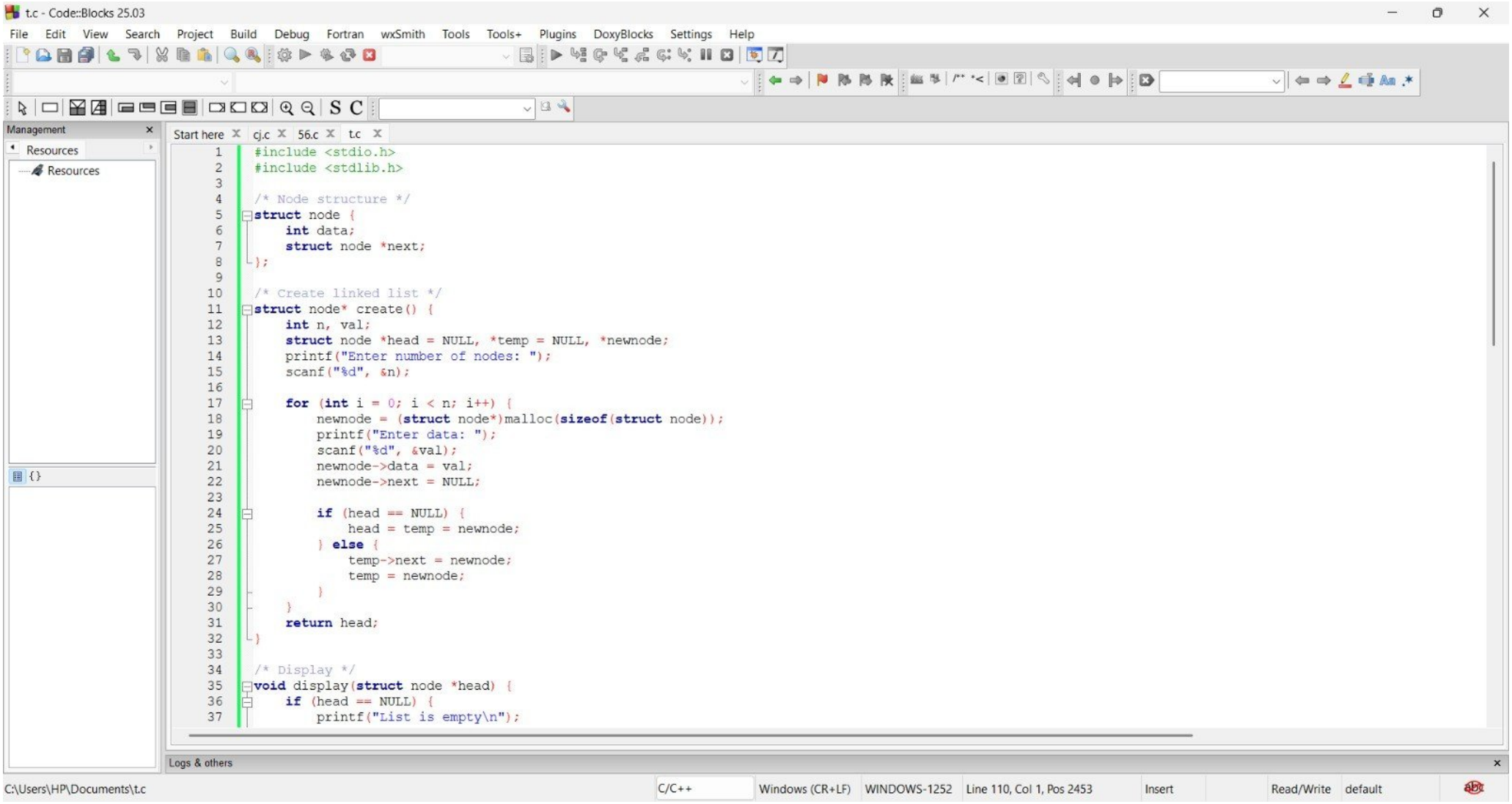
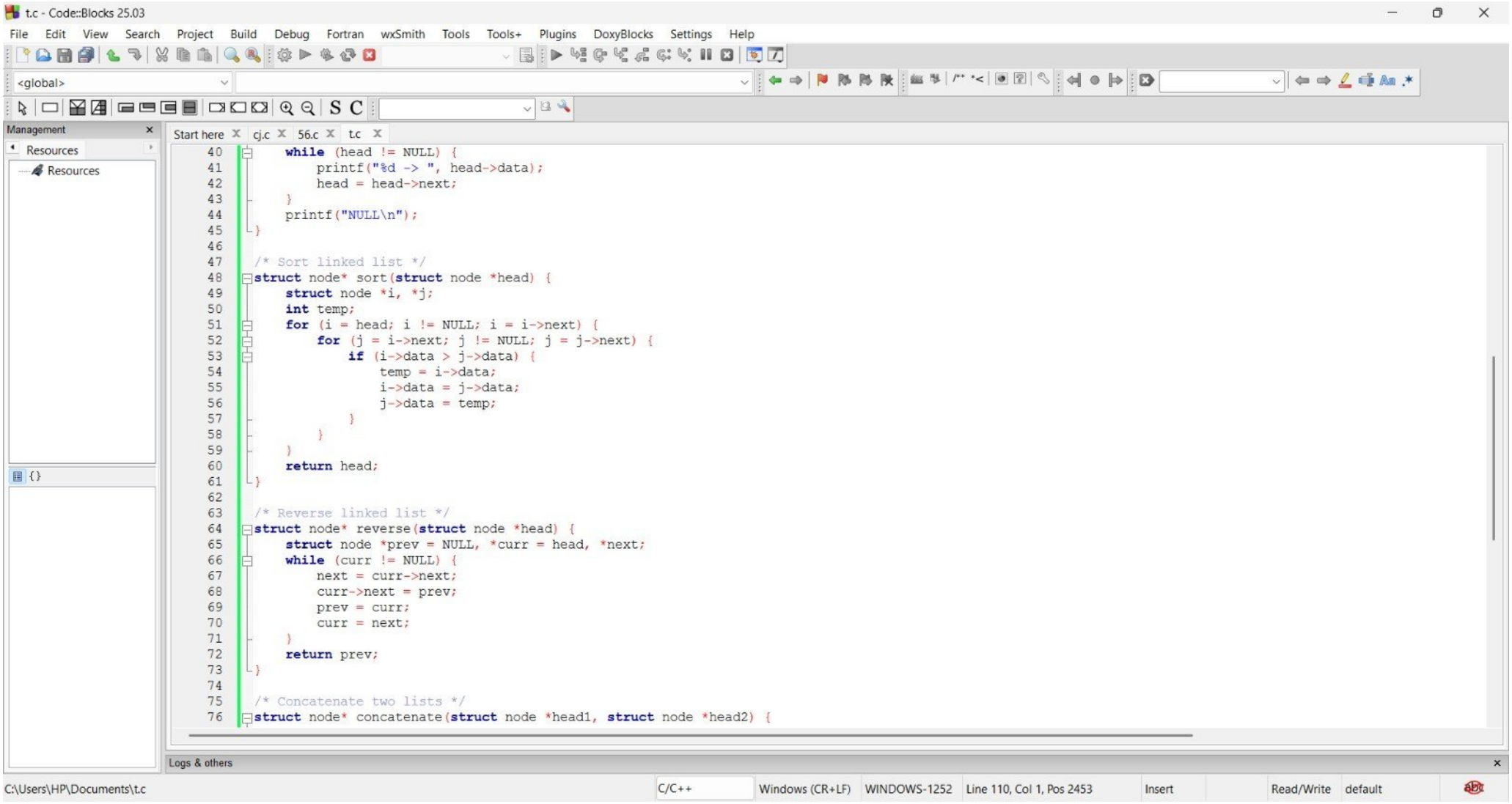
