```c
#include <stdio.h>
#define size 3
void insertq(int[], int);
void deleteq(int[]);
void display(int[]);

int front =  - 1;
int rear =  - 1;

int main()
{
    int n, ch;
    int queue[size];
    do
    {
        printf("\n\n Circular Queue:\n1. Insert \n2. Delete\n3. Display\n4. Exit");
printf("\n enter your choice:");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1:
                printf("\nEnter number: ");
                scanf("%d", &n);
                insertq(queue, n);
                break;
            case 2:
                deleteq(queue);
                break;
            case 3:
                display(queue);
                break;
        }
    }while (ch != 0);
}
```

```
31                  break;
32              }
33          }while (ch != 0);
34      }
35
36
37      void insertq(int queue[], int item)
38      {
39          if ((front == 0 && rear == size - 1) || (front == rear + 1))
40          {
41              printf("queue is full");
42              return;
43          }
44          else if (rear ==  - 1)
45          {
46              rear++;
47              front++;
48          }
49          else if (rear == size - 1 && front > 0)
50          {
51              rear = 0;
52          }
53          else
54          {
55              rear++;
56          }
57          queue[rear] = item;
58      }
59
60      void display(int queue[])
61      {
62          int i;
63          printf("\n");
64          if (front > rear)
65          {
66              for (i = front; i < size; i++)
```

```
64      if (front > rear)
65      {
66          for (i = front; i < size; i++)
67          {
68              printf("%d ", queue[i]);
69          }
70          for (i = 0; i <= rear; i++)
71              printf("%d ", queue[i]);
72      }
73      else
74      {
75          for (i = front; i <= rear; i++)
76              printf("%d ", queue[i]);
77      }
78  }
79
80  void deleteq(int queue[])
81  {
82      if (front ==  - 1)
83      {
84          printf("Queue is empty ");
85      }
86      else if (front == rear)
87      {
88          printf("\n %d deleted", queue[front]);
89          front =  - 1;
90          rear =  - 1;
91      }
92      else
93      {
94          printf("\n %d deleted", queue[front]);
95          front++;
96      }
97  }
98
```

```
 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
 enter your choice:1

Enter number: 20


 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
 enter your choice:1

Enter number: 30


 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
 enter your choice:1

Enter number: 40


 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
 enter your choice:1
```

```
 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
 enter your choice:1

Enter number: 50
queue is full

 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
 enter your choice:2

 20 deleted

 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
 enter your choice:2

 30 deleted

 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
 enter your choice:2

 40 deleted

 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
 enter your choice:2
Queue is empty

 Circular Queue:
1. Insert
2. Delete
3. Display
4. Exit
 enter your choice:2
Queue is empty
```