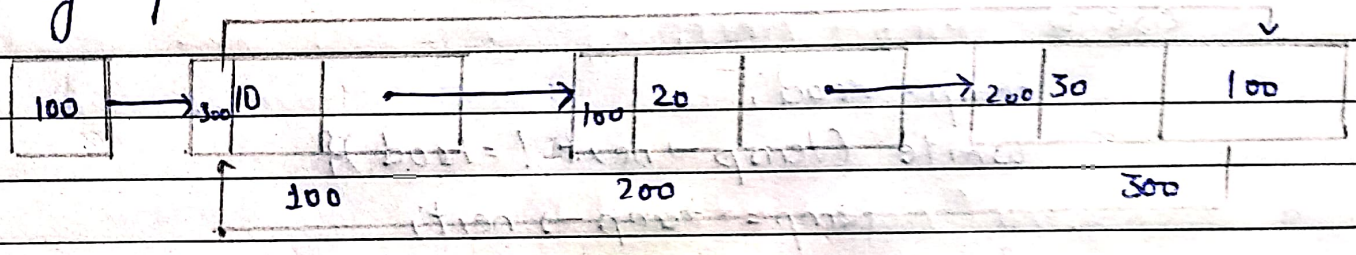
Name. Hitendra Sisodia
SAP Id. 500091910

PAGE NO.
DATE : / /

Assignment -3

Ans 1. Circular doubly linked list has properties of both doubly linked list and circular linked list in which two consecutive elements are linked or connected by previous and next pointer and the last node pointer points to first node by next pointer and also the first node points to the last node by previous node.




```
#include <stdio.h>
struct node
{
    int data;
    struct node *prev;
    struct node * next;
};
{ void insertionhead(struct node **head);
{
    struct node *ptr, *temp;
    int item;
    ptr= (struct node *) malloc (size of (struct node));
    if (ptr= NULL){
        printf ("FULL");
    }
```

```
else{
    scanf("%d", item);
    ptr->data = item;
}
if (head == NULL){
    head = ptr;
    ptr -> next = head;
    ptr -> prev = head;
}
else {
    temp = head;
    while (temp -> next != head){
        temp = temp -> next;
    }
    temp -> next = ptr;
    ptr -> prev = temp;
    head -> prev = ptr;
    ptr -> next = head;
    head = ptr;
}
void insertionend(struct node **head);
{
  struct node *ptr, *temp;
  int item;
  ptr = (struct node *) malloc (sizeof (struct node));
  if (ptr = NULL) {
      printf("FULL");
  }
  else {
      scanf("%d", &item);
```

```
        ptr -> data = item;
        if (head == NULL){
            head = ptr;
            ptr -> next = head;
            ptr -> prev = head;
        }
        else {
            temp = head;
            while (temp -> next != head){
                temp = temp -> next;
            }
            temp -> next = ptr;
            ptr -> prev = temp;
            head -> prev = ptr;
            ptr -> next = head;
        }
    }
}
void deletion_head (struct node **head);
{
    struct node *temp;
    if (head == NULL){
        printf ("Empty");
    }
    else if (head -> next == head){
        head = NULL;
        free (head);
    }
```

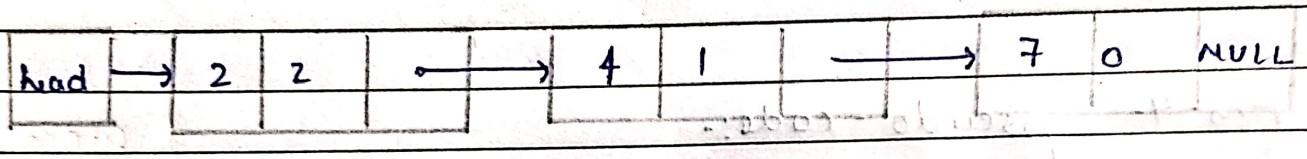
```
else {
    temp = head;
    while (temp -> next != head) {
        temp = temp -> next;
    }
    temp -> next = head -> next;
    head -> next -> prev = temp;
    free (head);
    head = temp -> next;
  }
}
void deletionend (node **head)
{
    struct node *ptr;
    if (head == NULL) {
        printf ("Empty");
    }
    else if (head -> next == head) {
        head = NULL;
        free (head);
    }
    else {
        ptr = head;
        if (ptr -> next != head) {
            ptr = ptr -> next;
        }
        ptr -> prev -> next = head;
        head -> prev = ptr -> prev;
        free (ptr);
    }
}
```

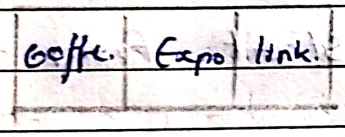
```
void display(struct node **head)
{
    struct node *ptr;
    ptr = head;
    while (ptr->next != head){
        printf("%d", ptr->data);
        ptr = ptr->next;
    }
}

int main()
{
    struct node *head = NULL;
    insertionhead(&head, 24);
    insertionend(&head, 23);
    insertionend(&head, 54);
    deletionhead(&head);
    deletionend(&head);
}
```

PAGE NO.
DATE : / /

Ans-2: An polynomial can be Represented using linked list by Consisting of 3 nodes sections in each node.
The linked list used to represent polynomial of any degree.
An consist of coefficient, Exponent and address of next node.

Eg $2n^2 + 4n^1 + 7n^0$



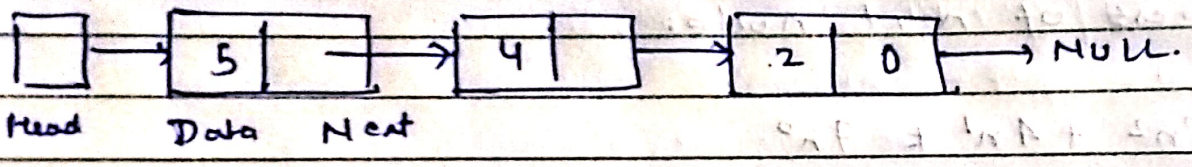


The node sections
Each node divided into 3 sections.
1) Coefficient of first exponent variable.
2) Exponent of first variable.
3) Address of Next nodes.

Applications:

1) Implementation of Stacks and Queues.
2) Implementation of graphs
3) Dynamic memory allocation.
4) Maintaining directory of Names.

A Linked list is a linear data structure, in which the elements are not stored at contiguous memory locations.
The elements in a linked list are linked using pointers as shown in the below: ~~image~~.



Ans-4- pseudo-code:-

① For Reversing linked we start from first node.
② Swap the previous and next pointer fields of current node.
③ Move position of current pointer to its next node, now current prev holds address of next node.
④ Repeat step 2-3 unhill current pointer becomes NULL.
⑤ When current pointer becomes null look at current pointer pointing to last node of linked list
⑥ And, In last swap first and last pointer than we are done.

PAGE NO.
DATE : / /

Ans 3- Algorithum to Removing Duplicate items:..

Step 1: Traverse the list from head to last node.

Step 2: While traversing then Compare Each node with the next node.

Step 3: If the data of the next node is the same as the current node then delete the next node.

Step 4: Before deleting a node we need to store the next pointer of the node.