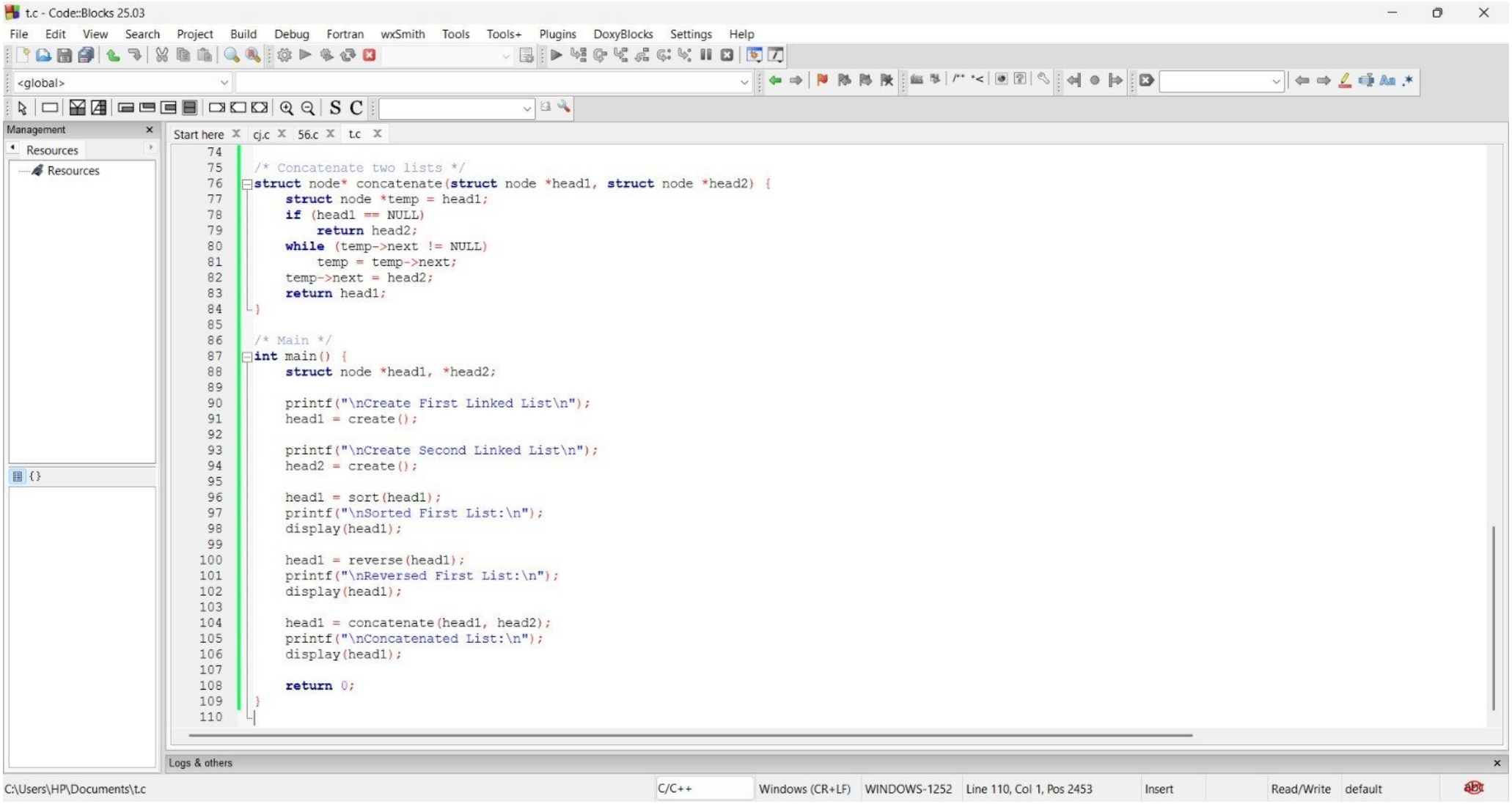
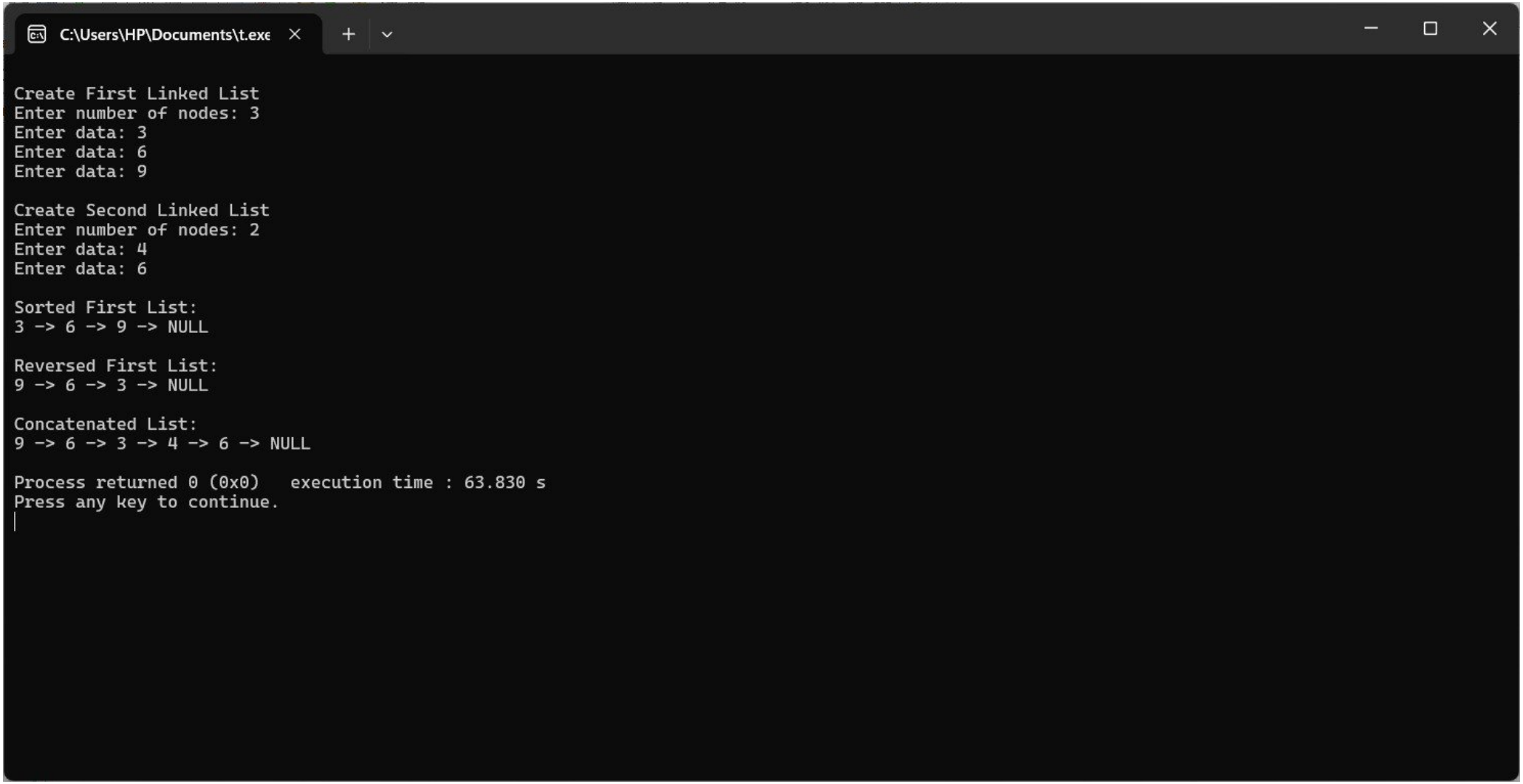
4	2	5	WAP to Implement Singly Linked List with following operations a) Create a linked list. b) Insertion of a node at first position, at any position and at end of list. Display the contents of the linked list.
