```
#define BUFFER_SIZE 10
#include<stdlib.h>
#include<stdio.h>
#include<pthread.h>
#include<semaphore.h>
#define TRUE 1
pthread_mutex_t mutex; //The mutex lock
sem_t full,empty;//semaphores
int buffer[BUFFER_SIZE];
int counter; //buffer counter

pthread_t tid; //ThreadID

void *consumer(void *param);
void *producer(void *param); //The producer thread
void insert_item(int);
int remove_item();
void initialize()
{
        pthread_mutex_init(&mutex,NULL); //Create the mutex lock
        sem_init(&full,0,0);//Initialize full semaphore to 0
        sem_init(&empty,0,BUFFER_SIZE); //Initialize empty semaphore to BUFFER_SIZE
        counter=0;
}


/*Produce Thread*/


void *producer(void *param)
{
        int item;
        int waittime;                       //Next item is produced after waittime
        waittime=rand()%5;
        sleep(waittime);
        item=rand()%10;            //Generate an item to insert in the buffer
        sem_wait(&empty);                   //wait for queue to become empty,using
semaphore
        pthread_mutex_lock(&mutex);       //Acquire the buffer
        printf("Producer produced %d\n",item);
        insert_item(item);                  //Insert in the queue
        pthread_mutex_unlock(&mutex);     //Release the buffer
        sem_post(&full);               //Signal the semaphore full
}


/*Consumer thread */


void *consumer(void *param)
{
        int item;
        int waittime;                       //Next item is consumed after waittime
        waittime=rand() %3;
        sleep(waittime);
        sem_wait(&full);          //wait if the buffer is full
        pthread_mutex_lock(&mutex);//Acquire the buffer
        item=remove_item();
        printf("Consumer Consumed:%d\n",item);
        pthread_mutex_unlock(&mutex);     //Release the buffer
        sem_post(&empty);                   //Signal empty
}
```

```
//Add item to buffer


void insert_item(int item)
{
        buffer[counter++]=item;
}

//Remove an item from the buffer

int remove_item()
{
        return(buffer[--counter]);
}

int main()
{

        int n1;//No. of producers
        int n2;//No. of consumers
        int i;
        printf("Enter no.of producers:\n");
        scanf("%d",&n1);
        printf("Enter no.of consumer:\n");
        scanf("%d",&n2);
        initialize();

        //Create producers threads

        for(i=0;i<n1;i++)
        pthread_create(&tid,NULL,producer,NULL);

        //Create consumers threads

        for(i=0;i<n2;i++)
        pthread_create(&tid,NULL,consumer,NULL);
        sleep(50);
        exit(0);
}
```