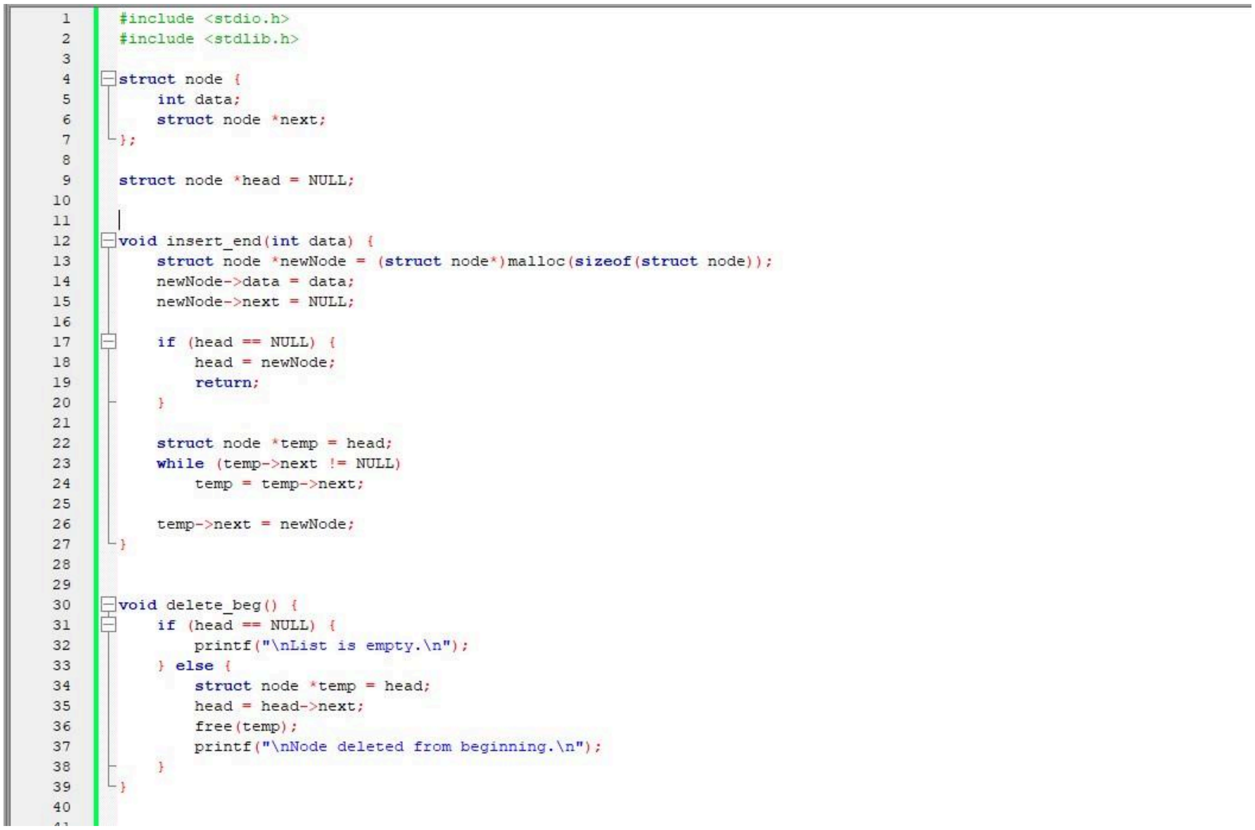
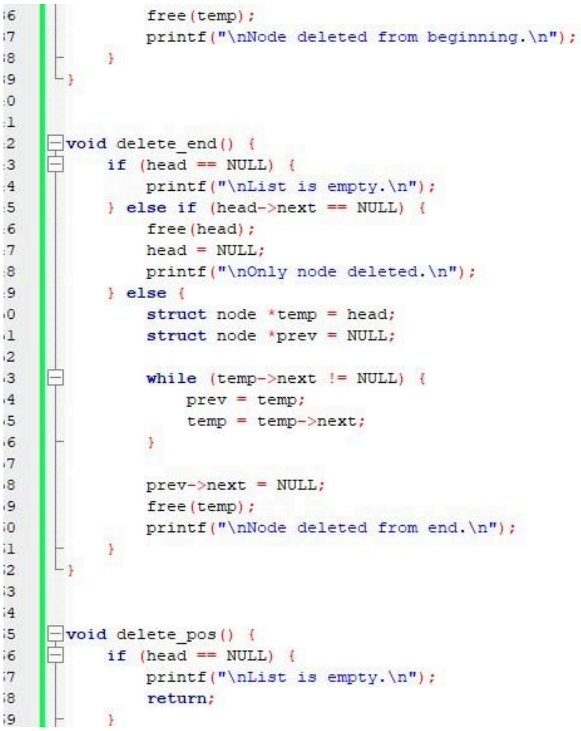
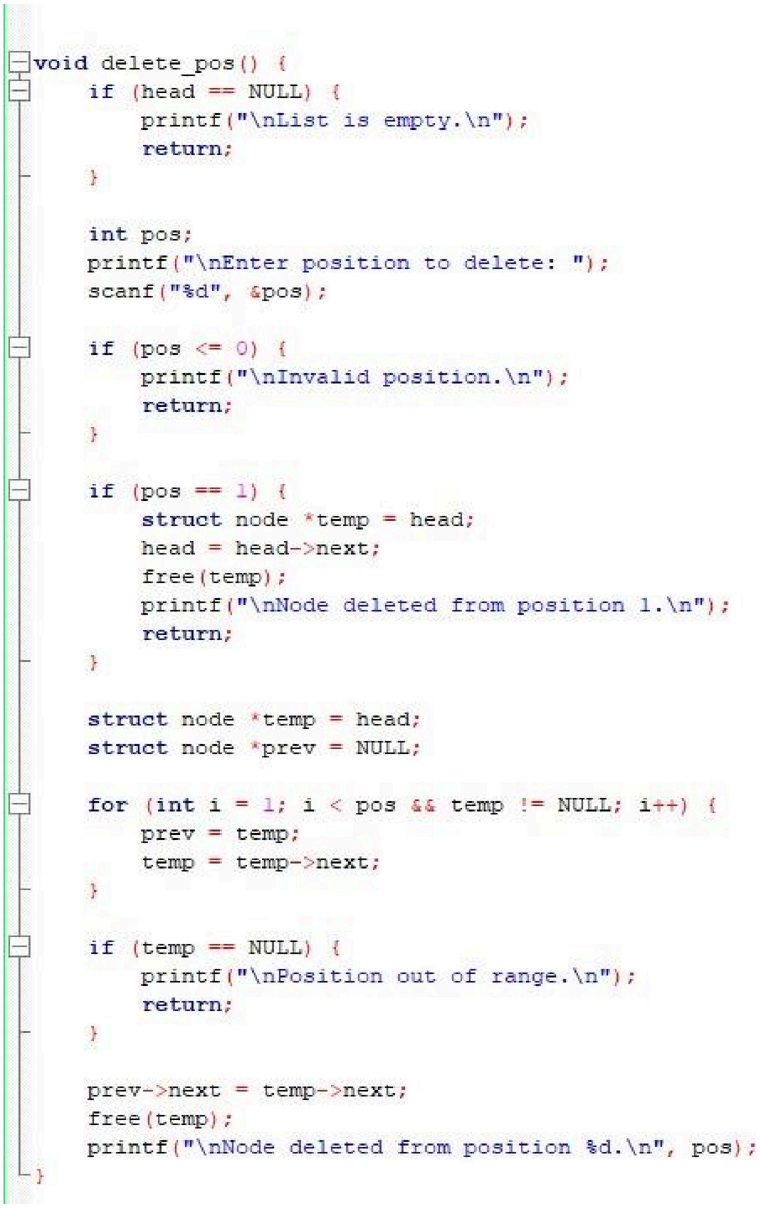


```c
#include <stdio.h>
#include <stdlib.h>

struct node {
    int data;
    struct node *next;
};

struct node *head = NULL;


void insert_end(int data) {
    struct node *newNode = (struct node*)malloc(sizeof(struct node));
    newNode->data = data;
    newNode->next = NULL;

    if (head == NULL) {
        head = newNode;
        return;
    }

    struct node *temp = head;
    while (temp->next != NULL)
        temp = temp->next;

    temp->next = newNode;
}


void delete_beg() {
    if (head == NULL) {
        printf("\nList is empty.\n");
    } else {
        struct node *temp = head;
        head = head->next;
        free(temp);
        printf("\nNode deleted from beginning.\n");
    }
}

```

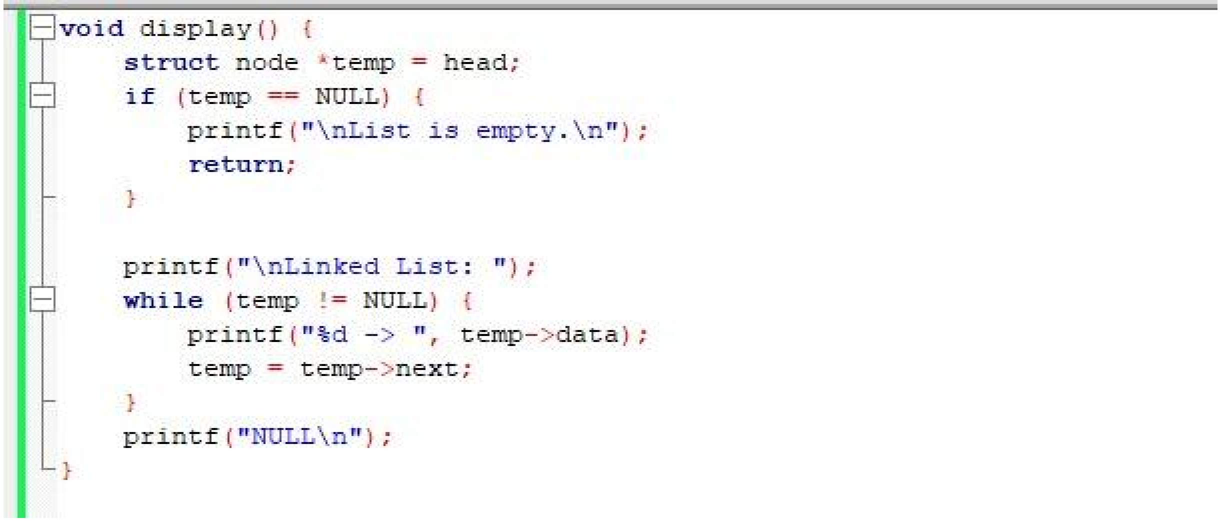


```c
        free(temp);
