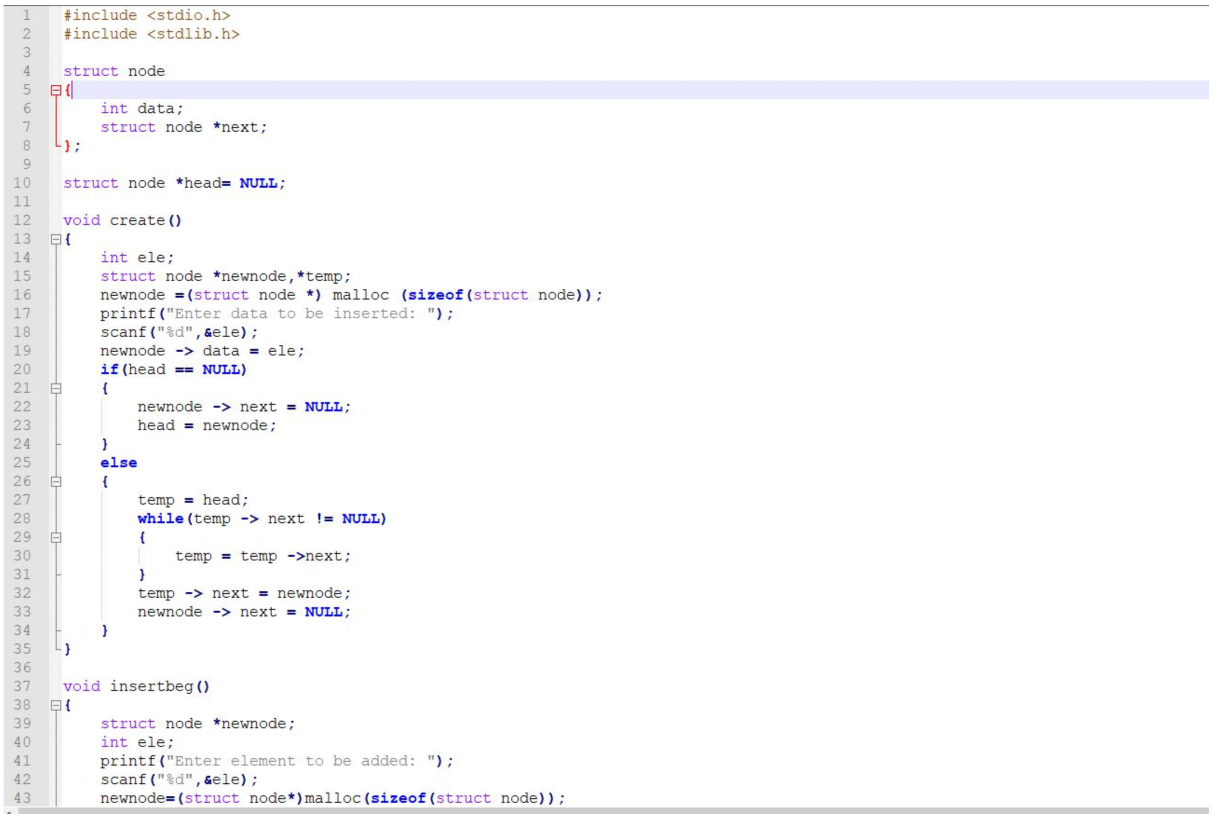
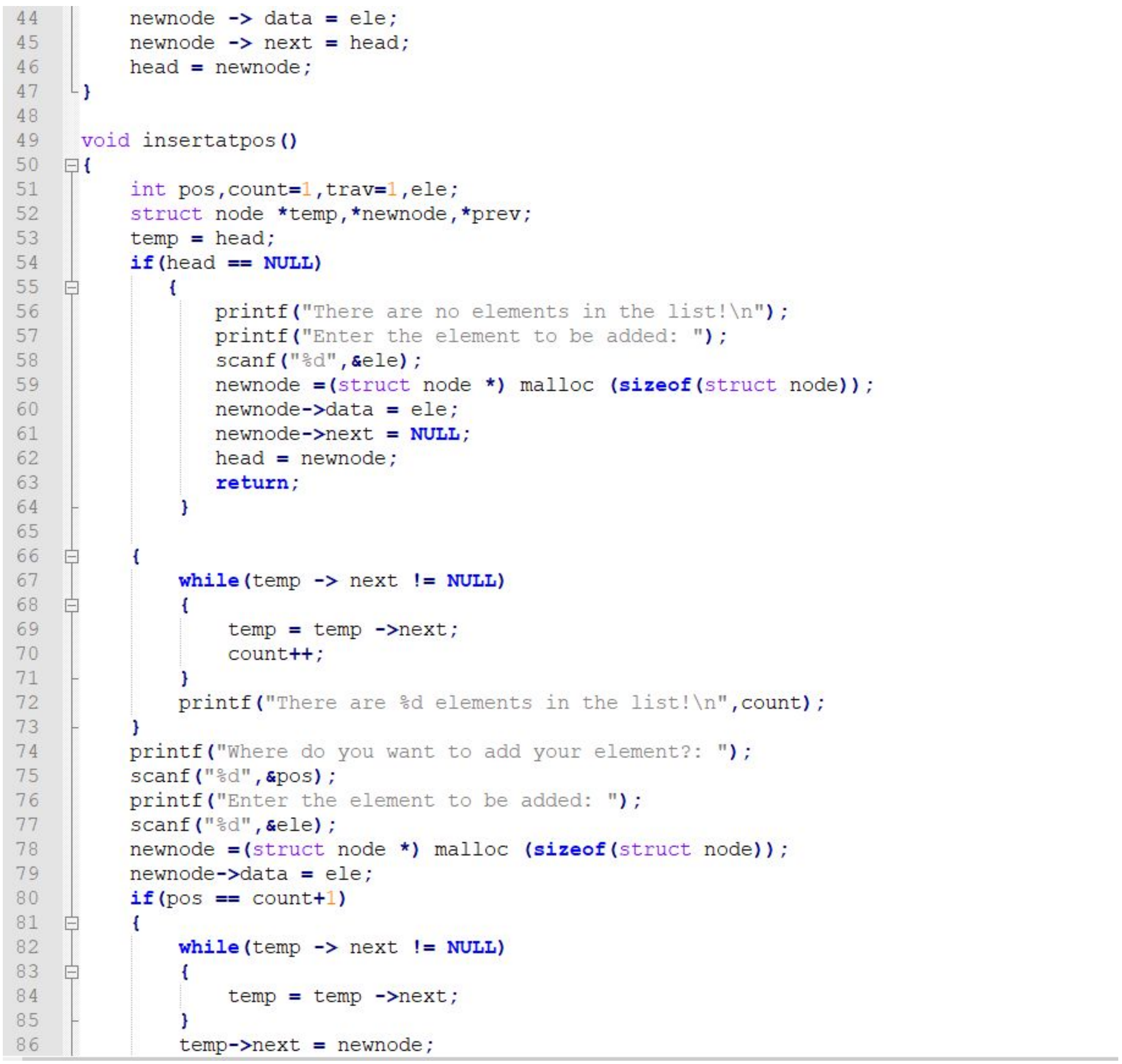
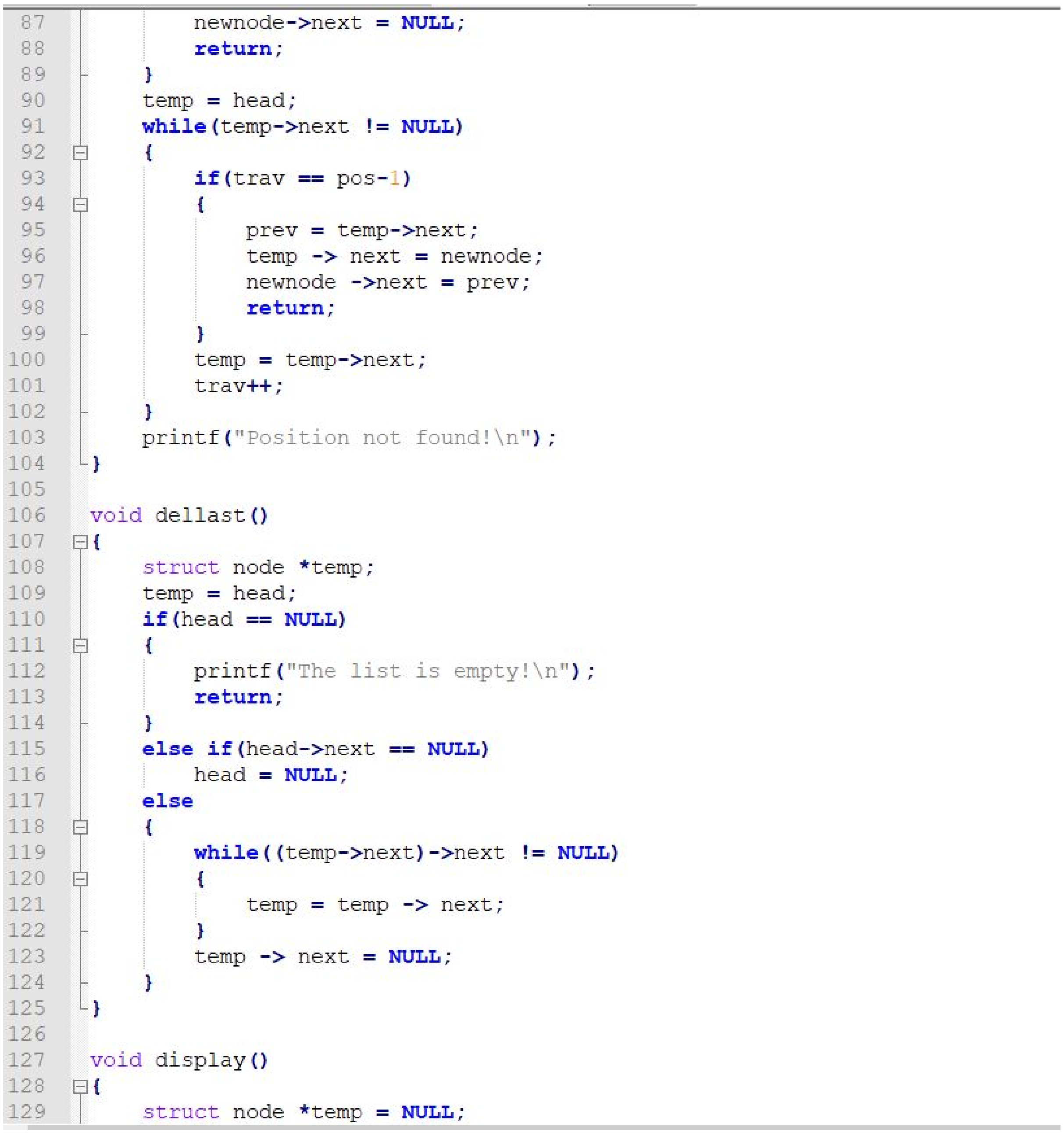


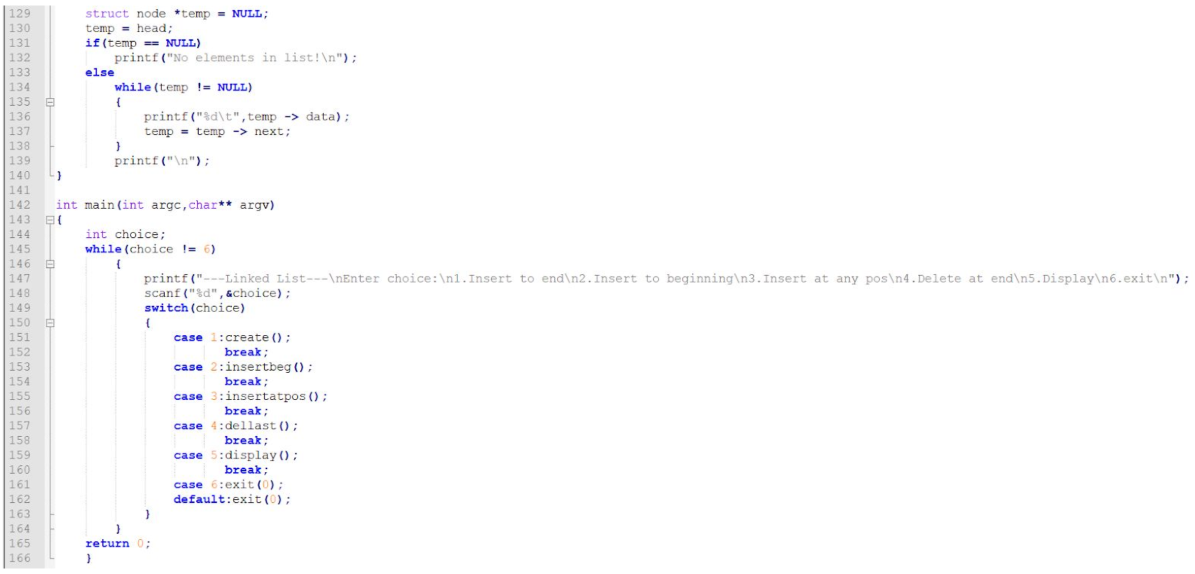
```c
#include <stdio.h>
#include <stdlib.h>

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

struct node *head= NULL;

void create()
{
    int ele;
    struct node *newnode,*temp;
    newnode =(struct node *) malloc (sizeof(struct node));
    printf("Enter data to be inserted: ");
    scanf("%d",&ele);
    newnode -> data = ele;
    if(head == NULL)
    {
        newnode -> next = NULL;
        head = newnode;
    }
    else
    {
        temp = head;
        while(temp -> next != NULL)
        {
            temp = temp ->next;
        }
        temp -> next = newnode;
        newnode -> next = NULL;
    }
}

void insertbeg()
{
    struct node *newnode;
    int ele;
    printf("Enter element to be added: ");
    scanf("%d",&ele);
    newnode=(struct node*)malloc(sizeof(struct node));
