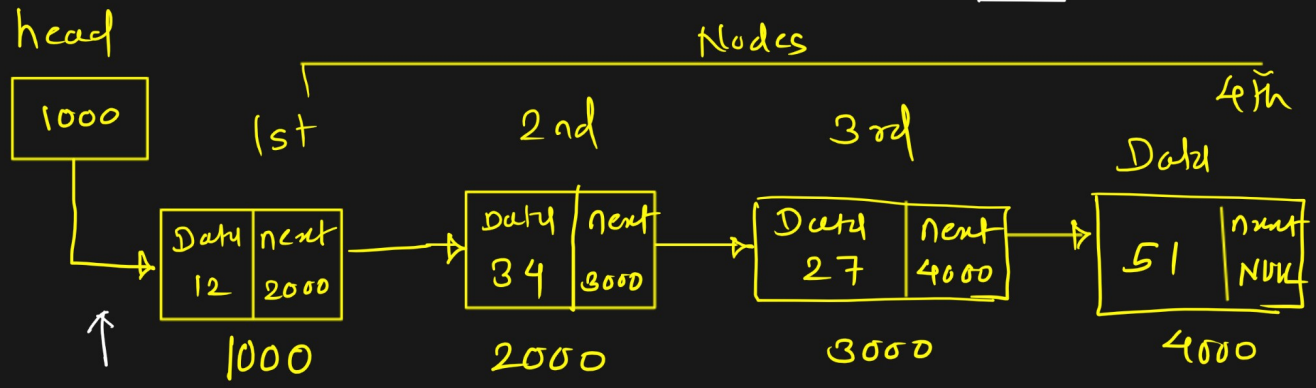
Day - 17 Linked List



Operation :-

1) Insert
2) Delete
3) Display
4) Search

1) Insert

- A) Insert at First
- B) Insert at Last
- C) Insert in between

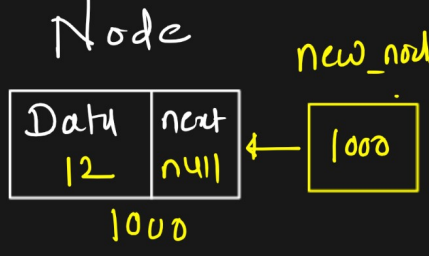
Delete

- Delete 1st node
- Delete last node
- Delete any node

A) Insert :-

1) Insert At First

i) Create first of all node




```
Node new_node = new Node(12);
```

(1000)

```
class Node
{
    int data;
    Node next;
    public Node(int data)
    {
        this.data = data;
        next = null;
    }
}
```

ii) Create a linked list :-

Node head

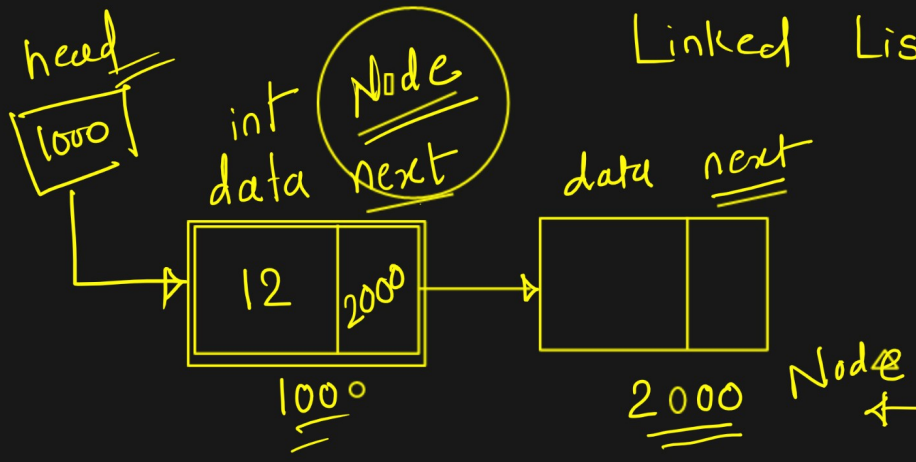
null ⇒ Empty Linked List

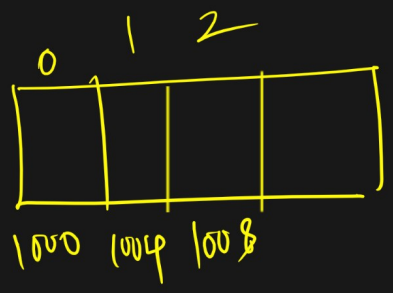
```
Linked_List l = new Linked_List();
```

