Queue Operations

```
#include<iostream>
using namespace std;
#include<stdlib.h>

class queue {
    int q[5], front, rear;
    public:
    queue();
    void qInsert();
    void qDelete();
    void qDisplay();
};

queue::queue() {
    front = rear = -1;
}

void queue::qInsert() {
    int num;
    if (rear == 4) {
        cout<<"Queue Overflow\n";
        return;
    }
    cout<<"Enter the number to be inserted: ";
    cin>>num;
    q[++rear] = num;
    if (front == -1)
    front = 0;
}

void queue::qDelete() {
    int num;
    if (front == -1) {
        cout<<"Queue Underflow\n";
        return;
    }

    num = q[front];
    cout<<"The deleted element is: "<<num;
    if (front == rear) {
```

```
        front = rear = -1;
    } else {
        front++;
    }
}

void queue::qDisplay() {
    if (front == -1) {
        cout<<"The queue is empty\n";
        return;
    }

    for (int i = front; i <= rear; i++)
    cout<<q[i]<<endl;
}


int main() {
    queue q;
    int ch;
    do {
        cout<<"\nQUEUE MENU\n1. Push\n2. Pop\n3. Display\n4. Exit\nEnter your choice: ";
        cin>>ch;

        switch (ch) {
            case 1: q.qInsert();
                    break;
            case 2: q.qDelete();
                    break;
            case 3: q.qDisplay();
                    break;
            case 4: exit(0);
            default: cout<<"\nInvalid choice";
        }
    } while (ch > 0);

    return 0;
}
```