

```

1  /*
2  队列
3
4  队列:
5      具有一定操作约束的线性表
6      只能在一端插入, 在另一端删除
7      数据插入: 入队
8      数据删除: 出队
9      先进先出
10
11  队列的顺序存储实现:
12      由一个一维数组 && 一个记录头元素位置的变量front && 一个记录尾元素位置的变量rear
13  */
14
15
16  #include<stdio.h>
17  #include<malloc.h>
18  #define MaxSize 5
19
20  typedef int ElementType;
21  typedef struct QNode *Queue;
22  struct QNode{
23      ElementType Data[MaxSize];
24      int front; // 记录队头
25      int rear; // 记录队尾
26  };
27
28  Queue CreateQueue(); // 初始化队列
29  void AddQ(Queue Q, ElementType item); // 入队
30  int IsFull(Queue Q); // 判断队列是否已满
31  ElementType DeleteQ(Queue Q); // 出队
32  int IsEmpty(Queue Q); // 判断队列是否为空
33
34  // 初始化
35  Queue CreateQueue()
36  {
37      Queue Q;
38      Q = (Queue)malloc(sizeof(struct QNode));
39      Q->front = -1;
40      Q->rear = -1;
41      return Q;
42  }
43
44  // 判断队列是否已满
45  int IsFull(Queue Q)
46  {
47      // front == rear时, 队列可以是空的, 也可以是满的
48      // 为了进行上述两种状态的区分, 可以:
49      // 设置size记录队列中存放元素的个数
50      // 设置tag表示最近一次操作是入队还是出队
51      // 牺牲一个存储空间, 如下
52      return ((Q->rear+1)%MaxSize == Q->front);
53  }
54
55  // 判断队列是否为空
56  int IsEmpty(Queue Q)
57  {
58      return (Q->front == Q->rear);
59  }
60
61  // 入队, 队尾指针后移, 然后元素入队
62  void AddQ(Queue Q, ElementType item)
63  {
64      if(IsFull(Q))
65      {
66          printf("队列满");
67          return;
68      }
69      else
70      {
71          Q->rear = (Q->rear+1)%MaxSize;
72          Q->Data[Q->rear] = item;
73      }
74  }

```

```

74
75 // 出队，队首指针后移，然后出队
76 ElementType DeleteQ(Queue Q)
77 {
78     if(IsEmpty(Q))
79     {
80         printf("队列空");
81         return 0;
82     }
83     else
84     { // front指向队列头的前一个元素
85         // 所以front+1后，正好指向当前要出队的元素
86         Q->front = (Q->front+1)%MaxSize;
87         return Q->Data[Q->front];
88     }
89 }
90
91 void print(Queue Q)
92 {
93     if(IsEmpty(Q))
94         printf("队空");
95     else
96     {
97         int i=Q->front;
98         while(i != Q->rear)
99         {
100             i = (i+1)%MaxSize;
101             printf("%d ", Q->Data[i]);
102         }
103     }
104     printf("\n");
105 }
106
107 int main()
108 {
109     Queue Q;
110     Q = CreateQueue();
111     print(Q);
112     AddQ(Q, 3);
113     printf("3入队\n");
114     print(Q);
115     AddQ(Q, 5);
116     printf("5入队\n");
117     print(Q);
118     AddQ(Q, 11);
119     printf("11入队\n");
120     print(Q);
121     printf("%d出队\n", DeleteQ(Q));
122     print(Q);
123     printf("%d出队\n", DeleteQ(Q));
124     print(Q);
125     return 0;
126 }
127

```