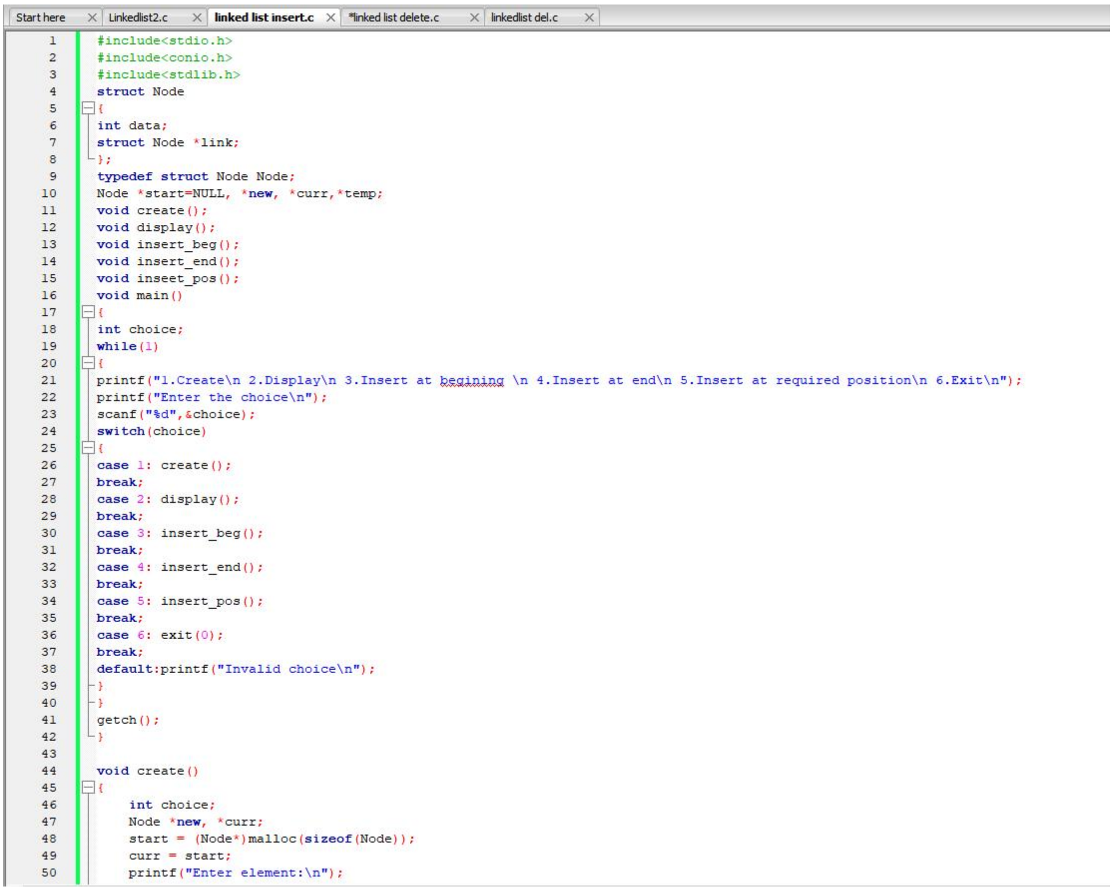
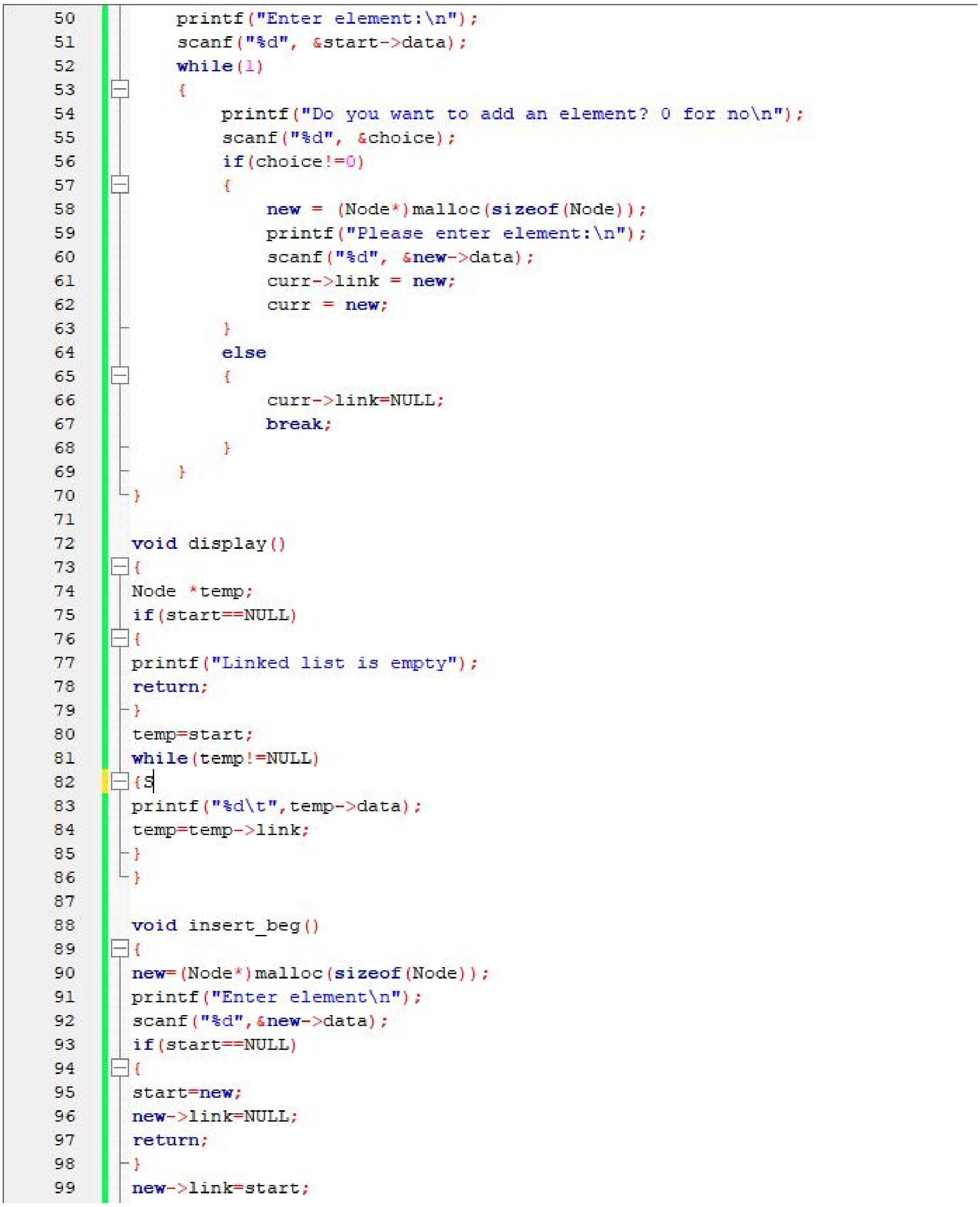
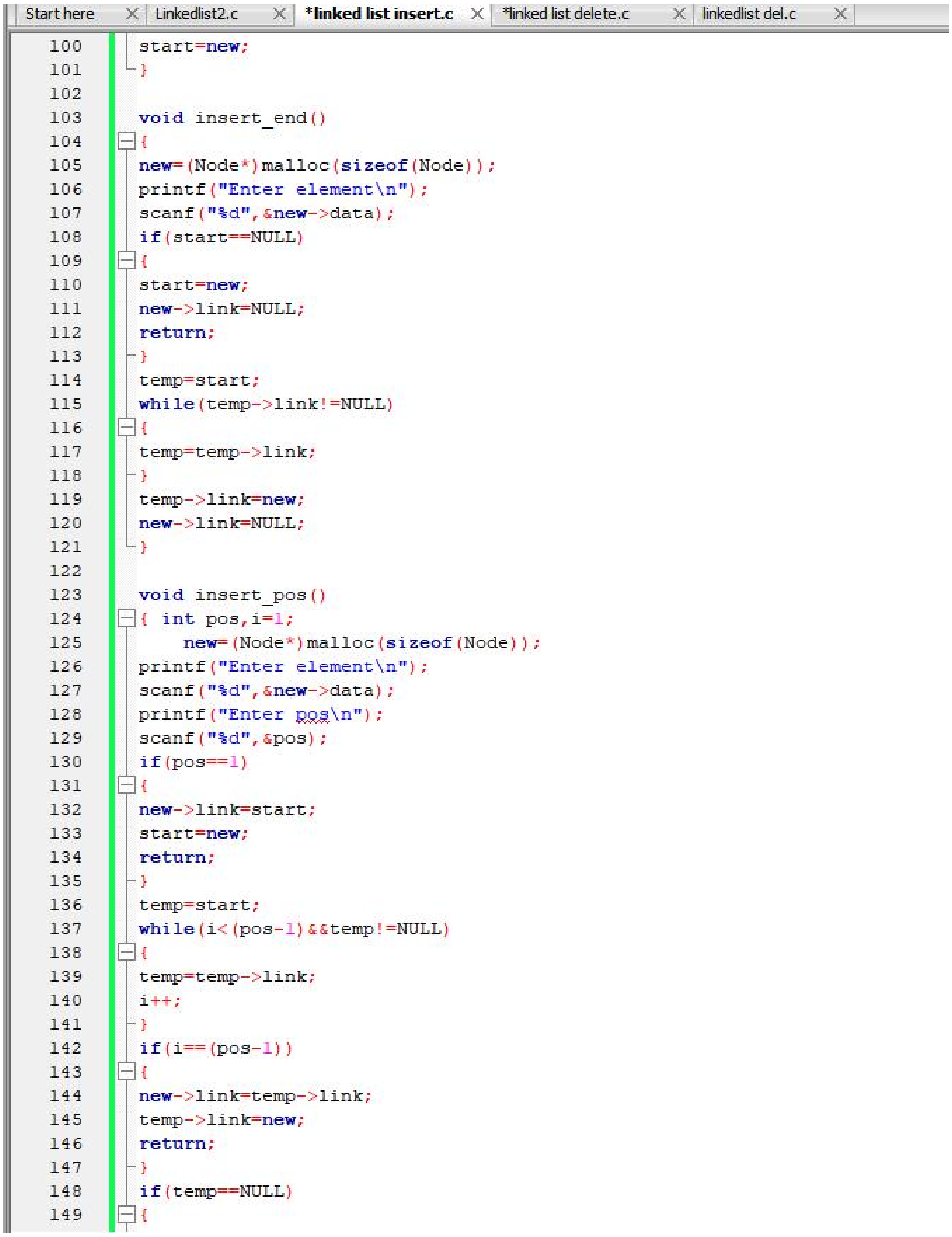
PROGRAM 1



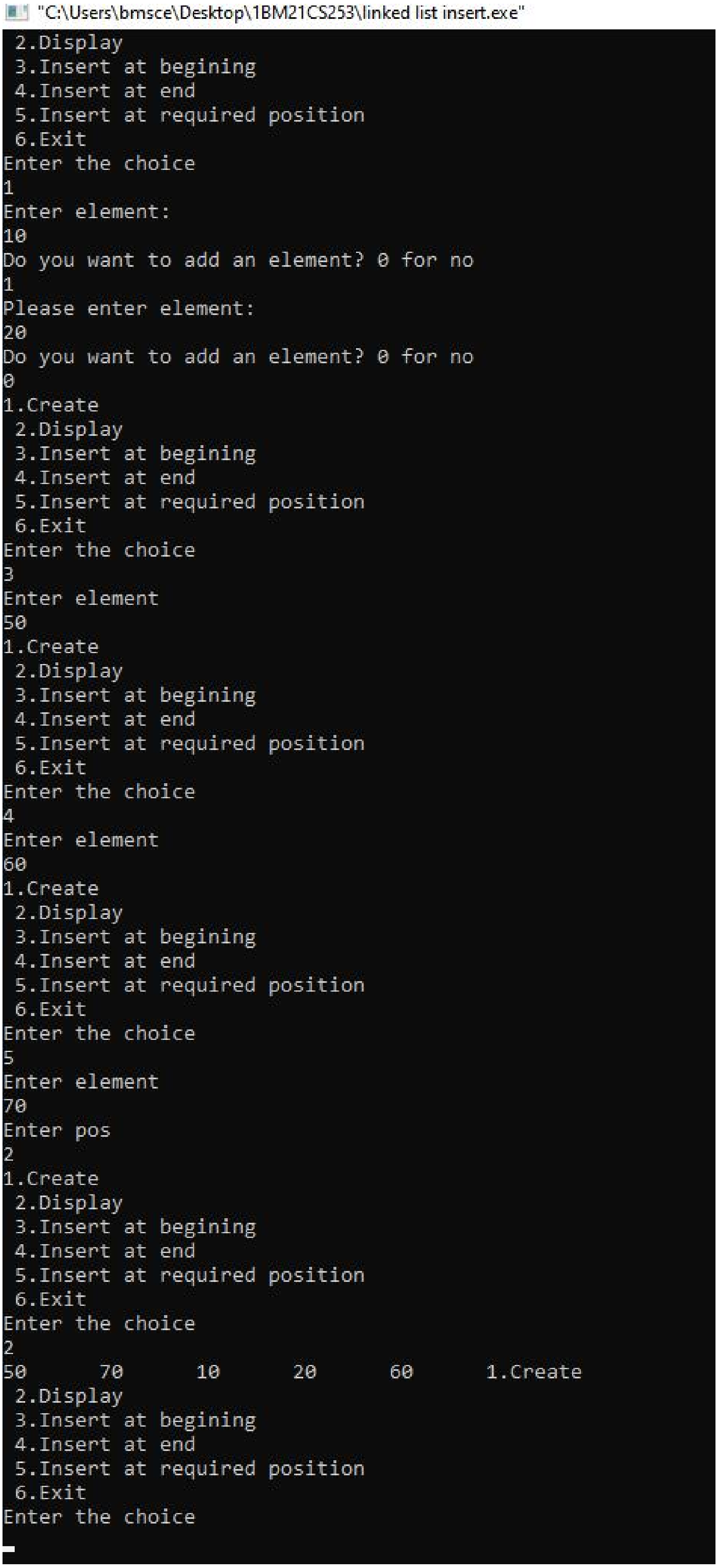
```c
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct Node
{
int data;
struct Node *link;
};
typedef struct Node Node;
Node *start=NULL, *new, *curr,*temp;
void create();
void display();
void insert_beg();
void insert_end();
void inseet_pos();
void main()
{
int choice;
while(1)
{
printf("1.Create\n 2.Display\n 3.Insert at begining \n 4.Insert at end\n 5.Insert at required position\n 6.Exit\n");
printf("Enter the choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1: create();
break;
case 2: display();
break;
case 3: insert_beg();
break;
case 4: insert_end();
break;
case 5: insert_pos();
break;
case 6: exit(0);
break;
default:printf("Invalid choice\n");
}
}
getch();
}

void create()
{
    int choice;
    Node *new, *curr;
