6(a)_ WAP to Implement Single Link List with following operations: Sort the linked list, Reverse the linked list, Concatenation of two linked lists. #include <stdlib.h>

```
/* Definition of node */
struct node {
   int data;
   struct node *next;
};

/* Insert at end */
struct node* insertEnd(struct node *head, int value) {
   struct node *newnode = (struct node*)malloc(sizeof(struct node));
   newnode->data = value;
   newnode->next = NULL;

   if (head == NULL)
      return newnode;

   struct node *temp = head;
   while (temp->next != NULL)
      temp = temp->next;

   temp->next = newnode;
   return head;
}

/* Display list */
void display(struct node *head) {
```

```
    if (head == NULL) {
        printf("List is empty\n");
        return;
    }
    while (head != NULL) {
        printf("%d -> ", head->data);
        head = head->next;
    }
    printf("NULL\n");
}

/* Sort linked list (Bubble Sort) */
struct node* sortList(struct node *head) {
    struct node *i, *j;
    int temp;

    for (i = head; i != NULL; i = i->next) {
        for (j = i->next; j != NULL; j = j->next) {
            if (i->data > j->data) {
                temp = i->data;
                i->data = j->data;
                j->data = temp;
            }
        }
    }
    return head;
}
```

```
/* Reverse linked list */
struct node* reverseList(struct node *head) {
   struct node *prev = NULL, *current = head, *next = NULL;

   while (current != NULL) {
      next = current->next;
      current->next = prev;
      prev = current;
      current = next;
   }
   return prev;
}

/* Concatenate two linked lists */
struct node* concatenate(struct node *head1, struct node *head2) {
   if (head1 == NULL)
      return head2;

   struct node *temp = head1;
   while (temp->next != NULL)
      temp = temp->next;

   temp->next = head2;
   return head1;
}

/* Main function */
int main() {
```

```
struct node *list1 = NULL, *list2 = NULL;

/* Create first list */
list1 = insertEnd(list1, 30);
list1 = insertEnd(list1, 10);
list1 = insertEnd(list1, 20);

printf("First Linked List:\n");
display(list1);

/* Sort list */
list1 = sortList(list1);
printf("\nSorted Linked List:\n");
display(list1);

/* Reverse list */
list1 = reverseList(list1);
printf("\nReversed Linked List:\n");
display(list1);

/* Create second list */
list2 = insertEnd(list2, 40);
list2 = insertEnd(list2, 50);

printf("\nSecond Linked List:\n");
display(list2);

/* Concatenate lists */
```

```
    list1 = concatenate(list1, list2);

    printf("\nConcatenated Linked List:\n");

    display(list1);


    return 0;

}
```