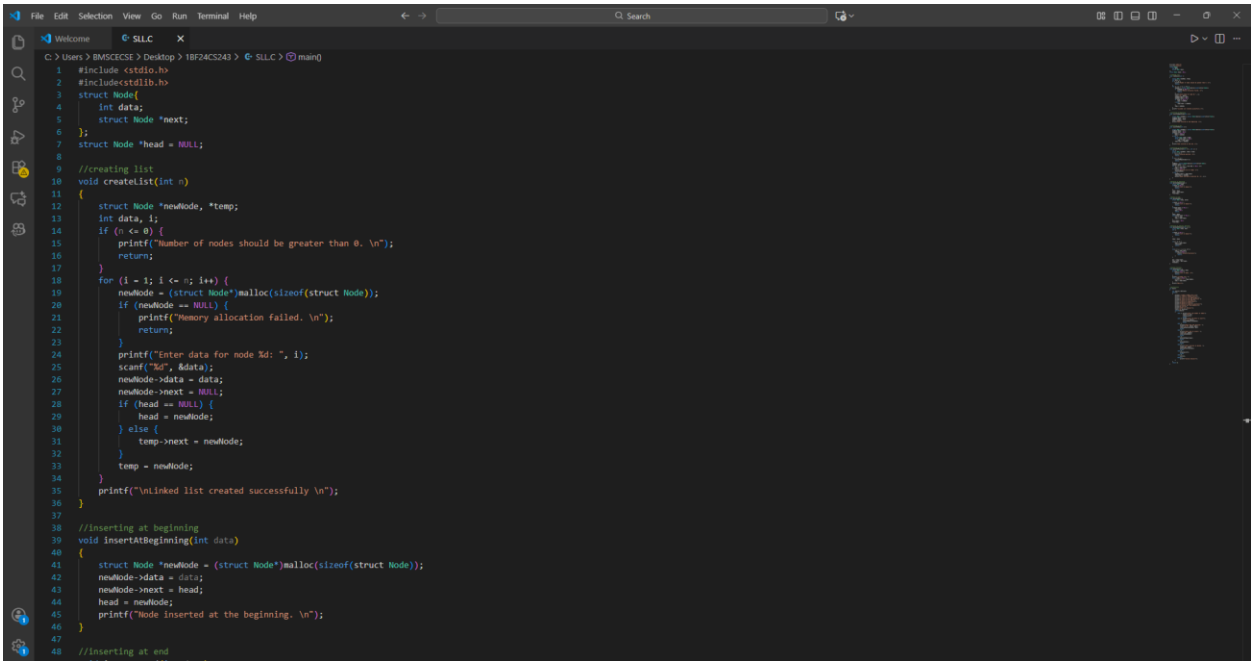
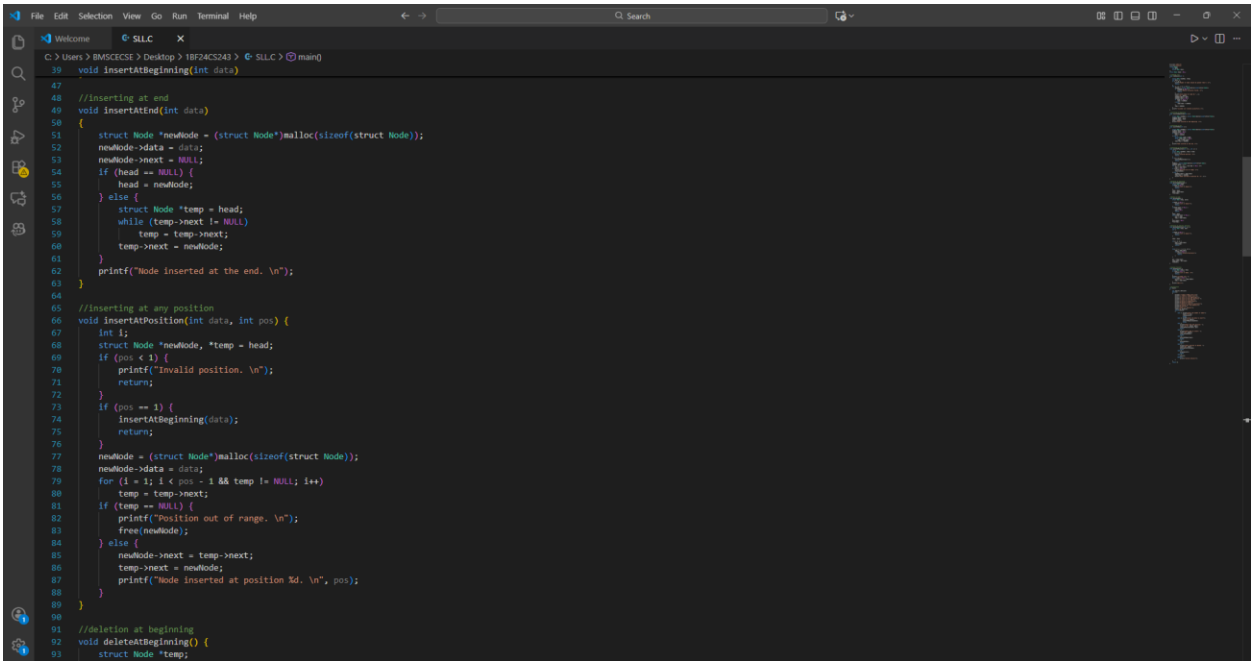
LAB(18-11-2025)

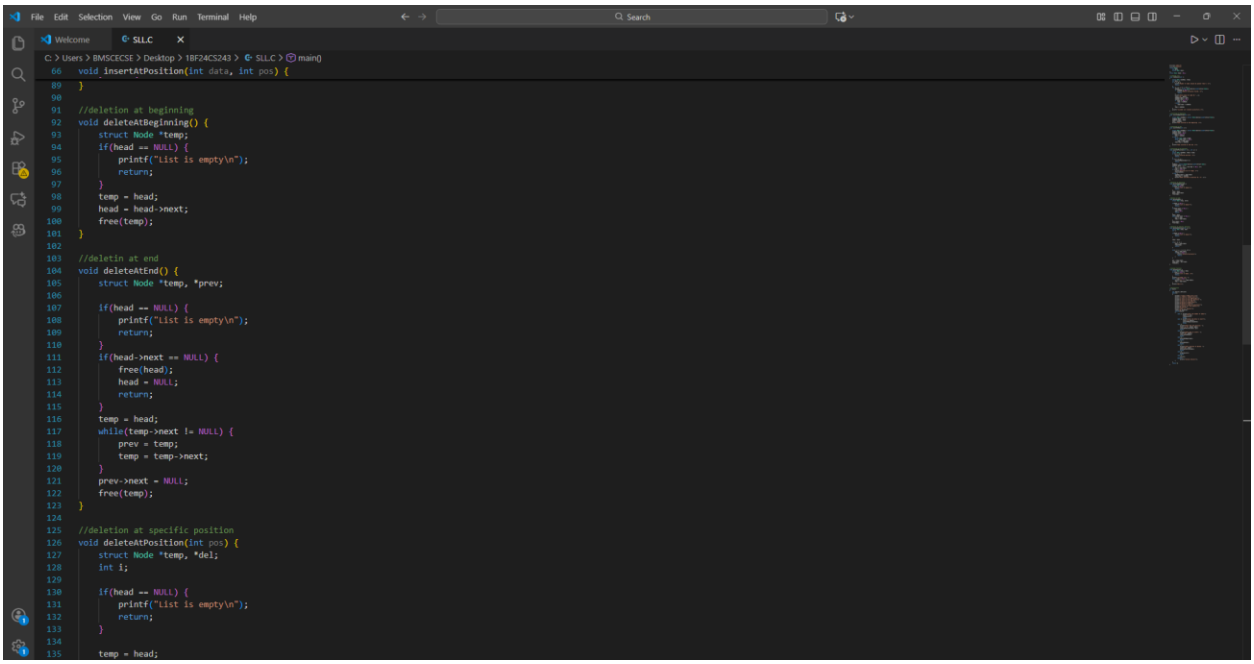
INPUT



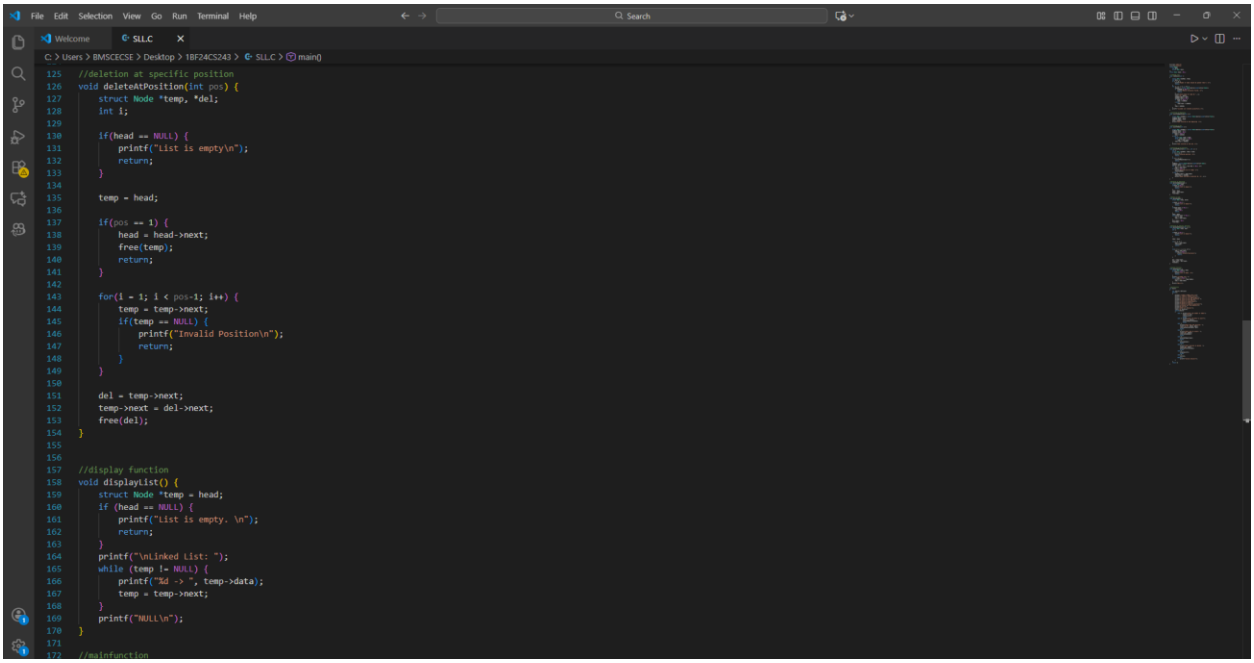
```c
#include <stdio.h>
#include<stdlib.h>
struct Node{
    int data;
    struct Node *next;
};
struct Node *head = NULL;

//creating list
void createList(int n)
{
    struct Node *newNode, *temp;
    int data, i;
    if (n <= 0) {
        printf("Number of nodes should be greater than 0. \n");
        return;
    }
    for (i = 1; i <= n; i++) {
        newNode = (struct Node*)malloc(sizeof(struct Node));
        if (newNode == NULL) {
            printf("Memory allocation failed. \n");
            return;
        }
        printf("Enter data for node %d: ", i);
        scanf("%d", &data);
        newNode->data = data;
        newNode->next = NULL;
        if (head == NULL) {
            head = newNode;
        } else {
            temp->next = newNode;
        }
        temp = newNode;
    }
    printf("\nLinked list created successfully \n");
}

//inserting at beginning
void insertAtBeginning(int data)
{
    struct Node *newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = head;
