```c
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
    struct node *next;
};
void insert();
void display();
void del();

struct  node *rear=NULL, *front =NULL;

int main(int argc, char **argv)
{
;
    int choice;
    char ch = 'Y';
    do
    {
    printf("\nQueue implementation using linked list\n");
    printf("\n1. Create \n2. Display  \n3. Delete  \n4. Exit \n");
    printf("\nEnter your choice : ");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1: insert(); break;
        case 2: display();break;
        case 3: del(); break;
        case 4:
            ch = 'n';
            break;
    }
    }while(ch=='y'||ch=='Y');
}
```

```
    }
    }while(ch=='y'||ch=='Y');
}

void insert()
{
    struct  node *newnode;
    newnode=(struct node *) malloc(sizeof(struct node));
    printf("Enter the element:\n");
    scanf("%d",&newnode->data);
    newnode->next=NULL;

    if(rear==NULL)
    {
        rear=newnode;
        front=newnode;

    }
    else
    {
        rear->next=newnode;
        rear=newnode;
    }
}

void del()
{
    if(front==NULL)
    {
        printf("Queue is empty\n");return;
    }

    else
    {
        printf("Deleted ele is %d",front->data);
        if(front==rear)
```

reverse.c X | mergesort.c X | concatination.c X

```
{
    if(front==NULL)
    {
        printf("Queue is empty\n");return;
    }

    else
    {
        printf("Deleted ele is %d",front->data);
        if(front==rear)
        {
            printf("Queue is empty\n");
            front=NULL; rear=NULL;
        }
        else
        front=front->next;
    }
}

void display()
{
    struct node *temp;
    if(front ==NULL)
    {
        printf("Queue is empty");
        return;
    }
    temp=front;
    while (temp !=NULL)
    {
        printf("%d ",temp->data);
        temp=temp->next;
    }

}
```