```c
/*
队列的链式存储结构：
    用一个单链表来实现
    插入和删除操作分别在链表的两头进行
    队列指针front指向链表的头
    队列指针rear指向链表的尾
*/


#include<stdio.h>
#include<malloc.h>

typedef int ElementType;
typedef struct QNode *Queue;
struct Node{
    ElementType Data;
    struct Node *Next;
};
struct QNode{ // 队列链结构
    struct Node *rear;  // 指向队尾结点
    struct Node *front; // 指向队头结点
};

Queue CreateQueue(); // 初始化队列
int IsEmpty(Queue Q); // 判断队列是否为空
void AddQ(Queue Q, ElementType item); // 入队
ElementType DeleteQ(Queue Q); // 出队

// 初始化
Queue CreateQueue()
{
    Queue Q;
    Q = (Queue)malloc(sizeof(struct QNode));
    Q->front = NULL;
    Q->rear = NULL;
    return Q;
}

// 是否为空
int IsEmpty(Queue Q)
{
    return (Q->front == NULL);
}

// 入队
void AddQ(Queue Q, ElementType item)
{
    struct Node *node;
    node = (struct Node*)malloc(sizeof(struct Node));
    node->Data = item;
    node->Next = NULL;
    if(IsEmpty(Q))
    {
        Q->front = node;
        Q->rear = node;
    }
    else
    {
        Q->rear->Next = node; // 将结点入队
        Q->rear = node; // rear 仍然保持最后
    }
}

// 出队
ElementType DeleteQ(Queue Q)
{
    struct Node *headNode;
    ElementType headData;
    if(IsEmpty(Q))
    {
        printf("队列空");
        return -1;
    }
    headNode = Q->front;
    if(Q->front == Q->rear)
    { // 队列中只有一个元素，直接将队列置空
        Q->front = Q->rear = NULL;
    }
    else
    {
        Q->front = Q->front->Next;
    }
    headData = headNode->Data;
    free(headNode);
    return headData;
}

void print(Queue Q)
{
    if(IsEmpty(Q))
        printf("队空");
    else
    {
        struct Node *p;
        int p_Data;
        p = Q->front;
        while(p)
        {
            printf("%d ", p->Data);
            p = p->Next ;
        }
    }
    printf("\n");
}

int main()
{
    Queue Q;
    Q = CreateQueue();
    print(Q);
    printf("入队5\n");
    AddQ(Q, 5);
    print(Q);
    printf("入队4\n");
    AddQ(Q, 4);
    print(Q);
    printf("入队3\n");
    AddQ(Q, 3);
    print(Q);
    printf("出队%d\n", DeleteQ(Q));
    print(Q);
    printf("出队%d\n", DeleteQ(Q));
    print(Q);
    printf("出队%d\n", DeleteQ(Q));
    print(Q);
    printf("%d\n", DeleteQ(Q));
    return 0;
}
```