    head = newNode;
    printf("Node inserted at the beginning. \n");
}

//inserting at end
```



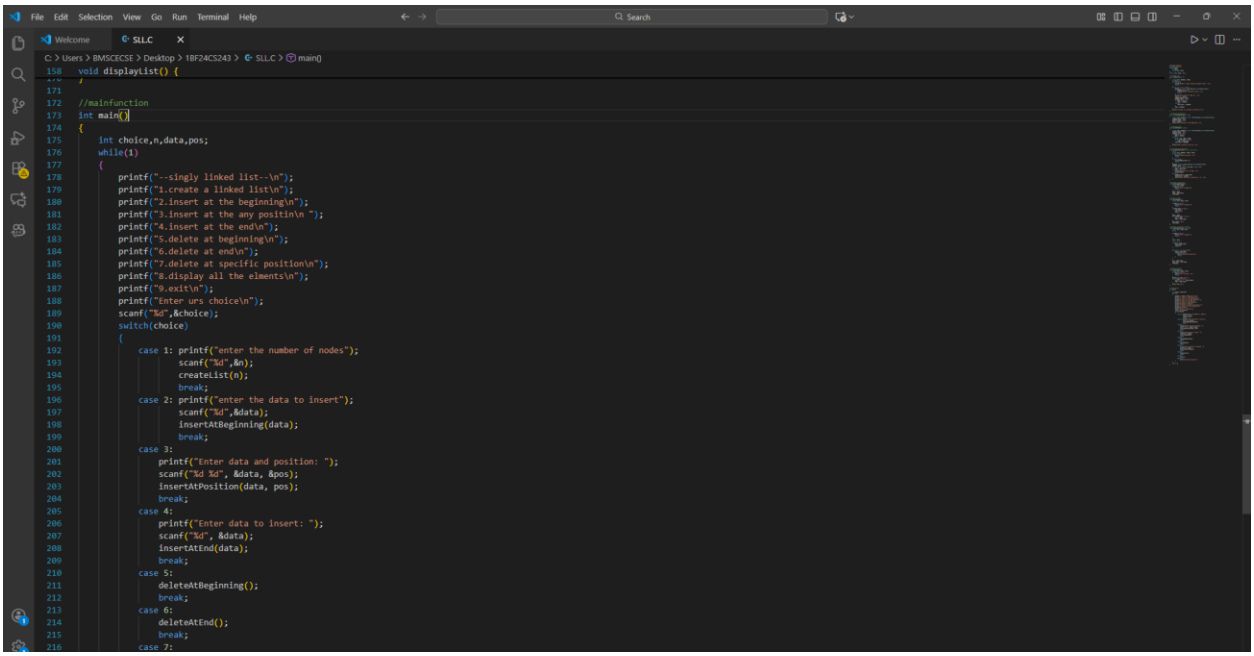
```c
void insertAtBeginning(int data)

//inserting at end
void insertAtEnd(int data)
{
    struct Node *newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    if (head == NULL) {
        head = newNode;
    } else {
        struct Node *temp = head;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = newNode;
    }
    printf("Node inserted at the end. \n");
}

//inserting at any position
void insertAtPosition(int data, int pos) {
    int i;
    struct Node *newNode, *temp = head;
    if (pos < 1) {
        printf("Invalid position. \n");
        return;
    }
    if (pos == 1) {
        insertAtBeginning(data);
        return;
    }
    newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    for (i = 1; i < pos - 1 && temp != NULL; i++)
        temp = temp->next;
    if (temp == NULL) {
        printf("Position out of range. \n");
        free(newNode);
    } else {
        newNode->next = temp->next;
        temp->next = newNode;
        printf("Node inserted at position %d. \n", pos);
