```c
//stack

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

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

void push(struct node **headptr,int value)
{
    struct node *newnode,*temp;
    newnode = (struct node*)malloc(sizeof(struct node));
    newnode->data = value;
    newnode->next = NULL;
    temp = *headptr;
    if(temp == NULL)
    {
        *headptr = newnode;
    }
    else
    {
        while(temp->next != NULL)
            temp = temp->next;
        temp->next = newnode;
    }
}
void pop(struct node **headptr)
{
    struct node *temp;
    temp = *headptr;
    if(temp == NULL)
    {
        printf("The list is Empty!!!\n");
        return;
    }
    else if(temp->next == NULL)
    {
        *headptr = NULL;
        printf("Last Element has been Deleted\n");
        return;
    }
```

```
45      else
46      {
47          while((temp->next)->next != NULL)
48              temp = temp->next;
49          temp->next = NULL;
50          printf("Top Element has been Deleted\n");
51      }
52  }
53  void display(struct node *temp)
54  {
55      if(temp == NULL)
56      {
57          printf("The list is Empty!!!\n");
58          return;
59      }
60      else
61      {
62              while(temp!=NULL)
63          {
64              printf("%d\t",temp->data);
65              temp = temp->next;
66          }
67          printf("\n");
68      }
69  }
70
71  int main(int argc, char **argv)
72  {
73      struct node *head = NULL;
74      int choice,ele;
75      while(choice != 4)
76      {
77          printf("Enter choice 1)Push 2)Pop 3)Display 4)Exit: ");
78          scanf("%d",&choice);
79          switch(choice)
80          {
81              case 1:printf("Enter value:");scanf("%d",&ele);push(&head,ele);break;
82              case 2:pop(&head);break;
83              case 3:display(head);break;
84              case 4:exit(0);
85              default:exit(0);
86          }
87      }
88      return 0;
```