VARUN BHARGAVA – 241010282
DATA STRUCTURES TASK-4

Task 01: Reverse a Linked List:

(https://github.com/varunnnb/dsa-sem3-iiitnr/blob/main/lab4/lab4-1.c)

Write a program to:

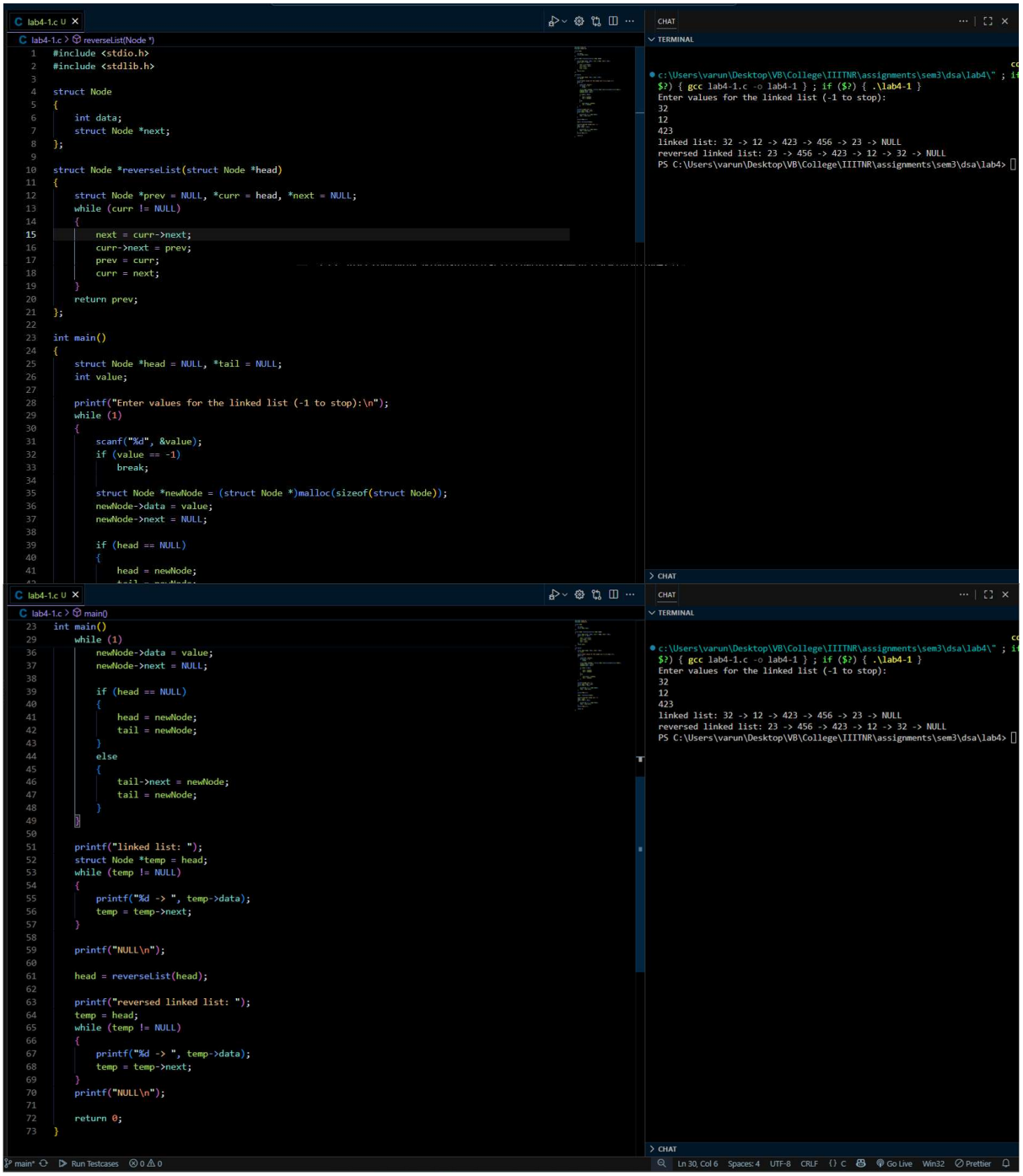
1. Take input from the user to create a singly linked list.

