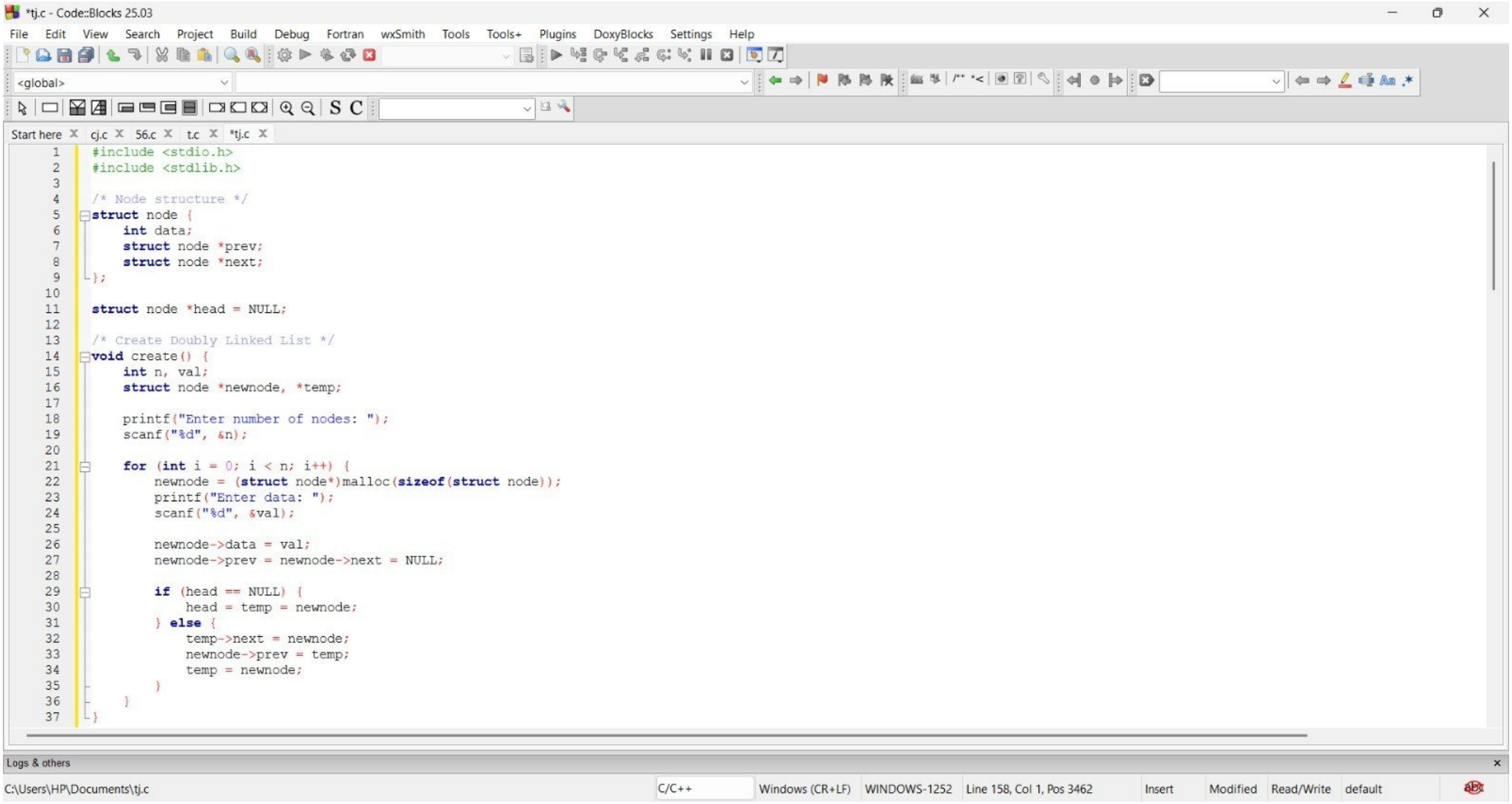
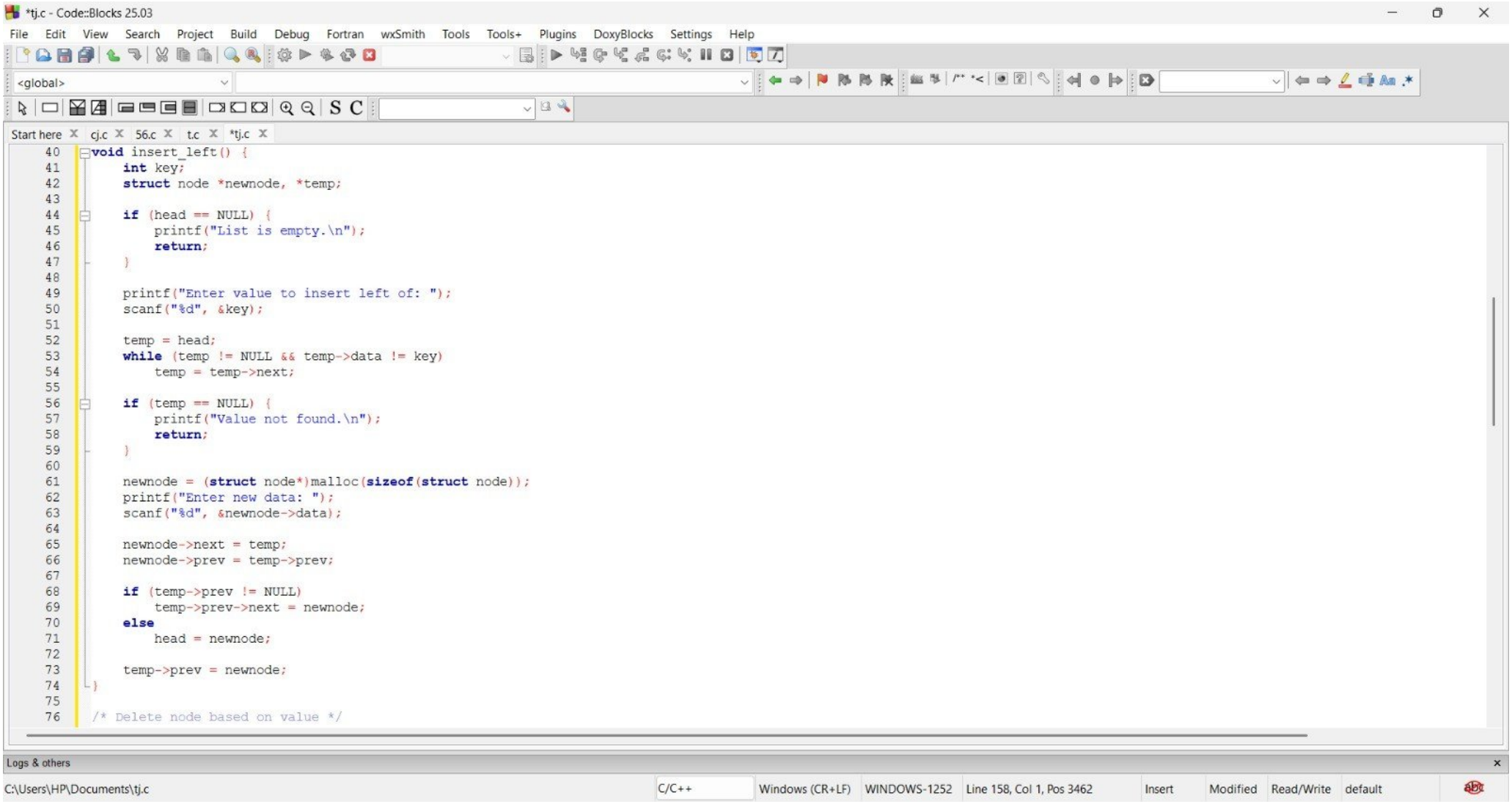
