```
#include<stdio.h>
#include<stdlib.h>
#include<sys/ipc.h>
#include<sys/shm.h>
#include<pthread.h>
#include<semaphore.h>

key_t key;
int shmid,size,in=0,out=0;
sem_t full;
sem_t empty;
pthread_mutex_t mutex;

void producer()
{
       for(int i=0;i<size;i++)
   {
              int *buffer = (int *)shmat(shmid, (void*) 0, 0);
              int item=rand()%10;

              sem_wait(&empty);
              pthread_mutex_lock(&mutex);

              buffer[in]=item;
              printf("Producer produced %d in buffer at %d\n", item, in);
              in=(in+1)%size;

              pthread_mutex_unlock(&mutex);
              sem_post(&full);

              shmdt(buffer);
      }
}
void consumer()
{
      for(int i=0;i<size;i++)
  {
              int *buffer=(int *)shmat(shmid, (void*) 0, 0);

              sem_wait(&full);
              pthread_mutex_lock(&mutex);

              int item=buffer[out];
              buffer[out]=0;
              printf("Consumer consumed %d from buffer at %d\n",item,out);
              out=(out+1)%size;

              pthread_mutex_unlock(&mutex);
              sem_post(&empty);

              shmdt(buffer);
      }
}
```

```
void main()
{
    key=ftok("shmfile",65);
    shmid=shmget(key,1024,0666|IPC_CREAT);

    printf("Enter buffer size:");
    scanf("%d",&size);
    printf("\n");

    sem_init(&full,0,0);
    sem_init(&empty,0,size);
    pthread_mutex_init(&mutex,NULL);

    pthread_t ptid, ctid;

    pthread_create(&ptid,NULL, (void*) producer, NULL);
    pthread_create(&ctid,NULL, (void*) consumer, NULL);

    pthread_join(ptid, NULL);
    pthread_join(ctid, NULL);

    sem_destroy(&full);
    sem_destroy(&empty);
    pthread_mutex_destroy(&mutex);
    shmctl(shmid, IPC_RMID, NULL);
}
```