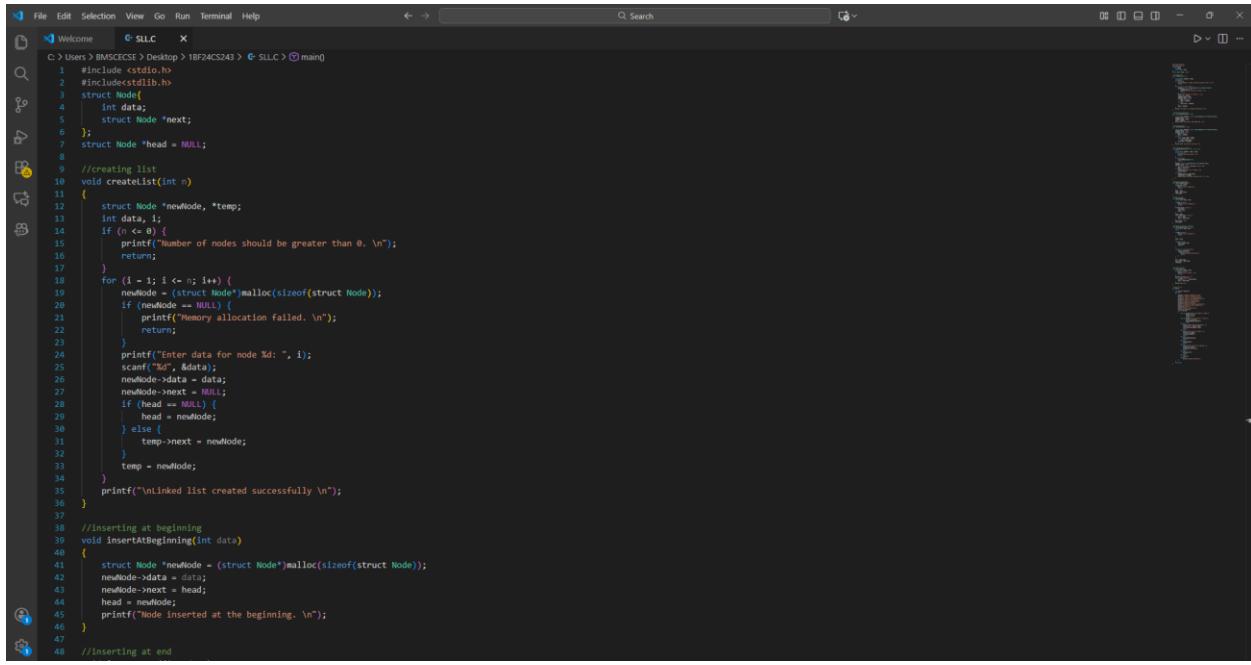
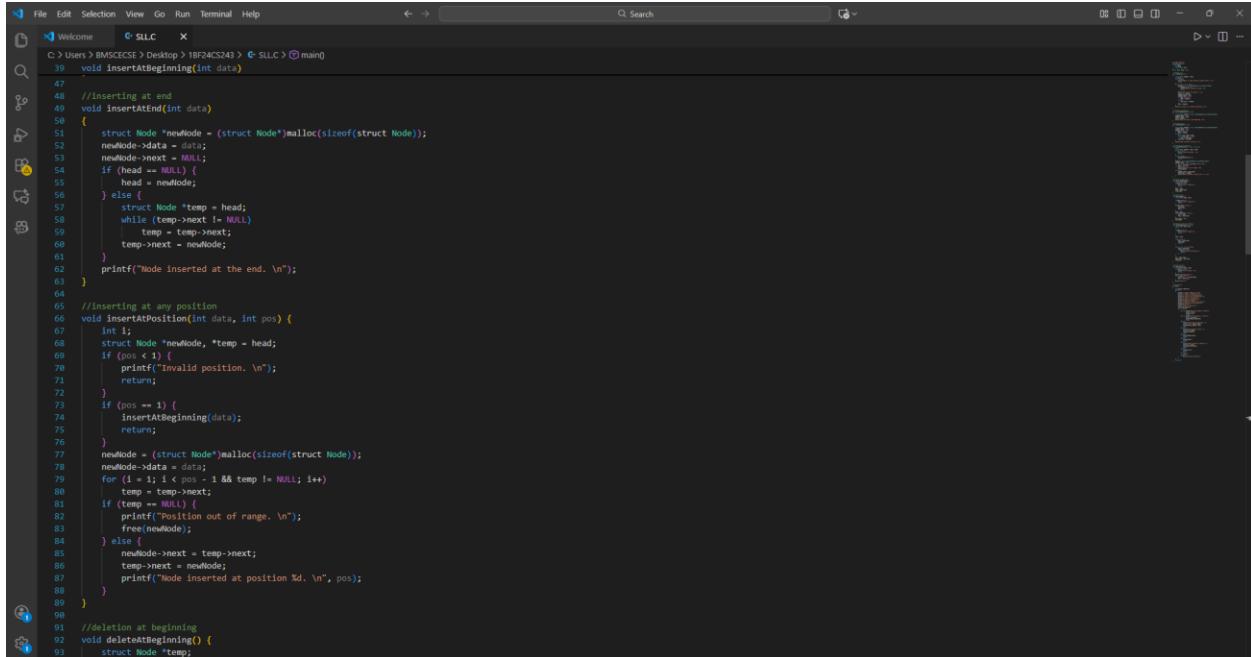


LAB(18-11-2025)

INPUT



```
C > Users > BMSCSCE > Desktop > 1BF24CS243 > C-SLLC > main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 struct Node {
4     int data;
5     struct Node *next;
6 };
7 struct Node *head = NULL;
8
9 //creating list
10 void createlist(int n)
11 {
12     struct Node *newNode, *temp;
13     int data, i;
14     if (n <= 0) {
15         printf("Number of nodes should be greater than 0. \n");
16         return;
17     }
18     for (i = 1; i <= n; i++) {
19         newNode = (struct Node*)malloc(sizeof(struct Node));
20         if (newNode == NULL) {
21             printf("Memory allocation failed. \n");
22             return;
23         }
24         printf("Enter data for node %d: ", i);
25         scanf("%d", &data);
26         newNode->data = data;
27         newNode->next = NULL;
28         if (head == NULL) {
29             head = newNode;
30         } else {
31             temp->next = newNode;
32         }
33         temp = newNode;
34     }
35     printf("\nLinked list created successfully \n");
36 }
37
38 //inserting at beginning
39 void insertAtBeginning(int data)
40 {
41     struct Node *newNode = (struct Node*)malloc(sizeof(struct Node));
42     newNode->data = data;
43     newNode->next = head;
44     head = newNode;
45     printf("Node inserted at the beginning. \n");
46 }
47
48 //inserting at end
```



```
C > Users > BMSCSCE > Desktop > 1BF24CS243 > C-SLLC > main.c
39 void insertAtBeginning(int data)
40 {
41
42
43
44
45
46
47
48 //inserting at end
49 void insertAtEnd(int data)
50 {
51     struct Node *newNode = (struct Node*)malloc(sizeof(struct Node));
52     newNode->data = data;
53     newNode->next = NULL;
54     if (head == NULL) {
55         head = newNode;
56     } else {
57         struct Node *temp = head;
58         while (temp->next != NULL)
59             temp = temp->next;
60         temp->next = newNode;
61     }
62     printf("Node inserted at the end. \n");
63 }
64
65 //inserting at any position
66 void insertAtPosition(int data, int pos) {
67     int i;
68     struct Node *newNode, *temp = head;
69     if (pos < 1) {
70         printf("Invalid position. \n");
71         return;
72     }
73     if (pos == 1) {
74         insertAtBeginning(data);
75         return;
76     }
77     newNode = (struct Node*)malloc(sizeof(struct Node));
78     newNode->data = data;
79     for (i = 1; i < pos - 1 && temp != NULL; i++)
80         temp = temp->next;
81     if (temp == NULL) {
82         printf("Position out of range. \n");
83         free(newNode);
84     } else {
85         newNode->next = temp->next;
86         temp->next = newNode;
87         printf("Node inserted at position %d. \n", pos);
88     }
89 }
90
91 //deletion at beginning
92 void deleteAtBeginning() {
93     struct Node *temp;
```

```
C:\Users> BMSCECSE > Desktop > 1BF24CS243 > C_SLLC > main.c
66 void insertAtPosition(int data, int pos) {
89 }
90
91 //Deletion at beginning
92 void deletedAtBeginning() {
93     struct Node *temp;
94     if(head == NULL) {
95         printf("List is empty\n");
96         return;
97     }
98     temp = head;
99     head = head->next;
100    free(temp);
101 }
102
