// Write a menu driven program to implement insertion, deletion and display operations on circular queue and priority queue using arrays.

```
#include <stdio.h>
int n, max;
int front=-1, rear=-1, front_pq=-1, rear_pq=-1;
struct item
{
   int data;
   int prio;
};
struct item pqueue[10];
int cqueue[10];

void insert_cq(int element)
{
   if(front==-1 && rear==-1)
   {
      front=0;
      rear=0;
      cqueue[rear]=element;
   }
   else if((rear+1)%n==front)
   {
      printf("Queue overflow\n");
   }
   else
   {
      rear=(rear+1)%n;
      cqueue[rear]=element;
   }
}

int delete_cq()
{
   if((front==-1) && (rear==-1))
   {
      printf("Queue underflow\n");
   }
   else if(front==rear)
   {
      printf("The dequeued element is %i\n", cqueue[front]);
      front=-1;
      rear=-1;
   }
   else
   {
      printf("The dequeued element is %i\n", cqueue[front]);
         front=(front+1)%n;
   }
}

void display_cq()
{
   if (front == -1)
   {
```

```
    printf("Queue is Empty\n");
}
printf("Elements in Circular Queue are:\n");
if (rear >= front)
{
    for (int i = front; i <= rear; i++)
        printf("%d ",cqueue[i]);
}
else
{
    for (int i = front; i < n; i++)
        printf("%i ", cqueue[i]);

    for (int i = 0; i <= rear; i++)
        printf("%i ", cqueue[i]);
    }
    printf("\n");
}

void insert_pq(int dat, int pri)
{
    int i, loc;
    if(front_pq==0&&rear_pq==max-1)
    {
        printf("Priority Queue is full\n");
    }
    else if(front_pq==-1)
    {
    front_pq=0;
    rear_pq=0;
    pqueue[rear_pq].data=dat;
    pqueue[rear_pq].prio=pri;
}
else
{
    if(rear_pq==max-1)
    {
        for(i=front_pq; i<=rear_pq; i++)
        {
            pqueue[i-1]=pqueue[i];
        }
        front_pq--;
        rear_pq–;
    }
    for(i=rear_pq; i>=front_pq; i–)
    {
    if(pqueue[i].prio<pri)
    {
        break;
    }
}
loc=i+1;
for(i=rear; i>=loc; i–)
{
    pqueue[i+1]=pqueue[i];
}
```

```
        pqueue[loc].data=dat;
        pqueue[loc].prio=pri;
        rear_pq++;
    }
}

void delete_pq()
{
    if(front_pq==-1&&rear_pq==-1)
    {
        printf("The Priority Queue is empty\n");
    }
    else
    {
        printf("Data deleted is: %i\n", pqueue[front_pq].data);
        printf("Priority of the data deleted is: %i\n", pqueue[front_pq].prio);
        front_pq++;
    }
}

void display_pq()
{
    if(front_pq==-1&&rear_pq==-1)
    {
        printf("The Priority Queue is empty\n");
    }
    else
    {
        for(int i=front_pq; i<=rear_pq; i++)
        {
            printf("%i\t%i\n", pqueue[i].data, pqueue[i].prio);
        }
    }
}
void main()
{
    int x=0, ch1;
    while(x==0)
    {
        printf("1. Circular Queue\n2. Priority Queue\n3. Exit\n");
        printf("Enter a choice:\n");
        scanf("%i", &ch1);
        if(ch1==1)
        {
            printf("Enter the size of Circular Queue:\n");
            scanf("%i", &n);
            int c=0;
            while(c==0)
            {
                int ch2;
                printf("1. Insert\n2. Delete\n3. Display\n4. Exit\n");
                printf("Enter the choice:\n");
                scanf("%i", &ch2);
                if(ch2==1)
                {
                    int t;
```

```
            printf("Enter an element:\n");
            scanf("%i", &t);
            insert_cq(t);
        }
        else if(ch2==2)
        {
            delete_cq();
        }
        else if(ch2==3)
        {
            display_cq();
        }
        else
        {
            c++;
        }
    }
}
else if(ch1==2)
{
    printf("Enter the size of priority queue:\n");
    scanf("%i", &max);
    int p=0;
    while(p==0)
    {
        int ch2;
        printf("1. Insert\n2. Delete\n3. Display\n4. Exit\n");
        printf("Enter the choice:\n");
        scanf("%i", &ch2);
        if(ch2==1)
        {
            int a, b;
            printf("Enter the element and its priority:\n");
            scanf("%i%i", &a, &b);
            insert_pq(a, b);
        }
        else if(ch2==2)
        {
            delete_pq();
        }
        else if(ch2==3)
        {
            display_pq();
        }
        else
        {
            p++;
        }
    }
    else
    {
        printf("Exiting the program\n");
        x++;
    }
}
```

}