    }
}

//deletion at beginning
void deleteAtBeginning() {
    struct Node *temp;
```



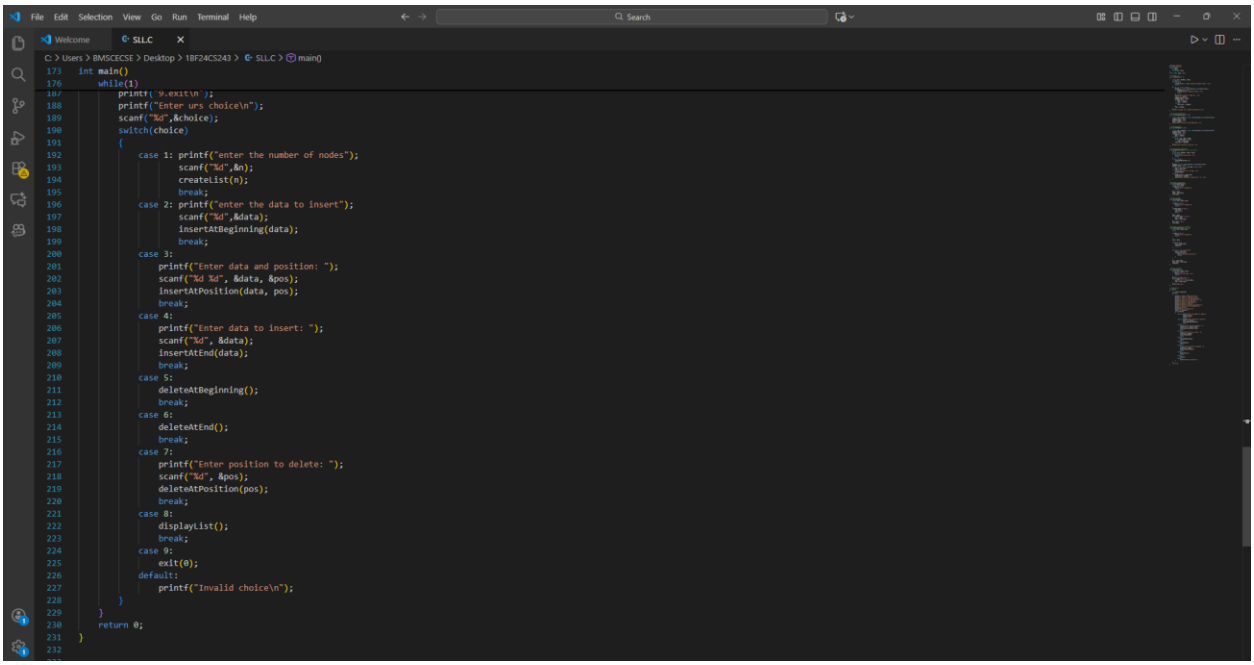
```
void insertAtPosition(int data, int pos) {
}

//deletion at beginning
void deleteAtBeginning() {
    struct Node *temp;
    if(head == NULL) {
        printf("List is empty\n");
        return;
    }
    temp = head;
    head = head->next;
    free(temp);
}

//deletin at end
void deleteAtEnd() {
    struct Node *temp, *prev;

    if(head == NULL) {
        printf("List is empty\n");
        return;
    }
    if(head->next == NULL) {
        free(head);
        head = NULL;
        return;
    }
    temp = head;
    while(temp->next != NULL) {
        prev = temp;
        temp = temp->next;
    }
    prev->next = NULL;
    free(temp);
}

//deletion at specific position
void deleteAtPosition(int pos) {
    struct Node *temp, *del;
    int i;

    if(head == NULL) {
        printf("List is empty\n");
        return;
    }

    temp = head;
```



```
//deletion at specific position
void deleteAtPosition(int pos) {
    struct Node *temp, *del;
    int i;

    if(head == NULL) {
        printf("List is empty\n");
