Start here | Linked_lists_sort_reverse_concatenate.c

```c
// Sneha ID 1BM24CS185
#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 insertEnd(struct Node **head, int data) {
    struct Node *newNode = createNode(data);
    if (*head == NULL) {
        *head = newNode;
    } else {
        struct Node *temp = *head;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = newNode;
    }
}

void display(struct Node *head) {
    if (head == NULL) {
        printf("List is empty\n");
    } else {
        struct Node *temp = head;
        while (temp != NULL) {
            printf("%d ", temp->data);
            temp = temp->next;
        }
        printf("\n");
    }
}

```

Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppCheck/Vera++ messages | Cscope | Debugger | DoxyBlocks | Fortran info | Closed files list | Thread search

Editorfile	Project	Projectfile

Start here | Linked_lists_sort_reverse_concatenate.c

```c
void sortList(struct Node *head) {
    if (head == NULL) return;
    struct Node *current = head, *index = NULL;
    int temp;

    while (current != NULL) {
        index = current->next;
        while (index != NULL) {
            if (current->data > index->data) {
                temp = current->data;
                current->data = index->data;
                index->data = temp;
            }
            index = index->next;
        }
        current = current->next;
    }
}

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;
}

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;
}

```

Logs & others

CodeBlocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppCheck/Vera++ messages | Cscope | Debugger | DoxyBlocks | Fortran info | Closed files list | Thread search

Editorfile	Project	Projectfile

```
        temp = temp->next;
    temp->next = head2;
    return head1;
}

int main() {
    struct Node *list1 = NULL, *list2 = NULL;
    int choice, value, list_choice;

    while (1) {
        printf("\nMenu:\n");
        printf("1. Insert into List 1\n");
        printf("2. Insert into List 2\n");
        printf("3. Display Lists\n");
        printf("4. Sort List\n");
        printf("5. Reverse List\n");
        printf("6. Concatenate Lists\n");
        printf("7. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter value to insert in List 1: ");
                scanf("%d", &value);
                insertEnd(&list1, value);
                break;
            case 2:
                printf("Enter value to insert in List 2: ");
                scanf("%d", &value);
                insertEnd(&list2, value);
                break;
            case 3:
                printf("List 1: ");
                display(list1);
                printf("List 2: ");
                display(list2);
                break;
            case 4:
                printf("Sort which list (1 or 2)? ");
                scanf("%d", &list_choice);
```

Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppCheck/Vera++ messages | Cscope | Debugger | DoxyBlocks | Fortran info | Closed files list | Thread search

Project	Projectfile

```
        temp = temp->next;
    temp->next = head2;
    return head1;
}

int main() {
    struct Node *list1 = NULL, *list2 = NULL;
    int choice, value, list_choice;

    while (1) {
        printf("\nMenu:\n");
        printf("1. Insert into List 1\n");
        printf("2. Insert into List 2\n");
        printf("3. Display Lists\n");
        printf("4. Sort List\n");
        printf("5. Reverse List\n");
        printf("6. Concatenate Lists\n");
        printf("7. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter value to insert in List 1: ");
                scanf("%d", &value);
                insertEnd(&list1, value);
                break;
            case 2:
                printf("Enter value to insert in List 2: ");
                scanf("%d", &value);
                insertEnd(&list2, value);
                break;
            case 3:
                printf("List 1: ");
                display(list1);
                printf("List 2: ");
                display(list2);
                break;
            case 4:
                printf("Sort which list (1 or 2)? ");
                scanf("%d", &list_choice);
```

Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppCheck/Vera++ messages | Cscope | Debugger | DoxyBlocks | Fortran info | Closed files list | Thread search

Project	Projectfile

Start here | Linked_lists_sort_reverse_concatenate.c

```
104                 printf("Enter value to insert in List 2: ");
105                 scanf("%d", &value);
106                 insertEnd(&list2, value);
107                 break;
108             case 3:
109                 printf("List 1: ");
110                 display(list1);
111                 printf("List 2: ");
112                 display(list2);
113                 break;
114             case 4:
115                 printf("Sort which list (1 or 2)? ");
116                 scanf("%d", &list_choice);
117                 if (list_choice == 1)
118                     sortList(list1);
119                 else
120                     sortList(list2);
121                 break;
122             case 5:
123                 printf("Reverse which list (1 or 2)? ");
124                 scanf("%d", &list_choice);
125                 if (list_choice == 1)
126                     list1 = reverseList(list1);
127                 else
128                     list2 = reverseList(list2);
129                 break;
130             case 6:
131                 list1 = concatenate(list1, list2);
132                 list2 = NULL;
133                 printf("Lists concatenated. List 1 now contains both lists.\n");
134                 break;
135             case 7:
136                 exit(0);
137             default:
138                 printf("Invalid choice. Try again.\n");
139         }
140     }
141     return 0;
142 }
143
```

Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppCheck/Vera++ messages | Cscope | Debugger | DoxyBlocks | Fortran info | Closed files list | Thread search

Editorfile	Project	Projectfile

C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 128, Col 48, Pos 3571 | Insert | Read/Write | default

```
Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 1
Enter value to insert in List 1: 23

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 2
Enter value to insert in List 2: 65

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 3
List 1: 23
List 2: 65

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 1
Enter value to insert in List 1: 55

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 2
Enter value to insert in List 2: 78

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
```

```
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 2
Enter value to insert in List 2: 78

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 3
List 1: 23 55
List 2: 65 78

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 5
Reverse which list (1 or 2)? 2

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 3
List 1: 23 55
List 2: 78 65

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 6
Lists concatenated. List 1 now contains both lists.

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 1
```

```
C:\Users\Admin\Desktop\Nidhi_1BM24CS185\Linked_lists_sort_reverse_concatenate.exe
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 3
List 1: 23 55
List 2: 78 65

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 6
Lists concatenated. List 1 now contains both lists.

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 1
Enter value to insert in List 1: 23

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice: 3
List 1: 23 55 78 65 23
List 2: List is empty

Menu:
1. Insert into List 1
2. Insert into List 2
3. Display Lists
4. Sort List
5. Reverse List
6. Concatenate Lists
7. Exit
Enter your choice:
```