**Assignment-3**

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

**position starts at 0.**

**Ans:**

To add to the beginning of the list, we will need to do the following:

1-> **Create a new item and set its value.**

2-> **Link the new item to point the head of the list.**

3-> **Set the head of the list to be our new item.**

/\* Given a reference (pointer to pointer) to the head of a list

   and an int,  inserts a new node on the front of the list. \*/

void insertAny(struck Node\*\* head\_ref, new\_data)

{

//***Allocate node***

Struct Node\* new\_node = (struct Node\*)malloc(sizeof(struct Node));

//***Put in the data***

new\_node->next = new\_data;

//***Make next of new node as head***

new\_node->next = (\*head\_ref);

}

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

**Ans:** To delete a node from the linked list, we need to do the following steps.

**1-> Find the previous node of the node to be deleted.   
2-> Change the next of the previous node.   
3-> Free memory for the node to be deleted.**

/\* Given a reference (pointer to pointer) to the head of a list

    and a key, deletes the first occurrence of key in linked list \*/

void delete\_beg(struct Node \*\*head\_ref, int key)

{

    // ***Store head node***

    struct Node\* temp = \*head\_ref, \*prev;

    // ***If head node itself holds the key to be deleted***

    if (temp != NULL && temp->data == key)

    {

        \*head\_ref = temp->next;   // ***Changed head***

        free(temp);               // ***free old head***

        return;

    }

**Qus-3:** **Write a function “delete\_end()” for deleting a node from the end of the linked list.**

**Ans:**

void delete\_end()

{

struct node \*temp, \*ptr ;

if(start == NULL)

{

printf(“list is empty: \n”);

exit(0);

}

else if(start->next == NULL)

{

ptr = start;

start = NULL;

printf(“The Deleted Element is: %d”, ptr->info);

free(ptr);

}

else

{

ptr = start;

while(ptr->next != NULL)

{

temp = ptr;

ptr = ptr->next;

}

temp->next = NULL;

printf(“The Deleted Element is: %d”, ptr->info);

free(ptr);

}

}