103 //Deletion at end
104 void deletedAtEnd() {
105     struct Node *temp, *prev;
106
107     if(head == NULL) {
108         printf("List is empty\n");
109         return;
110     }
111     if(head->next == NULL) {
112         free(head);
113         head = NULL;
114         return;
115     }
116     temp = head;
117     while(temp->next != NULL) {
118         prev = temp;
119         temp = temp->next;
120     }
121     prev->next = NULL;
122     free(temp);
123 }
124
125 //Deletion at specific position
126 void deletedAtPosition(int pos) {
127     struct Node *temp, *del;
128     int i;
129
130     if(head == NULL) {
131         printf("List is empty\n");
132         return;
133     }
134     temp = head;
135 }
```

```
C:\Users> BMSCECSE > Desktop > 1BF24CS243 > C_SLLC > main.c
125 //Deletion at specific position
126 void deletedAtPosition(int pos) {
127     struct Node *temp, *del;
128     int i;
129
130     if(head == NULL) {
131         printf("List is empty\n");
132         return;
133     }
134     temp = head;
135
136     if(pos == 1) {
137         head = head->next;
138         free(temp);
139         return;
140     }
141
142     for(i = 1; i < pos-1; i++) {
143         temp = temp->next;
144         if(temp == NULL) {
145             printf("Invalid Position\n");
146             return;
147         }
148     }
149
150     del = temp->next;
151     temp->next = del->next;
152     free(del);
153 }
154
155
156 //Display function
157 void display() {
158     struct Node *temp = head;
159     if (head == NULL) {
160         printf("List is empty. \n");
161         return;
162     }
163
164     printf("Unlinked List: ");
165     while (temp != NULL) {
166         printf("%d -> ", temp->data);
167         temp = temp->next;
168     }
169     printf("NULL\n");
170 }
171
172 //main function
```

```
C:\Users> BMGECSE> Desktop> 1BF24CS243 > SLLC > main()
158 void displayList() {
159     /
160     //mainfunction
161     int main()
162     {
163         int choice,n,data,pos;
164         while(1)
165         {
166             printf("--singly linked list--\n");
167             printf("1.create a linked list\n");
168             printf("2.insert at the beginning\n");
169             printf("3.insert at any position\n");
170             printf("4.insert at the end\n");
171             printf("5.delete at beginning\n");
172             printf("6.delete at end\n");
173             printf("7.display all the elements\n");
174             printf("8.exit\n");
175             printf("Enter ur choice\n");
176             scanf("%d",&choice);
177             switch(choice)
178             {
179                 case 1: printf("enter the number of nodes");
180                 scanf("%d",&n);
181                 createList(n);
182                 break;
183                 case 2: printf("enter the data to insert");
184                 scanf("%d",&data);
185                 insertAtBeginning(data);
186                 break;
187                 case 3:
188                     printf("Enter data and position: ");
189                     scanf("%d %d",&data,&pos);
190                     insertAtPosition(data, pos);
191                     break;
192                 case 4:
193                     printf("Enter data to insert: ");
