

WAP to simulate the working of a circular queue of integers using an array providing the following operations,  
a) Insert b) Delete c) Display and also print appropriate queue empty and queue overflow conditions.

code:

```
#include <stdio.h>
#define size 3
int item, front = 0, rear = -1, q[size], count = 0;

void insertrear()
{
    if (count == size)
    {
        printf("Queue Overflow\n"); return;
    }
    rear = (rear + 1) % size;
    q[rear] = item;
    count++;
}

int deletefront()
{
    if (count == 0)
        return -1;
    item = q[front];
    front = (front + 1) % size;
    count = count - 1;
    return item;
}

void display()
{
    int i, f;
    if (count == 0)
```

```

{
    printf("Queue Empty\n");
    return;
}

f = front
for (i = 1 ; i <= count ; i++)
{
    printf("%d\n", q[f]);
    f = (f+1) % size;
}

}
void main()
{
    int ch;
    for (;;)
    {
        printf("\n 1. Insert Rear\n 2. Delete Front\n 3. Display\n 4. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1: printf("Enter Item: ");
                    scanf("%d", &item); insertrear(); break;
            case 2 : item = deletefront();
                    if (item == -1)
                        printf("Queue Empty\n");
                    else
                        printf("Item Deleted = %d\n", item);
                    break;
            case 3 : display();
                    break;
            default : exit(0);
        }
    }
}

```



```
1 #include<stdio.h>
2 #define size 3
3 int item,front=0,rear=-1,q[size],count=0;
4 void insertrear()
5 {
6     if(count==size)
7     {
8         printf("Queue Overflow\n");
9         return;
10    }
11    rear=(rear+1)%size;
12    q[rear]=item;
13    count++;
14}
15 int deletefront()
16 {
17     if(count==0)
18     return -1;
19     item=q[front];
20     front=(front+1)%size;
21     count=count-1;
22     return item;
23}
24 void display()
25 {
26     int i,f;
27     if(count==0)
28     {
29         printf("Queue Empty\n");
30         return;
31     }
32     f=front;
33     for(i=1;i<=count;i++)
34     {
35         printf("%d\n",q[f]);
36         f=(f+1)%size;
37     }
38}
39 void main()
40 {
41     int ch;
42     for(;;)
43     {
44         printf("\n1.Insert Rear\n2.Delete Front\n3.Display\n4.Exit\n");
45         printf("Enter Choice:");
46         scanf("%d",&ch);
47         switch(ch)
48         {
49             case 1:printf("Enter Item:");
50                     scanf("%d",&item);
51                     insertrear();
52                     break;
53             case 2:item=deletefront();
54                     if(item==-1)
55                         printf("Queue Empty\n");
56                     else
57                         printf("Item Deleted =%d\n",item);
58                     break;
59             case 3:display();
60                     break;
61             default:exit(0);
62         }
63     }
64 }
```

```
× Terminal

1.Insert Rear
2.Delete Front
3.Display
4.Exit
Enter Choice:1
Enter Item:10

1.Insert Rear
2.Delete Front
3.Display
4.Exit
Enter Choice:1
Enter Item:20

1.Insert Rear
2.Delete Front
3.Display
4.Exit
Enter Choice:1
Enter Item:30

1.Insert Rear
2.Delete Front
3.Display
4.Exit
Enter Choice:1
Enter Item:40
Queue Overflow

1.Insert Rear
2.Delete Front
3.Display
4.Exit
Enter Choice:3
10
20
30

1.Insert Rear
2.Delete Front
3.Display
4.Exit
Enter Choice:2
Item Deleted =10

1.Insert Rear
2.Delete Front
3.Display
4.Exit
Enter Choice:2
Item Deleted =20

1.Insert Rear
2.Delete Front
3.Display
4.Exit
Enter Choice:2
Item Deleted =30

1.Insert Rear
2.Delete Front
3.Display
4.Exit
Enter Choice:2
Queue Empty

1.Insert Rear
2.Delete Front
3.Display
4.Exit
Enter Choice:3
Queue Empty

1.Insert Rear
2.Delete Front
3.Display
4.Exit
Enter Choice:4

Process finished.
```

```

DeQueue.c
Saved

1 #include<stdio.h>
2 #define qsize 5
3 int f=0,r=-1,ch;
4 int item,q[10];
5 int isfull()
6 {
7     return(r==qsize-1)?1:0;
8 }
9 int isempty()
10 {
11     return(f>r)?1:0;
12 }
13 void insert_rear()
14 {
15     if(isfull())
16     {
17         printf("Queue Overflow\n");
18         return;
19     }
20     r=r+1;
21     q[r]=item;
22 }
23 void delete_front()
24 {
25     if(isempty())
26     {
27         printf("Queue Empty\n");
28         return;
29     }
30     printf("Item Deleted = %d\n",q[(f++)]);
31     if(f>r)
32     {
33         f=0;
34         r=-1;
35     }
36 }
37 void insert_front()
38 {
39     if(f!=0)
40     {
41         f=f-1;
42         q[f]=item;
43         return;
44     }
45     else if((f==0)&&(r== -1))
46     {
47         q[++(r)]=item;
48         return;
49     }
50     else
51     {
52         printf("Insertion not possible\n");
53     }
54 }
55 void delete_rear()
56 {
57     if(isempty())
58     {
59         printf("Queue Empty\n");
60         return;
61     }
62     printf("Item Deleted = %d\n",q[(r--)]);
63     if(f>r)
64     {
65         f=0;
66         r=-1;
67     }
68 }
69 void display()
70 {
71     int i;
72     if(isempty())
73     {
74         printf("Queue Empty\n");
75         return;
76     }
77     for(i=f;i<=r;i++)
78     {
79         printf("%d\n",q[i]);
80     }
81 }
82 void main()
83 {
84     for(;;)
85     {
86         printf("\n1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
87         printf("Enter Choice:");
88         scanf("%d",&ch);
89         switch(ch)
90         {
91             case 1:printf("Enter Item:");
92                     scanf("%d",&item);
93                     insert_rear();
94                     break;
95             case 2:printf("Enter Item:");
96                     scanf("%d",&item);
97                     insert_front();
98                     break;
99             case 3:delete_rear();
100                    break;
101             case 4:delete_front();
102                    break;
103             case 5:display();
104                    break;
105             default:exit(0);
106         }
107     }
108 }

