```
# include<stdio.h>
# define MAX 5

int cqueue_arr[MAX];
int front = -1;
int rear = -1;

/*Begin of insert*/
void insert(int item)
{
        if((front == 0 && rear == MAX-1) || (front == rear+1))
        {
                printf("Queue Overflow \n");
                return;
        }
        if (front == -1)  /*If queue is empty */
        {
                front = 0;
                rear = 0;
        }
        else
        {
                if(rear == MAX-1)       /*rear is at last position of queue */
                        rear = 0;
                else
                        rear = rear+1;
        }
        cqueue_arr[rear] = item ;
}
/*End of insert*/

/*Begin of del*/
void del()
{
        if (front == -1)
        {
                printf("Queue Underflow\n");
                return ;
        }
        printf("Element deleted from queue is : %d\n",cqueue_arr[front]);
        if(front == rear) /* queue has only one element */
        {
                front = -1;
                rear=-1;
        }
        else
        {
                if(front == MAX-1)
                        front = 0;
                else
                        front = front+1;
        }
}
/*End of del() */

/*Begin of display*/
void display()
```

```
{
	int front_pos = front,rear_pos = rear;
	if(front == -1)
	{
		printf("Queue is empty\n");
		return;
	}
	printf("Queue elements :\n");
	if( front_pos <= rear_pos )
		while(front_pos <= rear_pos)
		{
			printf("%d ",cqueue_arr[front_pos]);
			front_pos++;
		}
	else
	{
		while(front_pos <= MAX-1)
		{
			printf("%d ",cqueue_arr[front_pos]);
			front_pos++;
		}
		front_pos = 0;
		while(front_pos <= rear_pos)
		{
			printf("%d ",cqueue_arr[front_pos]);
			front_pos++;
		}
	}
	printf("\n");
}
/*End of display*/

/*Begin of main*/
int main()
{
	int choice,item;
	do
	{
		printf("1.Insert\n");
		printf("2.Delete\n");
		printf("3.Display\n");
		printf("4.Quit\n");

		printf("Enter your choice : ");
		scanf("%d",&choice);

		switch(choice)
		{
			case 1 :
				printf("Input the element for insertion in queue : ");
				scanf("%d", &item);

				insert(item);
				break;
			case 2 :
				del();
				break;
```

```
                case 3:
                        display();
                        break;
                case 4:
                        break;
                        default:
                        printf("Wrong choice\n");
                }
        }while(choice!=4);

        return 0;
}
```

```
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice : 1
Input the element for insertion in queue : 34
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice : 1
Input the element for insertion in queue : 45
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice : 3
Queue elements :
34 45
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice : 4

Process returned 0 (0x0)   execution time : 19.484 s
Press any key to continue.
```