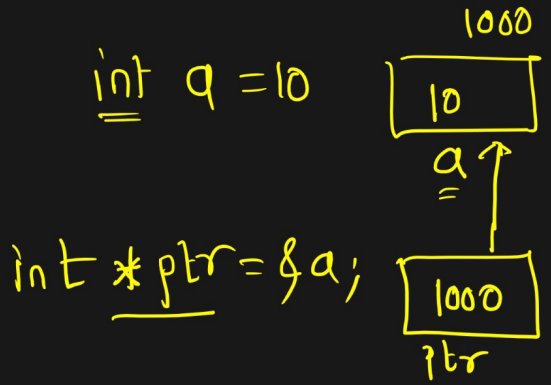
```
class Linked_List
{
    Node head;
    public Linked_List()
    {
        head = null;
    }
}
```

Node

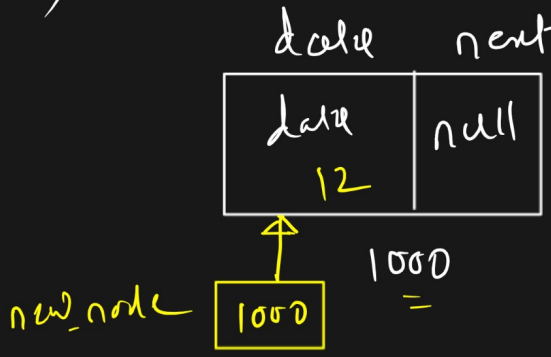
Linked List ⇒ Contigeous Memory not present







i) Create a node



Node new_node = new Node(12);

```
class Node
{
    int data;
    Node next;
    public Node(int data)
    {
        this.data = data;
        next = null;
    }
}
```

ii) Create Empty Linked List

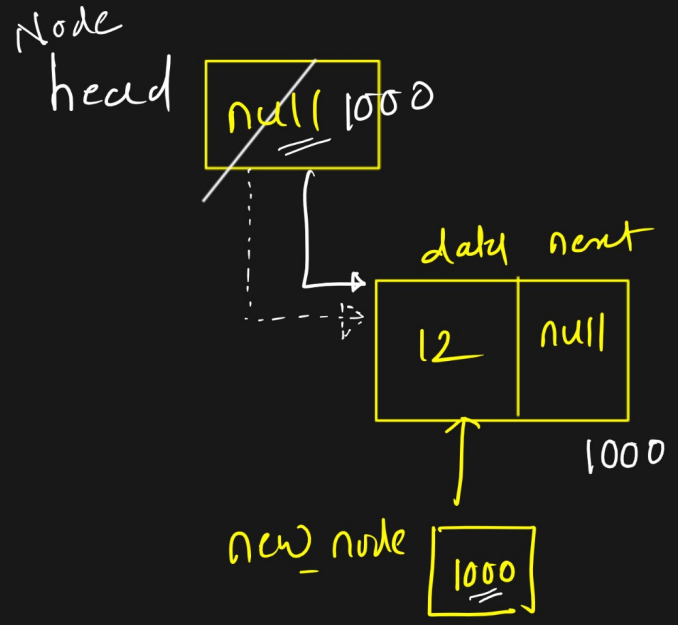
Node
head → null

// Empty Linked List

```
class Linked_List
{
    Node head;
    public Linked_list()
    {
        head = Null;
    }
    :
    continue;
```

