Question Nos.

Marks Awarded

Q2.

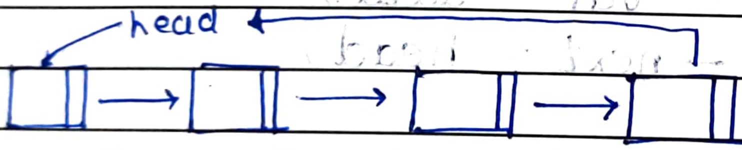
1. Insertion at the start

```
struct node {
    int val;
    ~~struct node *prev~~
    struct node *next;
  };

function Insertatstart (struct node ** head, data
  {
    struct node * newnode;
    newnode → val = data;
    newnode → next = head;
    ~~while (~~
    struct node temp = head
     while ( temp → next != head )
              temp = temp → next
     temp → next = newnode;
  }
```

```
~~function DeletionOfLastnode (struct node ** head, d~~
~~{ struct node prev~~
~~  struct node * temp = head.~~
~~  while (temp → next != head )~~
~~            temp = temp → next~~
~~  temp → next = prev → next~~
~~            prev = temp~~
~~  prev → next = head.~~
```

Question Nos | Marks Awarded

```
free(temp)

function DeletionOfLastNode(struct node **head)
{
    struct node prev
    struct node temp=head
    while (temp->next != head)
        prev->next = temp.
        temp => temp->next
    prev->next = head.
    free(temp)
}.
```



above is a circular link list in this the end node contains the memory location of the head instead of NULL pointer so we can use this to traverse to the last node until the next points to the head.

Insertion at start

create a newnode then traverse to the end node and set the newnode ~~next~~ as the pointer to the last node

Deletion of Last node

traverse to end store the previous node free the Last node and set previous node to head.