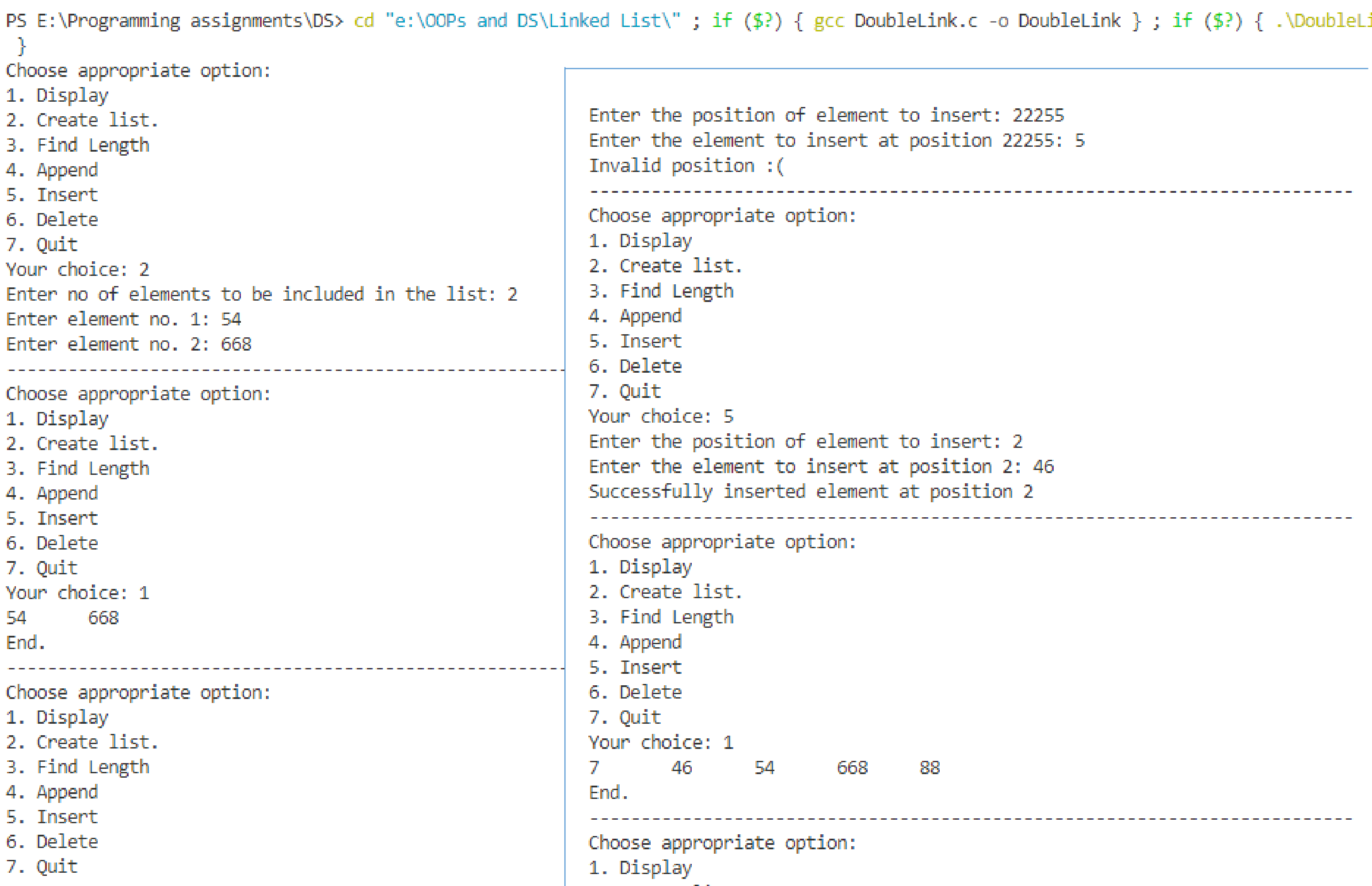
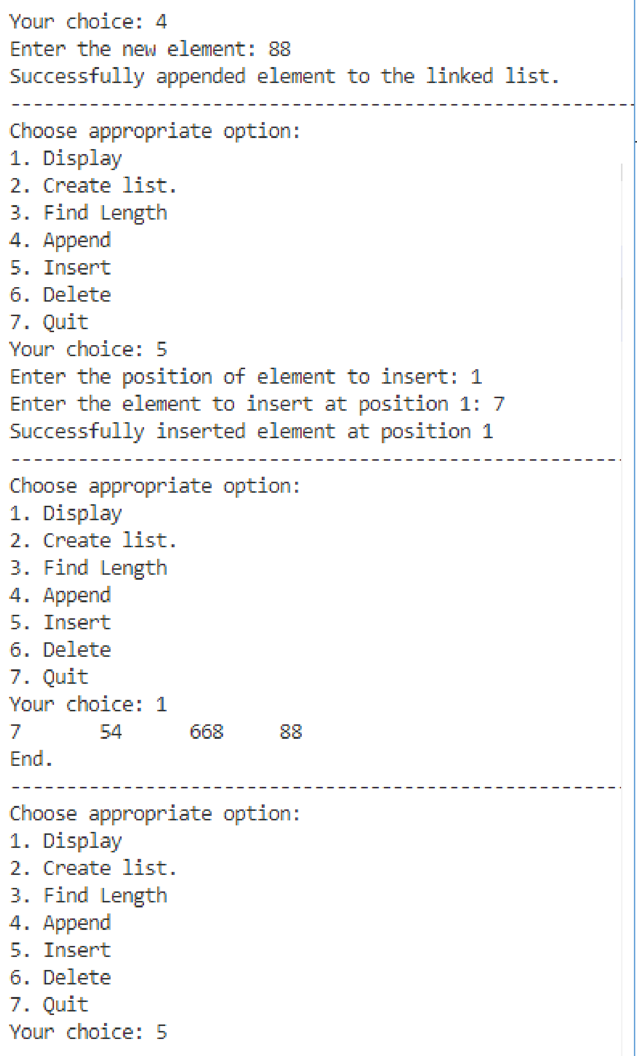
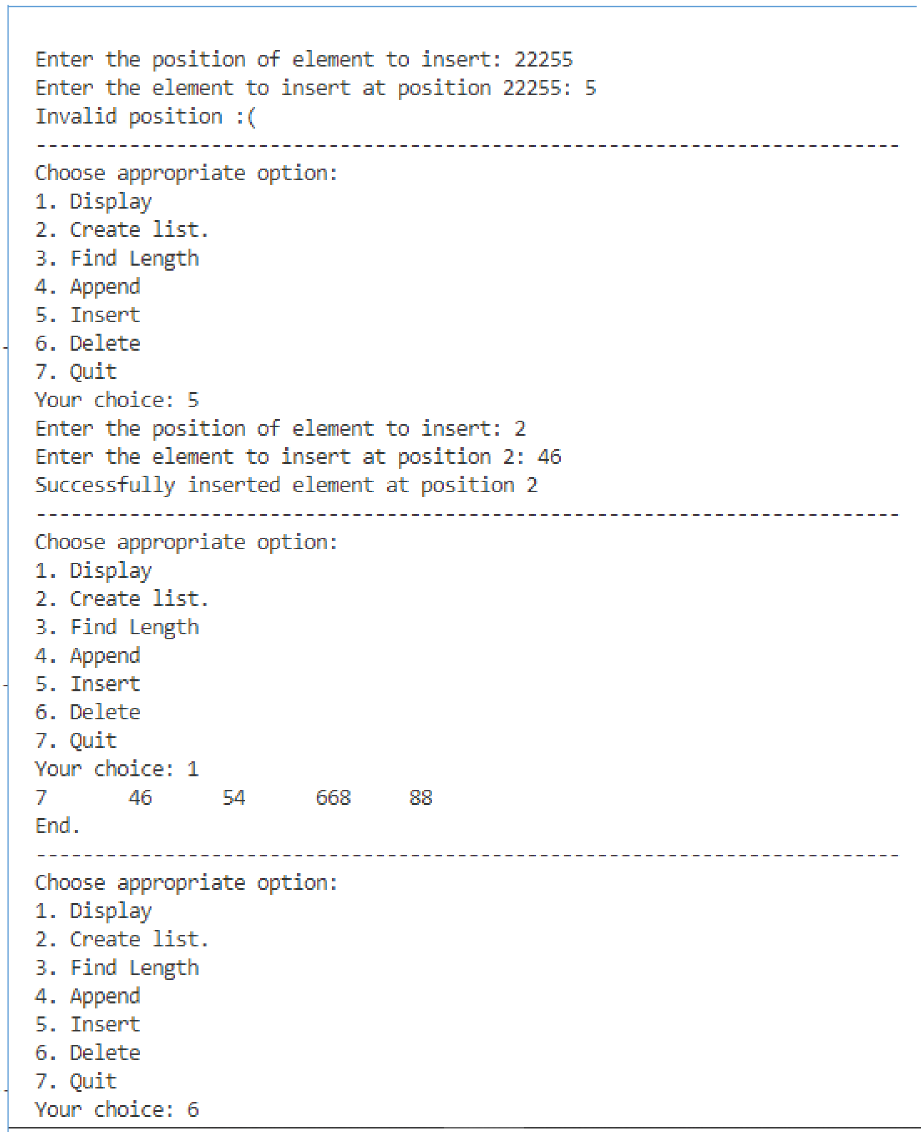
```c
        switch (ch) {
            case 1:
                display();
                break;
            case 2:
                create();
                break;
            case 3:
                printf("\nLength is %d.\n",length());
                break;
            case 4:
                printf("Enter the new element: ");
                scanf("%d",&el);
                append(el);
                printf("Successfully appended element to the linked list.\n");
                break;
            case 5:
                printf("Enter the position of element to insert: ");
                scanf("%d",&pos);
                printf("Enter the element to insert at position %d: ", pos);
                scanf("%d",&el);
                insert(pos, el);
                break;
            case 6:
                printf("Enter the position of element you want to delete: ");
                scanf("%d",&pos);
                delete(pos);
                break;
            case 7:
                printf("Bye\n");
                return;
                break;
            default:
                printf("Invalid choice !  Enter a valid choice.\n");
                break;
        }
        printf("--------------------------------------------------------------------
------\n");
    }
}
```



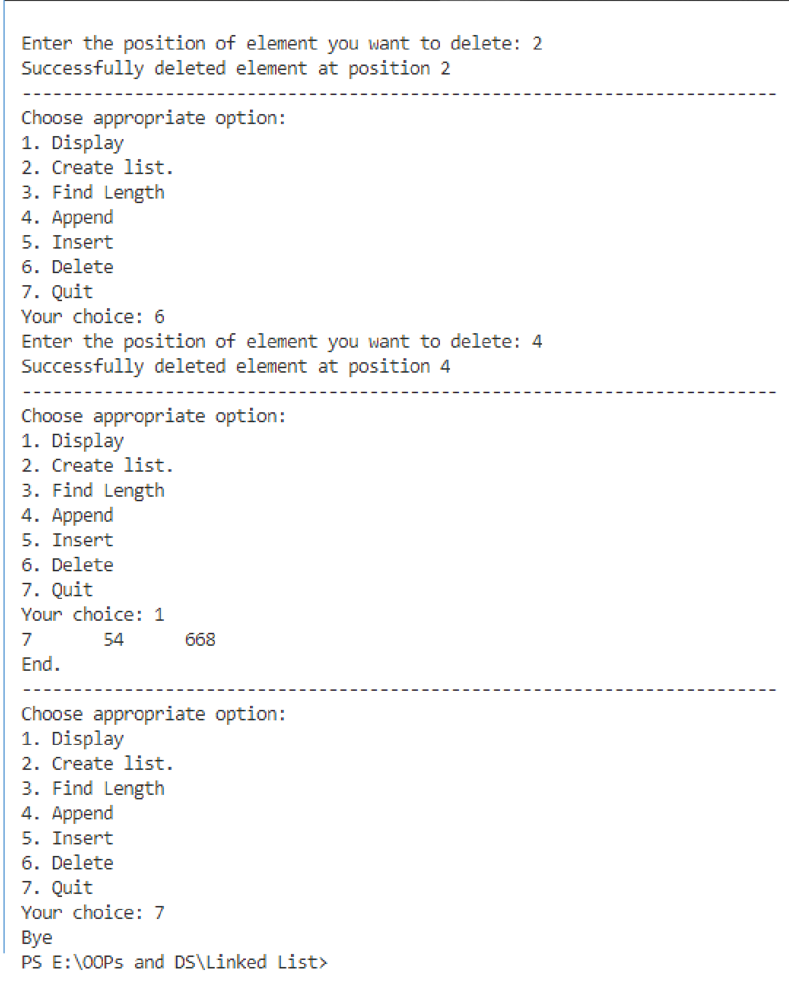
```
PS E:\Programming assignments\DS> cd "e:\OOPs and DS\Linked List\" ; if ($?) { gcc DoubleLink.c -o DoubleLink } ; if ($?) { .\DoubleLi
 }
Choose appropriate option:
1. Display
2. Create list.
3. Find Length
4. Append
5. Insert
6. Delete
7. Quit
Your choice: 2
Enter no of elements to be included in the list: 2
Enter element no. 1: 54
Enter element no. 2: 668
------------------------------------------------------------------------
