

```

1 //Lab Program 8
2 //stack and Queue implementation
3 #include <stdio.h>
4 #include <stdlib.h>
5 struct node{
6     int data;
7     struct node *next;
8 };
9
10 void insertend(struct node **headptr){
11     struct node *newnode,*temp;
12     int value;
13     printf("Enter value: ");
14     scanf("%d",&value);
15     newnode = (struct node*)malloc(sizeof(struct node));
16     newnode->data = value;
17     newnode->next = NULL;
18     temp = (*headptr);
19     if(*headptr == NULL)
20         (*headptr) = newnode;
21     else{
22         while(temp->next != NULL)
23             temp = temp->next;
24         temp->next = newnode;
25     }
26 }
27 void deletefront(struct node **headptr){
28     if((*headptr) == NULL)
29         printf("The list is empty\n");
30     else if((*headptr)->next == NULL)
31         (*headptr) = NULL;
32     else{
33         (*headptr) = (*headptr)->next;
34     }
35 }
36 void deleteend(struct node **headptr){
37     struct node *temp;
38     temp = (*headptr);
39     if((*headptr) == NULL)
40         printf("The list is empty\n");
41     else if((*headptr)->next == NULL)
42         (*headptr) = NULL;
43     else{
44         temp = *headptr;

```

```

45     while((temp->next)->next != NULL)
46         temp = temp->next;
47     temp->next = NULL;
48 }
49 }
50 void display(struct node *temp){
51     if(temp == NULL){
52         printf("The list is empty\n");
53         return;
54     }
55     else{
56         while(temp != NULL){
57             printf("%d\t",temp->data);
58             temp = temp->next;
59         }
60         printf("\n");
61     }
62 }
63 int main(int argc,char **argv){
64     struct node *head1=NULL,*head2=NULL;
65     int choice;
66     while(choice != 7){
67         printf("Enter choice: 1)pushstack 2)popstack 3)displaystack 4)EnQ 5)DeQ 6)displayQ 7)exit : ");
68         scanf("%d",&choice);
69         switch(choice){
70             case 1:insertend(&head1);break;
71             case 2:deleteend(&head1);break;
72             case 3:display(head1);break;
73             case 4:insertend(&head2);break;
74             case 5:deletefront(&head2);break;
75             case 6:display(head2);break;
76             case 7:
77                 default:exit(0);
78         }
79     }
80 }

```