WAP to Implement Singly Linked List with following operations

a) Create a linked list.

b) Deletion of first element, specified element and last element in the list.

c) Display the contents of the linked list.

```
#include <stdio.h>
#include <stdlib.h>

struct node{
   int info;
   struct node* next;
};

struct node* createlk(){
   struct node*p;
   struct node*start=NULL;
   struct node*last;
   int item;
   printf("enter -999 to exit\n");
   scanf("%d",&item);
   while(item!=-999){
      p=(struct node*)malloc(sizeof(struct node));
      p->info=item;
      if(start==NULL){
         p->next=NULL;
         start=p;
         last=p;
      }
```

```
        else{
          p->next=NULL;
          last->next=p;
          last=p;
        }
        scanf("%d",&item);
    }
    return start;
}

struct node * deletefirst(struct node * start){
    struct node * temp;

    if(start==NULL){
        printf("linked list is empty\n");
    }
    else if(start->next==NULL){
        temp=start;
        start=NULL;
        free(temp);
    }
    else{
        temp=start;
        start=start->next;
        free(temp);
    }
    return start;
```

```
}

struct node * deletelast(struct node * start){
   struct node * prev,*temp;
   if(start==NULL){
      printf("linked list is empty \n");
   }
   else if(start->next==NULL){
      temp=start;
      start=NULL;
      free(temp);
   }
   else{
      temp=start;
      while(temp->next!=NULL){
         prev=temp;
         temp=temp->next;

      }
      prev->next=NULL;
      free(temp);
   }
   return start;
}

struct node * delelement(struct node * start,int element){
   struct node *prev, *temp;
```

```
if(start == NULL){
   printf("linked list is empty \n");
   return start;
}

temp = start;



if(temp->info == element){
   start = start->next;
   free(temp);
   return start;
}



while(temp != NULL && temp->info != element){
   prev = temp;
   temp = temp->next;
}

if(temp == NULL){
   printf("element not found\n");
}
else{
   prev->next = temp->next;
   free(temp);
}
```

```
    return start;
}

void displaylk(struct node*start){
  struct node*temp;
  if(start==NULL){
    printf("linked list is empty\n");
  }
  else{
    temp=start;
    printf("elements are\n");
    while(temp!=NULL){
       printf("%d\n",temp->info);
       temp=temp->next;
    }
  }
}

int main(){
    struct node* head=NULL;
    int choice;
    int ele;
    int val;
    while(1){
    printf(" Linked list operations \n");
    printf(" 1)Create linked list \n" );
    printf(" 2)delete at first  \n" );
    printf(" 3)delete at last  \n" );
```

```
printf(" 4)delete element \n" );
printf(" 5)Display \n" );
printf(" 6)Exit \n ");

printf("Enter your choice \n");
scanf("%d",&choice);
switch(choice){

   case 1:
   head=createlk();
   break;

   case 2:
   head=deletefirst(head);
   break;

   case 3:
   head=deletelast(head);
   break;

   case 4:
   printf("enter value to delete: \n");
   scanf("%d",&ele);
   head=delelement(head,ele);
   break;

   case 5:
   displaylk(head);
```

```
        break;

        case 6:
        printf("Exiting program \n");
        return 0;

        default:
        printf("Invalid choice \n");
    }

}
return 0;
}
```

Output:

```
PS C:\Users\n6787\OneDrive\Desktop\c> cd "c:\Users\n6787\OneDrive\Desktop\c\big.c\" ; if ($?) { gcc linked2.c -o linked2 } ; if ($?) { .\linked2 }
 Linked list operations
 1)Create linked list
 2)delete at first
 3)delete at last
 4)delete element
 5)Display
 6)Exit
 Enter your choice
1
enter -999 to exit
5
4
3
2
1
-999
 Linked list operations
 1)Create linked list
 2)delete at first
 3)delete at last
 4)delete element
 5)Display
 6)Exit
 Enter your choice
2
 Linked list operations
 1)Create linked list
 2)delete at first
 3)delete at last
 4)delete element
 5)Display
 6)Exit
 Enter your choice
3
```

```
 Linked list operations
 1)Create linked list
 2)delete at first
 3)delete at last
 4)delete element
 5)Display
 6)Exit
 Enter your choice
4
enter value to delete:
4
 Linked list operations
 1)Create linked list
 2)delete at first
 3)delete at last
 4)delete element
 5)Display
 6)Exit
 Enter your choice
5
elements are
3
2
 Linked list operations
 1)Create linked list
 2)delete at first
 3)delete at last
 4)delete element
 5)Display
 6)Exit
 Enter your choice
6
Exiting program
```