```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#define BUFFER_SIZE 5
sem_t empty, full, mutex;
int buffer[BUFFER_SIZE];
int in = 0, out = 0;
void *producer(void *arg) {
  int item = 1;
  while (1) {
    sem_wait(&empty);
    sem_wait(&mutex);
    buffer[in] = item;
    printf("Producer produced item: %d\n", item);
    in = (in + 1) % BUFFER_SIZE;
    sem_post(&mutex);
    sem_post(&full);
    item++;
  }
}
void *consumer(void *arg) {
  while (1) {
    sem_wait(&full);
```

```
sem_wait(&mutex);
    int item = buffer[out];
    printf("Consumer consumed item: %d\n", item);
    out = (out + 1) % BUFFER_SIZE;
    sem_post(&mutex);
    sem_post(&empty);
  }
}
int main() {
  pthread_t producer_thread, consumer_thread;
  sem_init(&empty, 0, BUFFER_SIZE);
  sem_init(&full, 0, 0);
  sem_init(&mutex, 0, 1);
  pthread_create(&producer_thread, NULL, producer, NULL);
  pthread_create(&consumer_thread, NULL, consumer, NULL);
  pthread_join(producer_thread, NULL);
  pthread_join(consumer_thread, NULL);
  sem_destroy(&empty);
  sem_destroy(&full);
  sem_destroy(&mutex);
  return 0;
}
```