Choose appropriate option:
1. Display
2. Create list.
3. Find Length
4. Append
5. Insert
6. Delete
7. Quit
Your choice: 1
54      668
End.
------------------------------------------------------------------------
Choose appropriate option:
1. Display
2. Create list.
3. Find Length
4. Append
5. Insert
6. Delete
7. Quit
```



```
Your choice: 4
Enter the new element: 88
Successfully appended element to the linked list.
------------------------------------------------------------------------
Choose appropriate option:
1. Display
2. Create list.
3. Find Length
4. Append
5. Insert
6. Delete
7. Quit
Your choice: 5
Enter the position of element to insert: 1
Enter the element to insert at position 1: 7
Successfully inserted element at position 1
------------------------------------------------------------------------
Choose appropriate option:
1. Display
2. Create list.
3. Find Length
4. Append
5. Insert
6. Delete
7. Quit
Your choice: 1
7       54      668     88
End.
------------------------------------------------------------------------
Choose appropriate option:
1. Display
2. Create list.
3. Find Length
4. Append
5. Insert
6. Delete
7. Quit
Your choice: 5
```



```
Enter the position of element to insert: 22255
Enter the element to insert at position 22255: 5
Invalid position :(
------------------------------------------------------------------------
Choose appropriate option:
1. Display
2. Create list.
3. Find Length
4. Append
5. Insert
6. Delete
7. Quit
Your choice: 5
Enter the position of element to insert: 2
Enter the element to insert at position 2: 46
Successfully inserted element at position 2
------------------------------------------------------------------------
Choose appropriate option:
1. Display
2. Create list.
3. Find Length
4. Append
5. Insert
6. Delete
7. Quit
Your choice: 1
7       46      54      668     88
End.