```

```

x Terminal
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter Choice:1
Enter Item:10

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter Choice:1
Enter Item:20

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter Choice:1
Enter Item:30

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter Choice:5
10
20
30

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter Choice:2
Enter Item:40
Insertion not possible

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter Choice:3
Item Deleted = 30

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter Choice:5
10
20

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter Choice:4
Item Deleted = 10

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter Choice:5
20

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter Choice:2
Enter Item:30

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter Choice:4
Item Deleted = 30

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter Choice:4
Item Deleted = 20

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
Enter Choice:4
Queue Empty

```



```

← Input_Restricted_Dequeue.c
Saved

1 #include<stdio.h>
2 #define qsize 5
3 int f=0,r=-1,ch;
4 int item,q[5];
5 int isfull()
6 {
7     return(r==qsize-1)?1:0;
8 }
9 int isempty()
10 {
11     return(f>r)?1:0;
12 }
13 void insert_rear()
14 {
15     if(isfull())
16     {
17         printf("Queue Overflow\n");
18         return;
19     }
20     r=r+1;
21     q[r]=item;
22 }
23 void delete_front()
24 {
25     if(isempty())
26     {
27         printf("Queue Empty\n");
28         return;
29     }
30     printf("Item deleted is %d\n",q[(f)++]);
31     if(f>r)
32     {
33         f=0;
34         r=-1;
35     }
36 }
37
38 void delete_rear()
39 {
40     if(isempty())
41     {
42         printf("Queue Empty\n");
43         return;
44     }
45     printf("Item deleted is %d\n",q[(r)--]);
46     if(f>r)
47     {
48         f=0;
49         r=-1;
50     }
51 }
52 void display()
53 {
54     int i;
55     if(isempty())
56     {
57         printf("Queue Empty\n");
58         return;
59     }
60     for(i=f;i<=r;i++)
61         printf("%d\n",q[i]);
62 }
63 void main()
64 {
65     for(;;)
66     {
67         printf("\n1.Insert Rear\n2.Delete Rear\n3.Delete Front\n4.Display\n5.Exit\n");
68         printf("Enter Choice:");
69         scanf("%d",&ch);
70         switch(ch)
71         {
72             case 1:printf("Enter Item:");
73                     scanf("%d",&item);
74                     insert_rear();
75                     break;
76             case 2:delete_rear();
77                     break;
78             case 3:delete_front();
79                     break;
80             case 4:display();
81                     break;
82             default:exit(0);
83         }
84     }
85 }