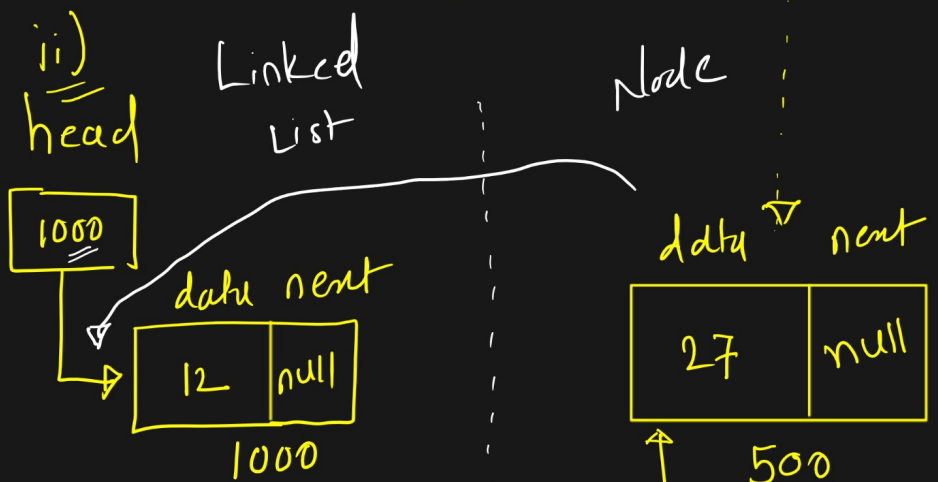
iii) Insert

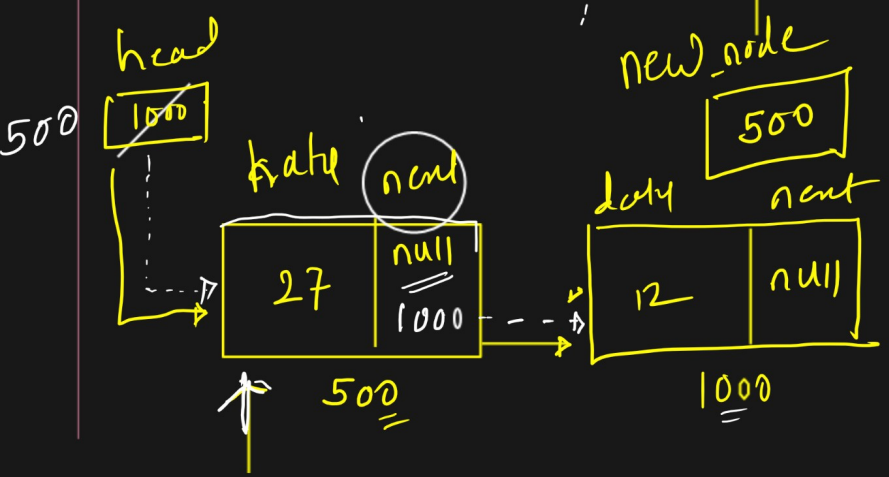
1) Insert At First Pos

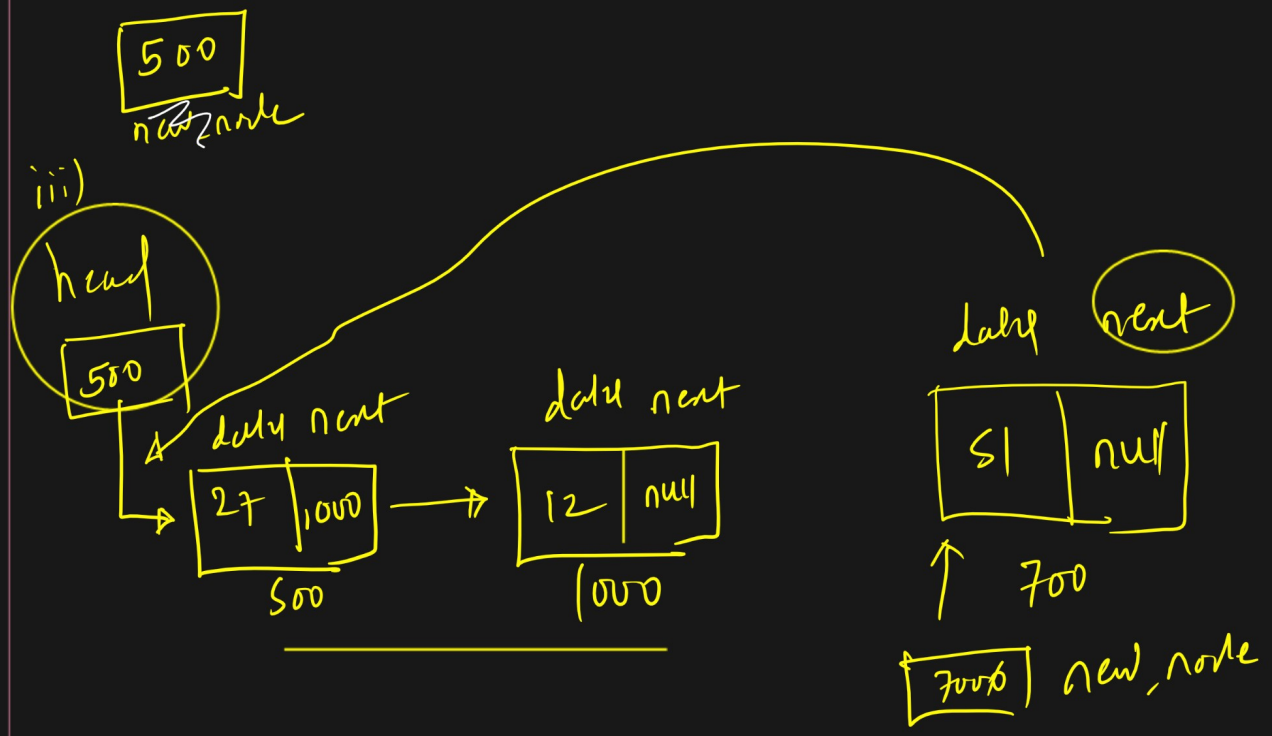



```
public void insert_At_First(int num)
{
    Node new_node = new Node(num);

    if (head == null) // Linked list is empty
    {
        head = new_node;
    }
    else
    {
