stack.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

singly linked list

Management

Projects Files FSymbols

Workspace

Start here | stack.c

```
36  rear->next = newnode;
37  rear = newnode;
38  }
39  printf("Enqueued %d into queue\n", value);
40  }
41  void dequeue() {
42  struct node *temp;
43  if (front == NULL) {
44  printf("Queue is empty\n");
45  } else {
46  temp = front;
47  printf("Dequeued element: %d\n", front->data);
48  front = front->next;
49  free(temp);
50  }
51  }
52  int main() {
53  int choice, value;
54  while (1) {
55  printf("\n1. Push (Stack)");
56  printf("\n2. Pop (Stack)");
57  printf("\n3. Enqueue (Queue)");
58  printf("\n4. Dequeue (Queue)");
59  printf("\n5. Exit");
60  printf("\nEnter choice: ");
61  scanf("%d", &choice);
62  switch (choice) {
63  case 1:
64  printf("Enter value: ");
65  scanf("%d", &value);
66  push(value);
67  break;
68  case 2:
69  pop();
70  break;
71  case 3:
72  printf("Enter value: ");
73  scanf("%d", &value);
74  enqueue(value);
75  break;
76  case 4:
77  dequeue();
78  break;
79  case 5:
80  exit(0);
81  default:
82  printf("Invalid choice\n");
83  }
84  }
85  }
86
```

