MULTI THREADING APPLICATION IN C++

CODE:

```
#include <iostream>
#include <thread>
#include <mutex>
#include <condition_variable>
#include <queue>
std::mutex mtx;
std::condition_variable cv;
std::queue<int> sharedQueue;
const int MAX_QUEUE_SIZE = 5;
void producer() {
    for (int i = 1; i <= 10; ++i) {
        std::unique_lock<std::mutex> lock(mtx);
        // Wait if the queue is full
        cv.wait(lock, [] { return sharedQueue.size() < MAX_QUEUE_SIZE; });
        sharedQueue.push(i);
        std::cout << "Produced: " << i << std::endl;
        lock.unlock();
        cv.notify_all();  // Notify the consumer that the queue is not empty
    }
}
void consumer() {
    for (int i = 1; i <= 10; ++i) {
        std::unique_lock<std::mutex> lock(mtx);
        // Wait if the queue is empty
        cv.wait(lock, [] { return !sharedQueue.empty(); });
        int item = sharedQueue.front();
        sharedQueue.pop();
        std::cout << "Consumed: " << item << std::endl;
        lock.unlock();
        cv.notify_all();  // Notify the producer that the queue is not full
    }
}
```

```
int main() {
    std::thread producerThread(producer);
    std::thread consumerThread(consumer);
    producerThread.join();
    consumerThread.join();
    return 0;
}
```