```
/******** Circular queue  ***********/
#include<iostream>
using namespace std;
class CQueue
{
   int queue[5],front,rear,n;
   public:
     CQueue()
     {
        n=5;
        front=-1;
        rear=-1;
     }
     void enqueue(int data);
     int dequeue();
     void display();
};
void CQueue::enqueue(int data)
{
   if(front==-1 && rear==-1)
   {
      front=rear=0;
      queue[rear]=data;
   }
   else if(((rear+1)%n)==front)
   {
      cout<<"queue is full";
   }
   else
   {
      rear=(rear+1)%n;
      queue[rear]=data;
   }
}
int CQueue::dequeue()
{
   int data;
   if(front==-1)
   {
      cout<<"Underflow";
   }   else if(front==rear)
   {
      data=queue[front];
      front=rear=-1;
   }
   else
   {
      data=queue[front];
```

```
        front=(front+1)%n;
    }
    return data;
}
void CQueue::display()
{
    int i=front;
    if(front==-1 && rear==-1)
    {
        cout<<"\n queue is empty";
    }
    else
    {
        while(i!=rear)
        {
            cout<<queue[i]<<" ";
            i=(i+1)%n;
        }
        cout<<queue[rear];
    }
}
int main()
{
    int ch,data;
    CQueue q=CQueue();
    do
    {
        cout<<"\n 1. Enqueue(insertion)";
        cout<<"\n 2. Dequue(deletion";
        cout<<"\n 3. Display";
        cout<<"\n 4. Exit";
        cout<<"\n enter your choice: ";
        cin>>ch;
        switch(ch)
        {
            case 1:
                    cout<<"\n enter data: ";
                    cin>>data;
                    q.enqueue(data);
                    break;
            case 2:
                    data=q.dequeue();
                    cout<<"\n deleted data is: "<<data;
                    break;
            case 3:
                    q.display();
                    break;
        }
    }while(ch!=4);
    return 0;
}
```