

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  #define QueueLimit 20
6
7  typedef int QueueElementType;
8
9  typedef struct {
10     int Front, Rear;
11     QueueElementType Element[QueueLimit];
12 } QueueType;
13
14 typedef enum {FALSE, TRUE} boolean;
15
16 void CreateQ(QueueType *Queue);
17 void RemoveQ(QueueType *Queue, QueueElementType *Item);
18 void AddQ(QueueType *Queue, QueueElementType Item);
19 void TraverseQ(QueueType Queue);
20 boolean EmptyQ(QueueType Queue);
21 boolean FullQ(QueueType Queue);
22 void Move(QueueType *Queue, int number);
23
24 int main()
25 {
26     QueueType EvenQueue, OddQueue;
27     int i, x, numberOfEvenItems, numberOfOddItems;
28
29     /*????????? ?????????? ?????????? ??????????*/
30     srand(time(NULL));
31
32     CreateQ(&EvenQueue);
33     CreateQ(&OddQueue);
34
35     /*????????????? ???? ?????????? ??? ?? ?????? ??????? ? ??????????
36     ??? ?????????? ???? ?????????? ??????*/
37     for(i=0; i < 20; i++)
38     {
39         x=rand()%20;
40
41         if(x%2)
42             AddQ(&OddQueue, x);
43         else
44             AddQ(&EvenQueue, x);
45     }
46
47     /*????????? ??? ?????????? ??? ?????? ??? ??? ???????????????? ???*/
48     printf("Size of EvenQueue:%d\n", EvenQueue.Rear);
49     TraverseQ(EvenQueue);
50
51     /*????????? ??? ?????????? ??? ?????? ??? ??? ???????????????? ???*/
52     printf("Size of OddQueue:%d\n", OddQueue.Rear);
53     TraverseQ(OddQueue);
54
55     /*????????????? ??????? ??????? ??? ?? 1 ??? ?? ?????????? ??? ?????? ???
56     ?? ??? ???????????? move ?????????????? ??? ???????????????????? ?? ???????
57     ????? ??? ?????? ??? ??????*/
58     numberOfEvenItems=1+rand()%EvenQueue.Rear;
59     printf("Random number of items=%d\n", numberOfEvenItems);
60     Move(&EvenQueue, numberOfEvenItems);
61     TraverseQ(EvenQueue);
62
63     /*????????????? ??????? ??????? ??? ?? 1 ??? ?? ?????????? ??? ?????? ???
64     ?? ??? ???????????? move ?????????????? ??? ???????????????????? ?? ???????
65     ????? ??? ?????? ??? ??????*/

```

```

67     numberOfOddItems=1+rand()%OddQueue.Rear;
68     printf("Random number of items=%d\n",numberOfOddItems);
69     Move(&OddQueue,numberOfOddItems);
70     TraverseQ(OddQueue);
71
72     return 0;
73 }
74
75 void CreateQ(QueueType *Queue)
76 {
77     Queue->Front = 0;
78     Queue->Rear = 0;
79 }
80
81 void RemoveQ(QueueType *Queue, QueueElementType *Item)
82 {
83     if(!EmptyQ(*Queue))
84     {
85         *Item = Queue ->Element[Queue -> Front];
86         Queue ->Front  = (Queue ->Front + 1) % QueueLimit;
87     }
88     else
89         printf("Empty Queue");
90 }
91
92 void AddQ(QueueType *Queue, QueueElementType Item)
93 {
94     int NewRear;
95
96     if(!FullQ(*Queue))
97     {
98         NewRear = (Queue ->Rear + 1) % QueueLimit;
99         Queue ->Element[Queue ->Rear] = Item;
100         Queue ->Rear = NewRear;
101     }
102     else
103         printf("Full Queue");
104 }
105
106 void TraverseQ(QueueType Queue) {
107     int current;
108     current = Queue.Front;
109     while (current != Queue.Rear) {
110         printf("%d ", Queue.Element[current]);
111         current = (current + 1) % QueueLimit;
112     }
113     printf("\n");
114 }
115
116 boolean EmptyQ(QueueType Queue)
117 {
118     return (Queue.Front == Queue.Rear);
119 }
120
121 boolean FullQ(QueueType Queue)
122 {
123     return ((Queue.Front) == ((Queue.Rear +1) % QueueLimit));
124 }
125
126 /*????? ?????????? ????? ??? ?? ??????????? ?? ?????? ??? ???????
127 ??? ???? ?????????????? ??? ???? ?????????? ?? ??? ???? ?????????????????????
128 ??? ????? ??? ??????? ??????*/
129 void Move(QueueType *Queue,int number)
130 {
131     QueueType TempQueue;
132     int i;

```

```
133
134     CreateQ(&TempQueue);
135
136     for(i=0;i<number;i++)
137     {
138         AddQ(&TempQueue,Queue ->Element[Queue ->Front]);
139         RemoveQ(&(*Queue),&Queue ->Element[Queue ->Front]);
140     }
141
142     for(i=0;i<number;i++)
143     {
144         AddQ(&(*Queue),TempQueue.Element[TempQueue.Front]);
145         RemoveQ(&TempQueue,&TempQueue.Element[TempQueue.Front]);
146     }
147 }
```