Singly Linked List — Deletion

IBM19CS047
Derek Stanley
Kannathe

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

void delfun()
{ struct node *temp, *del = NULL;
  if(head == NULL)
  { printf("Empty List.\n"); return; }
  temp = head;
  int ch, ele;
  printf("Delete at:\n1. Front\n2. Back\n3. Desired Element\n Enter
  choice :");
  scanf("%d",&ch);
  { case 1: del = head;
            head = head->next;
            printf("Node Deleted.");
            break;
    case 2: while(temp->next->next!=NULL)
            { temp = temp->next;}
            del = temp->next;
            temp->next = NULL;
            printf("Node Deleted\n.");
            break;
```


Derek

```
case 3: printf(" Enter the element to delete\n");
        scanf(" %d", &ele);
        if(head -> data == ele)
        { head = head -> next; }
        else
        while (temp -> next != NULL)
        { if (temp -> next -> data == ele)
          { del = temp -> next;
            temp -> next = del -> next; }
          else
          { temp = temp -> next; }
        }

default: if (ch != 4)
         { printf("\n Enter valid choice "); }
  }
if(del == NULL)
{ printf(" Element not found in the List \n"); }
}
void insert()
{ struct node *temp, *newnode;
  int item;
  newnode = (struct node *) malloc(sizeof(struct node));
  printf(" Enter the data:");
  scanf("%d ", &item);
  newnode -> data = item;
  if(head == NULL)
  { head = newnode; }
  else
  { temp = head;
```


Derek

1BM19CS045
Derek Stanley
Kannatha

```
while (temp -> next != NULL)
{ temp = temp -> next; }
temp -> next = newnode;
newnode -> next = NULL;
printf(" Node created\n");
}
}
```


Derek