EXPERIMENT NO.4.4

CIRCULAR QUEUE:

```
#include<iostream>
using namespace std;
#define max 100 class
abc
{ public:

int front,rear; int
        arr[max]; int
        n,ch,data;



void initial()
        { front=-1; rear=-1; cout<<"Enter size of
                queue:\n";
                cin>>n; }


void menu()
{
do
                { cout<<"Select and option\n";
                        cout<<"1. Enqueue\n";
                        cout<<"2. Dequeue\n";
                        cout<<"3. Display\n";
                        cout<<"4. Exit\n"; cin>>ch;
                        switch(ch)
                        { case 1: enqueue();
                                break;
                                case 2: dequeue();
                                        break;
                                case 3: display();
                                        break;
                                case 4:

break;
}
}
while(ch!=4);
```

```
} void enqueue()
                { cout<<"Enter data:\n";
                        cin>>data;

if(front == 0 && rear== n-1)
                        { cout<<"Queue overflow\n";
} else
                        { if(front==-1)
                                { front = 0;
} rear=(rear+1)%n;
cout<<"arr["<<rear<<"]  =  "<<data<<endl; arr[rear]=data;

}
}

void dequeue()
        { if(front==-1)
                { cout<<"Queue underflow\n";

} else
                { cout<<arr[front]<<" is deleted\n";
                        arr[front]=0; if(front==n-1)
                        { front=0;
}
else if(rear == front)
                        { front = rear = -1;
} else
                        { front=(front+1)%n;
}
}
}
void display()
        { if(front<rear)
                { for(int i=front;i<=rear;i++)
                        { cout<<arr[i]<<" ";
} cout<<endl;
} if(rear<front)
                { for(int i=front;i<=rear;i++)
                        { cout<<arr[i]<<" ";
                        }
for(int i=rear;i<=front;i++)
                        { cout<<arr[i]<<" ";
                        }
```

```
}


}


}; int
main()
{ abc ob;
ob.initial(); ob.menu();
        return 0;
}
```

OUTPUT:
Enter size of queue:
3
Select and option
1. Enqueue
2. Dequeue
3. Display
4. Exit
1 Enter
data: 10
arr[0] = 10 Select
and option
1. Enqueue
2. Dequeue
3. Display
4. Exit
1 Enter
data: 20
arr[1] = 20 Select
and option
1. Enqueue
2. Dequeue
3. Display
4. Exit
1 Enter
data: 30
arr[2] = 30 Select
and option
1. Enqueue
2. Dequeue

3. Display
4. Exit
3
10 20 30
Select and option
1. Enqueue
2. Dequeue
3. Display
4. Exit
2
10 is deleted
Select and option
1. Enqueue
2. Dequeue
3. Display
4. Exit
2
20 is deleted