DST

PROGRAM 01

Singly Linked List

In [4]:

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

# Class SSL (Singly Linked List)
class SSL:
    def __init__(self):
        self.head = None

    # Function to display the list
    def display(self):
        temp = self.head
        while temp:
            print(temp.data, end=" --> ")
            temp = temp.next
        print("None")

    def insert_at_start(self, data):
        new_node = Node(data)
        new_node.next = self.head
        self.head = new_node

    def insert_at_end(self, data):
        new_node = Node(data)
        if not self.head:
            self.head = new_node
            return
        temp = self.head
        while temp.next:
            temp = temp.next
        temp.next = new_node

    def delete(self, key):
        temp = self.head
        if temp and temp.data == key:
            self.head = temp.next
            return
        prev = None
        while temp and temp.data != key:
            prev = temp
            temp = temp.next
        if not temp:
            print(f"{key} not found")
            return
        prev.next = temp.next

linked_list = SSL()

linked_list.insert_at_start(10)
linked_list.insert_at_start(20)
linked_list.display()
linked_list.insert_at_end(30)
linked_list.display()
linked_list.delete(10)
linked_list.display()

```

```
20 --> 10 --> None
20 --> 10 --> 30 --> None
20 --> 30 --> None
```

In [ ]:
```
1
```