194                     scanf("%d",&data);
195                     insertAtEnd(data);
196                     break;
197                 case 5:
198                     deleteAtBeginning();
199                     break;
200                 case 6:
201                     deleteAtEnd();
202                     break;
203                 case 7:
204                     break;
205             }
206         }
207     }
208     return 0;
209 }
```

```
C:\Users> BMGECSE> Desktop> 1BF24CS243 > SLLC > main()
173 int main()
174 {
175     while(1)
176     {
177         printf("1.create a linked list\n");
178         printf("2.insert at the beginning\n");
179         printf("3.insert at any position\n");
180         printf("4.insert at the end\n");
181         printf("5.delete at beginning\n");
182         printf("6.delete at end\n");
183         printf("7.display all the elements\n");
184         printf("8.exit\n");
185         printf("Enter ur choice\n");
186         scanf("%d",&choice);
187         switch(choice)
188         {
189             case 1: printf("enter the number of nodes");
190                 scanf("%d",&n);
191                 createList(n);
192                 break;
193                 case 2: printf("enter the data to insert");
194                 scanf("%d",&data);
195                 insertAtBeginning(data);
196                 break;
197                 case 3:
198                     printf("Enter data and position: ");
199                     scanf("%d %d",&data,&pos);
200                     insertAtPosition(data, pos);
201                     break;
202                 case 4:
203                     printf("Enter data to insert: ");
204                     scanf("%d",&data);
205                     insertAtEnd(data);
206                     break;
207                 case 5:
208                     deleteAtBeginning();
209                     break;
210                 case 6:
211                     deleteAtEnd();
212                     break;
213                 case 7:
214                     printf("Enter position to delete: ");
215                     scanf("%d",&pos);
216                     deleteAtPosition(pos);
217                     break;
218                 case 8:
219                     displayList();
220                     break;
221                 case 9:
222                     exit(0);
223                 default:
224                     printf("Invalid choice\n");
225         }
226     }
227 }
228 return 0;
229 }
```

OUTPUT

```
File Edit Selection View Go Run Terminal Help ← → Q Search
Welcome to SLLC X
C:\Users\BMSCESE\Desktop>1BF24CS243> SLLC > main()
171 int main()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\BMSCESE> cd "C:\Users\BMSCESE\Desktop\1BF24CS243\output"
PS C:\Users\BMSCESE> cd "C:\Users\BMSCESE\Desktop\1BF24CS243\output"
PS C:\Users\BMSCESE\Desktop\1BF24CS243\output> & .\SLL.exe
--singly linked list
PS C:\Users\BMSCESE> cd "C:\Users\BMSCESE\Desktop\1BF24CS243\output"
PS C:\Users\BMSCESE\Desktop\1BF24CS243\output> & .\SLL.exe
PS C:\Users\BMSCESE> cd "C:\Users\BMSCESE\Desktop\1BF24CS243\output"
PS C:\Users\BMSCESE\Desktop\1BF24CS243\output> & .\SLL.exe
PS C:\Users\BMSCESE\Desktop\1BF24CS243\output> & .\SLL.exe
1.create a linked list
