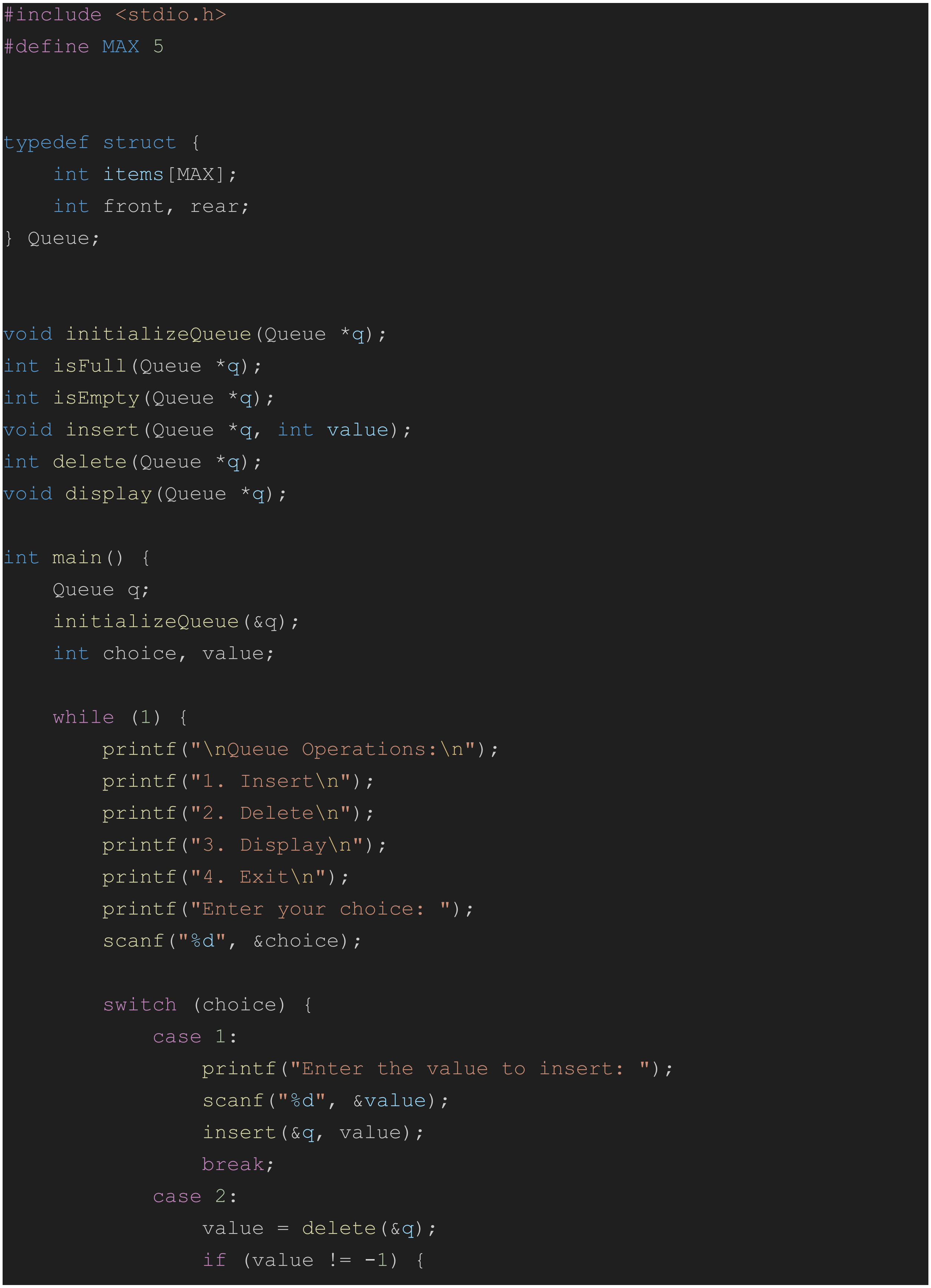
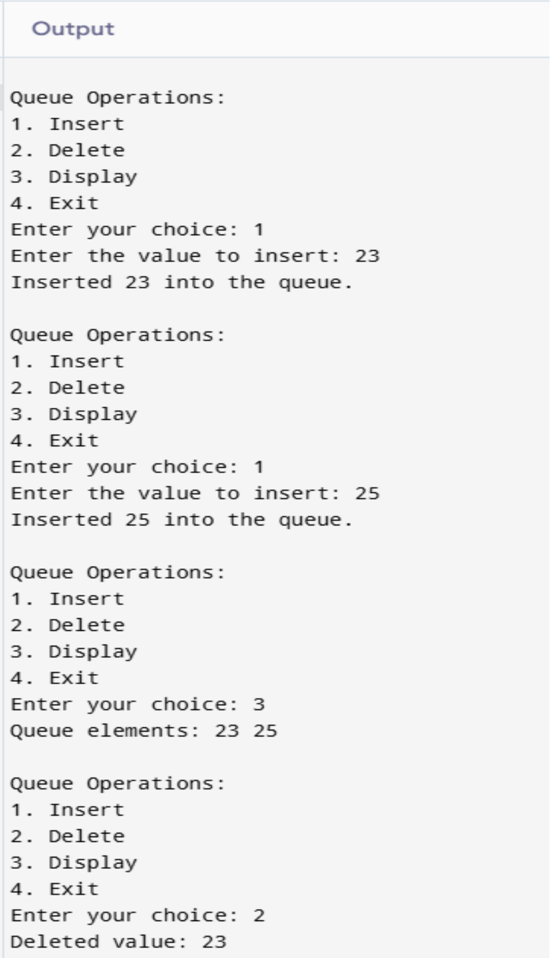
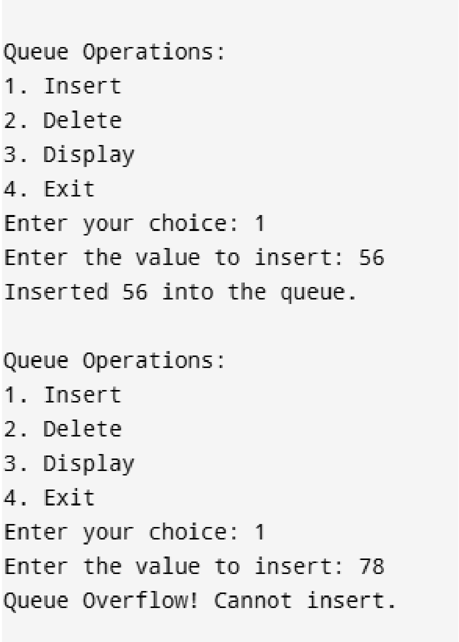


```
#include <stdio.h>
#define MAX 5


typedef struct {
    int items[MAX];
    int front, rear;
} Queue;


void initializeQueue(Queue *q);
int isFull(Queue *q);
int isEmpty(Queue *q);
void insert(Queue *q, int value);
int delete(Queue *q);
void display(Queue *q);

int main() {
    Queue q;
    initializeQueue(&q);
    int choice, value;

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

        switch (choice) {
            case 1:
                printf("Enter the value to insert: ");
                scanf("%d", &value);
                insert(&q, value);
                break;
            case 2:
                value = delete(&q);
                if (value != -1) {
                    printf("Deleted value: %d\n", value);
                }
                break;
            case 3:
                display(&q);
                break;
            case 4:
                printf("Exiting...\n");
                return 0;
            default:
                printf("Invalid choice! Try again.\n");
        }
    }
}

void initializeQueue(Queue *q) {
    q->front = -1;
    q->rear = -1;
}

int isFull(Queue *q) {
    return q->rear == MAX - 1;
}

int isEmpty(Queue *q) {
    return q->front == -1 || q->front > q->rear;
}


void insert(Queue *q, int value) {
    if (isFull(q)) {