------------------------------------------------------------------------
Choose appropriate option:
1. Display
2. Create list.
3. Find Length
4. Append
5. Insert
6. Delete
7. Quit
Your choice: 6
```



```
Enter the position of element you want to delete: 2
Successfully deleted element at position 2
------------------------------------------------------------------------
Choose appropriate option:
1. Display
2. Create list.
3. Find Length
4. Append
5. Insert
6. Delete
7. Quit
Your choice: 6
Enter the position of element you want to delete: 4
Successfully deleted element at position 4
------------------------------------------------------------------------
Choose appropriate option:
1. Display
2. Create list.
3. Find Length
4. Append
5. Insert
6. Delete
7. Quit
Your choice: 1
7       54      668
End.
------------------------------------------------------------------------
Choose appropriate option:
1. Display
2. Create list.
3. Find Length
4. Append
5. Insert
6. Delete
7. Quit
Your choice: 7
Bye
PS E:\OOPs and DS\Linked List>
```

```c
void delete(int pos){
    if(pos==0 || pos>length()){
        printf("Invalid position!\n");
        return;
    }
    if(root==NULL){
        printf("There is no element to delete!\n");
        return;
    }
    if(pos==1){
        if(root->next!=NULL){
        root=root->next;
        root->prev=NULL;
        }
        else{
            root=NULL;
            last=NULL;
        }
        printf("Successfully deleted\n");
        return;
    }
    current=root;
    int i=1;
    while(i<pos-1){
        current=current->next;
        i++;
    }
    current->next=current->next->next;
    if(current->next==NULL)
        last=current;
    else
        current->next->prev=current;

    printf("Successfully deleted element at position %d\n", pos);
}
void main(){

    while(1){
        int ch=7;
        printf("Choose appropriate option: \n");
        printf("1. Display\n");
        printf("2. Create list.\n");
        printf("3. Find Length\n");
        printf("4. Append\n");
        printf("5. Insert\n");
        printf("6. Delete\n");
        printf("7. Quit\n");
        printf("Your choice: ");
```

```c
void create(){
    printf("Enter no of elements to be included in the list: ");
    scanf("%d",&pos);
    for(int i=1; i<=pos; i++){
        printf("Enter element no. %d: ",i);
        scanf("%d",&el);
        append(el);
    }
}
void insert(int pos, int ele){
    if(pos>length()+1 || pos==0){
        printf("Invalid position :(\n");
        return;
    }
    if(root==NULL){
        printf("No elements in the list, so adding at beginning :)\n");
        append(ele);
        return;
    }
    if(pos>length()){
        append(ele);
        printf("Successfully inserted element at position %d\n", pos);
        return;
    }
    new = (struct node*)malloc(sizeof(struct node));
    new->data=ele;
    if(pos==1){
        new->next=root;
        new->prev=NULL;
        root->prev=new;
        root=new;
        printf("Successfully inserted element at position %d\n", pos);
        return;
    }
    int ps=pos;
    current=root;
    while(pos!=1){
        current=current->next;
        pos--;
    }
    new->next=current;
    new->prev=current->prev;

    current->prev->next=new;
    current->prev=new;

    printf("Successfully inserted element at position %d\n", ps);
}
```

Code :

```c
#include<stdio.h>

#include<stdlib.h>
int el, pos;
struct node{
    int data;
    struct node *prev, *next;
}*root=NULL, *last=NULL, *new=NULL, *current=NULL;
int length(){
    if(root==NULL)
        return 0;
    int count = 0;
    current=root;
    while (current!=NULL)
    {
        count++;
        current = current->next;
    }
    return count;
}
void display(){
    if(root==NULL){
        printf("No elements in the list to display. :(\n");
        return;
    }
    current = root;
    while(current!=NULL){
        printf("%d\t",current->data);
        current = current->next;
    }
    printf("\nEnd.\n");
}
void append(int ele){
    new = (struct node*)malloc(sizeof(struct node));
    new ->data=ele;
    new ->next=NULL;
    if(root==NULL){
        root=new;
        new->prev=NULL;
    }
    else{
    new->prev=last;
    last->next=new;
    }
    last=new;
}
```