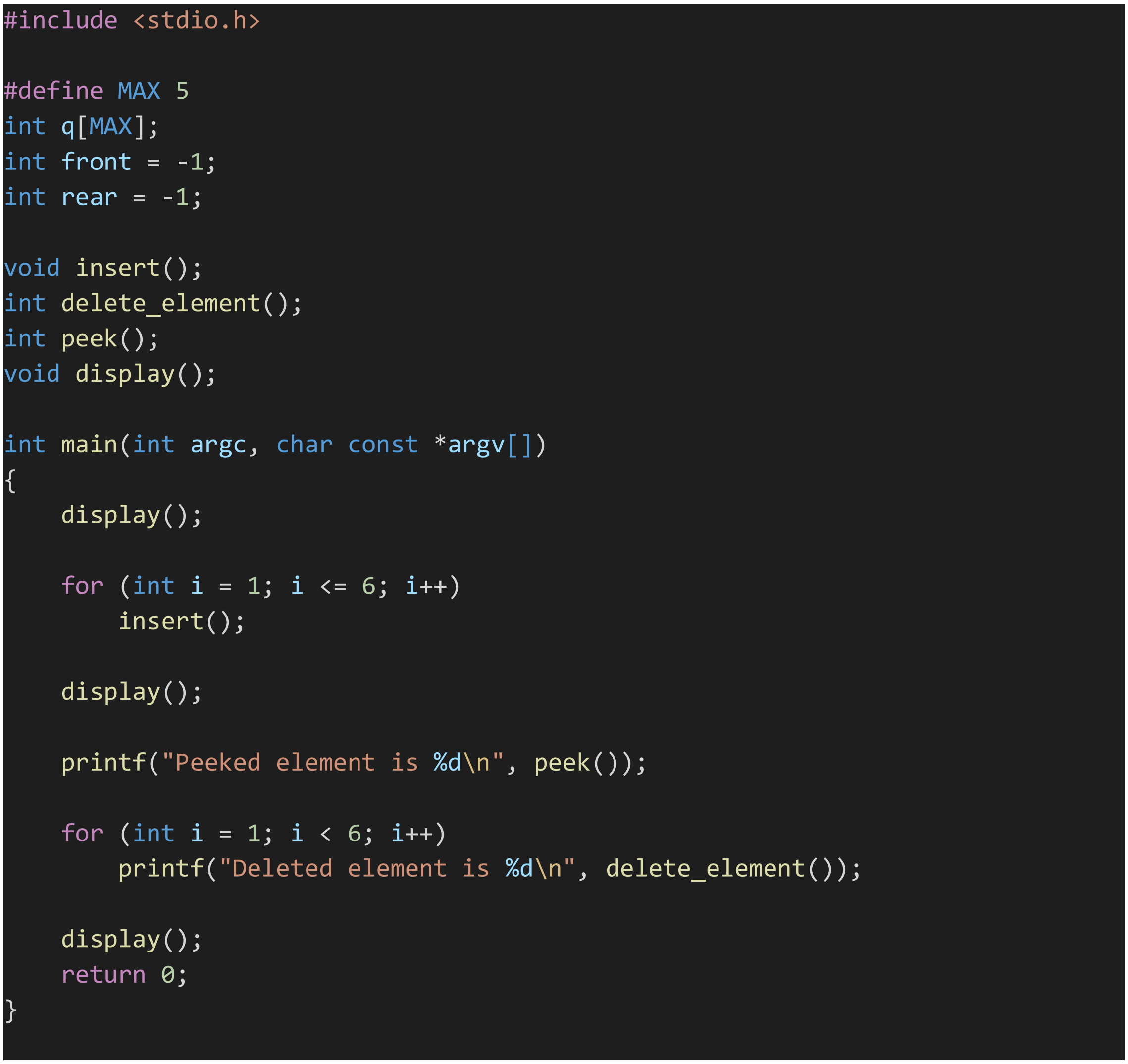
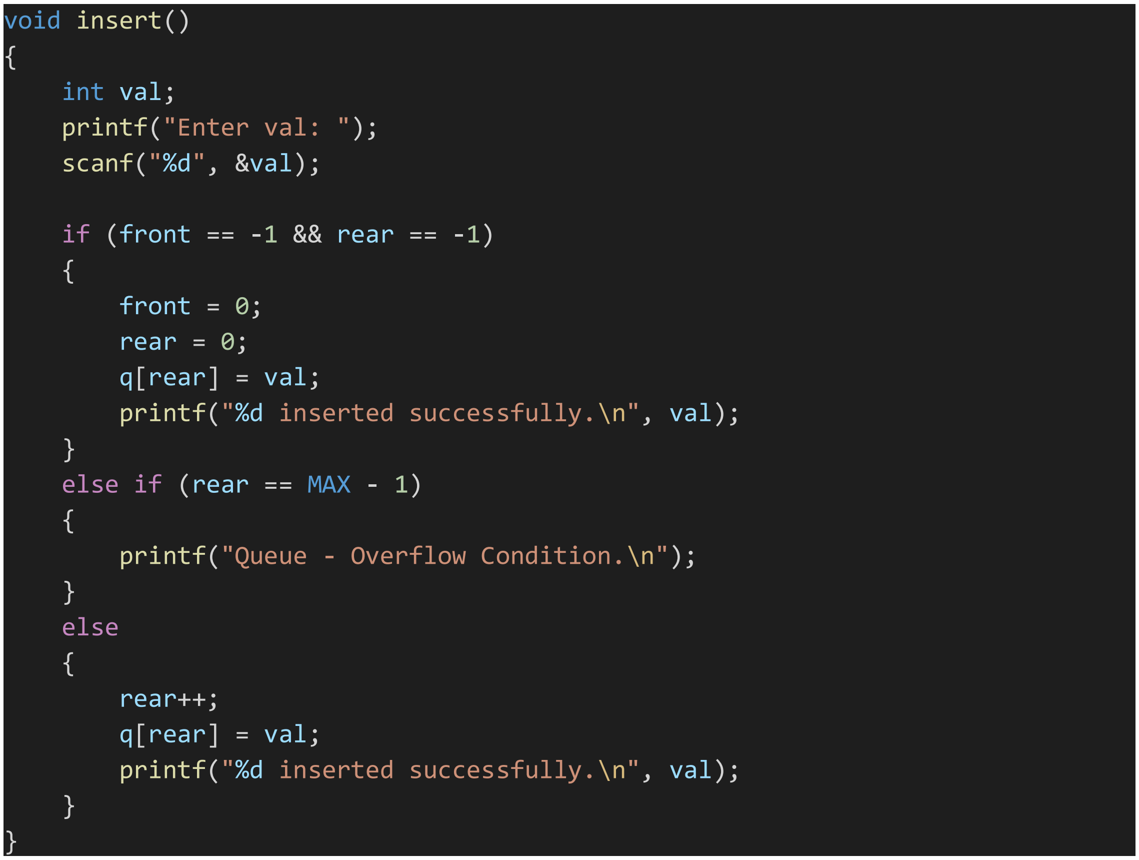
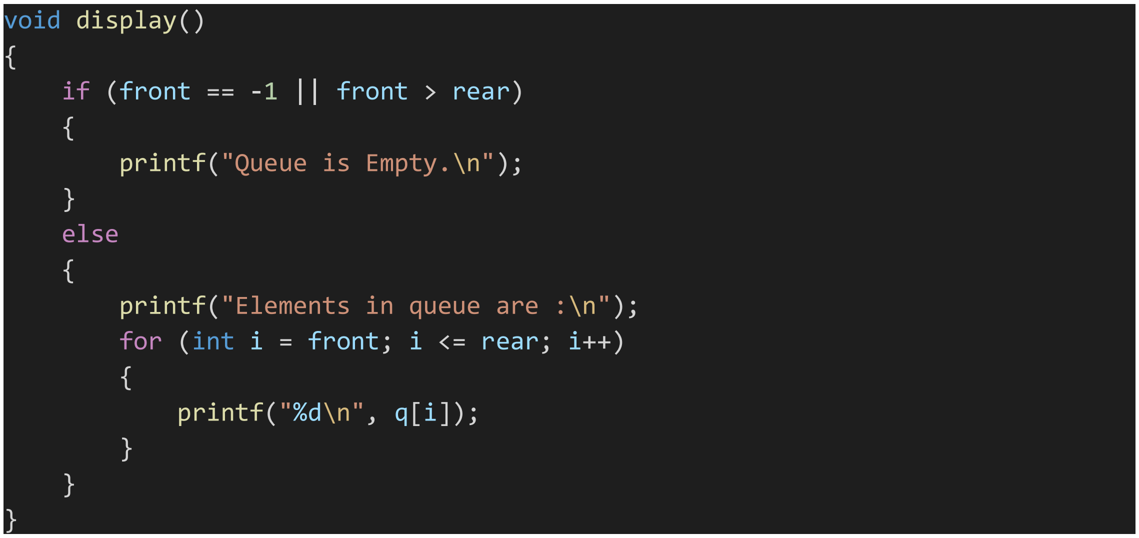


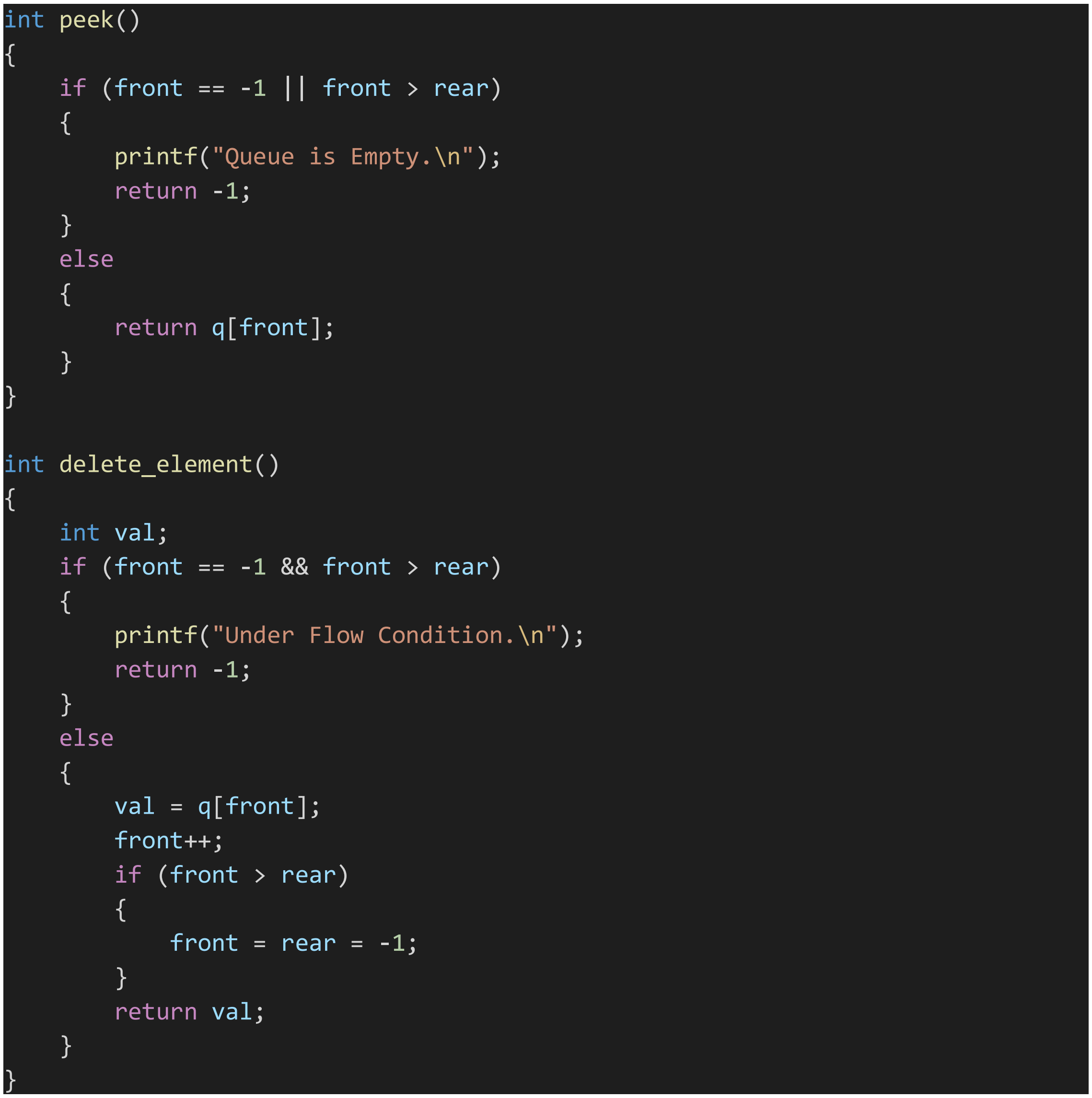
```c
#include <stdio.h>

#define MAX 5
int q[MAX];
int front = -1;
int rear = -1;

void insert();
int delete_element();
int peek();
void display();

int main(int argc, char const *argv[])
{
    display();

    for (int i = 1; i <= 6; i++)
        insert();

    display();

    printf("Peeked element is %d\n", peek());

    for (int i = 1; i < 6; i++)
        printf("Deleted element is %d\n", delete_element());

    display();
    return 0;
}
```



```c
void insert()
{
    int val;
    printf("Enter val: ");
    scanf("%d", &val);

    if (front == -1 && rear == -1)
    {
        front = 0;
        rear = 0;
        q[rear] = val;
        printf("%d inserted successfully.\n", val);
    }
    else if (rear == MAX - 1)
    {
        printf("Queue - Overflow Condition.\n");
    }
    else
    {
        rear++;
        q[rear] = val;
        printf("%d inserted successfully.\n", val);
    }
}
```



```c
void display()
{
    if (front == -1 || front > rear)
    {
        printf("Queue is Empty.\n");
    }
    else
    {
        printf("Elements in queue are :\n");
        for (int i = front; i <= rear; i++)
        {
            printf("%d\n", q[i]);
        }
    }
}
```



```c
int peek()
{
    if (front == -1 || front > rear)
    {
        printf("Queue is Empty.\n");
        return -1;
    }
    else
    {
        return q[front];
    }
}

int delete_element()
{
    int val;
    if (front == -1 && front > rear)
    {
        printf("Under Flow Condition.\n");
        return -1;
    }
    else
    {
        val = q[front];
        front++;
        if (front > rear)
        {
            front = rear = -1;
        }
        return val;
    }
}
```