**Q1. Insert in linked list**

struct node{

int data;

struct node \*next;

};

struct node \*start = NULL;

int totalnumnode = -1;

struct node \*createnode(){

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

newnode->next = NULL;

int data;

printf("Enter the data \n ");

scanf("%d",&data);

newnode->data = data;

return newnode;

}

void insert\_any(){

int pos;

printf("Enter the position u insert node \n");

scanf("%d",&pos);

node \*temp = start;

node \*newnode = createnode();

if(start == NULL){

start = newnode;

printf("node is Inserted \n");

}else{

if(pos >=1){

int counter = 1;

temp = start;

while(counter < pos){

temp = temp->next;

counter++;

}

newnode->next = temp->next;

temp->next = newnode;

printf("Node inserted \n");

}else{

printf("Invalid Position \n");

}

}

}

**Q2. Delete begining in linked list**

void delete\_beg(){

if(start == NULL){

printf("List is empty");

return;

}else

{

node \*temp = start;

start = start->next;

free(temp);

printf("Start node deleted \n");

}

}

**Q3. Delete end in linked list**

void delete\_end(){

if(start == NULL){

printf("List is empty");

return;

}

if(start->next == NULL){

free(start);

start = NULL;

printf("Last node deleted \n");

return;

}

else{

node \*tempnode = start;

while(tempnode->next->next != NULL){

tempnode = tempnode->next;

}

tempnode->next = NULL;

free(tempnode->next);

printf("Last node deleted \n");

}

}

