Step 1: Define Structure

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

        typedef Node node;
        node *list= NULL,*tail=NULL;
```

Step 2: Write Function

```
        void insertAtHead()
        {
                node *p;
                int ele;
                printf("\n Enter Element to insert At Head:\n");
                scanf("%d",&ele);
                p=(node*)malloc(sizeof(node));
                p->info = ele;
                p->next=list;
                if(NULL==list)
                {

                        tail=list=p;
                }
                else
                        list = p;

                printf("\n %d is added to the Linked List\n",list->info);
        }

int deleteFromHead()
{
        int ele;

        if(NULL == list )
        {
                printf("\nThere are no element in the Linked List to delete\n");
                return NULL;
        }

        else
        {
```

```
                ele = list->info;
                list=list->next;
                //return ele;
        }
        return ele;
}

void insertAtTail()
        {
                node *p;
                int ele;
                printf("\n Enter Element to insert At Tail:\n");
                scanf("%d",&ele);
                p=(node*)malloc(sizeof(node));
                p->info = ele;
                p->next=NULL;
                if(NULL==tail)
                        tail=list=p;
                else
                {
                        tail->next = p;
                        tail=p;
                }

                printf("\n %d is added to the tail of a Linked List\n",tail->info);
        }
int deleteFromTail()
{
        int ele;

        if(NULL ==  tail )
        {
                printf("\nThere are no element in the Linked List to delete\n");
                return NULL;
        }

        else
        {
                for(node *temp=list; temp->next!=NULL; temp=temp->next)
                ;
                ele = tail->info;
                free(tail);
                tail = temp;
```

```
                list=list->next;
                //return ele;
        }
        return ele;
}

void displayAll()
{
        printf("\n The Element Stored in Linked List Are:\n");
        for(node * temp = list ; temp!= NULL; temp= temp->next)
                printf("%d\t",temp->info);
}

step 3: Write void main
void main()
{
        char ch;
        int choice;
    //  clrscr();
        do{
                clrscr();
                printf("\n\tMENU\n");
                printf("1 -> Insert a Node at Head\n");
                printf("2 -> Delete a Node From Head\n");
                printf("3 -> Insert a Node at Tail\n");
                printf("4 -> Delete a Node from Tail\n");
                printf("5 -> Display All\n");
                printf("Enter your Choice\n");
                scanf("%d",&choice);
                switch(choice)
                {
                        case 1:
                                insertAtHead();
                                break;
                        case 2:
                                if(list== NULL)
                                        printf("Linked List is Empty:\n");
                                else
                                        printf("%d is deleted From Linked
List\n",deleteFromHead());
                                break;
                        case 3:
```

```
                        insertAtTail();
                        break;
                case 4:
                        if(tail == NULL)
                                printf("Linked List is Empty:\n");
                        else
                                printf("%d is deleted From Linked List\n",deleteFromTail());
                        break;
                case 5:
                        if(list== NULL)
                                printf("Linked List is Empty:\n");
                        else
                                displayAll();
                        break;
            }
            fflush(stdin);
            printf("Do you want to continue:(Y/N)\n");
            ch = toupper(getchar());
        }while(ch=='Y');


    //  getch();
}
```