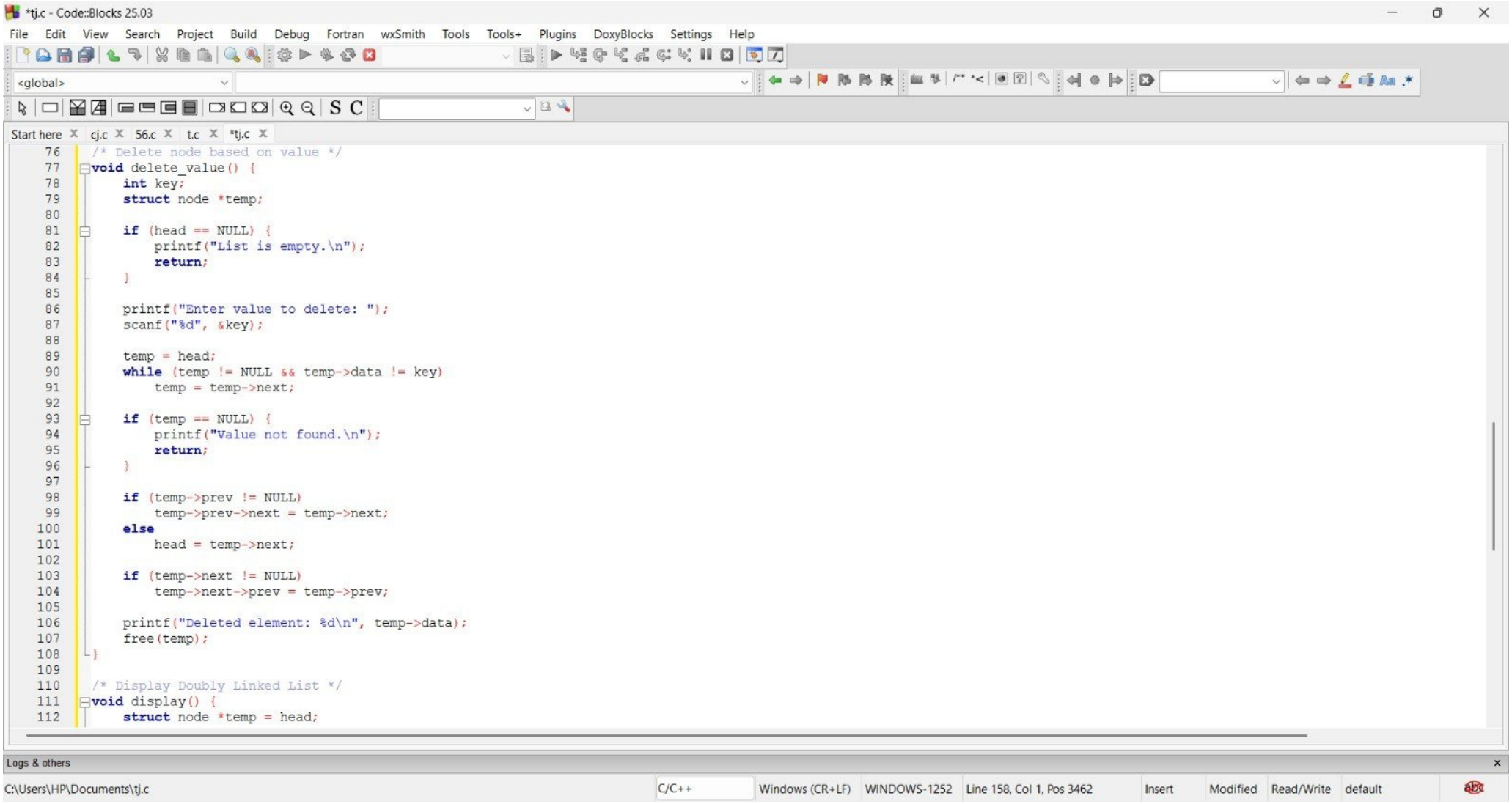
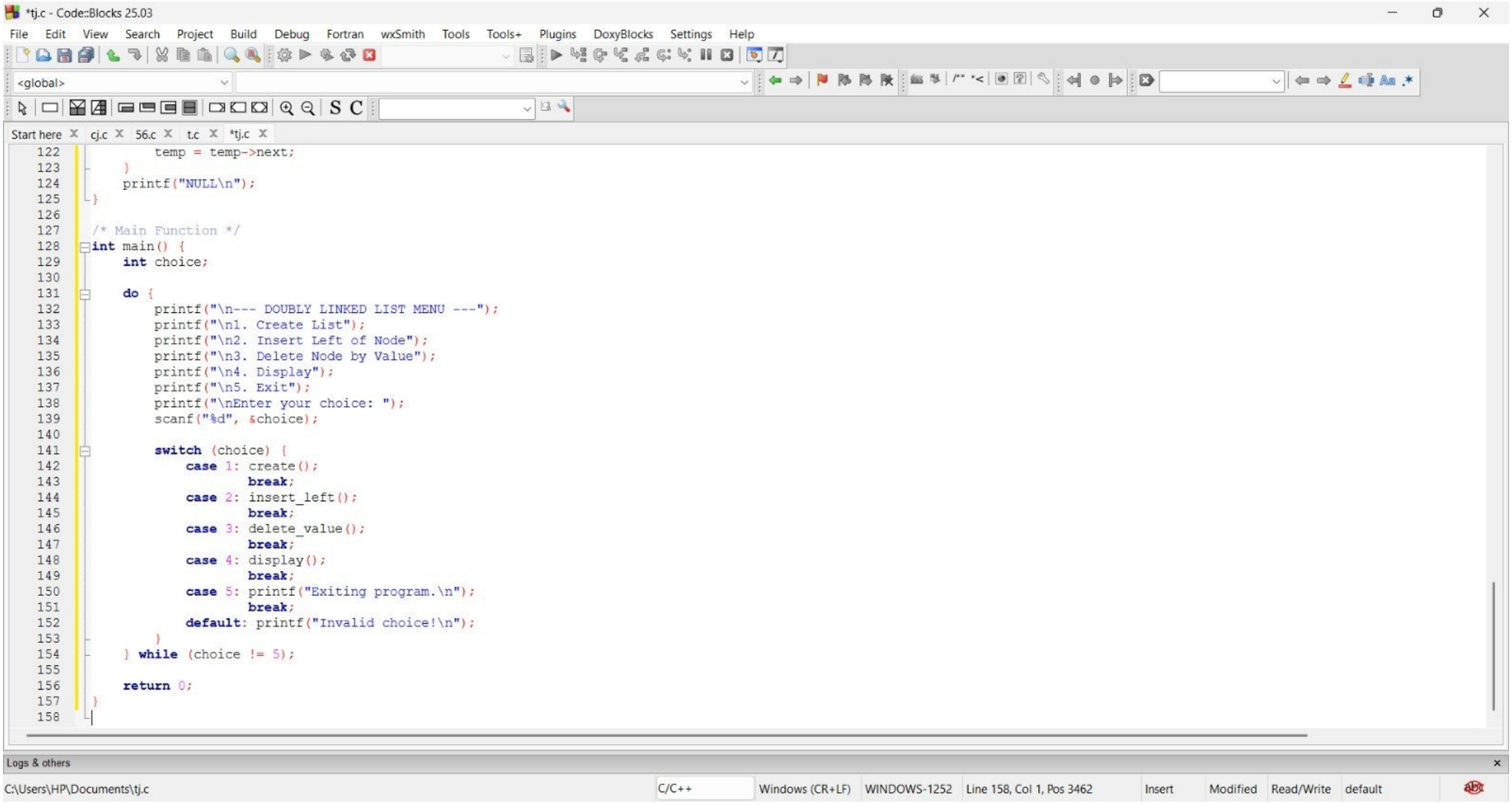
