Expt. No. 7 Aditi Akarsh 1BM19CS007 Date Page No. 1

```
#include<stdlib.h>
#include<stdio.h>
#include<malloc.h>
Struct node{
        int data;
        struct node *next; };
Struct node *start = NULL, *start1 = NULL
Struct node *create_ll (Struct node*);
Struct node *create_ll2(Struct node*);
Struct node *display (Struct node*);
Struct node *insert (Struct node*);
Struct node *delete (Struct node*);
Struct node *reverse (Struct node*);
Struct node *concat (Struct node*, Struct node*)
Struct node *sort (Struct node*);
int main(){
    int option;
do{
    printf("\n1: Create \n2:Display 3. insert\n
    delete\n5. reverse 6. concat 7. sort 8. exit");
    printf("Enter choice");
    scanf("%d",&option);
    switch(option){

case 1: start=create_ll(start);
        printf("Linked list created");
        break;
case 2: start=display(start);
        break;
```

Teacher's Signature

```
Case 3: start=insert(start);
        break;
Case 4: start=delete(start);
        break;
Case 5: start=reverse(start);
        break;
Case 6: start = concat(start,start1);
        break;
Case 7: start = sort(start);
        break;
    }
} while(option!=8);
return 0;
}

struct node *create_ll(struct node *start){
  struct node *ptr, *new_node;
    int num;
  printf("Enter -1 to end");
  printf("Enter data");
  scanf("%d", &num);
while(num!=-1){
  new_node=(struct node*)malloc
  (sizeof(struct node));
new_node->data=num;
  if(start==NULL){
    new_node->next=NULL;
    start=new_node;
  }
```

```
else {
        ptr = start;
        while(ptr -> next != NULL)
            ptr = ptr -> next;
            ptr -> next = new_node;
            new_node -> next = NULL;
        }
        printf("Enter data")
        scanf(" %d", &num);
    }
    return start;
}

struct node *display (struct node *start){
        struct node *ptr;
        ptr = start;
        while(ptr != NULL)
    {
        printf("\t%d", ptr -> data);
        ptr = ptr -> next;
    }
    return start;
}

struct node *insert (struct node *start){
    struct node *new node;
    int num;
    printf("Enter data");
    scanf("%d", &num);
```

```
newnode = (struct node *) malloc( size of (struct node));
newnode -> data = num;
new node -> next = start;
 start = newnode
 return start;
}

struct node* delete (struct node * start){
        struct node * ptr;
        ptr = start;
        start = start -> next;
        free (ptr);
        return start;
};

struct node * reverse (struct node * start){
    struct node * prev = NULL, * current = start,
    * next = NULL;
    while(current != NULL)
    {
        next = current -> next;
        Current -> next = prev;
        prev = current; current = next; }
        start = prev;
        return start;
}
```

Teacher's Signature

Expt. No. 7 Date

1BM19CS007 Page No. 6

```
struct node *sort(struct node *start) {
                 struct node *start1) {
    struct node *ptr1, *ptr2;
    int temp;
    ptr1 = start;
    while (ptr1->next != NULL) {
        ptr = ptr1->next;
    while (ptr2 != NULL) {
        if (ptr1->data > ptr2->data) {
            temp = ptr1->data;
            ptr1->data = ptr2->data;
            ptr2->data = temp; }
            ptr2 = ptr2->next; }
        ptr1 = ptr1->next;
    } return start;
}

struct node * concat (struct node *start,
                      struct node *start1) {
    struct node *ptr;
    ptr = start;
    printf("Enter second list");
    start1 = create_ll(start1);
    while(ptr->next != NULL)
        ptr = ptr->next;
        ptr->next = start1;
        return start;
}
```

Teacher's Signature