        return;
    }

    temp = head;

    if(pos == 1) {
        head = head->next;
        free(temp);
        return;
    }

    for(i = 1; i < pos-1; i++) {
        temp = temp->next;
        if(temp == NULL) {
            printf("Invalid Position\n");
            return;
        }
    }

    del = temp->next;
    temp->next = del->next;
    free(del);
}


//display function
void displayList() {
    struct Node *temp = head;
    if (head == NULL) {
        printf("List is empty. \n");
        return;
    }
    printf("\nLinked List: ");
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

//mainfunction
```

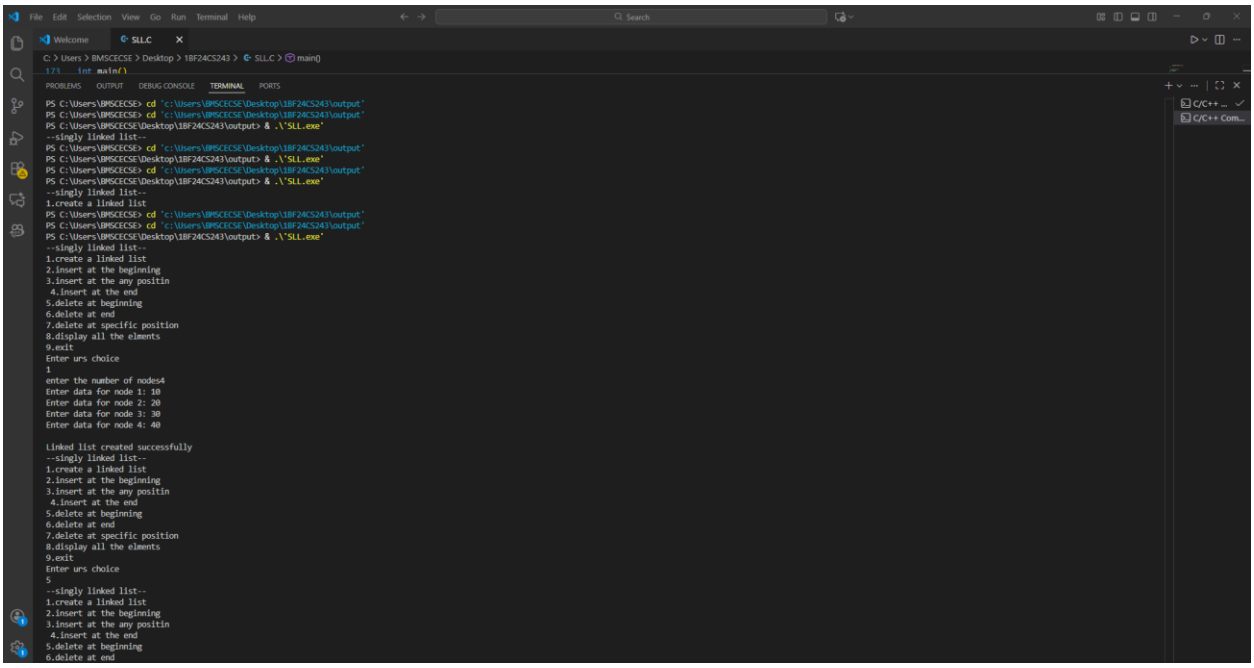


```
158  void displayList() {
172  //mainfunction
173  int main()
174  {
175      int choice,n,data,pos;
176      while(1)
177      {
178          printf("--singly linked list--\n");
179          printf("1.create a linked list\n");
180          printf("2.insert at the beginning\n");
181          printf("3.insert at the any positin\n ");
182          printf("4.insert at the end\n");
183          printf("5.delete at beginning\n");
184          printf("6.delete at end\n");
185          printf("7.delete at specific position\n");