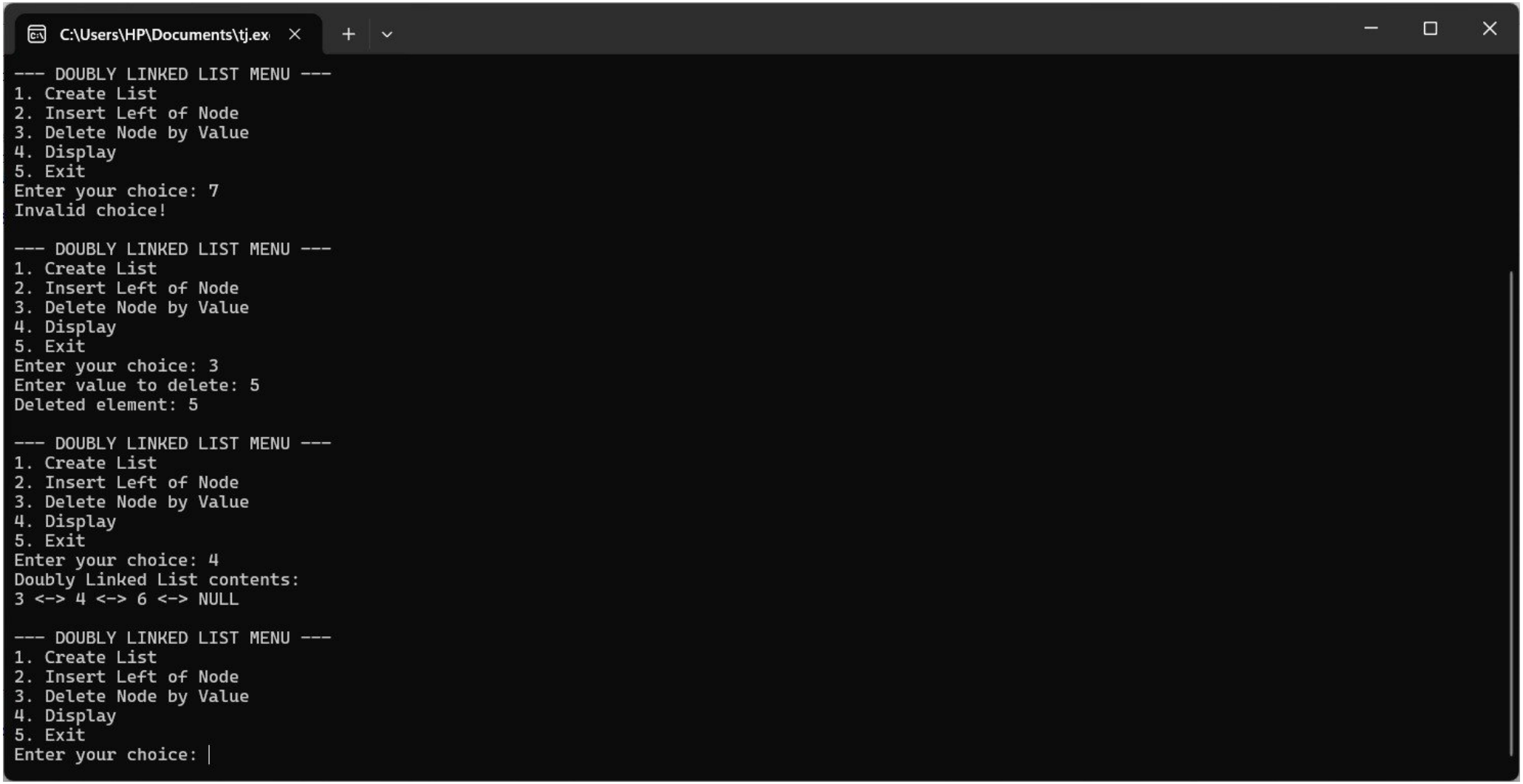
6	3	5	a) WAP to Implement Single Link List with following operations: Sort the linked list, Reverse the linked list, Concatenation of two linked lists.
