```c
#include<stdio.h>
#define q_size 5
int item,front=-1,rear=-1,q[10];
void insert_rear()
{
    if(rear==q_size-1)
    {
        printf("queue overflow\n");
        return;
    }
    rear=rear+1;
    q[rear]=item;
}
 int delete_front()
 {
     if(front>rear)
     {
         front=0;
         rear=-1;
     }
     return q[front++];
 }
 void displayQ()
 {
     int i;
     if(front>rear)
     printf("queue empty\n");
     else
     for(i=front;i<=rear;i++)
     {
         printf("%d\n",q[i]);
     }
 }
 void main()
 {
     int choice;

     for(;;)
     {
         printf("\n1.insertrear\n2.deletefront\n3.display\n4.exit\n");
         printf("enter the choice\n");
         scanf("%d",&choice);
         switch(choice)
         {
             case 1: printf("enter the item to be inserted\n");
                     scanf("%d",&item);
                     insert_rear();
                     break;
             case 2:
                     item=delete_front();
                     if(item==-1)
                     if(item==-1)
                     printf("empty queue\n");
                     else
                     printf("element deleted: %d",item);
                     break;
             case 3:
                     displayQ();
                     break;
             default:
                     printf("end of operation\n");


         }
     }

 }
```

```
1.insertrear
2.deletefront
3.display
4.exit
enter the choice
1
enter the item to be inserted
10

1.insertrear
2.deletefront
3.display
4.exit
enter the choice
1
enter the item to be inserted
20

1.insertrear
2.deletefront
3.display
4.exit
enter the choice
1
enter the item to be inserted
30

1.insertrear
2.deletefront
```

```
enter the choice
3
0
10
20
30
40
50

1.insertrear
2.deletefront
3.display
4.exit
enter the choice
2
element deleted: 0
1.insertrear
2.deletefront
3.display
4.exit
enter the choice
2
element deleted: 10
1.insertrear
2.deletefront
3.display
4.exit
enter the choice
2
element deleted: 20
```

```
1.insertrear
2.deletefront
3.display
4.exit
enter the choice
2
element deleted: 20
1.insertrear
2.deletefront
3.display
4.exit
enter the choice
3
30
40
50

1.insertrear
2.deletefront
3.display
4.exit
enter the choice
4
end of operation

1.insertrear
2.deletefront
3.display
4.exit
```

```c
#include<stdio.h>
#include<stdlib.h>
# define q_size 5
int item,front=0,rear=-1,q[q_size],count=0;
void insert_rear()
{
    if(count==q_size)
    {
    printf("queue underflow\n");
    return;
    }
    rear=(rear+1)%q_size;
    q[rear]=item;
    count++;
}
int delete_front()
{
    if(count==0)
    return -1;
    item=q[front];
    front=(front+1)%q_size;
    count=count-1;
    return item;
}
void displayQ()
{
    int i,f;
    if(count==0)
    {
        printf("queue is empty\n");
        return;
```

```
    }
    f=front;
    printf("contents of queue\n");
    for(i=1;i<=count;i++)
    {
        printf("%d\n",q[f]);
        f=(f+1)%q_size;
    }
}
void main()
{
    int choice;

    for(;;)
    {
        printf("\n1.insertrear\n2.deletefront\n3.display\n4.exit\n");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: printf("enter the item to be inserted\n");
                    scanf("%d",&item);
                    insert_rear();
                    break;
            case 2: item=delete_front();
                    if(item==-1)
                    printf("\n empty queue");
                    else
                    printf("item deleted=%d\n",item);
                    break;
            case 3:
                    displayQ();
```

```
            case 3:
                    displayQ();
                    break;
            default:
                    printf("enf of operation");
                    exit(0);


        }
    }

}
```

```
1.insertrear
2.deletefront
3.display
4.exit
1
enter the item to be inserted
1

1.insertrear
2.deletefront
3.display
4.exit
1
enter the item to be inserted
3

1.insertrear
2.deletefront
3.display
4.exit
1
enter the item to be inserted
5
```

```
enter the item to be inserted
5

1.insertrear
2.deletefront
3.display
4.exit
1
enter the item to be inserted
7

1.insertrear
2.deletefront
3.display
4.exit
1
enter the item to be inserted
9

1.insertrear
2.deletefront
```

```
1.insertrear
2.deletefront
3.display
4.exit
3
contents of queue
1
3
5
7
9

1.insertrear
2.deletefront
3.display
4.exit
2
item deleted=1

1.insertrear
2.deletefront
3.display
```

```
2.deletefront
3.display
4.exit
2
item deleted=3

1.insertrear
2.deletefront
3.display
4.exit
3
contents of queue
5
7
9

1.insertrear
2.deletefront
3.display
4.exit
4
enf of operation
```