lab4-1.c U

lab4-1.c > main()

```c
#include <stdio.h>
#include <stdlib.h>

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

int main()
{
    struct Node *head = NULL, *tail = NULL;
    int value;

    printf("Enter values for the linked list (-1 to stop):\n");
    while (1)
    {
        scanf("%d", &value);
        if (value == -1)
            break;

        struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
        newNode->data = value;
        newNode->next = NULL;

        if (head == NULL)
        {
            head = newNode;
            tail = newNode;
        }
        else
        {
            tail->next = newNode;
            tail = newNode;
        }
    }

```

2. Reverse the linked list.
3. Display the original and reversed linked list.


lab4-1.c U
lab4-1.c > reverseList(Node *)
#include <stdio.h>
#include <stdlib.h>
struct Node
{
int data;
struct Node *next;
};
struct Node *reverseList(struct Node *head)
{
struct Node *prev = NULL, *curr = head, *next = NULL;
while (curr != NULL)
{
next = curr->next;
curr->next = prev;
prev = curr;
curr = next;
}
return prev;
};
int main()
{
struct Node *head = NULL, *tail = NULL;
int value;
printf("Enter values for the linked list (-1 to stop):\n");
while (1)
{
scanf("%d", &value);
if (value == -1)
break;
struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
newNode->data = value;
newNode->next = NULL;
if (head == NULL)
{
head = newNode;
CHAT
TERMINAL
c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4\" ; if
$?) { gcc lab4-1.c -o lab4-1 } ; if ($?) { .\lab4-1 }
Enter values for the linked list (-1 to stop):
32
12
423
linked list: 32 -> 12 -> 423 -> 456 -> 23 -> NULL
reversed linked list: 23 -> 456 -> 423 -> 12 -> 32 -> NULL
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4>
CHAT
lab4-1.c U
lab4-1.c > main()
int main()
while (1)
newNode->data = value;
newNode->next = NULL;
if (head == NULL)
{
head = newNode;
tail = newNode;
}
else
{
tail->next = newNode;
tail = newNode;
}
}
printf("linked list: ");
struct Node *temp = head;
while (temp != NULL)
{
printf("%d -> ", temp->data);
temp = temp->next;
}
printf("NULL\n");
head = reverseList(head);
printf("reversed linked list: ");
temp = head;
while (temp != NULL)
{
printf("%d -> ", temp->data);
temp = temp->next;
}
printf("NULL\n");
return 0;
}
CHAT
TERMINAL
c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4\" ; if
$?) { gcc lab4-1.c -o lab4-1 } ; if ($?) { .\lab4-1 }
Enter values for the linked list (-1 to stop):
32
12
423
linked list: 32 -> 12 -> 423 -> 456 -> 23 -> NULL
reversed linked list: 23 -> 456 -> 423 -> 12 -> 32 -> NULL
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4>
CHAT
main* Run Testcases 0 0
Ln 30, Col 6 Spaces: 4 UTF-8 CRLF C Go Live Win32 Prettier

Task 02: Check if Linked List is a Palindrome:

(https://github.com/varunnnb/dsa-sem3-iiitnr/blob/main/lab4/lab4-2.c)

Write a program to:

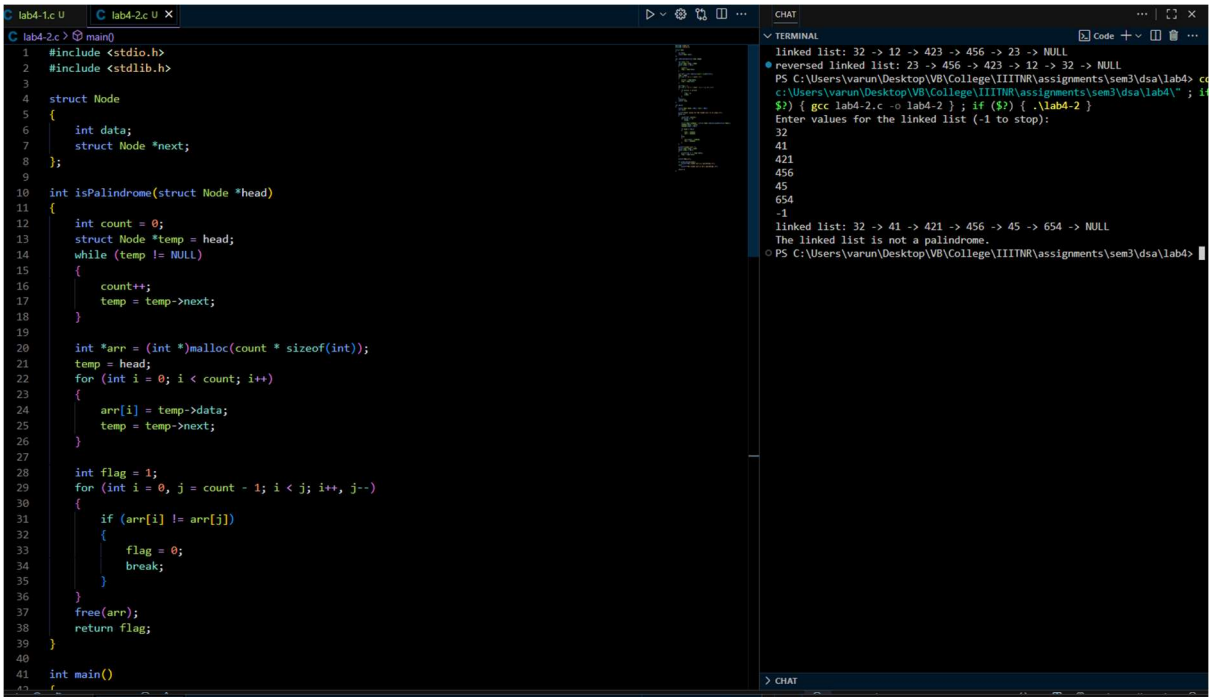
1. Take user input to create a singly linked list.

