Code:

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

sem_t wrt;
pthread_mutex_t mutex;
int cnt = 1;
int numreader = 0;

void *writer(void *wno)
{
   sem_wait(&wrt);
   cnt = cnt*2;
   printf("Writer %d modified cnt to %d\n",(*((int *)wno)),cnt);
   sem_post(&wrt);

}
void *reader(void *rno)
{
   // Reader acquire the lock before modifying numreader
   pthread_mutex_lock(&mutex);
   numreader++;
   if(numreader == 1) {
      sem_wait(&wrt); // If this id the first reader, then it will block the writer
   }
   pthread_mutex_unlock(&mutex);
   // Reading Section
   printf("Reader %d: read cnt as %d\n",*((int *)rno),cnt);

   // Reader acquire the lock before modifying numreader
   pthread_mutex_lock(&mutex);
   numreader--;
   if(numreader == 0) {
      sem_post(&wrt); // If this is the last reader, it will wake up the writer.
   }
   pthread_mutex_unlock(&mutex);
}

int main()
{

   pthread_t read[10],write[5];
   pthread_mutex_init(&mutex, NULL);
   sem_init(&wrt,0,1);

   int a[10] = {1,2,3,4,5,6,7,8,9,10}; //Just used for numbering the producer and consumer

   for(int i = 0; i < 10; i++) {
      pthread_create(&read[i], NULL,  reader, (void *)&a[i]);
   }
   for(int i = 0; i < 5; i++) {
      pthread_create(&write[i], NULL, writer, (void *)&a[i]);
   }
```

```
    for(int i = 0; i < 10; i++) {
        pthread_join(read[i], NULL);
    }
    for(int i = 0; i < 5; i++) {
        pthread_join(write[i], NULL);
    }

    pthread_mutex_destroy(&mutex);
    sem_destroy(&wrt);

    return 0;

}
```

Output:

```
@somesh4545 ➜ /workspaces/TE-Labs/OSL (main) $ gcc -pthread 4-b.c && ./a.out
Reader 1: read cnt as 1
Reader 2: read cnt as 1
Reader 3: read cnt as 1
Reader 4: read cnt as 1
Reader 5: read cnt as 1
Reader 8: read cnt as 1
Reader 9: read cnt as 1
Reader 7: read cnt as 1
Reader 10: read cnt as 1
Writer 1 modified cnt to 2
Writer 2 modified cnt to 4
Writer 3 modified cnt to 8
Writer 4 modified cnt to 16
Writer 5 modified cnt to 32
Reader 6: read cnt as 32
```