		5	b) WAP to Implement Single Link List to simulate Stack & Queue Operations.

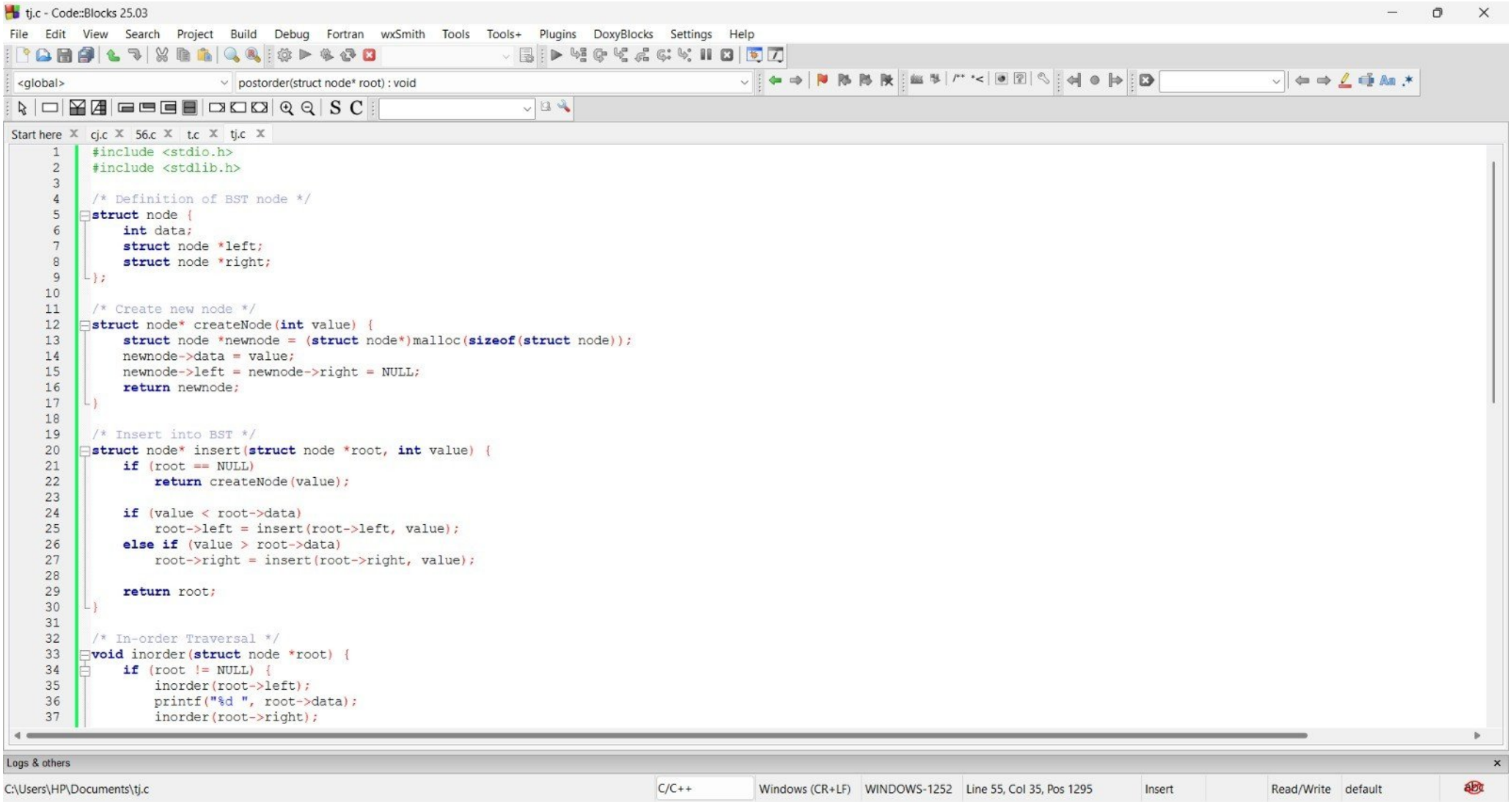

*tj.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here
cj.c
56.c
t.c
*tj.c
#include <stdio.h>
#include <stdlib.h>
/* Node structure */
struct node {
int data;
struct node *prev;
struct node *next;
};
struct node *head = NULL;
/* Create Doubly Linked List */
void create() {
int n, val;
struct node *newnode, *temp;
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->prev = newnode->next = NULL;
if (head == NULL) {
head = temp = newnode;
} else {
temp->next = newnode;
newnode->prev = temp;
temp = newnode;
}
}
}
Logs & others
C:\Users\HP\Documents\tj.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 158, Col 1, Pos 3462
Insert
Modified
Read/Write
default

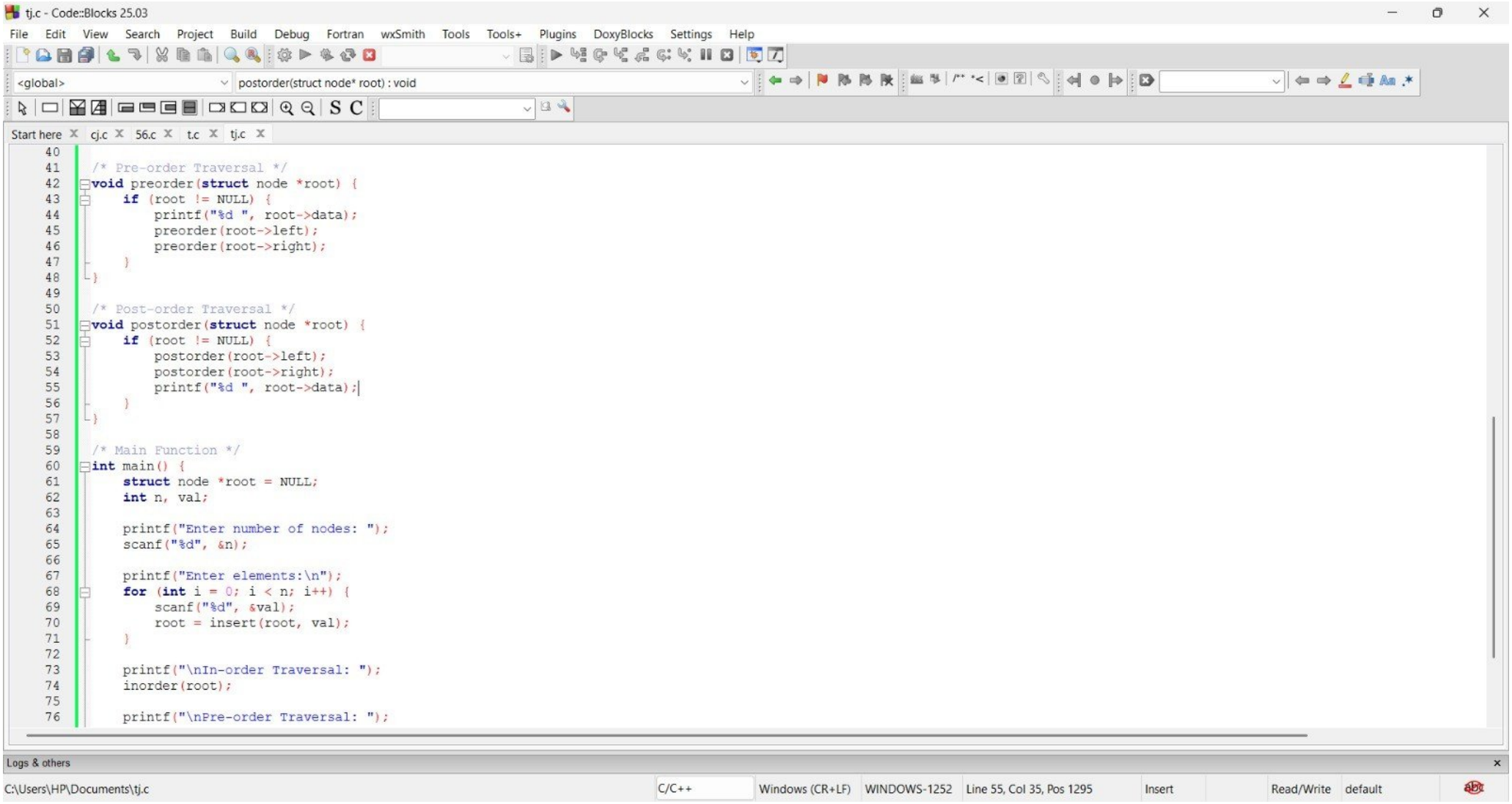

*tj.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here cj.c 56.c t.c *tj.c
40 void insert_left() {
41 int key;
42 struct node *newnode, *temp;
43
44 if (head == NULL) {
45 printf("List is empty.\n");
46 return;
47 }
48
49 printf("Enter value to insert left of: ");
50 scanf("%d", &key);
51
52 temp = head;
53 while (temp != NULL && temp->data != key)
54 temp = temp->next;
55
56 if (temp == NULL) {
57 printf("Value not found.\n");
58 return;
59 }
60
61 newnode = (struct node*)malloc(sizeof(struct node));
62 printf("Enter new data: ");
63 scanf("%d", &newnode->data);
64
65 newnode->next = temp;
66 newnode->prev = temp->prev;
67
68 if (temp->prev != NULL)
69 temp->prev->next = newnode;
70 else
71 head = newnode;
72
73 temp->prev = newnode;