186          printf("8.display all the elments\n");
187          printf("9.exit\n");
188          printf("Enter urs choice\n");
189          scanf("%d",&choice);
190          switch(choice)
191          {
192              case 1: printf("enter the number of nodes");
193                      scanf("%d",&n);
194                      createList(n);
195                      break;
196              case 2: printf("enter the data to insert");
197                      scanf("%d",&data);
198                      insertAtBeginning(data);
199                      break;
200              case 3:
201                  printf("Enter data and position: ");
202                  scanf("%d %d", &data, &pos);
203                  insertAtPosition(data, pos);
204                  break;
205              case 4:
206                  printf("Enter data to insert: ");
207                  scanf("%d", &data);
208                  insertAtEnd(data);
209                  break;
210              case 5:
211                  deleteAtBeginning();
212                  break;
213              case 6:
214                  deleteAtEnd();
215                  break;
216              case 7:
```

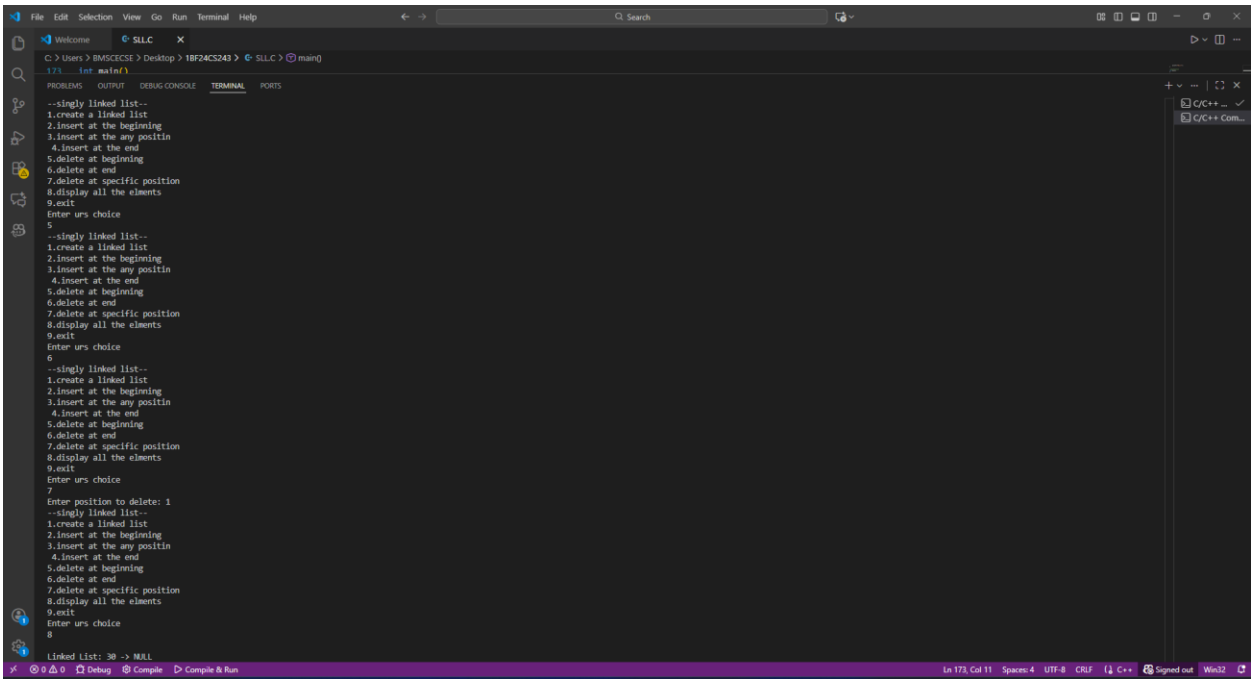


```
173  int main()
176      while(1)
188          printf("Enter urs choice\n");
189          scanf("%d",&choice);
190          switch(choice)
191          {
192              case 1: printf("enter the number of nodes");
193                      scanf("%d",&n);
194                      createList(n);
195                      break;
196              case 2: printf("enter the data to insert");
197                      scanf("%d",&data);
198                      insertAtBeginning(data);
199                      break;
200              case 3:
201                  printf("Enter data and position: ");
202                  scanf("%d %d", &data, &pos);
203                  insertAtPosition(data, pos);
204                  break;
205              case 4:
206                  printf("Enter data to insert: ");
207                  scanf("%d", &data);
208                  insertAtEnd(data);
209                  break;
210              case 5:
211                  deleteAtBeginning();
212                  break;
213              case 6:
214                  deleteAtEnd();