    start = (Node*)malloc(sizeof(Node));
    curr = start;
    printf("Enter element:\n");
```



```c
    printf("Enter element:\n");
    scanf("%d", &start->data);
    while(1)
    {
        printf("Do you want to add an element? 0 for no\n");
        scanf("%d", &choice);
        if(choice!=0)
        {
            new = (Node*)malloc(sizeof(Node));
            printf("Please enter element:\n");
            scanf("%d", &new->data);
            curr->link = new;
            curr = new;
        }
        else
        {
            curr->link=NULL;
            break;
        }
    }
}

void display()
{
Node *temp;
if(start==NULL)
{
printf("Linked list is empty");
return;
}
temp=start;
while(temp!=NULL)
{S
printf("%d\t",temp->data);
temp=temp->link;
}
}

void insert_beg()
{
new=(Node*)malloc(sizeof(Node));
printf("Enter element\n");
scanf("%d",&new->data);
if(start==NULL)
{
start=new;
new->link=NULL;
return;
}
new->link=start;
```



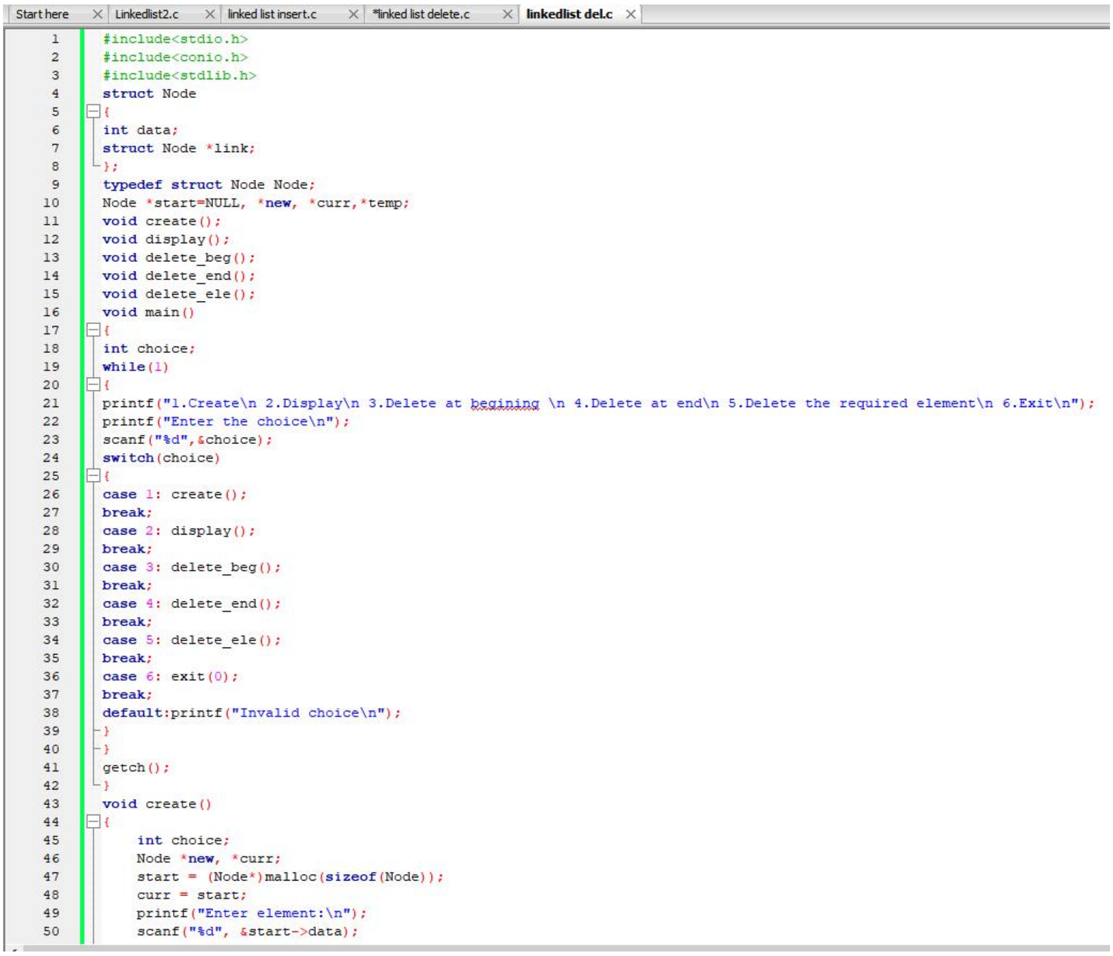
```
start=new;
}

void insert_end()
{
new=(Node*)malloc(sizeof(Node));
printf("Enter element\n");
scanf("%d",&new->data);
if(start==NULL)
{
start=new;
new->link=NULL;
return;
}
temp=start;
while(temp->link!=NULL)
{
temp=temp->link;
}
temp->link=new;
new->link=NULL;
}

void insert_pos()
{ int pos,i=1;
    new=(Node*)malloc(sizeof(Node));
printf("Enter element\n");
scanf("%d",&new->data);
printf("Enter pos\n");
scanf("%d",&pos);
if(pos==1)
{
new->link=start;
start=new;
return;
}
temp=start;
while(i<(pos-1)&&temp!=NULL)
{
temp=temp->link;
i++;
}
if(i==(pos-1))
{
new->link=temp->link;
temp->link=new;
return;
}
if(temp==NULL)
{
```



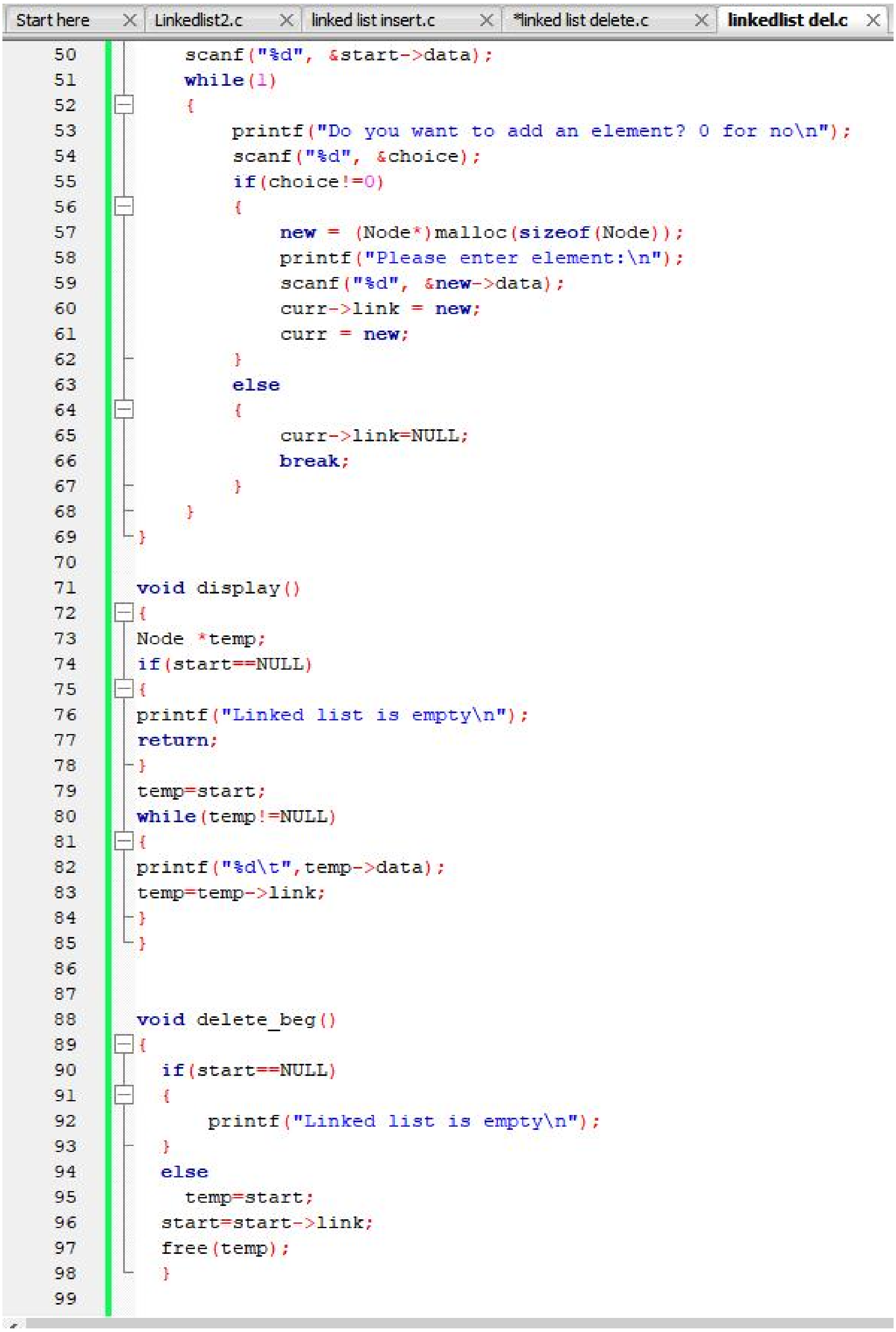
"C:\Users\bmsce\Desktop\1BM21CS253\linked list insert.exe"

```
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at required position
 6.Exit
Enter the choice
1
