

## LAB PROGRAM 6B

WAP to Implement Single Link List to simulate Stack & Queue Operations.

```
#include <stdio.h>

#include <stdlib.h>

struct Node {
    int data;
    struct Node* next;
};

struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}

void append(struct Node** head, int data) {
    struct Node* newNode = createNode(data);
    if (*head == NULL) {
        *head = newNode;
        return;
    }
    struct Node* temp = *head;
    while (temp->next != NULL) {
        temp = temp->next;
    }
    temp->next = newNode;
}

void display(struct Node* head) {
```

```

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

```

```

void 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;
            }
        }
    }
}

```

```

void 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;
    }
    *head = prev;
}

```

```
}
```

```
struct Node* concatenate(struct Node* head1, struct Node* head2) {  
    if (head1 == NULL) return head2;  
    if (head2 == NULL) return head1;  
  
    struct Node* temp = head1;  
    while (temp->next != NULL) {  
        temp = temp->next;  
    }  
    temp->next = head2;  
    return head1;  
}
```

```
int main() {  
    struct Node* list1 = NULL;  
    struct Node* list2 = NULL;  
    int n1, n2, data, i;  
  
    printf("Enter the number of elements in List 1: ");  
    scanf("%d", &n1);  
    printf("Enter the elements of List 1:\n");  
    for (i = 0; i < n1; i++) {  
        scanf("%d", &data);  
        append(&list1, data);  
    }  
  
    printf("Enter the number of elements in List 2: ");  
    scanf("%d", &n2);  
    printf("Enter the elements of List 2:\n");  
    for (i = 0; i < n2; i++) {
```

```
        scanf("%d", &data);
        append(&list2, data);
    }

    printf("List 1: ");
    display(list1);

    printf("List 2: ");
    display(list2);

    sortList(&list1);
    printf("Sorted List 1: ");
    display(list1);

    reverseList(&list1);
    printf("Reversed List 1: ");
    display(list1);

    struct Node* mergedList = concatenate(list1, list2);
    printf("Concatenated List: ");
    display(mergedList);

    return 0;
}
```