        printf("\nNode deleted from beginning.\n");
    }
}
```

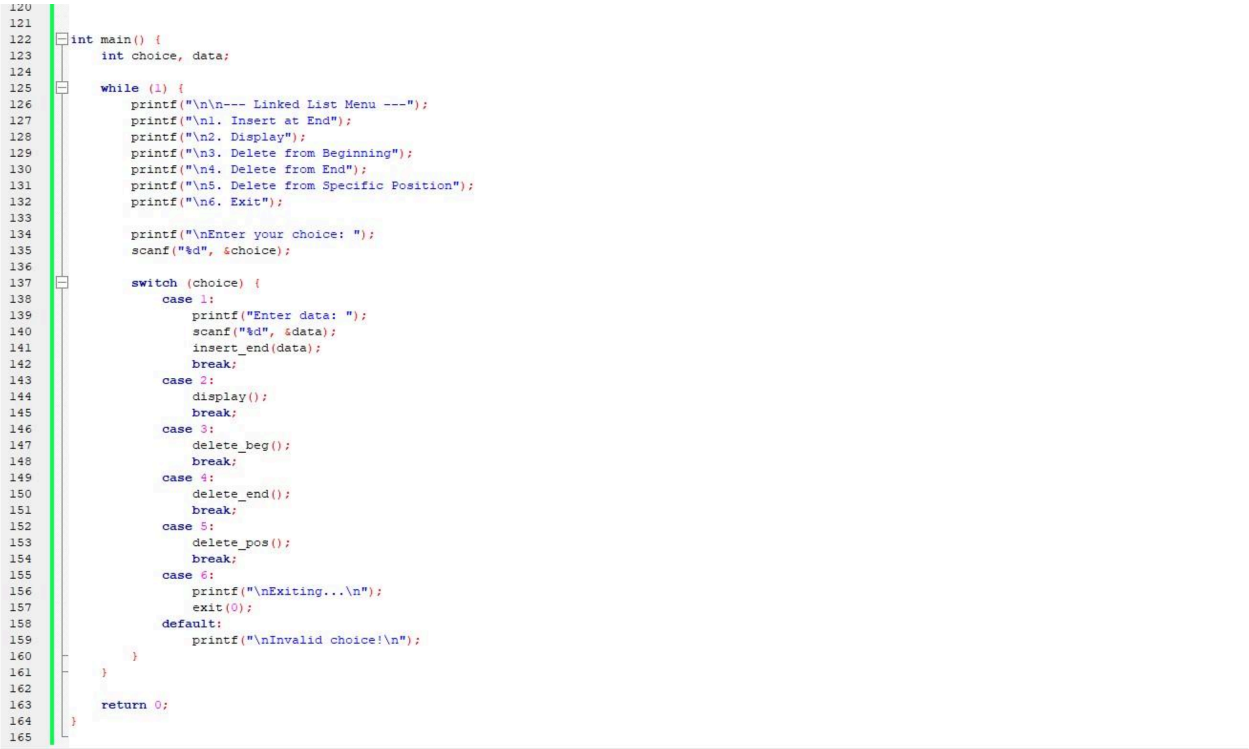
```c


void delete_end() {
    if (head == NULL) {
        printf("\nList is empty.\n");
    } else if (head->next == NULL) {
        free(head);
        head = NULL;
        printf("\nOnly node deleted.\n");
    } else {
        struct node *temp = head;
        struct node *prev = NULL;

        while (temp->next != NULL) {
            prev = temp;
            temp = temp->next;
        }

        prev->next = NULL;
        free(temp);
        printf("\nNode deleted from end.\n");
    }
}


void delete_pos() {
    if (head == NULL) {
        printf("\nList is empty.\n");
        return;
    }
```



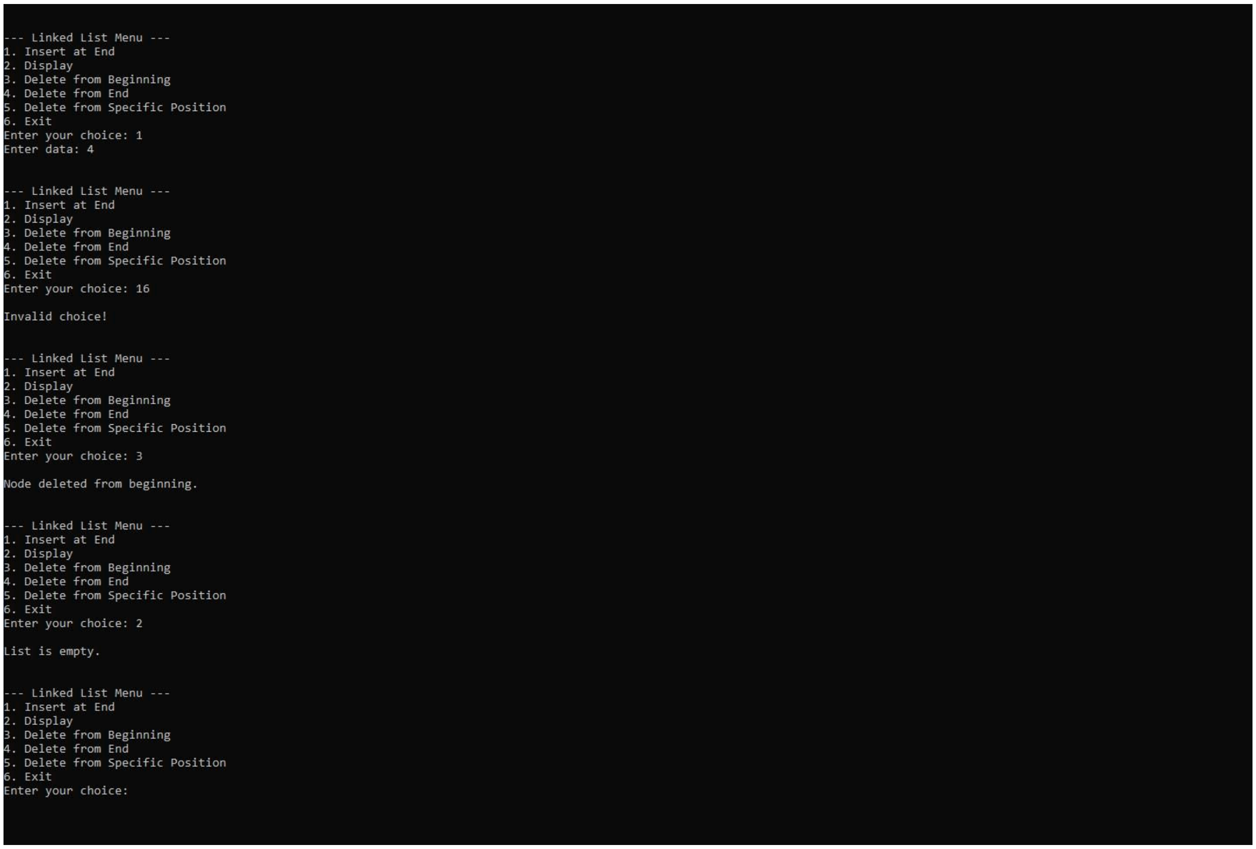
```c
void delete_pos() {
    if (head == NULL) {
        printf("\nList is empty.\n");
        return;
    }

    int pos;
    printf("\nEnter position to delete: ");
    scanf("%d", &pos);