Enter element:
10
Do you want to add an element? 0 for no
1
Please enter element:
20
Do you want to add an element? 0 for no
0
1.Create
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at required position
 6.Exit
Enter the choice
3
Enter element
50
1.Create
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at required position
 6.Exit
Enter the choice
4
Enter element
60
1.Create
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at required position
 6.Exit
Enter the choice
5
Enter element
70
Enter pos
2
1.Create
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at required position
 6.Exit
Enter the choice
2
50      70      10      20      60      1.Create
 2.Display
 3.Insert at begining
 4.Insert at end
 5.Insert at required position
 6.Exit
Enter the choice
```

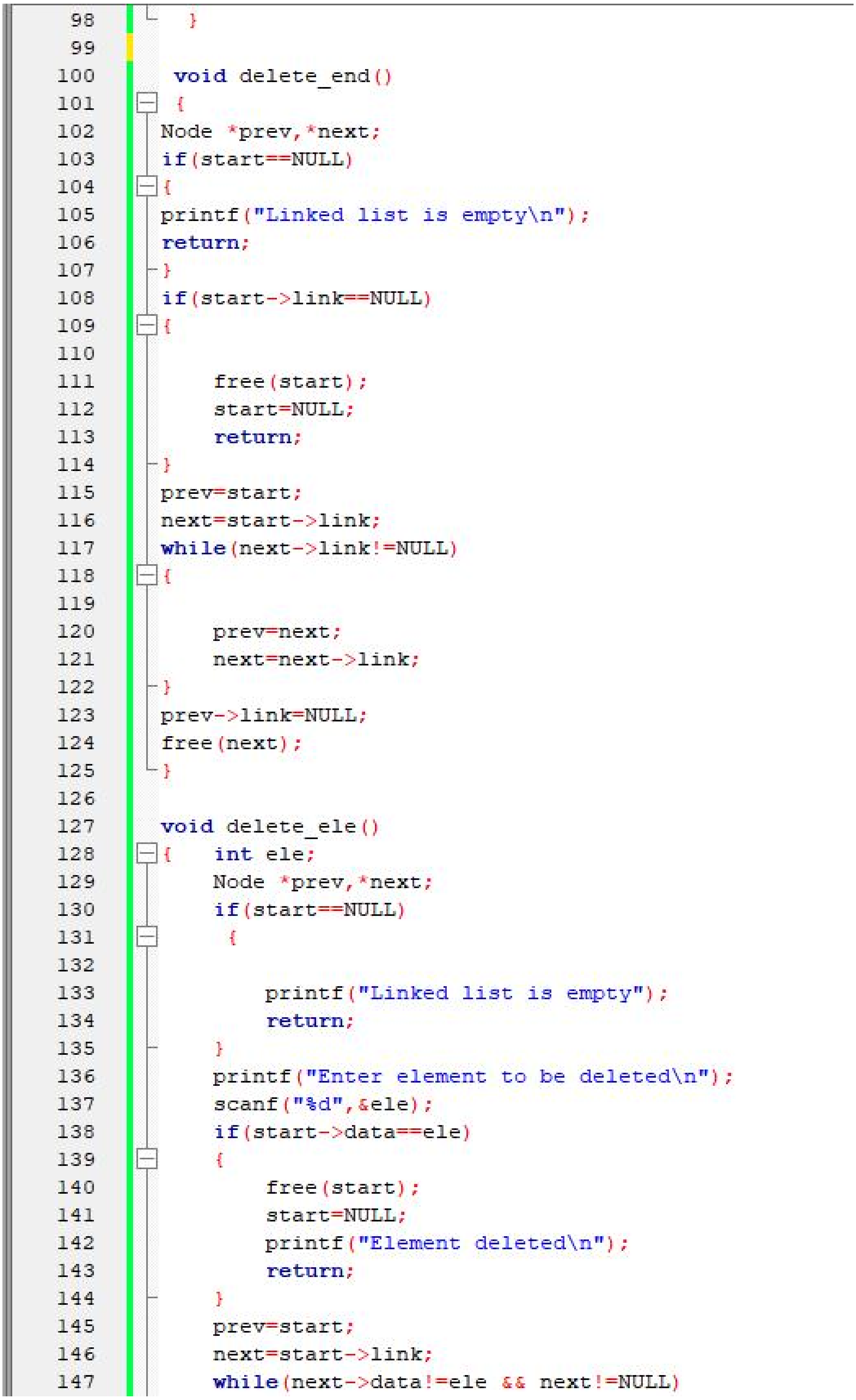
Program 2



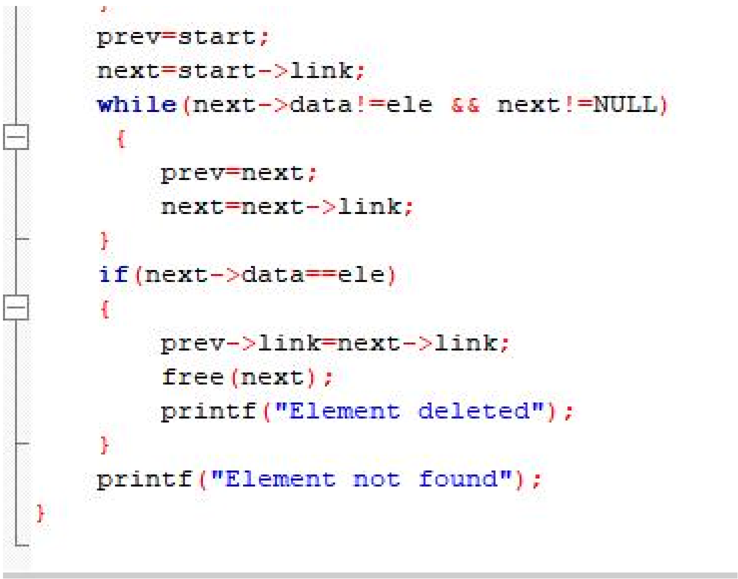
```c
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct Node