74 }
75
76 /* Delete node based on value */
Logs & others
C:\Users\HP\Documents\tj.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 158, Col 1, Pos 3462
Insert
Modified
Read/Write
default

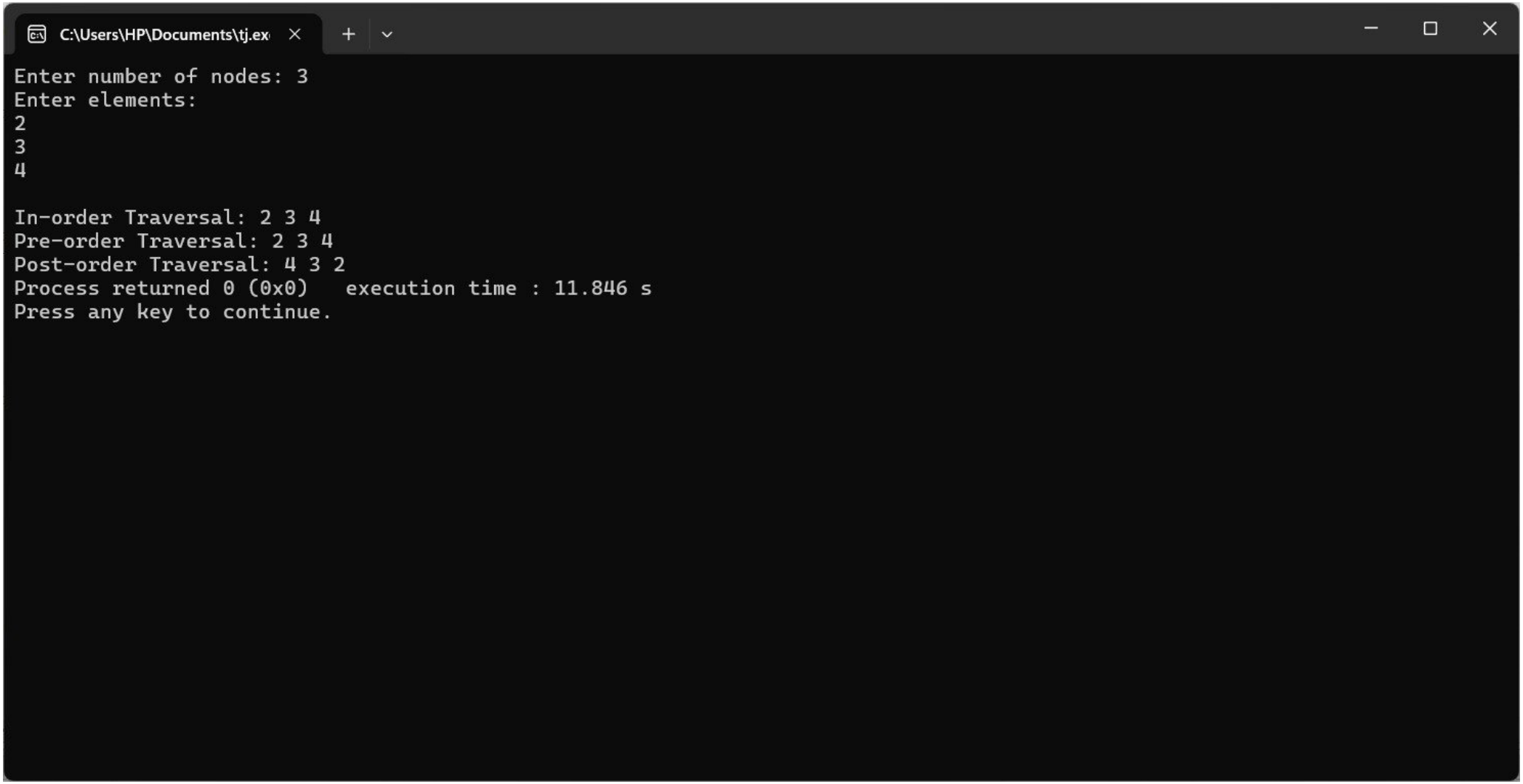

*tj.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here cj.c 56.c t.c *tj.c
/* Delete node based on value */
void delete_value() {
int key;
struct node *temp;
if (head == NULL) {
printf("List is empty.\n");
return;
}
printf("Enter value to delete: ");
scanf("%d", &key);
temp = head;
while (temp != NULL && temp->data != key)
temp = temp->next;
if (temp == NULL) {
printf("Value not found.\n");
return;
}
if (temp->prev != NULL)
temp->prev->next = temp->next;
else
head = temp->next;
if (temp->next != NULL)
temp->next->prev = temp->prev;
printf("Deleted element: %d\n", temp->data);
free(temp);
}
/* Display Doubly Linked List */
void display() {
struct node *temp = head;
Logs & others
C:\Users\HP\Documents\tj.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 158, Col 1, Pos 3462
Insert
Modified
Read/Write
default


*tj.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Start here
cj.c
56.c
t.c
*tj.c
temp = temp->next;
}
printf("NULL\n");
}
/* Main Function */
int main() {
int choice;
do {
printf("\n--- DOUBLY LINKED LIST MENU ---");
printf("\n1. Create List");
printf("\n2. Insert Left of Node");
printf("\n3. Delete Node by Value");
printf("\n4. Display");
printf("\n5. Exit");
printf("\nEnter your choice: ");
scanf("%d", &choice);
switch (choice) {
case 1: create();
break;
case 2: insert_left();
break;
