3\text{None}] \rightarrow [4300]$
100 800

Therefore we have inserted the node in the list

Code -

class Node:

```
def __init__(self,d):  
    self.data=d  
    self.next=None
```

class Linked list:

```
def __init__(self):  
    self.head=None  
def insert_end(self,d):  
    New node=Node(d)  
    if self.head is None:  
        self.head=New node
```

else:

temp=self.head

while temp.next is not None:

temp=temp.next

temp.next=New node

def print(self):

temp=self.head

while temp is not None:

print(temp.data,end="→")

temp=temp.next

def insert(self,d,n,i):

New node=Node(d)

i=1

if self.head is None:

print("No linked list")

def i in range(i+1,n):

Output:

Created LL: 2 → 4 → 5 → 6
updated LL: 2 → 3 → 4 → 5 → 6

temp = self.head
while i < j and temp.next is not None:
 temp = temp.next
 temp = temp.next
 i = i + 1

Newnode.next = temp
temp.i.next = New node

i =

L1.insert Begin()

j = i + 1

L1.insert End()

n = int(input("Enter N:"))

L1.Linked List()

for i in range(0, n):

d = int(input("Enter data:"))

L1.Insert Begin(d)

print("Created linked list")

L1.print()

o = int(input("Enter data!"))

N = int(input("Enter position:"))

L1.insert(o, N, n)

print("updated linked list")

L1.print()