PS C:\Users\BMSCESE> cd "C:\Users\BMSCESE\Desktop\1BF24CS243\output"
PS C:\Users\BMSCESE\Desktop\1BF24CS243\output> & .\SLL.exe
--singly linked list
1.create a linked list
2.insert at the beginning
3.insert at the any positin
4.insert at the end
5.delete at the beginning
6.delete at end
7.delete at specific position
8.display all the elements
9.exit
Enter urs choice
1
enter the number of nodes4
Enter data for node 1: 10
Enter data for node 2: 20
Enter data for node 3: 30
Enter data for node 4: 40
Linked list created successfully
--singly linked list
1.create a linked list
2.insert at the beginning
3.insert at the any positin
4.insert at the end
5.delete at beginning
6.delete at end
7.delete at specific position
8.display all the elements
9.exit
Enter urs choice
5
--singly linked list--
1.create a linked list
2.insert at the beginning
3.insert at the any positin
4.insert at the end
5.delete at beginning
6.delete at end
```

```
File Edit Selection View Go Run Terminal Help ← → Q. Search
Welcome 6: SLLC ×
C:\Users\BMSCESE\Desktop>1BF24CS243 > 6: SLLC > main()
171 [int main()]
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
--singly linked list--
1.create a linked list
2.insert at the beginning
3.insert at the any positin
4.insert at the end
5.delete at beginning
6.delete at end
7.delete at specific position
8.display all the elements
9.exit
Enter urs choice
5
--singly linked list--
1.create a linked list
2.insert at the beginning
3.insert at the any positin
4.insert at the end
5.delete at beginning
6.delete at end
7.delete at specific position
8.display all the elements
9.exit
Enter urs choice
6
--singly linked list--
1.create a linked list
2.insert at the beginning
3.insert at the any positin
4.insert at the end
5.delete at beginning
6.delete at end
7.delete at specific position
8.display all the elements
9.exit
Enter urs choice
7
Enter position to delete: 1
--singly linked list--
1.create a linked list
2.insert at the beginning
3.insert at the any positin
4.insert at the end
5.delete at beginning
6.delete at end
7.delete at specific position
8.display all the elements
9.exit
Enter urs choice
8
Linked List: 30 -> NULL
x 0 A ⚡ Debug ⚡ Compile ⚡ Compile & Run
Ln 173, Col 11 Spaces:4 UTF-8 CRLF (4 C++ Signed out Win32
```

The screenshot shows a terminal window titled "SLLC" with the following content:

```
C:\> users > BMSCSESE > Desktop > 1BF24CS243 > SLLC > main()
```

TERMINAL PORTS

```
171 int main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
6.delete at end  
7.delete at specific position  
8.display all the elements  
9.exit  
Enter lrs choice  
7  
Enter position to delete: 1  
--singly linked list--  
1.create a linked list  
2.insert at the beginning  
3.insert at the any positin  
4.insert at the end  
5.delete at beginning  
6.delete at end  
7.delete at specific position  
8.display all the elements  
9.exit  
Enter lrs choice  
8  
Linked List: 30 -> NULL  
--singly linked list--  
1.create a linked list  
2.insert at the beginning  
3.insert at the any positin  
4.insert at the end  
5.delete at beginning  
6.delete at end  
7.delete at specific position  
8.display all the elements  
9.exit  
Enter lrs choice  
9
```

PS C:\Users\BMSCSESE\Desktop\1BF24CS243\output>