case 3: delete_value();
break;
case 4: display();
break;
case 5: printf("Exiting program.\n");
break;
default: printf("Invalid choice!\n");
}
} while (choice != 5);
return 0;
}
Logs & others
C:\Users\HP\Documents\tj.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 158, Col 1, Pos 3462
Insert
Modified
Read/Write
default


C:\Users\HP\Documents\tj.ex
--- DOUBLY LINKED LIST MENU ---
1. Create List
2. Insert Left of Node
3. Delete Node by Value
4. Display
5. Exit
Enter your choice: 7
Invalid choice!
--- DOUBLY LINKED LIST MENU ---
1. Create List
2. Insert Left of Node
3. Delete Node by Value
4. Display
5. Exit
Enter your choice: 3
Enter value to delete: 5
Deleted element: 5
--- DOUBLY LINKED LIST MENU ---
1. Create List
2. Insert Left of Node
3. Delete Node by Value
4. Display
5. Exit
Enter your choice: 4
Doubly Linked List contents:
3 <-> 4 <-> 6 <-> NULL
--- DOUBLY LINKED LIST MENU ---
1. Create List
2. Insert Left of Node
3. Delete Node by Value
4. Display
5. Exit
Enter your choice:


tj.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
postorder(struct node* root) : void
Start here
cj.c
56.c
t.c
tj.c
#include <stdio.h>
#include <stdlib.h>
/* Definition of BST node */
struct node {
int data;
struct node *left;
struct node *right;
};
/* Create new node */
struct node* createNode(int value) {
struct node *newnode = (struct node*)malloc(sizeof(struct node));
newnode->data = value;
newnode->left = newnode->right = NULL;
return newnode;
}
/* Insert into BST */
struct node* insert(struct node *root, int value) {
if (root == NULL)
return createNode(value);
