```
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node*next;
};

struct Queue {
     struct Node*front;
      struct Node*rear;
};

void initQueue(struct Queue*q) {
    q->front=NULL;
    q->rear=NULL;
}

int isEmpty(struct Queue*q) {
    return(q->front==NULL);
}

void enQueue(struct Queue*q, int value) {
    struct Node*newNode=(struct Node*)malloc(sizeof(struct Node));

    if(newNode==NULL) {
        printf("Heap Overflow\n");
        return;
    }
    newNode->data=value;
    newNode->next=NULL;

    if(q->rear==NULL) {
        q->front=q->rear=newNode;
        return;
    }

    q->rear->next=newNode;
    q->rear=newNode;
}

void deQueue(struct Queue*q) {
    if(isEmpty(q)) {
        printf("Queue Underflow\n");
        return;
    }

    struct Node*temp=q->front;
    printf("Deleted element:%d\n,temp->data");

    q->front=q->front->next;

    if(q->front=NULL) {
      q->rear==NULL;
    free(temp);
    }
}

void peek(struct Queue*q) {
    if(isEmpty(q)) {
        printf("Queue is Empty\n");
        return;
    }
    printf("Front element:%d\n,q->front->data");
}

void display(struct Queue*q) {
    if(isEmpty(q)) {
        printf("Queue is Empty\n");
        return;
    }

    struct Node*temp=q->front;
    printf("Queue elements:");
    while(temp!=NULL) {
        printf("%d->",temp->data);
        temp=temp->next;
    }
    printf("NULL\n");
}
    int main() {
        struct Queue q;
        initQueue(&q);
        enQueue(&q,10);
        enQueue(&q,20);
        enQueue(&q,30);
        display(&q);
        deQueue(&q);
        display(&q);
    peek(&q);
    return 0;
}
```