        printf("Queue Overflow! Cannot insert.\n");
        return;
    }
    if (q->front == -1) {
        q->front = 0;
    }
    q->items[++q->rear] = value;
    printf("Inserted %d into the queue.\n", value);
}
```

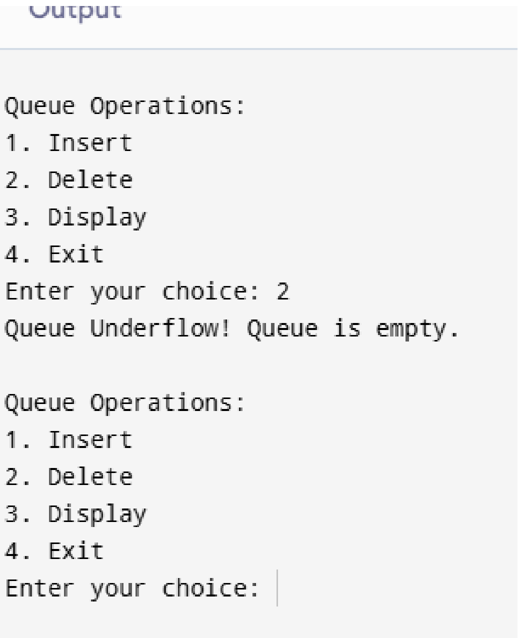
```c
int delete(Queue *q) {
    if (isEmpty(q)) {
        printf("Queue Underflow! Queue is empty.\n");
        return -1;
    }
    return q->items[q->front++];
}


void display(Queue *q) {
    if (isEmpty(q)) {
        printf("Queue is empty.\n");
        return;
    }
    printf("Queue elements: ");
    for (int i = q->front; i <= q->rear; i++) {
        printf("%d ", q->items[i]);
    }
    printf("\n");
}
```



```
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to insert: 23
Inserted 23 into the queue.

Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to insert: 25
Inserted 25 into the queue.

Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
Queue elements: 23 25

Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Deleted value: 23
```



```
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to insert: 56
Inserted 56 into the queue.

Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 1
Enter the value to insert: 78
Queue Overflow! Cannot insert.
```



```
Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 2
Queue Underflow! Queue is empty.

Queue Operations:
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice:
```