```
1
    队列的链式存储结构:
2
3
        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
4
5
        队列指针front指向链表的头
        队列指针rear指向链表的尾
6
7
8
9
10
    #include<stdio.h>
    #include<malloc.h>
11
12
13
    typedef int ElementType;
14
    typedef struct QNode *Queue;
15
    struct Node{
16
        ElementType Data;
17
        struct Node *Next;
18
    };
19
    struct QNode { // 队列链结构
20
        struct Node *rear; // 指向队尾结点
        struct Node *front; // 指向队头结点
21
22
    };
23
    Queue CreateQueue(); // 初始化队列
24
25
    int IsEmpty (Queue Q); // 判断队列是否为空
    void AddQ(Queue Q, ElementType item); // \lambda \%
26
27
    ElementType DeleteQ(Queue Q); // 出队
28
29
    // 初始化
30
    Queue CreateQueue()
31
32
        Queue Q;
33
        Q = (Queue)malloc(sizeof(struct QNode));
        Q->front = NULL;
34
35
        Q->rear = NULL;
36
        return Q;
37
    }
38
39
    // 是否为空
40
    int IsEmpty (Queue Q)
41
    {
42
        return (Q->front == NULL);
43
    }
44
    // 入队
45
46
    void AddQ(Queue Q, ElementType item)
47
48
        struct Node *node;
49
        node = (struct Node*)malloc(sizeof(struct Node));
50
        node->Data = item;
51
        node->Next = NULL;
52
        if(IsEmpty(Q))
53
            Q->front = node;
54
55
            Q->rear = node;
56
        }
57
        else
58
        {
            Q->rear->Next = node; // 将结点入队
59
60
            Q->rear = node; // rear 仍然保持最后
61
        }
62
    }
63
64
    // 出队
65
    ElementType DeleteQ(Queue Q)
66
67
        struct Node *headNode;
68
        ElementType headData;
69
        if(IsEmpty(Q))
70
        {
71
            printf("队列空");
            return -1;
73
        }
```

```
74
          headNode = Q->front;
 75
          if(Q->front == Q->rear)
 76
          { // 队列中只有一个元素,直接将队列置空
 77
              Q->front = Q->rear = NULL;
 78
          }
 79
          else
 80
          {
 81
              Q->front = Q->front->Next;
 82
          }
 83
          headData = headNode->Data;
 84
          free(headNode);
 85
          return headData;
 86
      }
 87
 88
      void print(Queue Q)
 89
      {
 90
          if(IsEmpty(Q))
 91
              printf("队空");
 92
          else
 93
          {
 94
              struct Node *p;
 95
              int p_Data;
              p = Q \rightarrow front;
 96
 97
              while(p)
 98
                  printf("%d ", p->Data);
 99
100
                  p = p-Next;
101
102
          printf("\n");
103
104
      }
105
106
      int main()
107
      {
108
          Queue Q;
109
          Q = CreateQueue();
110
          print(Q);
          printf("入队5\n");
111
112
          AddQ(Q, 5);
113
          print(Q);
114
          115
          AddQ(Q, 4);
          print(Q);
116
117
          printf("\太队3\n");
118
          AddQ(Q, 3);
119
          print(Q);
          printf("出队%d\n", DeleteQ(Q));
120
121
          print(Q);
122
          printf("出队%d\n", DeleteQ(Q));
123
          print(Q);
124
          printf("出队%d\n", DeleteQ(Q));
125
          print(Q);
          printf("%d\n", DeleteQ(Q));
126
127
          return 0;
128
      }
129
```