215                  break;
216              case 7:
217                  printf("Enter position to delete: ");
218                  scanf("%d", &pos);
219                  deleteAtPosition(pos);
220                  break;
221              case 8:
222                  displayList();
223                  break;
224              case 9:
225                  exit(0);
226              default:
227                  printf("Invalid choice\n");
228          }
229      }
230      return 0;
231  }
```

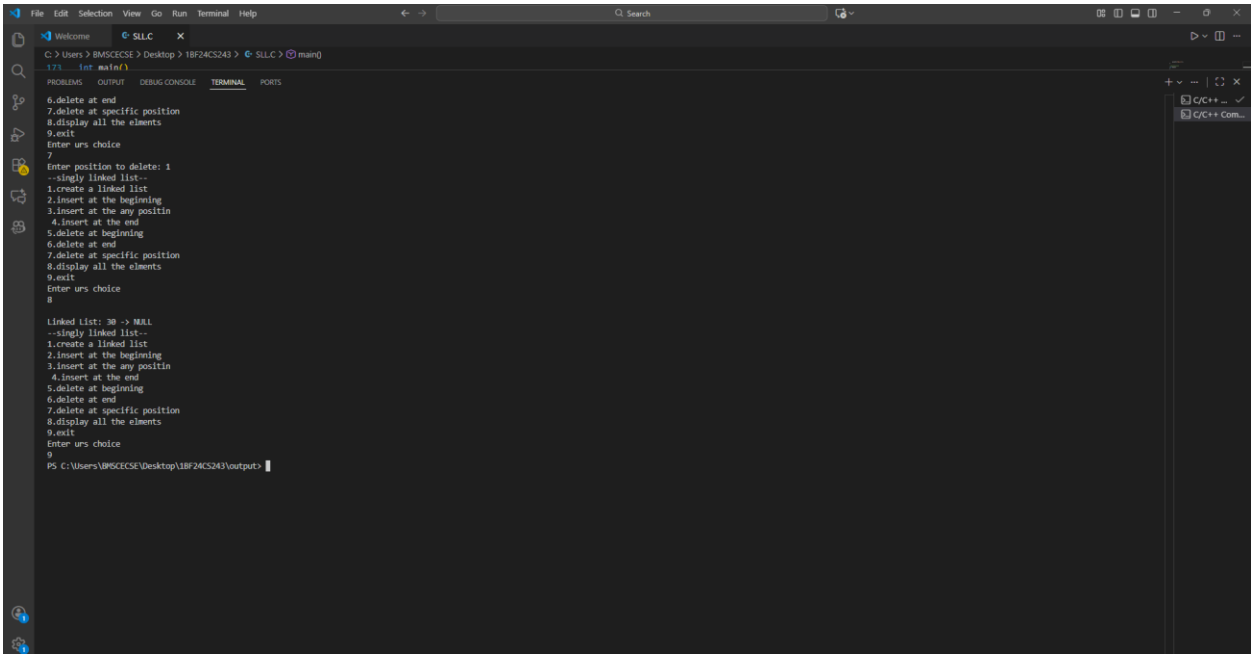
OUTPUT



```
PS C:\Users\BMSCECSE> cd "c:\Users\BMSCECSE\Desktop\18F24CS243\output"
PS C:\Users\BMSCECSE> cd "c:\Users\BMSCECSE\Desktop\18F24CS243\output"
PS C:\Users\BMSCECSE\Desktop\18F24CS243\output> & .\'SLL.exe'
--singly linked list--
PS C:\Users\BMSCECSE> cd "c:\Users\BMSCECSE\Desktop\18F24CS243\output"
PS C:\Users\BMSCECSE\Desktop\18F24CS243\output> & .\'SLL.exe'
PS C:\Users\BMSCECSE> cd "c:\Users\BMSCECSE\Desktop\18F24CS243\output"
PS C:\Users\BMSCECSE\Desktop\18F24CS243\output> & .\'SLL.exe'
--singly linked list--
1.create a linked list
PS C:\Users\BMSCECSE> cd "c:\Users\BMSCECSE\Desktop\18F24CS243\output"
PS C:\Users\BMSCECSE> cd "c:\Users\BMSCECSE\Desktop\18F24CS243\output"
PS C:\Users\BMSCECSE\Desktop\18F24CS243\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 beginning
6.delete at end
7.delete at specific position
8.display all the elments
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 elments
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
```



```
--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 elments
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 elments
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 elments
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 elments
9.exit
Enter urs choice
8

Linked List: 30 -> NULL
```


File Edit Selection View Go Run Terminal Help
Search
Welcome
SLLC
C: > Users > BMSCECSE > Desktop > 1BF24CS243 > SLLC > main()
173 int main()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
C/C++ ...
C/C++ Com...
6.delete at end
7.delete at specific position
8.display all the elments
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 elments
9.exit
Enter urs 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 elments
9.exit
Enter urs choice
9
PS C:\Users\BMSCECSE\Desktop\1BF24CS243\output>