AKASH
1BM18CS008

DATE

```
struct node
{
int data;
struct node * next;
};
struct node * head = NULL;

void create()
{
  int ele;
  struct node * newnode, * temp;
  new node = (struct node *) malloc (size of struct node));
  printf("Enter data to be inserted: ");
  scanf("%d", &ele);
  new node -> data = ele;
  if (head == NULL)
  {
    new node -> next = NULL;
    head = new node;
  }
  else
  {
    temp = head;
    while (temp -> next != NULL)
        temp = temp -> next;
      temp -> next = new node;
      new node -> next = NULL;
  }
}
void delfront()
{
  struct node * temp, *d = NULL;
  int ele;
  if (head == NULL)
  {
```

```
        printf("list is empty !!! \n");
        return ;
    }
temp = head;
printf("Enter value to be removed: ");
scanf("%d", &ele);
if (head -> data == ele)
{
    del = head;
    head = head -> next;
}
while (temp -> next != NULL)
{
    if (temp -> next -> data == ele)
    {
    del = temp -> next;
    if (next == NULL)
    temp -> next = NULL;
else
    temp = temp -> next;
}
if (del == NULL)
    printf("Elements has not been found");
}
void display()
{
    struct node *temp = NULL;
    temp = head;
    if (temp == NULL)
        printf("No elements in list! \n");
    else
        while (temp != NULL)
        {
            printf("%d \t", temp -> data);
            temp = temp -> next;
```

DATE

```
    }
    printf("\n");
}
void insertbeg()
{
    struct node *newnode;
    int ele;
    printf("Enter element to be added");
    scanf("%d", &ele);
    newnode = (struct node*) malloc(sizeof struct node));
    newnode -> data = ele;
    newnode -> next = head;
    head = newnode;
}
```