**Q1.**

**void** insert\_any(**int** item)

    {

        struct node \*ptr = (struct node \*) malloc (sizeof(struct node));

    struct node \*temp;

**int** i,loc;

**if**(ptr == NULL)

        {

            printf("\nOVERFLOW");

        }

**else**

        {

            printf("Enter the location");

        scanf("%d",&loc);

        ptr->data = item;

            temp=head;

**for**(i=0;i<loc;i++)

            {

                temp = temp->next;

**if**(temp == NULL)

                 {

                   printf("\ncan't insert\n");

**return**;

 }

            }

            ptr ->next = temp ->next;

            temp ->next = ptr;

        printf("\nNode inserted");

        }

    }

**Q2.**

void delete\_beg()

{

struct node \*toDelete;

if(head == NULL)

{

printf("List is already empty.");

}

else

{

toDelete = head;

head = head->next;

printf("\nData deleted = %d\n", toDelete->data);

free(toDelete);

}

}

**Q3.**

void deleteLastNode()

{

struct node \*toDelete, \*secondLastNode;

if(head == NULL)

{

printf("List is already empty.");

}

else

{

toDelete = head;

secondLastNode = head;

while(toDelete->next != NULL)

{

secondLastNode = toDelete;

toDelete = toDelete->next;

}

if(toDelete == head)

{

head = NULL;

}

else

{

secondLastNode->next = NULL;

}

free(toDelete);

}

}