if (value < root->data)
root->left = insert(root->left, value);
else if (value > root->data)
root->right = insert(root->right, value);
return root;
}
/* In-order Traversal */
void inorder(struct node *root) {
if (root != NULL) {
inorder(root->left);
printf("%d ", root->data);
inorder(root->right);
Logs & others
C:\Users\HP\Documents\tj.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 55, Col 35, Pos 1295
Insert
Read/Write
default


tj.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
postorder(struct node* root) : void
Start here
cj.c
56.c
t.c
tj.c
/* Pre-order Traversal */
void preorder(struct node *root) {
if (root != NULL) {
printf("%d ", root->data);
preorder(root->left);
preorder(root->right);
}
}
/* Post-order Traversal */
void postorder(struct node *root) {
if (root != NULL) {
postorder(root->left);
postorder(root->right);
printf("%d ", root->data);
}
}
/* Main Function */
int main() {
struct node *root = NULL;
int n, val;
printf("Enter number of nodes: ");
scanf("%d", &n);
printf("Enter elements:\n");
for (int i = 0; i < n; i++) {
scanf("%d", &val);
root = insert(root, val);
}
printf("\nIn-order Traversal: ");
inorder(root);
printf("\nPre-order Traversal: ");
Logs & others
C:\Users\HP\Documents\tj.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 55, Col 35, Pos 1295
Insert
Read/Write
default


C:\Users\HP\Documents\tj.ex
Enter number of nodes: 3
Enter elements:
2
3
4
In-order Traversal: 2 3 4
Pre-order Traversal: 2 3 4
Post-order Traversal: 4 3 2
Process returned 0 (0x0)   execution time : 11.846 s
Press any key to continue.