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

 struct pthreading {
    int f,buff[5];
sem_t mutex,full,empty;
    int r;
    };

void *producer(void *arguments){
int i;
struct pthreading  *args=(struct pthreading *)arguments;
for(i=0;i<5;i++){
sem_wait(&args -> empty);
sem_wait(&args -> mutex);
printf("produced %d",i);
args -> buff[(++args -> r)%5]=i;
sleep(1);
sem_post(&args -> mutex);
sem_post(&args -> full);
 }
}


 void *consumer(void *arguments){
    int item,i;
struct pthreading  *args=(struct pthreading *)arguments;
```

```
for(i=0;i<5;i++){
sem_wait(&args -> full);
sem_wait(&args -> mutex);
printf("consumed %d",item);
item=args -> buff[(++args -> f)%5];
sleep(1);
sem_post(&args -> mutex);
sem_post(&args -> empty);
 }
void Sem_init(sem_t *sem, int pshared, unsigned int value);

}


      int main() {

        pthread_t d1,d2;
       struct pthreading *a;
              sem_init(&(a -> mutex),0 ,1);
           sem_init(&(a -> full),0 ,1);
          sem_init(&(a -> empty),0,5);
             pthread_create(&d1,NULL,&producer,NULL);
             pthread_create(&d1,NULL,consumer,NULL);
             pthread_join(d1,NULL);
          pthread_join(d2,NULL);
         }
```