```c
#include<stdio.h>
#define MAX 5
void insert(int val, int *rear, int *front, int queue[]) {
    if (*rear == MAX - 1) {
        printf("The queue is full\n");
    } else {
        if (*front == -1 && *rear == -1) {
            *front = *rear = 0;
        } else {
            *rear = *rear + 1;
        }
        queue[*rear] = val;
        printf("%d inserted into the queue\n", val);
    }
}
void delete(int *front, int *rear, int queue[]) {
    if (*front == -1 || *front > *rear) {
        printf("The queue is empty\n");
    } else {
        int val = queue[*front];
        printf("%d deleted from the queue\n", val);
        *front = *front + 1;
        if (*front > *rear) {
            *front = *rear = -1;
        }
    }
}
void display(int *rear, int *front, int queue[]) {
    if (*front == -1) {
        printf("The queue is empty\n");
    } else {
        printf("Queue elements are: ");
        for (int i = *front; i <= *rear; i++) {
            printf("%d ", queue[i]);
        }
    }
}
```

```c
int main() {
    int choice = 0, val;
    int queue[MAX];
    int front = -1, rear = -1;

        do{
        printf("Queue Operations:\n");
        printf("1. Insert\n");
        printf("2. Delete\n");
        printf("3. Display\n");
        printf("4. Exit\n");
        printf("Enter your option: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter the value to insert:\n ");
                scanf("%d", &val);
                insert(val, &rear, &front, queue);
                break;
            case 2:
                delete(&front, &rear, queue);
                break;
            case 3:
                display(&rear, &front, queue);
                break;
            case 4:
                printf("Exiting the program. \n");
                break;

        }
    }while(choice!=4);

    return 0;
}
```

```
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 1
Enter the value to insert:
 9
9 inserted into the queue
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 1
Enter the value to insert:
 8
8 inserted into the queue
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 1
Enter the value to insert:
 7
7 inserted into the queue
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 1
Enter the value to insert:
 6
6 inserted into the queue
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 1
Enter the value to insert:
 5
5 inserted into the queue
```

```
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 3
Queue elements are: 9 8 7 6 5 Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 1
Enter the value to insert:
 5
The queue is full
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 2
9 deleted from the queue
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 3
Queue elements are: 8 7 6 5 Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 2
8 deleted from the queue
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 2
7 deleted from the queue
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 2
6 deleted from the queue
```

```
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 2
5 deleted from the queue
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 2
The queue is empty
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your option: 4
Exiting the program.

Process returned 0 (0x0)   execution time : 52.901 s
Press any key to continue.
```