stack.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

singly linked list

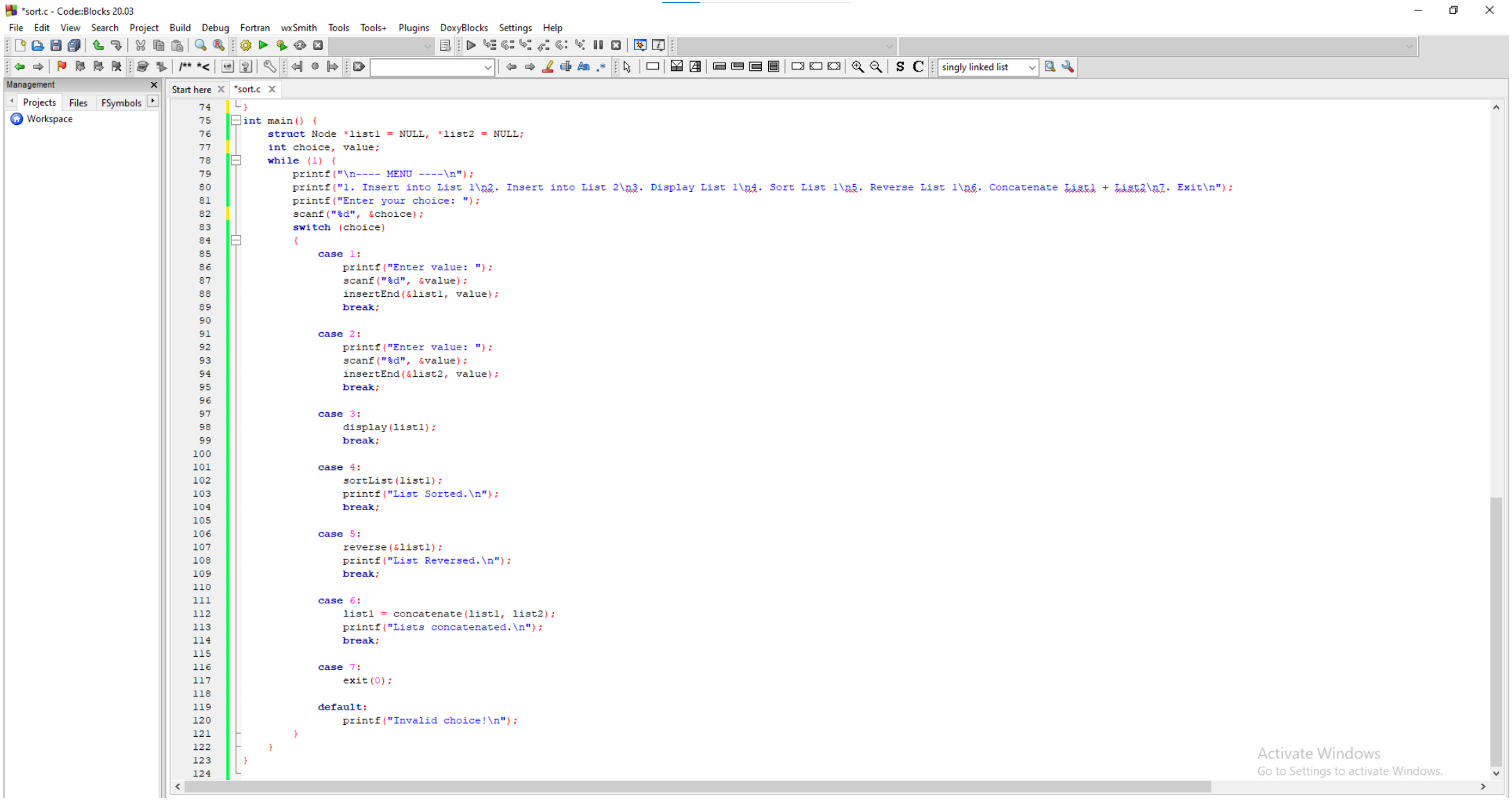
Management

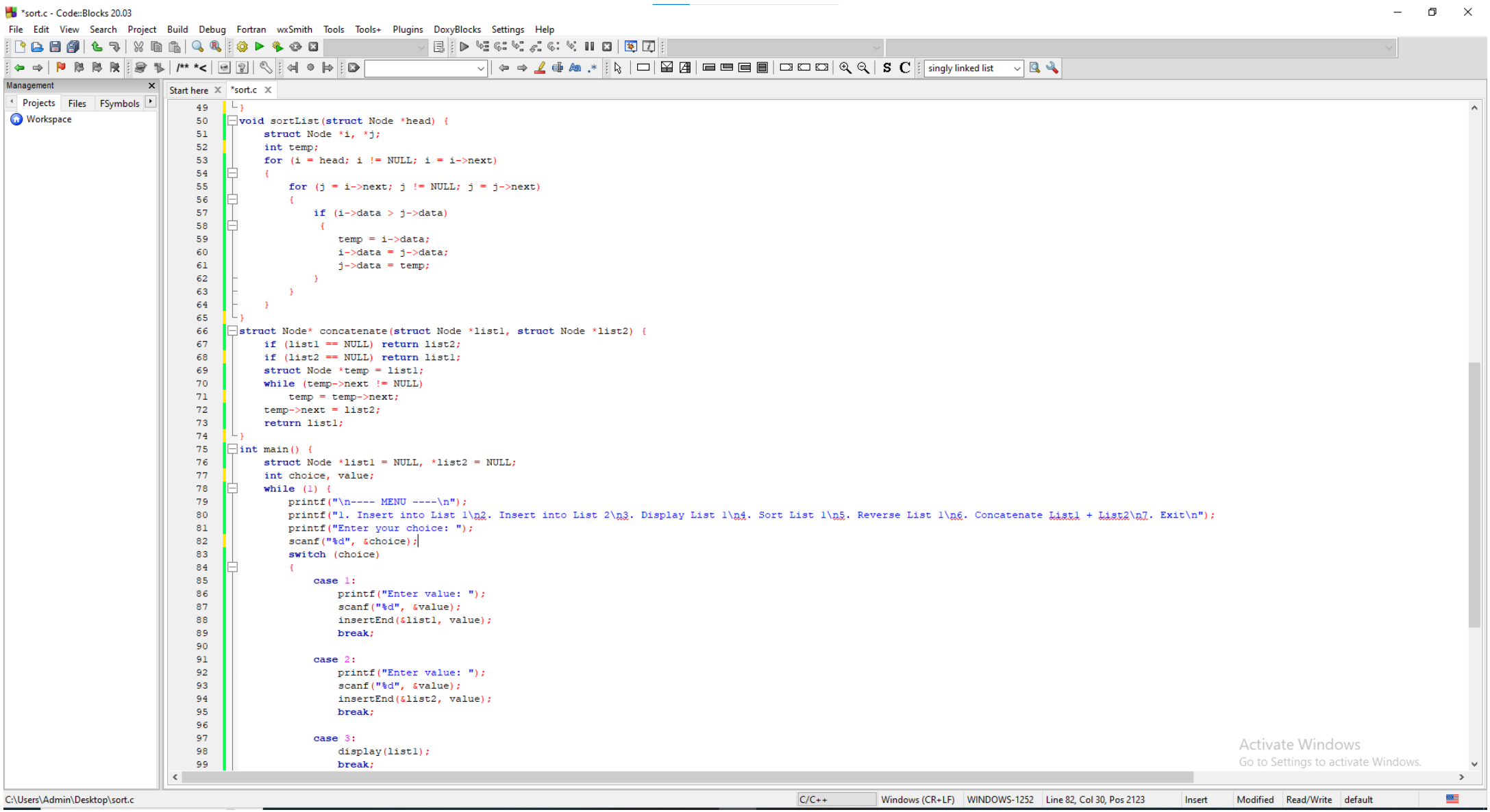
Projects Files FSymbols

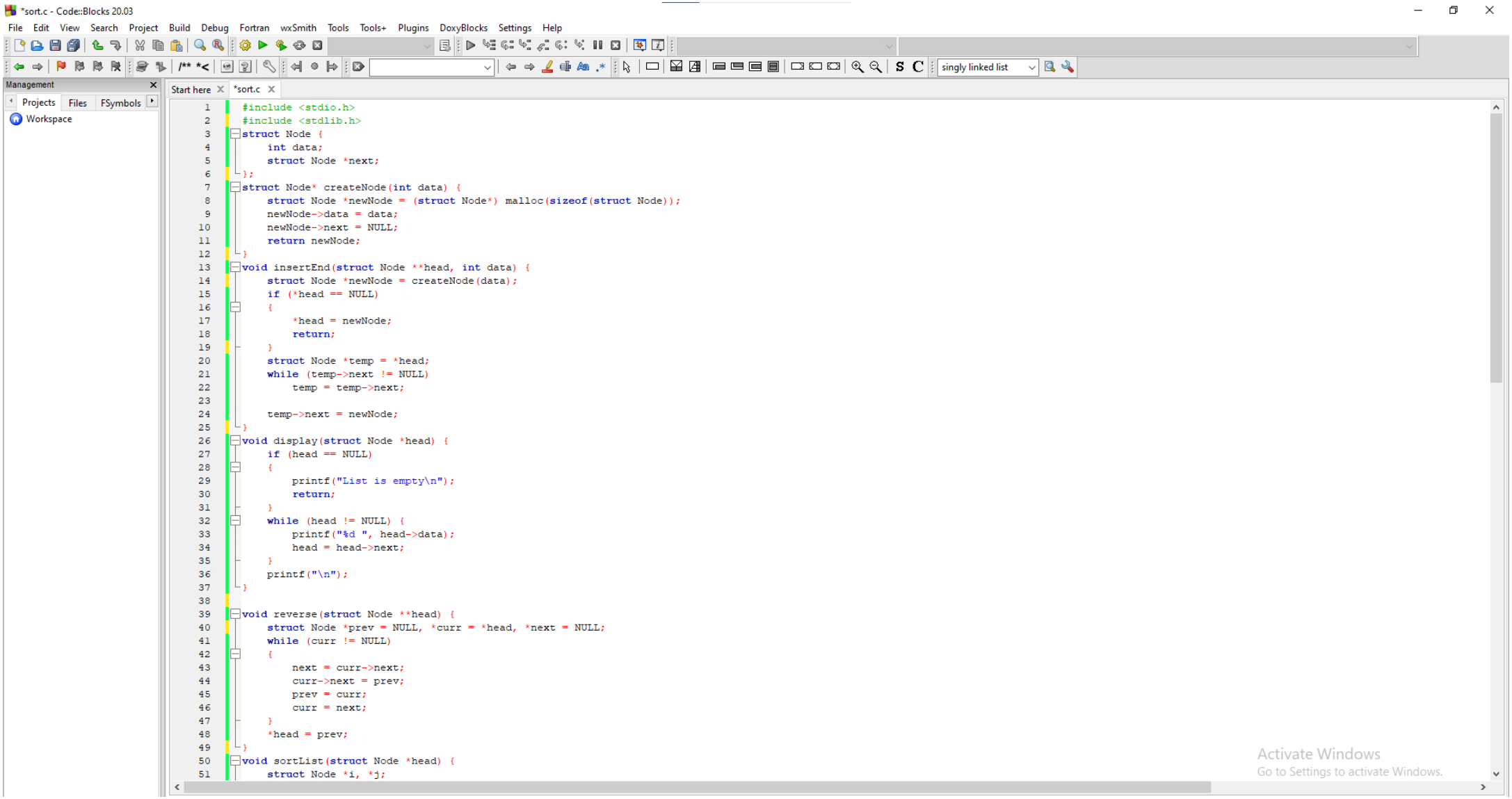
Workspace

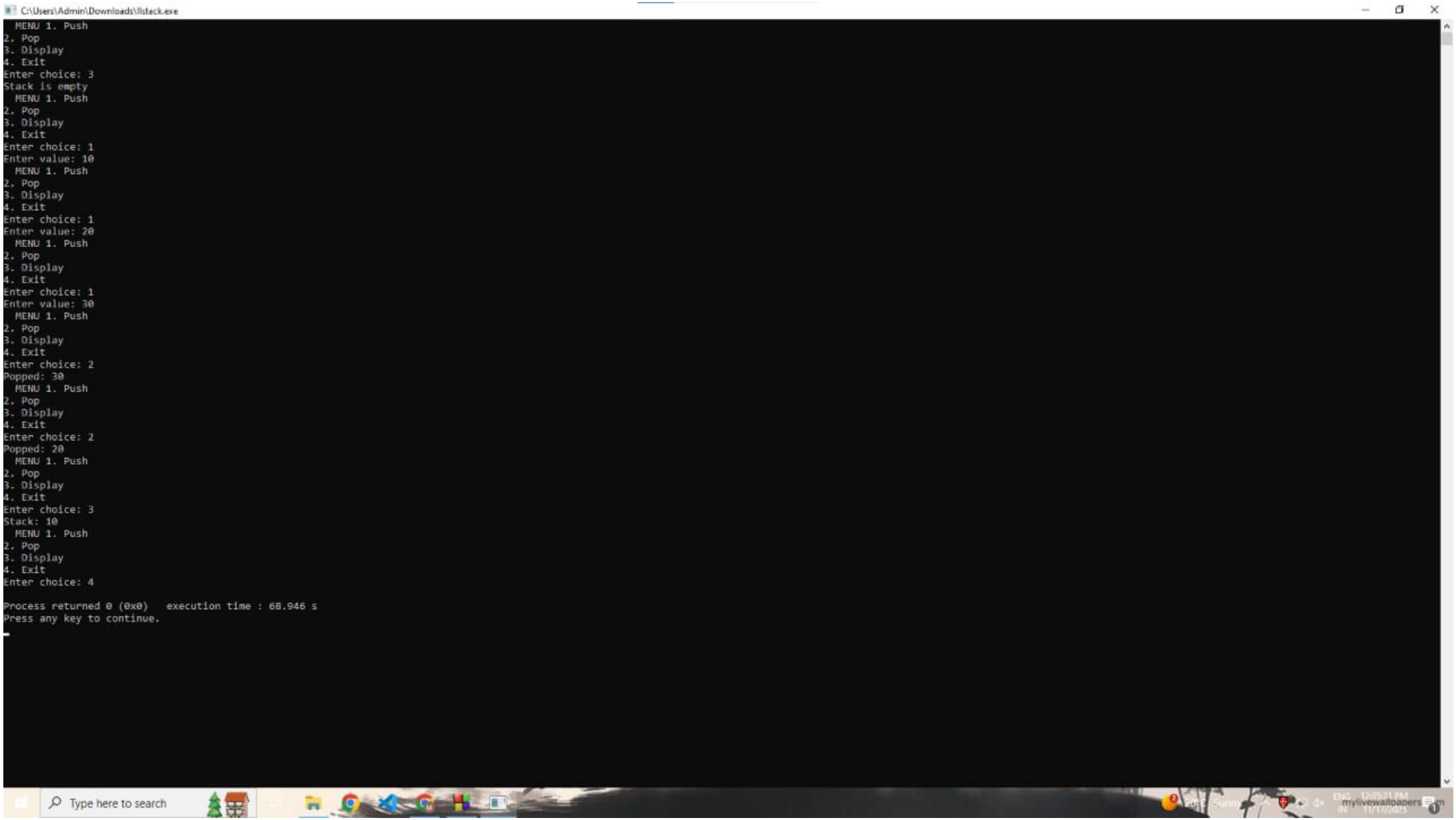
