**Dequeues** → To delete front, delete rear, insert front, insert rear, display.

① 
```c
#include <stdio.h>
#include <conio.h>
#include <process.h>
#define qsize 5
int f=0, r=-1, ch;
int item, q[10];
int isfull() {
    return (r==qsize-1)? 1:0;
}
int isempty() {
    return (f>r)? 1:0;
}
void insert_rear() {
    if(isfull()) {
        printf("Queue Overflow\n");
        return;
    }
    r=r+1;
    q[r] = item;
}
void delete_front() {
    if(isempty()) {
        printf("Queue empty\n");
        f=0;
        r=-1;
        return;
    }
    printf("item deleted is %d \n", q[f++]);
}
void insert_front() {
    if(f != 0) {
        f = f-1;
        q[f] = item;
        return;
    }
}
```

```c
else if ((f == 0) && (r == -1)) {
    q[++(r)] = item;
    return;
}
else
    printf("Insertion is not possible\n");
}
void delete_rear() {
    if (isempty()) {
        printf("Queue is empty\n");
        f = 0;
        r = -1;
        return;
    }
    printf("Item delleted is %d\n",
                    q[(r)--]);
}
void display() {
    int i;
    if (is empty()) {
        printf("Queue empty \n");
        return;
    }
    for (i = f; i <= r; i++)
        printf("%d\n", q[i]);
}
int main() {
    for(;;) {
        printf("1. insert-rear \n 2. insert-from
             \n 3. delete-rear \n 4. delete_front
             \n 5. display \n 6. exit");
        printf("Enter choice \n");
        scanf("%d", &ch);
```

```c
switch (ch) {
    case 1 : printf ("Enter the item \n");
             scanf ("%d", &item);
             insert_rear ();
             break;
    case 2 : printf ("enter the item \n");
             scanf ("%d", &item);
             insert_front ();
             break;
    case 3 : deleterear ();
             break;
    case 4 : deletefront ();
             break;
    case 5 : display ();
             break;
    default : exit (0);
}
}
return 0;
}
```