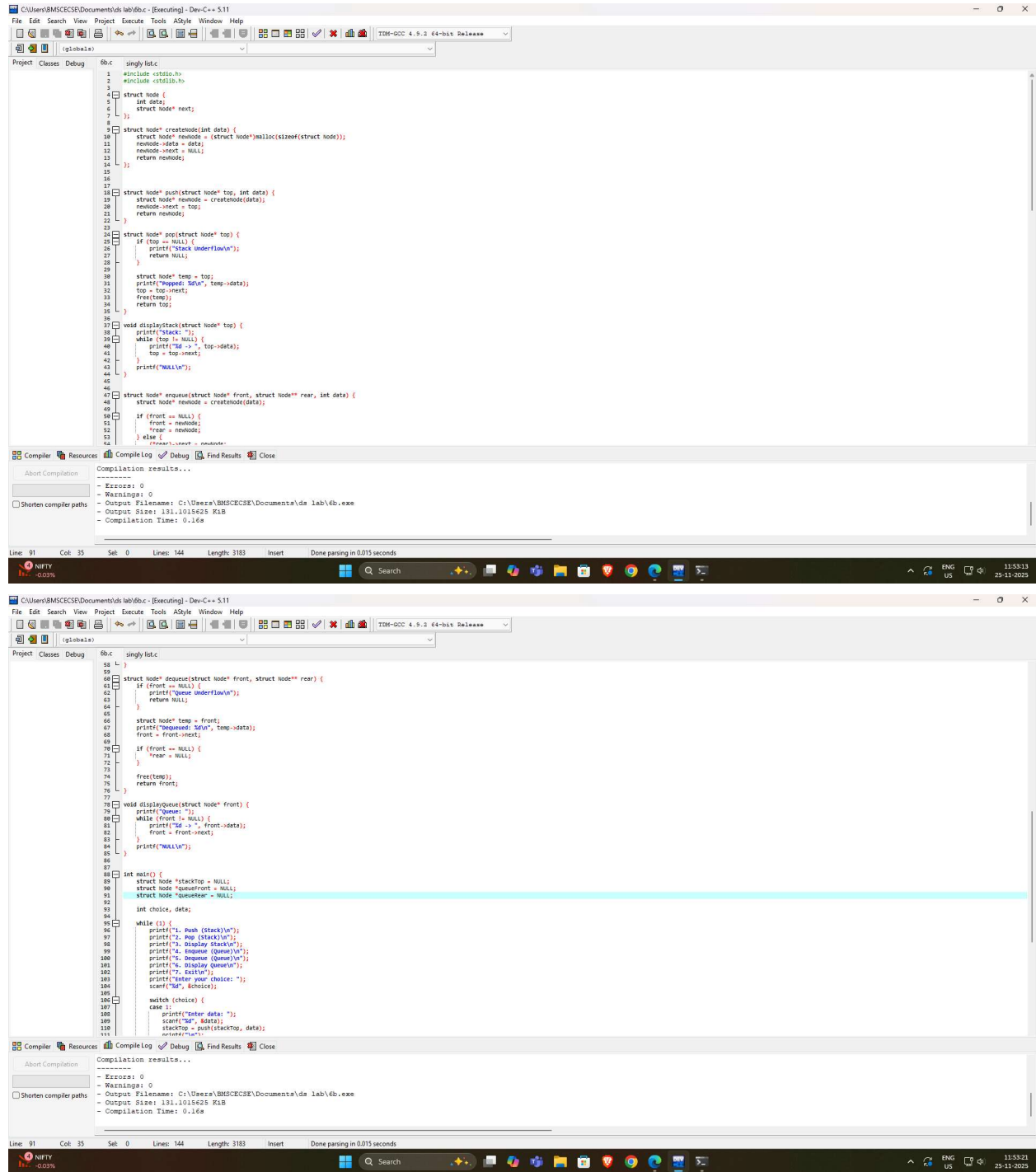
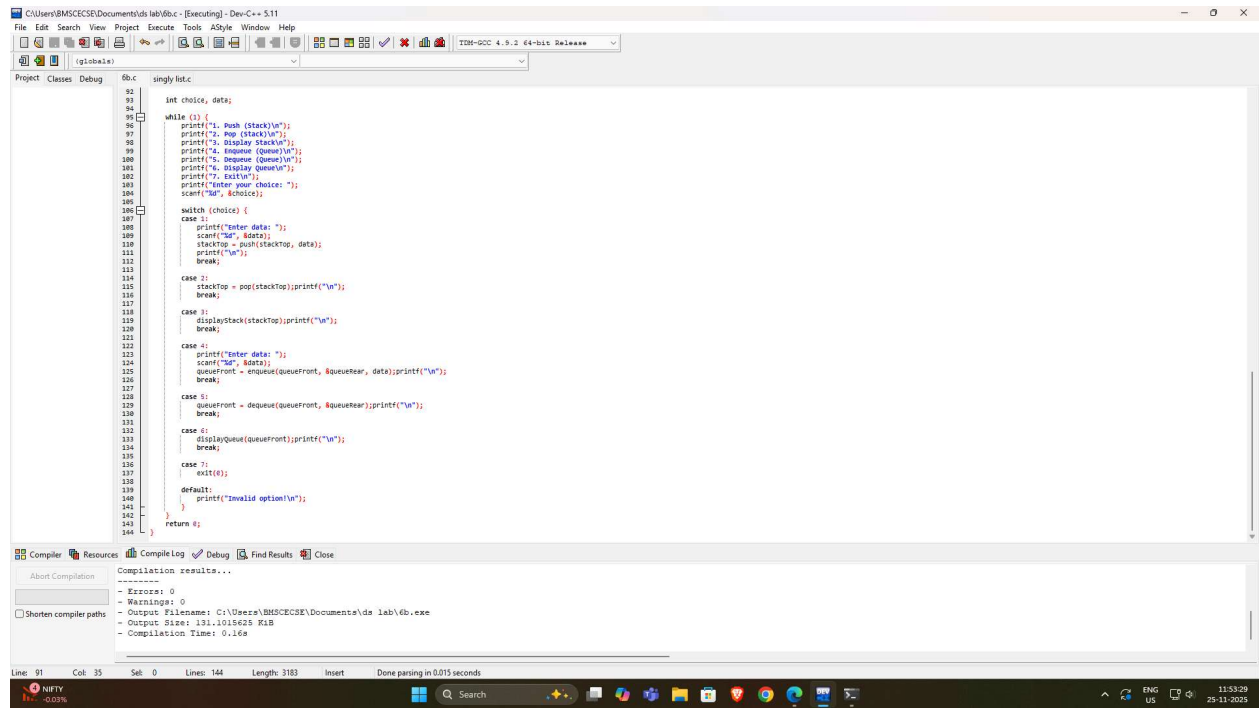