```



```c
    newnode -> data = ele;
    newnode -> next = head;
    head = newnode;
}

void insertatpos()
{
    int pos,count=1,trav=1,ele;
    struct node *temp,*newnode,*prev;
    temp = head;
    if(head == NULL)
       {
           printf("There are no elements in the list!\n");
           printf("Enter the element to be added: ");
           scanf("%d",&ele);
           newnode =(struct node *) malloc (sizeof(struct node));
           newnode->data = ele;
           newnode->next = NULL;
           head = newnode;
           return;
        }

    {
        while(temp -> next != NULL)
        {
            temp = temp ->next;
            count++;
        }
        printf("There are %d elements in the list!\n",count);
    }
    printf("Where do you want to add your element?: ");
    scanf("%d",&pos);
    printf("Enter the element to be added: ");
    scanf("%d",&ele);
    newnode =(struct node *) malloc (sizeof(struct node));
    newnode->data = ele;
    if(pos == count+1)
    {
        while(temp -> next != NULL)
        {
            temp = temp ->next;
        }
        temp->next = newnode;
```



```c
        newnode->next = NULL;
        return;
    }
    temp = head;
    while(temp->next != NULL)
    {
        if(trav == pos-1)
        {
            prev = temp->next;
            temp -> next = newnode;
            newnode ->next = prev;
            return;
        }
        temp = temp->next;
        trav++;
    }
    printf("Position not found!\n");
}

void dellast()
{
    struct node *temp;
    temp = head;
    if(head == NULL)
    {
        printf("The list is empty!\n");
        return;
    }
    else if(head->next == NULL)
        head = NULL;
    else
    {
        while((temp->next)->next != NULL)
        {
            temp = temp -> next;
        }
        temp -> next = NULL;
    }
}

void display()
{
    struct node *temp = NULL;
```



```c
    struct node *temp = NULL;
    temp = head;
    if(temp == NULL)
        printf("No elements in list!\n");
    else
        while(temp != NULL)
        {
            printf("%d\t",temp -> data);
            temp = temp -> next;
        }
        printf("\n");
}

int main(int argc,char** argv)
{
    int choice;
    while(choice != 6)
        {
            printf("---Linked List---\nEnter choice:\n1.Insert to end\n2.Insert to beginning\n3.Insert at any pos\n4.Delete at end\n5.Display\n6.exit\n");
            scanf("%d",&choice);
            switch(choice)
            {
                case 1:create();
                       break;
                case 2:insertbeg();
                       break;
                case 3:insertatpos();
                       break;
                case 4:dellast();
                       break;
                case 5:display();
                       break;
                case 6:exit(0);
                default:exit(0);
            }
        }
    return 0;
    }
```