{
int data;
struct Node *link;
};
typedef struct Node Node;
Node *start=NULL, *new, *curr,*temp;
void create();
void display();
void delete_beg();
void delete_end();
void delete_ele();
void main()
{
int choice;
while(1)
{
printf("1.Create\n 2.Display\n 3.Delete at begining \n 4.Delete at end\n 5.Delete the required element\n 6.Exit\n");
printf("Enter the choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1: create();
break;
case 2: display();
break;
case 3: delete_beg();
break;
case 4: delete_end();
break;
case 5: delete_ele();
break;
case 6: exit(0);
break;
default:printf("Invalid choice\n");
}
}
getch();
}
void create()
{
    int choice;
    Node *new, *curr;
    start = (Node*)malloc(sizeof(Node));
    curr = start;
    printf("Enter element:\n");
    scanf("%d", &start->data);
```



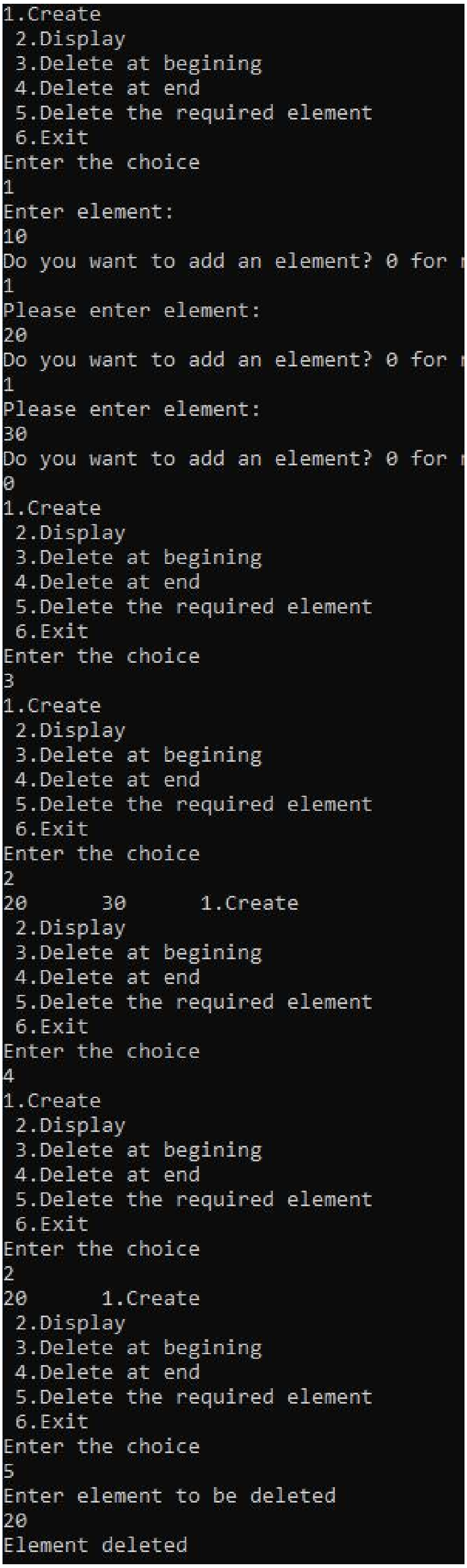
```
    scanf("%d", &start->data);
    while(1)
