**Assignment 3**

**Question 1: Write a function “insert\_any()” for inserting a node at any given position of the linked list. Assume position starts at 0.**

**Function:**

void insert\_any(int data,int position)

{

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

ptr->data=data; //Creating a new node

int i;

struct node \*temp=head;

if(position==0)

{

ptr->next=temp;

head=ptr;

return;

}

for(i=1;i<position-1;i++)

{

temp=temp->next;

}

ptr->next=temp->next;

temp->next=ptr;

}

**Question 2:** **Write a function “delete\_beg()” for deleting a node from the beginning of the linked list.**

**Function:**

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);

/\* Clears the memory occupied by first node\*/

free(toDelete);

printf("SUCCESSFULLY DELETED FIRST NODE FROM LIST\n");

}

}

**Question 3:**

**Write a function “delete\_end()” for deleting a node from the end of the linked list.**

**Function:**

void deleteLastNode()

{

struct node \*toDelete, \*secondLastNode;

if(head == NULL)

{

printf("List is already empty.");

}

else

{

toDelete = head;

secondLastNode = head;

/\* Traverse to the last node of the list \*/

while(toDelete->next != NULL)

{

secondLastNode = toDelete;

toDelete = toDelete->next;

}

if(toDelete == head)

{

head = NULL;

}

else

{

/\* Disconnect link of second last node with last node \*/

secondLastNode->next = NULL;

}

/\* Delete the last node \*/

free(toDelete);

printf("SUCCESSFULLY DELETED LAST NODE OF LIST\n");

}

}