```c
//Queue

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

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

void enQ(struct node **headptr,int value)
{
    struct node *newnode;
    newnode = (struct node*)malloc(sizeof(struct node));
    newnode->data = value;
    newnode->next = NULL;
    if(*headptr == NULL)
        *headptr = newnode;
    else
    {
        newnode->next = *headptr;
        *headptr = newnode;
    }
}
void deQ(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;
    }
    else
    {
        while((temp->next)->next != NULL)
            temp = temp->next;
```

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