Producer-Consumer Problem

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

#define BUFFER_SIZE 5

int buffer[BUFFER_SIZE];
int in = 0;
int out = 0;
int produced_count = 0;
int consumed_count = 0;

int max_items;

pthread_mutex_t mutex;
pthread_cond_t full;
pthread_cond_t empty;

void* producer(void* arg) {
   int item = 1;

   while (produced_count < max_items) {
      pthread_mutex_lock(&mutex);

      while (((in + 1) % BUFFER_SIZE) == out) {
         pthread_cond_wait(&empty, &mutex);
      }

      buffer[in] = item;
      printf("Produced: %d\n", item);
      item++;
      in = (in + 1) % BUFFER_SIZE;

      produced_count++;

      pthread_cond_signal(&full);
      pthread_mutex_unlock(&mutex);
   }

   pthread_exit(NULL);
}
```

```c
void* consumer(void* arg) {
   while (consumed_count < max_items) {
      pthread_mutex_lock(&mutex);

      while (in == out) {
         pthread_cond_wait(&full, &mutex);
      }

      int item = buffer[out];
      printf("Consumed: %d\n", item);
      out = (out + 1) % BUFFER_SIZE;

      consumed_count++;

      pthread_cond_signal(&empty);
      pthread_mutex_unlock(&mutex);
   }

   pthread_exit(NULL);
}

int main() {
   pthread_t producerThread, consumerThread;


   printf("Enter the number of items to produce and consume: ");
   scanf("%d", &max_items);

   pthread_mutex_init(&mutex, NULL);
   pthread_cond_init(&full, NULL);
   pthread_cond_init(&empty, NULL);

   pthread_create(&producerThread, NULL, producer, NULL);
   pthread_create(&consumerThread, NULL, consumer, NULL);

   pthread_join(producerThread, NULL);
   pthread_join(consumerThread, NULL);

   pthread_mutex_destroy(&mutex);
   pthread_cond_destroy(&full);
   pthread_cond_destroy(&empty);

   return 0;
}
```

Output :

```
Enter the number of items to produce and consume: 6
Produced: 1
Produced: 2
Produced: 3
Produced: 4
Consumed: 1
Consumed: 2
Consumed: 3
Consumed: 4
Produced: 5
Produced: 6
Consumed: 5
Consumed: 6
```