```
/*. Implement Linked queue*/

#include <bits/stdc++.h>
using namespace std;

struct QNode {
    int data;
    QNode* next;
    QNode(int d)
    {
        data = d;
        next = NULL;
    }
};

struct Queue {
    QNode *front, *rear;
    Queue() { front = rear = NULL; }

    void enQueue(int x)
    {

        // Create a new LL node
        QNode* temp = new QNode(x);

        // If queue is empty, then
        // new node is front and rear both
        if (rear == NULL) {
            front = rear = temp;
            return;
        }

        // Add the new node at
        // the end of queue and change rear
        rear->next = temp;
        rear = temp;
    }

    // Function to remove
    // a key from given queue q
    void deQueue()
    {
        // If queue is empty, return NULL.
        if (front == NULL)
            return;

        // Store previous front and
        // move front one node ahead
        QNode* temp = front;
        front = front->next;

        // If front becomes NULL, then
        // change rear also as NULL
        if (front == NULL)
            rear = NULL;

        delete (temp);
    }
};
```

```
// Driver code
int main()
{

    Queue q;
    q.enQueue(10);
    q.enQueue(20);
    q.deQueue();
    q.deQueue();
    q.enQueue(30);
    q.enQueue(40);
    q.enQueue(50);
    q.deQueue();
    cout << "Queue Front : " << (q.front)->data << endl;
    cout << "Queue Rear : " << (q.rear)->data;
}
```