```

```
× Terminal

1.Insert Rear
2.Delete Rear
3.Delete Front
4.Display
5.Exit
Enter Choice:1
Enter Item:10

1.Insert Rear
2.Delete Rear
3.Delete Front
4.Display
5.Exit
Enter Choice:1
Enter Item:20

1.Insert Rear
2.Delete Rear
3.Delete Front
4.Display
5.Exit
Enter Choice:1
Enter Item:30

1.Insert Rear
2.Delete Rear
3.Delete Front
4.Display
5.Exit
Enter Choice:2
Item deleted is 30

1.Insert Rear
2.Delete Rear
3.Delete Front
4.Display
5.Exit
Enter Choice:4
10
20

1.Insert Rear
2.Delete Rear
3.Delete Front
4.Display
5.Exit
Enter Choice:3
Item deleted is 10

1.Insert Rear
2.Delete Rear
3.Delete Front
4.Display
5.Exit
Enter Choice:4
20

1.Insert Rear
2.Delete Rear
3.Delete Front
4.Display
5.Exit
Enter Choice:2
Item deleted is 20

1.Insert Rear
2.Delete Rear
3.Delete Front
4.Display
5.Exit
Enter Choice:2
Queue Empty

1.Insert Rear
2.Delete Rear
3.Delete Front
4.Display
5.Exit
Enter Choice:4
Queue Empty
```



```

Output_Restricted_Dequeue.c
Saved

1 #include<stdio.h>
2 #define qsize 5
3 int f=0,r=-1,ch;
4 int item,q[10];
5 int isfull()
6 {
7     return(r==qsize-1)?1:0;
8 }
9 int isempty()
10 {
11     return(f>r)?1:0;
12 }
13 void insert_rear()
14 {
15     if(isfull())
16     {
17         printf("Queue Overflow\n");
18         return;
19     }
20     r=r+1;
21     q[r]=item;
22 }
23 void delete_front()
24 {
25     if(isempty())
26     {
27         printf("Queue Empty\n");
28         return;
29     }
30     printf("Item Deleted = %d\n",q[(f)++]);
31     if(f>r)
32     {
33         f=0;
34         r=-1;
35     }
36 }
37 void insert_front()
38 {
39     if(f!=0)
40     {
41         f=f-1;
42         q[f]=item;
43         return;
44     }
45     else if((f==0)&&(r==qsize-1))
46     {
47         q[++(r)]=item;
48         return;
49     }
50     else
51         printf("Insertion not possible\n");
52 }
53 void display()
54 {
55     int i;
56     if(isempty())
57     {
58         printf("Queue Empty\n");
59         return;
60     }
61     for(i=f;i<=r;i++)
62         printf("%d\n",q[i]);
63 }
64 void main()
65 {
66     for(;;)
67     {
68         printf("\n1.insert_rear\n2.insert_front\n3.delete_front\n4.display\n5.exit\n");
69         printf("Enter Choice:");
70         scanf("%d",&ch);
71         switch(ch)
72         {
73             case 1:printf("Enter Item:");
74                     scanf("%d",&item);
75                     insert_rear();
76                     break;
77             case 2:printf("Enter Item:");
78                     scanf("%d",&item);
79                     insert_front();
80                     break;
81             case 3:delete_front();
82                     break;
83             case 4:display();
84                     break;
85             default:exit(0);
86         }
87     }
88 }

```

```
× Terminal

1.insert_rear
2.insert_front
3.delete_front
4.display
5.exit
Enter Choice:1
Enter Item:10

1.insert_rear
2.insert_front
3.delete_front
4.display
5.exit
Enter Choice:2
Enter Item:20
Insertion not possible

1.insert_rear
2.insert_front
3.delete_front
4.display
5.exit
Enter Choice:1
Enter Item:30

1.insert_rear
2.insert_front
3.delete_front
4.display
5.exit
Enter Choice:4
10
30

1.insert_rear
2.insert_front
3.delete_front
4.display
5.exit
Enter Choice:3
Item Deleted = 10

1.insert_rear
2.insert_front
3.delete_front
4.display
5.exit
Enter Choice:3
Item Deleted = 30

1.insert_rear
2.insert_front
3.delete_front
4.display
5.exit
Enter Choice:3
Queue Empty

1.insert_rear
2.insert_front
3.delete_front
4.display
5.exit
Enter Choice:4
Queue Empty

1.insert_rear
2.insert_front
3.delete_front
4.display
5.exit
Enter Choice:5

Process finished.
```