Start here | stack.c

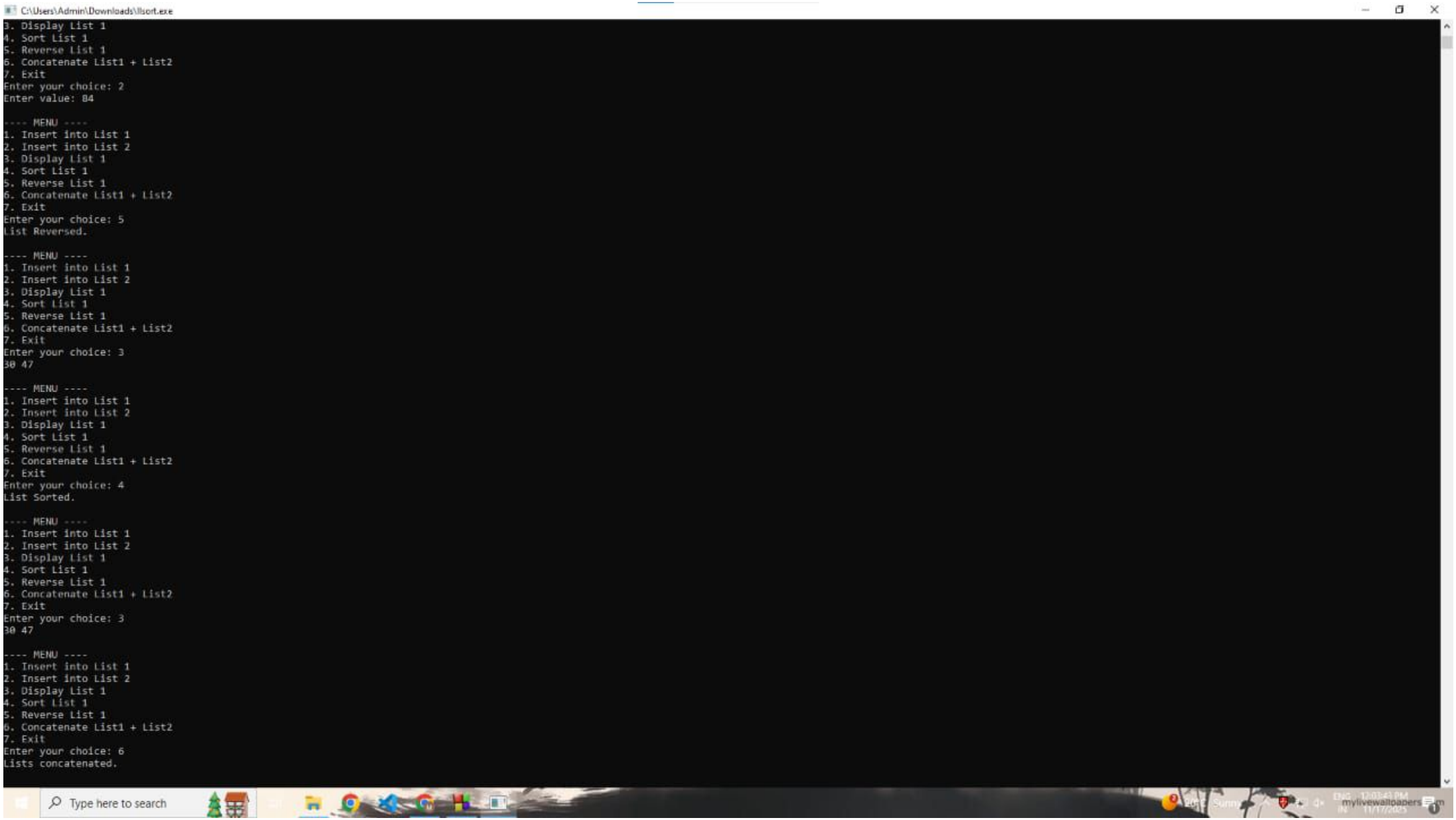
```c
#include <stdio.h>
#include <stdlib.h
struct node {
int data;
struct node *next;
};
struct node *top = NULL;
struct node *front = NULL, *rear = NULL;
void push(int value) {
struct node *newnode;
newnode = (struct node *)malloc(sizeof(struct node));
newnode->data = value;
newnode->next = top;
top = newnode;
printf("Pushed %d into stack\n", value);
}
void pop() {
struct node *temp;
if (top == NULL) {
printf("Stack is empty\n");
} else {
temp = top;
printf("Popped element: %d\n", top->data);
top = top->next;
free(temp);
}
}
void enqueue(int value) {
struct node *newnode;
newnode = (struct node *)malloc(sizeof(struct node));
newnode->data = value;
newnode->next = NULL;
if (front == NULL) {
front = rear = newnode;
} else {
rear->next = newnode;
rear = newnode;
}
printf("Enqueued %d into queue\n", value);
}
void dequeue() {
struct node *temp;
if (front == NULL) {
printf("Queue is empty\n");
} else {
temp = front;
printf("Dequeued element: %d\n", front->data);
front = front->next;
free(temp);
}
}
```