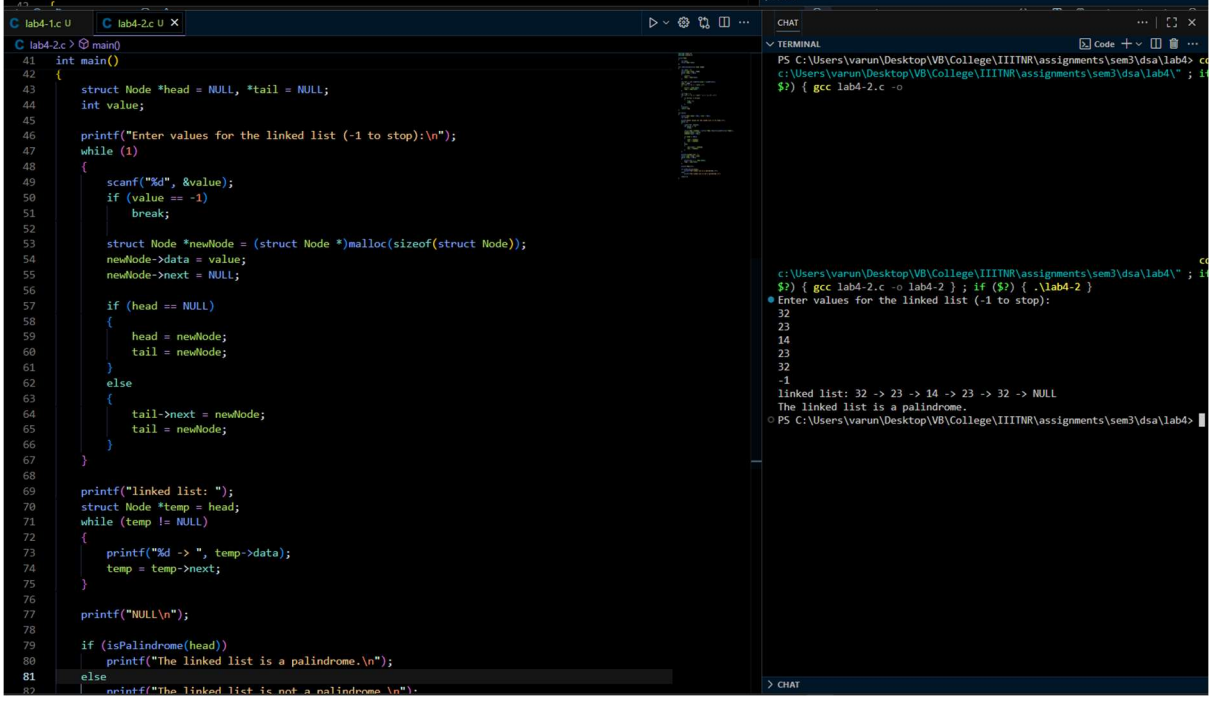
lab4-1.c U lab4-2.c U

lab4-2.c > ...

```c
#include <stdio.h>
#include <stdlib.h>

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

int main()
{
    struct Node *head = NULL, *tail = NULL;
    int value;

    printf("Enter values for the linked list (-1 to stop):\n");
    while (1)
    {
        scanf("%d", &value);
        if (value == -1)
            break;

        struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
        newNode->data = value;
        newNode->next = NULL;

        if (head == NULL)
        {
            head = newNode;
            tail = newNode;
        }
        else
        {
            tail->next = newNode;
            tail = newNode;
        }
    }

```

2. Determine if the linked list is a palindrome (reads the same forward and backward).
3. Display an appropriate message indicating whether the linked list is a palindrome or not.


lab4-1.c U
lab4-2.c U
lab4-2.c > main()
#include <stdio.h>
#include <stdlib.h>
struct Node
{
int data;
struct Node *next;
};
int isPalindrome(struct Node *head)
{
int count = 0;
struct Node *temp = head;
while (temp != NULL)
{
count++;
temp = temp->next;
}
int *arr = (int *)malloc(count * sizeof(int));
temp = head;
for (int i = 0; i < count; i++)
{
arr[i] = temp->data;
temp = temp->next;
}
int flag = 1;
for (int i = 0, j = count - 1; i < j; i++, j--)
{
if (arr[i] != arr[j])
{
flag = 0;
break;
}
}
free(arr);
return flag;
}
int main()
CHAT
TERMINAL
Code
linked list: 32 -> 12 -> 423 -> 456 -> 23 -> NULL
reversed linked list: 23 -> 456 -> 423 -> 12 -> 32 -> NULL
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4> cd
c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4\" ; if
$?) { gcc lab4-2.c -o lab4-2 } ; if ($?) { .\lab4-2 }
Enter values for the linked list (-1 to stop):
32
41
421
456
45
654
-1
linked list: 32 -> 41 -> 421 -> 456 -> 45 -> 654 -> NULL
The linked list is not a palindrome.
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4>
CHAT