6b WAP to Implement Single Link List to simulate Stack & Queue Operations.



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct node {
5     int data;
6     struct node* next;
7 };
8
9 struct node* createNode(int data) {
10     struct node* newNode = (struct node*)malloc(sizeof(struct node));
11     newNode->data = data;
12     newNode->next = NULL;
13     return newNode;
14 }
15
16 struct node* push(struct node* top, int data) {
17     struct node* newNode = createNode(data);
18     newNode->next = top;
19     return newNode;
20 }
21
22 struct node* pop(struct node* top) {
23     if (top == NULL) {
24         printf("Stack Underflow\n");
25         return NULL;
26     }
27     struct node* temp = top;
28     printf("Popped: %d\n", temp->data);
29     top = top->next;
30     free(temp);
31     return top;
32 }
33
34 void displayStack(struct node* top) {
35     printf("Stack: ");
36     while (top != NULL) {
37         printf("%d -> ", top->data);
38         top = top->next;
39     }
40     printf("NULL\n");
41 }
42
43 struct node* enqueue(struct node* front, struct node** rear, int data) {
44     struct node* newNode = createNode(data);
45     if (front == NULL) {
46         *rear = newNode;
47         front = newNode;
48     } else {
49         (*rear)->next = newNode;
50         *rear = newNode;
51     }
52 }
53
54 struct node* dequeue(struct node* front, struct node** rear) {
55     if (front == NULL) {
56         printf("Queue Underflow\n");
57         return NULL;
58     }
59     struct node* temp = front;
60     printf("Dequeued: %d\n", temp->data);
61     front = front->next;
62     if (front == NULL) {
63         *rear = NULL;
64     }
65     free(temp);
66     return front;
67 }
68
69 void displayQueue(struct node* front) {
70     printf("Queue: ");
71     while (front != NULL) {
72         printf("%d -> ", front->data);
73         front = front->next;
74     }
75     printf("NULL\n");
76 }
77
78 int main() {
79     struct node* stackTop = NULL;
80     struct node* queueFront = NULL;
81     struct node* queueRear = NULL;
82     int choice, data;
83
84     while (1) {
85         printf("\n 1. Push (Stack)\n");
86         printf("\n 2. Pop (Stack)\n");
87         printf("\n 3. Display Stack\n");
88         printf("\n 4. Enqueue (Queue)\n");
89         printf("\n 5. Dequeue (Queue)\n");
90         printf("\n 6. Display Queue\n");
91         printf("\n 7. Exit\n");
92         printf("Enter your choice: ");
93         scanf("%d", &choice);
94
95         switch (choice) {
96             case 1:
97                 printf("Enter data: ");
98                 scanf("%d", &data);
99                 stackTop = push(stackTop, data);
100                 printf("Pushed\n");
101             case 2:
102                 stackTop = pop(stackTop);
103                 printf("Popped\n");
104             case 3:
105                 displayStack(stackTop);
106             case 4:
107                 enqueue(queueFront, &queueRear, data);
108                 printf("Enqueued\n");
109             case 5:
110                 queueFront = dequeue(queueFront, &queueRear);
111                 printf("Dequeued\n");
112             case 6:
113                 displayQueue(queueFront);
114             case 7:
115                 return 0;
116             default:
117                 printf("Invalid choice\n");
118         }
119     }
120 }
```



```
C:\Users\BMSCECSB\Documents x + v
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 1
Enter data: 12

1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 1
Enter data: 23

1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 2
Popped: 23

1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 3
Stack: 12 -> NULL

1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 4
Enter data: 45

1. Push (Stack)
```

```
C:\Users\BMSCECSB\Documents x + v
Enter data: 45

1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 4
Enter data: 56

1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 5
Dequeued: 45

1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 6
Queue: 56 -> NULL

1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter your choice: 7

-----
Process exited after 17.39 seconds with return value 0
Press any key to continue . . .
```