classmate
Date
Page

15/01/24 DA WEEK-6

1 Implement Single linked list to Sort, Reverse, Concatenate of two linked lists.

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

//list node structure
 struct Node {
   int data;
   struct Node* next;
  };

void append(struct Node** head_ref, int
  new_data){
     struct Node* new_node = (struct Node*)
     malloc(sizeof(struct Node));
     struct Node* last = *head_ref;
     new_node -> data = new_data;
     new_node -> next = NULL;

     if(*head_ref == NULL) {
       *head_ref = new_node;
       return;
     }
```

[illegible] 23/1/24

```
while (last -> next != NULL) {
      last = last -> next;
  }
  last -> next = new_node;
}

void printlist(struct Node *node){
    while(node != NULL) {
    printf("%d->", node -> data);
    node = node -> next;
  }
  printf("NULL\n");
}

void sortlist(struct Node** head_ref){
    if(*head_ref == NULL) {
       return;
  }
  int swapped, temp;
  struct Node* ptr1;
  struct Node* lptr = NULL;

do{
  swapped = 0;
  ptr1 = *head_ref;
while(ptr1->next!=lptr){
    if(ptr1->data>ptr1->next->data){
        temp=ptr1->data;
        ptr1->data=ptr1->next->data;
        ptr1->next->data=temp;
        swapped=1;
    }
    ptr1=ptr1->next;
}
lptr=ptr1;
}while(swapped);
}


void reverselist(struct Node** head_ref){
    struct Node* prev=NULL;
    struct Node* current=*head_ref;
    struct Node* next=NULL;

    while(current!=NULL){
        next=current->next;
        current->next=prev;
        prev=current;
        current=next;
    }
    *head_ref=prev;
}
```

```
void concatenateLists(struct Node **
head1, struct Node *head2){
    if(*head1 == NULL){
      *head1 = head2);
     return;
    }

   struct Node* temp = *head1;
     while(temp->next != NULL){
        temp = temp->next;
     }
     temp->next = head2;
  }

int main(){
     struct Node* list1 = NULL
     struct Node* list2 NULL

     int n, data;
      printf("Enter the no of elements for list1");
      scanf("%d",&n);

       printf("Enter the element for list1:\n");
       for(int i=0;i<n;i++){
```

```
scanf("%d", &data);
    append(&list1, data);
}
 printf("Enter the elements for list 2:\n");
 for (int i=0; i<n; ++i){
     scanf("%d", &data);
     append(&list2, data);
  }


 printf("Orginal list1:");
  printlist(list1);
  printf("Orginal list2");
  printlist(list2);

sortlist(&list1);
sortlist(&list2);

 printf("\n Sorted list1");
 printlist(list1)
printf("Sorted list:2");
printlist(list2);

concatenatelists(&list1, list2);
 printf("\n concatenated list:");
  printlist(list1);
```

```
    reverse list(&list1);

    printf("\n reversed list:");
    printf(list2);

    return 0;
}
```

Output :- Enter the numbers of elements for list 1: 3
Enter the elements for list 1: 1,2,3
Enter the elements for list 2: 4,5,6

Original list1: 1 2 3
Original list2: 4,5 6

Sorted list: 1 2 3
Reversed list: 6 5 4

Concatenated list: 1 2 3 4 5 6