

```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#define QUE_SIZE 3
```

```
int item, front = 0, rear = -1, q[10];
```

```
void insertrear () {
```

```
if (rear == QUE_SIZE - 1) {
```

```
printf ("queue overflowin");
```

```
return;
```

```
}
```

```
rear += 1;
```

```
q[rear] = item;
```

```
}
```

```
int deletefront () {
```

```
if (front > rear) {
```

```
front = 0;
```

```
rear = -1;
```

```
return -1;
```

```
}
```

```
return q[front++];
```

```
}
```

```
void display () {
```

```
int i;
```

```
if (front > rear)
```

```
printf ("\n..d\n", q[i]);
```

```
}
```

```
void main () {
```

```
int choice;
```

```
clrscr();
```

```
for(;;) {
    printf ("1:insert item 2:delete front 3:display\n"
           "4:exit\n");
    printf ("enter choice\n");
    scanf ("%d", &choice);
    switch (choice) {
        case 1: printf ("enter item to be inserted\n");
                  scanf ("%d", &item);
                  insertrear();
                  break;
        case 2: item = deletefront();
                  if (item == -1)
                      printf ("queue is empty\n");
                  else
                      printf ("item deleted = %d\n", item);
                  break;
        case 3: display();
                  break;
        case 4: exit();
    }
}
```

```
1 #include <stdio.h>
2 #include <conio.h>
3 #include <process.h>
4 #define QUE_SIZE 3
5 int item,front=0,rear=-1,q[10];
6 void insertrear() {
7     if(rear==QUE_SIZE-1) {
8         printf("Queue overflow\n");
9         return;
10    }
11    rear+=1;
12    q[rear]=item;
13 }
14 int deletefront() {
15     if(front>rear) {
16         front=0;
17         rear=-1;
18         return -1;
19     }
20     return q[front++];
21 }
22 void display() {
23     int i=0;
24     if(front>rear)
25
26     printf("%d\n",q[i]);
27 }
28 int main() {
29     int choice;
30     for(;;) {
31         printf("1:insert rear\n2:delete front\n3:display\n4:exit\n");
32         scanf("%d",&choice);
33         switch(choice) {
34             case 1:printf("Enter item\n");
35             scanf("%d",&item);
36             insertrear();
37             break;
```

```
14 int deletefront() {
15     if(front>rear) {
16         front=0;
17         rear=-1;
18         return -1;
19     }
20     return q[front++];
21 }
22 void display() {
23     int i=0;
24     if(front>rear)
25
26     printf("%d\n",q[i]);
27 }
28 int main() {
29     int choice;
30     for(;;) {
31         printf("1:insert rear\n2:delete front\n3:display\n4:exit\n");
32         scanf("%d",&choice);
33         switch(choice) {
34             case 1:printf("Enter item\n");
35             scanf("%d",&item);
36             insertrear();
37             break;
38             case 2:item=deletefront();
39             if(item==-1)
40                 printf("Queue is empty\n");
41             else
42                 printf("item deleted=%d",item);
43             break;
44             case 3:display();
45             break;
46             case 4:return 0;
47         }
48     }
49 }
```

```
1:insert rear
2:delete front
3:display
4:exit
1
Enter item
1
1:insert rear
2:delete front
3:display
4:exit
1
Enter item
2
1:insert rear
2:delete front
3:display
4:exit
1
Enter item
3
1:insert rear
2:delete front
3:display
4:exit
1
Enter item
4
Queue overflow
1:insert rear
2:delete front
3:display
4:exit
3
1:insert rear
2:delete front
3:display
4:exit
2
item deleted=11:insert rear
2:delete front
3:display
4:exit
2
item deleted=21:insert rear
2:delete front
3:display
4:exit
2
item deleted=31:insert rear
2:delete front
3:display
4:exit
2
Queue is empty
```