    if (pos <= 0) {
        printf("\nInvalid position.\n");
        return;
    }

    if (pos == 1) {
        struct node *temp = head;
        head = head->next;
        free(temp);
        printf("\nNode deleted from position 1.\n");
        return;
    }

    struct node *temp = head;
    struct node *prev = NULL;

    for (int i = 1; i < pos && temp != NULL; i++) {
        prev = temp;
        temp = temp->next;
    }

    if (temp == NULL) {
        printf("\nPosition out of range.\n");
        return;
    }

    prev->next = temp->next;
    free(temp);
    printf("\nNode deleted from position %d.\n", pos);
}
```



```
void display() {
    struct node *temp = head;
    if (temp == NULL) {
        printf("\nList is empty.\n");
        return;
    }

    printf("\nLinked List: ");
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}
```



```c

int main() {
    int choice, data;

    while (1) {
        printf("\n\n--- Linked List Menu ---");
        printf("\n1. Insert at End");
        printf("\n2. Display");
        printf("\n3. Delete from Beginning");
        printf("\n4. Delete from End");
        printf("\n5. Delete from Specific Position");
        printf("\n6. Exit");

        printf("\nEnter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter data: ");
                scanf("%d", &data);
                insert_end(data);
                break;
            case 2:
                display();
                break;
            case 3:
                delete_beg();
                break;
            case 4:
                delete_end();
                break;
            case 5:
                delete_pos();
                break;
            case 6:
                printf("\nExiting...\n");
                exit(0);
            default:
                printf("\nInvalid choice!\n");
        }
    }

    return 0;
}
```



```
--- Linked List Menu ---
1. Insert at End
2. Display
3. Delete from Beginning
4. Delete from End
5. Delete from Specific Position
6. Exit
Enter your choice: 1
Enter data: 4


--- Linked List Menu ---
1. Insert at End
2. Display
3. Delete from Beginning
4. Delete from End
5. Delete from Specific Position
6. Exit
Enter your choice: 16

Invalid choice!


--- Linked List Menu ---
1. Insert at End
2. Display
3. Delete from Beginning
4. Delete from End
5. Delete from Specific Position
6. Exit
Enter your choice: 3

Node deleted from beginning.


--- Linked List Menu ---
1. Insert at End
2. Display
3. Delete from Beginning
4. Delete from End
5. Delete from Specific Position
6. Exit
Enter your choice: 2

List is empty.


--- Linked List Menu ---
1. Insert at End
2. Display
3. Delete from Beginning
4. Delete from End
5. Delete from Specific Position
6. Exit
Enter your choice:
```