		5	Program - Leetcode platform


t.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Resources
Resources
Start here
cj.c
56.c
t.c
#include <stdio.h>
#include <stdlib.h>
/* Node structure */
struct node {
int data;
struct node *next;
};
/* Create linked list */
struct node* create() {
int n, val;
struct node *head = NULL, *temp = NULL, *newnode;
printf("Enter number of nodes: ");
scanf("%d", &n);
for (int i = 0; i < n; i++) {
newnode = (struct node*)malloc(sizeof(struct node));
printf("Enter data: ");
scanf("%d", &val);
newnode->data = val;
newnode->next = NULL;
if (head == NULL) {
head = temp = newnode;
} else {
temp->next = newnode;
temp = newnode;
}
}
return head;
}
/* Display */
void display(struct node *head) {
if (head == NULL) {
printf("List is empty\n");
Logs & others
C:\Users\HP\Documents\t.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 110, Col 1, Pos 2453
Insert
Read/Write
default


t.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Management
Resources
Resources
Start here
cj.c
56.c
t.c
40 while (head != NULL) {
41 printf("%d -> ", head->data);
42 head = head->next;
43 }
44 printf("NULL\n");
45 }
46
47 /* Sort linked list */
48 struct node* sort(struct node *head) {
49 struct node *i, *j;
50 int temp;
51 for (i = head; i != NULL; i = i->next) {
52 for (j = i->next; j != NULL; j = j->next) {
53 if (i->data > j->data) {
54 temp = i->data;
55 i->data = j->data;
56 j->data = temp;
57 }
58 }
59 }
60 return head;
61 }
62
63 /* Reverse linked list */
64 struct node* reverse(struct node *head) {
65 struct node *prev = NULL, *curr = head, *next;
66 while (curr != NULL) {
67 next = curr->next;
68 curr->next = prev;
69 prev = curr;
70 curr = next;
71 }
72 return prev;
73 }
74
75 /* Concatenate two lists */
76 struct node* concatenate(struct node *head1, struct node *head2) {
Logs & others
C:\Users\HP\Documents\t.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 110, Col 1, Pos 2453
Insert
Read/Write
default


t.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Management
Resources
Resources
Start here
cj.c
56.c
t.c
/* Concatenate two lists */
struct node* concatenate(struct node *head1, struct node *head2) {
struct node *temp = head1;
if (head1 == NULL)
return head2;
while (temp->next != NULL)
temp = temp->next;
temp->next = head2;
return head1;
}
/* Main */
int main() {
struct node *head1, *head2;
printf("\nCreate First Linked List\n");
head1 = create();
printf("\nCreate Second Linked List\n");
head2 = create();
head1 = sort(head1);
printf("\nSorted First List:\n");
display(head1);
head1 = reverse(head1);
printf("\nReversed First List:\n");
display(head1);
head1 = concatenate(head1, head2);
printf("\nConcatenated List:\n");
display(head1);
return 0;
}
Logs & others
C:\Users\HP\Documents\t.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 110, Col 1, Pos 2453
Insert
Read/Write
default


C:\Users\HP\Documents\t.exe
Create First Linked List
Enter number of nodes: 3
Enter data: 3
Enter data: 6
Enter data: 9
Create Second Linked List
Enter number of nodes: 2
Enter data: 4
Enter data: 6
Sorted First List:
3 -> 6 -> 9 -> NULL
Reversed First List:
9 -> 6 -> 3 -> NULL
Concatenated List:
9 -> 6 -> 3 -> 4 -> 6 -> NULL
Process returned 0 (0x0) execution time : 63.830 s
Press any key to continue.