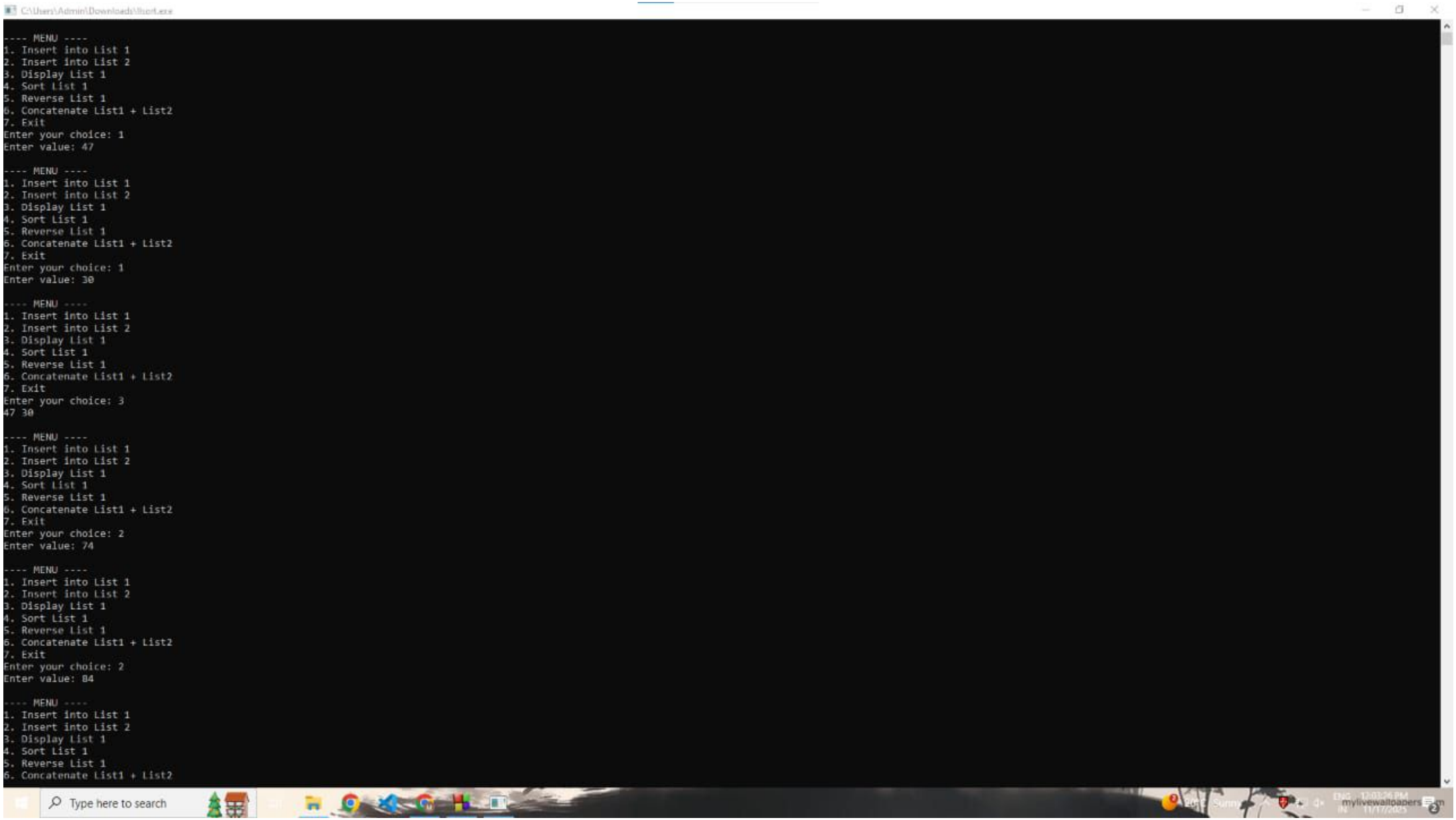

*sort.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
singly linked list
Management
Projects Files FSymbols
Workspace
Start here
*sort.c
}
int main() {
struct Node *list1 = NULL, *list2 = NULL;
int choice, value;
while (1) {
printf("\n---- MENU ----\n");
printf("1. Insert into List 1\n2. Insert into List 2\n3. Display List 1\n4. Sort List 1\n5. Reverse List 1\n6. Concatenate List1 + List2\n7. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);
switch (choice)
{
case 1:
printf("Enter value: ");
scanf("%d", &value);
insertEnd(&list1, value);
break;
case 2:
printf("Enter value: ");
scanf("%d", &value);
insertEnd(&list2, value);
break;
case 3:
display(list1);
break;
case 4:
sortList(list1);
printf("List Sorted.\n");
break;
case 5:
reverse(&list1);
printf("List Reversed.\n");
break;
case 6:
list1 = concatenate(list1, list2);
printf("Lists concatenated.\n");
break;
case 7:
exit(0);
default:
printf("Invalid choice!\n");
}
}
}
Activate Windows
Go to Settings to activate Windows.


