NAME :- TASMIYA FATHIMA
USN :- 1BM19CS172

M T W T F S S

COMPASS

Date:

LAB-7

WAP to implement singly linked list with following operations

a) Create a linked list.
b) Insertion of a node at first position, at any position and at end of list.
c) Deletion of first element, specified element and last element in the list.
d) Display the contents of the linked list.

```
struct Node
{
  int data;
  struct Node *next;
}

void insert_beg(? struct Node **head)
{
  struct Node *ptr;
  int value;
  ptr= (struct Node *)malloc (sizeof (struct Node));
  if(ptr
      ptr->do
  printf (" Enter the value to be added");
   scanf ("%d", & value);
   ptr->data = value;
   ptr -> next = (*head);
   (*head)= ptr;
  }
```

```
Void insert_end ( struct Node ** head)
{
struct Node * new_node = (struct Node*) malloc
                          (sizeof (struct Node));

struct Node * last = * head .
~~new_node→ data =~~
int value;
printf (" Enter the value to be inserted\n").
scanf(" % d", & value);
new_node → data = value;
new_node → next = NULL;

if (* head == NULL)
 {
   * head = new_node;
   return;
 }
else {
while ( last → next != NULL )
  last = last → next;
  last → next = new_node;
  return;
}


 Void insert_specific (int position )
{
struct node * new_node = (struct node *) malloc( sizeof
                          ( struct node )).

 int value;
printf (" Enter the value to be inserted ").
scanf (" ~~&d~~ % d", & value).
   new_node → data = value;
struct node * temp = head;
if (position == 1)
{
    new_node -> next = temp;
    head = new_node;
    return;
}

for (i=1; i< position-1; i++)
{
    temp = temp -> next;
}
new_node -> next = temp -> next;
temp -> next = new_node;
}

void delete_front ()
{
  struct node * ptr;
 if (head == NULL)
 {
    printf ("List is empty");
 }

else
  {
     ptr = head;
     head = ptr -> next;
     free (ptr);
printf ("\n Node deleted from the beginning");
}
}
```

```
Void delete_end()
{
    struct node * ptr, * ptr1;
      if (head == NULL)
       {
            printf ("\n list is empty");
       }
    else if (head -> next == NULL)
     {
           head = NULL;
           free (head);
     printf ("Only node of the list ");
  }
  else
    {
        ptr = head;
        while ( ptr -> next != NULL)
         {
              ptr1 = ptr ;
               ptr = ptr -> next ;
         }
     ptr1 -> next = NULL;
       free (ptr);
   printf (" Deleted Node from the list \n");
  }
 }


 Void delete_speific()
 {
    Struct node * ptr, * ptr1;
    int loc, j;
    Scanf (" %d", & loc) ;
      ptr = head ;
   for(i=0; i< loc; i++)
```

```
{
  ptr1= ptr;
   ptr=   ptr→ next;

  if (ptr==~~!~~ NULL)
 {
  printf(" There are less than %d elements
          in the list \n", loc);
 }
 }
 ptr1→ next = ptr→ next;
 free (ptr)
 printf ("\n Deleted %d node", loc);
}


void  display ()
{
    Struct   node * ptr;
      ptr= head;
       if( ptr==NULL)
      {
         printf (" Nothing  to print ");
    else
      {
         printf (" Values are " );
             while (ptr!= NULL)
             {
                 printf ("\n %d", ptr→data);
                    ptr=ptr→ next;
             }
            }
           }
```