

# Program 6a

```
CUUsers\BMS\CECSP\Documents\lab\sort, reverse,concat.oprc - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools ASStyle Window Help
(globals)
Project Classes Debug [?] sort, reverse,concat.oprc
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct Node {
5     int data;
6     struct Node* next;
7 };
8
9 // Function to create a new node
10 struct Node* createNode(int data) {
11     struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
12     newNode->data = data;
13     newNode->next = NULL;
14     return newNode;
15 }
16
17 // Function to insert node at the end
18 struct Node* insertEnd(struct Node* head, int data) {
19     struct Node* newNode = createNode(data);
20     if (head == NULL)
21         return newNode;
22
23     struct Node* temp = head;
24     while (temp->next != NULL)
25         temp = temp->next;
26
27     temp->next = newNode;
28     return head;
29 }
30
31 // Function to display linked list
32 void display(struct Node* head) {
33     struct Node* temp = head;
34     while (temp != NULL) {
35         printf("%d -> ", temp->data);
36         temp = temp->next;
37     }
38     printf("NULL\n");
39 }
```

```
CUUsers\BMS\CECSP\Documents\lab\sort, reverse,concat.oprc - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools ASStyle Window Help
(globals)
Project Classes Debug [?] sort, reverse,concat.oprc
40
41 // Function to sort linked list (ascending)
42 void sortList(struct Node* head) {
43     struct Node* i, *j;
44     int temp;
45     for (i = head; i != NULL; i = i->next) {
46         for (j = i->next; j != NULL; j = j->next) {
47             if (i->data > j->data) {
48                 temp = i->data;
49                 i->data = j->data;
50                 j->data = temp;
51             }
52         }
53     }
54 }
55
56 // Function to reverse linked list
57 struct Node* reverseList(struct Node* head) {
58     struct Node* prev = NULL, *curr = head, *next = NULL;
59     while (curr != NULL) {
60         next = curr->next;
61         curr->next = prev;
62         prev = curr;
63         curr = next;
64     }
65     return prev;
66 }
67
68 // Function to concatenate two linked lists
69 struct Node* concatenate(struct Node* list1, struct Node* list2) {
70     if (list1 == NULL) return list2;
71     struct Node* temp = list1;
72     while (temp->next != NULL)
73         temp = temp->next;
74     temp->next = list2;
75     return list1;
76 }
```

```

76 }
77
78 // Main function
79 int main() {
80     struct Node *list1 = NULL, *list2 = NULL;
81     int choice, data;
82
83     while (1) {
84         printf("1. Insert in List 1\n");
85         printf("2. Insert in List 2\n");
86         printf("3. Display List 1\n");
87         printf("4. Display List 2\n");
88         printf("5. Sort List 1\n");
89         printf("6. Reverse List 1\n");
90         printf("7. Concatenate List1 & List2\n");
91         printf("8. Exit\n");
92         printf("Enter your choice: ");
93         scanf("%d", &choice);
94
95         switch (choice) {
96             case 1:
97                 printf("Enter data: ");
98                 scanf("%d", &data);
99                 list1 = insertEnd(list1, data);
100                 break;
101             case 2:
102                 printf("Enter data: ");
103                 scanf("%d", &data);
104                 list2 = insertEnd(list2, data);
105                 break;
106             case 3:
107                 printf("List 1: ");
108                 display(list1);
109                 break;
110             case 4:
111                 printf("List 2: ");
112                 display(list2);
113                 break;
114             case 5:
115                 sortList(list1);
116                 printf("List 1 Sorted.\n");
117                 break;
118             case 6:
119                 list1 = reverseList(list1);
120                 printf("List 1 Reversed.\n");
121                 break;
122             case 7:
123                 list1 = concatenate(list1, list2);
124                 printf("After Concatenation: ");
125                 display(list1);
126                 break;
127             case 8:
128                 exit(0);
129             default:
130                 printf("Invalid choice!\n");
131         }
132     }
133
134     return 0;

```

```

101
102     printf("Enter data: ");
103     scanf("%d", &data);
104     list2 = insertEnd(list2, data);
105     break;
106 case 3:
107     printf("List 1: ");
108     display(list1);
109     break;
110 case 4:
111     printf("List 2: ");
112     display(list2);
113     break;
114 case 5:
115     sortList(list1);
116     printf("List 1 Sorted.\n");
117     break;
118 case 6:
119     list1 = reverseList(list1);
120     printf("List 1 Reversed.\n");
121     break;
122 case 7:
123     list1 = concatenate(list1, list2);
124     printf("After Concatenation: ");
125     display(list1);
126     break;
127 case 8:
128     exit(0);
129 default:
130     printf("Invalid choice!\n");
131 }
132
133
134     return 0;

```

**Output:**

```
C:\Users\BMSCE\Documents\ X + v - □ X

----- MENU -----
1. Insert in List 1
2. Insert in List 2
3. Display List 1
4. Display List 2
5. Sort List 1
6. Reverse List 1
7. Concatenate List1 & List2
8. Exit
Enter your choice: 1
Enter data: 23

----- MENU -----
1. Insert in List 1
2. Insert in List 2
3. Display List 1
4. Display List 2
5. Sort List 1
6. Reverse List 1
7. Concatenate List1 & List2
8. Exit
Enter your choice: 1
Enter data: 45

----- MENU -----
1. Insert in List 1
2. Insert in List 2
3. Display List 1
4. Display List 2
```

```
C:\Users\BMSCE\Documents\ X + v - □ X

----- MENU -----
1. Insert in List 1
2. Insert in List 2
3. Display List 1
4. Display List 2
5. Sort List 1
6. Reverse List 1
7. Concatenate List1 & List2
8. Exit
Enter your choice: 2
Enter data: 34

----- MENU -----
1. Insert in List 1
2. Insert in List 2
3. Display List 1
4. Display List 2
5. Sort List 1
6. Reverse List 1
7. Concatenate List1 & List2
8. Exit
Enter your choice: 67
Invalid choice!

----- MENU -----
1. Insert in List 1
2. Insert in List 2
3. Display List 1
4. Display List 2
```

```
C:\Users\BMSCE\Documents\ x + v
----- MENU -----
1. Insert in List 1
2. Insert in List 2
3. Display List 1
4. Display List 2
5. Sort List 1
6. Reverse List 1
7. Concatenate List1 & List2
8. Exit
Enter your choice: 6
List 1 Reversed.

----- MENU -----
1. Insert in List 1
2. Insert in List 2
3. Display List 1
4. Display List 2
5. Sort List 1
6. Reverse List 1
7. Concatenate List1 & List2
8. Exit
Enter your choice: 7
After Concatenation: 98 -> 45 -> 23 -> 34 -> 67 -> NULL

----- MENU -----
1. Insert in List 1
2. Insert in List 2
3. Display List 1
4. Display List 2
```