        new_node.next = head;
                  null  1000
        head = new_node;
                   500
    }
}
```





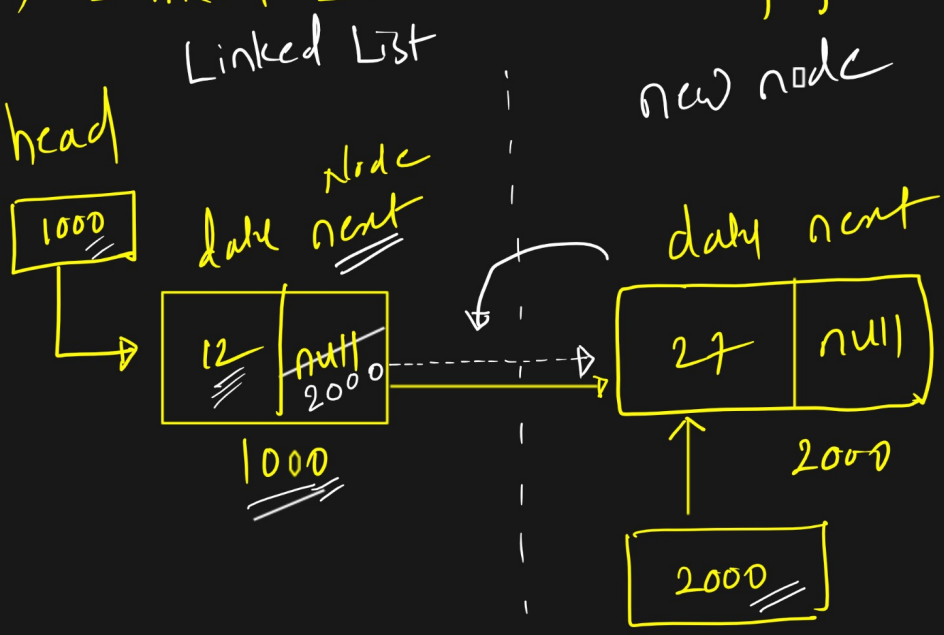


ii) Insert At last

A) Linked List is empty crnot :-


```
public void insert_At_Last(int num)
{
    Node new_node = new Node(num);
    if (head == null)
    {
        head = new_node;
    }
```