*sort.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
singly linked list
Management
Projects Files FSymbols
Workspace
Start here
*sort.c
}
void sortList(struct Node *head) {
struct Node *i, *j;
int temp;
for (i = head; i != NULL; i = i->next)
{
for (j = i->next; j != NULL; j = j->next)
{
if (i->data > j->data)
{
temp = i->data;
i->data = j->data;
j->data = temp;
}
}
}
}
struct Node* concatenate(struct Node *list1, struct Node *list2) {
if (list1 == NULL) return list2;
if (list2 == NULL) return list1;
struct Node *temp = list1;
while (temp->next != NULL)
temp = temp->next;
temp->next = list2;
return list1;
}
int main() {
struct Node *list1 = NULL, *list2 = NULL;
int choice, value;
while (1) {
printf("\n---- MENU ----\n");
printf("1. Insert into List 1\n2. Insert into List 2\n3. Display List 1\n4. Sort List 1\n5. Reverse List 1\n6. Concatenate List1 + List2\n7. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);
switch (choice)
{
case 1:
printf("Enter value: ");
scanf("%d", &value);
insertEnd(&list1, value);
break;
case 2:
printf("Enter value: ");
scanf("%d", &value);
insertEnd(&list2, value);
break;
case 3:
display(list1);
break;
Activate Windows
Go to Settings to activate Windows.
C:\Users\Admin\Desktop\sort.c
C/C++
Windows (CR+LF)
WINDOWS-1252
Line 82, Col 30, Pos 2123
Insert
Modified
Read/Write
default


*sort.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
singly linked list
Management
Projects Files FSymbols
Workspace
Start here
*sort.c
#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;
return;
}
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");
return;
}
while (head != NULL) {
printf("%d ", head->data);
head = head->next;
}
printf("\n");
}
void reverse(struct Node **head) {
struct Node *prev = NULL, *curr = *head, *next = NULL;
while (curr != NULL)
{
next = curr->next;
curr->next = prev;
prev = curr;
curr = next;
}
*head = prev;
}
void sortList(struct Node *head) {
struct Node *i, *j;
Activate Windows
Go to Settings to activate Windows.


C:\Users\Admin\Downloads\llstack.exe
MENU 1. Push
2. Pop
3. Display
4. Exit
Enter choice: 3
Stack is empty
MENU 1. Push
2. Pop
3. Display
4. Exit
Enter choice: 1
Enter value: 10
MENU 1. Push
2. Pop
3. Display
4. Exit
Enter choice: 1
Enter value: 20
MENU 1. Push
2. Pop
3. Display
4. Exit
Enter choice: 1
Enter value: 30
MENU 1. Push
2. Pop
3. Display
4. Exit
Enter choice: 2
Popped: 30
MENU 1. Push
2. Pop
3. Display
4. Exit
Enter choice: 2
Popped: 20
MENU 1. Push
2. Pop
3. Display
4. Exit
Enter choice: 3
Stack: 10
MENU 1. Push
2. Pop
3. Display
4. Exit
Enter choice: 4
Process returned 0 (0x0) execution time : 68.946 s
Press any key to continue.
Type here to search


C:\Users\Admin\Downloads\llsort.exe
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 2
Enter value: 84
---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 5
List Reversed.
---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 3
30 47
---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 4
List Sorted.
---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 3
30 47
---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 6
Lists concatenated.
Type here to search


C:\Users\Admin\Downloads\llsort.exe
---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 1
Enter value: 47
---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 1
Enter value: 30
---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 3
47 30
---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 2
Enter value: 74
---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 2
Enter value: 84
---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
Type here to search

C:\Users\Admin\Downloads\llsort.exe

```
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 4
List Sorted.

---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 3
30 47

---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 6
Lists concatenated.

---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 3
30 47 74 84

---- MENU ----
1. Insert into List 1
2. Insert into List 2
3. Display List 1
4. Sort List 1
5. Reverse List 1
6. Concatenate List1 + List2
7. Exit
Enter your choice: 7

Process returned 0 (0x0)   execution time : 157.337 s
Press any key to continue.
```