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

sem_t mutex; sem_t empty; sem_t full;
int buffer[8];
pthread_t p[5]; pthread_t c[5];

void PRODUCER(int *p)
{       int a[10],i=0,n=*(int*)p;
        while(i<=5)
        {       sem_wait(&empty);
                sem_wait(&mutex);
                a[i]=3;
                printf("Producer %d Produced Item %d\n",n,i);
                sleep(1);
                i++;
                buffer[i]=a[i];
                sem_post(&mutex);
                sem_post(&full);    }       }
```

```
void CONSUMER(void *p)
{       int b[10],i=0,n=*(int*)p;
        while(i<=5)
        {       sem_wait(&full);
                sem_wait(&mutex);
                printf("Consumer %d Consumes Item %d\n",n,i);
                sleep(1);
                b[i]=buffer[i];
                i++;
                sem_post(&mutex);
                sem_post(&empty);       }       }

void main()
{       int n;
        sem_init(&mutex,0,1);
        sem_init(&empty,0,5);
        sem_init(&full,0,0);
        for(n=0;n<5;n++)
        {       pthread_create(&p[n],0,(void *)PRODUCER,(void *)&n);
                pthread_create(&c[n],0,(void *)CONSUMER,(void *)&n);   }
                while(1);       }
```