ii) Linked List is not empty



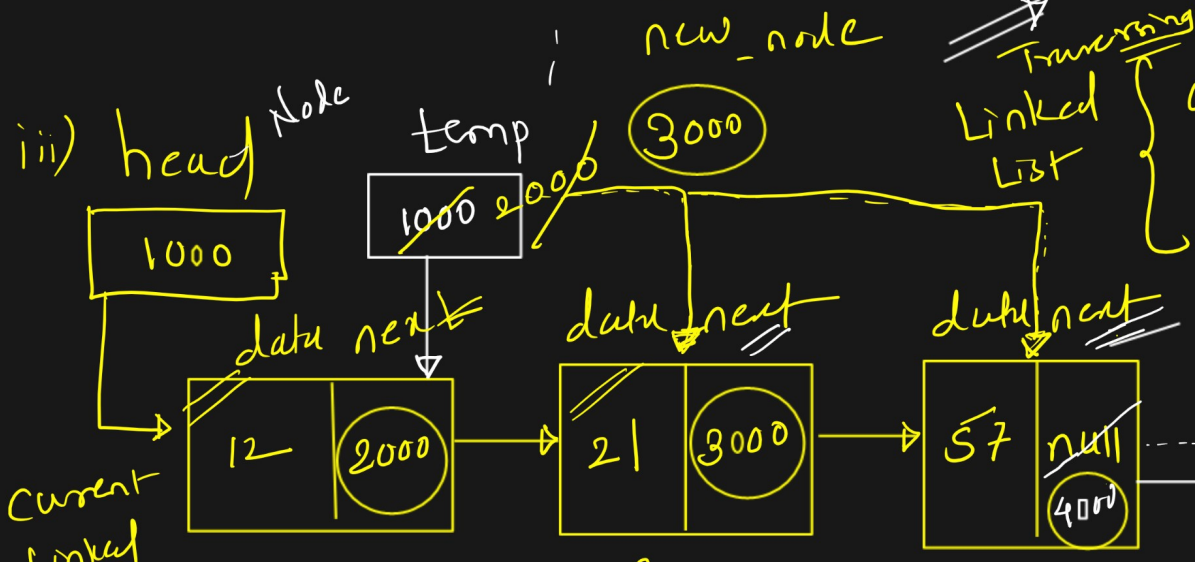

```
else if ( head.next == null)
{
    head.next = new_node;
}
```

1000 ← 2000

```
else
{
    Node temp = head ;
```

1000 ⇒ 1000

iii) cond^n



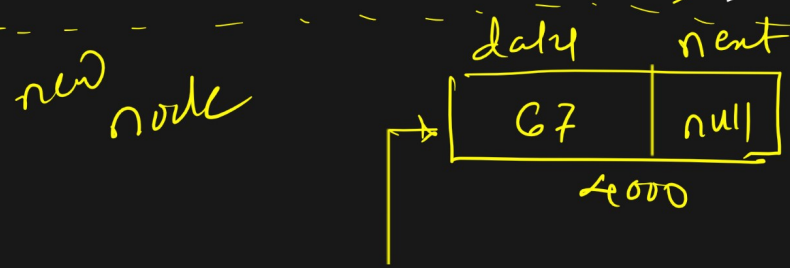

```
    while (temp.next != null)
    {
        temp = temp.next ;
    }
```

2000

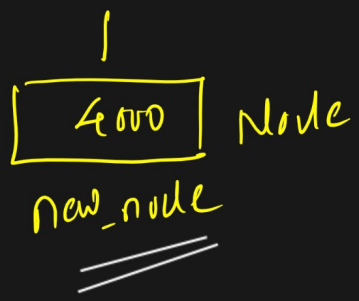
```
    temp.next = new_node;
```

3000 → null 4000

4000




```
}
```


Good
Node
new_node