

C SORT.c > sort(Node *)

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 struct Node{
5     int data;
6     struct Node* next;
7 };
8
9 struct Node* createNode(int data){
10    struct Node *newNode = (struct Node*)malloc(sizeof(struct Node));
11    newNode->data = data;
12    newNode->next = NULL;
13    return newNode;
14 }
15 struct Node* createList(){
16    struct Node *head = NULL, *temp, *newNode;
17    int n, data;
18    printf("Enter number of nodes: ");
19    scanf("%d",&n);
20    for(int i =0;i < n; i++){
21        printf("Enter data: ");
22        scanf("%d",&data);
23        newNode = createNode(data);
24        if(head == NULL) head = newNode;
25        else {
26            temp = head;
27            while(temp->next != NULL)
28                temp = temp->next;
29            temp->next = newNode;
30        }
31    }
32    return head;
33 }
34 void display(struct Node *head){
35    while(head != NULL){
36        printf("%d->",head->data);
37        head = head->next;
38    }
39    printf("NULL\n");
40 }
41 void sort(struct Node* head){}
42    struct Node *i, *j;
43    int temp;
44    for(i = head; i != NULL; i = i->next){
45        for(j = i->next; j != NULL; j = j->next){
46            if(i->data > j->data){
47                temp = i->data;
48                i->data = j->data;
49                j->data = temp;
50            }
51        }
52    }
53 }
```

C SORT.c > sort(Node *)

```
51     struct Node *reverse(struct Node* head){
52         struct Node *prev = NULL, *curr = head,
53         *next = NULL;
54         while(curr != NULL){
55             next = curr->next;
56             curr->next = prev;
57             prev = curr;
58             curr= next;
59         }
60         return prev;
61     }
62     struct Node *concatenate(struct Node *head1, struct Node *head2){
63         if(head1 == NULL) return head2;
64         if(head2 == NULL) return head1;
65         struct Node *temp = head1;
66         while(temp->next != NULL)
67             temp = temp->next;
68         temp->next = head2;
69         return head1;
70     }
71     int main(){
72         struct Node *list1 = NULL, *list2 = NULL;
73         printf("Create List 1: \n");
74         list1 = createList();
75
76         printf("\nCreate List 2: \n");
77         list2 = createList();
78
79         printf("\nList 1: ");
80         display(list1);
81
82         printf("List 2: ");
83         display(list2);
84
85         sort(list1);
86         printf("\nList 1 After sorting: ");
87         display(list1);
88
89         list1 = reverse(list1);
90         printf("List1, After Reversing: ");
91         display(list1);
92
93         list1 = concatenate(list1, list2);
94         printf("After concatenation (list1 + list2): ");
95         display(list1);
96         return 0;
97     }
```

PROBLEMS 8

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

QUERY RESULTS

```
PS C:\Users\gsm22\OneDrive\Documents\DS> cd "c:\Users\gsm22\OneDrive\Do
Create List 1:
Enter number of nodes: 4
Enter data: 23
Enter data: 2
Enter data: 65
Enter data: 12

Create List 2:
Enter number of nodes: 3
Enter data: 86
Enter data: 59
Enter data: 36

List 1: 23->2->65->12->NULL
List 2: 86->59->36->NULL

List 1 After sorting: 2->12->23->65->NULL
List1, After Reversing: 65->23->12->2->NULL
After concatenation (list1 + list2): 65->23->12->2->86->59->36->NULL
PS C:\Users\gsm22\OneDrive\Documents\DS>
```