    {
        printf("Do you want to add an element? 0 for no\n");
        scanf("%d", &choice);
        if(choice!=0)
        {
            new = (Node*)malloc(sizeof(Node));
            printf("Please enter element:\n");
            scanf("%d", &new->data);
            curr->link = new;
            curr = new;
        }
        else
        {
            curr->link=NULL;
            break;
        }
    }
}

void display()
{
Node *temp;
if(start==NULL)
{
printf("Linked list is empty\n");
return;
}
temp=start;
while(temp!=NULL)
{
printf("%d\t",temp->data);
temp=temp->link;
}
}


void delete_beg()
{
  if(start==NULL)
  {
      printf("Linked list is empty\n");
  }
  else
    temp=start;
  start=start->link;
  free(temp);
  }

```



```c
  }

 void delete_end()
 {
Node *prev,*next;
if(start==NULL)
{
printf("Linked list is empty\n");
return;
}
if(start->link==NULL)
{

    free(start);
    start=NULL;
    return;
}
prev=start;
next=start->link;
while(next->link!=NULL)
{

    prev=next;
    next=next->link;
}
prev->link=NULL;
free(next);
}

void delete_ele()
{    int ele;
    Node *prev,*next;
    if(start==NULL)
     {

        printf("Linked list is empty");
        return;
    }
    printf("Enter element to be deleted\n");
    scanf("%d",&ele);
    if(start->data==ele)
    {
        free(start);
        start=NULL;
        printf("Element deleted\n");
        return;
    }
    prev=start;
    next=start->link;
    while(next->data!=ele && next!=NULL)
```



```
    prev=start;
    next=start->link;
    while(next->data!=ele && next!=NULL)
     {
        prev=next;
        next=next->link;
    }
    if(next->data==ele)
    {
        prev->link=next->link;
        free(next);
        printf("Element deleted");
    }
    printf("Element not found");
}
```



```
1.Create
 2.Display
 3.Delete at begining
 4.Delete at end
 5.Delete the required element
 6.Exit
Enter the choice
1
Enter element:
10
Do you want to add an element? 0 for
1
Please enter element:
20
Do you want to add an element? 0 for
1
Please enter element:
30
Do you want to add an element? 0 for
0
1.Create
 2.Display
 3.Delete at begining
 4.Delete at end
 5.Delete the required element
 6.Exit
Enter the choice
3
1.Create
 2.Display
 3.Delete at begining
 4.Delete at end
 5.Delete the required element
 6.Exit
Enter the choice
2
20      30      1.Create
 2.Display
 3.Delete at begining
 4.Delete at end
 5.Delete the required element
 6.Exit
Enter the choice
4
1.Create
 2.Display
 3.Delete at begining
 4.Delete at end
 5.Delete the required element
 6.Exit
Enter the choice
2
20      1.Create
 2.Display
 3.Delete at begining
 4.Delete at end
 5.Delete the required element
 6.Exit
Enter the choice
5
Enter element to be deleted
20
Element deleted
```