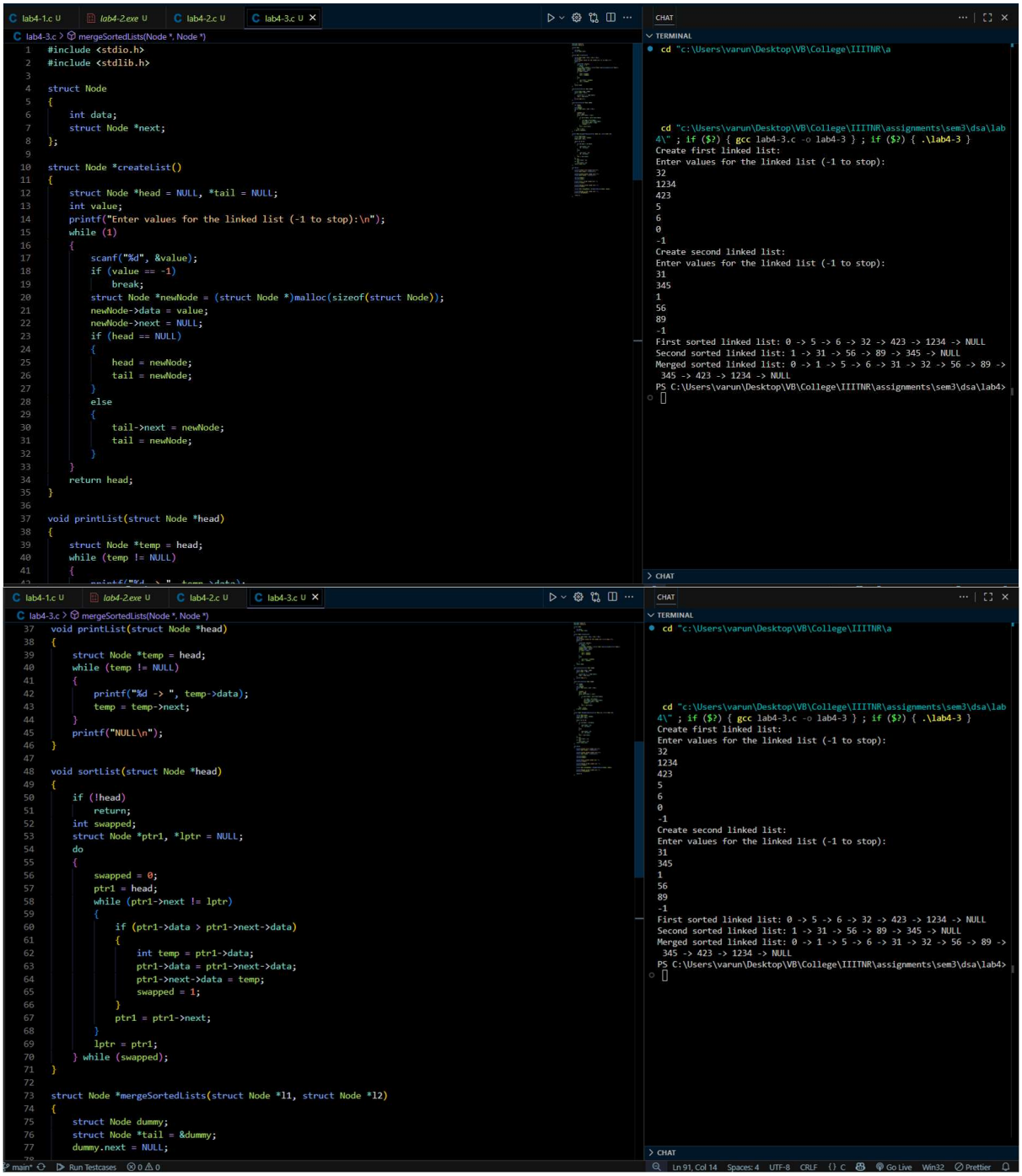

lab4-1.c U
lab4-2.c U
lab4-2.c > main()
int main()
{
struct Node *head = NULL, *tail = NULL;
int value;
printf("Enter values for the linked list (-1 to stop):\n");
while (1)
{
scanf("%d", &value);
if (value == -1)
break;
struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
newNode->data = value;
newNode->next = NULL;
if (head == NULL)
{
head = newNode;
tail = newNode;
}
else
{
tail->next = newNode;
tail = newNode;
}
}
printf("linked list: ");
struct Node *temp = head;
while (temp != NULL)
{
printf("%d -> ", temp->data);
temp = temp->next;
}
printf("NULL\n");
if (isPalindrome(head))
printf("The linked list is a palindrome.\n");
else
CHAT
TERMINAL
Code
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4> cd
c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4\" ; if
$?) { gcc lab4-2.c -o
c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4\" ; if
$?) { gcc lab4-2.c -o lab4-2 } ; if ($?) { .\lab4-2 }
Enter values for the linked list (-1 to stop):
32
23
14
23
32
-1
linked list: 32 -> 23 -> 14 -> 23 -> 32 -> NULL
The linked list is a palindrome.
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4>
CHAT

Task 03: Merge Two Linked Lists:

(https://github.com/varunnnb/dsa-sem3-iiitnr/blob/main/lab4/lab4-3.c)

Write a program to:

1. Take user input to create two unsorted singly linked lists.
2. Merge the two linked lists in ascendinpg order into a single sorted linked list.
3. Display the final merged sorted linked list.



```c
#include <stdio.h>
#include <stdlib.h>

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

struct Node *createList()
{
    struct Node *head = NULL, *tail = NULL;
    int value;
    printf("Enter values for the linked list (-1 to stop):\n");
    while (1)
    {
        scanf("%d", &value);
        if (value == -1)
            break;
        struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
        newNode->data = value;
        newNode->next = NULL;
        if (head == NULL)
        {
            head = newNode;
            tail = newNode;
        }
        else
        {
            tail->next = newNode;
            tail = newNode;
        }
    }
    return head;
}

void printList(struct Node *head)
{
    struct Node *temp = head;
    while (temp != NULL)
    {
```

```
cd "c:\Users\varun\Desktop\VB\College\IIITNR\a

 cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4\" ; if ($?) { gcc lab4-3.c -o lab4-3 } ; if ($?) { .\lab4-3 }
Create first linked list:
Enter values for the linked list (-1 to stop):
32
1234
423
5
6
0
-1
Create second linked list:
Enter values for the linked list (-1 to stop):
31
345
1
56
89
-1
First sorted linked list: 0 -> 5 -> 6 -> 32 -> 423 -> 1234 -> NULL
Second sorted linked list: 1 -> 31 -> 56 -> 89 -> 345 -> NULL
Merged sorted linked list: 0 -> 1 -> 5 -> 6 -> 31 -> 32 -> 56 -> 89 ->
 345 -> 423 -> 1234 -> NULL
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4>
```

```c
void printList(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)
{
    if (!head)
        return;
    int swapped;
    struct Node *ptr1, *lptr = NULL;
    do
    {
        swapped = 0;
        ptr1 = head;
        while (ptr1->next != lptr)
        {
            if (ptr1->data > ptr1->next->data)
            {
                int temp = ptr1->data;
                ptr1->data = ptr1->next->data;
                ptr1->next->data = temp;
                swapped = 1;
            }
            ptr1 = ptr1->next;
        }
        lptr = ptr1;
    } while (swapped);
}

struct Node *mergeSortedLists(struct Node *l1, struct Node *l2)
{
    struct Node dummy;
    struct Node *tail = &dummy;
    dummy.next = NULL;
```

```
cd "c:\Users\varun\Desktop\VB\College\IIITNR\a

 cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4\" ; if ($?) { gcc lab4-3.c -o lab4-3 } ; if ($?) { .\lab4-3 }
Create first linked list:
Enter values for the linked list (-1 to stop):
32
1234
423
5
6
0
-1
Create second linked list:
Enter values for the linked list (-1 to stop):
31
345
1
56
89
-1
First sorted linked list: 0 -> 5 -> 6 -> 32 -> 423 -> 1234 -> NULL
Second sorted linked list: 1 -> 31 -> 56 -> 89 -> 345 -> NULL
Merged sorted linked list: 0 -> 1 -> 5 -> 6 -> 31 -> 32 -> 56 -> 89 ->
 345 -> 423 -> 1234 -> NULL
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4>
```


lab4-1.c U
lab4-2.exe U
lab4-2.c U
lab4-3.c U
lab4-3.c > mergeSortedLists(Node *, Node *)
struct Node *mergeSortedLists(struct Node *l1, struct Node *l2)
{
struct Node dummy;
struct Node *tail = &dummy;
dummy.next = NULL;
while (l1 && l2)
{
if (l1->data < l2->data)
{
tail->next = l1;
l1 = l1->next;
}
else
{
tail->next = l2;
l2 = l2->next;
}
tail = tail->next;
}
if (l1)
tail->next = l1;
if (l2)
tail->next = l2;
return dummy.next;
}
int main()
{
printf("Create first linked list:\n");
struct Node *head1 = createList();
printf("Create second linked list:\n");
struct Node *head2 = createList();
sortList(head1);
sortList(head2);
printf("First sorted linked list: ");
printList(head1);
CHAT
TERMINAL
cd "c:\Users\varun\Desktop\VB\College\IIITNR\a
cd "c:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4\" ; if ($?) { gcc lab4-3.c -o lab4-3 } ; if ($?) { .\lab4-3 }
Create first linked list:
Enter values for the linked list (-1 to stop):
32
1234
423
5
6
0
-1
Create second linked list:
Enter values for the linked list (-1 to stop):
31
345
1
56
89
-1
First sorted linked list: 0 -> 5 -> 6 -> 32 -> 423 -> 1234 -> NULL
Second sorted linked list: 1 -> 31 -> 56 -> 89 -> 345 -> NULL
Merged sorted linked list: 0 -> 1 -> 5 -> 6 -> 31 -> 32 -> 56 -> 89 -> 345 -> 423 -> 1234 -> NULL
PS C:\Users\varun\Desktop\VB\College\IIITNR\assignments\sem3\dsa\lab4>
main*
Run Testcases
Ln 91, Col 14
Spaces: 4
UTF-8
CRLF
C
Go Live
Win32
Prettier
void sortList(struct Node *head)
Generate Copilot summary