Application to demonstrate process/thread synchronization using semaphores and mutex. Implement, reader-writer or producer-consumer.

1) Reader-writer

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

int count=0,rcount=0;
sem_t mutex,wr;

void* writer(void *p){
int* i =(int*)p;
sem_wait(&wr);
printf("\nWriter %d writes page number %d",*i,++count);
sem_post(&wr);
}
void* reader(void* p){
int* i =(int*)p;
sem_wait(&mutex);
rcount++;
if(rcount==1)
sem_wait(&wr);
sem_post(&mutex);
printf("\nReader %d reads page number %d ",*i,count);
sem_wait(&mutex);
rcount--;
if(rcount==0)
sem_post(&wr);
sem_post(&mutex);
}
int main(){
sem_init(&mutex,0,1);
sem_init(&wr,0,1); int a[6]={1,2,3,1,2,3};
pthread_t p[6];
for(int i=0;i<3;i++) pthread_create(&p[i],NULL,writer,&a[i]);
for(int i=3;i<6;i++) pthread_create(&p[i],NULL,reader,&a[i]);
for(int i=0;i<6;i++) pthread_join(p[i],NULL);
}
```

Producer consumer:

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

```
#include<stdlib.h>
#define buffersize 10
pthread_mutex_t mutex;
pthread_t tidP[20],tidC[20];
sem_t full,empty;
int counter;
int buffer[buffersize];

void initialize()
{
        pthread_mutex_init(&mutex,NULL);
        sem_init(&full,1,0);
        sem_init(&empty,1,buffersize);
        counter=0;
}

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

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

void * producer (void * param)
{
        int waittime,item,i;
        item=rand()%5;
        waittime=rand()%5;
        sem_wait(&empty);
        pthread_mutex_lock(&mutex);
        printf("\nProducer has produced item: %d\n",item);
        write(item);
        pthread_mutex_unlock(&mutex);
        sem_post(&full);
}

void * consumer (void * param)
{
        int waittime,item;
        waittime=rand()%5;
        sem_wait(&full);
        pthread_mutex_lock(&mutex);
        item=read();
        printf("\nConsumer has consumed item: %d\n",item);
        pthread_mutex_unlock(&mutex);
        sem_post(&empty);
}

int main()
{
        int n1,n2,i;
        initialize();
        printf("\nEnter the no of producers: ");
        scanf("%d",&n1);
        printf("\nEnter the no of consumers: ");
        scanf("%d",&n2);
        for(i=0;i<n1;i++)
                pthread_create(&tidP[i],NULL,producer,NULL);
        for(i=0;i<n2;i++)
                pthread_create(&tidC[i],NULL,consumer,NULL);
        for(i=0;i<n1;i++)
                pthread_join(tidP[i],NULL);
        for(i=0;i<n2;i++)
                pthread_join(tidC[i],NULL);

        //sleep(5);
        exit(0);
}
```