```
#include <stdio.h>
#define size 5
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 \n0. Exit");
                printf("\nEnter Choice 0-3?: ");
                scanf("%d", &ch);
                switch (ch)
                {
                        case 1:
```

```
            printf("\n Enter number:");
            scanf("%d", &n);
            insertq(queue, n);
            break;
        case 2:
            deleteq(queue);
        case 3:
            display(queue);
            break;
    }
  }while(ch!=0);
}

void insertq(int queue[], int item)
{
    if((front == 0 && rear == size-1) || (front == rear+1))
    {
        printf("queue is full");
        return;
    }
    else if (rear == -1)
    {
        rear++;
        front++;
    }
    else if (rear == size-1 && front > 0)
    {
        rear = 0;
    }
    else
    {
        rear++;
    }
    queue[rear] = item;
```

```
}
void display(int queue[])
{
    int i;
    printf("\n");
    if(front > rear)
    {
        for(i= front; i< size ;i++)
        {
            printf("%d", queue[i]);
        }
        for(i=0; i<= rear; i++)
            printf("%d", queue[i]);
    }
    else
    {
        for(i= front; i<= rear; i++)
            printf("%d", queue[i]);
    }
}
void delete(int queue[])
{
    if (front== -1)
    {{
        printf("Queue is empty");
    }
    else if(front == rear)
    {
        printf("\n%d deleted", queue[front]);
        front= -1;
        rear= -1;
    }
    else
    {
        printf("\n%d deleted", queue[front]);
        front++;
    }
}
```

```
 Circular Queue:
1. Insert
2. Delete
3. Display
0. Exit
Enter Choice 0-3? : 1

Enter number: 4


 Circular Queue:
1. Insert
2. Delete
3. Display
0. Exit
Enter Choice 0-3? : 1

Enter number: 5


 Circular Queue:
1. Insert
2. Delete
3. Display
0. Exit
Enter Choice 0-3? : 3

4 5

 Circular Queue:
1. Insert
2. Delete
3. Display
0. Exit
```

```
 Circular Queue:
1. Insert
2. Delete
3. Display
0. Exit
Enter Choice 0-3? : 2

 4 deleted

 Circular Queue:
1. Insert
2. Delete
3. Display
0. Exit
Enter Choice 0-3? : 2

 5 deleted

 Circular Queue:
1. Insert
2. Delete
3. Display
0. Exit
Enter Choice 0-3? : 2
Queue is empty

 Circular Queue:
1. Insert
2. Delete
3. Display
0. Exit
Enter Choice 0-3? :
```